УДК 681.3

Б. Б. НЕСТЕРЕНКО, М. А. НОВОТАРСКИЙ (Институт математики НАН Украины)

ФОРМАЛЬНЫЕ СРЕДСТВА ДЛЯ МОДЕЛИРОВАНИЯ СЛОЖНЫХ ДИСКРЕТНЫХ СИСТЕМ

У роботі розглянуті принципи формального представлення дискретних систем та їх моделей. Дано визначення прямої, зворотної і взаємної подібності між системою та моделлю. Коротко описана структура формальних засобів та їх еволюція. Запропоновано нову версію мереж для моделювання складних дискретних систем з асинхронною взаємодією компонентів.

В работе рассмотрены принципы формального представления дискретных систем и их моделей. Даны определения прямого, обратного и взаимного подобия между системой и ее моделью. Кратко описана структура формальных средств и их эволюция. Предложена новая версия сетей для моделирования сложных дискретных систем с асинхронным взаимодействием компонентов.

In the paper principles of formal representation of discrete systems and their models are considered. Definitions of straightforward similarity, back similarity and bisimilarity between systems and their models are given. The structure of formal means and their evolution is briefly described. The new version of networks for modeling complex discrete systems with asynchronous interaction of components is offered.

Введение

За последние годы, как в сфере разработки новых транспортных средств, так и в сфере их эксплуатации стремительно возрос интерес к технологиям моделирования, анализа и управления сложными системами. Типичными примерами таких систем являются гибкие производственные комплексы, телекоммуникационные сети, системы параллельной обработки информации, логистические системы. Как правило, все они имеют искусственное происхождение, складываются из конечного количества ресурсов (средств производства, коммуникационных каналов, средств обработки информации), которые совместно используются несколькими пользователями (типами материалов, информационными пакетами, задачами) и вносят вклад в достижение общей цели (изготовление продукции, передача набора информационных пакетов от передатчика к приемнику, решение задачи управления сложными транспортными системами).

Общим для упомянутых систем является то, что они эволюционируют благодаря осуществлению ряда событий, связанных с активностями ресурсов (например: изготовление элемента продукции, успешная передача информационного пакета, обработка этапа задачи управления). События происходят в дискретные моменты времени, а продолжительность интервалов между событиями может быть детерминированной или носить вероятностный характер. Событием будем называть некоторое действие или спонтанное явление. События происходят мгновенно и, в зависимости от внутренних условий в системе, могут вызывать или не вызывать изменение состояний ее компонентов. Основываясь на данных абстракциях, опишем два наиболее распространенных подхода к формализации сложных дискретных систем.

Формальное представление дискретных систем

Формально дискретные системы с событиями задают кортежем из двух элементов: $(S,\to)$, где S – множество состояний, $\to\subseteq S\times S$ – бинарное отношение, которое задано на S и определяет множество переходов. Элементом множества состояний S есть множество $p=\{p_i\}_{i=1}^{N}$, где p_i – состояние i-го компонента, N – количество компонентов системы. Двумя разными состояниями системы $p,q\in S$ будем называть такие состояния, которые отличаются хотя бы одним состоянием компонента: $p_i\neq q_i$ при $p_i\in p, q_i\in q, 1\le i\le N$. Говорят, что существует переход (p,q) между состоянием p и состоянием q, если существует событие, свершение которого служит причиной изменения состояния системы p на состояние q. Если $p,q\in S$, а $(p,q)\in\to$, то факт существования перехода из состояния p к состоянию q может быть представлен в виде: $p\to q$. Системы с данным формальным описанием также называют немаркированными системами с переходами.

Для представления систем с параллельной обработкой событий применяют формальное описание в виде кортежа:

$$\left(S, M, \left\{\xrightarrow{\mu} \middle| \; \mu \in M\right\}\right),$$

где S – множество состояний, M – множество меток, $\xrightarrow{\mu} \subseteq S \times M \times S$ – тернарное отношение, которое задает переходы между состояниями модели под воздействием меток. Если

$$p,q \in S \text{ , } \mu \in M \text{ и } (p,q) \in \xrightarrow{\mu},$$

то переход из состояния p к состоянию q под воздействием метки μ обозначают $p \xrightarrow{\mu} q$. Такие системы называют маркированными системами с переходами. Маркированная система с переходами может быть сведенной к немаркированной системе с переходами при условии, что множество M состоит лишь из одного элемента.

Поскольку иерархичность является одним из важных свойств сложных систем, справедливо также обратить внимание на особенности формального описания агрегативных систем, то есть таких систем, компоненты которых сами подходят под определение системы. Компонент агрегативной сложной системы включает переходы, а потому его состояние не может быть сведенным к статическому набору параметров.

Если динамика модели состоит в эволюции от одного события к следующему, то такие модели называют ориентированными на активности. Последовательность активностей называют процессом.

В общем случае под процессом понимают последовательность естественных явлений или искусственно спроектированных изменений, направленных на достижение определенного результата. Итак, состояние сложной системы с событиями представляют процессом, состоящим из последовательности активностей, которые активируются и деактивируются под воздействием событий.

Агрегативную маркированную систему с переходами зададим кортежем

$$\left(P, M, \left\{\xrightarrow{\mu} \middle| \; \mu \in M\right\}\right),$$

где P – множество состояний компонентов, выраженных множеством процессов, M – множество меток, $\xrightarrow{\mu} \subseteq P \times M \times P$ – тернарное отношение, задающее переходы между состояниями компонентов под воздействием меток.

Динамика систем с дискретными событиями характеризуется также синхронизацией и параллелизмом. Синхронизация проявляется через согласование во времени некоторой совокупности событий, предопределяющих переход системы от предшествующего этапа ее эволюции к последующему. Например, транспортное средство может находиться в бездействующем состоянии до того момента, пока не станут доступными все условия, необходимые для продолжения движения. О наличии параллелизма в системе говорят в случае, если существует некоторое множество свободных ресурсов, допускающих одновременное использование. Например, в системе управления полетами очередная задача может быть назначена одному из свободных на данный момент диспетчеров.

Проблемы подобия моделей

Под моделированием будем понимать процесс построения адекватного отображения наиболее важных сторон дискретной системы с некоторой заданной точностью и проведение экспериментов на построенной модели для получения информации об объекте исследования [1]. Процесс адекватного отображения состоит в применении таких процедур построения модели, которые разрешают отвергнуть несущественные компоненты системы и упростить те компоненты, которые играют важную роль. При этом необходимо сохранить подобие модели объекту моделирования. Будем различать прямое, обратное и взаимное подобие между моделью и описываемой системой.

Между данной маркированной системой с переходами $(S, M, \rightarrow)$ и моделью $(S', M', \rightarrow)$ существует прямое подобие, если для некоторого бинарного отношения R на S, $R \subseteq S \times S$ существует соответствующее бинарное отношение R' на S', $R' \subseteq S' \times S'$, в котором для каждой пары состояний $p,q \in S$, $(p,q) \in R$ $\forall$ $\mu \in M$ существует такая пара состояний $p',q' \in S'$, $(p',q') \in R'$ $\forall$ $\mu' \in M'$, что для каждого перехода $p \xrightarrow{\mu} q$ существует соответствующий переход $p' \xrightarrow{\mu'} q'$.

Свойство прямого подобия, как правило, является достаточным при условии использования модели для анализа характеристик системы как объекта моделирования. Если же использовать результаты моделирования на этапе проектирования системы, то нужно быть уверенным в том, что между моделью и системой существует обратное подобие.

Между данной моделью $(S', M', \rightarrow)$ и маркированной системой с переходами $(S, M, \rightarrow)$ существует обратное подобие, если для некоторого бинарного соотношения R' на S', $R' \subseteq S' \times S'$ существует соответствующее бинарное соотношение R на S, $R \subseteq S \times S$, в котором для каждой пары состояний $p', q' \in S'$, $(p', q') \in R'$ $\forall$ $\mu' \in M'$ существует такая пара состояний $p, q \in S$, $(p, q) \in R$ $\mu \in M$, что для каждого перехода $p' \xrightarrow{\mu'} q'$ существует соответствующий переход $p \xrightarrow{\mu} q$.

Маркированная система с переходами $(S, M, \rightarrow)$ и модель $(S', M', \rightarrow)$ взаимно подобны, если существует прямое и обратное подобие между данной системой и данной моделью. Понятие взаимного подобия лежит в основе построения и применения моделей и охватывает все классы эквивалентности, которые возникают в данном случае. Основными из этих классов являются эквивалентность типа «модель-система», которую определяют с помощью прямого и обратного подобия, а также эквивалентности типа «модель-модель» и «состояние-состояние». В общем случае эквивалентность может рассматриваться как взаимное сходство поведения объектов наблюдения. Поэтому определение взаимного подобия процессов агрегативной системы

$$\left(P, M, \left\{ \xrightarrow{\mu} \middle| \ \mu \in M \right\}\right)$$

рассматривается как общий случай взаимного подобия.

Два процесса $p, q \in P$ будем называть взаимно подобными тогда и только тогда, когда существует некоторое соотношение $R \subseteq P \times P$, такое, что при условии $p, q \in R$ для всех $\mu \in M$:

1. Каждому переходу $p \xrightarrow{\mu} p'$ соответствует некоторый процесс q', для которого существует переход

$$q \xrightarrow{\mu} q' \text{ и } (p', q') \in R.$$

2. Каждому переходу $q \xrightarrow{\mu} q'$ соответствует некоторый процесс p', для которого существует переход

$$p \xrightarrow{\mu} p' \text{ и } (p', q') \in R.$$

Упомянутые подобия базируются на установлении соответствия между теми компонентами и переходами, которые принадлежат определенному соотношению R.

Формальные средства описания моделей

Модель системы для дальнейшего ее исследования должна быть определенным образом формализована. Такой подход разрешает применить методы, с помощью которых возможно понять или осознать сущность объекта моделирования. Формализм обеспечивает описание структуры заданных классов объектов и задает однозначные правила взаимодействия между ними. Структура инструментария формального описания служит основой для определения базиса сложности объектов описываемого класса. Большинство современных формальных средств описания имитационных моделей с дискретными событиями базируется на положении общей теории систем, которое гласит, что реальные системы могут функционировать по одинаковым правилам и демонстрировать одинаковое поведение даже в том случае, когда они физически не подобны. Упомянутый принцип построения модели породил целую серию формальных описаний дискретных систем (рисунок), которые, как правило, характеризуются аналитическим и графическим представлениями.

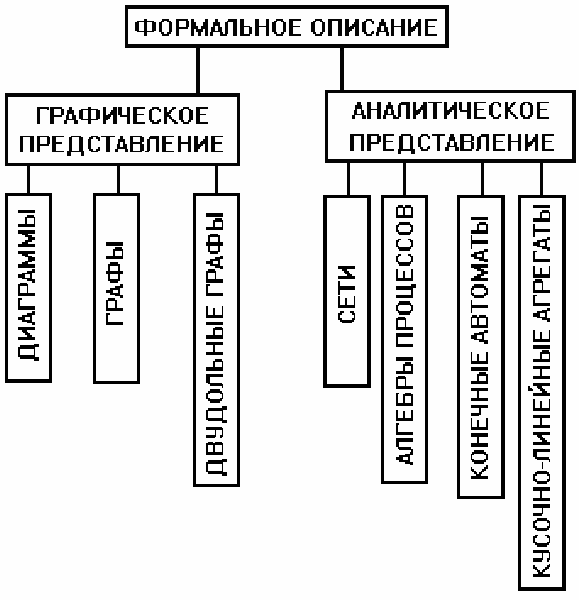

Рис. Структура формального описания моделей дискретных систем

Следует отметить, что не существует жестко установленной связи между аналитическим представлением и его графической интерпретацией. Однако аналитическая модель должна учитывать количество элементов графического представления. Поэтому графы и диаграммы ориентированы на описание систем типа $(S,\rightarrow)$, которые эволюционируют путем перехода от одного состояния к другому под воздействием событий. Принципиальное отличие двудольных графов состоит в том, что они, в отличие от обыкновенных графов и диаграмм, включают три базовых графических элемента (позиции, ребра и переходы). Это дает возможность отображать маркированные системы типа

$$\left(S,M,\left\{\xrightarrow{\mu}\middle|\mu\in M\right\}\right)$$

и агрегативные системы

$$\left(P,M,\left\{\xrightarrow{\mu}\middle|\mu\in M\right\}\right).$$

Эволюция применения двудольных графов берет свое начало от сетей Петри [2], принципиальное отличие которых от ранее известных формальных средств состоит в том, что они ориентированы на описание параллельных процессов и структур. Сети Петри – это современный формальный инструмент исследования сложных систем путем моделирования [3]. Модель в данном случае представляет в формализованном виде основные свойства систем, которые подлежат изучению. Манипулируя параметрами такой модели, можно получить новые знания о системе как объекте исследования и о характере ее связей с окружающей средой.

Существенные недостатки сетей Петри, связанные с громоздкостью графического описания модели из-за жестких правил срабатывания переходов, послужили основанием к дальнейшему совершенствованию сетевых подходов. Разнообразные модификации сетей Петри [4] позволили улучшить их характеристики при моделировании узкого круга задач. Существенной попыткой модернизировать сети Петри можно считать E-сети [5], которые позволили снизить громоздкость графического представления путем введения ряда специализированных позиций и расширения набора правил срабатывания переходов. Для моделирования сложных систем с иерархической структурой были предложены PRO-сети [6]. Основная продуктивная идея их создания состояла в стремлении отойти от жестких правил срабатывания переходов для получения гибких механизмов описания объектов моделирования.

Общим недостатком рассмотренных подходов к графическому представлению дискретных моделей является отсутствие четкого механизма моделирования процессов взаимодействия между компонентами сложной системы. В связи с этим предложена новая версия сетей – асинхронные PRO-сети (сокращенно – APRO-сети).

APRO-сети

Представим APRO-сеть в виде кортежа:

$$\Phi=\left(P,T,F,M,V\right), \tag{1}$$

где $P=\{p_i\}_{i=1}^{n}$ – конечное множество позиций; $T=\{t_j\}_{j=1}^{m}$ – конечное множество переходов; $F=P\times T\bigcup T\times P$ – множество ребер между переходами и позициями; $M=\left\{\left(p_k,\{\mu_l\}_{l=1}^{Max\_p_k}\right)\right\}_{k=1}^{n}$ – конечное множество маркировок; $V=\left(\Theta,\Psi,\Lambda\right)$ – множество глобальных переменных.

Позиции APRO-сети:

$$p_i=\{c_i,q_i\}, \tag{2}$$

где $c_i=\{\mathrm{Id}\_p_i,\varphi\_p_i,\mathrm{Cur}\_p_i,\mathrm{Max}\_\_p_i\}$ – параметры позиции, которые включают: Id_p_i – идентификатор позиции, φ_p_i – множество допустимых типов меток, Cur_p_i – текущее количество меток на позиции, Max_p_i – максимально допустимое количество меток; q_i – список меток, размещенных на данной позиции.

Переходы APRO-сети:

$$t_j=\{\chi_j,\rho_j,\pi_j\}\ , \tag{3}$$

где $\chi_i=\left\{\mathrm{Id}\_t_j,\tau\_t_j,\Delta\_t_j,\mathrm{Cur}\_t_j,\mathrm{Max}\_t_j\right\}$ – араметры перехода, которые включают: Id_t_j – идентификатор перехода, τ_t_j – локальный счетчик времени, Δ_t_j – период простоя, Cur_t_j – идентификатор текущего процесса, Max_t_j – максимально допустимое количество процессов; ρ_j – процедура активации перехода; π_j – процедура обслуживания перехода.

Ребра APRO-сети $\left(p_i,t_j\right)$ задают матрицей инцидентности H с элементами:

$$H\left(p_i,t_j\right)=\begin{cases}-1,\ \left(p_i,t_j\right)\in F,\\ +1,\ \left(p_i,t_j\right)\in F^{-1},\\ 0,\quad \left(p_i,t_j\right)\notin F,\left(p_i,t_j\right)\notin F^{-1}\ ,\end{cases} \tag{4}$$

Будем использовать ${}^{\bullet}p_i$ для обозначения входного множества переходов позиции p_i, то есть такого подмножества $\{t_j\}_{j=1}^{m}$, для которого существуют ребра от t_j в p_i при $H(p_i,t_j)>0\big|_{j=\overline{1,m}}$. Выходное множество переходов $p_i^{\bullet}$ для позиции p_i обозначается подобным образом как подмножество переходов $\{t_j\}_{j=1}^{m}$, для которых существуют ребра от p_i в t_j при $H(p_i,t_j)<0\big|_{j=\overline{1,m}}$. Сокращенно правила формирования входного и выходного множеств переходов для позиции p_i представим в виде:

$$\begin{cases} {}^{\bullet}p_i = \{ t_j | H(p_i,t_j) > 0,\, 1 \le j \le m \}, \\ p_i^{\bullet} = \{ t_j | H(p_i,t_j) < 0,\, 1 \le j \le m \}. \end{cases}$$

Аналогично формируются входные и выходные подмножества позиций для данного перехода t_j:

$$\begin{cases} {}^{\bullet}t_j = \{ p_i | H(p_i,t_j) < 0,\, 1 \le i \le n \}, \\ t_j^{\bullet} = \{ p_i | H(p_i,t_j) > 0,\, 1 \le i \le n \}. \end{cases}$$

Метки APRO-сети:

$$\mu_k = \{\lambda_k, \alpha_k\}, \tag{5}$$

где $\lambda_k = \{ \mathrm{Id}\_\mu_k, \tau\_\mu_k, \varphi\_\mu_k \}$ – параметры метки, которые включают: Id_μ_k – идентификатор метки, τ_μ_k – время создания метки, φ_μ_k – тип метки, α_k – множество атрибутов метки.

Множество меток формирует глобальную маркировку $M_{\Phi} = \left\{ \left(p_k, \{\mu_l\}_{l=1}^{Cur\_p_k} \right) \right\}_{k=1}^{n}$ с текущим количеством меток $M(p_i)$ на позиции p_i сети Φ.

Множество глобальных переменных:

$$V = (\Theta, \Psi, \Lambda), \tag{6}$$

где Θ – подмножество показателей производительности, Ψ – подмножество показателей реактивности, Λ – подмножество показателей использования.

Рассмотрим основные аспекты динамики APRO-сети, адаптировав основные положения семантики последовательных шагов [7] к предложенной версии сетей. Упомянутая семантика базируется на понятии мультимножества, формально определяемого в виде пары (X,β), где X – некоторое базовое множество элементов; $\beta : X \to N$ – функция, задающая отображение X на пространство натуральных чисел N. Количество вхождений $x \in X$ в данном случае будет определяться функцией $\beta(x) \in N$. Шагом U будем называть мультимножество переходов $U = (X,\beta)$ при $X \subset T$, $X = \{t_j\}$, активирующихся при условии выполнения данного неравенства для позиции p_i:

$$M(p_i) \ge \left| \sum_{t_j \in X} H(p_i,t_j) \cdot \beta(t_j) \right|. \tag{7}$$

Входное множество шага U объединяет все позиции переходов, входящих в данный шаг:

$$\begin{aligned} {}^{\circ}U = \{ (p_i,\phi_i) | \, & p_i \in {}^{\bullet}t_j,\, t_j \in X, \\ & \phi_i = \textstyle\sum_{t_j \in X} \beta(t_j),\, 1 \le i \le n,\, 1 \le j \le m \} \end{aligned} \tag{8}$$

Аналогично определяем выходное подмножество шага U:

$$\begin{aligned} U^{\circ} = \{ (p_i,\phi_i) | \, & p_i \in t_j^{\bullet},\, t_j \in X, \\ & \phi_i = \textstyle\sum_{t_j \in X} \beta(t_j),\, 1 \le i \le n,\, 1 \le j \le m \} \end{aligned} \tag{9}$$

При условии активации и выполнении условий срабатывания может сработать переход t_j. В результате этого происходит смена количества меток на связанной с данным переходом позиции:

$$M'(p_i) := M(p_i) + H(p_i,t_j) \tag{10}$$

В соответствии с (4) для входной позиции p_i перехода t_j $H(p_i,t_j) = -1$, а для выходной позиции p_k данного перехода $H(p_k,t_j) = 1$. Поэтому при срабатывании перехода t_j общее количество меток на входных позициях уменьшается на единицу, а общее количество меток на выходных позициях увеличивается на единицу. В результате срабатывания шага U получим изменение маркирования с $M(U)$ на $M'(U)$ в соответствии с выражением:

$$\begin{aligned} M'(U) := M(U) + & \textstyle\sum_{p_i \in {}^{\circ}U} H(p_i,t_j) + \\ & + \textstyle\sum_{p_k \in U^{\circ}} H(p_k,t_j) \end{aligned} \tag{11}$$

Такое изменение маркирования в результате срабатывания шага U обозначим $M|U\rangle M'$. Тогда эволюция маркирований σ под действием последовательности шагов $\sigma = U_1 \cdots U_k$ может быть записана:

$$M|\sigma\rangle M' = M|U_1\rangle M_1 \cdots M_{k-1}|U_k\rangle M'. \tag{12}$$

Действия перехода по перемещению меток образуют цикл активности перехода. Типичный цикл активности состоит из трех этапов: активация перехода, работа перехода и деактивация перехода.

Активация происходит только при выполнении необходимого и достаточного условий запуска. Для APRO-сети необходимое условие активации всех переходов шага U имеет следующий вид:

$$|U| = \left| \left\{ \lfloor t_j \rfloor \Big| \left|{}^{\bullet}t_j\right| \geq 1, \sum_{i=1}^{\left|{}^{\bullet}t_j\right|} Cur\_p_i \geq 1 \right\}_{t_j \in U} \right|, \quad (13)$$

где $\lfloor t_j \rfloor$ – переход t_j в состоянии активации.

Достаточное условие активации переходов шага U:

$$|U| = \left| \left\{ \lfloor t_j \rfloor \;\Big|\; |A_j| \leq Max \right\}_{t_j \in U} \right|, \quad (14)$$

где A_j – частично-упорядоченное множество активностей, соответствующих переходу t_j

Проверку упомянутых условий активации выполняет процедура ρ_j для каждого перехода. Основные ее функции сводятся к анализу атрибутов меток, размещенных на входных позициях с целью поиска корректной метки. Если такая метка найдена, то переход начинает реализацию рабочей стадии с помощью запуска процедуры обслуживания перехода π_j. Эта процедура состоит из процедуры анализа типов и процедур локальных процессов:

$$\pi_j = (Analyse, pr_1, \cdots, pr_k), \quad (15)$$

где $k \leq \text{Max}\_t_j$.

Процедура $Analyse(\varphi\_\mu_l)$, $1 \leq l \leq \text{Max}\_p_i$ выполняет функцию селектора процессов перехода t_j. Процедуры $pr_1,..., pr_i,..., pr_k$ реализуют процессы обработки рабочей нагрузки. Например, при построении модели цифровой обработки сигналов упомянутые процедуры реализуют алгоритм быстрого преобразования Фурье. После выполнения произвольной процедуры pr_i всегда запускается процедура Transit, которая реализует этап деактивации перехода.

Деактивация – завершающий этап активности перехода. Этот этап, как правило, сопровождается размещением меток на его выходных позициях. Для размещения произвольной метки на выходной позиции необходимо выполнение таких условий:

1. Метка должна быть «созревшей», то есть процедура $Analyse$, управляющая выполнением основных блоков процедуры π_j, должна успешно завершить свою работу. В случае успешного завершения формируются атрибуты метки α_k и параметров Id_μ_k, φ_μ_k.
2. Количество меток на целевой исходной позиции не должно превышать максимально допустимое количество Max_p_i.
3. Тип метки φ_μ_k должен входить во множество допустимых типов меток для данной позиции: $\varphi\_\mu_k \in \varphi\_p_i$.

В случае успешного размещения метки на выходной позиции создается событие активации в том переходе, для которого данная позиция является входной.

Признаком завершения процесса моделирования является отсутствие условий активации для всех переходов сети.

Выводы

В работе подчеркивается актуальность построения моделей дискретных систем, эволюционирующих путем свершения событий, связанных с активностями ресурсов. Рассмотрены способы формального представления таких систем, использующие маркирование, состояния и переходы для описания компонентов системы, их функционирования и взаимодействия. Дано определение прямого, обратного и взаимного подобия между системой и ее моделью. Приведен краткий обзор, отображающий логику развития инструментов формального описания дискретных моделей. Предложена новая версия сетей, APRO-сети, ориентированная на описание параллельных процессов и структур с асинхронным взаимодействием компонентов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бусленко Н.П., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем.– М.: Сов. радио, 1973.– 439 с.
2. Питерсон Дж. Теория сетей Петри и моделирование систем.– М.: Мир, 1984.– 264 с.
3. Котов В.Е. Сети Петри.– М.: Наука, 1984.– 160 с.
4. Jensen K. Coloured Petri Nets // Lecture Notes in Computer Science.– 1987.– vol.254.– P.248-299.
5. Nutt G.J. Evaluation nets for computer system performance analysis // 1972 Fall Joint Computer Conference, AFIPS Conference Proceedings. – 1973. – vol.41. – P.279-286.
6. Нестеренко Б.Б., Новотарский М.А. Мультипроцессорные системы.– Киев: Институт математики, 1995.– 408 с.
7. Boesh F.T., Wang J. Reliable circulant networks with minimum transmission delay // IEEE Trans. Cir. And Sys.– 1984.– CAS-32.– P.1286-1291.

Поступила в редколлегию 17.04.07.