

## ОБ ОДНОЙ МЕТОДИКЕ ПРОЕКТИРОВАНИЯ СУПЕРВИЗОРНОГО УПРАВЛЕНИЯ ДЛЯ ДИСКРЕТНО-СОБЫТИЙНЫХ СИСТЕМ<sup>1</sup>

*Аннотация.* В работе предлагается методика проектирования супервизоров для предотвращения запрещенных состояний для неуправляемых и управляемых дискретно-событийных систем. Рассматриваются вопросы поиска и навигации по пространству состояний в прямом и обратном направлениях, вычисления шагов в линейной структуре. Затронуты вопросы реализации систем безопасности. Приводится пример системы двух выталкивателей для демонстрации предложенной методики.

*Ключевые слова:* дискретно-событийные системы, супервизорное управление, сетевые системы «условие–событие», запрещенная маркировка, методика проектирования, линейный шаг, частичная маркировка, выталкиватель, граф достижимости критических маркировок.

*Abstract.* In the paper a technique for the design of supervisors for the avoidance of forbidden states for uncontrolled (plant) and controlled discrete event systems is proposed. Problems of forward and backward search and a navigation in the state space as well as a computation of steps in a linear structure are considered. The paper touches upon questions of the implementation of safety systems too. For the illustration of the suggested technique an example of a two-pushers system is given.

*Keywords:* discrete event systems, supervisory control, net condition/event systems, forbidden marking, design technique, linear step, partial marking, pusher, reachability graph of critical markings.

### Введение

В настоящее время в связи с внедрением компьютерных технологий и средств автоматизации во все сферы жизни все больший интерес проявляется к изучению и исследованию дискретно-событийных систем (ДСС). Для решения задач проектирования управляющих систем для ДСС была разработана теория супервизорного управления [1]. Основной задачей данной теории является синтез супервизора по заданной модели ДСС и спецификации желаемого поведения системы.

Хорошей идеей в области проектирования супервизоров ДСС является использование сетей Петри и производных от них моделей [2]. Например, в работе [3] для решения задач супервизорного управления используются «сетевые системы условие–событие» (NCES-сети), а в [4] – безопасные NCES-сети (sNCES-сети), являющиеся расширением сетей Петри.

В данной работе предлагается методика проектирования неблокируемых максимально разрешающих супервизоров для предотвращения запрещенных состояний, основанная на использовании расширенных sNCES-сетей.

<sup>1</sup> Работа выполнена в рамках проекта «Разработка комплекса формальных моделей и их трансформаций для проектирования распределенных информационно-управляющих систем промышленной автоматике» аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы (2009–2010 годы)», номер гос. регистрации НИР 01200952061.

В дальнейшем супервизоры подобного типа будем также называть контроллерами безопасности. Основные отличия предлагаемого подхода от подхода, изложенного в [4], заключаются в следующем:

1) в качестве области применения супервизорного управления рассматриваются произвольные ДСС – как управляемые, так и неуправляемые;

2) для решения задачи предотвращения используется изменение маркировки произвольных позиций (в том числе внутренних), а не изменение значений входов модулей, что позволяет работать с моделями произвольной структуры, в том числе с замкнутыми моделями;

3) для моделирования ДСС с супервизорным управлением (ДССсСУ) используются *sNCES*-сети, расширенные производственными правилами, в то время как в работе [4] не предусматривается отдельное моделирование ДССсСУ;

4) для поиска запрещающих решений используются реверсивные *sNCES*-сети (*RsNCES*-сети) [5], а не обычные *sNCES*-сети и обратный поиск в них, что позволяет наиболее полно и математически точно решать обратные задачи на сетях. Некоторые теоретические вопросы синтеза контроллеров безопасности в рамках предлагаемой методики были рассмотрены в работе [6], а вопросы реализации – в работе [7]. В данной работе также освещается ряд вопросов, не затронутых в [5–7].

### 1 Концепция подхода

В дальнейшем в качестве области применения предлагаемых методов будем рассматривать в общем случае управляемые системы (УС), включающие в себя систему управления (СУ), систему передачи данных (СПД) и систему оборудования (СО). СПД часто исключается из рассмотрения.

Суть предлагаемого подхода к проектированию супервизоров заключается:

1) в расширении исходной модельной нотации средствами, допускающими модификацию модели;

2) в разработке методов построения правил корректировки функционирования системы на уровне расширенной модели, приводящих к удовлетворению предъявляемых требований;

3) в разработке методов реализации и встраивания системы корректировки поведения на основе разработанных правил.

Основные моменты подхода представлены на рис. 1.

Преимущества изложенного подхода состоят в следующем:

1) моделирующие морфизмы  $F1$ ,  $F2$  и  $F3$  в процессе синтеза супервизора остаются неизменными. Варьируемой частью являются производственные правила. Это допускает легкость реконфигурации системы по отношению к целям супервизорного управления;

2) в ряде случаев разработка локальной системы корректировки проще и дешевле, чем перепроектирование и разработка всей УС в целом или даже ее части (например, СУ).

Примем следующие предположения:

– в качестве основополагающей модельной формы используются безопасные *NCES*-сети;

– желаемое поведение системы задается набором маркировок, которые соответствуют запрещенным состояниям системы.

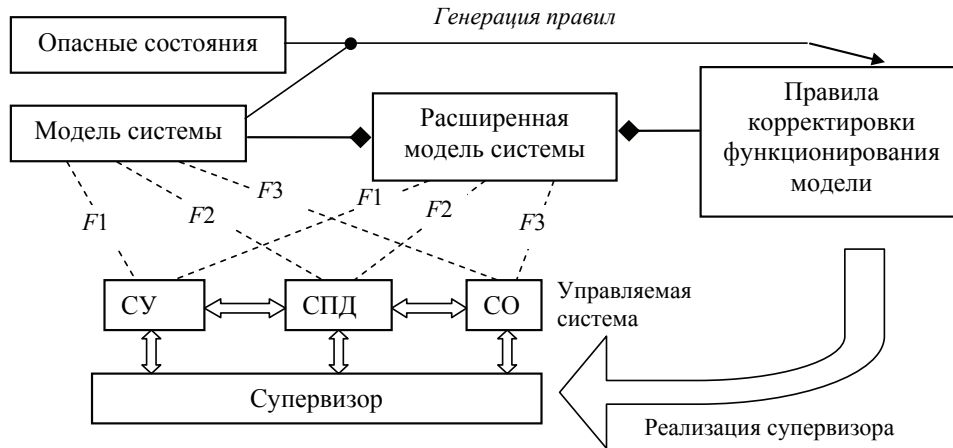


Рис. 1 Основные элементы подхода к проектированию супервизорного управления

Таким образом, основные этапы методики проектирования безопасных УС можно определить следующим образом:

- 1) построение *sNCES*-модели системы, поведение которой надо скорректировать;
- 2) формулировка запрещенных состояний (маркировок);
- 3) построение расширенной *sNCES*-модели системы (включая разработку правил корректировки функционирования системы);
- 4) реализация расширенной *sNCES*-модели системы путем программно-аппаратной реализации соответствующего супервизора и его встраивания в существующую систему.

## 2 Поиск и навигация в пространстве состояний

В основе построения запрещающих правил лежат механизмы поиска и навигации в пространстве частичных маркировок системы как в прямом направлении, так и в обратном. Двигаясь от опасной маркировки в обратном направлении, мы ищем маркировки (граничные), в которых можем запретить переход назад, уже в прямом направлении. Ранее были введены реверсивные частично-маркированные *sNCES*-сети (*RsNCES*-сети), позволяющие по заданной целевой частичной маркировке найти частичные маркировки, достижимые из нее (в обратном направлении) [5].

Для решения задачи поиска маркировок, достижимых из заданной начальной частичной маркировки в прямом направлении, ниже кратко вводятся частично-маркированные *sNCES*-сети (*DsNCES*-сети). Введение *DsNCES*- и *RsNCES*-сетей (кроме всего прочего) позволяет формализовать комбинированный поиск в сетях, одна часть которого начинается от начальной маркировки, а другая часть – от целевой. *DsNCES*-сеть определяется следующим набором:  $N = (P, T, F, C, E, em, m_0)$ , где все компоненты кортежа имеют тот же смысл, что и в обычных *sNCES*-сетях [4, 8], за исключением  $m_0$ . Начальная маркировка  $m_0$  является частичной и определяется как  $m_0: P \rightarrow \{0, 1, *\}$ . В *DsNCES*-сетях определяются различные уровни допустимости перехода: «безусловно разрешен», «условно разрешен», «не определен», «условно за-

прещен», «безусловно запрещен». Определение шага в *DsNCES*-сетях в принципе совпадает с определением шага в *sNCES*-сетях.

Обозначим через  $PG_t^*$  множество неопределенно маркированных входных и выходных позиций переходов событийного графа максимально разрешенных шагов (СГМРШ) [8], образуемых триггерным переходом  $t \in TR \subseteq T$ , где  $TR$  – множество триггерных переходов;  $\sigma_t^*: PG_t^* \rightarrow \{0, 1\}$  – подстановка;  $[\sigma_t^*]$  – множество всех возможных подстановок на множестве  $PG_t^*$ . Шаг  $\xi$  называется разрешенным при маркировке  $m$  и подстановке  $\sigma \in [\sigma_t^*]$ ,  $t \in \xi \cap TR$ , если каждый переход из  $\xi$  является разрешенным со стороны позиций при маркировке  $m$  и подстановке  $\sigma$  и не существует множества переходов  $\xi' = \xi \cup \{t\}$ , которое также является шагом и каждый переход которого разрешен. Разрешенный шаг может сработать. При срабатывании шага  $\xi$  при маркировке  $m$  последняя изменяется на семейство маркировок  $(m_i)$ , минимально покрывающее все маркировки, достижимые из маркировки  $m$  при срабатывании шага  $\xi$  при всех возможных подстановках  $\sigma \in [\sigma_t^*]$ ,  $t \in \xi \cap TR$ . Для *DsNCES*-сети, изображенной на рис. 2, в качестве примера приведено пространство состояний (рис. 3) при начальной маркировке  $m_0 = (10^{**})$ .

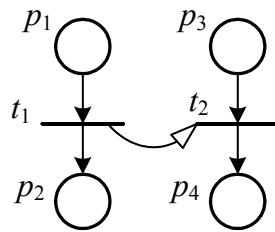


Рис. 2 Простая (\*)*sNCES*-сеть

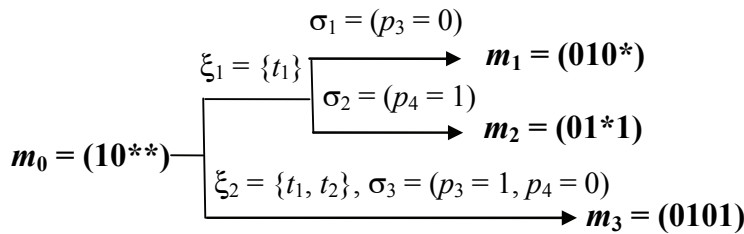


Рис. 3 Пространство состояний для *DsNCES*-сети, приведенной на рис. 2

Для эффективного вычисления разрешенных шагов и семейства маркировок  $(m_i)$  необходимо воспользоваться особенностями *NCES*-сетей, при интерпретации которых важна как разрешенность, так и запрещенность перехода. Для перехода, имеющего  $n$  входных и  $m$  выходных неопределенно-маркированных позиций, имеется одна разрешающая и  $(n + m)$  минимальных запрещающих подстановок. Очевидно, что  $n + m + 1 \leq 2^{n+m}$ , где в правой части указано общее число подстановок. При больших числах разница будет существенной. Очевидно, чтобы запретить переход, достаточно обнулить маркировку одной из его входных позиций или установить в 1 маркировку одной из его выходных позиций.

Ниже приводится ряд утверждений, касающихся связи прямых и обратных шагов. При упоминании обратного шага будем иметь в виду сеть  $RsNCES$ , а прямого – сеть  $DsNCES$ . В утверждениях используется отношение покрываемости одной маркировки другой. Запись  $m_a \angle m_b$  означает, что маркировка  $m_b$  покрывается маркировкой  $m_a$  (иначе – маркировка  $m_b$  конкретизирует маркировку  $m_a$ ) [5]. На рис. 4–6 для обозначения отношения покрываемости используются штриховые дуги.

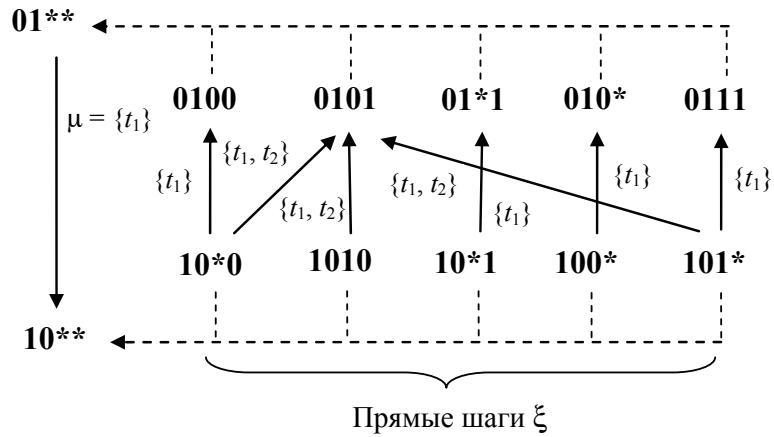


Рис. 4 Соотношение обратного шага  $\mu = \{t_1\}$  и прямых шагов на его основе для сетевой модели, изображенной на рис. 2

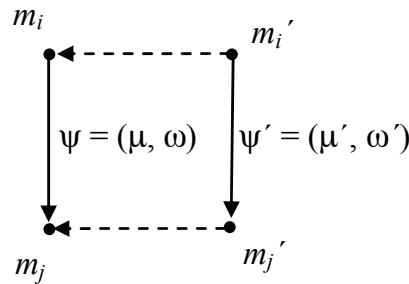


Рис. 5 Суть утверждения 2

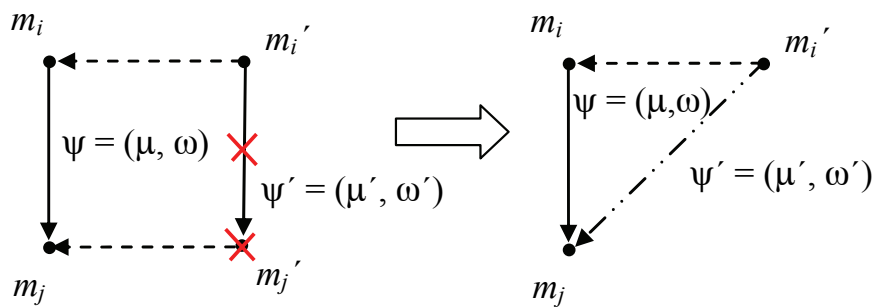


Рис. 6 Иллюстрация утверждения 5

**Утверждение 1** (о взаимосвязи срабатываний обратного и прямого шагов). Пусть срабатывание обратного усеченного допустимого шага  $\mu$  при маркировке  $t_i$  изменяет ее на маркировку  $t_j$ . Пусть  $t_j \angle t_j'$ . Тогда срабатывание прямого разрешенного шага  $\xi \supseteq \mu$  при маркировке  $t_j'$  изменяет ее на маркировку  $t_i'$  такую, что  $t_i \angle t_i'$ . Примеры, иллюстрирующие утверждение 1, можно найти на рис. 4.

**Утверждение 2** (о частном обратном срабатывании). Пусть срабатывание конфигурации  $\psi = (\mu, \omega)$ , которая включает обратный усеченный допустимый шаг  $\mu$  и запрещающую контекстную маркировку (ЗКМ)  $\omega$ , при маркировке  $t_i$  изменяет ее на маркировку  $t_j$ . Пусть  $t_i \angle t_i'$ . Тогда срабатывание при маркировке  $t_i'$  конфигурации  $\psi' = (\mu', \omega')$ , удовлетворяющей следующим условиям: 1)  $\mu' \supseteq \mu$ ; 2) при  $\mu' \supset \mu$  шаг  $\mu$  является максимальным среди всех допустимых шагов при маркировке  $t_i$ ; 3)  $\omega \angle \omega'$ , – изменяет ее на маркировку  $t_j'$  такую, что  $t_j \angle t_j'$  (рис. 5). При этом конфигурацию  $\psi'$  будем называть *частной* по отношению к конфигурации  $\psi$ .

**Утверждение 3** (о частном прямом срабатывании). Пусть срабатывание прямого разрешенного шага  $\xi$  при маркировке  $t_i$  изменяет ее на маркировку  $t_j$ . Пусть  $t_i \angle t_i'$ . Тогда срабатывание прямого шага  $\xi' \supseteq \xi$  при маркировке  $t_i'$  изменяет ее на маркировку  $t_j'$  такую, что  $t_j \angle t_j'$ .

**Утверждение 4** (об условии максимально разрешающего поведения) [7]. Пусть:

1) потенциально опасная маркировка  $t$  достижима в результате срабатывания обратных шагов  $\mu_1, \mu_2, \dots, \mu_n$ , и не существуют другие обратные шаги, ведущие в эту маркировку;

2) в результате срабатывания обратных шагов  $\mu_1, \mu_2, \dots, \mu_n$  возникают прямые шаги  $\xi_1, \xi_2, \dots, \xi_m$ , ведущие в безопасные маркировки.

Тогда для обеспечения максимально разрешающего поведения (по отношению к маркировке  $t$ ) необходимо и достаточно, чтобы предотвращение прямых шагов на основе обратных шагов  $\mu_1, \mu_2, \dots, \mu_n$  не предотвращало бы прямые шаги  $\xi_1, \xi_2, \dots, \xi_m$ . Для того чтобы условие максимально разрешающего поведения относилось ко всей сети, необходимо, чтобы условие утверждения 4 было действительно для каждой граничной маркировки.

**Утверждение 5** (об отсечениях). Пусть при построении графа обратной достижимости, или графа достижимости критических маркировок (ГДКМ), встречается ситуация, когда текущая маркировка  $t_i'$  покрывается ранее достигнутой маркировкой  $t_i$ . Пусть при маркировке  $t_i'$  возможна конфигурация  $\psi'$ , являющаяся частной по отношению к конфигурации  $\psi$ , которая возможна при маркировке  $t_i$ . Тогда маркировку  $t_j'$  можно отсечь (без влияния на результат поиска множества граничных маркировок) (рис. 6). Следует отметить, что при необходимости сохранить в графе обратной достижимости информацию об отсечении можно провести дугу специального вида от вершины  $t_i'$  к вершине  $t_j$  и пометить ее символом  $\psi'$  (рис. 6). Утверждение об отсечениях во многом основывается на утверждении 2. В действительности мы игнорируем частное решение, ссылаясь на то, что есть более общее решение.

### 3 Вычисление шагов в линейной структуре

На практике из шагов различных топологий наиболее часто встречается так называемый линейный шаг, когда множество переходов связано в цепоч-

ку с использованием событийных дуг (рис. 7). Для упрощения на рис. 7 используются переходы, имеющие по одной входной и выходной позиции. Обозначим:  $s_k = \{t_1, t_2, \dots, t_k\}$  – множество переходов цепочки, начинающейся в  $t_1$  и заканчивающейся в  $t_k$ . В соответствии с определением шага в sNCES-сетях [4, 8] очевидно, что в приведенной на рис. 7 сетевой модели шагом является любое из множеств  $s_1, s_2, \dots, s_k$ .

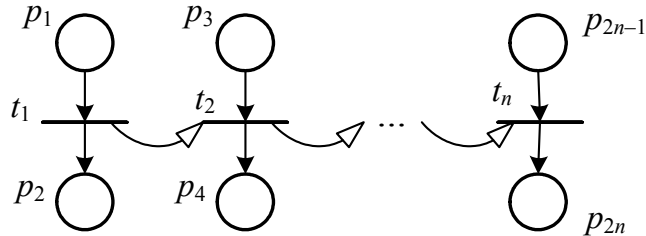


Рис. 7 Линейный шаг

Разрешенным (в прямом направлении) шагом называется шаг  $s_k$  такой, что разрешенными являются все переходы  $t_1, t_2, \dots, t_k$ , входящие в шаг; следующий переход цепочки  $t_{k+1}$  не разрешен (если такой переход существует). Этот разрешенный шаг является единственным и максимальным.

Шаг  $s_k$  является допустимым обратным шагом в линейной структуре (рис. 7), если его крайний переход  $t_k$  является безусловно допустимым, а переходы  $t_1, t_2, \dots, t_{k-1}$  – допустимыми. Запрещающая контекстная маркировка (ЗКМ) для допустимого обратного шага  $s_k$  ( $k < n$ ) формируется следующим образом. Пусть переходы  $t_{k+1}, t_{k+2}, \dots, t_{k+r}$  являются неопределенными, а переход  $t_{k+r+1}$  является условно разрешенным или разрешенным (в прямом направлении). Тогда число ЗКМ для шага  $s_k$  определяется как  $(2r + b)$ , где  $b = 0$ , если переход  $t_{k+r+1}$  является условно разрешенным, и  $b = 1$  – иначе. Процедуры получения вариантов ЗКМ определяются как  $m(p_{2i-1}) = 1$  и  $m(p_{2i}) = 0$ , где  $i = k + 1, k + 2, \dots, k + r$ . Для условно разрешенного перехода  $t_{k+r+1}$  ЗКМ определяется как  $m(p_{2*(k+r)+1}) = 1$  или  $m(p_{2*(k+r)+1}) = 0$ . Следует отметить, что при отсутствии неопределенных переходов  $t_{k+1}, t_{k+2}, \dots, t_{k+r}$  полагается  $r = 0$ , а при отсутствии условно разрешенного перехода  $t_{k+r+1}$  ЗКМ для шага  $s_k$  не строится и считается равной  $\emptyset$ .

#### 4 Вопросы реализации супервизорного управления

Реализация встроенной системы безопасности во многом будет определяться формой представления управляющей программы и продукционных правил. Архитектурные решения встраивания системы безопасности можно разделить на последовательные и параллельные. Последовательное решение предполагает чисто программную реализацию контроллеров безопасности, при этом контроллер процессов и контроллер безопасности выполняются на одном и том же оборудовании, что может привести к нарушению временных ограничений вследствие перегрузки аппаратных ресурсов. При параллельном решении для выполнения функций контроллера процессов и контроллера безопасности должны использоваться разные аппаратные средства, в том числе процессоры.

Пример распараллеливания работы контроллера процессов (на основе ПЛК) и контроллера безопасности представлен на рис. 8. Традиционные функции ПЛК – ввод входных переменных, выполнение алгоритмов, служебные операции, вывод выходных переменных – отмечены на рис. 8 идентификаторами  $F1$ ,  $F2$ ,  $F3$  и  $F4$  соответственно. Для реализации параллельного выполнения функций контроллеров процессов и безопасности может быть выбрана двухмашинная структура «ведущий–ведомый» с общей многоканальной памятью (рис. 9). Более подробно вопросы реализации рассмотрены в работе [7].

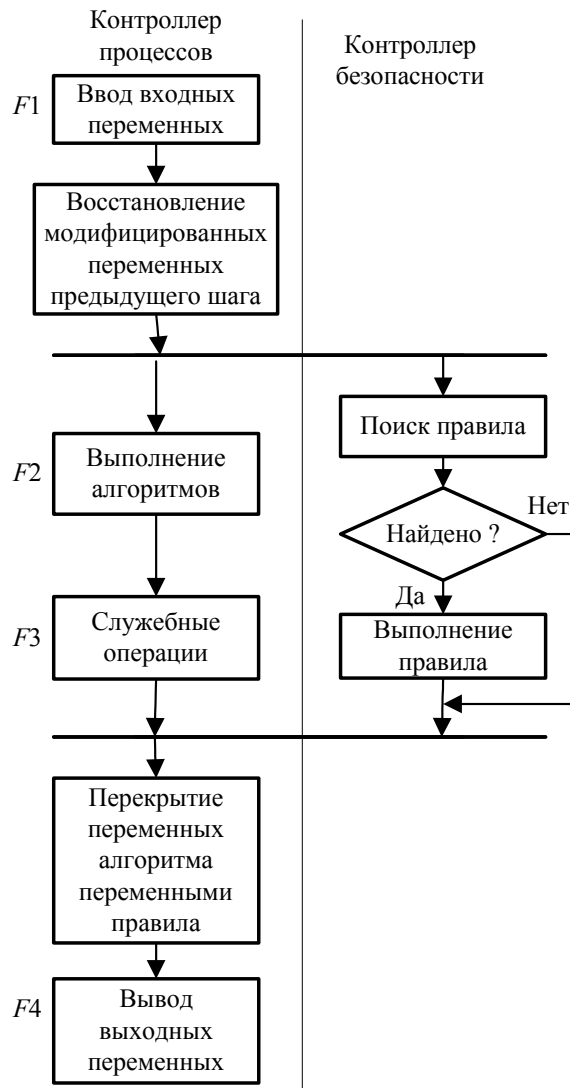


Рис. 8 Параллельный алгоритм функционирования котроллеров процесса и безопасности

### 5 Пример. Система двух выталкивателей

В качестве иллюстративного примера рассмотрим систему из двух выталкивателей, предназначенную для транспортировки деталей (рис. 10). Как



видно из рис. 10 траектория движения детали Г-образна и проходит через ряд характерных мест 1, 2 и 3. Система двух выталкивателей использовалась в ряде работ в качестве эталонного примера [9].

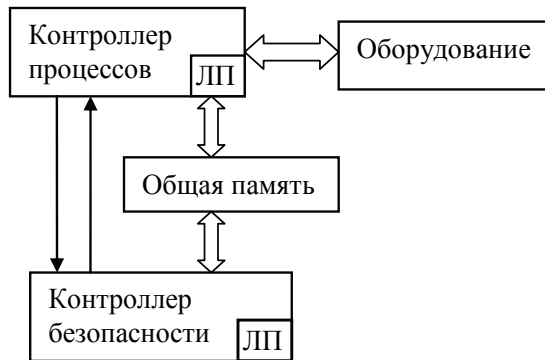


Рис. 9 Двухмашинная структура с общей памятью для встраивания контроллера безопасности

Исполнительный механизм выталкивателя позволяет выдвигать поршень выталкивателя и втягивать его. Для этого используются два управляющих сигнала – «Вперед» и «Назад». Каждый из выталкивателей снабжен двумя датчиками, сигнализирующими о нахождении поршня выталкивателя в одном из двух крайних положений – «Втянут» и «Выдвинут». В представленной системе двух выталкивателей имеется одна аварийная ситуация, когда оба выталкивателя находятся в выдвинутом состоянии (рис. 11).

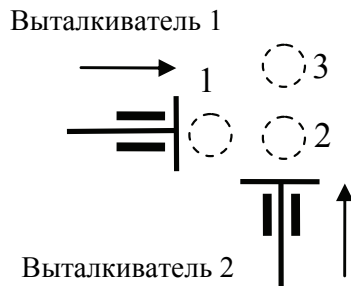


Рис. 10 Исходная ситуация



Рис. 11 Аварийная ситуация

Модульная *NCES*-модель замкнутой системы двух выталкивателей, включающей управляющий контур, представлена на рис. 12. Для упрощения модели считается, что выталкиватель может быть только в двух (крайних) состояниях «Втянут» и «Выдвинут». Каждый *NCES*-модуль имеет имя экземпляра и имя типа, указанные в верхней части модуля. Модуль типа *Pusher* представляет собственно выталкиватель, модуль типа *PControl* – управляющую часть системы, а модуль типа *WPTrace* – модель передвижения детали. Целевая маркировка показана на рис. 12 с использованием меток в виде заливтых кружков, а возможная начальная – в виде окружностей. Опасная ситуация описывается частичной маркировкой  $m(p9) = 1, m(p14) = 1$ .

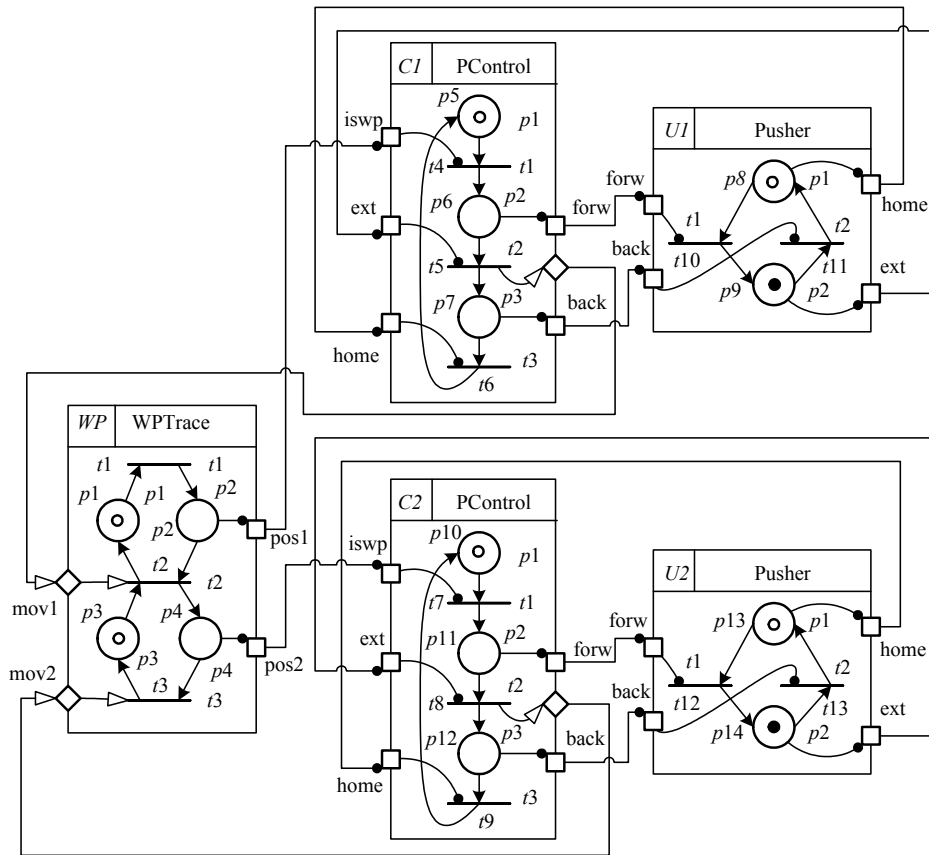


Рис. 12 NCES-модель замкнутой системы двух выталкивателей

В модуле типа Pusher маркировка  $m(p1) = 1$  определяет состояние выталкивателя «Втянут», а  $m(p2) = 1$  – «Выдвинут». Для перехода между этими двумя состояниями используются переходы  $t1$  и  $t2$ . Позиции модуля PControl определяют состояние последовательного циклического алгоритма управления выталкивателем. Если в начальном состоянии алгоритма (позиция  $p1$ ) перед поршнем выталкивателя имеется деталь (сигнал условия на входе  $iswp$ ), то в выталкиватель выдается команда на движение поршня вперед (через выход  $forw$ ). Когда поршень будет выдвинут (определяется через вход  $ext$ ), состояние алгоритма изменяется с  $p2$  на  $p3$ , и через выход  $back$  выдается команда на движение поршня назад. Как только выталкиватель окажется в состоянии «Втянут», алгоритм переходит в начальное состояние.

Модуль WPTrace моделирует конвейерное продвижение детали по местам. При этом деталь переходит на следующее место, если оно не занято. Маркированная позиция  $p2$  ( $p3$ ) свидетельствует о том, что деталь находится на первом (втором) месте. Триггерный переход  $t1$  моделирует поступление детали в систему извне. Следует отметить, что в некоторых системах наличие/отсутствие детали перед поршнем может определяться только по модели, без использования датчиков. В этом случае модель передвижения детали должна быть неотъемлемой составной частью контроллера.

Для построения графа обратной достижимости для модели системы двух выталкивателей использовался алгоритм из работы [10]. На рис. 13 при-

ведена часть построенного графа обратной достижимости. Для построения ГДКМ использовалась модификация алгоритма, приведенного в [6]. Граничными маркировками считаются не только те маркировки, из которых можно запретить прямые шаги на основе всех обратных шагов (тип 1), но и те, из которых можно запретить прямые шаги на основе только некоторых обратных (тип 2). Но при этом не должно быть путей, состоящих из незапрещаемых прямых шагов, в неграничные маркировки. Модификация алгоритма позволяет учесть утверждение 5 и исключить возможную потерю некоторых решений.

Для построения ГДКМ задаются контролируемые позиции, т.е. позиции, маркировку которых система безопасности может изменить самостоятельно. Допустим, что контролируемыми позициями являются позиции  $p_2$  (полное имя  $WP.p_2$ ) и  $p_4$  (полное имя  $WP.p_4$ ). Кроме того, считается, что все позиции являются обозримыми. Сгенерированный ГДКМ приведен на рис. 12 как часть графа обратной достижимости. Он выделен штрихпунктирным контуром. Найденные граничные маркировки представлены в табл. 1. На рис. 12 граничные маркировки первого типа обведены прямоугольным контуром, а второго типа – окружным. Обратные шаги, для которых возможно запрещение соответствующих прямых шагов, отмечены пунктирной линией. Для предотвращения прямых шагов в граничной маркировке  $m_{14}$  генерируется следующее правило:

**ЕСЛИ**  $m(p_2) = 1 \ \& \ m(p_5) = 1 \ \& \ m(p_6) = 0 \ \& \ m(p_8) = 1 \ \& \ m(p_9) = 0 \ \& \ m(p_{11}) = 1 \ \& \ m(p_{13}) = 1 \ \& \ m(p_{14}) = 0$ , **ТО**  $m(p_2) = 0$ .

Данное правило предотвращает прямое срабатывания перехода  $t_4$  в данном состоянии.

Таблица 1

$m_3$	*1**10*10***1
$m_6$	***1****110*10
$m_{14}$	*1**10*10*1*10
$m_{16}$	***1*1*1010*10

Как отмечалось в работе [6], размерность ГДКМ, как правило, значительно меньше размерности графа обратной достижимости, что делает такой подход к синтезу довольно привлекательным. Шагом к его усовершенствованию для сетей небольшой размерности можно было бы считать удаление граничных маркировок, недостижимых из начальной (в  $DsNCES$ -сетях), а также ранжирование всех граничных маркировок. Для построения графа обратной достижимости и ГДКМ использовалась специально разработанная для этих целей программа на языке *C*.

### Заключение

В данной работе были рассмотрены основные положения методики проектирования супервизоров для предотвращения запрещенных состояний на основе модифицированных *NCES*-сетей. Направлением дальнейших исследований является расширение предложенной методики для нечеткого супервизорного управления.

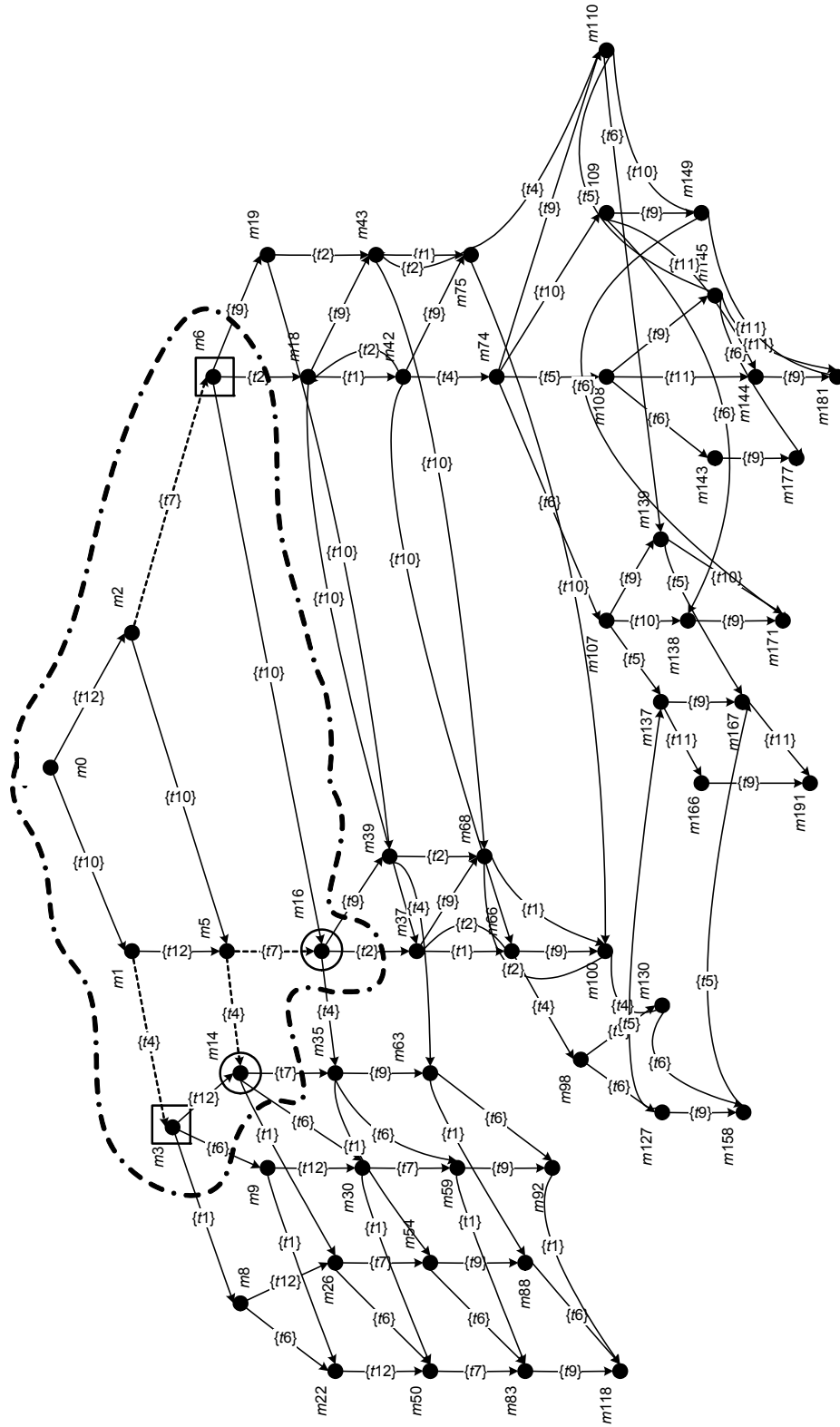


Рис. 13 Граф обратной достижимости и ГДКМ

**Список литературы**

1. **Ramadge, P. J.** The Control of Discrete Event Systems / P. J. Ramadge, W. M. Wonham // Proc. IEEE. – 1989. – V. 77. – Issue 1. – P. 81–98.
2. **Uzam, M.** A New Petri-Net-based Synthesis Technique for Supervisory Control of Discrete Event Systems / M. Uzam, A. H. Jones // Turk. J. Elec.Engin. – V. 10. – № 1. – 2002.
3. **Hanisch, H.-M.** Synthesis of Supervisory Controllers Based on a Novel Representation of Condition/Event Systems / H.-M. Hanisch, M. Rausch // IEEE Int. Conf. on Systems, Man and Cybernetics. – Vancouver, Canada, 1995.
4. **Missal, D.** Synthesis of distributed controllers by means of a monolithic approach / D. Missal, H.-M. Hanisch // Proc. 11th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA'2006). – Prague, 2006. – P. 356–363.
5. **Дубинин, В. Н.** Реверсивные частично маркированные *sNCES*-сети / В. Н. Дубинин, Х.-М. Ханиш, Д. Миссал // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 2. – С. 22–31.
6. **Дубинин, В. Н.** Использование реверсивных частично маркированных *sNCES*-сетей в синтезе контроллеров безопасности для дискретно-событийных систем / В. Н. Дубинин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2008. – № 1. – С. 25–35.
7. **Дубинин, В. Н.** Об одной методике проектирования супервизоров для предотвращения запрещенных состояний в дискретно-событийных системах / В. Н. Дубинин // Современные информационные технологии : труды Международной научно-технической конференции. – Вып. 8. – Пенза, 2008. – С. 143–159.
8. **Dubinín, V.** Event graph-based approach to interpretation of NCES models / V. Dubinín, H.-M. Hanisch, D. Missal // Новые информационные технологии и системы : труды VII Международной научно-технической конференции. – Пенза, 2006. – Ч. 1. – С. 172–186.
9. **Pinzon, L.** A comparative study of synthesis methods for discrete event controllers / L. Pinzon, M. Jafari, H.-M. Hanisch, T. Boucher. – Rutcor Research Report : Rutgers University, 1997. – 54 p.
10. **Дубинин, В. Н.** Интерпретация реверсивных частично маркированных *sNCES*-сетей / В. Н. Дубинин, Х.-М. Ханиш, Д. Миссал // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 3. – С. 23–34.

---

**Дубинин Виктор Николаевич**

кандидат технических наук, доцент,  
кафедра вычислительной техники,  
Пензенский государственный  
университет

E-mail: victor\_n\_dubinin@yahoo.com

---

**Dubinín Viktor Nikolaevich**

Candidate of engineering sciences,  
associate professor, sub-department  
of computer science, Penza State University

---

УДК 519.714

**Дубинин, В. Н.**

**Об одной методике проектирования супервизорного управления для дискретно-событийных систем / В. Н. Дубинин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 3 (11). – С. 12–24.**