

УДК 004.825

## Системы, основанные на знаниях, с использованием раскрашенных сетей Петри

*Чернецкая Д.И., магистр*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Системы автоматического управления»*

*Научный руководитель: Суханов В.А., к.т.н., доцент  
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Системы автоматического управления»*

[suhanov@iu1.bmstu.ru](mailto:suhanov@iu1.bmstu.ru)

В настоящее время интеллектуальные системы являются неотъемлемыми компонентами развитых автоматизированных систем управления. Среди основных моделей представления знаний в интеллектуальных системах выделяют системы продукций, системы фреймов, семантические сети, лены, нейронные сети. Продукционные модели можно считать наиболее распространенными моделями представления знаний. Для представления набора продукционных правил и структуризации интеллектуальной системы могут использоваться раскрашенные сети Петри.

Целью данной работы является разработка продукционной модели представления знаний на примере конкретной предметной области – модель «Клиент-Сервер».

Для достижения поставленной цели решались следующие задачи:

- изучение подходов к продукционным системам, основанным на знаниях (СОЗ);
- исследование различных возможностей раскрашенных сетей Петри для систем с продукционной моделью представления знаний;
- апробация предложенного подхода на конкретном примере – модель «Клиент-Сервер» с помощью программного комплекса CPN Tools 4.0.

Общая структура СОЗ содержит несколько функциональных блоков (подсистем), часть которых объединена в функциональные группы (интеллектуальный модуль, пользовательский интерфейс блока). В структуре СОЗ можно выделить три основных укрупненных блока: базу данных, базу знаний и решающий блок.

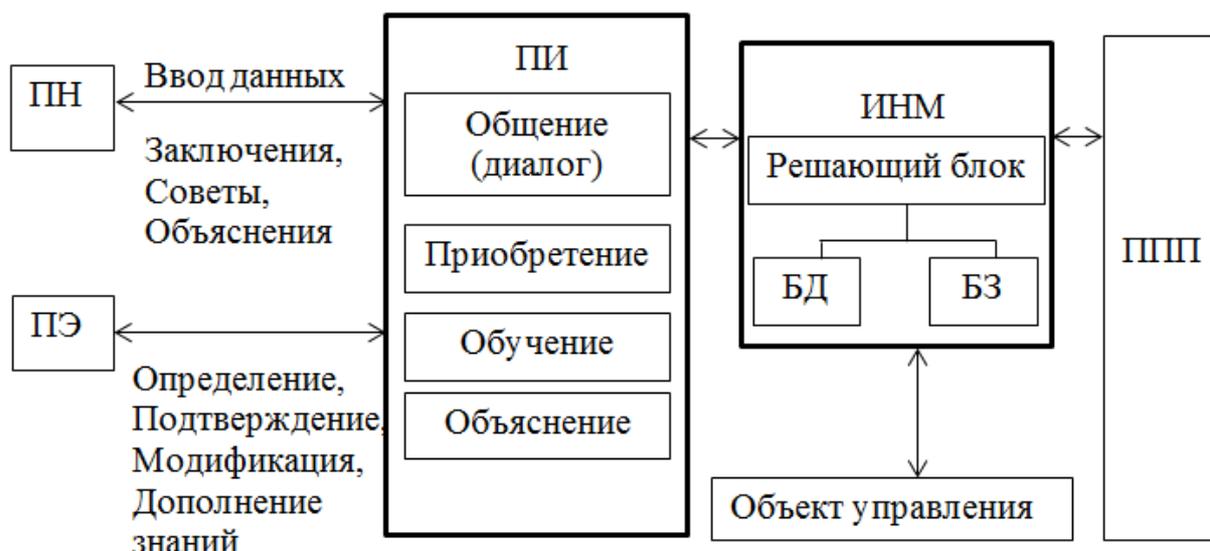


Рис. 1. Обобщенная функциональная схема СОЗ

На рис. 1 представлено: ПН – пользователь-неспециалист; ПЭ – пользователь-эксперт, ПИФ – пользовательский интерфейс; ИНМ – интеллектуальные модули, общение (диалог) – диалоговый компонент; обучение – компонент обучения; приобретение – компонент приобретения знаний; объяснение – компонент объяснения; ППП – пакеты прикладных программ.

В настоящее время разработано множество моделей представления знаний, но наиболее распространенными являются продукционные модели. Продукционная модель – это модель, основанная на правилах, позволяющая представить знание в виде предложений типа: «ЕСЛИ условие, ТО действие».

Любое правило состоит из одной или нескольких пар «атрибут-значение». В рабочей памяти систем, основанных на продукционных моделях, хранятся пары атрибут-значение, истинность которых установлена в процессе решения конкретной задачи к некоторому текущему моменту времени. Содержимое рабочей памяти изменяется в процессе решения задачи. Это происходит по мере срабатывания правил. Правило срабатывает, если при сопоставлении фактов, содержащихся в рабочей памяти, с antecedентом анализируемого правила имеет место совпадение, при этом заключение сработавшего правила заносится в рабочую память. Поэтому в процессе логического вывода объём фактов в рабочей памяти, как правило, увеличивается (уменьшаться он может в том случае, если действие какого-нибудь правила состоит в удалении фактов из рабочей памяти). В процессе логического вывода каждое правило из базы правил может сработать только один раз.

Основные достоинства систем, основанных на продукционных моделях, связаны с простотой представления знаний и организации логического вывода.

При разработке небольших систем (десятки правил) проявляются в основном положительные стороны продукционных моделей знаний, однако при увеличении объема знаний заметными становятся слабые стороны, в частности, усложняется оценка целостного образа знаний, снижается эффективность обработки знаний.

Сети Петри (СП) – это математический аппарат двудольных графов, используемый для анализа, моделирования и представления причинно-следственных связей в сложных распределенных дискретных системах, процессах параллельной обработки и синхронизации, что является весьма важным, в частности, для продукционных систем. При этом общее описание процессов может быть представлено в форме двудольных ориентированных графов. Графы переходов и конечные автоматы являются частными случаями СП.

Важно, что СП описывают не только сам процесс, но и управление этим процессом.

Представление набора продукционных правил и структуризация их в форме СП могут использоваться в качестве способа внутреннего представления знаний в СОЗ.

СП представляет собой граф, который обладает следующими свойствами:

- 1) Каждая из вершин графа является либо позицией, либо переходом;
- 2) Граф является ориентированным, каждая дуга соединяет позицию с переходом, либо переход с позицией;

- 3) Каждой из позиций ставится в соответствие некоторое целое неотрицательное число  $N$ , при этом говорят, что в этой позиции находится  $N$  меток; это число может быть равно 0, т. е. меток в позиции нет; в процессе работы сети, количество меток в позиции может изменяться;

- 4) Каждой дуге ставится в соответствие некоторое натуральное число, которое называют кратностью данной дуги; при кратности 1 обозначение 1 обычно опускается; иногда для общности вводят кратность 0, которая соответствует отсутствию дуги; кратность дуги в процессе работы сети не изменяется.

Формально СП может быть представлена в виде четверки  $G = (P, T, I, O)$  или графически в форме двудольных ориентированных графов (на рисунке 2.), где  $P = \{p_1, \dots, p_n\}$ ,  $P \neq \emptyset$  – множество вершин, называемых позициями (местами);  $T = \{t_1, \dots, t_m\}$ ,  $T \neq \emptyset$  – множество вершин, называемых переходами;  $I, O$  – входная и выходная функции, которые

представляют собой соответственно прямую и обратную функции (матрицы) инцидентности:

I:  $P \times T \rightarrow \{0, 1\}$  – описывает входные для переходов дуги;

O:  $T \times P \rightarrow \{0, 1\}$  – описывает выходные для переходов дуги.

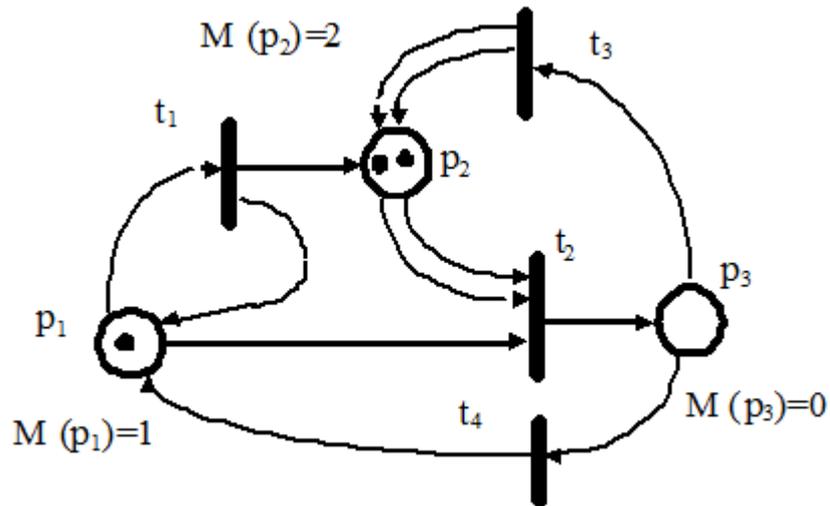


Рис. 2. Графическое представление СП

Таким образом, отношение инцидентности можно задать, указав для каждого перехода  $t_j$  два вектора –  $I(t_j)$  и  $O(t_j)$  размерностью  $n$ , где  $I(t_j)$  содержит сведения о входных для  $t_j$  дугах для всех позиций сети,  $O(t_j)$  – о выходных дугах.

Для каждого перехода  $t_j$  можно определить множества входных  $I(t_j)$  и выходных  $O(t_j)$  позиций:

$$I(t_j) = \{ p_i \in P / I(p_i, t_j) = 1 \}, O(t_j) = \{ p_i \in P / O(t_j, p_i) = 1 \}, \text{ где } i = \overline{1, n}, j = \overline{1, m}$$

Если дуга двудольного графа направлена из  $i$ -ой позиции в  $j$ -ый переход, то  $I(p_i, t_j) = 1$ , иначе  $I(p_i, t_j) = 0$ , где  $i = \overline{1, n}, j = \overline{1, m}$ . Аналогично – для других элементов.

Для описания динамических свойств сети (моделируемой системы) вводится функция  $M$  разметки сети  $M: P \rightarrow \{0, 1, 2, \dots\}$ . С помощью функции  $M_i = M(p_i)$ , где  $i = \overline{1, n}$ , позиции помечаются целыми неотрицательными числами. При графическом задании СП разметка отображается соответствующим числом точек  $M_i = M(p_i)$  внутри соответствующих кружков – позиций  $p_i$ . Эти точки называют метками (фишками, маркерами).

Разметку можно представить также в векторной форме:

$$M = (M_1, M_2, \dots, M_n),$$

где  $M_i = M(p_i)$ .

Размеченная сеть обозначается в виде:

$G=(P,T,I,O,M_0)$ ,

где  $M_0$  - начальная разметка.

Срабатывание какого-либо перехода  $t_j$  в размеченной сети ведет к смене разметки.

Аналогично для каждой позиции  $p_i$  можно определить множества входных и выходных переходов  $-I(p_i)$  и  $O(p_i)$  соответственно,  $i = \overline{1, n}$ .

Можно получить математическое описание функционирования СП в виде уравнения, описывающего изменение разметки в результате срабатывания перехода. Новая разметка  $M(k+1)$ , т.е. разметка после срабатывания перехода  $t_j$ , может быть определена по следующему правилу:

$$M(k+1) = M(k) - I(t_j) + O(t_j) = M(k) + \Phi(t_j),$$

где  $M(k)$  – вектор текущей разметки (состояние СП);  $\Phi(t_j) = O(t_j) - I(t_j)$  – столбец матрицы  $\Phi$ , соответствующий переходу  $t_j$ .

Это уравнение описывает изменение разметки сети при срабатывании одного  $j$ -го перехода, т.е. не носит общего характера.

Срабатывание перехода – неделимое действие, изменяющее разметку входных и выходных позиций, при этом событие, изображаемое переходом, изменяет состояние (емкость) непосредственно связанных с ним условий так, что емкость предусловий, вызвавших реализацию этого события, уменьшается, а емкость постусловий, на которые оно влияет, увеличивается. Функционирование размеченной СП представляет собой поочередный запуск переходов. В каждый момент времени какие-то переходы могут быть запущены, а какие-то – нет. Другими словами, это процесс изменения разметки (начиная от  $M_0$ ), осуществляемый по правилам:

1. Если для перехода  $t_j$  выполняется необходимое условие срабатывания, то происходит срабатывание этого перехода, в результате чего изменяется разметка во всех входных и выходных позициях только этого перехода:
  - а. Из каждой позиции, которая является входной для данного перехода, изымается число меток, равное кратности соответствующей дуги.
  - б. В каждую выходную для данного перехода позицию добавляется число меток, равное кратности соответствующей дуги.
2. Если два или более перехода могут одновременно сработать, и они не имеют общих входных позиций, то их срабатывание независимо и осуществляется в любой последовательности или параллельно.
3. Если несколько переходов могут сработать и имеют общую входную позицию, то сначала срабатывает только один переход, любой из них; при этом может

оказаться, что, сработав, этот переход лишит другие переходы возможности сработать - этим моделируется конфликтная ситуация (она устраняется вне формализма СП).

4. Функционирование СП продолжается до тех пор, пока существует хотя бы один переход, готовый к срабатыванию, и останавливается, если ни один из ее переходов не может сработать.

Представление какой-либо системы в виде СП основано на двух основополагающих понятиях: событиях и условиях. События – это действия, имеющие место в системе; их возникновением управляет состояние системы, которое может быть описано множеством условий.

Ситуации в системе формируются с помощью локальных операций, называемых условиями реализации событий. Условие – это предикат или логическое описание состояния системы; оно может принимать значения либо "истина", либо "ложь" (1 и 0). Условие соответствует таким ситуациям в системе, как наличие данных для операции в программе, состояние некоторого регистра, наличие детали на конвейере и т.п.

Условие имеет емкость, которая задается равной 0, если оно не выполнено; 1 – если выполнено; N – если условие выполнено с N-кратным запасом (N – целое положительное число). Емкости условий соответствуют функциям разметки позиций  $M_i$ . Т.к. события являются действиями, то они могут происходить или не происходить. Для их выполнения необходимы соответствующие условия, называемые предусловиями событий. Возникновение события может вызвать нарушение (изменение) предусловий и привести к выполнению других условий – постусловий.

Таким образом, определенные сочетания условий разрешают реализоваться некоторому событию, а реализация события изменяет некоторые условия, т.е. события взаимодействуют с условиями, а условия – с событиями.

При моделировании СП метки часто соответствуют объектам, передаваемым от компонента к компоненту системы (данные в информационных системах, детали, ресурсы и т.д.). Эти объекты могут иметь дополнительные атрибуты, позволяющие различать их и использовать эти различия для управления функционированием системы.

Раскрашенные сети Петри – это графоориентированный язык для проектирования, описания, имитации и контроля распределенных и параллельных систем. Графическими примитивами показывается течение процесса, а конструкциями специального языка имитируется необходимая обработка данных. Сеть представляет собой направленный граф с двумя типами вершин – позициями и переходами, при этом дуги не могут

соединять вершины одного типа, т.е. граф является двудольным. Множество позиций (обозначаются эллипсом) описывают состояния системы. Переходы (обозначают прямоугольниками) описывают условия изменения состояний. Позиции называются входными для конкретного перехода, если направление дуги, указывает на переход. Позиции называются выходными для перехода, если дуга ведет от перехода к позиции.

В отличие от «классических» сетей Петри, в раскрашенных немаловажную роль играет типизация данных, основанная на понятии множества цветов, которое аналогично типу в декларативных языках программирования. Соответственно, для манипуляции цветом применяют переменные, функции и другие элементы, известные из языков программирования. Ключевой элемент РСП – позиция – имеет определенное значение из множества цветов.

Для отражения динамических свойств в сеть Петри введено понятие разметки сети, которая реализуется с помощью, так называемых фишек, размещаемых в позициях. Цвет позиции определяет тип фишек, которые могут там находиться. Конкретизация фишки, находящейся в данной позиции определяется инициализирующим выражением начальной разметки или формируется в результате правильного выполнения шага итерации сети Петри.

Сеть представляет собой асинхронную систему, в которой фишки перемещаются по позициям через переходы. Переход может сработать (т.е. переместить фишку из входной позиции в выходную для данного перехода), если во всех входных позициях для данного перехода присутствует хотя бы одна фишка и выполнено логическое выражение, ограничивающее переход (спусковая функция).

Дуги могут иметь пометки в виде выражений (переменных, констант или функций), определенных для множества цветов, и использоваться, либо для "вычленения" компонент сложного цвета фишек при определении условия срабатывания перехода, либо для изменения цвета фишки следующей позиции после срабатывания перехода.

Таким образом, РСП представляется четверкой:  $G = (P, T, I, O)$ , где  $P = \{p_1, \dots, p_N\}$ ,  $N \geq 0$  – конечное множество позиций,  $T = \{t_1, \dots, t_M\}$ ,  $M \geq 0$  – конечное множество переходов;  $P \cap T = \emptyset$ ;  $I : T \rightarrow P^\infty$  – входная функция-отображение множества переходов в помеченные комплекты позиций  $I = \{I_1, \dots, I_L\}$ ;  $O : T \rightarrow P^\infty$  – выходная функция-отображение множества переходов в помеченные комплекты позиций;  $O = \{O_1, \dots, O_L\}$ ,  $L = |D|$ ;  $D = \{d_1, \dots, d_L\}$  – множество пометок (цветов). В случае  $|D|=1$  мы имеем дело с «классическими» СП.

Позиция  $p_i$  является входной позицией перехода  $t_j$  в том случае, если  $\exists l: l = \overline{1, L}$  такое, что  $p_i \in I_L(t_j)$ . Позиция  $p_i$  является выходной позицией перехода  $t_j$  в том случае, если  $\exists l: l = \overline{1, L}$  такое, что  $p_i \in O_L(t_j)$ . Входы и выходы переходов представляют собой совокупность помеченных позиций.

Кратность входной позиции  $p_i \in I_L(t_j)$  из  $l$ -го комплекта позиций  $(P^\infty)^l$  для перехода  $t_j$  есть число появлений в позиции в данном входном комплекте перехода, и обозначается  $\#(p_i, I^l(t_j))$ . Аналогично  $\#(p_i, O^l(t_j))$  – кратность выходной позиции в  $l$ -м комплекте [2].

Для управления динамикой РСП используют маркировку  $M$ , являющуюся матрицей размерности  $N \times L$ :  $M = \{M_{il}\}$ , где  $N = |P|$  и каждое  $M_{il}$  принадлежит множеству неотрицательных целых чисел,  $i = \overline{1, N}$ ,  $l = \overline{1, L}$ . Выполнение помеченной сети Петри осуществляется с учетом количества, пометок и распределения фишек в сети. Сеть Петри выполняется посредством запусков переходов. Переход запускается удалением фишек из его входных позиций и образованием новых фишек, помещаемых в его выходные позиции. Переход может запускаться только в том случае, когда он разрешен.

Переход  $t_j \in T$  в маркированной РСП  $G = (P, T, I, O)$  с маркировкой  $M_{il}$  является разрешенным, если  $\forall p_i \in I^l(t_j)$  выполняется условие  $M_{il} \geq \#(p_i, I^l(t_j))$ ,  $l = \overline{1, L}$ .

Переход запускается удалением разрешающих фишек из его входных позиций и последующим помещением в каждую выходную позицию по одной фишке для каждой дуги. Если какая-либо входная позиция перехода не обладает достаточным количеством фишек с нужной пометкой, то переход не разрешен и не может быть запущен.

Переход  $t_j$  в маркированной помеченной сети Петри с маркировкой  $M_{il}$  может быть запущен всякий раз, когда он разрешен. В результате запуска разрешенного перехода  $t_j$  образуется новая маркировка  $M^*_{il}$  определяемая соотношением:

$$M^*_{il} = M_{il} - \#(p_i, I^l(t_j)) + \#(p_i, O^l(t_j)).$$

Теория РСП разрабатывается более 20 лет рабочей группой (CPN Group) университета Орхуса (University of Aarhus, Denmark) под руководством профессора Курта Йенсена (Kurt Jensen). Ими разработана основная модель, включающая использование типов данных и иерархических конструкций, определены концепции динамических свойств, развивается теория методов анализа.

Также в рамках работы CPN Group был разработан программный продукт CPN Tools – специальная моделирующая система, которая использует язык сетей Петри для описания моделей. Система свободно распространяется для некоммерческих организаций. Уровень предоставляемого сервиса позволяет классифицировать CPN Tools как промышленную моделирующую систему. Она была использована в большом количестве

реальных проектов, особенно в области телекоммуникаций. В последнее время корпорация Nokia применяет CPN Tools для управляемой моделью разработки нового поколения мобильных телефонов.

Простейшая сеть терминалов удаленных операций имеет топологию «звезда»: несколько терминалов подключены к одному серверу, но не связаны между собой напрямую (рис. 3). Физическая природа канала связи в рамках рассматриваемой задачи значения не имеет, однако необходимо учесть требования оптимального использования ресурсов. Фактически это означает неспособность сети функционировать при полной нагрузке, когда все терминалы одновременно пытаются установить связь с сервером.

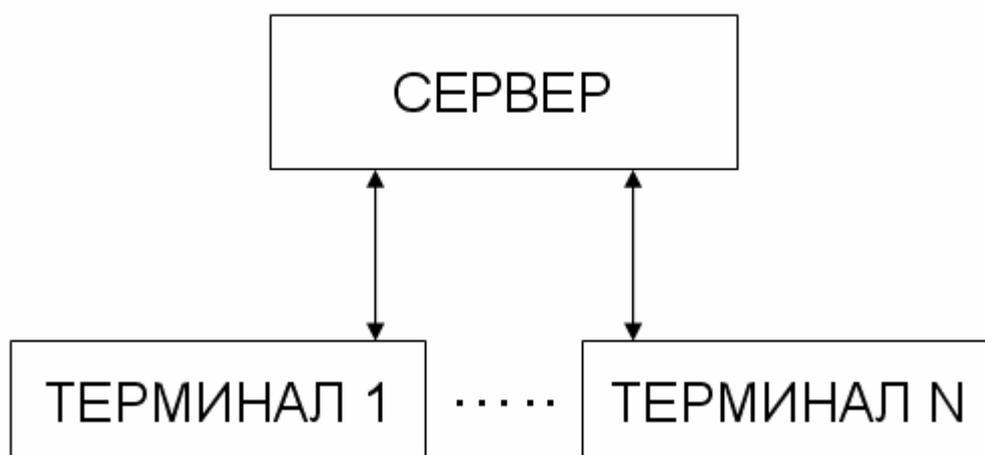


Рис. 3. Топология сети

Поскольку подобная ситуация маловероятна и наличие непрерывного канала связи не является обязательным, целесообразно устанавливать соединение терминала с сервером по мере необходимости (в момент использования терминала).

В случае многотерминальной системы обработки данных действующими лицами (т.е. внешними относительно системы сущностями) являются пользователи терминалов, а базовых вариантов использования два: запрос некоторой информации, необходимой для совершения операции, и сама операция.

Запрос данных и запрос-уведомление об операции подразумевают пересылку информации от клиента (терминала) к серверу и обратно. При этом клиент и сервер можно представить абстракцией черного ящика. Модульный подход к проектированию системы подразумевает выделение функций, выполняемых модулем и правил взаимодействия с ним, а внутренняя реализация этих функций вторична. Описав входы и выходы каждого модуля, из них, возможно, скомпоновать модель всей системы, что упрощает процесс разработки и тестирования. Корректная работа модуля гарантирована

соответствием его внутренней реализации заданному внешнему интерфейсу. Минимальный набор сообщений в коммуникации терминала и сервера довольно прост.

В процессе взаимодействия терминал посылает определенные сигналы серверу (запросы данных, запросы операций, уведомления о совершении операций) и получает также строго определенный набор ответных сигналов (запрошенные данные, подтверждения операций).

На основании функционирования данной сети была сформирована база производственных правил.

*Таблица 1*

#### Производственные правила

Номер правила	Формулировка правила
1	Если пользователь сделал запрос и терминал исправен, то формируется запрос для сервера.
2	Если запрос сформирован и существует свободный канал связи, то сервер готов принимать новые запросы, а терминал получает запрос на выполнение операции.
3	Если сервер свободен и последняя операция успешно завершена, то канал связи освобождается.
4	Если поступил запрос на выполнение операции, то: а) Операция успешно выполнена и терминал считается освободившемся; б) Операция не была завершена успешно и канал связи освобождается.

СП, реализующая данную производственную систему, представлена на рис.4.

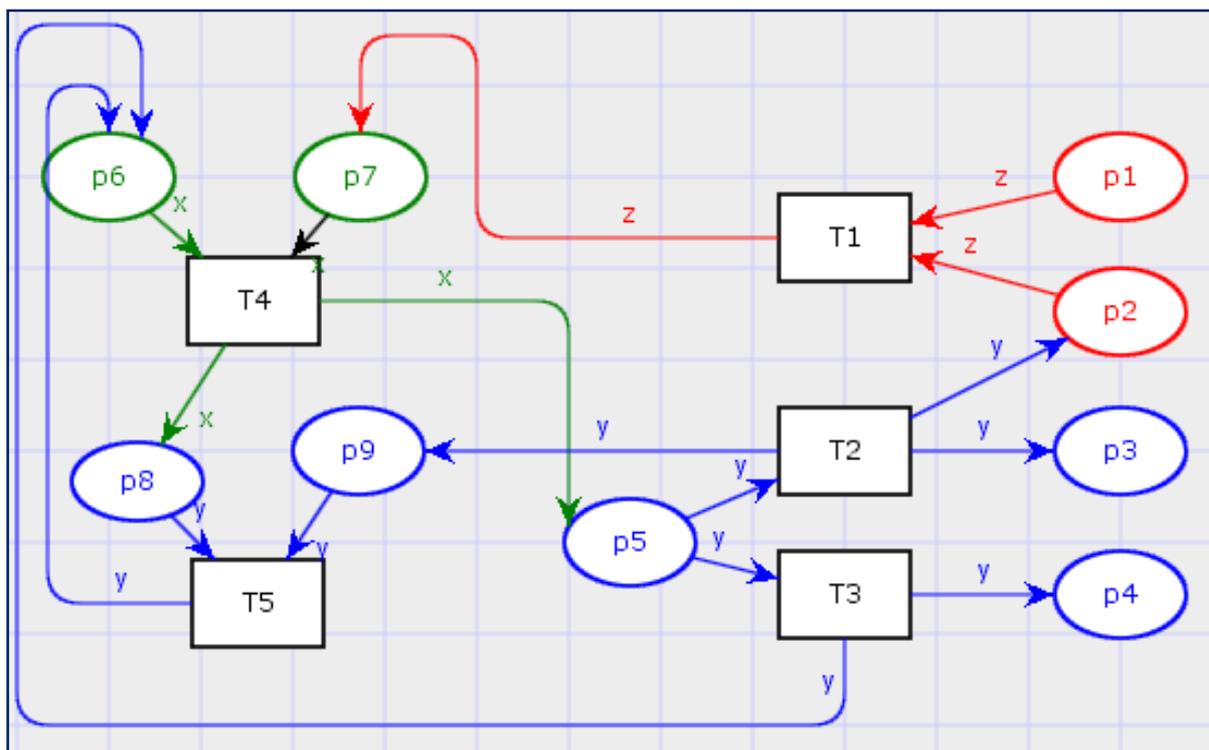


Рис. 4. РСП как динамическая модель модулей системы

Таблица 2

Описание позиций СП

Номер позиции	Интерпретация
1	Пользователь терминала совершил запрос информации
2	Терминал исправлен
3	Операция прошла успешно
4	Отказ в выполнении операции
5	Уведомление терминала о запросе
6	Готовность канала связи
7	Входящий запрос от пользователя сформирован
8	Готовность системы к дальнейшей работе
9	Операция закончена

## Описание переходов СП

Номер перехода	Интерпретация
1	Формирование запроса
2	Обработка запроса
3	Освобождение канала связи
4	Удачное завершение операции
5	Отказ в выполнении операции

Основные выводы:

- 1) Рассмотрены возможности продукционного подхода в СОЗ и модификация способа ПЗ (структурирование правил с помощью математического аппарата СП).
- 2) Исследованы возможности РСП для систем с продукционной системой представления знаний. Преимуществом РСП перед обычными СП является более компактный и удобный язык моделирования, а также возможность использования дополнительных атрибутов для разделения функционального управления системой.
- 3) Описана динамическая модель «Клиент-Сервер». Сформирована база продукционных правил для работы данной системы. На основе продукционного подхода построена РСП.

### Список литературы

1. Рыбина Г.В. Основы построения интеллектуальных систем. М.: Финансы и статистика; ИНФРА-М, 2010. 432 с.
2. Плотников В.Н., Суханов В.А. Системы, основанные на знаниях. М.: Изд-во МГТУ им. Н.Э. Баумана, 1995. 89 с.
3. Питерсон Дж. Теория сетей Петри и моделирование систем: пер. с англ. М.: Мир, 1984. 264с. [James L. Peterson - Petri net theory and the modeling of systems. Prentice-Hall, 1981, 241 p.]
4. Зайцев Д.А., Шмелева Т.Р. Моделирование телекоммуникационных систем в CPN Tools, Одесса, 2008, 67 с.
5. Зимаев И.В., Воевода А.А. Моделирование асинхронной сети автоматов обработки данных // Сборник научных трудов НГТУ. 2009. № 4 (58). С.147–154.