

Д. В. Пащенко, Д. А. Трокоз

## ПРОБЛЕМЫ ПОСТРОЕНИЯ МНОГОПОТОЧНОЙ МОДЕЛИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭКСПЕРТНОЙ СИСТЕМЫ АВИАЦИОННЫХ РАДИОЛОКАЦИОННЫХ КОМПЛЕКСОВ

*Аннотация.* Рассматриваются проблемы построения многопоточной модели программного обеспечения экспертной системы авиационного комплекса радиолокационного дозора и наведения, решение которых позволит приступить к непосредственной разработке модели его экспертной системы.

*Ключевые слова:* радиолокационный комплекс, модель, сеть Петри, экспертные системы, теория автоматов.

*Abstract.* The article deals with the problem of multi-threaded software model design for the expert system of aviation radar patrol and tracking. The solution of the problem will immediately encourage the development of an expert system.

*Key words:* radar system, model, Petri nets, expert systems, theory of automata.

### Введение

Авиационные комплексы радиолокационного дозора и наведения (АК РЛДН) предназначены для обнаружения и сопровождения воздушных, наземных и надводных целей, а также управления самолетами истребительной и ударной авиации при их наведении на эти цели. Для оценки качества выполнения полетных заданий и технического состояния радиотехнических комплексов (РТК) используют экспертные системы АК РЛДН.

В настоящее время ведется разработка АК РЛДН нового поколения, для которого требуется разработать экспертную систему для проведения объективного контроля действий операторов и состояния РТК. Основной проблемой реализации такой системы является жесткое требование к времени выполнения необходимой обработки зарегистрированной информации, так как общий объем регистрируемых данных может достигать нескольких терабайт.

### 1. Описание системы

Разрабатываемая экспертная система представляет собой аппаратно-программный комплекс, который способен обрабатывать постполетную информацию, зарегистрированную АК РЛДН, в различной форме. Обработка информации состоит из нескольких этапов [1].

На первом этапе зарегистрированная информация с бортового устройства регистрации (БУР) на съемном твердотельном накопителе переносится на наземный комплекс обработки и дешифрирования информации (НКОД), который и представляет собой экспертную систему АК РЛДН (рис. 1). На этом этапе происходит разбор информации по типам кодограмм (датаграмм специального вида) и загрузка всей необходимой информации с БУРа в базу данных (БД) НКОД.

На втором этапе производится табличная обработка информации (рис. 2), которая позволяет производить селекцию и сортировку данных по заданным параметрам. Кроме того, имеется возможность совместной обработки кодограмм различных типов и объединения их по общим полям.

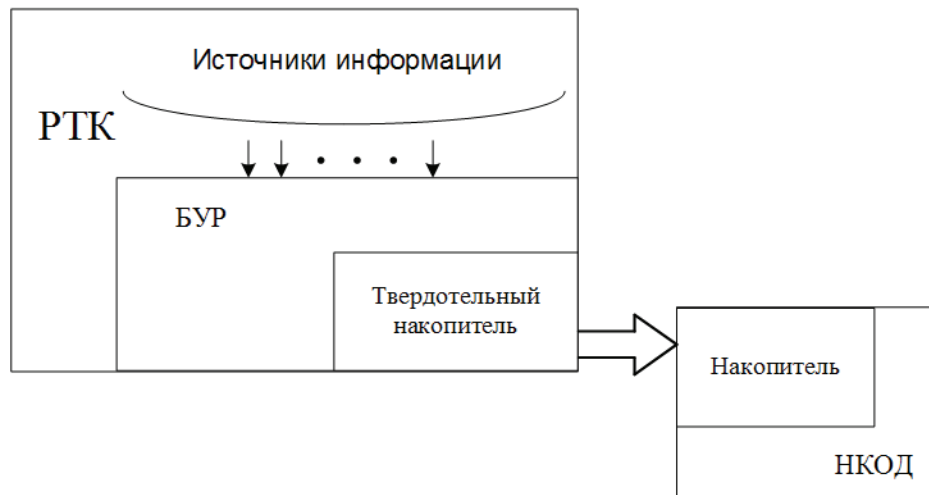


Рис. 1. Перенос информации с БУР на НКОД

Параметры селекции данных

| Параметр | Минимальное значение | Максимальное значение |
|----------|----------------------|-----------------------|
| fx       | 10000.0              | 100000.0              |
| fy       | 20000.0              | 40000.0               |

Добавить параметр  
Изменить параметр  
Удалить параметр  
Селекция

| header_t                | t         | fD         | fFi   | fVr       | fEps   | ↑ fx      | fy        | fN  |
|-------------------------|-----------|------------|-------|-----------|--------|-----------|-----------|-----|
| 23.10.2009 14:01:12.375 | 58320,870 | 165600,000 | 1,446 | -168,915  | -0,067 | 10098,576 | 32367,832 | 2   |
| 23.10.2009 14:21:38.386 | 59546,785 | 153300,000 | 0,850 | -180,764  | -0,046 | 10130,555 | 21126,800 | 6   |
| 23.10.2009 13:51:09.164 | 57717,590 | 119850,000 | 0,766 | 115,490   | -0,060 | 11263,525 | 37441,700 | 3   |
| 23.10.2009 13:05:55.738 | 55004,344 | 123000,000 | 0,842 | -127,440  | -0,026 | 11586,970 | 32217,719 | 7   |
| 23.10.2009 13:05:55.738 | 55004,297 | 118950,010 | 0,812 | -127,060  | -0,145 | 11668,777 | 26824,600 | -7  |
| 23.10.2009 13:52:16.164 | 57784,684 | 113400,000 | 0,789 | 121,415   | -0,048 | 12211,100 | 25072,781 | 4   |
| 23.10.2009 14:15:57.585 | 59205,977 | 124050,000 | 0,828 | 145,750   | -0,055 | 12357,263 | 30478,463 | 2   |
| 23.10.2009 14:22:16.183 | 59584,440 | 157800,000 | 0,899 | -180,765  | -0,015 | 12528,481 | 27770,633 | 4   |
| 23.10.2009 14:21:57.183 | 59565,630 | 155850,000 | 0,861 | -181,240  | -0,129 | 13222,668 | 23044,200 | -12 |
| 23.10.2009 14:01:40.777 | 58349,170 | 171750,000 | 1,458 | -175,220  | -0,047 | 13814,350 | 36672,570 | -10 |
| 23.10.2009 14:23:03.183 | 59631,773 | 161100,000 | 0,957 | -181,240  | -0,031 | 14244,690 | 37083,940 | 1   |
| 23.10.2009 13:06:05.535 | 55013,934 | 125550,010 | 0,838 | -128,326  | -0,144 | 14922,350 | 32478,035 | -7  |
| 23.10.2009 13:06:14.937 | 55023,540 | 126450,016 | 0,846 | -127,060  | -0,047 | 15862,573 | 32400,453 | 4   |
| 23.10.2009 13:52:06.562 | 57775,110 | 123375,000 | 0,793 | 139,920   | -0,045 | 17644,818 | 33912,664 | -6  |
| 23.10.2009 13:48:16.164 | 57544,640 | 125100,000 | 0,464 | -59,350   | -0,081 | 17746,734 | 35778,938 | -1  |
| 23.10.2009 13:51:37.562 | 57736,793 | 126149,990 | 0,766 | 116,600   | -0,045 | 17797,066 | 39098,410 | 4   |
| 23.10.2009 13:52:25.761 | 57794,260 | 121425,000 | 0,790 | 127,800   | -0,083 | 18796,190 | 29579,861 | 2   |
| 23.10.2009 13:06:24.738 | 55033,098 | 127950,016 | 0,835 | -128,240  | -0,080 | 19087,229 | 31278,863 | -7  |
| 23.10.2009 14:14:03.382 | 58991,740 | 129149,990 | 0,971 | -1434,327 | -0,074 | 19406,736 | 39699,883 | 7   |

Рис. 2. Табличная обработка информации

Третий этап обработки заключается в представлении результатов селекции табличных данных в графическом виде (рис. 3). Каждый из объектов в графическом виде представляет собой трассу (множество точек траектории движения объекта). Для любой (дискретной) точки трассы в любой (дискретный) момент времени возможно получение информации о положении, скорости, азимуте и т.п.

Второй и третий этапы требуют значительного участия человека для анализа полетной информации. Для проведения контроля в короткие сроки в НКОД предусмотрена функция автоматизированного контроля аппаратуры и выполнения полетного задания. Этот контроль производится согласно заранее заданному сценарию и не требует от пользователя каких-либо действий.

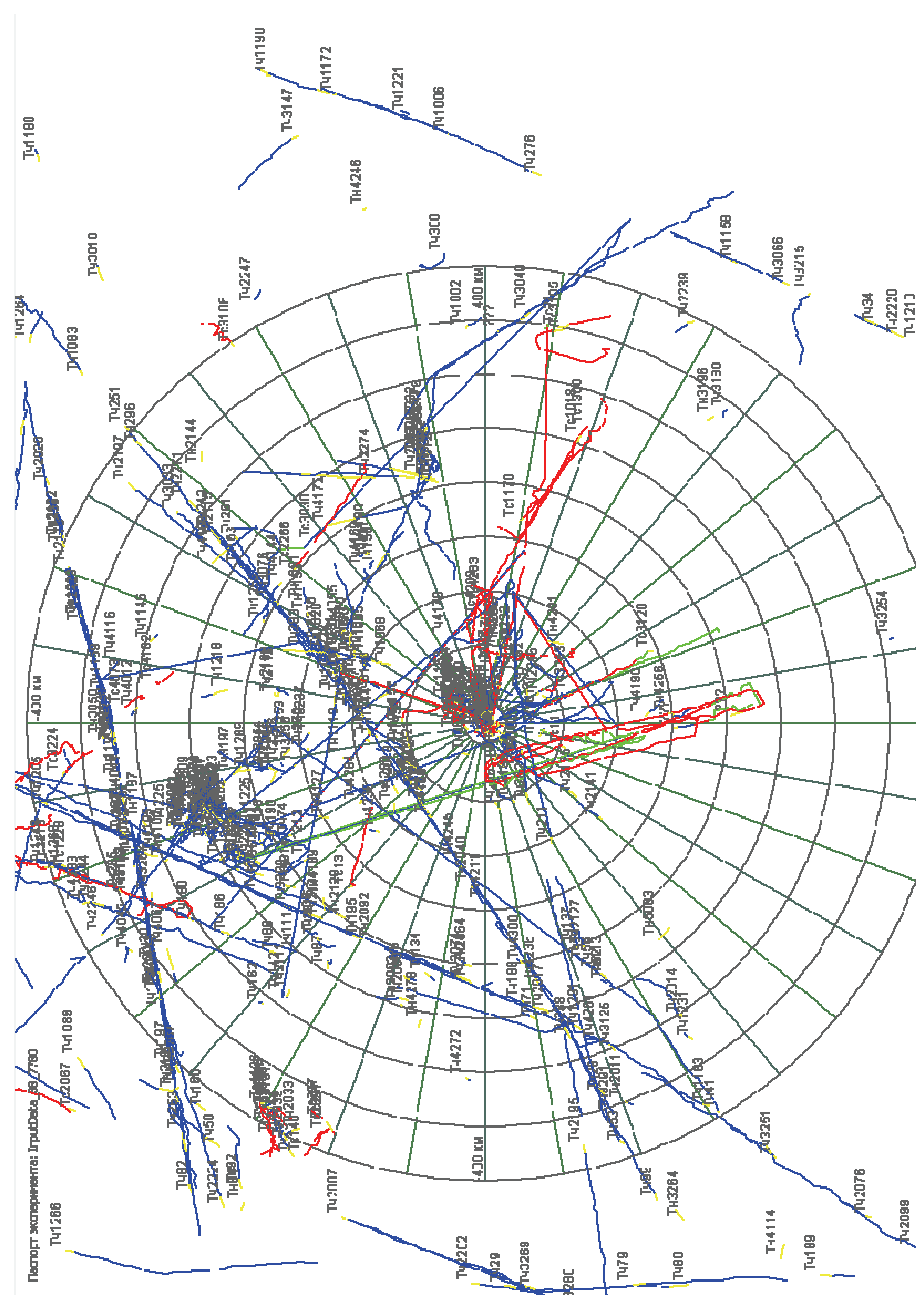


Рис. 3. Графическое представление информации

## **2. Анализ особенностей формализации системы**

Таким образом, для получения основного объема результатов обработки необходимо произвести две операции:

- разбор и загрузку данных с БУРа в БД НКОД;
- автоматизированный контроль аппаратуры и выполнения полетного задания.

Проблема, которая возникает при реализации такой экспертной системы, связана с большим объемом обрабатываемых данных [2]. Каждая из двух операций, описанных выше, обрабатывает практически все данные с БУРа объемом несколько терабайт [3]. В программное обеспечение (ПО) разрабатываемой экспертной системы включены алгоритмы параллельной обработки информации, позволяющие увеличить производительность комплекса. Несмотря на то, что число одновременно выполняющихся потоков в системе (ОС МСВС 3.0) практически не ограничено, одновременный запуск большого числа потоков приводит к большим вычислительным затратам ОС на переключение между ними, что вызывает снижение производительности системы в целом. Чтобы не допустить этого, в ПО экспертной системы встроен специальный механизм управления потоками (диспетчер потоков). Каждый новый поток, который требуется запустить на исполнение, передается диспетчеру потоков, а он, в свою очередь, уже обеспечивает, чтобы число одновременно исполняемых потоков было равно числу процессоров. В соответствии вышеописанным алгоритмом может быть сформулирована лемма об управлении потоками в многопоточной системе.

***Лемма.** Максимальное быстродействие системы достигается, когда число одновременно выполняющихся потоков равно числу процессоров в системе.*

Следует отметить, что такой способ организации управления потоками возможен, если потоки выполняют продолжительные действия, как, например, обработку больших объемов данных. Управление подобным образом потоками прорисовки окон или обработки нажатия клавиш может привести к нестабильной работе системы.

## **3. Выбор аппарата для моделирования**

Описанный выше алгоритм управления потоками был реализован в ПО экспертной системы АК РЛДН. Новой проблемой явилась необходимость выбора аппаратной части комплекса под разработанное алгоритмическое решение для обеспечения максимального быстродействия.

На работу параллельного алгоритма влияет множество аппаратно-программных факторов, совместное влияние которых невозможно проанализировать эмпирически. Наиболее эффективным методом решения проблемы является моделирование ПО экспертной системы АК РЛДН. При этом возникает проблема выбора математической модели, которая оказывает существенное влияние на адекватность – соответствие модели реальной системе [4]. Рассмотрим наиболее распространенные средства для имитационного моделирования систем параллельной обработки:

- язык GPSS;
- конечные автоматы;
- сети Петри.

Язык GPSS позволяет строить модели систем параллельной обработки. Так как язык предназначен для моделирования систем массового обслуживания, на нем легко изменять количественные параметры аппаратной части системы, например число процессоров в системе или жестких дисков в рейд массиве и их быстродействие, но не структурную организацию. В программе на языке GPSS достаточно сложно представить непосредственно процессы обработки данных на уровне алгоритмов. Кроме того, модель представляет собой программу, а значит, не имеет графической интерпретации, что затрудняет процесс разработки модели и снижает наглядность модели в целом.

Конечные автоматы имеют как формальное, так и графическое представление, что является несомненным плюсом. Они позволяют построить модели систем параллельной обработки, однако, чтобы изменить число параллельных процессов в такой модели, требуется внести существенные изменения в саму модель. Кроме того, попытка разработки сложной модели на конечном автомате приведет к быстрому росту числа состояний автомата, что в итоге сделает разработку такой модели крайне сложным занятием.

Сети Петри, так же как и предыдущие инструменты, позволяют построить модель системы многопоточной обработки данных. Сеть Петри обычно представляется в графической форме, но также имеет и формальный вариант представления. Такой класс сетей, как раскрашенные временные сети Петри, позволяет строить параллельные системы любой сложности, а их мощность эквивалентна машине Тьюринга. В отличие от конечных автоматов, в сетях Петри потоки представляются фишками, а не ветвями сети, что позволяет управлять числом потоков в модели путем изменения начальной маркировки в позициях сети.

Сравнивая перечисленные выше варианты моделей системы, можно сделать вывод, что для построения модели ПО экспертной системы АК РЛДН наиболее удобным инструментом являются сети Петри. В отличие от языка GPSS, они имеют удобную графическую форму представления, что ускоряет процесс разработки модели, а также увеличивает наглядность модели в целом. Если же сравнить сети Петри с конечными автоматами, то первые обладают большей гибкостью в представлении потоков в модели, а также изменении их числа, что необходимо в процессе моделирования.

Выбрав математическую модель, мы сталкиваемся с проблемой выбора системы моделирования, которая позволила бы не только представить сеть Петри необходимого вида (временные раскрашенные сети Петри), но и обладала возможностью проведения имитационно-математического моделирования. Из множества систем моделирования на основе сетей Петри, имеющих свободную лицензию, наиболее мощным и распространенным продуктом является CPN Tools. Моделирующая система CPN Tools разработана в университете Орхуса (Дания). Язык описания моделей в CPN Tools представляет собой сочетание графа сети Петри и языка программирования CPN ML (для описания условий, функций и типов данных).

Кроме того, CPN Tools обладает богатыми возможностями для проведения имитационно-математического моделирования, позволяющими, используя визуальные элементы управления, оперативно изменять структуру и свойства моделируемых объектов.

После выбора системы моделирования необходимо рассмотреть различные способы реализации модели системы на сетях Петри. При разработке модели сложной системы возникает проблема представления компонентов системы: как представить разнородные части реальной системы в модели, чтобы получить, с одной стороны, адекватную модель многопоточной системы, а с другой – приемлемое время моделирования. Кроме того, нужно обеспечить гибкость модели, одновременно не перегружая ее лишними вспомогательными позициями и переходами, которые не обладают информативной ценностью. Чтобы совместить все эти противоречивые требования, нужно применить поэтапное проектирование модели.

Чтобы стало понятно, как должна функционировать многопоточная модель системы, рассмотрим построение модели однопоточной системы на сетях Петри. В этом случае позиции будут выступать в роли элементов хранения данных, представленных фишками, а переходы – в роли процессов или действий, осуществляемых над данными. При этом каждая фишка будет относиться к определенному множеству цветов – типу данных.

Обработка данных в модели происходит при переходе, т.е. перемещении фишки из одной позиции в другую. Она заключается в изменении цветов входных фишек (поступающих на вход перехода), а также непускании на выход некоторых входных фишек и создании новых выходных фишек. Это позволяет моделировать процесс преобразования данных для любой однопоточной системы.

При создании многопоточной системы необходимо ввести специальный тип фишек, который представляет потоки в модели. Это фишки множества цветов целых чисел, обладающих свойством времени:

$$colset\ Thread = StructThread\ timed,$$

где *StructThread* – структура, содержащая номер и информационные поля потока. В многопоточной модели лишь множество цветов потоков должно обладать свойством времени. При этом каждый тип данных, который участвует в многопоточной обработке, будет преобразован в следующую структуру:

$$colset\ TYPE\_T = record\ thread:Thread*data:TYPE,$$

где *TYPE* – базовый тип данных, который будет участвовать в многопоточной обработке; *Thread* – поток, который производит обработку данных.

Используя такое представление, получаем два вида множеств цветов: первый – типы данных, второй – состоящий из одного специального множества, представляющего потоки модели. В многопоточной модели для каждой позиции, хранящей фишки для нескольких потоков, создается позиция, хранящая потоки. Переход будет разрешен, если входные фишки данных соответствуют входной фишке потока. Проследить работу модели можно, наблюдая за изменением кортежа, содержащего количество фишек потоков. При этом потоки влияют лишь на данные, которые им разрешено обрабатывать, исходя из топологии модели.

Для реализации функционирования модели необходимо для каждого перехода задать следующие условия:

$$(vThread = \#thread\ x_1),$$

$andalso (vThread = \#thread\ x_2),$

...

$andalso (vThread = \#thread\ x_n),$

где  $x_i$  –  $i$ -е имя входной переменной;  $n$  – число входных переменных.

Множество условий переходов обеспечит соответствие фишек-данных и фишек-потоков, что обеспечит независимую обработку данных, относящуюся к разным потокам.

В любой системе с многопоточностью существуют этапы обработки, которые в единый момент времени могут производиться лишь одним из потоков (например, чтение данных с диска). Для построения модели с многопоточной обработкой требуется ввод в модель критической секции – участка сети Петри, в котором одновременно может находиться лишь один поток (в общем случае ограниченное количество потоков).

Для реализации критической секции вводятся две позиции:

- *Free* (критическая секция свободна);
- *Take* (критическая секция занята).

В добавление к позициям вводится особое множество цветов *Mark*, которое содержит единственный цвет:

$colset\ Mark = with\ mark.$

Число фишек типа *Mark* в позиции *Free* в начальной расстановке фишек указывает на число критических ресурсов, т.е. число потоков, которые могут одновременно находиться в критической секции. Пока в позиции *Free* есть фишки, в критическую секцию могут поступать потоки. При поступлении очередного потока в критическую секцию одна фишка *Mark* переходит из позиции *Free* в позицию *Take*. Когда поток выходит из критической секции, т.е. освобождает критический ресурс, одна фишка *Mark* из позиции *Take* переходит в позицию *Free*.

Пример реализации многопоточной обработки данных с использованием сети Петри в визуальном редакторе CPN Tools представлен на рис. 4.

Входная информация поступает в переход *processing data* одновременно с обрабатываемым ее потоком. При этом происходит сопоставление потока и данных. Поскольку количество потоков может быть больше, чем критических ресурсов, в модели вводится критическая секция. Критическая секция реализуется с помощью переходов и позиций *Take* и *Free*, как описано выше. Подобный подход позволяет строить модели многопоточных систем, имеющих критические ресурсы.

### Заключение

Рассмотренные в данной статье решения проблемы построения многопоточной модели ПО экспертной системы АК РЛДН позволили приступить к непосредственной разработке модели ПО экспертной системы АК РЛДН, которая ведется в настоящее время.





**Список литературы**

1. **Мачалин, В. А.** Стратегия параллельной обработки массивов данных в системе объективного контроля радиотехнического комплекса радиолокационного дозора и наведения / В. А. Мачалин, А. Н. Токарев, Д. А. Трокоз, Д. В. Пашенко, М. П. Синев // Радиотехника. – 2010. – Вып. 8. – С. 51–55.
2. Приказ министра обороны РФ от 24.09.2004 № 275 об утверждении федеральных авиационных правил производства полетов государственной авиации (зарегистрировано в Минюсте РФ 10.11.2004 № 6110) // Законодательство РФ 2004 г. (часть 1). – URL: <http://2004-1.xof.ru/lib/?tm=151&vp=akt15155> (Дата обращения 23.09.2010).
3. **Пашенко, Д. В.** Объективный контроль состояния авиационных радиолокационных комплексов / Д. В. Пашенко // Проблемы автоматизации и управления : труды Международной научно-технической конференции. – Пенза, 2009. – С. 55–59.
4. **Трокоз, Д. А.** Стратегия параллельной обработки массивов данных в системе объективного контроля радиотехнического комплекса радиолокационного дозора и наведения / В. А. Мачалин, Д. В. Пашенко, Д. А. Трокоз, М. Н. Синев // Вопросы радиоэлектроники. Сер. ЭВТ. – 2009. – Вып. 4. – С. 139–144.

---

**Пашенко Дмитрий Владимирович**

кандидат технических наук, доцент,  
кафедра вычислительной техники,  
Пензенский государственный  
университет

E-mail: [Dmitry.pashchenko@gmail.com](mailto:Dmitry.pashchenko@gmail.com)

**Pashchenko Dmitry Vladimirovich**

Candidate of engineering sciences, associate  
professor, sub-department of computer  
science, Penza State University

**Трокоз Дмитрий Анатольевич**

магистрант, Пензенский  
государственный университет

E-mail: [vt@pnzgu.ru](mailto:vt@pnzgu.ru)

**Trokoz Dmitry Anatolyevich**

Graduate student,  
Penza State University

---

УДК 007.51

**Пашенко, Д. В.**

**Проблемы построения многопоточной модели программного обеспечения экспертной системы авиационных радиолокационных комплексов / Д. В. Пашенко, Д. А. Трокоз // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2011. – № 2 (18). – С. 21–29.**