

## Представление имитационных моделей дискретно – событийных систем с помощью CASE средства E-NetSim

### Аннотация

*Рассмотрены существующие программно-технические комплексы, реализующие анализ исследуемой системы при помощи теории Оценочных Сетей. Описана система имитационного моделирования E-NetSi., Рассматриваемое CASE средство предназначено для создания и исследования имитационных моделей дискретно-событийных систем, представленных в виде Оценочных Сетей Петри. Показан механизм взаимодействия пользователя с системой при создании моделей и задании ее основных параметров. Показана возможность простого, интуитивно понятного описания сложных функциональных и логических связей в создаваемой модели.*

### Ключевые слова

Программно-технический комплекс, CASE-средство, Сети Петри, Оценочные Сети, моделирующая мощьность, диалоговый интерфейс, система имитационного моделирования.

### 1. Постановка проблемы в общем виде

Создание сложных распределенных информационно-технических систем невозможно без глубокого изучения процессов взаимодействия между отдельными составляющими в таких системах. Характерными особенностями таких систем являются глубокий параллелизм, взаимная неопределенность состояний, отсутствие однозначной зависимости между событиями и условиями их порождающими, многообразие вариантов возможного развития каждого процесса и системы в целом [ ].

Использование сетей Петри, особенно их расширений E-сетей, по мнению группы разработчиков, позволяет приблизиться к созданию инструментального средства, позволяющего с единых позиций создавать комплексное описание системы, проводить ее исследование и выработать рекомендации по дальнейшему развитию [ ].

В настоящее время разработано множество программных средств, позволяющих производить анализ сложных систем, базируясь на теории сетей Петри и их расширениях [ ], однако для пользования ними разработчик помимо знания самой методологии СП должен разбираться в области программирования, что в значительной мере ограничивает область их использования. Таким образом, задача разработки функционально мощного и в тоже время предельно простого и понятного программно-технического комплекса, позволяющего проводить анализ системы при помощи E-сетей, является актуальной.

### 2. Анализ последних исследований

Анализ существующего программного обеспечения, позволяющего проводить исследования системы с использованием теории Оценочных Сетей, показал, что на данный момент не существует относительно простого для пользователя программного средства [1-3]. Существует множество интересных средств, как правило, разработанных в 80х-90х годах, функционально мощных, однако не обладающих гибким графическим интерфейсом и уже давно морально устаревших. Современное программное обеспечение носит скорее демонстративный характер, единственным применением которого может быть визуализация принципов работы Оценочных сетей в образовательных целях для большей наглядности. Другими распространенными средствами имитационного моделирования в наше время являются Языки имитационного моделирования (ЯИМ)[4], такие как: GPSS, Simula, SimScript проч. Также был рассмотрен программный комплекс MATLAB [5,6], в основу которого положен язык высокого уровня для выполнения технических и научных вычислений. Обладая высокой моделирующей мощью, функциональным, математическим аппаратом, комплекс является одним из лучших средств исследования различных систем. Однако, наряду с достоинствами комплекса одним из очевидных недостатков является сложность и длительность изучения.

### 3. Цели и задачи исследования

Целью работы является описание моделирующей мощности разработанного ранее программно-технического комплекса E-NetSim, раскрывающего имитационные возможности математического аппарата E-сетей.

#### 4. Основной материал

Мобильная инструментальная система имитационного моделирования предназначена для создания и исследования имитационных моделей дискретно-событийных систем. В настоящее время разработано множество программно-технических комплексов, позволяющих производить анализ сложных систем, базируясь на теории Сетей Петри и их расширений, однако для пользования ими разработчик помимо знания самой методологии Оценочных Сетей (ОС) должен разбираться в области программирования, что в значительной мере ограничивает область их использования. Основным условием использования рассматриваемой системы имитационного моделирования является то, что пользователь должен быть квалифицирован в области исследуемой системы, и быть знакомым с принципами построения моделей при помощи оценочных сетей.

Модель создается на базе основных элементов оценочных сетей: *позиции(условия), переходы (действия), дуги, маркеры.*

Состояние системы отображается в модели совокупностью выполнившихся условий. Выполнение условия моделируется маркировкой соответствующей позиции. Происходящие в системе события в модели представляются с помощью переходов, которые срабатывают под воздействием одного или нескольких условий. Основными параметрами перехода являются: *время работы, условия для срабатывания, функция преобразования, тип перехода*

Время работы перехода – количество тактов моделирования, которые необходимы для переноса маркеров из входных позиций перехода в выходные.

Функция преобразования является математической зависимостью, выполняющей преобразования над заданными атрибутами маркеров, при срабатывании перехода.

Все переходы, используемые в оценочных сетях, делятся на два вида: *безусловные (переходы типа T, F, J) и условные (переходы типа X и Y).*

Дуги служат для образования логических связей типа «предусловие» и «постусловие» для переходов.

Маркеры, которые перемещаются по сети, кроме отображения выполнения условия могут отождествляться с объектами существующими в исследуемой системе и нести в себе один или более параметров, каждый из которых может принимать дискретный набор значений (атрибутов). В

предлагаемой системе моделирования предусмотрена возможность гибкого задания количества атрибутов маркеров и их значений.

Существует несколько подходов формального представления E-сети, будем использовать следующее представление:

$$E = \langle (P, B, R), T, I(T), O(T), Z, V, Q, \psi, M \rangle,$$

где:

$P = \{P_i\}$  - конечное непустое множество позиций;

$B = \{b_k\} \in P$  - множество периферийных, не внутренних:  $I(b_k) = 0$  либо  $O(b_k) = 0$  и  $I(b_k) \cup O(b_k) \neq 0$  позиций, которые используются в сети для определения связей с внешней средой;

$R = \{r_m\} \subset P$  - множество решающих позиций, на которых определяется множество Q;

$T = \{t_n\}$  - непустое конечное множество переходов;

$I(T), O(T)$  - функция связи переходов и позиций соответственно по входам и выходам;

$z: T \rightarrow R^+$  - функция с помощью которой задается время выполнения переходов;

$v = \{V_{is}\}$  - конечное непустое множество переменных (атрибутов маркера), изменяющих свои значения при выполнении сети;

$Q = \{q_n\}$  - конечное множество решающих процедур, с помощью которых задаются правила выполнения условных переходов;

$\psi = \{\psi_n\}$  - множество процедур перехода, выполняющих преобразования над атрибутами маркеров;

$M = (M_0(p_i)), p_i \in P, M_0(p_i) \in \{0, 1\}$  - начальная маркировка сети.

Переход срабатывает при наличии маркера во входной позиции и отсутствии его в выходной позиции. На примере безусловного T перехода это продемонстрировано на рисунке 4.1. Формально это можно записать так  $M_1(p_1)\{0\}, M_1(p_2)\{1\}$

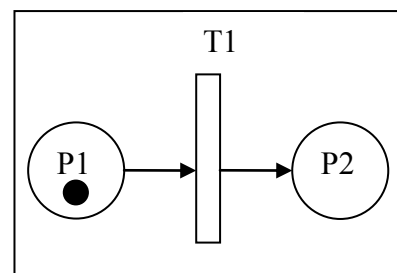


Рисунок 4.1 – T-переход

Формальное описание процесса преобразования значений атрибутов инициализирующего маркера представлено ниже:

$$V_{is} = \begin{cases} \Psi_k(R), k = s \\ V_{is}, k \neq s \end{cases}, s = \overline{1, n}, \quad (4.1)$$

где:

R-параметры, от которых зависит функция преобразования значения атрибута  $\Psi_k$ , В качестве параметра может выступать: значение изменяемого

атрибута, значения других атрибутов маркера, параметром может быть случайная величина с известным законом распределения;

$V_{is}$  - значение  $s$ -го атрибута маркера, находящегося в позиции  $i$ ;

$n$  - количество атрибутов маркера, находящегося в позиции  $i$ .

Время работы перехода зависит от функции  $z: T \rightarrow R^+$ . Предлагается в CASE-средстве реализовать возможность гибкого управления временем работы перехода. Для этого можно учитывать значения атрибутов маркеров, которые активизируют переход. При этом появляется возможность изменять время работы переходов по заданному закону, что в значительной мере повышает моделирующие возможности системы.

Условные переходы типа X и Y помимо перечисленных особенностей в своей структуре имеют условный переключатель, задающий направление ветвления или выборки в зависимости от условия. Рассмотрим процесс срабатывания X перехода, представленного на рисунке 4.2. В общем случае разрешающая процедура может быть сколь угодно сложной, но сущность ее работы заключается в проверке условий разветвления потока маркеров.

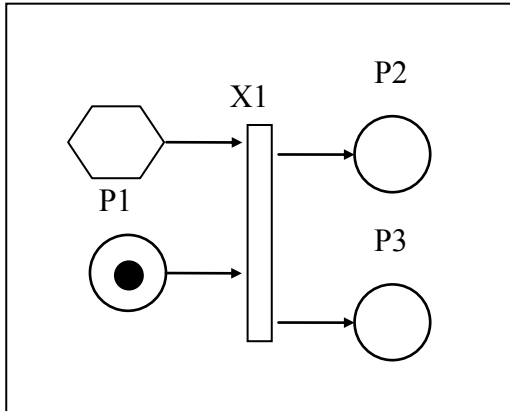


Рисунок 4.2 – переход типа X

Ниже приведено формальное описание работы X перехода в случае, когда выходные позиции не содержат маркеров:

$$\begin{cases} M_1(p_1) = M_0(p_1) - 1 \\ M_1(p_2) = M_0(p_2) + 1 \\ M_1(p_3) = M_0(p_3) \end{cases} \quad (4.2)$$

$$, \forall V_{1s} \in V : q_1(q_1(p_2), \dots, q_n(p_2)) \in Q, \quad s = 0, n'$$

где:

$n'$  - количество проверяемых атрибутов (условием

для маркирования определенной позиции могут служить значения нескольких атрибутов иницирующего маркера);

$q_1$  - конечное множество решающих процедур  $q_i(p_i)$ , одним из параметров которых является позиция, которая будет маркирована в результате работы перехода и удовлетворения условий.

При функционировании перехода типа Y, рисунок 4.3, наблюдается противоположная ситуация. Когда маркирована только одна из входных позиций  $M_x(p_i) = 1, p_i \in I(t_m)$ , где  $t_m$  - описываемый переход типа Y, работа перехода схожа с работой T-перехода. Сложность возникает в случае, когда маркированы две и более позиций, связанных по входам, и состоит в определении очередности извлечения маркеров.

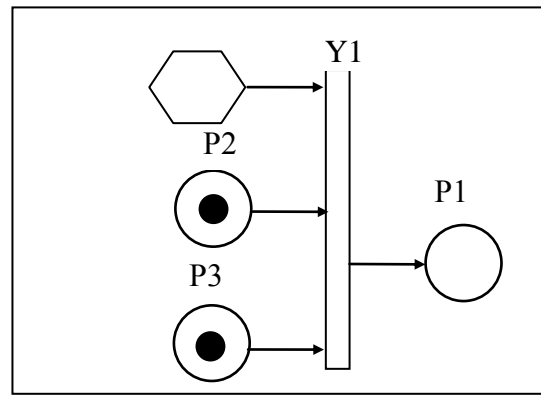


Рисунок 4.3 – переход типа Y

Формальное определение решения такой ситуации представлено ниже:

$$\begin{cases} M_1(p_1) = M_0(p_1) + 1 \\ M_1(p_2) = M_0(p_2) - 1 \\ M_1(p_3) = M_0(p_3) \end{cases} \quad (4.3)$$

$$, \forall V_{2s} \in V : q_1(q_1(p_1), \dots, q_n(p_1)) \in Q$$

$$, \forall V_{3s} \in V : q_1(q_1(p_1), \dots, q_n(p_1)) \in Q, \quad s = 0, n'$$

Приведенный подход довольно трудоемок, в связи с чрезвычайно большим объемом информации, которую необходимо ввести пользователю при задании параметров модели. В большинстве известных системах имитационного моделирования на базе сетей Петри задание этих параметров осуществляется с помощью специализированного языка, что требует специальных навыков. В предлагаемой системе имитационного моделирования используется интерактивная схема описания модели. Суть ее заключается в объединении графического и математического методов. При помощи графического редактора из готовых простейших элементов E-сети создается модель исследуемой системы, а при помощи

окон свойств элементов, вносится информация о функциональных и параметрических характеристиках каждого объекта в модели.

Средство обладает гибким графическим интерфейсом, позволяющим продемонстрировать динамическое изменение положения того или иного маркера относительно позиций и обеспечить хорошую визуализацию процессов происходящих в модели. Пользователю достаточно правильно выделить «причинно-следственные» связи в исследуемом объекте и создать имитационную модель.

Задание математических соотношений, определяющих поведение отдельных элементов моделируемой системы пользователем, также максимально упрощено и сведено к некоторому набору простых, логически понятных, действий. Например, для того, чтобы описать срабатывание перехода под воздействием некоторых факторов, достаточно просто определить решающую позицию и задать условие, или некоторую совокупность условий, приводящих к запуску перехода. При этом сами условия задаются при помощи простых математических соотношений и неравенств. То же самое можно сказать и о функции перехода, то есть преобразованиях, выполняемых над атрибутами маркера, вследствие срабатывания перехода. Существует также возможность динамического изменения времени срабатывания перехода. Для этого в свойствах перехода в поле «время срабатывания» необходимо ввести переменную, которая является атрибутом маркера, инициирующего работу перехода. После этого при попадании маркера во входную позицию такого перехода происходит динамическое изменение времени работы перехода.

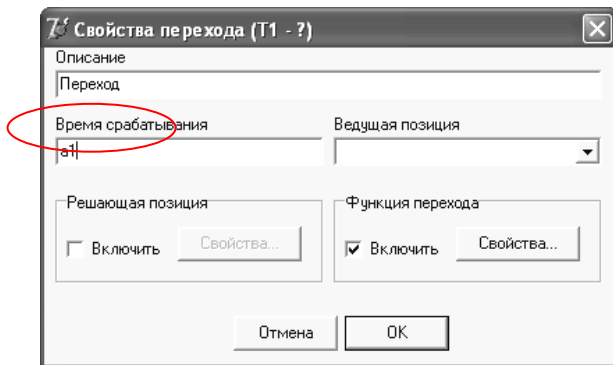


Рисунок 4.4

Основными источниками маркеров в системе являются генераторы. Генератор представляет собой базовый элемент теории E-сетей и является совокупностью простых элементов. Схематическое изображение Генератора представлено на рисунке 1.4

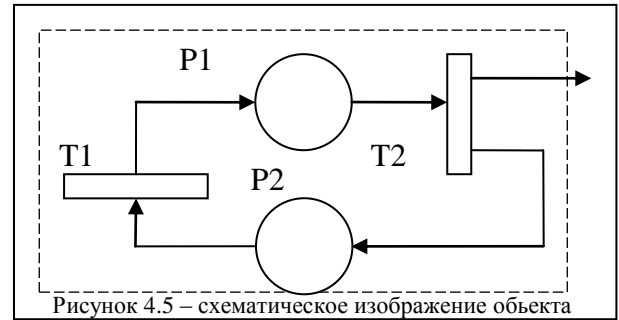


Рисунок 4.5 – схематическое изображение объекта

Генератор

В предлагаемом CASE-средстве для инициализации генератора, необходимо задать характеристики временного распределения и характеристики распределений атрибутов маркеров, создаваемых генератором. Количество атрибутов, при этом, неограниченно. В системе реализовано задание распределений, таких как экспоненциальное, равномерное, нормальное, также значение определяемого параметра может быть константой.

После того как модель построена и заданы математические соотношения, определяющие все её параметры, проводится проверка на корректность соединения элементов модели. В случае если будут найдены ошибки в определении элементов модели, в отдельном окне выводится соответствующая информация. Процесс управления моделированием также довольно гибок. В системе имеется возможность выбора нескольких режимов моделирования: пошаговый, позволяющий эксперту произвести «доводку модели» и проверку на правильность функционирования; моделирование выбранного количества шагов; прогон модели до заданного события, условия или момента времени. В последнем случае возможно установить требуемое время выполнения одного такта, тем самым ускоряя или замедляя процесс моделирования.

В конце эксперимента с моделью происходит генерация календаря событий, который хранит в себе полную информацию о всех имевших место событиях во время моделирования. Обработывая отчетную информацию, эксперт может получить интересующие его сведения, начиная от вероятности пребывания системы в том или ином состоянии до временных характеристик элементов исследуемой системы.

Как уже было сказано ранее, для описания модели используются математический и графический методы. В соответствии с ними, в средстве реализован графический редактор и набор диалоговых форм, позволяющих задать характеристики элемента в понятной для пользователя форме. Рассмотрим более подробно процесс описания модели при помощи диалоговых форм.

Основную часть главного окна средства занимает рабочая область редактора сети. На ней пользователь

при помощи кнопок управления и базовых элементов Оценочных Сетей методом Drag&Drop формирует структуру имитационной модели исследуемой системы или процесса. Чтобы добавить элемент на рабочую область редактора сети пользователю необходимо воспользоваться следующим меню, представленным на рисунке 4.6.

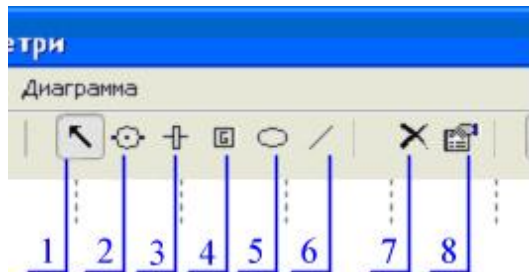


Рисунок 4.6

,где

1 – выбор объекта и перемещение по области редактора;

- 2 – добавление позиции;
- 3 – добавление перехода;
- 4 – добавление генератора;
- 5 – добавление очереди;
- 6 – добавление дуги (соединение разнотипных элементов);
- 7 – удаление выбранного объекта либо группы объектов;
- 8 – свойства выбранного объекта.

После того, как требуемый объект расположен в рабочей области, его можно перемещать, масштабировать, вращать, а также выполнять редактирование свойств элемента при помощи диалогового интерфейса, вид которого будет различен в зависимости от выбранного элемента.

На рисунке 4.7 представлена взаимосвязанная иерархия форм, используемых при задании свойств позиции:

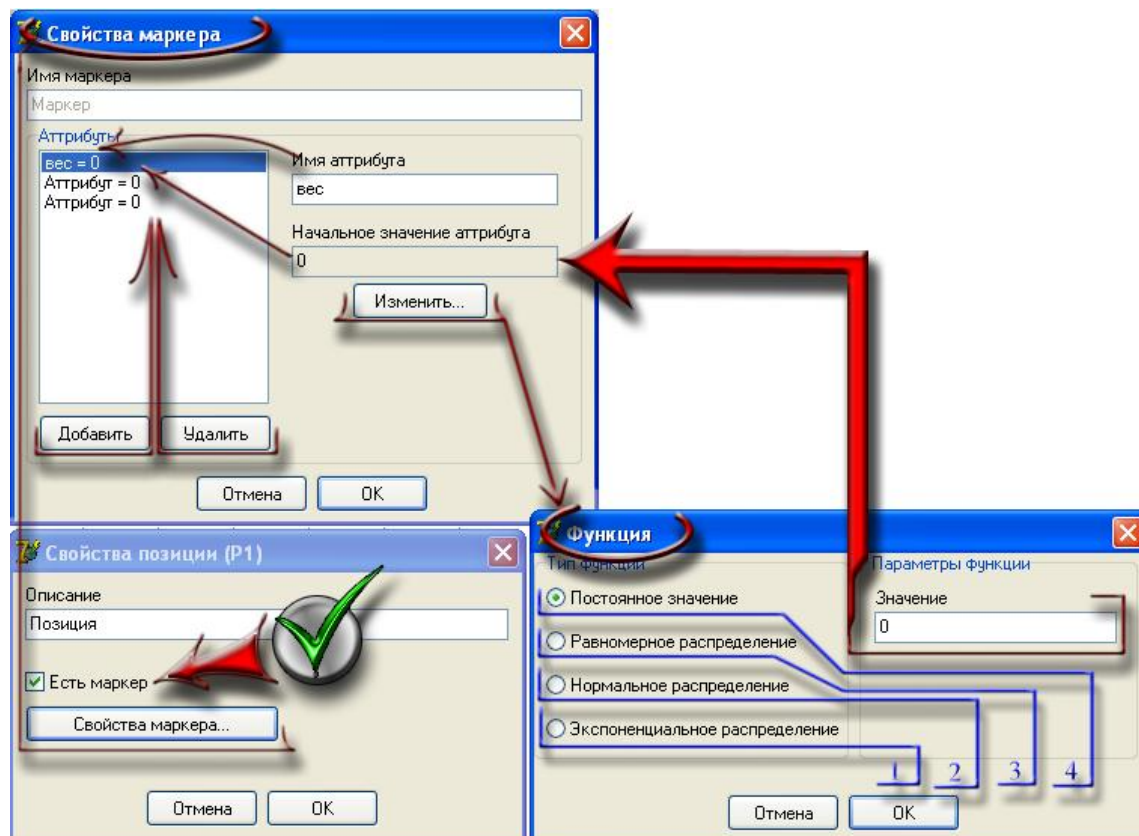


Рисунок 4.7 – задание свойств элемента Позиция

При инициализации редактирования свойств позиции на экране возникает диалоговая форма «Свойства позиции», позволяющая задавать смысловое описание позиции, а также добавить маркер стартовой разметки сети. Если в позицию принудительно помещается маркер, то возможна модификация его параметров. Для этого достаточно при помощи кнопки «Свойства маркера» вызвать одноименную форму.

Поле Имя маркера является неизменным, так как в процессе моделирования имена маркерам присваиваются автоматически и отличаются номером идентификатором. В левой области окна отображается список атрибутов маркера. Редактирование свойств атрибута выполняется при помощи двух полей – Имя атрибута и Начальное значение. Поле Имя атрибута позволяет дать смысловое описание характеристик маркера, зачастую заимствованное из терминологии из

предметной области исследуемой системы. Редактирование поля Начальное значение атрибута осуществляется при помощи формы «Описание функции», которая позволяет установить тип значения и его параметр. В средстве реализовано 3 наиболее часто используемых законов распределения случайных величин и функция постоянного значения. Выбор соответствующего типа функции осуществляется при помощи управляющих кнопок, представленных на рисунке как:

1 – значение атрибута является стохастической величиной, распределенной по экспоненциальному закону с соответствующим параметром;

2 – значение атрибута является стохастической величиной, распределенной по нормальному закону с соответствующими параметрами математического ожидания и дисперсии;

3 – значение атрибута является стохастической величиной, распределенной по равномерному закону с соответствующими параметрами минимального и максимального значений;

4 – значение атрибута является детерминированным.

Задание свойств генератора маркеров выполняется с помощью иерархии экранных форм представленных на рисунке 4.8.

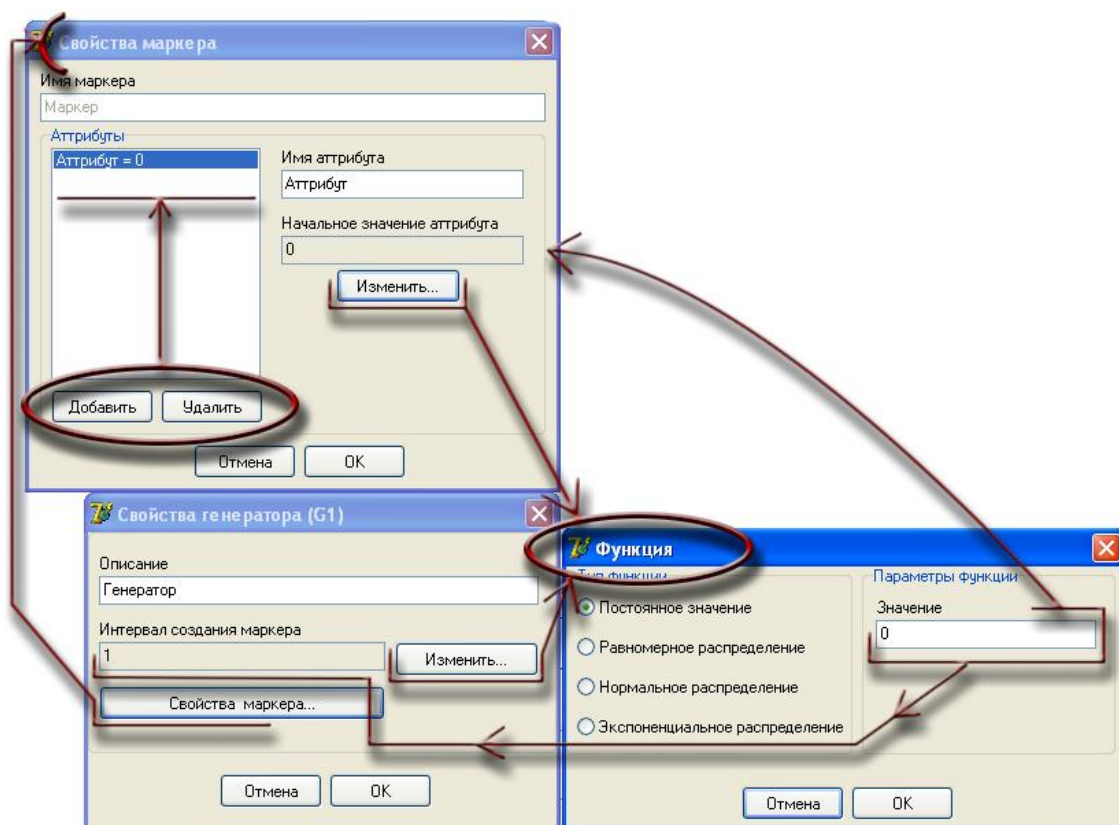


Рисунок 4.8 – задание свойств объекта Генератор

Для задания свойств генератора маркеров, необходимо указать интервал времени между появлением маркеров на выходе генератора, а также определить набор атрибутов характеризующих каждый создаваемый маркер. Значением поля Интервал создания маркера может также быть либо константа, либо случайная величина, соответствующая одному из трех законов распределения. Редактирование данного поля происходит при помощи описанного ранее диалога, представленного на рисунке 5. Аналогичным способом выполняется задания свойств генерируемых маркеров.

Объект Очередь в модели описывается с помощью форм, представленных на рисунке 4.9.

Очередь характеризуется размером, это максимальное число помещаемых в неё маркеров, и типом диспетчеризации это метод изъятия маркеров из очереди. Имеется задавать три наиболее часто встречаемых метода освобождения очереди – FIFO, LIFO, HPF. В отдельной области выводится информация о маркерах, присутствующих в очереди, а при помощи трех управляющих кнопок возможно добавление маркеров в очередь и редактирование значений их атрибутов при создании стартовой разметки сети.

На рисунке представлены по пунктам управляющие кнопки и поля, среди которых:

- 1 – тип диспетчеризации HPF;
- 2 – тип диспетчеризации LIFO;

3 – тип диспетчеризации FIFO;

4 – размер очереди, устанавливаемый перед началом процесса проигрывания модели.

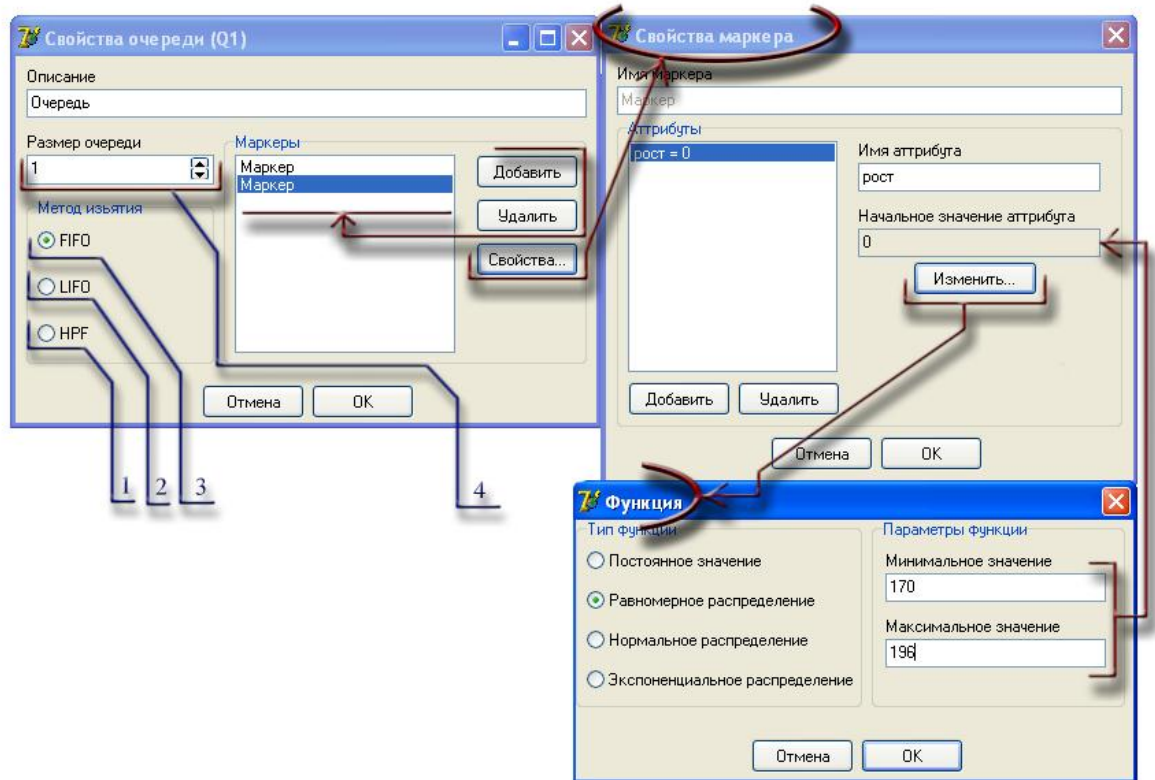


Рисунок 4.9 – задание свойств объекта Очередь

Описание объекта Переход может быть осуществлено при помощи следующего набора форм представленных на рисунке 4.10.

Характеристиками объекта Переход являются: время срабатывания Перехода это время, от момента инициализации работы, до срабатывания перехода; функция перехода это правило преобразования атрибутов маркера, выполняемое в результате запуска

перехода; решающая позиция это определение функции ветвления или выборки, для разрешения конфликтных ситуаций в переходах условного типа. Ведущая позиция задает номер позиции, из которой маркер переносится в выходную позицию перехода типа F. Время работы каждого перехода может быть величиной постоянной или изменяться динамически в процессе моделирования.

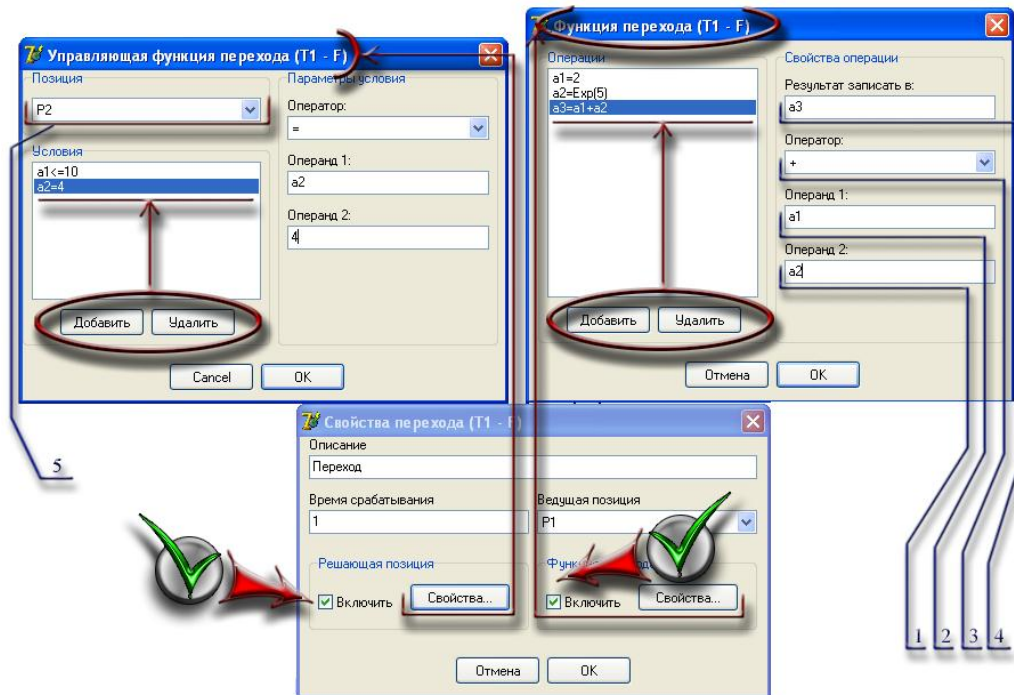


Рисунок 4.10 – задание свойств объекта Переход

Рисунок 4.11

Формы задания правил преобразования атрибутов маркера, при срабатывании перехода, и определения функции задающей правила работы условных переходов, схожи по своему интерфейсу. Для формирования функции необходимо воспользоваться специальными полями, представленными на рисунке 4.10, где:

- 1 – второй операнд;
- 2 – первый операнд;
- 3 – оператор преобразования, принимающий, в ситуации с функцией преобразования математический вид, а в ситуации управляющей функции – логический;
- 4 – результат, куда будет занесено полученное значение. В случае с управляющей функцией поле отсутствует.

Возможна гибкая модификация значения указанного атрибута маркера в соответствии со значением атрибута не входящего в состав данного маркера, а находящегося в другой позиции. Для этого в поле Операнд указывается составной путь к атрибуту, состоящий из: номера позиции и уникального идентификатора атрибута, разделенных точкой. Иллюстрация данного примера приведена ниже

Функционально форма «Управляющая функция перехода» разбита на три области: Позиция – привязка условий осуществляется с учетом позиций которые будут либо маркированы в результате конфликта, либо из которых будет изъят маркер; Условия – собственно конечное множество условий, разрешающих конфликтную ситуацию; область Оператор – как и в предыдущем диалоге (рисунок 4.11) выполняет функцию составления условий. При составлении функций разрешения конфликтных ситуаций также возможно применение составного пути, то есть когда проверяемое значение не принадлежит атрибуту маркера, инициировавшего работу перехода.

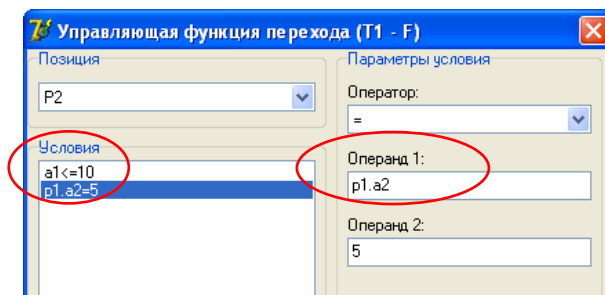
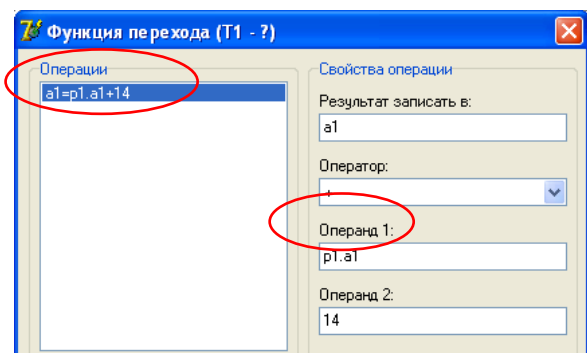


Рисунок 4.12



## 5. Выводы

Поведенный анализ позволяет сделать вывод о том, что использование Оценочных Сетей при исследовании сложных систем открывает возможности комплексного описания изучаемого объекта. Существующие средства программного представления моделей в виде E-сети

сложны в использовании и требуют специальной программной подготовки от пользователя. В работе описано CASE-средство предназначенное для создания и исследования имитационных моделей дискретно-событийных систем в виде оценочных сетей Петри. Приведено описание разработанного программно-

технического комплекса, описаны его функциональные и моделирующие возможности, показана диалоговая система взаимодействия с пользователем. В настоящее время происходит модернизация комплекса, путем добавления новых функциональных возможностей.

#### Список литературы

1. Питерсон Дж. Теория сетей Петри и моделирование систем. Пер. с англ. М.: Мир, 1984 г., -264с.
2. Котов В.Е. Сети Петри. – М.: Наука, 1984. – 160 с.
3. Костин А. Е., Шаньгин В. Ф. Организация и обработка структур данных в вычислительных системах: Учеб. пособ. для вузов. – М.: Высш. шк., 1987. – 248 с.
4. Варжапетян А. Г. Имитационное моделирование на GPSS/H: Учебное пособие; ГУАП. — СПб., 2007. — 384 с.: ил.
5. <http://www.mathworks.com/products/stateflow>.
6. Гультяев А. Визуальное моделирование в среде MATLAB: Учеб. пособ. – СПб: Питер, 2000. – 432 с.