опирается только на предыдущий материал, максимально точно и верно, чтобы делать однозначные выводы, что в значительной степени экономит мыслительный процесс.

3. После курса «Математика и информатика» студент не должен научиться решать математические задачи (хотя это само по себе очень полезно), а научиться понимать и применять математические понятия и методологию, увидеть возможности формализации вербальных построений в различных областях знаний.

Многим уже сейчас ясно, что дальнейшее развитие гуманитарных наук без математического моделирования и точных количественных методов исследования, широкого использования современных вычислительных средств просто невозможно. Следует, правда, признать, что математика пока не располагает средствами, в достаточной мере отвечающими потребностям этих наук. По всей видимости, дело здесь в том, что создание соответствующего аппарата может явиться только результатом вполне осознанных совместных действий, как математиков, так и тех ученых-гуманитариев.

В то же время, целый ряд ученых констатирует, что сегодня людям крайне необходимо построить модель будущего развития земного бытия, способной помочь найти ответ на такие вопросы: как человечество будет существовать уже в ближайшем будущем? и каков рациональный путь его развития? Для этого необходимо огромное количество интеллектуальной силы, не в последней степени и из гуманитарной сферы, что требует соответствующего элитарного образования. Без системного подхода, применения математики и информатики выполнение такой задачи невозможно.

АНАЛИЗ СВОЙСТВ СЕТЕЙ ПЕТРИ И ИХ ИНТЕРПРЕТАЦИЯ В ИССЛЕДОВАНИИ СЛОЖНЫХ СИСТЕМ

Воронова И. Е., Крюкова В. В.
Кузбасский государственный технический университет

Сети Петри - инструмент исследования систем. Моделирование с использованием сетей Петри позволяет изучить динамику функционирования исследуемого объекта и его поведение при различных начальных условиях. Объектом исследования являются поведенческие свойства сетей Петри.

Цель статьи – установить соответствие поведенческих свойств сети Петри соответствующим характеристикам моделей сложных систем.

Сеть Петри состоит из позиций (мест) (P), переходов (T) и ориентированных дуг, соединяющих места и переходы. Выполнение условия изображается *разметкой* соответствующего места, а именно помещением n маркеров в это место $(n>0)$. Условие не выполнено, если место не имеет фишек; условие выполнено, если место имеет одну фишку; условие имеет ёмкость n, если в месте находится n фишек. *Срабатывания* переходов соответствуют реализациям событий и приводят к изменению разметки мест, т.е. к локальному изменению условий в системе.

Сеть Петри - это набор $N = (P, T, F, W, M_0)$, где (P, T, F) - конечная сеть (множество $X = P \cup T$ конечно), а $W: F \to \mathbf{N} | \{0\}$ и $M_0: P \to \mathbf{N}$ - две функции, называемые соответственно *кратностью дуг* и *начальной разметкой* (где $\mathbf{N}$ – множество натуральных чисел, включая ноль) [Котов 1984: 16].

Если кратность всех дуг сети (n) равна 1, то такая сеть называется *ординарной*. Если кратность $n>1$, то число n записывается рядом с дугой или дуга заменяется пучком из n дуг.

Ниже изложены основные свойства сетей Петри, их формальные определения и соответствующие интерпретации характеристик моделей сложных систем.

Ограниченность. Позиция $p_i \in P$ сети Петри $N = (P,T,F,W,M_0)$, с начальной маркировкой M_0 является *k-ограниченной*, если $M'(p_i) \leq k$ для всех $M' \in R(N,M_0)$. Сеть называется *ограниченной*, если в любой её позиции может появиться ограниченное число маркеров. Исследование проблемы ограниченности сводится к анализу *дерева достижимости*. Алгоритм построения [Котов 1984: 16]:

1. Первоначально предполагается, что дерево содержит единственную вершину-корень M_0 и не имеет дуг.

2. Пусть M – вершина дерева, которая еще не объявлена листом (т.е. вершиной, из которой не исходит ни одна дуга). Возможны четыре случая:

а) Ни один из переходов сети не может сработать при разметке M. В этом случае вершина M объявляется листом.

б) На пути из корня дерева в вершину M существует вершина M' такая, что $M'=M$. Вершина M объявляется листом.

в) На пути из корня дерева в вершину M существует вершина M' такая, что $M'<M$. Для любого места p такого, что $M'(p)<M(p)$, значение соответствующей координаты M заменяется на ω (бесконечно большое число) и вершина M объявляется ω - листом.

г) Если ни один из вышеперечисленных случаев не имеет места, то M – внутренняя вершина дерева. Для каждого перехода $t \in T$ такого, что $M \geq {}^{\bullet}F(t)$ (где ${}^{\bullet}F(t)$ – кратность дуги, соединяющей p и t), в дерево добавляется новая вершина $M' = M - {}^{\bullet}F(t) + F^{\bullet}(t)$ ($F^{\bullet}(t)$ – кратность дуги, соединяющей t и p) и дуга, ведущая из M в M', помеченная символом t.

Из анализа дерева достижимости следует, что сеть N ограничена тогда и только тогда, когда дерево не содержит лист с разметкой ω.

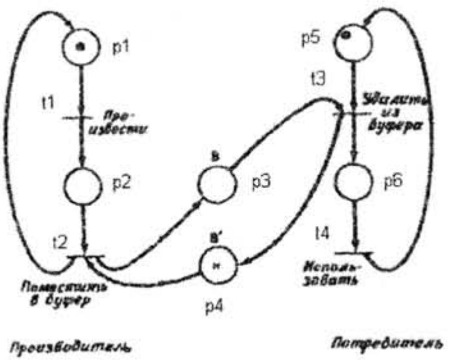


Рис. 1. *Задача о производителе/потребителе с ограниченным буфером*

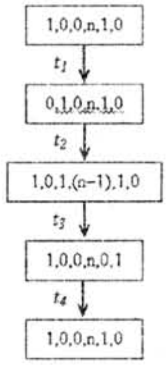


Рис. 2. *Фрагмент дерева достижимости сети на Рис. 1*

Ограниченность сети свидетельствует о конечности состояний отдельных элементов системы. Рассмотрим свойство ограниченности на примере социальной системы [Питерсон 1984: 65]. В системе производитель/потребитель (Рис.1) имеется буфер (***B***) ограниченного размера с n ячейками для хранения элементов данных, которые произведены, но еще не использованы (число заполненных ячеек), ***B'*** - количество пустых ячеек в буфере. Если потребитель работает медленно и буфер заполнен, производитель вынужден ждать. Если производитель попытается поместить еще один элемент данных в буфер, то он будет остановлен, так как в ***B'*** нет фишки, делающей этот переход разрешенным. На Рисунке 2 представлен фрагмент дерева достижимости для сети, представленной на Рис. 1.

Проанализируем свойство ограниченности на примере модели работы роторного экскаватора. Позиция моделирует погрузку экскаватором m базовых объемов угля на конвейер, с тем, чтобы загрузить уголь в бункер. Число маркеров в этой позиции соответствует количеству ковшей ёмкостью n базовых объёмов, достаточному для заполнения бункера. При анализе ограниченности сети определяют количество ковшей, необходимое для заполнения ёмкости m бункера.

Достижимость. Множество всех разметок, достижимых в сети N от начальной разметки M_0, называют *множеством достижимых разметок* $R(N,M_0)$ сети N. Разметка M *достижима* от разметки M_0, если существует последовательность запусков, приводящих от M_0 к M. Задача достижимости состоит в том, чтобы для сети N с маркировкой M_0 и маркировки M определить: $M \in R(N,M_0)$. Алгоритм анализа свойства достижимости также сводится к построению дерева достижимости. Если разметка M достижима от начальной разметки M_0, то существует узел с разметкой M', такой что $M \leq M'$. Анализ достижимости позволяет получить все допустимые и недопустимые состояния системы. Рассмотрим свойство достижимости на примере задачи распределения ресурсов между процессами вычислительной системы [Питерсон 1984: 85]. Пусть необходимо определить достижимое состояние, при котором в сети может возникнуть тупик (Рис. 3).

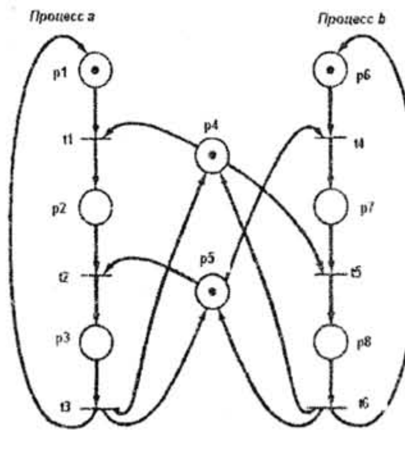


Рис. 3. *Модель распределения ресурсов*

Система имеет два различных ресурса q (моделируется p_4) и r (моделируется p_5) и два процесса a и b. Оба процесса нуждаются в обоих ресурсах. При срабатывании переходов t_1 и t_4, возникает ситуация, когда процесс a обладает ресурсом q и хочет получить r, процесс b обладает r и хочет получить q. Система заблокирована; никакой процесс продолжаться не может. Следовательно, в сети тупик может возникнуть, если достижимым является состояние (0,1,0,0,0,0,1,0).

Безопасность. Место p называется *безопасным*, если $\forall M \in R(N): M(p) \leq 1$, т.е. число фишек в нём никогда не превышает 1; соответственно *сеть безопасна,* если все её места безопасны. Алгоритм распознавания безопасности сети состоит в построении её дерева достижимости. Сеть с n местами безопасна, если и только если все вершины дерева достижимости представляют собой векторы из множества $\{0, 1\}^n$, т.е. векторы составленные из 0 и 1.

Безопасность – это частный случай более общего свойства ограниченности. Если интерпретировать позиции как условия, то свойство безопасности характеризует истинность (присутствие фишки в позиции) или ложность (отсутствие фишки) логически высказанного условия.

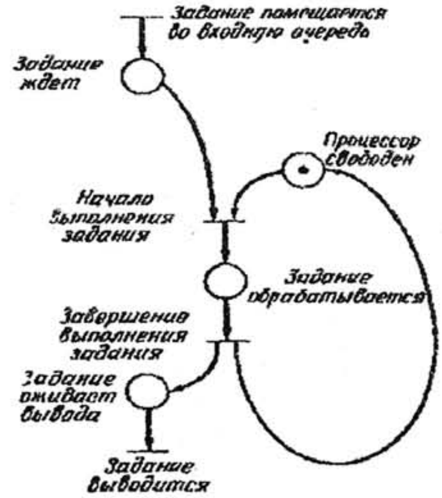
Рис. 4. *Модель работы бункера*

Рассмотрим свойство безопасности на примере модели работы бункера (Рис.4). Заполнение бункера углем посредством работы комбайна моделируется одной фишкой в соответствующей позиции. Сеть безопасна, если её маркировка соответствует режиму работы комбайна, при котором переполнение бункера исключается.

Проанализируем модель вычислительной системы применительно к свойству безопасности.

Задания поступают на устройство ввода (Рис. 5). Когда процессор свободен и в устройстве ввода есть задание, процессор начинает обработку задания. Когда задание выполнено, оно посылается в устройство вывода; процессор же либо продолжает обрабатывать другое задание, если оно имеется, либо ждет прихода задания, если устройство ввода еще не получило такового. Сеть безопасна, если буфер ограничен одной ячейкой для хранения задания.

Рис. 5. *Модель вычислительной системы*

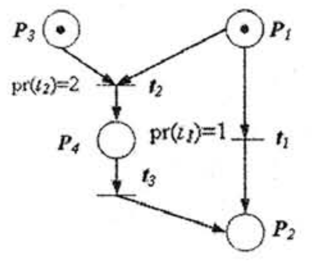


Живость. Переход t в сети Петри N называется *живым*, если $\forall M \in R(N), \exists M' \in R(N,M): M' \geq F(t)$, т.е. существует достижимая от M разметка M', при которой переход t может сработать. Сеть является *живой*, если все её переходы живы. Проблема живости сети сводится к проблеме достижимости некоторой разметки.

Алгоритм анализа свойства живости состоит в построении дерева достижимости. Поскольку число переходов сети N конечно, то живость сети можно установить, проверяя для каждого перехода, является ли он живым или потенциально мертвым (т.е. при любой разметке M' переход t не может сработать). Свойство живости позволяет выявить ситуации тупиковые для некоторых событий, т.е. ситуации, в которых после некоторой последовательности срабатываний переходов сети и соответствующих изменений ее разметки, некоторые переходы, в том числе те, которые уже срабатывали, больше не смогут сработать.

Рассмотрим модель появления исключительной ситуации [Тайлаков 1990: 25], применим её для процесса выполнения компьютерной программы. Переходу t_2 (Рис.6) присвоен более высокий приоритет, чем t_1 (pr(t_2)>pr(t_1)).

Рис. 6. *Модель появления исключительной ситуации в программе*



При нормальной работе программы сработает переход t_1 и маркер из p_1 (начального состояния участка программы) переместится в p_2 (конечное состояние). При возникновении исключительной ситуации (например, деления на ноль) срабатывает переход t_2. Маркеры переместятся в p_4, т.е. сработает блок обработки исключительных ситуаций, после чего программа продолжит свое выполнение (переход t_3). Сеть жива, если исключительная ситуация устранима и все операции выполняются.

Консервативность. $\sum_{p_i \in P} M(p_i) = \sum_{p_i \in P} M_0(p_i)$ Сеть Петри N с начальной маркировкой M_0 называется *консервативной,* если для всех $M \in R(N,M_0)$, т.е. если сумма маркеров во всех позициях при работе сети остается постоянной, и срабатывание перехода не меняет число фишек в сети.

Из определения следует, что число входов в каждый переход должно равняться числу выходов. Алгоритм анализа свойства консервативности состоит в построении дерева достижимости. Так как дерево достижимости конечно, для каждой маркировки можно вычислить взвешенную сумму. Если сумма одинакова для каждой достижимой маркировки, сеть - консервативная по отношению к данному весу. Если суммы не равны или в маркировке позиции имеется символ ω, сеть - неконсервативная.

Консервативность сети свидетельствует о невозможности уничтожения или возникновения дополнительных ресурсов. Применим свойство консервативности к задаче о чтении/записи [Питерсон 1984: 68], т.е. покажем, что фишки, представляющие ресурсы, никогда не создаются и не уничтожаются. Имеются процессы двух типов: процессы чтения и процессы записи (Рис. 7). Оба процесса совместно используют общий файл, переменную или элемент данных. Первоначально имеется 1 процесс чтения и 1 процесс записи. После того, как процесс чтения завершен, и общий ресурс освобожден, начинается процесс записи и т.д. Сеть консервативна, т.к. число фишек в сети остается постоянным.

Устойчивость. Переход t называется *устойчивым* в сети N, если

$\forall t' \in \setminus\{t\}, \forall M \in R(N) : (M \geq^{\bullet} F(t)) \wedge (M \geq^{\bullet} F(t')) \Rightarrow (M \geq^{\bullet} F(t) +^{\bullet} F(t'))$, т.е. если переход t может сработать, то никакой другой переход не может, сработав, лишить его этой возможности. *Сеть устойчива*, если все её переходы устойчивы. Алгоритм анализа свойства устойчивости состоит в построении дерева достижимости. Сеть устойчива, если из любой вершины дерева достижимости исходит только одна дуга, свидетельствующая о единственном возможном варианте разметки.

Устойчивость сети свидетельствует об отсутствии тупиковых ситуаций в процессе работы системы. Устойчивость характеризует точное и последовательное выполнение каждого процесса, который доводится до завершения, а новый процесс не запускается, пока не окончены все циклы первого.

Рассмотрим свойство устойчивости на примере модели сети Петри устройства управления асинхронной ЭВМ с конвейерной обработкой [Питерсон 1984: 49].

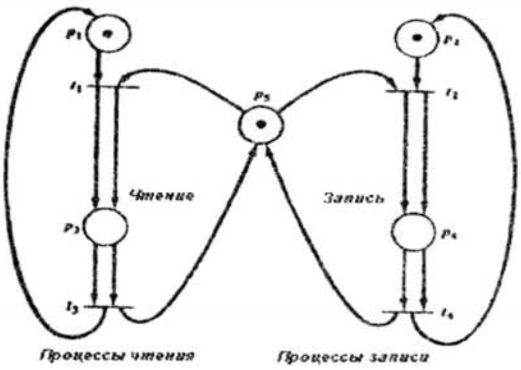


Рис. 7. *Моделирование процессов чтения и записи*

Результат шага k конвейерной обработки может быть послан на шаг *(k + 1)*, как только шаг k выполнен, а блок *(k + 1)* свободен (Рис. 8). Блок использует значение своего входного регистра для вычисления значения выходного регистра. Выходной регистр блока очищается путем пересылки содержимого во входной регистр следующего блока, затем новое входное значение появляется в его входном регистре. Сеть устойчива, т.к. завершение каждой операции строго предшествует началу последующей.

Рис. 8. *Модель устройства управления асинхронной ЭВМ с конвейерной обработкой*

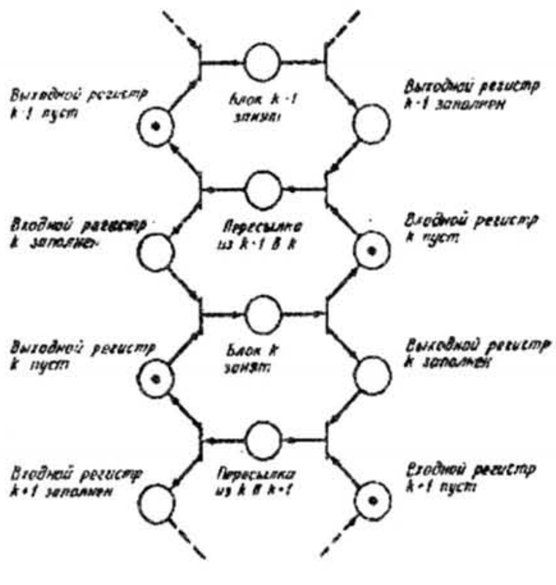
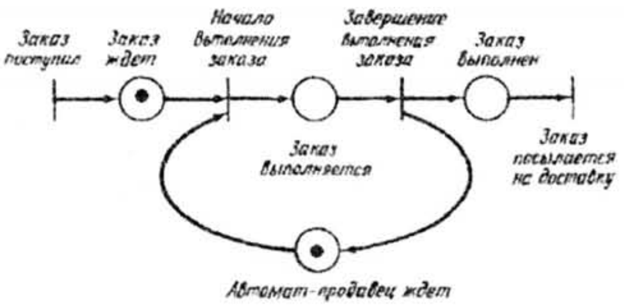


Рис. 9. *Модель работы автомата-продавца*



Например, каждый переход в модели работы автомата-продавца (Рис. 9) [Питерсон 1984: 37] соответствует завершению i-ой операции, начиная от поступления заказа и заканчивая отправкой заказа на доставку. Автомат-продавец находится в состоянии ожидания до тех пор, пока не появится заказ, который он выполняет и посылает на доставку. Сеть устойчива, когда завершение i-ой операции есть единственный исход её наступления.

Выводы

Установлено соответствие поведенческих свойств сети Петри параметрам технологических процессов сложных систем.

Анализ формальных свойств сетей Петри необходим для моделирования процессов, протекающих в сложных системах, т.к. позволяет исследовать поведение моделируемой системы и получить информацию о возможных исключительных ситуациях с целью корректировки модели и обеспечения рационального функционирования системы.

Список использованной литературы

1. **Котов, В. Е.** Сети Петри / В. Е. Котов. - М.: Наука, 1984. - 160 с.

2. **Питерсон, Дж.** Теория сетей Петри и моделирование систем / Дж. Питерсон: пер. с англ. М. В. Горбатовой, к.т.н. В. Л. Торхова, к.т.н. В. Н. Четверикова. – М.: Мир, 1984. – 264 с.

3. **Тайлаков, О. В.** Применение сетей Петри для моделирования гибких технологий подземных работ: Методические рекомендации / О. В. Тайлаков, В. Л. Конюх. – Кемерово: Институт угля Сибирского АН СССР, 1990. - 93 с.