

УДК 001.2: 165:167: 004.9

МОДЕЛИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СЕТЕЙ ПЕТРИ

Кудж С.А., д. тех.н., ректор, E-mail: mirearec1@yandex.ru
Логина А.С., магистрант, E-mail: melissa2033@rambler.ru
МГТУ МИРЭА, Москва, Россия

Аннотация. Статья описывает применение сетей Петри при моделировании процессов и при реализации информационного моделирования. Дается новая трактовка применения сетей Петри применительно к информационному моделированию. Описывается возможность применения сетей Петри для моделирования информационных ситуаций и информационных взаимодействий. Даются примеры применения сетей Петри для решения частных задач.

Ключевые слова: информация, моделирование, информационные технологии, информационные модели, информационные состояния информационное взаимодействие.

MODELING USING PETRI NETS

Kudzh S.A., D.ofSci.(Tech), rector, E-mail: mirearec1@yandex.ru
Loginova A.S., master student, E-mail: melissa2033@rambler.ru
MSTU MIREA, Moscow, Russia

Abstract. The article describes the use of Petri nets for modeling processes and the implementation of information modeling. Gives a new interpretation of the application of Petri nets in relation to information modeling. Describes the possibility of using Petri nets for modeling of information situations and information interactions. Give examples of the application of Petri nets for the solution of specific problems.

Keywords: information modeling, information technology, information model, information status information connectivity.

Введение. Теория сетей Петри делает возможным моделирование системы графическим представлением её в виде сети. Теория сетей Петри представляет собой механизм графической формализации процесса моделирования. Сети Петри — математический аппарат для моделирования динамических дискретных систем, впервые описанный Карлом Петри в 1962 году [1]. Он связан со взаимодействием событий в параллельных асинхронных дискретных системах и имеет сложную динамическую структуру. Информационные взаимодействия [2] описываются просто, если указывать не непосредственные связи между событиями, а те информационные ситуации [3], при которых данное событие может реализоваться. Информационные ситуации делятся на локальные и глобальные. Глобальные ситуации в системе формируются с помощью локальных операций, называемых условиями реализации событий. Предусловия события разрешают реализоваться некоторому событию, а

реализация события изменяет некоторые условия и создает постусловия события. В этом механизме события взаимодействуют с условиями, а условия взаимодействуют с событиями. Такая простая модель позволяет для решения задач и моделирования представить структуры систем из элементов двух типов – событий и условий. Удобный формальный механизм для этого, предложенный Петри, был развит А. Холтом, который назвал его сетью Петри [4]. В работе [5] сеть Петри связана с тензорным анализом. В данной работе показан более простой подход применения данного формализма для моделирования простых задач.

Формализм сетей Петри. В сетях Петри события и условия представлены абстрактными символами из двух непересекающихся алфавитов, называемых соответственно множеством переходов и множеством позиций. Имеется несколько формальных представлений сетей Петри: теоретико-множественное; графовое – бихроматический (двудольный ориентированный) граф; графическое; матричное.

Существуют разные виды сетей Петри: *Временная сеть* Петри — переходы обладают весом, определяющим продолжительность срабатывания (задержку). *Стохастическая сеть* Петри — задержки являются случайными величинами. *Функциональная сеть* Петри — задержки определяются как функции некоторых аргументов, например, количества меток в каких-либо позициях, состояния некоторых переходов.

Цветная сеть Петри — метки могут быть различных типов, обозначаемых цветами, тип метки может быть использован как аргумент в функциональных сетях. *Ингибиторная сеть* Петри — возможны ингибиторные дуги, запрещающие срабатывания перехода, если во входной позиции, связанной с переходом ингибиторной дугой, находится метка. *Иерархическая сеть* — содержит не мгновенные переходы, в которые вложены другие, возможно, также иерархические, сети. Срабатывание такого перехода характеризует выполнение полного жизненного цикла вложенной сети.

Сеть Петри состоит из трех элементов: множество мест S , множество переходов T и отношение инцидентности F . Кроме этого в них используют метки или метки (токены), которые обозначают черными кружочками и размещают в "местах" S . Эти метки характеризуют состояние системы, описываемое сетью Петри.

Применение сетей Петри к задачам моделирования. Сети Петри были разработаны и используются в основном для моделирования. С их помощью могут быть промоделированы многие системы в особенности системы с независимыми компонентами, например аппаратное и программное обеспечение ЭВМ, физические

системы, социальные и др. Сети Петри применяются для моделирования возникновения различных событий в системе. В частности, сети Петри могут моделировать поток информации и другие ресурсы системы.

Простое представление системы сетью Петри основано на двух основополагающих понятиях: событиях и условиях. События – это действия, имеющие место в системе. Возникновением событий управляет состояние системы. Состояние системы может быть описано множеством условий. Условие – предикат или логическое описание состояния системы. Условие может принимать, либо значение «истина», либо значение «ложь».

Так как события являются действиями, то они могут происходить. Для того чтобы событие произошло, необходимо выполнение соответствующих условий. Эти условия называются предусловиями события. Возникновение события может вызвать нарушение предусловий и может привести к выполнению других условий, постусловий.

В сети Петри условия моделируются позициями, события переходами. При этом входы перехода являются предусловиями соответствующего события; выходы – постусловиями. Возникновение события равносильно запуску соответствующего перехода. Выполнение условия представляется меткой в позиции, соответствующей этому условию. Запуск перехода удаляет разрешающие метки, представляющие выполнение предусловий и образует новые метки, которые представляют выполнение постусловий.

Одной из особенностей является свойственный сетям и их моделям параллелизм, или одновременность. В модели сети Петри два разрешенных невзаимодействующих события могут происходить независимо друг от друга. Синхронизировать события, пока это не требуется моделируемой системе, нет нужды. Но когда синхронизация необходима, моделировать её легко. Таким образом, сети Петри представляются идеальными для моделирования систем с распределенным управлением, в которых несколько процессов выполняются одновременно.

Другая важная особенность сетей Петри – это их асинхронная природа. В сети Петри отсутствует измерение времени или течение времени. Это отражает философский подход к понятию времени, с логической точки зрения, - это определение частичного упорядочения событий. В реальной жизни различные события укладываются в различные интервалы времени, и это отражено в модели сети Петри независимостью от времени управления последовательностью событий. Структура сети Петри такова, что содержит в себе всю необходимую информацию для определения

возможных последовательностей событий.

Выполнение сети Петри рассматривается как последовательность дискретных событий. Порядок выполнения событий является одним из возможных, допускаемых основной структурой. Это приводит к явной недетерминированности в выполнении сети Петри. Если в какой-то момент времени разрешено более одного перехода, то любой из нескольких возможных переходов может стать «следующим» запускаемым. Выбор запускаемого перехода осуществляется недетерминированным образом, т. е. случайно. Но все же выбор для запуска одного из разрешенных переходов детерминирован, но не на модели, просто потому что модель не дает полной информации о системе.

При анализе динамического поведения сети Петри, когда определяется последовательность запусков переходов, возникают некоторые трудности. Для простоты обычно вводят следующее ограничение. Запуск перехода (и соответствующего события) рассматривается как мгновенное событие, занимающее нулевое время, и возникновение двух событий одновременно невозможно.

Моделируемое таким образом событие называется примитивным (primitive events - PE) [6]. Примитивные события мгновенны и не одновременны. Это обосновывается тем, что время – непрерывная действительная переменная. Следовательно, если мы присвоим каждому событию время возникновения, то вероятность того, что любые две произвольно выбранные непрерывные действительные переменные совпадают, равна нулю, и следовательно, события не одновременны.

Не примитивными (not primitive events - NPE) называются такие события, длительность которых отлична от нуля. Они не являются не одновременными и, следовательно, могут пересекаться во времени. Не примитивное событие может быть представлено в виде двух примитивных событий: «начало не примитивного события», «конец не примитивного события» и условия «не примитивное событие происходит» [7].

Петри и другие [4-7] предложили представлять не примитивные события в сети Петри в виде прямоугольника, а примитивные планками. Это упрощает графический вид сети Петри. Прямоугольник может иметь значение при моделировании сложных систем на нескольких иерархических уровнях, он позволяет генерализовать в отдельный элемент целые подсети.

Не детерминированность и не одновременность запусков переходов в моделировании параллельной системы показываются двумя способами. Один из них представлен на рис. 1. В этой информационной ситуации два разрешенных перехода в

любом случае не влияют друг на друга, и в число возможных последовательностей событий входит последовательность, в которой первым срабатывает один переход, и последовательность, в которой первым будет запущен другой переход. Это называется одновременностью.

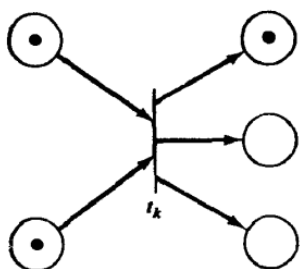


Рис. 1. Информационная ситуация "одновременность"

В исходном состоянии имеются два места S , переход обозначен t_k отношения инцидентности стрелками. Другая информационная ситуация, в которой одновременное выполнение затруднено и которая характеризуется невозможностью одновременного возникновения событий, показана на рис. 2. Здесь два разрешенных перехода находятся в конфликте. Может быть запущен только один переход, так как при запуске он удаляет метку из общего входа и запрещает другой переход.

Еще одним достоинством сетей Петри является то, что они моделируются с помощью достаточно простого языка, включающего простые (графические) информационные единицы [7]. Применение информационных единиц [8] позволяет использовать лингвистические модели для описания моделей и моделирования на этой основе.

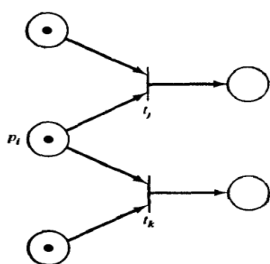


Рис. 2. Информационная ситуация "Конфликт"

Моделирование конечных автоматов с применением сетей Петри. *Конечный автомат* [10] – абстрактный автомат без выходного потока, число возможных состояний которого конечно. Результат работы автомата определяется по его конечному состоянию. Существуют различные варианты задания конечного автомата.

Например, конечный автомат может быть задан с помощью пяти параметров:

$$M = (Q, \Sigma, \delta, q_0, F),$$

где:

- Q - конечное множество состояний автомата;
- q_0 - начальное состояние автомата ($q_0 \in Q$);
- F - множество заключительных (или допускающих) состояний, таких что $F \subseteq Q$;
- Σ - допустимый входной алфавит (конечное множество допустимых входных символов), из которого формируются строки, считываемые автоматом;
- δ - заданное отображение множества $Q \times \Sigma$ во множество $\mathcal{P}(Q)$ подмножеств Q : $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$ (иногда δ называют функцией переходов автомата).

Важно отметить, что автоматы часто представляются в виде графов-переходов.

Моделирование в рамках сети Петри осуществляется посредством представления алфавитов автомата (входного или выходного) позициями или переходами (рис.3, Рис.4). Обычно входы представляются позициями. Приведём небольшой пример:

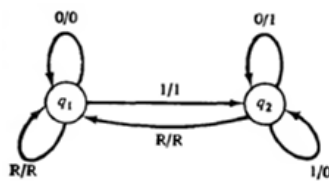


Рис. 3. Граф конечного автомата, вычисляющего дополнение до двух

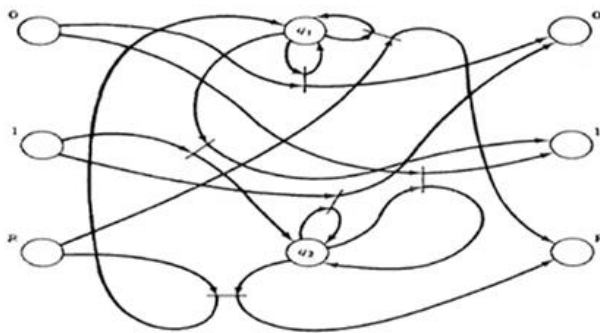


Рис. 4. Эквивалентная сеть Петри

Основным применением сетей Петри является возможность преобразования блок-схемы в сеть Петри. Поэтому в виде сети Петри можно представить любой готовый алгоритм, который построен на вычислениях и ветвлениях. Рис. 5 иллюстрирует правила перевода. Слева даны классические блок-схемы алгоритма, справа - их эквивалент в нотации сети Петри.

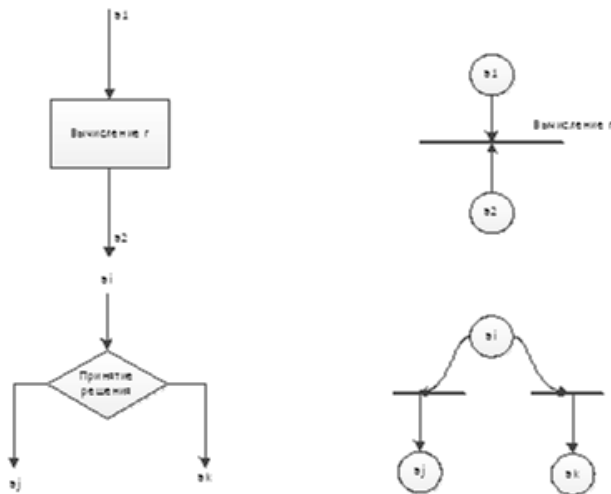


Рис. 5. Правила перевода блок-схемы в сеть Петри

Параллельные вычисления и синхронизация. Сети Петри являются инструментом моделирования процессов и состояний систем [11]. Большое применение сети Петри нашли при создании и выполнении параллельных ветвей различных вычислительных процессов. На рис. 6 изображена одна из типичных конструкций fork/join. Переход fork моделирует «разветвление» – создание из одной ветви выполнения двух параллельных ветвей. Это, как правило, реализуется путем создания одной дополнительной ветви вдобавок к существующей. Переход join, в свою очередь, осуществляет «слияние» двух ветвей по завершению их работы – уничтожение созданной параллельной ветви за ненадобностью.

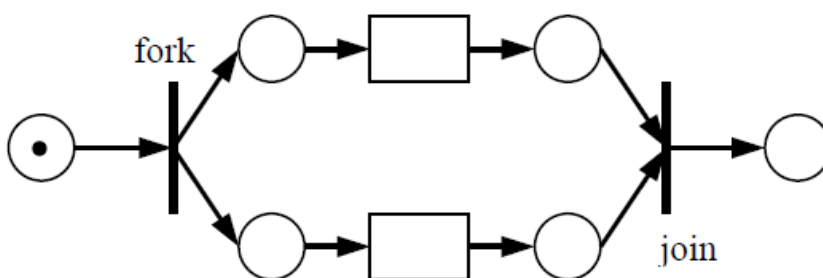


Рис. 6. Конструкция разветвления-слияния в формализме сети Петри.

Объекты синхронизации также легко моделируются сетями Петри. На рис. 7 изображен фрагмент сети Петри, моделирующий барьерную синхронизацию трех процессов.

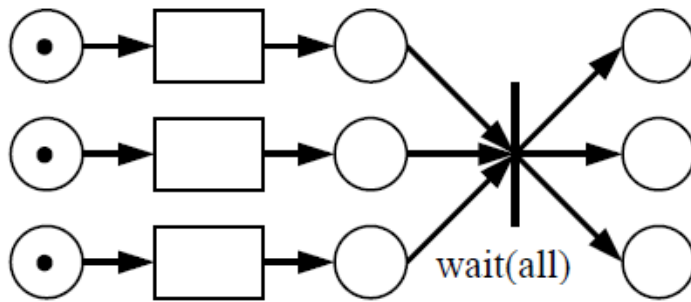


Рис. 7. Фрагмент сети Петри, моделирующий барьерную синхронизацию трех процессов

В начале работы сети разрешенными являются три длительных перехода, характеризующих выполнение некоторых подзадач. Три независимых процесса выполняют эти подзадачи, и лишь после их полного завершения становится разрешенным переход-барьер `wait(all)`. После его срабатывания все три процесса снова продолжают свою работу.

Похожим образом может быть смоделировано ожидание завершения любой из подзадач, первой завершившей свое выполнение. На рис. 8 показан такой фрагмент сети. При завершении выполнения любого длительного перехода становится разрешенным переход `wait(any)`. После его срабатывания дальнейшая работа продолжается, в то время как остальные подзадачи завершают свое выполнение. Дополнительная входная позиция перехода `wait(any)` защищает его от срабатывания при завершении остальных подзадач в случае, если в текущий момент ожидание не выполняется. В процессе дальнейшей работы сети метка в эту позицию возвращается, и тем самым разрешается ожидание и обработка результатов выполнения остальных подзадач.

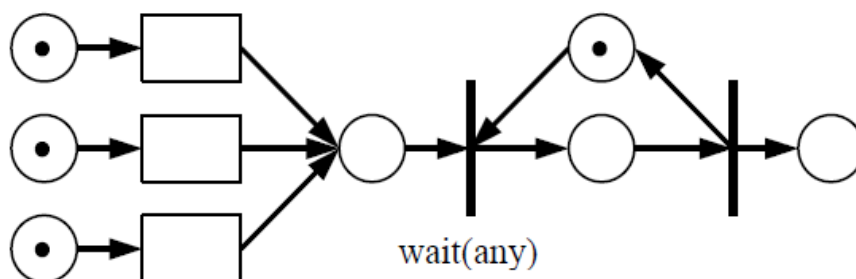


Рис. 8. Фрагмент сети, моделирующий ожидание завершения любой из подзадач

Другой пример синхронизации параллельных процессов – критическая секция.

Фрагмент сети на рис. 9 обеспечивает взаимоисключающий доступ к некоторому общему ресурсу для двух процессов. Первый процесс, захвативший доступ к критической секции, лишит метки соответствующую позицию *res*, тем самым запретив доступ к критической секции второму процессу до момента ее освобождения. В случае, когда в позиции *res* изначально более одной метки, такой фрагмент сети моделирует использование семафора – объекта синхронизации, позволяющего одновременный доступ к ресурсу нескольких процессов в количестве не более заданного.

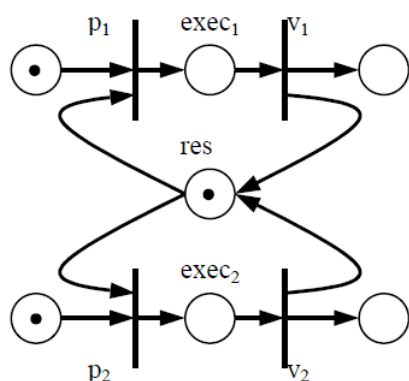


Рис. 9. Фрагмент сети обеспечивает взаимоисключающий доступ к некоторому общему ресурсу для двух процессов.

Важная особенность сетей Петри заключается в атомарности срабатывания перехода и, как следствие, возможности атомарного захвата одновременно нескольких ресурсов, что исключает взаимоблокировки процессов (dead lock).

Одной из классических задач, иллюстрирующих проблемы синхронизации процессов, является задача об обедающих философах. Ее постановка довольно проста и вкратце заключается в следующем. За круглым столом сидят пятеро философов, перед ними стоит блюдо спагетти. На столе лежат пять вилок - по одной между каждыми двумя соседними философами. По условию каждому философу для еды необходимо две вилки - лежащие непосредственно слева и справа от него. Каждый философ пребывает за столом в одном из двух состояний - размышляет или ест. В последнем случае оба его ближайших соседа размышляют, поскольку для еды им не хватает вилок.

Основная проблема, иллюстрируемая этой задачей - проблема возможности взаимоблокировки, которая была описана раньше. В случае реализации с последовательным захватом вилок возможна ситуация, когда одновременно все философы возьмут, к примеру, левую от себя вилку. Тогда ни один из них не сможет взять правую вилку, и каждый окажется заблокированным в ожидании ее

освобождения. На рис. 10 изображена сеть Петри, решающая задачу об обедающих философях.

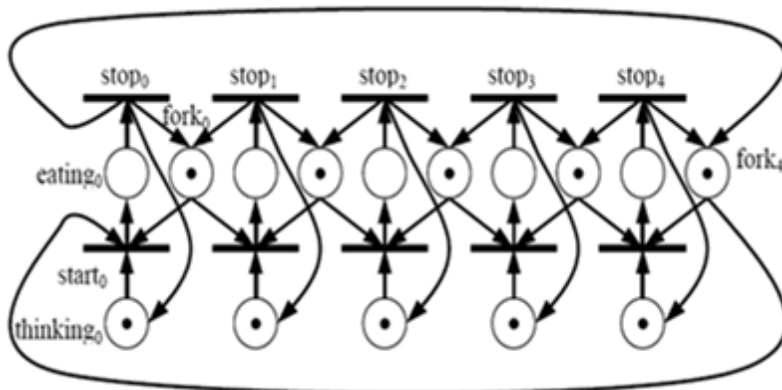


Рис. 10. Задача об обедающих философях

Позиции *eating* и *thinking* здесь характеризуют пребывание соответствующего философа в состоянии еды или размышлений. Позициями *fork* обозначается доступность вилок. Переходы *start* и *stop* характеризуют переход философа к еде и к размышлениям соответственно.

Однократный процесс приема пищи каждым философом представляет собой длительную операцию и может быть представлен в виде длительного составного перехода. В этом случае срабатывание перехода *stop* характеризует завершение приема пищи философом и, соответственно, завершение длительного перехода. Одним из довольно простых решений этой проблемы, хотя, конечно, далеко не самым лучшим, является следующее. К сети, изображенной на рис. 11, добавим барьерную синхронизацию после того, как каждый философ осуществит прием пищи по одному разу.

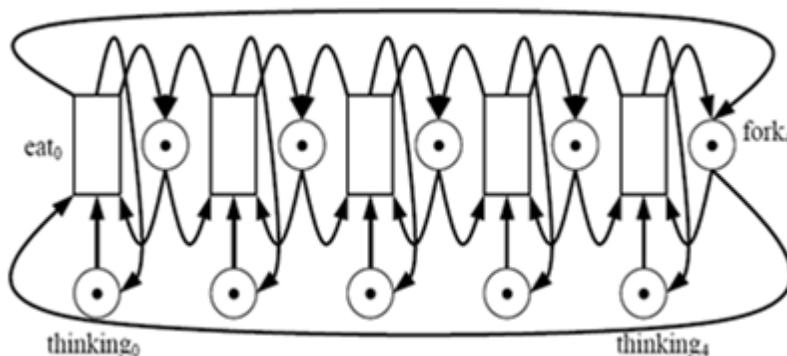


Рис. 11. Другое представление задачи

Для этого добавим к каждому философу по две позиции - позицию *todo*, количество фишек в которой отражает количество предстоящих подходов философа к еде до следующего барьера, и позицию *done*, отражающую количество выполненных подходов. Полученная сеть изображена на рис. 12.

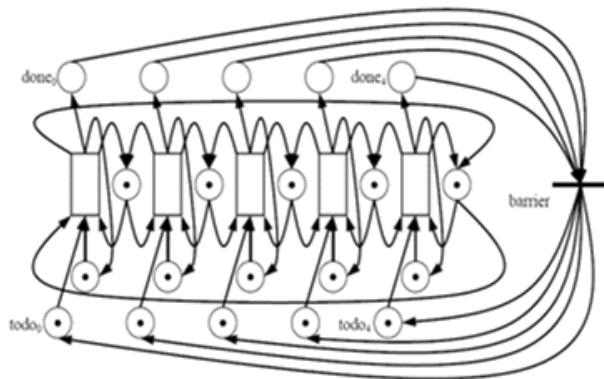


Рис. 12. Решение задачи вместе с барьером синхронизации

В случае если синхронизация после каждого подхода к еде является слишком частой мерой, количество фишек в каждой позиции *todo* может быть увеличено. Более того, начальные количества фишек в этих позициях могут быть заданы разными, в зависимости от потребности в интенсивном питании каждого из философов. В соответствии с начальным количеством фишек в позициях *todo* должна быть изменена и кратность дуг, ведущих в переход-барьер и из него.

С учетом такой модификации сети каждый философ будет получать доступ к еде в обычном порядке, пока не закончатся метки в соответствующей позиции *todo*, после чего остановится на размышлениях. Таким образом, пока каждый из пяти философов не получит доступ к еде заданное количество раз, переход к следующему циклу осуществлен не будет. Когда все метки из позиций *todo* постепенно переместятся в соответствующие позиции *done*, переход-барьер снова переместит их в позиции *todo*, после чего начнется следующий цикл приема пищи.

Заключение. Теория сетей Петри, хотя и возникла как самостоятельная теория [12], отражает некоторые современные общие тенденции анализа и моделирования. Первая тенденция это применение графических обозначений для формализации процессов и состояний систем. Близким аналогом этому является унифицированный язык визуального моделирования UML [13]. Вторая тенденция это применение информационных единиц как основы информационного языка и основы построения информационных конструкций [14] и информационных моделей [15].

Теория сетей Петри является популярным формализмом, предназначенным для

работы с параллельными и асинхронными системами [16]. Она содержит большое количество методов и средств анализа, позволяющих применять ее при моделировании и анализе ряда сложных систем и структурных логических моделей [17] сложных и информационных систем. Основными особенностями этой графической модели является возможность моделирования состояний и переходов между ними. Сеть Петри позволяет проводить анализ процессов за счёт разделения на позиции и переходы. Одна схема может многократно использоваться для описания разных состояний или разных информационных ситуаций. Именно на аспект информационного взаимодействия обращена данная работа. Это поможет в организации информационного моделирования на основе применения сетей Петри.

Список литературы

1. Petri, C. (1962), *Kommunikation mit Automaten*, PhD thesis, University of Bonn, Germany.
2. Tsvetkov V. Ya. Information interaction // *European Researcher*, 2013, Vol.(62), № 11-1, p.2573- 2577.
3. Tsvetkov V. Ya. Information Situation and Information Position as a Management Tool // *European Researcher*, 2012, Vol.(36), № 12-1, p.2166- 2170.
4. Meldman J. A., Holt A. W. Petri nets and legal systems // *Jurimetrics journal*. – 1971. – С. 65-75.
5. Кулагин В.П., Цветков В.Я. Философия сетей Петри // *Вестник МГТУ МИРЭА «MSTU MIREA HERALD»* 2014 - № 4 (5) - с.18-38.
6. Peterson J. L. Petri nets // *Computing Surveys*, Vol 9, No. 3, September 1977. - pp.223-252.
7. Ghanem N. et al. Representation and recognition of events in surveillance video using petri nets // *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on.* – IEEE, 2004. – С. 112-112.
8. Tsvetkov V. Ya. Information Units as the Elements of Complex Models // *Nanotechnology Research and Practice*, 2014, Vol.(1), № 1. p57-64.
9. Ozhereleva T. A. Systematics for information units // *European Researcher*, 2014, Vol.(86), № 11/1, pp. 1894-1900.
10. Alur R., Dill D. L. A theory of timed automata // *Theoretical computer science.* – 1994. – V. 126. , №. 2. –pp.183-235.
11. Цветков В.Я. Моделирование в автоматизации научных исследований и проектировании.- М.: ГКНТ, ВНИИЦентр, 1991. – 125с.

12. Котов В.Е. «Сети Петри» М.:Наука, 1984.-160с.
13. Fowler M. UML distilled: a brief guide to the standard object modeling language. – Addison-Wesley Professional, 2004.
14. Tsvetkov V. Ya. Information Constructions // European Journal of Technology and Design, 2014, Vol.(5), № 3- p147-152.
15. Breeden D., Viswanathan S. Why do firms hedge? An asymmetric information model //Fuqua School of Business, Working Paper. – 1998.
16. А.А. Лескин, П.А. Мальцев, А.М. Спиридонов «Сети Петри в моделировании и управлении» Л.: Наука, 1989.-133с.
17. Tsvetkov V.Ya. Logic units of information systems // European Journal of Natural History. – 2009. – № 2 . – p 99-100.