УДК: 621.623

ИМИТАЦИОННАЯ МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ПОДРАЗДЕЛЕНИЯМИ, РЕАЛИЗУЮЩАЯ СПОСОБЫ КООРДИНАЦИОННОГО УПРАВЛЕНИЯ

А.П. Богомолов, Д.В. Чернышов (Воронеж)

Традиционно многоуровневую систему управления (связи) иерархического типа представляют в виде набора типовых элементов органов (пунктов) управления, оснащенных типовыми средствами обработки и передачи информации и обладающих потенциальными техническими возможностями по пропускной способности в соответствии с их уровнем в структуре иерархии системы. При этом в строго иерархической системе, реализующей только способ централизованного управления, на всех уровнях иерархии не должны присутствовать линии связи между элементами одного уровня иерархии [1].

Однако жесткая централизация может неоправданно увеличить длительность циклов управления вследствии необходимости обращения нижестоящих по уровню элементов для согласования своих действий с соответствующим вышестоящим элементом, что одновременно приводит к существенной его перегрузке.

В [2] рассмотрена разновидность управления в иерархических многоуровневых системах, между компонентами которых есть конфликт, но нет антагонизма – координационное управление. Оно отличается от обычного управления следующими особенностями:

а) проблема координации возникает тогда, когда система управления состоит из нескольких органов управления (лиц, принимающих решения – регуляторов) различных звеньев иерархической системы, каждый из которых имеет дело с некоторой частью общего управляемого процесса или когда нижестоящие элементы (регуляторы) обладают определенной самостоятельностью при выборе управленческих решений;

б) при координации всегда существует вышестоящий решающий элемент, который имеет право вмешиваться в деятельность нижестоящих решающих элементов, не подменяя их и не возлагая на себя выполнение свойственных им управленческих функций.

Исключение свободы выбора управленческих решений на нижнем уровне снижает качество управления, поскольку сопровождается снятием ответственности с подчиненных при выполнении ими своих функциональных обязанностей. Вместе с тем, свобода выбора управленческих решений может привести к формированию у нижестоящих органов управления целей, иногда расходящихся с целью всей системы.

В соответствии с [2] применительно к управлению частями (подразделениями) группировки на театре войны рассмотрим следующие основные варианты реализации принципов координационного управления:

– «прямое управление»: все части (подразделения) на театре управляются с единого пункта управления (ПУ) (рисунок 1а);

– «создание коалиции»: все части (подразделения) одного операционного направления (вида вооруженных сил) выполняют поставленные с главного ПУ задачи в «традиционной» зоне ответственности этих подразделений, сообразуясь с текущими изменениями обстановки без вмешательства вышестоящего органа управления, при этом предоставляя часть помехового ресурса в общую «разведывательно-помеховую сеть», а остальные части (подразделения) на театре управляются непосредственно с главного ПУ (рисунок 1б);

– «наделение ответственностью»: часть сил выполняют задачи, поставленные органом управления более низкого звена управления в «традиционной» зоне ответственности этих подразделений, а некоторые подразделения управляются непосредственно с главного ПУ (рисунок 1в);

– «развязывание противоречий»: все части (подразделения) одного операционного направления (вида вооруженных сил) выполняют поставленные с главного ПУ задачи в «традиционной» зоне ответственности этих подразделений при определяющей роли вышестоящего органа управления в распределении задач и контроле эффективности их выполнения (рисунок 1г).

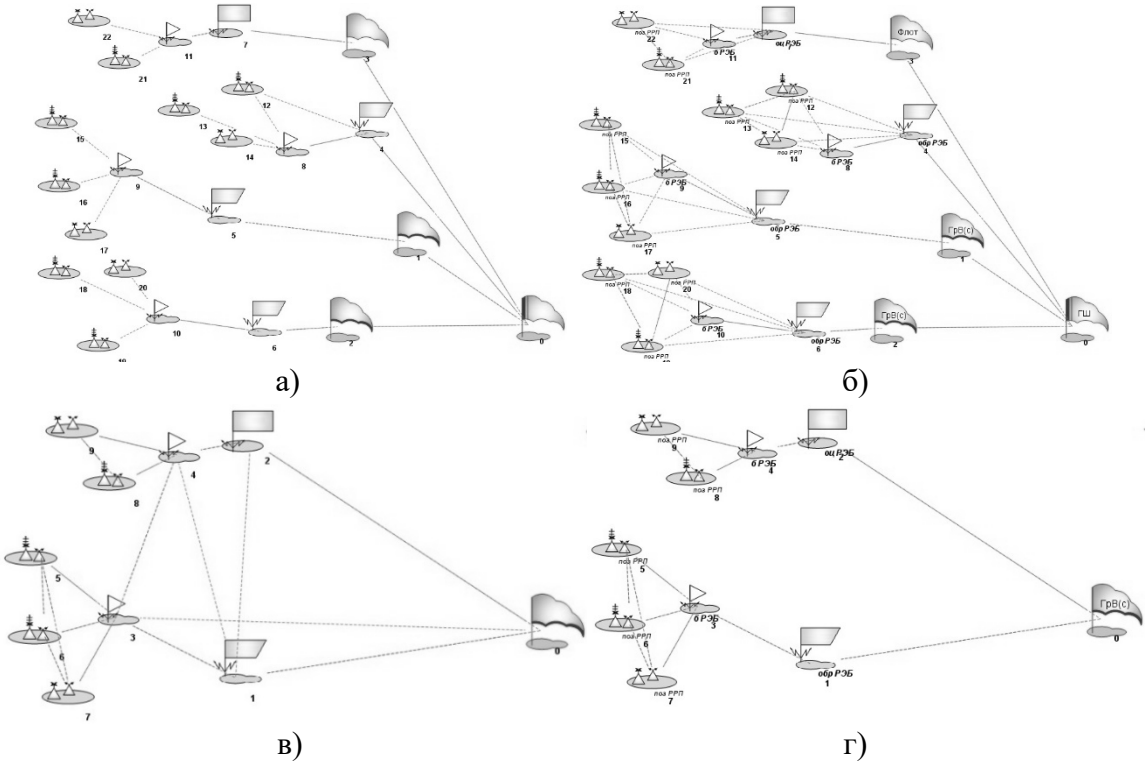


Рис. 1 – Варианты координационного управления

Анализ отмеченных способов управления показал, что при их реализации в системе иерархической системе управления предполагается наличие дополнительных связей (каналов взаимодействия) на одном уровне иерархии и параллельное ветвление древовидных иерархических связей.

Аналитическое решение задачи оценки эффективности функционирования системы такой сложной топологии традиционными методами весьма затруднительно и связано со сложностью декомпозиции такой системы в целом на совокупность однородных элементов и математического описания правил обработки входящих заявок в отдельных элементах системы методами систем массового обслуживания при неоднородных потоках [3-4].

Одним из возможных путей решения данной проблемы является компьютерное моделирование случайных процессов обмена потоками информации методами имитационного моделирования.

В данной статье авторами предложен подход к оценке эффективности предложенных способов управления, который реализован в модели [5] с одновременным применением двух методов имитационного моделирования: агентного подхода при представлении элементов структуры системы и дискретно-событийного – для

моделирования каналов (подсистем) обработки поступающей информации, ее обработки (принятия решений) и формирования команд управления (докладов) подчиненным (взаимодействующим) элементам системы в виде систем массового обслуживания с использованием элементов стандартной библиотеки среды моделирования AnyLogic.

На рисунке 2 приведен внешний вид системы управления сложной структуры, которая визуализируется в основном окне моделирования после загрузки ее топологии из файла формата Excel, где указаны количество и типы элементов, их координаты размещения, каналы приема и передачи команд управления (подчиненность) и каналы взаимодействия и их типовые параметры пропускной способности.

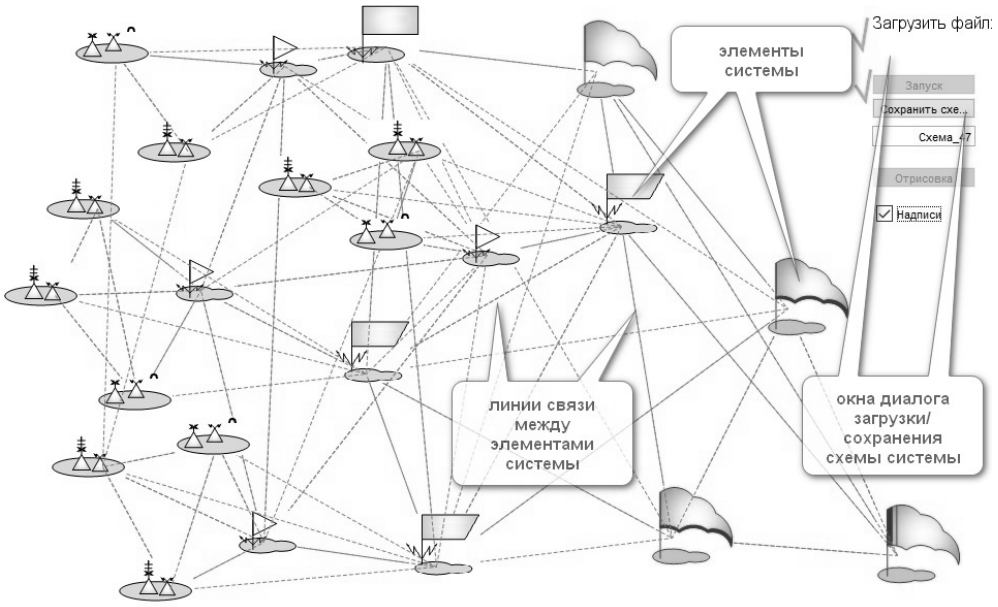


Рис. 2 – Внешний вид основного окна моделирования

На рисунке 3 приведены фрагменты Excel таблиц для хранения исходных данных модели о структуре системы управления (рисунок 3а) и порядка подчиненности и взаимодействия (рисунок 3б).

При этом порядок их взаимного размещения на экране может быть скорректирован путем перемещения помеченных мышью элементов по экрану по технологии Drag-and-Drop с использованием функции MouseDragged() основного класса Panel (отображения эксперимента и презентации агента) с последующим сохранением в файле формата Excel.

Алгоритм функционирования (приема и передачи команд управления) каждого элемента системы моделируется в виде агента, реализованного как модель совокупности каналов систем массового обслуживания с применением метода дискретно-событийного моделирования (рисунок 4). При этом в зависимости от уровня подчиненности по иерархии и типового предназначения (источник или обработчик информации) функционирование элементов каждого агента системы (отдельные каналы системы массового обслуживания) будет выполняться в соответствии с загруженными из файла базы данных параметрами (рисунок 3а).

id	name	master	type_pict	lat	lont	isx_com	isx_dokl
1	ГШ	0	0	904	536	1,00	0,00
2	Гр 1	1	11	686	519	1,00	2,00
3	обр 1 гр	1	11	798	297	2,00	2,00
4	обр 2 гр	1	12	613	67	2,00	2,00
5	обр ГШ	1	21	637	217	4,00	4,00
6	обр 1 гр	2	21	390	360	4,00	4,00
7	обр 2 гр	3	21	458	547	4,00	4,00
8	оц фл	4	31	410	43	4,00	4,00
9	б ГШ	5	31	507	245	6,00	6,00
10	б 1гр	6	31	230	279	6,00	6,00
11	б 2 гр	7	31	291	523	6,00	6,00
12	б фл	8	31	311	62	6,00	6,00
13	р РРП б ГШ 1	9	41	441	144	0,00	24,00
14	р РРП б ГШ 2	9	41	335	179	0,00	24,00

уровень иерархии элемента

тип знака элемента

наименование

координаты элемента в окне модели

интенсивности исходящих потоков команд/докладов

порядковый номер элемента

а) таблица состава и взаимного размещения элементов системы управления

	ГШ	ГрВ(с) 1	ГрВ(с) 2	Флот	обр ГШ	обр 1 гр	обр 2 гр	оц фл	б ГШ	б 1гр	б 2 гр	б фл	р РРП б -1
id	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	1	1	1	1	2	2	2	5	6	7	8	5
2	3	3	2	2	6	3	3	3	6	9	9	9	10
3	4	4	4	4	7	4	4	4	7	11	10	10	11
4	5	6	7	8	8	5	5	5	8	12	12	11	12
5	0	0	0	0	9	7	6	6	10	16	19	22	14
6					0	8	8	7	11	17	20	23	15
7						10	11	12	12	18	21	0	0
8						0	0	0	13	19	0		
9									14				
10									15				
11									0				

порядковый номер элемента

порядковые номера старших элементов

наименование элемента

признак окончания списка элементов

порядковые номера младших элементов

б) таблица порядка управления и взаимодействия

Рис. 3 – Excel таблицы для хранения исходных данных модели

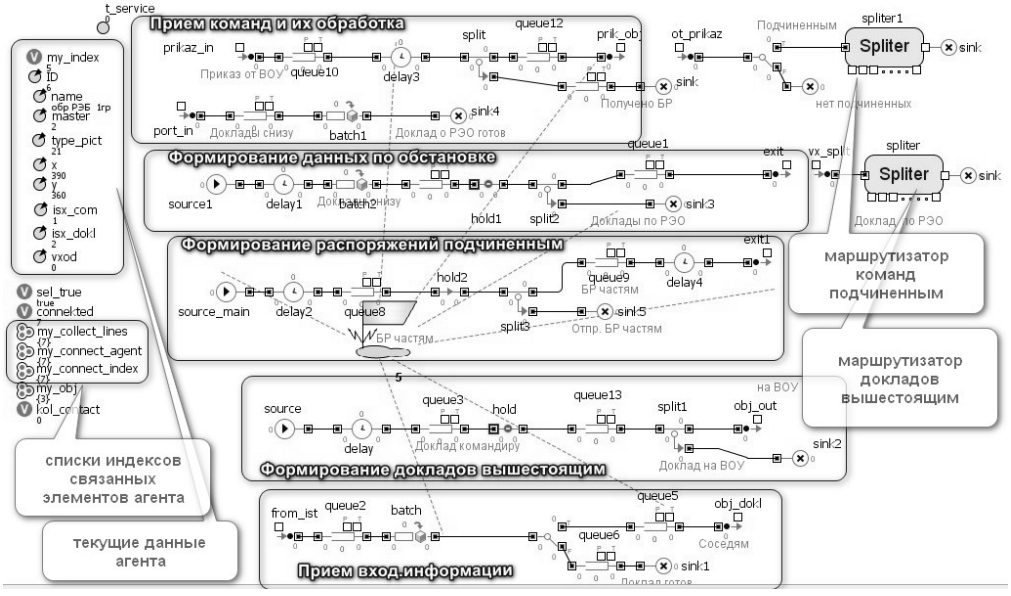


Рис. 4 – Модель агента типа «обработчик заявок»

Агент типа «обработчик заявок» представляет собой совокупность взаимосвязанных функциональных блоков:

– приема команд от вышестоящих органов (элементов) управления (входной порт типа Enter, накопитель типа Queue), их обработки (блок задержки типа Delay, накопитель

типа Queue) и принятия решения (переключатель типа SelectOutput, блок уничтожения заявок типа Sink);

– формирования данных об обстановке (источник заявок типа Source, блок задержки типа Delay, накопитель типа Queue), блок управления прохождением потока типа Hold, который управляется командой из предыдущего блока (приема команд от вышестоящих органов) при поступлении соответствующей команды (требование доклада) и принятия решения (переключатель типа SelectOutput, блок уничтожения заявок типа Sink) о типе получателя (вышестоящий или взаимодействующий элемент);

– формирования (ретрансляции) команд подчиненным (источник заявок типа Source, блок задержки типа Delay, накопитель типа Queue), блок управления прохождением потока типа Hold, который управляется командой из предыдущего блока (приема команд от вышестоящих органов) при поступлении соответствующей команды (получение приказа);

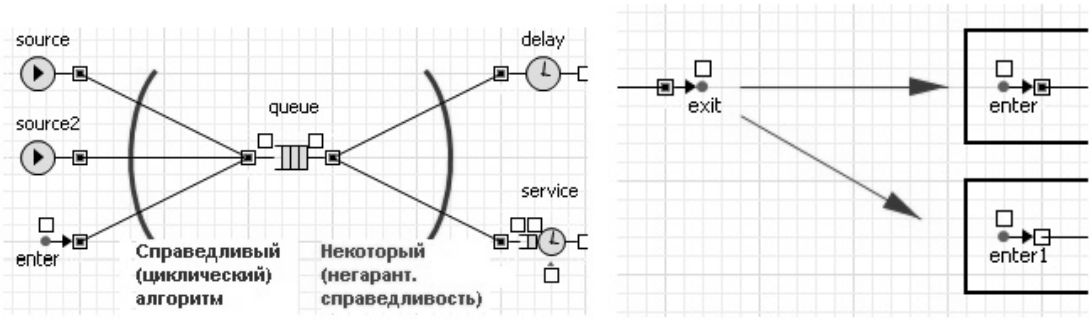
– формирования докладов вышестоящим (источник заявок типа Source, блок задержки типа Delay, накопитель типа Queue), блок управления прохождением потока типа Hold, который управляется командой из предыдущего блока (приема команд от вышестоящих органов) при поступлении соответствующей команды (требование доклада) и принятия решения (блок уничтожения заявок типа Sink);

– приема информации обстановки (входной порт типа Enter, накопитель типа Queue), обработки (формирования пакетов типа Batch) и принятия решения (переключатель типа SelectOutput, блок уничтожения заявок типа Sink).

Как видно из структуры агента «обработчик заявок» (рисунок 4) в состав описанных блоков включены выходные порты типа Exit для пересылки потоков заявок соответствующим элементам системы в соответствии с правилами маршрутизации (рисунок 3б) – все номера в столбце данного агента, которые меньше номера данного агента являются вышестоящими, а которые больше – подчиненные. Списки маршрутизации формируются в каждом агенте при инициализации структуры с помощью функции set_lines() в виде коллекций типа my_connect_agent (my_collect_lines).

В ходе реализации модели потребовалось решить проблему, обусловленную переменным числом управляемых (подчиненных, взаимодействующих) элементов системы при маршрутизации команд управления.

Суть проблемы заключается в том, что, если несколько выходных портов соединены с одним входным портом (рисунок 5а) и сразу несколько объектов хотят передать заявку, то выбор будет произведен «справедливым» образом согласно реализованному во входном порте циклическому алгоритму (round-robin) [6].

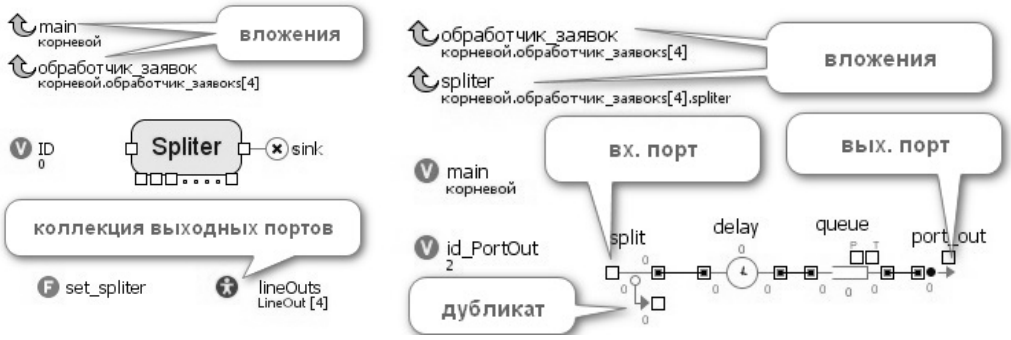


а) б)

Рис. 5 – Правила сложной маршрутизации заявок

Если один выходной порт соединен с несколькими входными портами, и сразу несколько портов готовы принять заявку, то тот порт, куда она будет переслана, будет зависеть от того, как исполняющий модуль среды AnyLogic обработает одновременные события. Разработчики среды моделирования рекомендуют исключать такие соединения и пользоваться объектом SelectOutput для соединения до 5-ти портов или механизмом сложной маршрутизации (рисунок 5б), чтобы иметь полный контроль над распределением заявок. В случае большого числа приемных портов использование нескольких объектов SelectOutput перестает быть элегантным, а в случае реплицированного принимающего объекта - так и вовсе неприменимо.

Для реализации маршрутизации заявок и устранения неоднозначности при одновременной рассылке команд управления в модели реализован агент типа Splitter (рисунок 6а), который представляет собой виртуальный маршрутизатор, состоящий из коллекции агентов типа line_Out_1s каналов отправки команд через порт типа Exit (port_out на рисунке 6б) в результате выполнения функции set_splitter(). Листинг кода программы функции set_splitter() приведен на рисунке 7.



а) б)

Рис. 6 – Модель агента Splitter (а) и вложенного агента элемента маршрутизатора (б)

В результате вложения в агенты типа «Обработчик заявок» двух нестандартных агентов «Splitter» (рисунок 4), реализующих функции виртуальных маршрутизаторов (рисунок 6а) команд управления для подчиненных и взаимодействующих элементов системы управления, каждый из которых включает коллекцию агентов виртуальных каналов отправки команд line_Out_1, устранена проблема неоднозначности при одновременной пересылки заявок группам элементов сложной системы.

```
// Создаем коллекцию агентов каналов Spliter1
for (int ii = 0; ii <= (get_Main().обработчик_заявок.get( (обработчик_заявок.ID-1) ).my_obj.size() - 1 ) ; ii++ )
{       add_lineOut_1s();    }
if ( lineOut_1s.size() > 0 )
{
for (int ii = 0; ii <= (get_Main().обработчик_заявок.get( (обработчик_заявок.ID-1) ).spliter1.lineOut_1s.size() - 1 );
       {   if ( ii == 0 )
        {  // первый элемент
  //  Selector.port_in коннектим с Selector.Split.in
         port_in.map(lineOut_1s.get(ii).split.in);    //      port_in.refreshConnections();
         if ( ii == (get_Main().обработчик_заявок.get( (обработчик_заявок.ID-1) ).spliter1.lineOut_1s.size() - 1))
       { //  Если один элемент lineOut_1s
           lineOut_1s.get(ii).split.outCopy.connect(sink.in);      }       }
        else  // остальные элементы
        {    // Selector.Split.outCopy предыдущего агента с Selector.Split.in текущего
          lineOut_1s.get(ii-1).split.outCopy.connect(lineOut_1s.get(ii).split.in);       }
        if ( ii == (get_Main().обработчик_заявок.get( (обработчик_заявок.ID-1) ).spliter1.lineOut_1s.size() - 1) )
        {  // последний элемент
 //  Split.in текущего коннектим с Split.outCopy предыдущего
        if ( ii != 0 )
             {    lineOut_1s.get(ii-1).split.outCopy.connect(lineOut_1s.get(ii).split.in);   }
 //  Selector.Split.outCopy коннектим с sink.in
         lineOut_1s.get(ii).split.outCopy.connect(sink.in);     }  }
```

Рис. 7 – Алгоритм построения виртуального маршрутизатора Splitter

Таким образом, при декомпозиции сложного объекта и реализации отдельных функций продемонстрировано на практике одно из важнейших преимуществ применения агентного подхода – возможность многократного вложения агентов друг в друга.

На рисунке 8 приведены окна вывода результатов моделирования отдельных функциональных блоков агента, в соответствующих закладках модели, что позволяет по итогам моделирования различных вариантов структуры системы управления, реализующих определенные способы координационного управления, произвести на основе полученных количественных оценок выбор наиболее рационального способа управления.

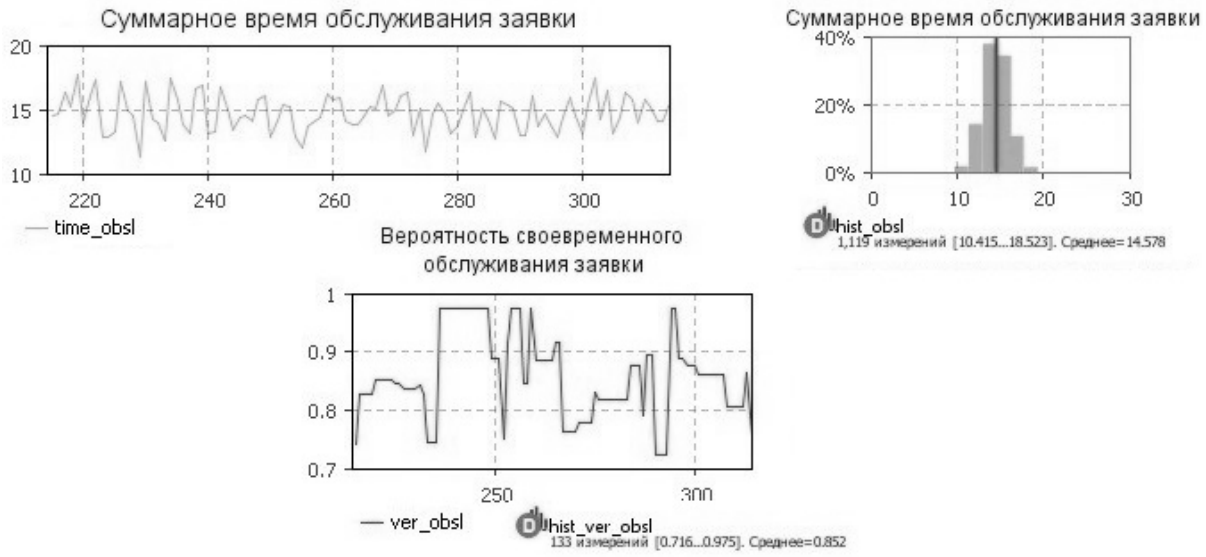


Рис. 8 – Внешний вид окна вывода результатов моделирования отдельных функциональных блоков агента

Таким образом, на примере имитационной модели функционирования системы управления, реализующей способы координационного управления, продемонстрирована возможность декомпозиции сложной организационно-технической системы военного назначения и реализации отдельных функций при ее функционировании за счет одновременного применения двух методов имитационного моделирования:

− агентного подхода при представлении элементов структуры системы многократное вложение агентов при моделировании виртуальных маршрутизаторов с переменной пропускной способностью для устранения неоднозначности при одновременной рассылке заявок (сообщений);

− дискретно-событийного моделирования с использованием элементов стандартной библиотеки среды моделирования AnyLogic для представления функциональных подсистем элементов системы управления в виде совокупности взаимосвязанных систем массового обслуживания каналов (подсистем) обработки поступающей информации, ее обработки (принятия решений) и формирования команд управления (докладов) подчиненным (взаимодействующим) элементам системы.

Литература

1. **Месарович М.** «Теория многоуровневых систем» // Месарович М., Мако Д., Такохара Д. /– М.: Мир, 1973, – 332с.
2. **Новосельцев В.И.** «Системный анализ: современные концепции» (изд.второе) / Новосельцев В.И. // Воронеж: «Кварта». – 2003. 360 с.
3. **Понтрягин Л.С.** «Математическая теория оптимальных процессов» / Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф // Физматгиз. 1961 г.
4. **Волгин Н.С.** «Исследование операций» ч.2. Санкт-Петербург: ВМА им. Н.Г. Кузнецова, 1999. – 334 с.ил.
5. «Имитационная модель функционирования системы управления подразделениями, реализующая способы координационного управления» (Модуль «Потоки и документооборот»), / Богомолов А.П., Бородавкин А.Н., Дубровин Ю.А., Кувалдин Г.В., Чернышов Д.В. // Свидетельство о государственной регистрации программы для ЭВМ №2024615425 от 06.03.24 г.
6. **Карпов Ю.** Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 [Текст] / Ю. Карпов. – СПб.: БХВ-Петербург, 2005. – 400 с.