

Считаем, что инфинитезимальная матрица $\Lambda(s)$ исследуемого процесса $\xi(t)$ непрерывна на $[s_0, t_0)$ и что для всех s таких, что $s_0 < s < t < t_0$, ее диагональные элементы удовлетворяют условию

$$|\lambda_{ii}(s)| \leq c(t) < \infty.$$

В этом случае исходный процесс $\xi(t)$ с матрицей $\Lambda(s)$, содержащей бесконечное число строк и столбцов, на любом отрезке $[s_0, t]$, $t < t_0$ можно с любой степенью точности аппроксимировать процессом $\hat{\xi}(t)$ с конечным числом состояний n . Инфинитезимальная матрица $\hat{\Lambda}(t)$ этого процесса есть левый верхний матричный блок матрицы $\Lambda(t)$, подвергнутой некоторой коррекции. Эта коррекция состоит в замене $\lambda_{ii}(t)$ величиной $\hat{\lambda}_{ii}(t) = \lambda_{ii}(t) + \Delta_i(t)$, где $\Delta_i(t) = \sum_{j=n+1}^{\infty} \lambda_{ij}(t)$, $i = 1, 2, \dots, n$.

Известно, что такая аппроксимация возможна при соответствующем выборе n для любого однородного процесса с инфинитезимальной матрицей $\Lambda(s)$, являющейся значением матрицы-функции $\Lambda(s)$ при любом фиксированном $s \in [s_0, t] \subset [s_0, t_0)$. Отсюда нетрудно получить, что существует n такое, что указанная выше аппроксимация исходного процесса $\xi(t)$ может быть реализована левым верхним диагональным блоком $\hat{\Lambda}(s)$ матрицы $\Lambda(s)$, имеющим для любого $s \in [s_0, t]$ одну и ту же размерность $n \times n$.

Очевидно, что выбор n , определяющий размерность матрицы $\hat{\Lambda}(s)$, зависит от точности аппроксимации исходного процесса с матрицей $\Lambda(s)$ процессом с матрицей $\hat{\Lambda}(s)$, содержащим n состояний.

Понятно, что с ростом n улучшается точность аппроксимации и что существует такая последовательность чисел $n_1, n_2, \dots, n_m, \dots$ ($n_m \rightarrow \infty, m \rightarrow \infty$),

что аппроксимирующие процессы с числом состояний $n_1, n_2, \dots, n_m, \dots$, построенные по тому же правилу, что и процесс $\hat{\xi}(t)$, при неограниченном росте n_m будут давать погрешность в аппроксимации, стремящуюся к нулю. Поскольку для каждого из аппроксимирующих процессов имеет место соотношение (2) [1], то оно же будет иметь место и для предельного процесса $\xi(t)$.

Отметим, что фокусировка и стабилизация марковского процесса могут быть реализованы за сколь угодно малый промежуток времени путем сообщения инфинитезимальной матрице $\Lambda(t)$ сильных возмущений, выбранных специальным образом.

Литература: 1. Дикарев В.А. Точки фокусировки и теоремы о существовании предельных вероятностей. Харьков, 1995, 9 с. Деп. в ГНТБ Украины 28.02.95 № 526-Ук95. 2. Розанов Ю.А. Теория вероятностей, случайные процессы и математическая статистика: Учебник для вузов. М.: Наука, 1985. 320 с.

Поступила в редколлегию 16.09.99

Рецензент: д-р техн. наук Гиль Н.И.

Басманов Алексей Евгеньевич, аспирант кафедры ПМ ХТУРЭ. Научная специальность: вычислительная математика. Научные интересы: марковские процессы и их приложения. Увлечения: шахматы. Адрес: Украина, 310128, Харьков, Садовый проезд 5, кв.10, тел. 97-23-77.

Герасин Сергей Николаевич, канд. физ.-мат. наук, доцент кафедры ВМ ХТУРЭ. Научные интересы: стохастические процессы. Увлечения: лыжный спорт. Адрес: Украина, 310166, Харьков, пр. Ленина 14, тел. (0572)40-93-72, (0572)72-12-38; e-mail: hm@kture.kharkov.ua.

Дикарев Вадим Анатольевич, д-р физ.-мат. наук, профессор кафедры ПМ ХТУРЭ. Научные интересы: теория вероятностей. Увлечения: туризм. Адрес: Украина, 310166, Харьков, пр. Ленина 14, тел. (0572)40-93-72

Родзинский Анатолий Анатольевич, аспирант кафедры ПМ ХТУРЭ. Научная специальность: математическое моделирование и вычислительные методы. Научные интересы: математические модели в технике и экономике. Увлечения: компьютеры. Адрес: Украина, 310166, Харьков, пр. Ленина 14, тел. (0572)40-93-72.

УДК 681.5.015(07)

ДИНАМИЧЕСКОЕ МОДЕЛИРОВАНИЕ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ ВИЗУАЛЬНЫМИ СРЕДСТВАМИ

КУЗЬМЕНКО В.М., ГНЕННЫЙ А.М.

Описывается применение идеографического подхода к языку GPSS, что позволило разработать систему визуального моделирования систем массового обслуживания Visual QSDM. Использование данного программного комплекса позволяет проводить моделирование достаточно сложных систем массового обслуживания без глубокого знания языков моделирования.

Решение смежных задач автоматизации исследования, управления и проектирования больших систем в настоящее время не может выполняться без использования методов имитационного моделирования. Для автоматизации динамического моделирова-

ния разработаны и используются специализированные пакеты программ, работа которых базируется на интерпретации программ-моделей, записанных посредством специального проблемно-ориентированного языка моделирования: GPSS, СЛАМ-2 и др. [1,2]. Однако создание программ-моделей требует взаимодействия специалистов различных областей: программистов, предметников, постановщиков и др. В связи с развитием самих моделируемых систем, необходимостью рассмотрения только отдельных аспектов этих систем программы-модели должны также изменяться, что требует постоянного взаимодействия этих типов специалистов. Альтернативой традиционной технологии создания имитационных моделей является идеографический подход [3], который позволяет создавать и модифицировать модель специалисту предметной области — непрограммисту.

Основой идеографического подхода служит использование идеографических элементов Id_i ($i=1, \dots, N$), которые, с одной стороны, являются визуальным изображением объектов моделируемой системы, а с другой — модулем, адекватно моделирующим этот объект. Идеографическая сеть Is , которая моделирует

техническую систему, представляется в этом случае ориентированным графом $G = \langle V, W \rangle$, где вершинам $v_j \in V$ соответствует идеограмма Id_j , а дугам $w_j \in W$ — переменные P , описывающие моделируемую систему.

Каждый идеографический элемент $Id_i \in Id$ обеспечивает вычисление функции $Y = f(X)$, где X — входные данные, Y — выходные результаты, которые имеют их области определения, соответственно A и B . В общем случае идеографический элемент $Id_i \in Id$ описывается планом отображения $\Pi_o(Id_i)$ — графическое изображение преобразования множества переменных $P = \{X, Y, S\}$, которое состоит из переменных входа X , выхода Y и внутренних переменных S , и планом содержания $\Pi_c(Id_i)$ — математическое выражение, которое задает преобразование множества переменных P . Количество и степень отношения между входными X и выходными Y переменными идеографического элемента Id_i определяется аксиомой вычислений $e(Id_i)$, а его количество входов — порогом вычислений $p(Id_i)$.

Множество идеографических элементов $Id_i \in Id$ ($i = 1, \dots, N$), которые описывают имитационную модель M системы, не содержит двух идеографических элементов, имеющих одинаковые план отображения, содержания и множество переменных P .

Комплексирование идеографических элементов Id_i, Id_j в идеографическую сеть Js осуществляется на основании следующих правил:

1. Количество соединений, объединяющих два смежных идеографических элемента (Id_i, Id_j) , не должно превышать их порог вычисления: $p = \min(Id_i, Id_j)$ $i = 1, \dots, N, j = 1, \dots, N$.

2. Объединение идеографических элементов (Id_i, Id_j) осуществляется на основе планов отображения $\Pi_o(Id_i), \Pi_o(Id_j)$, а проверка правильности — на основе планов содержания $\Pi_c(Id_i), \Pi_c(Id_j)$.

3. Выходные результаты идеографического элемента Id_i являются входными данными другого идеографического элемента Id_j , если они имеют совместную аксиому вычисления $e(Id_i, Id_j)$.

4. Идеографические элементы $Id_i \in Id$ в сети Js объединяются с учетом ориентации дуг.

5. Выходные результаты $Y(Id_i)$ идеографического элемента не могут быть входными данными $X(Id_i)$ этого же идеографического элемента.

Данный подход должен базироваться на использовании специального языка моделирования. При этом можно использовать существующие специальные языки моделирования или разрабатывать новые. Применение идеографического подхода к языку GPSS позволило разработать систему визуального моделирования систем массового обслуживания Visual QSDM. С использованием Visual QSDM методику имитационного моделирования условно можно разбить на следующие этапы:

1. Определение структуры моделируемой системы, декомпозиция системы на объекты в соответствии с требованиями языка моделирования, определение связей между ними.

2. Размещение идеографических элементов в соответствии со структурой модели.

3. Установка межэлементных связей, характеризующих маршруты перемещения транзактов в модели;

4. Установка параметров элементов схемы (их свойств).

5. Моделирование системы в соответствии с построенной схемой.

6. Получение результатов статистического анализа в виде графиков и гистограмм.

7. Анализ результатов, корректировка топологии модели и (или) свойств входящих в нее элементов, если это необходимо.

Объектная модель системы Visual QSDM приведена на рис. 1. К основным классам, составляющим ядро системы моделирования, относятся:

CBlock — абстрактный базовый класс;

CPropertyBag — класс манипулирования набором свойств;

CProp — класс реализации отдельного свойства элемента;

CTransact — класс транзактов;

CConnect — класс связей между элементами;

CEditable — класс управления отображением и редактированием;

CNoEnter, COneEnter, CNoExit, COneExit — классы-примеси, реализующие типичное поведение элементов;

CQueued — класс-примесь, реализующий обработку очередей транзактов;

CGenerate, CTerminate, CAdvance, CSeize, CRelease, CTable — классы типичных элементов.

Основные особенности системы Visual QSDM:

1. Базовая система — пакет динамического моделирования GPSS, что позволяет использовать ее специалистам, знакомым с GPSS.

2. Структура моделируемой системы строится из типовых элементов.

3. Каждый типовой элемент имеет множество входов и выходов.

4. Связи входов и выходов определяют перемещение потоков транзактов.

5. Есть возможность ограничить как количество транзактов, так и время функционирования системы.

6. Присвоение транзактам, которые генерируются различными источниками, различных цветов позволяет значительно упростить анализ поведения системы.

7. Результаты моделирования представляются в наглядной форме (в виде графиков, гистограмм и др.).

8. В результате моделирования могут быть получены статистические оценки поведения системы и отдельных ее элементов.

9. Наглядность представления структуры системы достигается путем цветового выделения функционально схожих элементов.

10. Перестройка структуры системы путем перетаскивания элементов и настройки связей позволяет сократить время подготовки к моделированию и общее время исследования системы:

Система функционирует под управлением наиболее распространенных операционных систем Windows 95, Windows NT 4.0 и Windows 98. Для разработки системы использовалась интегрированная система Microsoft Visual C++, что позволило получить максимальную производительность при моделировании.

Для построения моделей систем используется 7 разновидностей элементов, характеристики которых приведены в табл. 1.

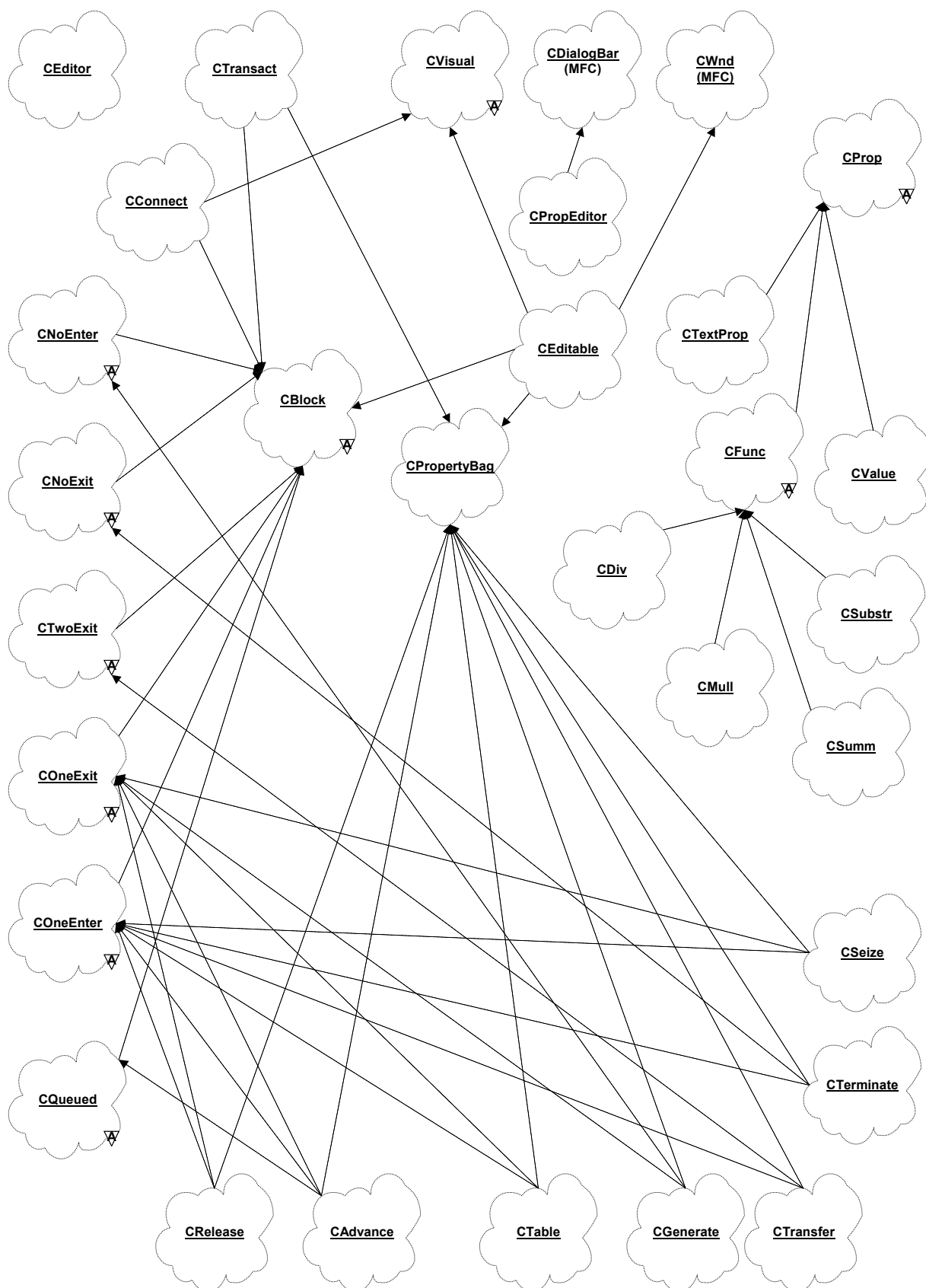


Рис. 1. Диаграмма иерархии классов

Кроме указанных свойств, каждый элемент имеет имя и обработчик для события, наступающего при входе очередного транзакта в элемент. В обработчиках событий реализуются арифметические, логические операции, а также операции присваивания свойствам элементов новых значений. Например : *on_enter:"user_param1:=transact.priority+1"*.

Для последующих реализаций системы моделирования, для реализации обработчиков событий предполагается использовать встроенный в Internet Explorer (а значит и в Windows 95) интерпретатор языка JavaScript.

Рассмотрим простейшую задачу имитационного моделирования: моделирование мойки автомобилей. Процесс мытья одного автомобиля занимает 10 ± 5

Таблица 1

Название	Назначение	Основные свойства
Вход	Запускает транзакты в систему	Среднее время поступления транзакта в систему, приоритет транзактов, количество входящих транзактов
Выход	Удаляет входящие транзакты из системы	Значение, на которое уменьшается глобальный счетчик транзактов
Задержка	Моделирует задержку транзакта при его обработке	Среднее время задержки
Ветвление	Меняет направление потока транзактов	Условия ветвления
Таблица	Собирает статистические данные	Имена регистрируемых параметров и их описание
Занять	Занимает указанное устройство	Имя занимаемого устройства
Освободить	Освобождает указанное устройство	Имя освобождаемого устройства

минут. Мойка оборудована стоянкой на 2 автомобиля, которые заезжают на мойку через 11 ± 5 минут. При отсутствии свободного места на стоянке автомо-

биль покидает систему. Определить количество не обслуженных в течение смены (8 ч.) автомобилей.

Построим модель данной системы и проведем моделирование ее поведения. Для этого спроектируем структуру системы из типовых элементов и установим маршруты перемещения транзактов. Полученная структура представлена на рис. 2.

Устанавливаем свойства блоков. Свойства элементов приведены в табл. 2.

Таблица 2

Элемент	Свойство	Значение
Вход	Count	0
	Interval	12
	Interval_v	4
	Name	Enter_1
Ветвление	To_exit_1	Moika.size==0
	Name	Trans_1
Занять	Size	3
	Name	Moika
	Enter_size	1
Задержка	Timeout	10
	Timeout_v	3
Освободить	Size	1
	Device	Moika
Выход №1	Count	1
Выход №2	Count	1

В результате моделирования получим количество автомобилей, не прошедших мойку, и диаграмму времени прохождения транзактов через модель. Диаграмма приведена на рис. 2.

Этот простейший пример демонстрирует простоту и наглядность моделирования систем массового обслуживания при помощи визуальной среды.

Использование данного программного комплекса позволяет проводить моделирование достаточно сложных систем массового обслуживания без глубокого знания языков моделирования. Для плодотворного

применения данной системы необходимы минимальные навыки работы на ЭВМ и минимальное представление о системе GPSS, т.е. понимание основных принципов ее функционирования. Пользователь ограждается от досконального изучения языка моделирования и длительного программирования на этом языке.

Анализ результатов разработки системы Visual QSDM выявил следующие пути дальнейшего усовершенствования подобных средств:

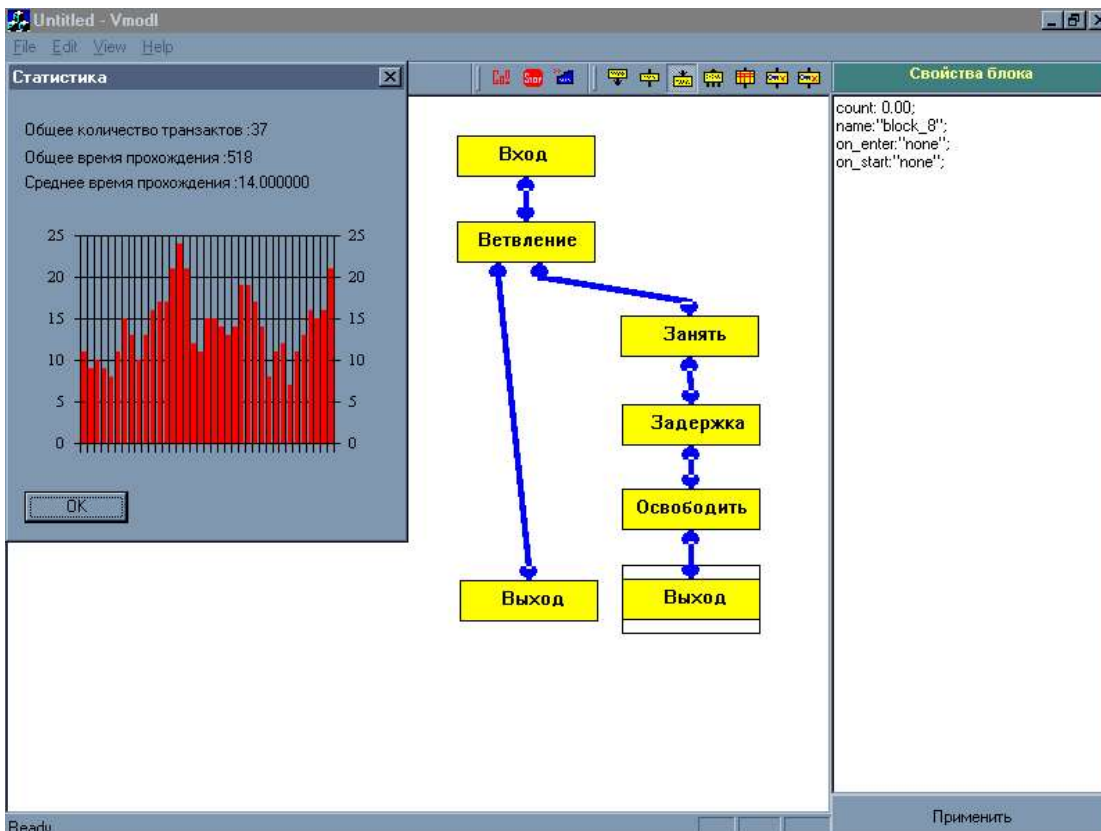


Рис.2. Пример отображения результатов моделирования

1. Более сильное абстрагирование от базового языка моделирования.

2. Повышение наглядности процесса моделирования системы.

3. Приближение системы обозначений, применяющихся в процессе построения модели, к обозначениям, применяемым в конкретной предметной области.

4. Расширение функциональных возможностей моделирования.

5. Усовершенствование математического аппарата, используемого в процессе моделирования.

6. Генерация кода на языке моделирования позволяет переносить модель в другие среды динамического моделирования.

Литература. 1. Солодовников И.В. Языки, программное обеспечение и организация систем имитационного моделирования. М.: Машиностроение, 1982. 149с. 2. Моделирование систем сбора и обработки данных / Под ред.

В.И. Маковицкого. М.: Наука, 1983. 129с. 3. Кузьменко В.М. Имитационное моделирование в решении задач оптимизации проектирования технологического оборудования // 3-я Всесоюзная конференция "Проектирование автоматизированных систем контроля и управления сложными объектами (г. Туапсе)". Тез. докл. Харьков, 1988. С 32.

Поступила в редколлегию 21.07.99

Рецензент: д-р техн. наук, проф. Нефедов Л.И.

Кузьменко Виктор Михайлович, канд. техн. наук, доцент, профессор кафедры системотехники ХТУРЭ. Научные интересы: математическое и имитационное моделирование технологических процессов. Адрес: Украина, 310726, Харьков, пр. Ленина 14, тел. 40-93-06, 19-76-36.

Гненный Андрей Михайлович, студент ХТУРЭ. Научные интересы: процессы и методы управления в автоматизированных системах. Адрес: Украина, 310726, Харьков, пр. Ленина 14, каф. системотехники, тел. 40-93-06, 49-52-56.

УДК 62.522

ПОЗИЦИОННАЯ СТРУКТУРА ДИСКРЕТНО-АНАЛОГОВОГО УПРАВЛЕНИЯ ПНЕВМОАГРЕГАТОВ

ВУРЬЕ Б. А.

Предлагается принципиально новая схемная организация пневмоагрегатов, позволяющая реализовать дискретно-аналоговое управление, а также метод синтеза данного класса систем.

В настоящее время при проектировании пневмоагрегатов технологического оборудования используются различные структурные схемы реализации.

Выбор той или иной структурной схемы может быть осуществлен в результате глубокого анализа технологического процесса, исходя из конкретных требований заказчика.

В настоящей статье приводится обобщенная схема структурной организации, позволяющая реализовать любой процесс управления пневмоагрегатов, имея минимальный набор структурных блоков. При этом указанные ниже отличия от существующих схемных решений позволяют получить минимизированную реализацию в заданном базисе устройств.

На рисунке представлена позиционная структурная реализация дискретно-аналогового управления пневмоагрегатов: 1 — блок совпадений включений, выполняющий функции конъюнкции; 2 — блок разделения отключений, выполняющий функции дизъюнкции; 3 — командоаппарат с унитарным кодированием внутренних состояний, причем верхний триггер имеет два инверсных выхода для реализации двух внутренних состояний пневмоагрегата. Каждый предыдущий триггер отключается сигналом с выхода последующего. Входные сигналы X с блока ИУ4 дискретного типа формируются от устройств, контролирующих положение данного ИУ. Выходные сигналы Z формируются в блоке 2, ИУ5 содержат следящие распределители. Пневмодвигатели в этом случае, от-

рабатывающие движения, имеют дискретный контроль своих положений, а устройства контроля вырабатывают дискретные сигналы. В ИУ6 контроль положения осуществляется датчиками, отрабатывающими аналоговые либо импульсные сигналы, которые преобразуются в дискретные, а затем в дискретную пневмосхему управления в блоке 7.

Дадим оценку сложности данной структуры, исходя из того, что каждое элементарное логическое действие принимается равным 1. Число элементов блока 1 в общем случае $(|X| \times T)$, где $|X|$ — число

