

ГЛАВА 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ И МЕТОДИЧЕСКИЕ ОСНОВЫ РЕШЕНИЯ ЗАДАЧ ОЦЕНКИ НАДЕЖНОСТИ, БЕЗОПАСНОСТИ И РИСКА СИСТЕМ НА ОСНОВЕ ТЕХНОЛОГИИ АВТОМАТИЗИРОВАННОГО МОДЕЛИРОВАНИЯ, РЕАЛИЗОВАННОЙ В ПРОГРАММНЫХ КОМПЛЕКСАХ RELEX, АСМ И RISK SPECTRUM.

1.1. ОСНОВНЫЕ ПОЛОЖЕНИЯ и МЕТОДИЧЕСКИЕ ОСНОВЫ РЕШЕНИЯ ЗАДАЧ ОЦЕНКИ НАДЕЖНОСТИ, БЕЗОПАСНОСТИ И РИСКА СИСТЕМ НА ОСНОВЕ ТЕХНОЛОГИИ АВТОМАТИЗИРОВАННОГО МОДЕЛИРОВАНИЯ, РЕАЛИЗОВАННОЙ В ПРОГРАММНЫХ КОМПЛЕКСАХ КОМПАНИИ "RELEX SOFTWARE"

1.1.1. Технология расчета надежности, безопасности, эффективности, основанная на совместном использовании блок-схем надежности, деревьев отказов, Марковских графов, реализованная в программном комплексе Relex (США)

1.1.1.1. Состав модулей программного комплекса Relex

В состав программного комплекса (ПК) Relex входят 8 аналитических модулей:

- Прогнозирование Безотказности (Reliability Prediction);
- Прогнозирование Ремонтопригодности (Maintainability Prediction);
- Анализа Видов, Последствий и Критичности Отказов (FMEA/FMECA);
- Блок-Схемы Надежности (Reliability Block Diagram);
- Деревья Отказов/Событий (Fault Tree/Event Tree);
- Марковский Анализ (Markov Analysis);
- Статистический Анализ (Weibull Analysis);
- Оценка стоимости срока службы (Life Cycle Cost).

ПК Relex может формироваться из произвольного количества взаимодействующих и разделяющих единую базу данных модулей:

Модуль прогнозирования надежности содержит модели для расчета показателей надежности элементов. В модуль включена обширная база данных (~300 000 наименований), содержащая классификационные признаки элементов и характеристики надежности. Расчеты проводятся в соответствии со стандартами: MIL-HDBK-217, Telcordia (Bellcore) TR-332, Prism, NSWC-98/LE1, CNET93, HRD5, 299B.

Модуль анализа ремонтпригодности реализует положения стандарта по исследованию ремонтпригодности систем - MIL-HDBK-472. Решаются задачи прогнозирования профилактического технического обслуживания.

Модуль анализа видов, последствий и критичности отказов. Анализ проводится в соответствии со стандартами MIL-STD-1629, SAE ARP 5580 и др. Производится ранжирование опасных отказов и их оценка по приоритетам рисков.

Модуль блок-схемы надежности используется для анализа сложных резервированных систем. Содержит как аналитические методы, так и методы моделирования Монте Карло.

Модуль дерева отказов/дерева событий. Реализованы процедуры для дедуктивного и индуктивного анализа развития отказов, событий в системе. Применяется для анализа надежности и безопасности. Содержит широкий набор логико-функциональных вершин.

Модуль марковского анализа. Реализовано марковское моделирование надежности, производительности, безопасности, рисков. Начиная с версии 7.7, добавлены марковские процессы с доходами. Позволяет учитывать сложные режимы функционирования, различные виды отказов, особенности обслуживания.

Модуль “Weibull”. Предназначен для обработки статистических данных испытаний, эксплуатации. Поддерживается большой спектр распределений.

Модуль экономических расчетов (LCC). Осуществляется оценка стоимости срока службы на всех этапах создания, эксплуатации, утилизации системы. Проводятся оценки экономических показателей продления ресурса. Имеется встроенный редактор формул, позволяющий реализовать свои экономические воззрения.

Модули содержат визуальный редактор задания модели анализируемой системы непосредственно на экране компьютера. Реализованы настраиваемые отчеты, вывод на графики, язык написания макросов. Система полностью документирована, содержит развитый Help. В ПК встроен мастер импорта/экспорта исходных данных в/из текстовых файлов, электронных таблиц, баз данных, BOM файлов.

В рамках данной НИР будет проанализирована работа модулей, реализующих методы системного анализа надежности, производительности, безопасности – блок-схемы надежности, дерева отказов, марковские процессы.

1.1.1.2. Краткое описание модулей системного анализа ПК Relex

1.1.1.2.1 Модуль блок-схем надежности

Модуль блок схем надежности (RBD) предназначен для исследования надежности и производительности резервированных, восстанавливаемых систем с произвольными законами распределения случайных времен работы до отказа и восстановления элементов. Вычислительный блок модуля осуществляет расчет показателей безотказности, готовности и производительности аналитическими методами и статистическим Монте-Карло моделированием с ускорением.

При расчетах надежности и производительности в RBD возможен учет следующих факторов:

- вид резервирования (постоянное, замещением, скользящее);
- вероятность и время успешного подключения резерва;
- нагруженность резерва;
- механизм проявления отказа;
- различные стратегии восстановления;
- наличие ЗИП, профилактического обслуживания и технических осмотров;

Результатом работы RBD является вычисление следующих показателей:;

- вероятность безотказной работы;
- средняя наработка до отказа;
- интенсивность отказов системы;
- коэффициент готовности (стационарный, нестационарный);
- параметр потока отказов;
- среднее число отказов;
- средняя наработка на отказ;

- производительность и др.

Для расчета сложных, но декомпозируемых систем, в RBD реализована вложенность блоков – каждый блок может быть представлен другой RBD, причем степень вложенности ограничивается лишь объемами оперативной памяти. Взаимодействие RBD с другими модулями Relex осуществляется через связь блоков с соответствующими объектами (элемент или сборка из модуля Прогнозирования Безотказности, объект модуля АВОПК, объект дерева отказов).

Начиная с версии 7.6, в Relex RBD введена возможность решения оптимальных задач надежности: определения числа резервных элементов, максимизирующего показатели надежности/производительности или минимизирующего стоимость системы; определение оптимальных периодов профилактического обслуживания или технических осмотров.

В версию 7.7 добавлен новый вид блок-схем – фазовые диаграммы, используемые для моделирования надежности систем, чье время функционирования можно разбить на стадии (фазы), каждая из которых характеризуется своей длительностью, величинами интенсивностей отказов элементов, надежностной структурой.

Для создания блок-схем и проведения соответствующих расчетов необходимо выполнить следующие шаги:

1. Создать новый или открыть имеющийся файл проекта Relex (*.grj). В верхнем левом окне проекта перейти на вкладку RBD Table. С помощью команды системного меню Insert/New RBD создать новую диаграмму и вставить новую запись в таблицу. Ввести в строку таблицы имя и описание создаваемой диаграммы.
2. Набрать блок-схему с помощью операций вставки блока (Insert/Figure или Ctrl+G) и соединения блоков в заданную конфигурацию. Соединение осуществляется щелчком мыши на выходе нужного блока и ведением (при нажатой левой клавиши мыши) возникшей соединительной линии (коннектора в терминологии Relex RBD) до входа другого блока. При организации соединений необходимо помнить, что разветвление связей может осуществляться только через специальные переходники (Junction в терминологии Relex RBD). Вставка переходника – Insert/Junction или Ctrl+J. Для схем с простыми соединениями (например, последовательных) можно включить режим автоматического соединения – RBD/Auto Connect Figures. При работе с визуальным редактором RBD можно пользоваться многоуровневой командой Undo (Ctrl+Z или кнопка на панели инструментов), командами перемещения, выделения, удаления, копирования, вырезки, вставки компонентов диаграмм.
3. Настроить визуальные свойства вершин. Для этого при выделенной вершине нажимается правая клавиша мыши и из всплывающего меню выбирается пункт “Visual Properties”. На вкладках появившейся диалоговой панели General, Text, Graphic, Size, Color, Text Display настраиваются цвет, форма и размер вершин, содержание и форматы надписей.
4. Настроить расчетные свойства вершин. Для этого при выделенной вершине нажимается правая клавиша мыши и из всплывающего меню выбирается пункт “Calculation Properties”. На вкладках появившейся диалоговой панели настраиваются расчетные свойства:
 - General – признак повторяемости, стоимость, производительность блока
 - Failure – вид и параметры функции распределения случайного времени отказа блока, механизм проявления отказа, коэффициент нагруженности; на этой же вкладке представляется возможность сопряжения с модулем обработки статистических данных об отказах данного блока













- Redundancy – вид резервирования и количество резервных компонентов блока, вероятность и время успешного подключения резерва
 - Maintenance – три группы свойств ремонтпригодности:
 - a) Corrective – на этой вкладке задаются параметры восстановления (число ремонтных бригад, момент начала восстановления, функция распределения и параметры случайного времени восстановления, процентное соотношение восстанавливаемых и невосстанавливаемых отказов на рассматриваемом периоде времени, различного рода стоимостные и ресурсные характеристики, нужные для решения оптимизационных задач)
 - b) Preventive – на этой вкладке задаются параметры профилактического обслуживания (периодичность, функция распределения и параметры случайного времени обслуживания, стоимостные и ресурсные характеристики профилактики)
 - c) Inspection - на этой вкладке задаются параметры технических осмотров, аналогичные параметрам профилактического обслуживания, и определяется режимы проведения осмотров (оперативный или с выключением осматриваемых компонентов)
 - Sparring – три группы свойств ЗИПа, определяющих количество, места расположения (Onsite/Offsite/Repair Shop), обновления, стоимостные и ресурсные характеристики
5. Если блок диаграммы предназначен для внедрения других объектов Relex, то вместо пункта 4 необходимо задать связь этого блока с нужным объектом (элемент, сборка, запись АВОПК, дерево отказов, другая RBD). Элемент или сборку можно перетащить на лист диаграммы с помощью операций перетаскивания (drag and drop). Связь с другими объектами устанавливается выбором пунктов всплывающего меню Set Data Link или Set Diagram Link.
6. Настроить свойства переходников. Для этого при выделенном переходнике нажимается правая клавиша мыши и из всплывающего меню выбирается пункт “Properties”. На вкладках появившейся диалоговой панели настраиваются
- визуальные свойства: форма, размер, цвет, расположение входов/выходов, надпись, формат надписи
 - свойства резервирования (redundancy): вид надежностного соединения блоков, входящих в данный переходник (параллельное, последовательное, замещением с произвольной степенью нагруженности)
7. Настроить свойства коннекторов. Для этого при выделенном коннекторе нажимается правая клавиша мыши и из всплывающего меню выбирается пункт “Properties”. На вкладках появившейся диалоговой панели настраиваются визуальные свойства (толщина, стиль, цвет), задается приоритет связи.
8. Провести расчеты надежности и производительности. Для этого необходимо:
- Выбрать пункт системного меню Project/Calculate, на появившейся панели многостраничного диалога выбрать вкладку Calculation Selection (Выбор Расчета) и поставить галочку в окошке выбора, соответствующем RBD.
 - На вкладке RBD задать интервал времени и количество промежуточных точек, на которых выводятся результаты вычислений. Здесь же формируется набор вычисляемых показателей безотказности, готовности, производительности.

- На вкладке RBD Advanced задать параметры Монте-Карло моделирования (число итераций, доверительные интервалы), выбрать тип сети – потоковая или электрическая (необходимо при вычислении показателей производительности).
- Нажать кнопку ОК для запуска вычислений. Результаты расчета можно просмотреть, выбрав пункт системного меню View/RBD Calculation Results. Для просмотра путей или сечений выбираются пункты меню View/Path Sets или View/Cut Sets соответственно.

9. Создать отчеты нужного формата с помощью Мастера Отчетов (File/Print Preview).

1.1.1.2.2 Модуль деревьев отказов

Модуль деревьев отказов (FT) предназначен для исследования надежности и безопасности систем. Модуль Relex FT лишен недостатков и ограничений классических деревьев отказов за счет введения новых логико-динамических операторов (вершин), учитывающих зависимость событий, временные соотношения, приоритеты. В таблице приведен список вершин и событий, реализованных в Relex.

| вершина | название | описание |
|-------------------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------|
|  | AND | логическое И |
|  | OR | логическое ИЛИ |
|  | NAND | логическое И-НЕ |
|  | NOR | логическое ИЛИ-НЕ |
|  | NOT | логическое НЕ |
|  | VOTING (k/n) | m/n голосование (мажоритарный выбор) |
|  | INHIBIT | логическое И с запрещающим входом (ингибиторное И) |
|  | XOR | исключающее ИЛИ |
|  | PRIORITY AND | приоритетное И (динамический оператор) |
|  | FDEP | учитывает зависимости между событиями и временной порядок (динамический оператор) |
|  | SPARE | учитывает состояние резерва, в частности, нагруженности, (динамический оператор) |
|  | SEQ | учитывает последовательность возникновения событий (динамический оператор) |

В модуле деревьев отказов Relex предусмотрено также моделирование с общими элементами в различных ветвях дерева и с общими причинами отказов. Для учета отказов по общей причине используется несколько моделей (в литературе они известны под названиями: β -факторная модель, MGL-модель, α -модель, BFR модель).

Введенные новые операторы, общие элементы и общие причины позволяют учитывать в модели многие особенности, присущие возникновению опасных последствий и технические, алгоритмические, организационные меры по обеспечению безопасности.

В модуле осуществляется расчет следующих показателей:

- вероятность отказа;
- неготовность;
- параметр потока отказов;
- среднее число отказов.

Значения показателей вычисляются как для вершинного события, так и для каждого промежуточного.

Кроме того, для каждого выделенного события (как вершинного, так и промежуточного) могут просматриваться и анализироваться наборы соответствующих минимальных сечений.

Если дерево отказов очень велико, то для увеличения скорости вычислений можно осуществить приближенную оценку без существенного снижения точности. Это достигается ограничением числа учитываемых сечений, пренебрегая сечениями с малой вероятностью, использованием методов суммирования сечений, ограничения числа пересечений, Эзари-Прошана.

Возможно проведение сравнения относительной важности различных событий, используя методы Бирнбаума, критичности, Фусселя-Везели.

Деревья отказов интегрированы с остальными модулями Relex. Любому событию дерева могут быть соотнесены

- элементы или сборки модуля прогнозирования безотказности;
- виды отказов из FMEA/FMECA;
- граф переходов из Relex Марков;

Само же дерево может быть соотнесено блокам модуля Relex RBD.

Для создания дерева и проведения соответствующих расчетов необходимо выполнить следующие шаги:

1. Создать новый или открыть имеющийся файл проекта Relex (*.rpj). В верхнем левом окне проекта перейти на вкладку Fault Tree Table. С помощью команды системного меню Insert/New Top Gate создать вершинное событие. И далее, следуя логике сверху-вниз, для каждой вершины-события определять входные события и их логику, вплоть до базисных, далее не развиваемых событий. Определить входы (Gate, Basic Event) можно:

- командами системного меню Insert/Input, Insert/Gate, Insert/Event
- на вкладке Inputs многостраничной диалоговой панели свойств, вызываемой двойным щелчком вершины или нажатием правой клавиши мыши

2. Описать визуальные и расчетные свойства вершин-событий.

3. Если событие предназначено для внедрения других объектов Relex, то вместо пункта 2 необходимо задать его связь с нужным объектом (элемент, сборка, запись АВОПК, марковский граф). Связь с другими объектами устанавливается выбором пунктов всплывающего меню Set Data Link.

4. Провести расчеты надежности и безопасности. Для этого необходимо:

- Выбрать пункт системного меню Project/Calculate, на появившейся панели многостраничного диалога выбрать вкладку Calculation Selection (Выбор Расчета) и поставить галочку в окошке выбора, соответствующем Fault Tree.

- На вкладке Fault Tree задать интервал времени и количество промежуточных точек, на которых выводятся результаты вычислений, сформировать набор вычисляемых показателей и настроить параметры вычислений.
- Нажать кнопку ОК для запуска вычислений. Результаты расчета можно увидеть, выбрав пункт системного меню View/Fault Tree Calculation Results. Выбрав пункт всплывающего меню Highlight Cut Sets, можно на построенном дереве просмотреть минимальные сечения.

5. Создать отчеты нужного формата с помощью Мастера Отчетов (File/Print Preview).

1.1.1.2.3 Модуль Марковского моделирования

Марковские процессы давно и с успехом применяются в моделировании и анализе надежности систем. Разрабатываемые с помощью этого аппарата модели являются динамическими и позволяют отражать необходимые временные условия и другие особенности, зависимости, которые конкретизируют траекторию переходов системы в пространстве возможных состояний, образованных отказами, восстановлением элементов.

В Relex реализованы марковские процессы с дискретным множеством состояний и непрерывным временем, позволяющие учитывать следующие особенности функционирования и резервирования систем:

- несовместные виды отказов элементов;
- последовательность возникновения отказов;
- изменение интенсивностей отказов элементов в зависимости от уже происшедших событий (в частности, степень нагруженности резерва);
- количество бригад по восстановлению (ограниченное, неограниченное восстановление);
- очередность восстановления;
- ограничения на ЗИП;
- различную эффективность функционирования в различных состояниях системы и доходы (потери) за переходы в состояния.

Известные проблемы марковского моделирования, связанные с размерностью и жесткостью порождаемых в процессе моделирования систем дифференциальных уравнений, успешно решены в Relex Markov.

Реализованные в Relex методы позволяют решать жесткие и плохо обусловленные системы, в частности, тестирование программы на примере с числом обусловленности порядка 10^{25} дает удовлетворительные результаты. Возможность получения решения систем с таким числом обусловленности вполне достаточна на практике. Relex, практически, не содержит ограничений на число состояний исследуемой системы. Оно ограничивается лишь размером оперативной памяти.

Вычисляемые показатели:

- вероятность каждого из состояний;
- вероятность безотказной работы (отказа) на заданном интервале времени;
- вероятность застать систему в заданный момент времени в работоспособном подмножестве состояний (коэффициент готовности);
- средняя наработка на отказ;
- средняя наработка до отказа;
- среднее время восстановления;
- параметр потока отказов;
- средняя эффективность (производительность) в заданный момент времени;
- средняя интегральная эффективность на заданном интервале времени.

Модуль содержит графический редактор, с помощью которого на экране компьютера набирается граф состояний и переходов марковского процесса. Каждой вершине и дуге задаются визуальные и вычислительные свойства. Такое задание марковского процесса намного эффективнее и нагляднее обычно применяемого заполнения инфинитезимальной матрицы.

Для создания марковского графа и проведения соответствующих расчетов необходимо выполнить следующие шаги:

1. Создать новый или открыть имеющийся файл проекта Relex (*.rj). В верхнем левом окне проекта перейти на вкладку Markov Table. С помощью команды системного меню Insert/New Markov Diagram создать новый марковский граф и вставить новую запись в таблицу. Ввести в строку таблицы имя и описание создаваемого графа.
2. Набрать блок-схему с помощью операций вставки состояний (Insert/State или Ctrl+G) и соединить состояния в соответствии с логикой переходов системы. Соединение осуществляется щелчком мыши на выбранном состоянии (вершине графа) и ведением (при нажатой левой клавиши мыши) возникшей соединительной линии (дуги графа) до нужного состояния. При организации соединений необходимо помнить, что дуги входят/выходят только в специальные маленькие треугольники, обрамляющие эллипсы состояний. При работе с визуальным редактором можно пользоваться многоуровневой командой Undo (Ctrl+Z или кнопка на панели инструментов), командами перемещения, выделения, удаления, копирования, вырезки, вставки дуг и вершин графа.
3. Настроить визуальные свойства вершин. Для этого при выделенной вершине нажимается правая клавиша мыши и из всплывающего меню выбирается пункт “Visual Properties”. На вкладках появившейся диалоговой панели General, Text, Graphic, Size, Color, Text Display настраиваются цвет, форма и размер вершин, содержание и форматы надписей.
4. Настроить расчетные свойства вершин. Для этого при выделенной вершине нажимается правая клавиша мыши и из всплывающего меню выбирается пункт “Calculation Properties”. На вкладках появившейся диалоговой панели настраиваются расчетные свойства:
 - State – тип состояния (работоспособное, отказовое, деградация);
 - Initial Condition - начальное условие ($0 \div 1$);
 - Cost - доход (убыток) за единицу времени пребывания в данном состоянии.
5. Настроить визуальные свойства соединений (дуг). Для этого при выделенной дуге нажимается правая клавиша мыши и из всплывающего меню выбирается пункт “Visual Properties”. На вкладках появившейся диалоговой панели настраиваются визуальные свойства (толщина, стиль, цвет дуги, формат надписи).
6. Настроить расчетные свойства соединений (дуг). Для этого при выделенной дуге нажимается правая клавиша мыши и из всплывающего меню выбирается пункт “Calculation Properties”. На вкладках появившейся диалоговой панели настраиваются расчетные свойства:
 - Rate - интенсивность перехода;
 - Cost per Transition – доход (штраф) за переход в состояние.
7. Провести расчеты надежности и эффективности. Для этого необходимо:
 - Выбрать пункт системного меню Project/Calculate, на появившейся панели многостраничного диалога выбрать вкладку Calculation Selection (Выбор Расчета) и поставить галочку в окошке выбора, соответствующем Markov.
 - На вкладке Markov задать интервал времени и количество промежуточных точек, на которых выводятся результаты вычислений, определить требуемую точность

расчетов. Здесь же формируется набор вычисляемых показателей безотказности, готовности, эффективности.

- Нажать кнопку ОК для запуска вычислений. Результаты расчета по системе в целом можно просмотреть, выбрав пункт системного меню View/Markov Calculation Results. Значений показателей по состояниям приводятся на вкладке Calculation Properties/Result.
8. Создать отчеты нужного формата с помощью Мастера Отчетов (File/Print Preview).

1.2. ОСНОВНЫЕ ПОЛОЖЕНИЯ И МЕТОДИЧЕСКИЕ ОСНОВЫ РЕШЕНИЯ ЗАДАЧ ОЦЕНКИ НАДЕЖНОСТИ, БЕЗОПАСНОСТИ И РИСКА СИСТЕМ НА ОСНОВЕ ТЕХНОЛОГИИ АВТОМАТИЗИРОВАННОГО СТРУКТУРНО-ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ, РЕАЛИЗОВАННОЙ В ПРОГРАММНЫХ КОМПЛЕКСАХ ПК АСМ КОМПАНИИ "СПИК СЗМА"

1.2.1. Содержание этапов ОЛВМ

Теоретической основой технологии автоматизированного структурно-логического моделирования является *общий логико-вероятностный метод* (ОЛВМ) моделирования и расчета надежности, живучести и безопасности структурно и качественно сложных системных объектов и процессов [3-5]. В ОЛВМ расчета надежности аппарат математической логики используется для первичного графического и аналитического описания условий реализации функций отдельными и группами элементов в проектируемой системе, а методы теории вероятностей и комбинаторики применяются для количественной оценки безотказности и/или опасности функционирования проектируемой системы в целом. Для использования ОЛВМ должны задаваться специальные структурные схемы функциональной целостности исследуемых систем, логические критерии их функционирования, вероятностные и другие параметры элементов.

В основе постановки и решения всех задач моделирования и расчета надежности систем с помощью ОЛВМ лежит так называемый событийно-логический подход. Этот подход предусматривает последовательное выполнение следующих четырех основных этапов ОЛВМ:

1. Этап структурно-логической постановки задачи, который включает в себя:
 - разделение всей рассматриваемой системы на конечное число N элементов $i = 1, 2, \dots, N$, каждый из которых представляется в модели надежности простым (бинарным) событием x_i с двумя возможными состояниями $\bar{x}_i = \{x_i, \bar{x}_i\}$, например, работоспособности/отказа, готовности/неготовности, поражения/не поражения и т.п. и заданными вероятностными параметрами $p_i(t)$, или $q_i(t) = 1 - p_i(t)$;
 - определение содержания и логических условий реализации y_i и/или не реализации \bar{y}_i выходных функций для каждого элемента в системе;
 - логически строгое вербальное и графическое (аналитическое) описание множества X отдельных элементов системы и множества условий Y реализации ими своих системных функций, которые в совокупности $G(X, Y)$ образуют специальную *схему функциональной целостности* (СФЦ) рассматриваемой системы;
 - логически строгое описание и задание с помощью отдельных или групповых выходных функций *логических критериев функционирования* (ЛКФ) системы $Y_F = Y_F(\{y_i\}, i = 1, 2, \dots, N)$ реализации основных функций и/или возникновения опасных состояний системы.
2. Этап логического моделирования, на котором с помощью специальных методов преобразования СФЦ и ЛКФ осуществляется построение логической *функции работоспособности системы* (ФРС) $Y_F = Y_F(\{\bar{x}_i\}, i = 1, 2, \dots, N)$. Логическая ФРС позволяет аналитически строго в компактной форме определить все комбинации состояний элементов $\bar{x}_i, i = 1, 2, \dots, N$, в которых (и только в которых) система реализует свою выходную функцию F .
3. Этап вероятностного моделирования, на котором с помощью специальных методов преобразования ФРС осуществляется построение многочлена расчетной *вероятност-*

ной функции (ВФ) $P_F(\{p_i(t), q_i(t)\}, i = 1, 2, \dots, H; t)$. Многочлен ВФ позволяет аналитически строго определить закон распределения времени безотказной работы системы по реализации выходной функции F , заданной логическим критерием функционирования.

4. Этап выполнения расчетов показателей надежности, которые выполняются на основе ВФ и заданных параметров надежности элементов.

В технологии автоматизированного структурно-логического моделирования вручную выполняется только первый этап структурно-логической постановки задач оценки надежности, безопасности и риска сложных систем. Этапы построения логических ФРС, многочленов ВФ и расчетов показателей выполняются с помощью специальных программных комплексов автоматизированного структурно-логического моделирования (ПК АСМ) [6-11, 19].

Главную творческую работу по постановке задачи и анализу результатов проектного расчета надежности выполняют проектировщик и системный аналитик. Они логически описывают проектируемую систему, строят СФЦ для каждой из ее главных подсистем и функций, определяют параметры надежности элементов, задают логические критерии реализации функций, анализируют результаты расчетов, вырабатывают и реализуют проектные решения и подготавливают отчет.

Вся громоздкая и трудоемкая работа по построению логических, расчетных математических моделей и вычислению показателей надежности в технологии АСМ выполняется автоматически, с помощью специальных программных средств комплекса автоматизированного структурно-логического моделирования.

1.2.1.1. Средства и методы постановки задач в ОЛВМ

Основное содержание этапа постановки задач расчета надежности проектируемой системы характеризуется схемой на рис.2.1.



Рис.2.1. Схема постановки задачи в ОЛВМ

Исходными данными для постановки задачи выступают требования ТЗ или ПОН и все знания и данные о проектируемой системе, методах, средствах и возможностях используемой технологии автоматизированного структурно-логического моделирования.

Исходные данные для постановки задачи готовит разработчик в свободной, но логически строгой форме описания организационно-технических методов и средств обеспечения надежности проектируемой системы. Оно включает в себя описание функциональной структуры системы, ее главных функций и аварийных ситуаций, состава и параметров надежности элементов. Обязательно должны быть сформулированы (текстуально и/или графически) условия, при выполнении которых каждый элемент реализует свое функциональное назначение в системе.

Окончательную формализованную постановку задачи осуществляет системный аналитик. На основе полученного от разработчика описания системы, для каждой ее функции он строит схемы функциональной целостности, задает логические критерии функционирования, фиксирует параметры надежности элементов и уточняет перечень рассчитываемых показателей надежности системы.

Главное место в постановке задач автоматизированного структурно-логического моделирования надежности занимает построение схем функциональной целостности для каждой заданной выходной функции проектируемой системы.

Изобразительные средства и типовые фрагменты СФЦ. Любой графический аппарат, в сущности, является языком описания структур систем. Как и все другие языки, он определяется алфавитом, синтаксисом и семантикой. Алфавит СФЦ образуют ее основные изобразительные средства, которые включают в себя два вида вершин, два вида выходов для представления исходящих ребер (дуг) и два вида входов для представления заходящих дуг.

1.2.1.1.1.1 Функциональные вершины СФЦ

Главное назначение функциональных вершин состоит в графическом обозначении прямых исходов бинарных событий $\tilde{x}_i = x_i$, которыми в СФЦ представляются элементы i исследуемой системы. На рис.2.2 приведено графическое представление и аналитическое описание функциональной вершины.



$i \in \mathbb{N}$ - номер функциональной вершины (элементов, событий);

$\tilde{x}_i = x_i$ - свершение собственного, элементарного события;

$p_i(t)$ - вероятность свершения собственного элементарного события i .

Рис.2.2. Обозначение функциональной вершины в СФЦ

Примерами событий, представляемых функциональными вершинами в структурных схемах систем могут быть:

- сохранение работоспособности технического средства в течение заданного времени его работы (наработки);
- отказ технического средства в любой момент заданного времени работы;
- принятие (или не принятие) некоторого решения на определенном этапе процесса управления системой;
- правильное выполнение заданной функции (или ошибка) оператора на заданном этапе управления системой;
- выполнение (или не выполнение) заданной функции данным техническим средством или подсистемой;
- поражение (или не поражение) объекта в результате возникновения аварийной ситуации или вредного воздействия и др.

1.2.1.1.1.2 Дизъюнктивные ребра (дуги) в СФЦ

Направленность дизъюнктивных ребер в СФЦ обозначается стрелками. Вершина, из которой дуга исходит, называется *обеспечивающей*, а вершина, в которую дуга заходит, – *обеспечиваемой*.

Дизъюнктивные ребра (дуги) в СФЦ предназначены для графического отображения следующих двух видов информации:

а) дуга, исходящая из вершины i , обозначается y_i (прямая выходная дуга) и представляет все условий, необходимые для реализации (выполнения) выходной функции этого элемента;

б) дуга y_j , исходящая из вершины j и заходящая в вершину i , обозначает одно из условий обеспечения реализации выходной функции y_i .

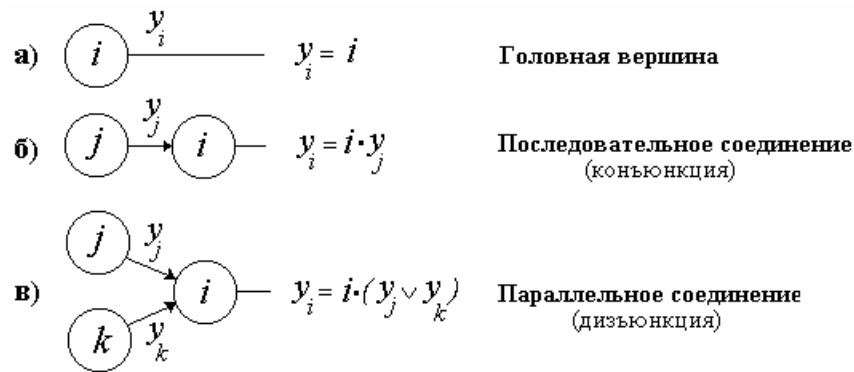


Рис.2.3. Типовые варианты использования дизъюнктивных дуг в СФЦ

Вариант структурной схемы на рис.2.3.а представляет вершину без заходящих в нее дуг (без обеспечения). Такие вершины называют *головными*. Элементы систем, представляемые в СФЦ головными вершинами, считаются достоверно обеспеченными. Это значит, что условия реализации прямой выходной функции y_i головной вершины определяется свершением только собственного события x_i . Аналитически такое условие определяется следующим логическим уравнением:

$$y_i = x_i \quad (1.2.1)$$

Схема на рис.2.3.б представляет одиночную связь функционального подчинения элемента i элементу j . Это означает, что прямая выходная функция y_i элемента i реализуется, если свершилось собственное событие x_i и реализована обеспечивающая функция y_j элемента j . Логическое уравнение в этом случае имеет вид

$$y_i = x_i \cdot y_j \quad (1.2.2)$$

Одиночная связь функционального подчинения в СФЦ эквивалентна последовательному соединению вершин в графах связности и деревьях событий.

Последняя схема, изображенная на рис.2.3.в., представляет условия дублированного (резервированного, дизъюнктивного) обеспечения элемента i системы хотя бы одной из двух интегративных функций y_j или y_k . Говорят, что между функциями y_j и y_k в системе реализованы дизъюнктивные (логическая операция "ИЛИ") организационные отношения. Логическое уравнение, соответствующее рис.П.3.в, имеет вид:

$$y_i = x_i \cdot (y_j \vee y_k) \quad (1.2.3)$$

Дизъюнктивные организационные отношения в СФЦ являются аналогами известных параллельных соединений элементов в графах связности.

Рассмотренные изобразительные средства и логические уравнения представляют все графические и логические возможности деревьев событий и графов связности, которые нашли широкое применение в известных структурных методах анализа надежности систем. Вместе с тем, аппарат СФЦ обладает рядом новых дополнительных возможностей. Они позволяют строить модели надежности систем со сложной логической организацией работы элементов.

1.2.1.1.1.3 Фиктивные вершины в СФЦ

В отличие от функциональных фиктивные вершины не представляют никаких элементов моделируемой системы и служат только для расширения возможностей графического представления сложных логических связей и отношений между различными интегративными функциями.

$$\varphi \quad \bigcirc \quad \begin{array}{l} x_\varphi = I \quad - \text{логическая единица, истина;} \\ p_\varphi = 1 \quad - \text{достоверное событие.} \end{array}$$

Рис.2.4. Обозначение фиктивной вершины в СФЦ

Собственно фиктивная вершина φ в СФЦ рассматривается как логическая константа **I** (истина), т.е. некоторое условное, достоверное событие, вероятность реализации которого $p_\varphi(t)$ принимается равной единице. В общем логико-вероятностном методе фиктивные вершины имеют следующее аналитическое определение.

$$x_\varphi = \mathbf{I} \text{ (истина); } p_\varphi(t) = p_\varphi = 1; \quad \overline{x_\varphi} = \mathbf{O} \text{ (ложь); } q_\varphi(t) = q_\varphi = 0 \quad (1.2.4)$$

Примеры использования фиктивных вершин в СФЦ приведены на рис.2.5.

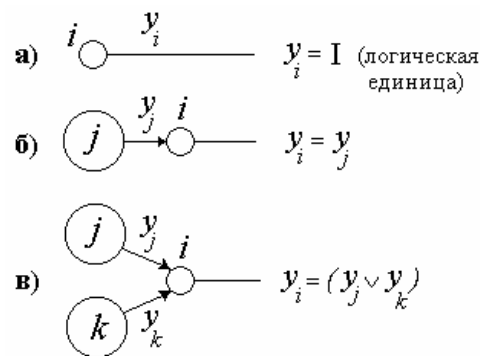


Рис.2.5. Типовые варианты использования фиктивных вершин в СФЦ

Из сравнения рис.2.5. и рис.2.3. следует, что логические уравнения выходных интегративных функций для фиктивных вершин отличаются от аналогичных уравнений для функциональных вершин только отсутствием в записи обозначений простых логических переменных x_i .

Фиктивные вершины в СФЦ являются аналогами точечных соединений дуг или логических объединений ребер, применяемых в графах связности и деревьях событий.

Введенные графические и аналитические обозначения фиктивных вершин позволяют использовать их в СФЦ совместно с функциональными вершинами, с сохранением всех правил логического и вероятностного моделирования.

Рассмотренных графических средств СФЦ достаточно, чтобы успешно решать все задачи расчета надежности систем, функционирование которых представляется последовательно-параллельными и мостиковыми графами связности.

1.2.1.1.1.4 Конъюнктивные дуги в СФЦ

Конъюнктивные дуги используются в СФЦ наравне с дизъюнктивными дугами (см.рис.2.3). Однако направленность связей логического подчинения на концах конъюнктивных дуг обозначается не стрелками, а точками. С помощью конъюнктивных дуг в СФЦ представляются условия, когда элементы параллельных ветвей не дублируют (не резервируют) друг друга, а должны функционировать все вместе, одновременно.

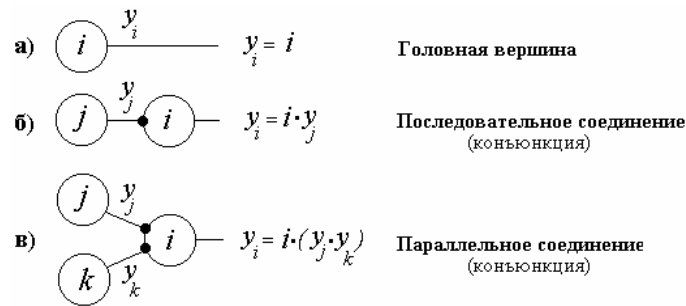


Рис.2.6. Типовые варианты использования конъюнктивных дуг в СФЦ

Сравнивая рис.2.6. и рис.2.3 видим, что, по определению, головные вершины и последовательные соединения, изображенные в СФЦ с помощью одиночных дизъюнктивных и одиночных конъюнктивных дуг, являются совершенно одинаковыми как в логическом, так и в вероятностном смысле.

Главное назначение конъюнктивных дуг в СФЦ состоит в обеспечении возможности корректного учета зависимостей, требующих одновременного функционирования нескольких параллельных элементов, ветвей или подсистем структурной схемы. Так логическое условие реализации выходной функции (y_i) схемы, изображенной на рис.2.6.в, состоит в одновременной реализации функций (y_j и y_k), двух разных элементов j и k , что не могло быть представлено раньше с помощью графов связности.

$$y_i = x_i \cdot (y_j \cdot y_k) \quad (1.2.5)$$

Конъюнктивные дуги в СФЦ позволяют иным, чем последовательное соединение, способом представить условия одновременной (конъюнктивной) реализации нескольких обеспечивающих функций в системе.

1.2.1.1.2 Использование дизъюнктивных и конъюнктивных дуг в СФЦ

Важное место в СФЦ занимают приемы представления сложных смешанных организационных отношений между несколькими обеспечивающими функциями вершин. В этом случае обеспечение элемента i определяется реализацией одновременно нескольких дизъюнктивных и конъюнктивных условий. Обобщенный вариант монотонной схемы такого смешанного фрагмента обеспечения в СФЦ изображен на рис.2.7.

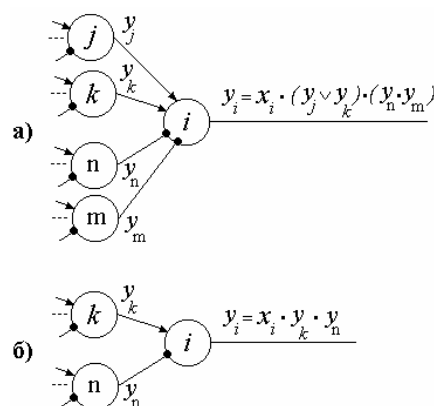


Рис.2.7. Варианты смешанного обеспечения вершин СФЦ

СФЦ на рис.2.7.а представляет фрагмент системы, где, например, исполнительный элемент i обеспечивается тремя видами ресурсов:

j, k - электроэнергией;

n - водой;

m - воздухом.

Условия обеспечения элемента i воздухом y_m и водой y_n не резервированы, а обеспечение электроэнергией осуществляется от двух дублированных источников y_j или y_k .

Следует отметить, что в смешанных организационных отношениях между группами дизъюнктивных и группами конъюнктивных заходящих дуг всегда устанавливается знак логического умножения (знак конъюнкции). Это означает, что общая функция смешанного обеспечения реализуется, если имеются одновременно - все без исключения функции конъюнктивной группы и хотя бы одна функция дизъюнктивной группы. Полезно осмыслить один вырожденный, частный случай смешанного обеспечения, СФЦ которого приведена на рис.2.7.б. Здесь присутствуют две одиночные заходящие дуги, обеспечивающие элемент i , одна дизъюнктивная y_k и одна конъюнктивная y_n . По правилам построения СФЦ между этими функциями в логическом уравнении записывается знак логического умножения.

1.2.1.1.3 Размножение функциональных вершин

Многие элементы в современных технических и организационных системах являются функционально сложными, т.е. могут выполнять (или не выполнять) не одну, а сразу несколько различных функций. Для реализации каждой из этих функций может потребоваться разное по составу и организации обеспечение. При этом отказ такого многофункционального элемента приводит к отказу реализации всего множества его различных выходных функций.

Деревьями событий и графами связности такие многофункциональные элементы не представляются. Для учета многофункциональных элементов в СФЦ реализована возможность размноженного представления (размножения) функциональных вершин. То есть, в СФЦ разрешается отображение одного многофункционального элемента системы с помощью нескольких размноженных функциональных вершин. Число размноженных вершин обычно равно числу разных функций данного элемента, различные условия реализации которых должны быть представлены в СФЦ.

Размноженные вершины должны иметь в СФЦ разные схемные номера. Эти номера объединяются в группу, которой сопоставляется один общий системный номер и параметр одного данного события. Типовая запись группы размноженных вершин следующая:

$$\{i, j, k, m, n\} = i, \quad p_i \quad (1.2.6)$$

В этой группе номер i является основным, системным. Он определяет индекс одного случайного события x_i безотказной работы многофункционального элемента и его вероятностного параметра p_i . Все остальные номера j, k, m, n группы являются дополнительными, схемными и определяют в СФЦ вершины, размноженные по разным функциям данного элемента i . Однако, все эти номера и вершины представляют с СФЦ одно единственное простое событие x_i с собственным вероятностным параметром p_i .

В одной СФЦ может быть несколько групп размноженных функциональных вершин. Вся информация о таких группах записывается отдельно от схемы в форме (1.2.6).

1.2.1.1.4 Представление логической операции инверсирования в СФЦ

Для реализации в ОЛВМ всех возможностей основного аппарата моделирования алгебры логики в СФЦ предусмотрена возможность графического представления логической операции "НЕ" (инверсирования). Таким образом, набор логических операций, графически представляемых в СФЦ, является функционально полным ("И", "ИЛИ", "НЕ").

Графическое отображение логической операции инверсирования реализовано с помощью еще одного, второго вида выхода дуги из вершин СФЦ, который назван инверсным выходом. Этот выход можно отображать на любой функциональной, фиктивной или размноженной вершине. Типовые примеры изображения инверсных выходов вершин в СФЦ приведены на рис.2.8.

СФЦ на рис.2.8.а отличается от СФЦ на рис.2.7.а наличием второго инверсного выхода \bar{y}_i функциональной вершины i . По смысловому содержанию инверсный выход является точной логической противоположностью прямого выхода из вершины, который применялся во всех ранее рассмотренных структурных схемах. Например, если прямой выход используется для обозначения условия работоспособности или безопасности системы, то инверсный выход точно и однозначно представляет условия неработоспособности этой системы или условия возникновения аварийной ситуации.

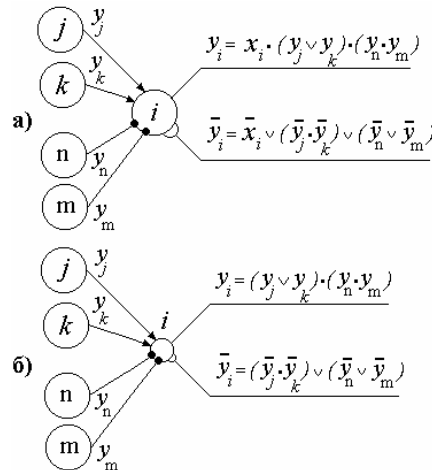


Рис.2.8. Примеры изображения инверсных выходов в СФЦ

На рис.2.8.б приведено изображение и логическое уравнение инверсного выхода из фиктивной вершины СФЦ.

1.2.1.1.5 Обобщенный структурный фрагмент СФЦ

Рассмотренные выше изобразительные средства СФЦ (см. рис.2.2-2.8 и выражения (2.1-2.5)), можно представить одним обобщенным структурным фрагментом и двумя базовыми аналитическими соотношениями (логическими уравнениями). Схема обобщенного структурного фрагмента СФЦ приведена на рис.2.9.

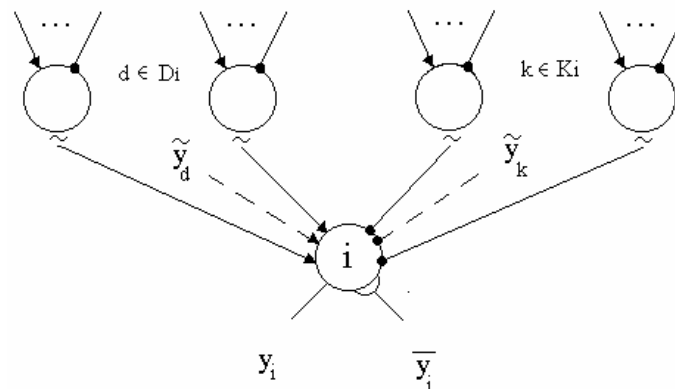


Рис.2.9. Схема обобщенного фрагмента СФЦ

На рис.2.9. в обобщенном виде представлены все рассмотренные выше типовые фрагменты СФЦ. Обеспечивающие вершины разделены на две группы - дизъюнктивную и конъюнктивную. Номера d обеспечивающих вершин дизъюнктивной группы составляют множество D_i , а номера k обеспечивающих вершин конъюнктивной группы составляют

множество K_i . С помощью знака “тильда” на схеме обозначена возможность использования в СФЦ любого из двух возможных выходов обеспечивающих вершин - прямого или инверсного. Знаком тильды отмечены и выходные интегративные функции обеспечивающих вершин, которые также могут быть прямыми или инверсными

$$\tilde{Y}_d = \{y_d, \bar{y}_d\}; \quad (1.2.7)$$

$$\tilde{Y}_k = \{y_k, \bar{y}_k\};$$

С учетом принятых обозначений интегративные функции прямого и инверсного выходов вершины i обобщенного фрагмента СФЦ определяются следующими универсальными, базовыми логическими уравнениями.

Для прямой выходной интегративной функции

$$a). y_i = i \cdot \left(\bigvee_{d \in D_i} \tilde{Y}_d \right) \cdot \left(\big\& \tilde{Y}_k \right), \quad i - \text{функциональная вершина} \quad (1.2.8)$$

$$b). y_i = \left(\bigvee_{d \in D_i} \tilde{Y}_d \right) \cdot \left(\big\& \tilde{Y}_k \right), \quad i - \text{фиктивная вершина}$$

Для инверсной выходной интегративной функции

$$a). \bar{y}_i = \bar{i} \cdot \left(\big\& \bar{\tilde{Y}}_d \right) \cdot \left(\bigvee_{k \in K_i} \bar{\tilde{Y}}_k \right), \quad i - \text{функциональная вершина} \quad (1.2.9)$$

$$b). \bar{y}_i = \left(\big\& \bar{\tilde{Y}}_d \right) \cdot \left(\bigvee_{k \in K_i} \bar{\tilde{Y}}_k \right), \quad i - \text{фиктивная вершина}$$

Выражения (2.8.a) и (2.9.a) определяют случай, когда вершина i функциональная, а (2.8.б) и (2.9.б), когда вершина i фиктивная (см. рис.2.4.).

В базовых аналитических уравнениях части, выделенные круглыми скобками, называют, соответственно, дизъюнктивной и конъюнктивной группами составляющих функции обеспечения (ФО) вершины i . В дизъюнктивные группы объединены выходные интегративные функции \tilde{Y}_d тех вершин (множество номеров $d \in D_i$), которые обеспечивают реализацию прямого выхода y_i вершины i по логике "ИЛИ", а в конъюнктивные группы объединены выходные интегративные функции \tilde{Y}_k тех вершин (множество номеров $k \in K_i$), которые обеспечивают реализацию прямого выхода y_i вершины i по логике "И".

Из (1.2.8) и (1.2.9) могут быть легко получены все рассмотренные выше частные фрагменты СФЦ и базовых логических соотношений. Например, если вершина i функциональная и в схеме ее обеспечения отсутствует (равна логической единице **1**) конъюнктивная часть ФО (в схеме нет заходящих дуг с точками), то (2.8.a) точно представляет аппарат графов связности (см. выражения (1.2.1)- (1.2.3)).

Если в схеме отсутствует дизъюнктивная часть ФО (нет заходящих дуг со стрелками), то из (2.8.a) получаем все типовые варианты использования конъюнктивных дуг в СФЦ (см. выражение (1.2.5)). Отсутствие в записях функций (1.2.8) и (1.2.9) всей ФО определяет головную (функциональную или фиктивную) вершину в СФЦ.

1.2.1.2. Методика построения СФЦ

1.2.1.2.1 Общие замечания и рекомендации

Построение схемы функциональной целостности является основой постановки задач проектного анализа надежности и безопасности систем. Как и любая другая постановочная задача, построение СФЦ, в своей главной содержательной части, является сугубо творческим, неформальным и, следовательно, абсолютно не алгоритмизируемым процессом. Постановку задачи проектного расчета надежности могут осуществлять профессиональные специалисты двух предметных областей, работающие совместно:

- разработчики (работники проектных отделов) и/или пользователи (заказчики), создающие и эксплуатирующие системные объекты и получившие подготовку по основам технологии автоматизированного структурно-логического моделирования;
- системные аналитики (обычно работники исследовательских отделов), профессионально владеющие технологией автоматизированного структурно-логического моделирования.

При построении СФЦ разработчик осуществляет идентификацию и предварительное общее, но логически строгое, описание условий реализации функций каждым элементом и исследуемой системой в целом. Системный аналитик выполняет окончательное строго формальное построение СФЦ по каждой функции проектируемой системы. После согласования этих СФЦ с разработчиком они используются в проектном расчете надежности системы.

При логическом описании и построении СФЦ необходимо учитывать ограничения и допущения, принятые в ОЛВМ моделирования и расчета надежности систем. Типовые допущения и ограничения следующие:

- бинарность, то есть возможность находиться только в двух несовместных состояниях, случайных событий $\tilde{x}_i = \{x_i, \bar{x}_i\}$, с помощью которых представляются (моделируются) элементы $i = 1, 2, \dots, N$ системы;
- независимость в совокупности всех выделенных бинарных событий, которая означает, что собственные вероятностные параметры надежности $p_i(t)$ соответствующих элементов не изменяется в зависимости от того, отказали или не отказали другие элементы системы;
- неограниченность процессов восстановления элементов в моделях надежности восстанавливаемых систем, которая означает, что интенсивность ремонтных работ любого элемента не зависит от того, сколько элементов системы уже ремонтируется;
- в моделях надежности законы распределения случайного времени безотказной работы и восстановления всех элементов принимаются экспоненциальными;
- статистическая устойчивость вероятностных характеристик элементов, которая означает, что на рассматриваемом интервале времени работы системы собственные исходные параметры законов распределений элементов не изменяются.

Указанные допущения и ограничения свойственны большинству известных методик расчета надежности. В технологии АСМ эти ограничения и допущения не являются абсолютными и характеризуют только типовой набор средств для проектного расчета надежности и безопасности объектов. При необходимости, многие из ограничений могут быть сняты путем совершенствования теории и адаптации технологических средств АСМ к конкретному классу задач проектной оценки надежности систем.

1.2.1.2.2 Основные этапы построения СФЦ

1. Выделение в проектируемой системе конечного числа элементов, которые в модели надежности представляются бинарными событиями, а в СФЦ функциональными вершинами. При этом следует учитывать следующие рекомендации:

- Каждый выделенный элемент примерно в два раза увеличивает расчетную модель надежности системы. Поэтому число элементов не следует делать очень большим. Вместе с тем, недостаточное число элементов может привести к недопустимому снижению точности модели и проектной оценки надежности системы. Искусство выделения достаточного числа элементов определяется опытом разработчиков и постоянным сравнительным анализом различных вариантов. Для ориентировки можно сообщить, что последние проектные расчеты надежности содержали в структурных моделях систем от нескольких десятков до одной - двух сотен элементов.

- Все выделенные элементы должны иметь наименования, соответствующие реальным объектам проектируемой системы. Одновременно, в логических описаниях элементов и вершин СФЦ должно быть четко и однозначно определено смысловое содержание каждого из двух состояний (несовместных исходов) всех бинарных событий.
- Выделение бинарных событий должно соответствовать указанным выше допущениям и ограничениям ОЛВМ моделирования и расчета надежности. Приемы и способы снятия каких-либо из этих допущений в отчетных материалах описываются отдельно.
- Для учета в моделях надежности многофункциональных элементов или подсистем выделяются группы размноженных функциональных вершин СФЦ и устанавливается соответствие их схемных и системных номеров вида (1.2.6).
- Все выделяемые бинарные события должны быть параметрически определяемыми. То есть должна быть обеспечена возможность обоснованного задания (нахождения, определения, вычисления и др.) числовых значений собственных вероятностных параметров надежности каждого элемента.
- Для каждой функции системы и/или варианта аварийной ситуации (если они заданы) выделяется набор (подмножество) необходимых бинарных событий. Затем для каждой функции (аварийной ситуации) разрабатывается логически строгое описание условий реализации системных функций элементами и соответствующий граф $G(X, Y)$ схемы функциональной целостности. При выполнении этой работы полезно учитывать следующие рекомендации:

Для каждого выделенного бинарного события x_i определяются:

а). смысловое содержание выходной функции y_i соответствующего элемента i в системе;

б). все логические условия реализации этой выходной функции в системе, сформулированные в терминах собственного события x_i и обеспечивающих функций y_j, y_k, \dots других элементов системы.

- Для представления сложных логических отношений и для повышения наглядности, в описание структуры системы могут вводиться фиктивные вершины, на которых осуществляется необходимая группировка логических условий.
- Главный критерий описания и построения СФЦ заключается в том, чтобы логически определить и графическими средствами СФЦ изобразить все условия реализации в системе прямых выходных функций каждой выделенной функциональной и каждой введенной фиктивной вершиной. После этого построение структурной модели $G(X, Y)$ считается законченным.
- Каждое законченное логическое описание и соответствующая СФЦ могут представлять одну или сразу несколько функций (или аварийных ситуаций). Поэтому для выполнения следующих этапов моделирования и расчета надежности необходимо задать один или сразу несколько логических критериев функционирования системы. Как описание ЛКФ представляет собой высказывание, в котором с помощью выходных функций $y_i, i = 1, 2, \dots, N$ одного или нескольких элементов задается условие функционирования всей системы. Аналитически ЛКФ представляет собой булеву функцию следующего вида:

$$Y_F = f(\{y_i\}, i = 1, 2, \dots, N) \quad (1.2.10)$$

Здесь F определяет номер функции (номер вида аварийной ситуации) системы, а N - общее число функциональных, размноженных и фиктивных вершин СФЦ.

2. Проверка полноты и правильности построения СФЦ и задания ЛКФ. Такая проверка должна выполняться постоянно в процессе проектного расчета надежности. Она предусматривает многократное, порой итерационное повторения всех этапов построения СФЦ, в целях ее уточнения, исправления, расширения и проверки.

1.2.1.2.3 Задание логических критериев и параметров элементов

После построения СФЦ, на завершающем этапе постановки задачи расчета надежности проектируемой системы, определяются параметры надежности элементов и логические критерии ее функционирования.

1.2.1.3. Средства и методы построения математических моделей

Проектный расчет надежности с помощью технологии АСМ предусматривает последовательное построение двух видов расчетных моделей:

- логической функции работоспособности системы

$$Y_F = Y_F(\{x_i\}, i = 1, 2, \dots, H);$$

- многочлена расчетной вероятностной функции

$$P_F(t) = p_F(\{p_i(t), q_i(t)\}, i = 1, 2, \dots, H; t).$$

1.2.1.3.1 Построение логических функций работоспособности систем

В качестве логической модели надежности системы используется булева функция, которую называют логической функцией работоспособности системы (ФРС). Логическая ФРС представляет множество состояний, в которых система реализует соответствующий критерий своего функционирования. Это множество иногда называют областью состояний работоспособности системы. Однако в ОЛВМ понятие логической ФРС принято толковать расширенно, как логическая форма представления тех состояний системы, в которых (и только в которых) реализуется заданный критерий ее функционирования. При этом, собственно логический критерий может определять самые разные свойства системы - работоспособность, неработоспособность, живучесть, безопасность, уровень эффективности, риск, готовность, неготовность и т.д.

На рис.2.10. изображена общая схема задачи определения логической ФРС. Исходными данными здесь являются СФЦ и ЛКФ системы.



Рис.2.10. Схема задачи определения логической ФРС

В основе процедуры определения логических ФРС лежат специальные методы и программные средства решения систем логических уравнений (СФЦ) для каждого заданного ЛКФ системы. Аппарат СФЦ и метод, алгоритм и программа определения монотонных и немонотонных ФРС являются единым целым и составляют основу ОЛВМ, технологии и всех программных комплексов автоматизированного структурно-логического моделирования систем (авторы и правообладатели Можаяев А.С. и Гладкова И.А. [10, 11]).

1.2.1.3.2 Построение многочленов вероятностных функций

В качестве расчетной вероятностной модели надежности системы используется многочлен вероятностной функции $P_F(t) = p_F(\{p_i(t), q_i(t)\}, i=1, 2, \dots, H; t)$, который должен точно определять (в рамках принятых ограничений и допущений) закон распределения времени безотказной работы системы, все состояния работоспособности которой представлены с помощью логической ФРС $Y_F = Y_F(\{\bar{x}_i\}, i=1, 2, \dots, H)$.

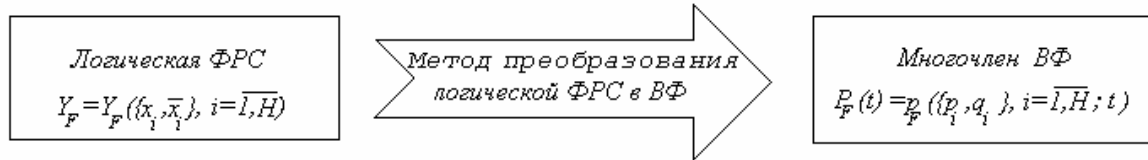


Рис.2.11. Схема задачи определения ВФ

Параметрами $p_i(t), q_i(t) = 1 - p_i(t)$ многочлена ВФ являются, в простейшем случае, собственные вероятности безотказной работы или коэффициенты готовности отдельных элементов системы. В общем случае, это законы распределения времени безотказной работы всех элементов проектируемой системы.

В настоящее время разработано большое количество методов преобразования логических ФРС в ВФ. Все они характеризуются высокой громоздкостью и трудоемкостью процедур перехода, и их практическое применение возможно только при полной автоматизации. В ПК АСМ реализованы два метода:

1. Комбинированный метод, алгоритм и программа преобразования логических ФРС в многочлены ВФ (автор Можаяев А.С.) [3-11];
2. Метод, алгоритм и программа ортогонализации логической ФРС (автор Татусьян О.В.) [19].

1.2.1.4. Методы вычисления показателей надежности и безопасности систем

Обобщенная схема завершающего четвертого этапа проектного расчета надежности приведена на рис.2.12.

В правой части схемы приведены типовые показатели надежности, вычисление которых предусмотрено требованиями государственных стандартов и руководящих документов по надежности автоматизированных систем управления. В технологии АСМ расчет указанных показателей выполняется на основе многочленов вероятностных функций, параметров надежности элементов и инверсированных логических функций работоспособности, представляющих все минимальные сечения отказов по каждой функции по проектируемой системе в целом.

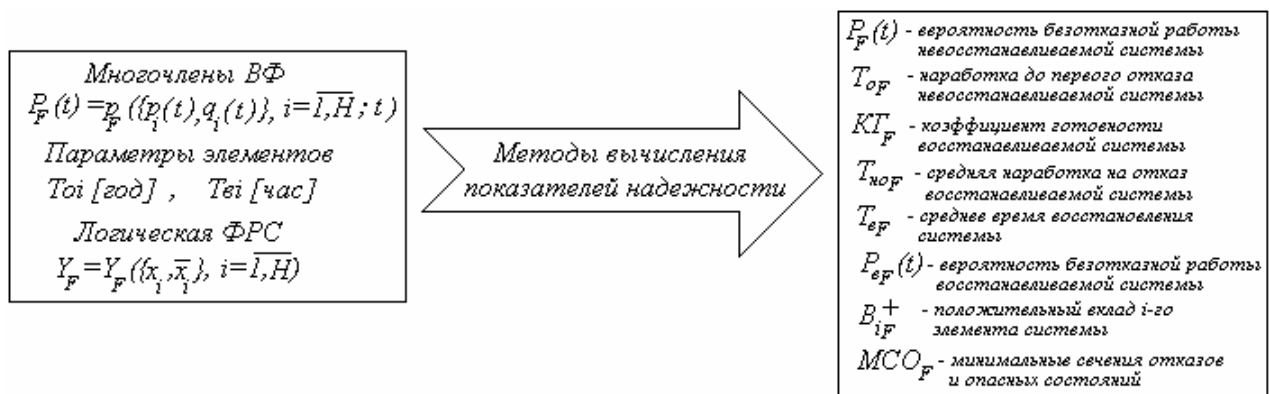


Рис.2.12. Схема задачи вычисления показателей надежности

1.2.1.4.1 Расчет безотказности невосстанавливаемых систем

Справка. Невосстанавливаемыми называют системные объекты, в которых на рассматриваемом интервале времени t функционирования все элементы $i=1,2,\dots,H$ могут независимо отказывать, с интенсивностью T_{oi} , но ни один из этих элементов после отказа не восстанавливается.

1.2.1.4.1.1 Расчет вероятности безотказной работы восстанавливаемой системы

Исходными данными для расчета этого показателя являются:

- многочлен $P_F(t)$ вероятностной функции рассматриваемого режима F функционирования исследуемой системы

$$P_F(t) = p_F(\{p_i(t), q_i(t)\}, i=1,2,\dots,H;t) \quad (1.2.11)$$

- числовые значения средней наработки до отказа T_{oi} всех элементов системы в годах;
- требуемое время t функционирования системы (наработка системы).

Сначала выполняется расчет вероятностей безотказной работы и отказа элементов системы для экспоненциального закона распределения времени их безотказной работы

$$\begin{aligned} p_i(t) &= e^{-\frac{1}{T_i}t}; \\ q_i(t) &= 1 - p_i(t). \end{aligned} \quad (1.2.12)$$

Подставляя аналитические выражения (1.2.12) в многочлен ВФ получаем закон распределения времени безотказной работы системы по реализации функции F . На основе этого закона выполняются расчеты вероятности безотказной работы системы для заданной наработки t .

Следует отметить, что при условии независимости отказов элементов подобная подстановка возможна не только для экспоненциального, но и для любых других законов распределения времени безотказной работы элементов, если вид и параметры этого закона известны. Причем, эти законы могут быть разными для различных элементов рассматриваемой системы в целом.

1.2.1.4.1.2 Расчет средней наработки до отказа

Аналитическое решение этой задачи связано с нахождением интеграла от многочлена (1.2.13) вероятностной функции надежности системы:

$$T_{oF} = \int_0^{\infty} P_{oF}(t) dt; \quad (1.2.13)$$

Если многочлен (1.2.11) представить в прямой форме без использования параметра $q_i(t)$ (заменив его на $1 - p_i(t)$), то интеграл (1.2.13) для экспоненциального закона берется непосредственно, в общем виде, и соответствующая расчетная формула средней наработки до первого отказа составляет

$$T_{oF} = \sum_{j=1}^M (zn_j) \frac{1}{\sum_{i \in K_j} \frac{1}{T_{oi}}} \quad (1.2.14)$$

Здесь M - число одночленов в преобразованном многочлене ВФ, (zn_j) - знак перед j -ым одночленом, а K_j - множество номеров элементов i , параметры $p_i(t)$ которых вошли в j -ый одночлен.

1.2.1.4.2 Расчет безотказности восстанавливаемых систем

Справка. Восстанавливаемыми называют системы, в которых все элементы, $i = 1, 2, \dots, H$, могут независимо отказывать с заданной интенсивностью $\lambda_i(t) = \lambda_i = const$, и неограниченно восстанавливаться, т.е. переходить в состояние работоспособности, с заданной интенсивностью восстановления $\mu_i(t) = \mu_i = const$.

1.2.1.4.2.1 Расчет коэффициентов готовности элементов

Надежность элементов в восстанавливаемой системе характеризуется двумя параметрами

$$T_{oi} = \frac{1}{\lambda_i}; \quad T_{ei} = \frac{1}{\mu_i}. \quad (1.2.15)$$

Здесь T_{oi} - называют наработкой на отказ восстанавливаемого элемента. Оно равно среднему времени безотказной работы элемента между двумя соседними отказами. Для экспоненциального закона оно численно совпадает со средней наработкой до первого отказа восстанавливаемого элемента. Параметр T_{ei} определяет среднее время восстановления отказавшего элемента.

В качестве обобщенного исходного параметра надежности восстанавливаемого элемента может выступать его коэффициент готовности равный

$$KG_i = \frac{T_{oi}}{T_{oi} + T_{ei}} = \frac{\mu_i}{\lambda_i + \mu_i} \quad (1.2.16)$$

Можно говорить о следующих двух смысловых значениях коэффициента готовности. Во-первых, его значение определяет ту среднюю долю времени ($KG_i \cdot t$) из общей наработки t , в течение которой элемент является работоспособным, т.е. выполняет свои функции в системе. Остальную часть времени $(1 - KG_i) \cdot t$ элемент находится в состоянии отказа, т.е. восстанавливается. Во-вторых, коэффициент готовности равен вероятности события - застать восстанавливаемый элемент i в любой момент времени t его функционирования, в состоянии работоспособности. Аналогично, коэффициент неготовности $1 - KG_i$ равен вероятности - застать элемент в любой момент времени в состоянии неработоспособности (отказа, восстановления). Для принятых допущений эта вероятность не зависит от времени и является постоянной, на всем интервале времени функционирования объекта, кроме очень короткого начального переходного периода.

1.2.1.4.2.2 Расчет коэффициента готовности восстанавливаемой системы

Существует явное подобие в постановке и решении задач моделирования и расчета вероятности безотказной работы $P_F(t)$ невосстанавливаемой системы и коэффициента готовности KG_F той же системы, но при условии неограниченного восстановления элементов. В первой задаче вычисления $P_F(t)$ производятся с помощью многочлена ВФ, путем подстановки в него вероятностей безотказной работы $p_i = p_i(t)$ невосстанавливаемых элементов. Во второй задаче вычисления KG_F производятся с помощью того же многочлена ВФ, только в качестве параметров элементов используются коэффициенты готовности $p_i = KG_i$ восстанавливаемых элементов. Это позволяет, почти без изменений, применять методику расчета вероятности безотказной работы невосстанавливаемой системы, для определения коэффициента готовности той же, но восстанавливаемой системы.

1. Определяется многочлен функции рассматриваемого режима функционирования исследуемой системы;
2. Задаются или вычисляются значения коэффициентов готовности $p_i = KG_i$ и коэффициентов неготовности $q_i = 1 - p_i = 1 - KG_i$ всех $i = 1, 2, \dots, H$ элементов системы;

3. Эти аналитические или числовые значения параметров элементов подставляются в многочлен вероятностной функции.

В результате аналитической подстановки получается расчетная формула (расчетная математическая модель) коэффициента готовности системы.

При подстановке в ВФ числовых значений коэффициентов готовности элементов производится расчет коэффициента готовности системы в целом.

Напомним, что рассмотренная методика правомерна, если допустима гипотеза о независимости отказов и неограниченного восстановлений всех элементов системы.

1.2.1.4.2.3 Расчет средней наработки между отказами и среднего времени восстановления

Коэффициент готовности является самой распространенной, но, к сожалению, далеко не самой полной характеристикой надежности восстанавливаемой системы. Так, в частности, знание только KG_F не позволяет определить такие важные характеристики, как среднюю наработку между отказами $T_{ноF}$, среднее время восстановления $T_{вF}$ и вероятность безотказной работы $P_{вF}(t)$ восстанавливаемой системы.

В технологии АСМ для расчета указанных показателей применяются известные и новые приближенные аналитические методы [14]:

$$T_{ноF} = KG_F * \left(\frac{1}{\sum_{i=1}^H \frac{\partial KG_F}{\partial KG_i} * \lambda_i * KG_i} \right) \quad (1.2.17)$$

$$T_{вF} = (1 - KG_F) * \left(\frac{1}{\sum_{i=1}^H \frac{\partial KG_F}{\partial KG_i} * \lambda_i * KG_i} \right)$$

1.2.1.4.2.4 Расчет вероятности безотказной работы восстанавливаемой системы

В основе расчета этой важной характеристики лежат следующие допущения:

- вероятность безотказной работы восстанавливаемой системы характеризует событие наступления первого, даже очень кратковременного отказа системы в целом;
- случайное время до наступления первого отказа восстанавливаемой системы распределено по экспоненциальному закону с параметром

$$\lambda_{\ominus} = \frac{1}{T_{ноF} + T_{вF}}$$

Тогда, оценочный расчет вероятности первого отказа восстанавливаемой системы выполняется по известной формуле для экспоненциального закона

$$P_{вF}(t) = e^{-\lambda_{\ominus} t} = e^{-\frac{1}{T_{ноF} + T_{вF}} t} \quad (1.2.18)$$

Все рассмотренные методы расчета показателей надежности невосстанавливаемых и восстанавливаемых систем в технологии АСМ полностью автоматизированы и выполняются с помощью ПК АСМ. Это позволяет на этапе проектирования выполнять оперативный многовариантный анализ надежности высокоразмерных системных объектов произвольной структуры и организации функционирования.

1.2.1.4.2.5 Расчет значимостей и вкладов элементов

В настоящее время в технологии и ПК АСМ реализованы вычисления трех показателей роли элементов в обеспечении надежности и безопасности исследуемых систем [4, 5].

1. Значимость элементов системы

В самом общем случае показатель значимости ξ_i отдельного элемента i исследуемой системы определяется следующим соотношением

$$\xi_i = \frac{P_c}{p_i} = 1 - \frac{P_c}{p_i} = 0 \quad ; \quad i = 1, 2, \dots, H \quad (1.2.19)$$

Здесь $\frac{P_c}{p_i} = 1$ - значение вероятностной характеристики системы при абсолютной надежности элемента i , а $\frac{P_c}{p_i} = 0$ - при достоверном отказе элемента i на рассматриваемом интервале t времени функционирования. Анализ определений значимости (1.2.19) позволяет заключить:

1. Величина значимости ξ_i точно равна изменению значения системной характеристики P_c вследствие изменения собственного параметра p_i от 0 до 1, и фиксированных значениях параметров всех других элементов системы.
2. Диапазон значений вероятностного показателя значимости ξ_i составляет $[-1, 0, +1]$ включительно.
3. Отрицательное значение $\xi_i < 0$ характеризует, так называемое, вредное (понижающее) влияние элемента i на вероятность реализации заданного режима работы системы. В этом случае, например, увеличение надежности самого элемента i безусловно приводит к уменьшению надежности P_c всей системы в целом, а точнее - рассматриваемого режима ее функционирования. Отрицательные значимости элементов характерны для немонотонных логико-вероятностных моделей систем.
4. Нулевое значение характеристики значимости $\xi_i = 0$ говорит о том, что данный элемент i является несущественным для реализации рассматриваемого режима функционирования системы в целом (элемент i лишний, ненужный)
5. Положительное значение $\xi_i > 0$ определяет то максимально возможное увеличение надежности P_c систем, которое она может получить, если изменить надежность только одного элемента i от нуля до единицы включительно.
6. Все элементы монотонных систем могут иметь только положительные или нулевые значения характеристик их значимости.
7. Для случая, когда процессы отказов (или отказов и восстановлений) всех элементов системы являются независимыми в совокупности, значимости (1.2.19) элементов системы равны соответствующим частным производным

$$\xi_i = \frac{\partial P_c}{\partial p_i} \quad ; \quad i = 1, 2, \dots, H \quad (1.2.20)$$

2. Положительные и отрицательные вклады элементов системы

Наряду с характеристиками значимости, в логико-вероятностных методах анализа систем, все большее применение начинают находить показатели положительного β_i^+ и отрицательного β_i^- вкладов и элементов, $i = 1, 2, \dots, H$. Дело в том, что показатель значимости ξ_i , по определению, не зависит от текущего значения собственного параметра p_i данного элемента и характеризует влияние на систему только теоретического, максимального, предельно возможного изменения этого параметра от 0 до 1. Однако, реальные воз-

возможности изменения собственного параметра элемента могут быть только от текущего значения p_i до 1, и от текущего значения p_i до 0. Поэтому, характеристики вкладов β_i^+ и β_i^- должны определять, на сколько изменится системный показатель P_c при указанных изменениях параметра p_i элемента i исследуемой системы. Основные расчетные формулы определения вкладов элементов следующие:

$$\beta_i^+ = \frac{P_c}{p_i} - P_c \quad ; \quad (1.2.21)$$

$$\beta_i^- = -(P_c - \frac{P_c}{p_i}) \quad ; \quad i=1,2,\dots,H \quad (1.2.22)$$

В отличие от показателя положительного вклада (1.2.21) при расчете показателя отрицательного вклада (1.2.22) принудительно изменяется знак. Это сделано для того, чтобы во всех показателях роли элементов положительные значения характеристик всегда означали увеличение P_c при соответствующих изменениях p_i от 0 до 1 – для ξ_i , от p_i до 1 – для β_i^+ и от p_i до 0 – для β_i^- , и наоборот.

Для анализа системы в целях повышения ее надежности или безопасности наиболее информативной представляется характеристика положительного вклада β_i^+ элемента, вычисленная на основе прямой модели системы в целом (безотказности, безопасности). Она представляет те, реальные возможности по изменению параметров элементов системы, которые могут оказать наиболее существенное практическое влияние на увеличение надежности или безопасности исследуемой системы в целом. Например, если p_i близка к 1, то даже при большой значимости этого элемента его реальный вклад в увеличение надежности системы может оказаться крайне незначительным, что и покажет β_i^+ . Для анализа возможностей наиболее эффективного уменьшения надежности или безопасности системы лучше использовать показатель отрицательного вклада элементов.

1.3. ОСНОВНЫЕ ПОЛОЖЕНИЯ И МЕТОДИЧЕСКИЕ ОСНОВЫ РЕШЕНИЯ ЗАДАЧ ОЦЕНКИ НАДЕЖНОСТИ, БЕЗОПАСНОСТИ И РИСКА СИСТЕМ НА ОСНОВЕ ТЕХНОЛОГИИ АВТОМАТИЗИРОВАННОГО МОДЕЛИРОВАНИЯ РЕАЛИЗОВАННОЙ В ПРОГРАММНОМ КОМПЛЕКСЕ «RISK SPECTRUM».

1.3.1. Краткая характеристика ПК Risk Spectrum.

В основе методологии моделирования и расчета показателей надежности и безопасности с помощью ПК Risk Spectrum лежит аппарат логико-вероятностных методов, использующих в качестве средства построения графических моделей безопасности (надежности) дерева событий (ДС) и дерева отказов (ДО), см. рис. 3.1.

Под деревом событий понимается **графическая** модель, описывающая **логику** развития различных вариантов аварийного процесса, вызываемого рассматриваемым исходным (инициирующим) событием аварии (ИС).

В ПК Risk Spectrum ДС представляется в виде таблицы, содержащей строку заголовков, поле, в котором помещен разомкнутый бинарный граф (собственно дерево событий), несколько столбцов с характеристиками конечных состояний моделируемого объекта, реализующихся в процессе осуществления аварийных последовательностей. В заголовке 1-го столбца таблицы указывается обозначение ИС (группы ИС). В последующих заголовках столбцов слева направо размещаются названия и условные обозначения промежуточных событий, соответствующих успешному или неуспешному выполнению функций безопасности, работоспособным или отказовым состояниям систем безопасности или отдельных компонентов (оборудования и технических средств), правильным или ошибочным действиям персонала. В столбцах, характеризующих конечные состояния, указываются номера КС, их условные обозначения, типы КС (например, КС с повреждением активной зоны – CD), вероятности реализации, возможно, логические формулы, соответствующие данным аварийным последовательностям (АП). Пример ДС представлен на рис. 3.2.

С помощью аварийных последовательностей на ДС отображаются варианты развития аварийного процесса. При этом под АП понимается последовательность событий, приводящих к некоторому конечному состоянию объекта, включающая исходное событие **аварии**, успешные или неуспешные срабатывания систем **безопасности** и действия личного состава (персонала) в процессе развития аварии.

Под деревом отказов понимается **графическая** модель, отображающая логику событий, приводящих к невыполнению заданной функции (отказу) системы вследствие возникновения различных комбинаций отказов оборудования и ошибок персонала. Пример ДО представлен на рис.3.3.

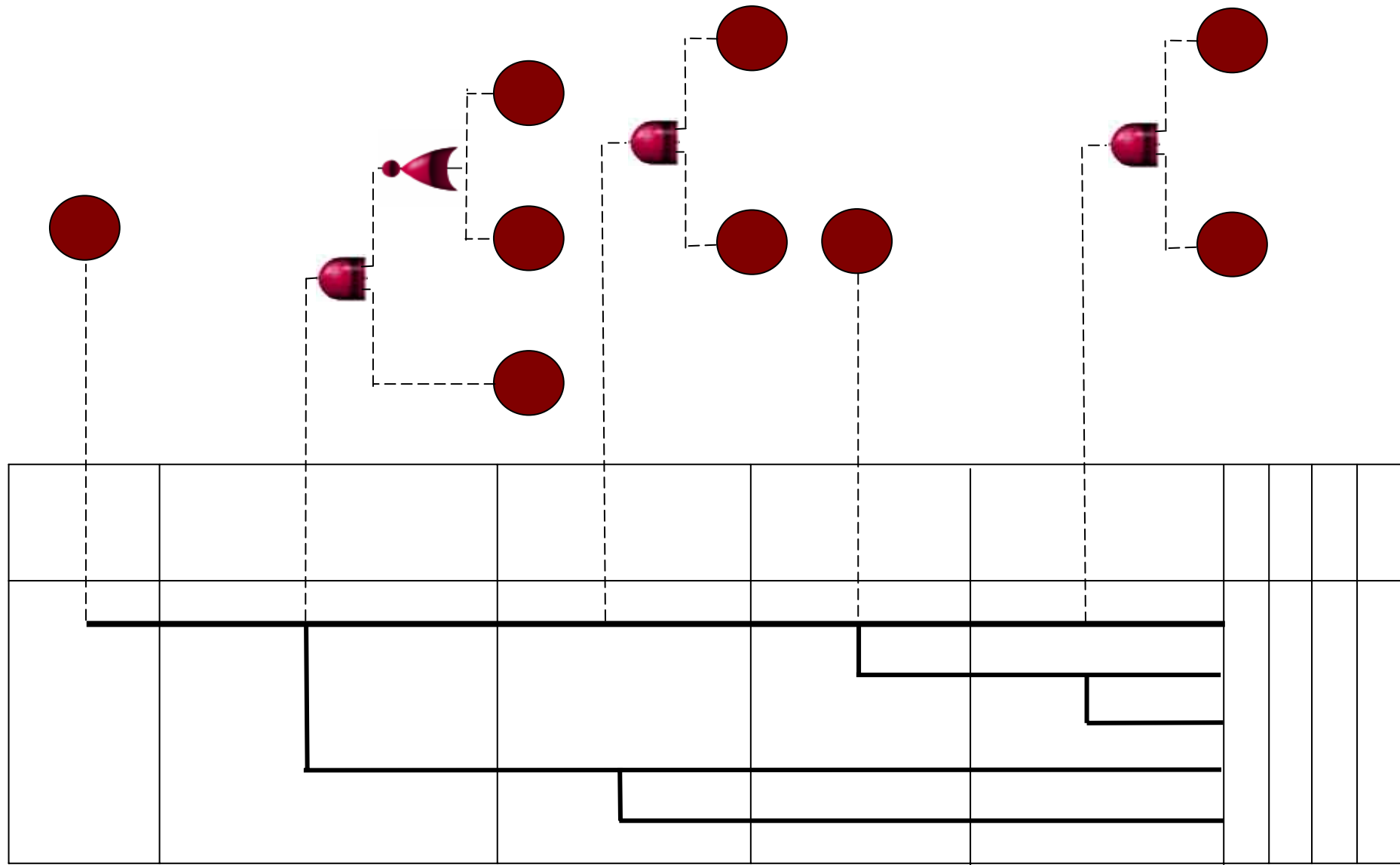


Рис. 3.1. Модель надежности (безопасности), представленная с помощью дерева событий и деревьев отказов

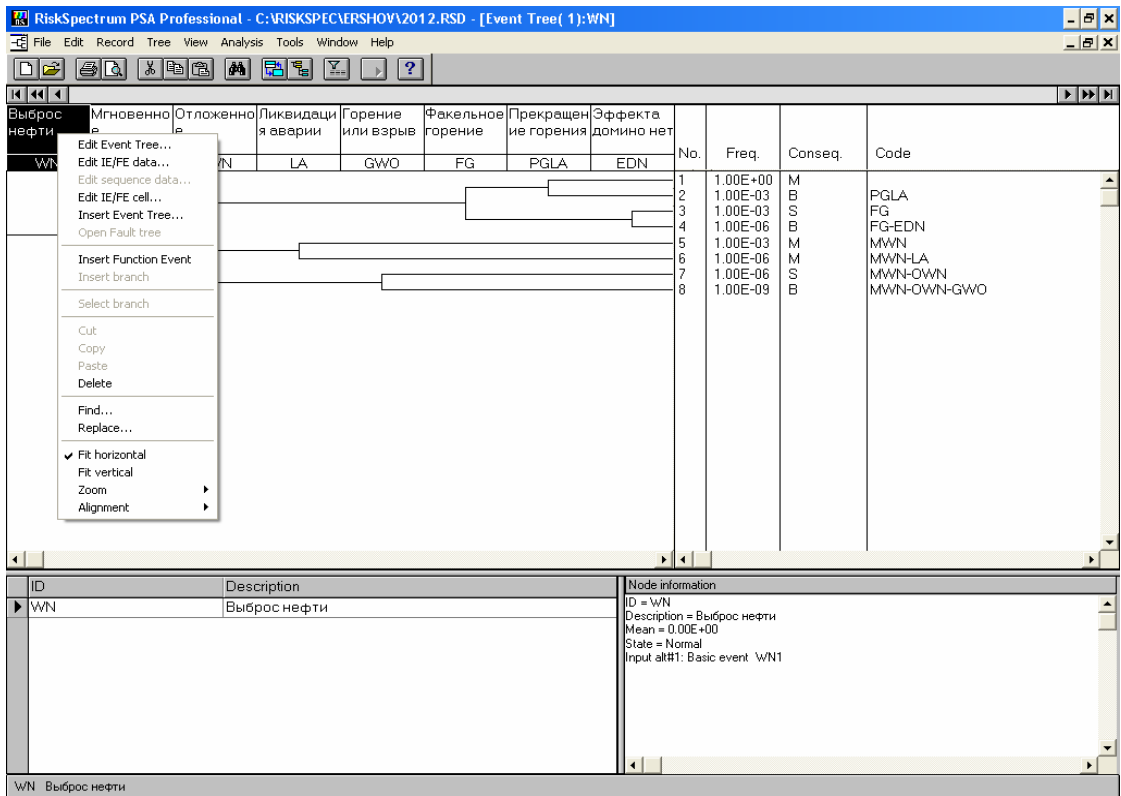


Рис. 3.2 Дерево событий в редакторе ДС

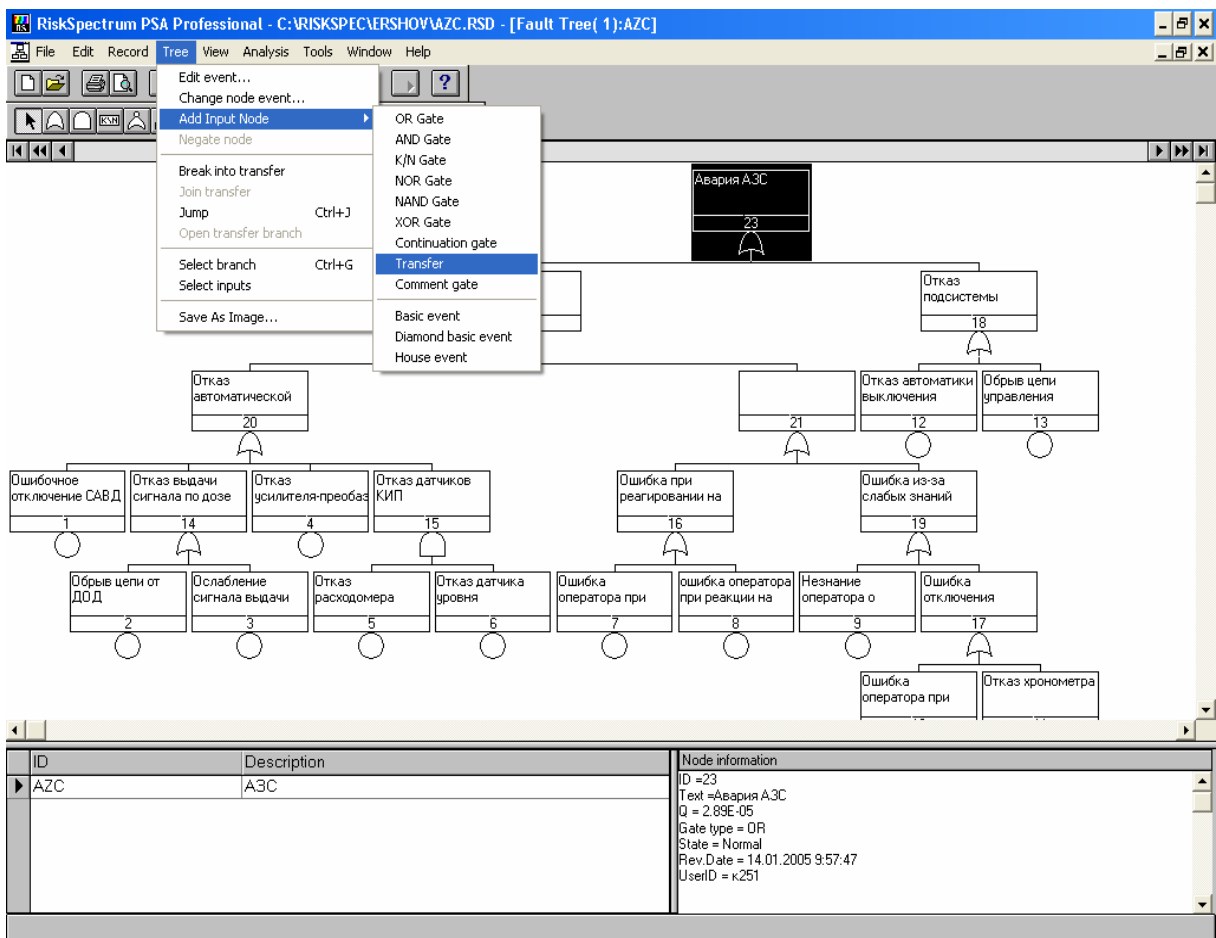



Рис. 3.3. Дерево отказов в редакторе ДО

В состав ДО входят графические элементы, служащие для отображения элементарных случайных событий (базисных событий) и для отображения логических операторов.

Как известно, логический оператор это математический (графический или буквенный) символ, служащий для отображения логической операции (конъюнкции, дизъюнкции, отрицания и т.д.), который произвольному набору случайных событий $\{A\}$, $\{B\}$, ..., $\{F\}$ ставит в соответствие сложное случайное событие $\{Z\}$. Т.о. в алгебре логики каждому логическому оператору соответствует булева функция некоторого числа логических переменных.

В графе, например, ДО, каждому логическому оператору булевой алгебры соответствует определенный графический элемент, что позволяет производить декомпозицию сложных событий на более простые (базисные или элементарные) события.

Пример: Запись $\{Z\} = \{A\} \cup \{B\}$ означает, что событие $\{Z\}$ реализуется при реализации любого из событий $\{A\}$ или $\{B\}$. Здесь обозначение \cup используется для отображения логической операции ИЛИ. В дереве отказов вместо \cup используется элемент вида: 

В таблице 3.1 представлены основные понятия и определения, используемые при описании теории ПК Risk Spectrum разработчиками комплекса.

Таблица 3.1.

| Элемент модели | Определение |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Дерево отказов | Древовидная структура. Начинается <i>верхним событием</i> , разбиваясь вниз по всем возможным причинам. |
| Верхнее событие | Нежелательное событие или состояние, для которого сформировано дерево отказов. <i>Верхнее событие</i> может также быть использовано, как более общий термин для всех типов нежелательных событий или конечных состояний в модели: <i>верхние события ДО</i> , <i>верхние события последовательности</i> и <i>верхние события последствия</i> . |
| Базисное событие | Событие «корневой причины» в дереве отказов, для которого не делается более детальная разработка логики дерева. Для каждого базисного события задаются <i>модель надежности</i> и набор <i>параметров надежности</i> . |
| Логический элемент (оператор) | Логические отношения между событиями в дереве отказов моделируются булевыми логическими операторами или <i>логическими элементами</i> . Каждый логический элемент имеет определенный тип оператора, например, ИЛИ, И, К/Н, НИ или И-НЕ. |
| Внешние условия | Внешние условия – специальный тип основных событий, которые могут иметь только два логических значения: ИСТИНА или ЛОЖЬ. |
| Параметр | Числовые значения, используемые в моделях надежности. В Risk Spectrum параметры – физически разделенные объекты данных с именем, точечной оценкой (среднее) значения и необязательным распределением неопределенности. |
| Дерево событий | Древовидная структура, начинающаяся с <i>исходного события</i> , очерчивающая все возможные сценарии, которые могут происходить в результате этого события. |
| Исходное событие | Событие или состояние, которое является «отправной точкой» в ДС, от которой начинаются все возможные сценарии. ИС - обычно нарушение или сбой. В Risk Spectrum ИС может быть <i>базисным событием</i> , <i>логическим элементом</i> дерева отказов или <i>последствием</i> , использованным в других деревьях событий. |

| Элемент модели | Определение |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Функциональное событие | События, которые воздействуют на результат сценариев, моделируемых в дереве событий. В Risk Spectrum функциональные события могут быть <i>логическими элементами</i> дерева отказов или <i>базисными событиями</i> . |
| Последовательность | Путь через ДС, сценарий. Последовательности в ДС описываются в терминах успеха или отказа <i>функциональных событий</i> , принадлежащих ДС. |
| Точка разветвления | Каждая точка в ДС, где последовательность разветвляется на две или больше последовательностей. Каждая точка разветвления соответствует <i>функциональному событию</i> . |
| Последствие | <i>Последствие</i> может быть назначено каждому конечному состоянию ДС. Для нескольких последовательностей может назначаться одно последствие. Это позволяет группировать последовательности, которые ведут к одному конечному состоянию, определяя, таким образом, уровень верхних событий в модели. |

К показателям надежности (безопасности), используемым в Risk Spectrum, относятся коэффициент неготовности, усредненный коэффициент неготовности, вероятность отказа, частота отказов.

Если значение коэффициента неготовности постоянно во времени, его значение равно величине усредненного коэффициента неготовности.

Если компонент или система являются невосстанавливаемыми, то коэффициент *неготовности* равен *вероятности отказа*.

Разработчики Risk Spectrum используют термины «неготовность» и «ненадежность» и для восстанавливаемых и для невосстанавливаемых компонентов. В научно-технической литературе обычно термин «неготовность» используется только для восстанавливаемых компонентов, в то время как «ненадежность» используется для невосстанавливаемых компонентов.

Также обстоит дело и с термином «частота» для восстанавливаемых и для невосстанавливаемых компонентов. В литературе, термин «частота отказов» используется только для невосстанавливаемых компонентов, в то время как «безусловная интенсивность отказов» используется для восстанавливаемых компонентов.

Термин «частота» используется в Risk Spectrum двумя немного разными способами:

1. Как тип параметра, который может быть использован в одной из моделей основного события. Эта модель названа по имени параметра, потому что это - единственный параметр в этой модели, т.о. мы имеем «модель основного события типа частота». В этом случае, «частота» относится к значению, которое является постоянным во времени. Фактически, это - параметр потока отказов Пуассона.

2. Исходное событие в ДС очень часто представляется процессом Пуассона. Традиционно при описании вероятностных параметров исходных событий используется термин

«частоты». Наиболее распространенная единица времени, используемая для этого типа частоты - «число событий в год».

Параметры «частоты» и тип «частота» основных событий (тип 5) предназначены для использования только для этого типа события.

Ожидаемое число возникновений отказов в единицу времени рассчитывается для верхнего события в анализе зависимости от времени. Оно также называется «безусловная интенсивность отказа».

1.3.2. Способы расчета вероятностных показателей надежности базисных событий.

В Risk Spectrum для базисных событий рассчитываются следующие характеристики надежности:

- Коэффициент неготовности в функции времени t (нестационарный коэффициент неготовности), $Q(t)$;
- Вероятность отказа или длительная установившаяся средняя неготовность (стационарный коэффициент неготовности), Q ;
- Безусловная интенсивность отказов во времени t , $W(t)$

В соответствии с этим каждому типу базисных событий задается определенная расчетная модель надежности. Каждая модель надежности имеет один или большее количество *параметров*. Система обозначений, используемая в формулах для расчета показателей надежности базисных событий в Risk Spectrum приведена в таблице 3.2.

Таблица 3.2.

| Параметр | Описание | Связь с параметром входа RS |
|-----------|------------------------------------------------|-----------------------------|
| q | Вероятность отказа на требование | q |
| λ | Интенсивность отказов | r |
| f | Частота | f |
| μ | Интенсивность восстановления | $1/TR$ |
| TR | Среднее время восстановление (MTTR) | TR |
| TI | Интервал между проверками | TI |
| TF | Время до первой проверки | TF |
| TM | Рассматриваемый промежуток времени (наработка) | TM |

Все модели имеют один или большее количество *обязательных параметров*. Некоторые из моделей надежности в Risk Spectrum гибки и охватывают немного разные ситуации моделирования. Эти модели содержат один или более *необязательных* параметров в дополнение к требуемым.

3.2. Типы и модели надежности базисных событий

Различают следующие типы и модели надежности базисных событий:

1.3.2.1. Модель непрерывно контролируемого восстанавливаемого компонента (тип 1)

Это - «нормальный» восстанавливаемый компонент. Используется экспоненциальное распределение и для процесса отказа и для процесса восстановления, то есть и интенсивность отказов и интенсивность восстановления постоянны.

Коэффициент неготовности $Q(t)$ для этого типа компонентов (оборудования) рассчитывается по следующей формуле:

$$Q(t) = q e^{-(\lambda+\mu)t} + \left(\frac{\lambda}{\lambda+\mu}\right) (1 - e^{-(\lambda+\mu)t}) \quad (1)$$

Обязательные параметры: λ, μ (г, TR)

Необязательные параметры: q

Второй член в формуле (1) - «традиционная» модель, используемая в большинстве программ анализа дерева отказов и во многих других типах исследований надежности. В начальный момент времени $K_{нг} = 0$, а затем быстро увеличивается до асимптотического установившегося значения.

Первый член необязательный и в большинстве случаев не используется. Если необязательный параметр q не определен (или задано $q = 0$), первый член исчезает. Однако, если $q > 0$, то в начальный момент $K_{нг} = q$, а затем уменьшается асимптотически до 0 со скоростью, зависящей от заданного значения интенсивности восстановления μ . Это может использоваться в моделях для компонентов, которые имеют начальную вероятность отказа (q) на требование на момент времени $t = 0$, а затем постоянную интенсивность отказов λ . Отказавшие элементы обоих типов могут быть восстановлены с постоянной интенсивностью восстановления μ .

Стационарный $K_{нг}$ для типа 1 модели надежности вычисляется по выражению:

$$Q = \frac{\lambda}{\lambda + \mu} \quad (2)$$

Безусловная интенсивность отказа $W(t)$ для типа 1 компонента рассчитывается по формуле:

$$W(t) = \lambda(1 - Q(t)) \quad (3)$$

Нормальное поведение (с $q = 0$) функции $W(t)$ состоит в том, что ее значение начинается с λ и затем медленно уменьшается до асимптотического значения. Если значение $K_{нг}$ невелико, то $W(t)$ приблизительно равна интенсивности отказов.

1.3.2.2. Модель периодически проверяемого компонента (тип 2)

Это наиболее сложная из моделей надежности в Risk Spectrum. В ее самой простой форме, при задании только интенсивности отказов и интервала между проверками, она соответствует «традиционной» модели для периодически проверяемого компонента. Для нее задают следующий набор обязательных и необязательных параметров:

Обязательные параметры: λ, T_I (г, TI)

Необязательные параметры: q, TR, TF

Модель предполагает экспоненциальное распределение для вероятности отказа (постоянная интенсивность отказов), постоянный установленный интервал между проверками, постоянное установленное время восстановления (если TR используется).

Чтобы упростить понимание этой модели, мы будем сначала представлять ее только с обязательными параметрами, то есть интенсивностью отказов и интервалом между проверками. КНГ в этом случае рассчитывается по формуле:

$$Q(t) = 1 - e^{-\lambda(t-T_i)}, \quad T_i = 0, TI, 2TI, \dots \quad (4)$$

Эта модель приводит к классической пилообразной кривой для функции неготовности.

Если задается параметр TF (время до первой проверки), то модель идентична, за исключением того, что моменты времени испытаний «смещены» на величину TF , то есть моменты времени испытаний - $T_i = 0, TF, TF + TI, TF + 2TI, \dots$

Среднюю неготовность Q_{mean} получают, интегрируя функцию $Q(t)$ по полному циклу испытания:

$$Q = \frac{1}{TI} \int_0^{TI} Q(t) dt = 1 - \frac{1}{\lambda TI} (1 - e^{-\lambda TI}) \quad (5)$$

Модель, описанная выше, предполагает, что время восстановления невелико (строго говоря $TR = 0$). Если это условие не имеет силы, то можно ввести в модель необязательный параметр TR . При этом учитывается вклад в значение коэффициента неготовности от времени ремонта.

В модели приняты следующие допущения:

Восстановление производится непосредственно после испытания, если при испытаниях был выявлен отказ компонента.

Продолжительность восстановления принято равным установленному времени TR .

Коэффициент неготовности $Q(t)$ для 2-го типа компонентов при учете времени восстановления TR вычисляется с помощью следующих формул:

$$Q(t) = 1 - e^{-\lambda t} \quad \text{for } t < TF \quad (1)$$

$$Q(t) = Q(TI) = 1 - e^{-\lambda TI} \quad \text{for } t = TF + nTI \quad (2)$$

$$Q(t) = Q(TI) + (1 - Q(TI))(1 - e^{-\lambda(t-TI)}) \quad \text{for } TI < t < TI + TR$$

$$Q(t) = 1 - e^{-\lambda(t-TI)} \quad \text{for } TI + TR < t < 2TI \quad (3)$$

Средняя неготовность, когда восстановление принято во внимание, вычисляется по формуле:

$$Q = 1 - \frac{1}{\lambda TI} (1 - e^{-\lambda TI}) + (1 - e^{-\lambda TI}) x \frac{TR}{TI} \quad (4)$$

Также можно учитывать необязательный постоянный вклад неготовности q . Это, например, вероятность отказа на требование, не выявляемая испытаниями. В этом случае значение поставляющей q добавляется ко всем формулам неготовности:

$$Q(t) = q + 1 - e^{-\lambda t} \quad \text{for } t < TF \quad (5)$$

$$Q(t) = Q(TI) = q + 1 - e^{-\lambda TI} \quad \text{for } t = TF + nTI \quad (6)$$

$$Q(t) = Q(TI) + (1 - Q(TI))(q + 1 - e^{-\lambda(t-TI)}) \quad \text{for } TI < t < TI + TR \quad (12)$$

$$Q(t) = q + 1 - e^{-\lambda(t-TI)} \quad \text{for } TI + TR < t < 2TI \quad (7)$$

Средняя неготовность в этом случае составляет:

$$Q = q + 1 - \frac{1}{\lambda TI} (1 - e^{-\lambda TI}) + (q + 1 - e^{-\lambda TI}) x \frac{Tr}{TI} \quad (8)$$

Безусловная интенсивность отказа $W(t)$ во всех случаях следует из нормальной формулы:

$$W(t) = \lambda(1 - Q(t)) \quad (9)$$

1.3.2.3. Модель с постоянным значением величины неготовности (тип 3)

Это простая, но часто используемая модель, которая использует постоянное значение неготовности q , как единственный параметр. Это обычно используется в ситуациях, где необходим учет «вероятности отказа на требование». Как правило, она используется при моделировании компонентов / систем, которые находятся в действии и изменяют состояние при поступлении соответствующего сигнала. Примеры: Отказ арматуры при открытии/закрытии, отказ выключателей, отказ двигателя при пуске/останове.

Параметры этой модели не зависят от времени работы. Формулы для неготовности $Q(t)$, длительной средней неготовности Q_{mean} и безусловной интенсивности отказа $W(t)$ имеют следующий вид:

$$Q(t) = q \quad (10)$$

$$Q = q \quad (11)$$

$$W(t) = 0 \quad (12)$$

1.3.2.4. Модель с заданным рассматриваемым промежутком времени (тип 4)

Эта модель, которая имеет сходство с моделью элементов 3 типа. Однако величина стационарного значения неготовности не задается, а рассчитывается по значениям интенсивности отказов и установленной времени работы компонента. Необязательная постоянная составляющая неготовности q может также быть добавлена.

Количественные характеристики параметров надежности рассчитываются по формулам:

$$Q(t) = q + 1 - e^{-\lambda TM} \quad (= \text{constant}) \quad (13)$$

$$Q = q + 1 - e^{-\lambda TM} \quad (= \text{constant}) \quad (20)$$

$$W(t) = 0 \quad (14)$$

Обязательные параметры: λ (r), ТМ

Необязательные параметры: q

1.3.2.5. Модель постоянного значения частоты (тип 5)

Эта модель используется, когда событие описывается процессом Пуассона, то есть события происходят с постоянной частотой (скоростью). Расчетные выражения при этом имеют следующий вид:

$$Q(t) = 0 \quad (22)$$

$$Q = 0 \quad (15)$$

$$W(t) = f \quad (16)$$

Эта модель обычно используется только для исходных событий. Исходные события обычно используются в деревьях событий, но они могут также быть использованы как основные события в деревьях отказов. Если они используются в деревьях отказов, должны соблюдаться определенные правила при дальнейших вычислениях. Эти правила будут описаны ниже при описании типов анализа в разделе MCS-анализа.

1.3.2.6. Модель невозстанавливаемого компонента (тип 6)

Это «традиционная» невозстанавливаемая модель компонента с показательной моделью отказа, то есть постоянной интенсивностью отказов. Как во многих других моделях компонентов в Risk Spectrum в нее добавлена возможность использования необязательной постоянной составляющей вероятности отказа. Это может быть использовано в ситуациях, где имеется вероятность отказа на требование во время $t = 0$.

Эта модель требует задания рассматриваемого промежутка времени Т, который является моментом времени, заданным в спецификации MCS-анализа. Для вычисления показателей надежности компонентов данного типа используются выражения.

$$Q(t) = q + 1 - e^{-\lambda t} \quad (17)$$

$$W(t) = \lambda(1 - Q(t)) \quad (18)$$

Обязательные параметры: λ (r)

Необязательные параметры: q

Для этой модели не имеет смысла определять длительную установившуюся среднюю неготовность Q, потому что неготовность увеличивается асимптотически до 1.

Окно выбора модели надежности для базисных элементов представлено на рис.3.4.

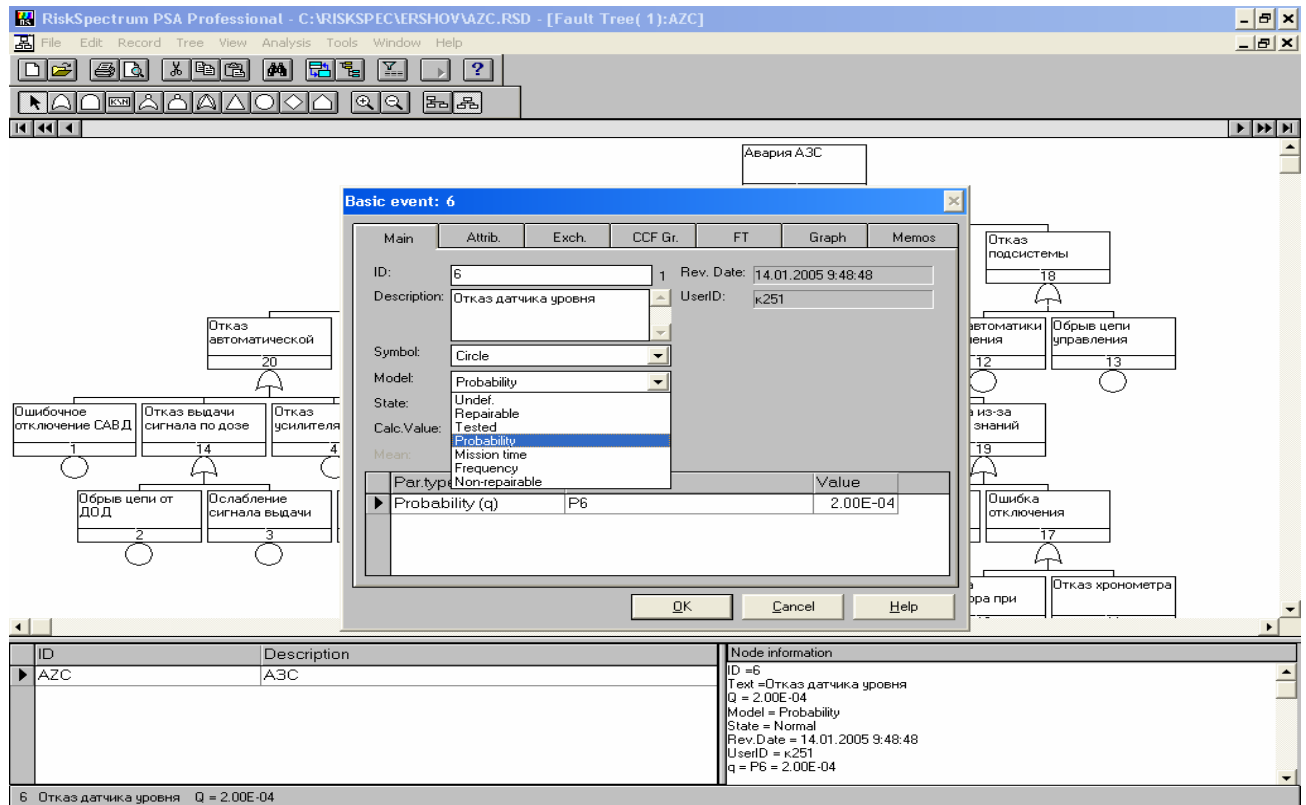


Рис. 3.4. Окно выбора модели надежности базисных элементов

1.3.3. Параметры моделей надежности

Параметр модели надежности в Risk Spectrum - в основном числовое значение, не обязательно включающее распределение неопределенности. Имя параметра может быть расценено как переменная, которая будет использоваться в математических формулах для вычислений надежности. Когда формула рассчитывается, имя переменной заменяется числовым значением, сохраненным в записи параметра.

Значение параметра определяется точечной оценкой (среднее) значения и, не обязательно, распределением неопределенности.

1.3.3.1. Типы параметров

Имеется 8 различных типов параметров в моделях надежности Risk Spectrum. Они перечислены в таблице 3.3. Там же указаны числовые ограничения для каждого параметра.

Таблица 3.3.

| Тип параметра | Обозначение | Пределы |
|------------------------------------|-------------|--------------------|
| Вероятность | q | $0 < q < 1$ |
| Интенсивность отказов, частота | r, f | $0 < r < 1.0E+32$ |
| Время восстановления (MTTR) | TR | $0 < TR < 1.0E+32$ |
| Интервал между проверками | TI | $0 < TI < 1.0E+32$ |
| Время до первой проверки | TF | $0 < TF < 1.0E+32$ |
| Рассматриваемый промежуток времени | TM | $0 < TM < 1.0E+32$ |
| CCF | * | $0 < * < 1$ |

* Параметры CCF могут быть различны, в зависимости от модели CCF. См. раздел, описывающий модели CCF.

1.3.3.2. Распределения неопределенностей для параметров моделей надежности

В таблице 3.4 перечислены используемые типы распределений неопределенности, а также параметры соответствующих распределений, которые должны быть определены для каждого параметра моделей надежности.

Таблица 3.4.

| Тип распределения | Параметры распределения |
|-------------------|-----------------------------------------|
| Логнормальное | Среднее, фактор ошибки (EF) |
| Бета | Среднее, α (параметр формы) |
| Гамма | Среднее, α (параметр формы) |
| Нормальное | Среднее, среднеквадратичное отклонение |
| Однородное | Минимум, максимум |
| Логоднородное | Минимум, максимум |
| Дискретное | По крайней мере два значения процентиля |

Примечание:

Средние значения параметров используются для MCS-анализа, анализа значимости, исследований чувствительности и анализа зависимости от времени. Распределения неопределенности используются только для анализа неопределенности.

1.3.4. Деревья отказов

При построении деревьев отказов в редакторе ДО ПК Risk Spectrum используется семь типов логических операторов:

Таблица 3.5.

| Тип логического оператора | Значение: логический элемент есть ИСТИНА если: |
|-------------------------------------|---------------------------------------------------------------------------|
| ИЛИ | По крайней мере одно входное событие ИСТИНА |
| И | Все входные события ИСТИНА |
| К-из-N (K/N) | По крайней мере K из N входных событий ИСТИНА * |
| НИ (ИЛИ-НЕ) | Ни одно из входных событий не есть ИСТИНА (все входные события ЛОЖЬ) |
| И-НЕ | Не все входные события ИСТИНА (По крайней мере одно входное событие ЛОЖЬ) |
| Неэквивалентность (исключающее ИЛИ) | Точно одно входное событие - ИСТИНА |
| Комментарий (не логика) | Единственное входное событие - ИСТИНА |

* Максимальное число входов и ограничения значения K зависят от возможного объема памяти и полного размера дерева отказов. K/N-логические элементы расширяются до эквивалентного числа ИЛИ - / И - логических элементов.

Обозначения логических операторов представлены в нижнем ряду панели инструментов на рис. 3.3.

НЕ - оператор

Помимо вышеперечисленных логических элементов, имеется еще один, дополнительный, логический оператор, который может быть использован в деревьях отказов: *НЕ - оператор*. Его можно считать логическим «инвертором». *НЕ - оператор* может быть исполь-

зован в каждом узле в дереве отказов, которое содержит основное событие или внешнее условие. Символ НЕ - *оператора* в Risk Spectrum - маленький круг, только выше узла.

НЕ – *оператор* имеет следующее влияние на логику, см. таблицу 3.6:

Таблица 3.6.

| Событие в узле | Узел (вход логического элемента) |
|----------------|----------------------------------|
| ИСТИНА | ЛОЖЬ |
| ЛОЖЬ | ИСТИНА |

В текущей версии Risk Spectrum отрицательная логика обрабатывается, как «псевдо-отрицательная логика». Это означает, что дополненные (отрицаемые) события обрабатываются следующим способом в МСО -исследованиях:

Любой набор сечений, содержащий взаимно исключающие события, удаляется. Такой набор сечений не может происходить, потому что взаимно исключающие события не могут происходить одновременно.

Дополненные события не включаются в наборы минимальных сечений, произведенные в МСО -анализе, то есть МСО содержат только события отказа и не любые события успеха.

«Псевдоотрицательная логика» обычно нужна из-за необходимости представить отрицательную логику в деревьях отказов: обработать взаимно исключающих событий таким способом, что комбинации (наборы сечений), содержащие взаимно исключающие события были невозможны.

Внешние условия

Внешнее условие может быть включено в логику дерева отказов, подобно базисному событию, но оно действует как «логический переключатель» который может иметь только одно из булевых значений ИСТИНА или ЛОЖЬ (ВКЛ/ВЫКЛ).

Внешние условия используют, чтобы изменить структуру дерева отказов. При этом могут быть получены различные вариации исходного дерева отказов. «Параметры настройки» внешнего условия (ИСТИНА / ЛОЖЬ) могут быть определены двумя способами:

Установить внешнее условие ИСТИНА или ЛОЖЬ в базе данных («жесткая» установка). Это означает, что это - всегда ИСТИНА или ЛОЖЬ во всех исследованиях, которые включают это внешнее условие.

Включить его в набор граничных условий (набор ГУ), где список внешних условий может быть определен, наряду с установкой ИСТИНЫ / ЛЖИ. Набор ГУ может тогда быть использован в случаях анализа, исходных событиях или функциональных событиях.

Внешние условия воздействуют на логику различных типов логических элементов так, как описано в таблице 3.7

Логические элементы, которые примут значения ИСТИНА или ЛОЖЬ из-за введения внешних условий, будут самостоятельно действовать как внешние условия и действие внешних условий может, таким образом, распространяться по структуре дерева отказов.

Во всех случаях внешние условия (и логические элементы, которые примут значения ИСТИНА или ЛОЖЬ) удаляются из структуры дерева после того, как их влияние на логику будет определено.

Таблица 3.7.

| | |
|------------------------------------|----------------------------------------------------------------------------------|
| Логический элемент «ИЛИ» | 1 или большее количество входов ИСТИНА == > ИСТИНА на выходе |
| | Все входы ЛОЖЬ == > ЛОЖЬ на выходе |
| | 1 или большее количество (но меньше чем все) входов ЛОЖЬ == > Никакого влияния |
| Логический элемент «И» | 1 или большее количество входов ЛОЖЬ == > ЛОЖЬ на выходе |
| | Все входы ИСТИНА == > ИСТИНА на выходе |
| | 1 или большее количество (но меньше чем все) входов ИСТИНА == > Никакого влияния |
| Логический элемент К/Ν | К или большее количество входов ИСТИНА == > ИСТИНА на выходе |
| | (N-K+1) или большее количество входов ЛОЖЬ == > ЛОЖЬ на выходе |
| | М. (где М. < К) входы ИСТИНА == > Логический элемент становится (К-1)/N |
| | М. (где М. < N-K+1) входы ЛОЖЬ == > Логический элемент становится К / (N-1) |
| Логический элемент НИ | 1 или большее количество входов ИСТИНА == > ЛОЖЬ на выходе |
| | Все входы ЛОЖЬ == > ИСТИНА НА ВЫХОДЕ |
| | 1 или большее количество (но меньше чем все) входов ЛОЖЬ == > Никакого влияния |
| Логический элемент И-НЕ | 1 или большее количество входов ЛОЖЬ == > ИСТИНА на выходе |
| | Все входы ИСТИНА == > ЛОЖЬ на выходе |
| | 1 или большее количество (но меньше чем все) входов ИСТИНА == > Никакого влияния |
| Логический элемент исключяющее ИЛИ | Точно 1 входная ИСТИНА == > ИСТИНА на выходе |
| | Больше чем 1 входы ИСТИНА == > ЛОЖЬ на выходе |
| | Все входы ЛОЖЬ == > ЛОЖЬ на выходе |
| | 1 или большее количество (но меньше чем все) входы ЛОЖЬ == > Никакого влияния |

Внешние условия могут также использоваться внутренне в логических элементах и основных событиях вместе с так называемыми заменами событий, как это описано ниже.

Замены событий

В Risk Spectrum Вы можете определить так называемые замены событий. Условная замена события определяется внешним условием (граничное условие, которое сообщает программе, когда замена событий должна быть использована) и основным событием (замена событий). Функция следующая:

В течение анализа дерева отказов, находят логический элемент или основное событие с внешним условием ИСТИНА в списке замен событий. Первоначальный логический элемент или основное событие заменяют тогда основным событием, указанным как замена событий. Это - очень удобный способ условно заменить одно событие другим.

Если имеется больше, чем одно граничное условие ИСТИНА в таблице замены событий для основного события или логического элемента, программа будет использовать **последнее** граничное условие ИСТИНА для замены.

1.3.5. Моделирование отказов по общей причине

1.3.5.1. Логика модели ООП

Особенность моделирования ООП, обеспеченного группами ООП - фактически создание такого же типа структуры дерева отказов, который обычно формируется вручную.

Функциональные возможности групп ООП поясним на примере. Предположим, что имеется четыре резервированных компонента - А, В, С и D, для которых мы моделируем ООП. Первоначально индивидуальные отказы этих четырех компонентов представлены четырьмя основными событиями в дереве отказов. Когда эти основные события (также называемые А, В, С, D, для простоты) существуют, Вы можете создавать запись группы ООП, где Вы определяете, что эти четыре основных события должны принадлежать определенной группе.

После обозначения группы ООП определяется модель ООП (бета-фактора, множественных греческих букв или альфа-фактора) и параметры выбранной модели ООП.

После окончания ввода данных по ООП программа автоматически создает следующие *события ООП*:

AB, AC, AD, BC, BD, CD, ABC, ABD, ACD, BCD, ABCD

(здесь, например, АВ означает ООП обоих компонентов - А и В)

На следующем шаге каждое из базисных событий, которые имелись в дереве отказов и подвержены ООП заменяется деревом с верхним событием «ИЛИ» (называемой логическим элементом ООП) куда входят индивидуальное событие отказа компонента и события ООП, включающие это базисное событие (отказ компонента). Например, основное событие А заменяется схемой «ИЛИ» с элементами А (индивидуальный отказ), АВ, АС, АД, АВС, ABD, ACD и ABCD, как входами.

1.3.5.2. Расчет вероятностей ООП

Квантификация различных событий ООП делается на основе модель надежности первичного базисного события в группе ООП, выбранной модели ООП и соответствующих параметров моделей.

Параметры моделей ООП, которые могут быть использованы в Risk Spectrum указаны в таблице 3.8

Таблица 3.8.

| Модель ССФ | Параметры ССФ |
|------------------------------------|---------------------|
| Бета-фактора | Бета, β |
| Множественных греческих букв (МГБ) | Бета, β |
| | Гамма, γ |
| | Дельта, δ |
| Альфа-фактора | Альфа 2, α_2 |
| | Альфа 3, α_3 |
| | Альфа 4, α_4 |

Комбинации событий ООП могут иметь порядок 2, 3 и N, где N - общее количество базисных событий, которые принадлежат группе. Если больше чем три компонента отказывают из-за ООП, это моделируется так, как будто **все** компоненты в группе отказывают. Другими словами, при моделировании ООП не делают различия между ООП порядка 4 или выше, даже если это теоретически возможно в моделях МГБ и альфа-фактора. В модели бета-фактора, по определению, различают только индивидуальные отказы и ООП, т.е. если происходит отказ по общей причине, то считается, что все компоненты отказывают одновременно.

Программа вычисляет неготовность для каждого события ООП, основываясь на модели надежности, основных параметрах надежности, модели ООП и параметрах ООП, которые определены в ООП-группе.

Эти вычисления делаются в соответствии с нижеприведенными формулам, в которых используется следующая система обозначений:

Таблица 3.9.

| | |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Q_{tot} | «Полная» неготовность для каждого основного события в группе ССФ. Эта неготовность рассчитывается, используя модель надежности и параметры, указанные в группе ССФ. |
| Q_k | Неготовность для события ССФ порядка k, то есть ССФ, включающего k компонентов (основные события). |
| N | Число основных событий в группе ССФ. |

Термин «неготовность» здесь используется в общем смысле. В зависимости от выбранного типа вычислений, она является или неготовностью в определенный момент времени ($Q_{tot}(t)$ и $Q_k(t)$), или длительной средней неготовностью ($Q_{tot, mean}$ и $Q_{k, mean}$)

В большинстве случаев, группы ООП не включают более четырех компонентов.

1.3.5.2.1 Модель бета-фактора

Общие формулы:

$$Q_k = (1 - \beta)Q_{tot} \quad k = 1 \quad (19)$$

$$Q_k = 0 \quad 1 < k < N \quad (20)$$

$$Q_k = \beta Q_{tot} \quad k = N \quad (29)$$

Формулы для ООП с четырьмя компонентами:

$$Q_1 = (1 - \beta) Q_{tot} \quad (30)$$

$$Q_2 = 0 \quad (31)$$

$$Q_3 = 0 \quad (32)$$

$$Q_4 = \beta Q_{tot} \quad (33)$$

1.3.5.2.2 Модель множественных греческих букв (МГБ)

Общие формулы:

$$Q_k = \frac{1}{\binom{m-1}{k-1}} \left(\prod_{i=1}^k \rho_i \right) (1 - \rho_{k+1}) Q_{tot}$$

$$\rho_1 = 1, \rho_2 = \beta, \rho_3 = \gamma, \dots, \rho_{N+1} = 0 \quad (34)$$

Формулы для ООП с четырьмя компонентами:

$$Q_1 = (1 - \beta) Q_{tot} \quad (35)$$

$$Q_2 = \frac{1}{3} \beta (1 - \gamma) Q_{tot} \quad (36)$$

$$Q_3 = \frac{1}{3} \beta \gamma (1 - \delta) Q_{tot} \quad (37)$$

$$Q_4 = \beta \gamma \delta Q_{tot} \quad (21)$$

1.3.5.2.3 Модель альфа-фактора

Общие формулы:

$$Q_k = \frac{k}{\binom{m-1}{k-1}} \frac{\alpha_k}{\alpha_{tot}} Q_{tot}$$

$$\alpha_{tot} = \sum_{k=1}^N k \alpha_k \quad (22)$$

Формулы для ООП с четырьмя компонентами:

$$Q_1 = \frac{\alpha_1}{\alpha_{tot}} Q_{tot} \quad (40)$$

$$Q_2 = \frac{2}{3} \frac{\alpha_2}{\alpha_{tot}} Q_{tot} \quad (41)$$

$$Q_3 = \frac{\alpha_3}{\alpha_{tot}} Q_{tot} \quad (42)$$

$$Q_4 = 4 \frac{\alpha_4}{\alpha_{tot}} Q_{tot} \quad (23)$$

Вычисление безусловной интенсивности отказа для событий ООП выполняется согласно тем же самым формулам, перечисленным выше для вычисления неготовности, но Q_{tot} заменяется W_{tot} , где W_{tot} - «полная» частота, рассчитываемая при использовании модели надежности и параметров, указанных в группе ООП. Это вычисление частоты соответствует формулам используемым для расчета показателей надежности основных событий.

1.3.5.3. Квантификация ООП для шахматного порядка испытаний

При вычислении неготовности для событий с ООП, у которых модель надежности «периодически проверяемая», и когда задан тип вычисления «Зависимость от времени» (то есть неготовность рассчитывается на определенный момент времени), параметры времени до первой проверки обрабатываются специальным способом, чтобы принять во внимание влияние шахматного порядка испытания компонентов в группе ООП:

- Для каждого события ООП, основные события, включенные в ту специфическую комбинацию проверяются на параметры TF (время до первой проверки). Программа решает, что **самое короткое** время прошло, если какой-либо из этих компонентов был проверен.
- Время, согласно вышеупомянутому используется, как «время после последнего испытания» (T_L) при вычислении $Q_{tot}(t)$ (см. главу «Модели основного события» для описания периодически проверяемой модели надежности основного события).

Влияние шахматного порядка испытаний (различные значения TF) будет принято во внимание логическим и непротиворечивым способом. Для каждого события ООП, **каждое** испытание **любого** из компонентов, включенных в событие ООП, рассматривается как испытание этого частного ООП.

1.3.6. Деревья событий

Как следует из рисунка 3.1 к каждому заголовку дерева событий «подключено» либо базисное событие, либо дерево отказов или, другими словами, каждый заголовок ДС имеет свой «вход».

1.3.6.1. Исходные события

Входами для исходного события в ДС Risk Spectrum могут служить:

- Базисное событие
- Логический оператор (гейт) дерева отказов
- Последствие (из другого ДС).

Если входом для ИС служит последствие, то ИС фактически заменяется верхним логическим элементом «дерева отказов последствия» (см. ниже), который объединяет по схеме ИЛИ все последовательности с этим последствием во всех других деревьях событий.

Можно вводить несколько строк входов для каждого ИС. Каждая строка это альтернативное исходное событие, и имеет индекс «Alt#», автоматически нумеруемый 1, 2, 3 ...

Каждое альтернативное ИС определяется входным событием (основное событие, логический оператор или последствие) и необязательным набором граничных условий (набор

ГУ). ИС может быть использовано в нескольких ДС, но каждое ДС может использовать только одну альтернативу (значение по умолчанию - Alt# = 1).

Если набор ГУ определен для переменной исходного события, которая «задействована» в дереве событий, этот набор ГУ действует по всему дереву событий (он действует слева направо по дереву событий).

Параметры настройки ИСТИНА / ЛОЖЬ в наборе ГУ в исходном событии останутся в этом значении для всех деревьев отказов последовательности и деревьев отказов последствия для этого дерева событий. Однако, если имеется любое последствие в этом дереве событий, которое используется, как исходное событие в другом дереве событий, эти граничные условия не будут «переходить» следующим деревьям событий в цепочке.

1.3.6.2. Функциональные события

Входами для функциональных событий (ФС) в Risk Spectrum могут служить:

- Основное событие
- Логический оператор дерева отказов

Основные события и логические элементы включаются в «деревья отказов последовательности», как описано ниже.

Для каждого ФС можно вводить несколько строк входов. Каждая строка есть альтернативное ФС и имеет индекс «Alt#», автоматически нумеруемый 1, 2, 3 ...

Каждое альтернативное ФС определяется входным событием (основное событие или логический оператор) и необязательным набором граничных условий (набор ГУ). Каждая ветвь дерева событий может использовать одну альтернативу (значение по умолчанию Alt# = 1). (Переменные ветви могут быть отредактированы путем двойного нажатия на точку ветвления.)

Если набор ГУ определен для переменной ФС, которая «задействована» в ветви, этот набор ГУ действует по всей ветви слева направо по дереву событий.

В дополнение к исходному событию, также можно активизировать набор граничных условий в каждой точке ветвления (см. переменные функционального события).

Набор граничных условий, активизированный в ветви, будет активен в этой ветви и во всех последующих точках ветвления, пока концы последовательности в этом дереве не будут достигнуты.

1.3.6.3. Учет вероятностей успешной реализации функционального события при расчетах вероятностей реализации конечных состояний

Вероятность реализации какого-либо конечного состояния (КС) в дереве событий рассчитывается, в общем случае, путем перемножения вероятностей реализации или не реализации функциональных событий, входящих в соответствующую аварийную последовательность. Однако при этом возможны различные варианты учета успехов для ФС(например, вероятностей успешного срабатывания систем безопасности АЭС).

В ПК Risk Spectrum используются пять возможных вариантов «обработки успеха» для функциональных событий:

- Пропуск успеха в ДС
- Логический успех ДС (значение по умолчанию)
- Логическая и простая квантификация
- Де Морган в ДО
- Де Морган в ДО и ДС

При выборе пункта «Пропуск успеха в ДС» программа не включает логику успеха в последовательности, в которые входят успешные исходы каких-либо ФС.

При выборе пункта «Логический успех в ДС», программа включает логику успеха в последовательности, в которые входят успешные исходы каких-либо ФС, но вероятности реализации этих событий успеха не учитываются при расчетах. Точнее, вероятности успеха обрабатываются так, как будто они равны 1.

При выборе пункта «Логическая и простая квантификация», программа включает логику успеха в последовательности, в которые входят успешные исходы каких-либо ФС, и вероятности реализации этих событий успеха учитываются при расчетах.

При выборе пункта «Де Морган в ДО», программа применит правила Де Моргана ко всем инвертированным логическим элементам в деревьях отказов. Это означает, что вся отрицательная логика будет перемещена до уровня основных событий в деревьях отказов. Инвертированные основные события будут сохраняться в сечениях, и они будут определены количественно.

При выборе пункта «Де Морган в ДО и ДС », программа применит правила Де Моргана ко всем инвертированным логическим элементам в деревьях отказов и ко всем инвертированным функциональным событиям в последовательностях дерева событий. Это означает, что вся отрицательная логика будет перемещена до уровня основных событий в деревьях отказов, включая деревья отказов, подключенные к инвертированным функциональным событиям. Инвертированные основные события будут сохраняться в сечениях, и они будут определены количественно.

Окно выбора варианта учета успешных исходов функциональных событий представлено на рисунке 3.5.

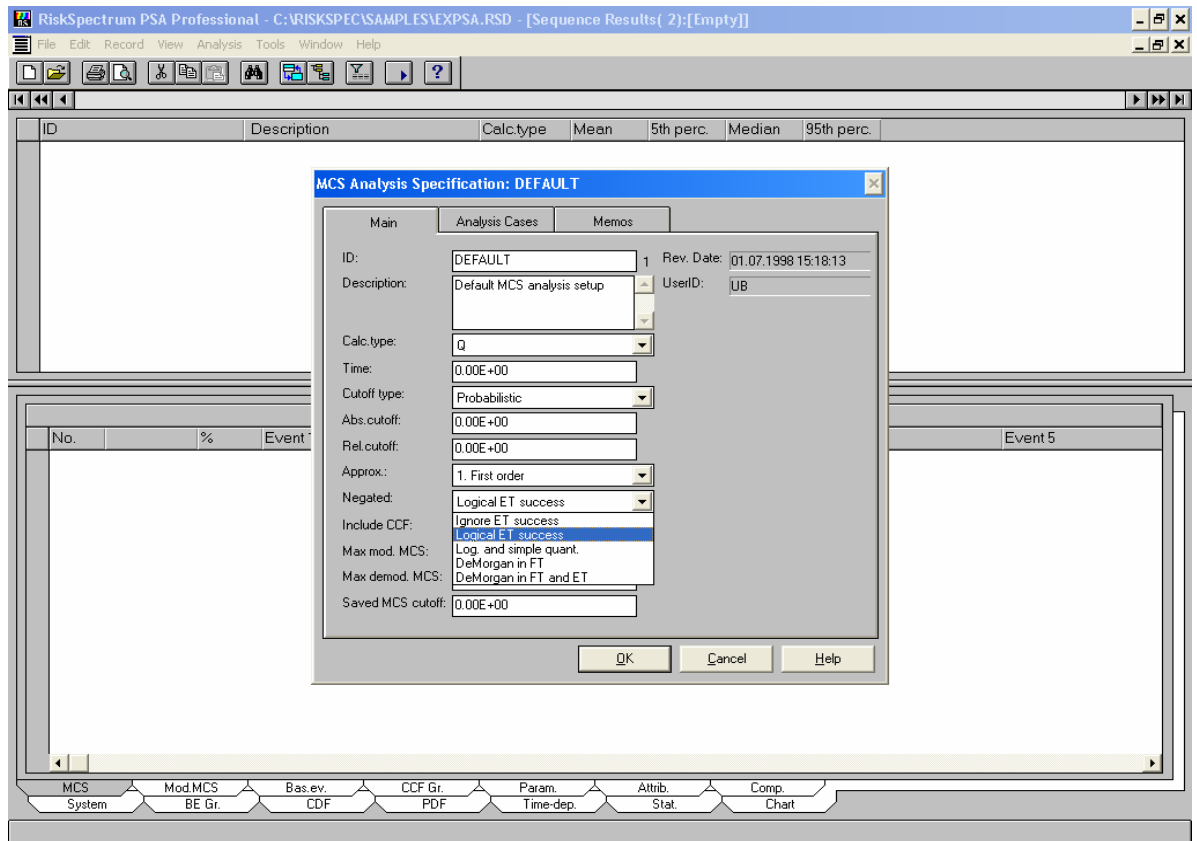


Рис. 3.5. Окно выбора варианта учета успешных исходов функциональных событий

1.3.7. Генерация и анализ минимальных сечений отказов

Цель генерации и анализа МСО состоит в получении, так называемых, наборов МСО деревьев отказов и выполнении расчетов точечной оценки вероятности реализации верхнего события ДО. ДО - графическое представление булевой функции и эта булева функция может быть преобразована, путем использования законов булевой алгебры, во множество МСО. МСО это комбинация базисных (элементарных) событий, реализация которых вызывает реализацию верхнего события. Термин «минимальное сечение» означает, что, если любое из базисных событий удалено из множества, верхнее событие будет не реализовано.

1.3.7.1. Краткая характеристика задачи генерации и анализа МСО

Анализ МСО состоит следующих главных этапов:

- Чтение информации о верхнем событии, подлежащем анализу. Если определена группа анализа, то чтение информации о списке верхних событий.
- Чтение технических требований для данного случая анализа.
- Чтение фундаментальных данных из полной базы данных проекта и сохранение ее во внутренней памяти. Эти данные включают средние значения параметров, модели основного события, параметры и определения логики логического оператора (тип логического элемента и исходные события).
- Формирование внутренней структуры данных, которая соответствует ДО, подлежащему анализу, начиная с указанного верхнего логического элемента. Для верхних событий ДО это

достаточно простая задача. Для верхних событий последовательностей и верхних событий последствий этот шаг включает формирование конструкции дерева отказов из логики дерева событий, что является значительно более сложной задачей. Важная часть этого шага состоит в изменении логики дерева отказов в соответствии с состояниями внешнего условия, указанными в определении верхнего события (верхние события ДО) или в ДС. Когда этот шаг закончен, результат - всегда ДО с одним верхним событием.

- Проверка дерева на циклы. Если в ДО имеются циклы, выполнение анализа будет прервано сообщением об ошибке, указывающим логические элементы, включенные в цикл.

- Вычисление «неготовности» для каждого базисного события в соответствии с заданным типом вычисления (среднее значение или функция времени) и моделью надежности.

- Проверка дерева для основных событий, которые включены в группы ООП. Эти основные события автоматически заменяются деревьями отказов, которые моделируют все возможные комбинации событий ООП. Вычисление неготовности для каждого события ООП в соответствии с заданной моделью надежности и моделью ООП, указанной в группах ООП.

- Реструктуризация и построение из модулей дерева отказов, чтобы получить более «оптимальную» конструкцию, т.к. это упрощает и ускоряет генерацию МСО на более поздней стадии. Наиболее важная часть этого шага - нахождение так называемых **модулей** в ДО. Модули - независимые ветви дерева, которые могут быть обработаны как основные события при генерации МСО.

- Генерация МСО для верхнего события ДО. При получении МСО задаются критерий останова и значение останова моделирования. Конечный продукт этого шага - список всех сохраненных МСО после того, как было применено отсечение. Эти МСО строятся из модулей, то есть если в дереве имеются модули, то эти модули не расшифровываются (не разворачиваются).

- Если анализу подвергается верхнее событие последовательности, то генерируются МСО для верхнего логического элемента успеха. Верхний логический элемент успеха - NOR-логический элемент, который является входным для верхнего логического элемента последовательности. Этот NOR-логический элемент имеет все функциональные события, которые являются успешными по последовательности, как входы. При этом в некоторых последовательностях не имеется успешных функциональных событий и, следовательно, не имеется верхнего логического элемента успеха. При генерации МСО для этого верхнего логического элемента успеха он обрабатывается по схеме «ИЛИ». Полученные МСО сохраняются как набор AND-модулей - входов в большой OR-модуль. Это называется «Успешный MCS-модуль».

- Квантификация (расчет вероятностных показателей надежности (безопасности) модулей) по «Успешным MCS-модулям». При квантификации успеха МСО результат – вероятность

того, что один или большее количество этих функциональных событий - сбой, Q . Нас интересует дополнение к Q (успех) = $1 - Q$.

- Вычисление вероятностного показателя верхнего события (неготовности или частоты) с учетом результатов вычисления вероятностных показателей модулей. При этом шаге могут быть использованы различные типы аппроксимаций. Если это - верхнее событие последовательности, это - частота последовательности вычисляется без учета успехов функциональных событий. В выводе программы этот результат называется Q (неудача).

- Вычисление конечного результата последовательности по выражению $Q_{TOP} = Q(\text{неудача}) * Q(\text{успех})$.

- Демодуляция (разворачивание) модулей в МСО. В результате выполнения этого шага генерируется список МСО, в котором все элементы сечений представлены базисными событиями. Это может приводить к существенному увеличению числа сечений по сравнению со списком МСО, построенным из модулей. Поэтому при этом шаге применяется отсечение, которое служит, чтобы ограничить число МСО до разумного количества.

- Сортировка МСО или по неготовности / частоте или по числу элементов, входящих в сечения (порядку сечений).

- Запись результат расчета вероятностного показателя верхнего события (неготовность / частота) и список МСО в файл результата.

1.3.7.2. Формирование структуры дерева

Построение ДО ведется последовательно сверху вниз – от верхнего события до базисных элементов. После окончания построения дерева производится анализ заданных внешних условий (Истина, Ложь, Не определено). На основании этого анализа структура ДО модернизируется, путем удаления ветвей первоначального ДО.

При выполнении анализа последовательностей или последствий, то есть тогда, когда используются деревья событий, программа автоматически преобразовывает логику дерева событий в эквивалентную логику дерева отказов. При этом программа принимает во внимание состояния успеха событий в последовательностях. Это делается путем использования отрицательной логики в структуре дерева, которую программа создает, основываясь на деревьях событий. Логика модели, используемая программой для исследований дерева событий, обрабатывается следующим способом:

Каждая последовательность преобразуется в эквивалентную конструкцию дерева отказов со схемой «И» наверху. Этот логический элемент называется «верхний логический элемент последовательности».

Входами к верхнему логическому элементу последовательности служат:

- Функциональные события, входящие в последовательность, с неуспешными исходами.

- NOR-логические элементы со всеми функциональными событиями, входящими в последовательность, с успешными исходами.

При этом события, для которых задано состояние «не значимо» в последовательности не включаются.

Для каждого дерева событий, включенного в анализ (всегда одно ДС в случае анализа последовательности и одно или более в случае анализа последствий), программа создает схему «ИЛИ» со всей сопутствующей последовательностью схемы «И» от этого дерева событий, как вход. Тогда схема «И» создается с этой схемой «ИЛИ», как входом, наряду с исходным событием. Верхняя схема «И», созданная таким образом, называется «верхний логический элемент дерева событий».

Для верхнего события последствия, все верхние логические элементы дерева событий есть входы в схему «ИЛИ».

Для деревьев событий, которые имеют исходное событие, определенное как последствие, программа создает конструкцию верхнего последствия таким же образом, как описано выше и затем использует верхний логический элемент этого дерева, как исходное событие.

Когда технические требования верхнего события и логика дерева событий преобразованы в дерево отказов, как описано выше, остальная часть МСО-анализа выполняется таким же образом, как и для деревьев отказов.

1.3.7.3. Вычисление неготовностей основного события

Неготовность рассчитывается и сохраняется для каждого основного события (для основных событий типа «частота» (тип 5) частота сохраняется вместо неготовности).

Вычисление неготовностей основного события делается в зависимости от заданного типа вычислений:

Тип вычисления: средняя неготовность

Неготовность для каждого основного события рассчитывается стационарного коэффициента неготовности. Эти средние неготовности затем используются как неготовности в дальнейших вычислениях для модулей, наборов сечений и верхнего события. Этот тип вычисления наиболее широко применяется в ВАБах атомных электростанций. Должно быть указано, однако, что он не совсем правилен, поскольку не дает правильной оценки средней неготовности верхнего события, когда в модель включены основные события с моделью 2 (периодически проверяемый). Например, средняя неготовность двух связанных по схеме И основных событий при этом выше, чем произведение средней неготовности для двух отдельных основных событий. Также влияние шахматного испытания (различное время до первой проверки - TF - для избыточных компонентов) не принято во внимание в этом типе вычисления средней неготовности. Ошибки, сделанные из-за этих упрощений, однако, обычно достаточно малы по сравнению с общей неопределенностью в ВАБ, результатами учета отказов по

общей причине и т.д. Если нужны точные результаты, используют специальный модуль анализа зависимости от времени, чтобы вычислить среднюю неготовность верхнего события.

Тип вычисления: зависимость от времени неготовность

Неготовность для каждого компонента рассчитывается в точке времени, указанной в технических требованиях анализа МСО. Неготовность верхнего события также рассчитывается в этом указанном моменте времени.

Тип вычисления: частота

При этом типе вычислений неготовности основного события рассчитываются, как при *вычислении средней неготовности*. Должны иметься основные события типа частоты (тип 5) в модели и если сечения с больше чем одним событием частоты сгенерированы, эти сечения удаляются.

1.3.7.4. Реструктурирование и модуляризация дерева отказов

В целях упрощения генерации наборов МСО выполняется несколько типов модификаций дерева отказов:

- объединение логических операторов идентичного типа.
- нахождение и создание модулей.
- разложение К/N-логических элементов.
- минимизация логических элементов на основе правил булевой алгебры.

Объединение логических операторов идентичного типа.

Если имеется схема «ИЛИ» или «И» (логический оператор G1), которая является входом другого логического оператора того же самого типа (логический оператор G2) и логический оператор G1 встречается только в одном месте дерева, то логический оператор G1 может быть удален, а его входы могут быть добавлены к входам «отца» логического оператора G1 – оператору G2.

Этот шаг уменьшает число логических элементов в дереве, делает поиск модулей (см. ниже) более эффективным и увеличивает скорость генерации набора сечений.

Поиск и создание модулей.

Модуль определяется, как поддерево (логический оператор с его полной подструктурой), которое удовлетворяет следующим правилам:

- Все составляющие его события являются или базисными событиями или другими модулями.
- Все составляющие его события встречаются в дереве отказов только по одному разу.

Из этого определения следует, например, что все дерево отказов может быть модулем, если все события в дереве отказов встречаются только однажды. При этом, не смотря на то что, хотя все входы в модуле должны встречаться только однажды, сам модуль может находиться во многих места дерева.

Модули находят, применяя несколько методов. Отдельные логические элементы в ДО могут быть непосредственно идентифицированы и помечены как модули. Такие модули легко находятся. Другие модули могут быть созданы путем изменения ДО и прибавления новых логических элементов (которые помечаются как модули).

Модули это специальный тип логического элемента, Типы логических операторов, которые могут быть модулями: схемы «ИЛИ», «И» и К/Н.

Разложение К/Н-логических элементов.

Каждый К/Н-логический элемент может быть разложен в эквивалентную комбинацию схемы «ИЛИ» со множеством схем «И» в качестве входов. Это разложение делается на этой стадии, потому что это упрощает дальнейшую обработку дерева.

Как указывалось выше, модули могут также иметь К/Н-тип, и такие К/Н-модули также разлагаются на ИЛИ - и И-модули таким же образом.

Минимизация логической функции на основе правил булевой алгебры

При минимизации логической функции применяются следующие правила:

Схемы «ИЛИ» проверяются на предмет наличия какого-либо входа по схеме «И» к логическому элементу, который может быть удален, применяя Булев закон поглощения:

$$A + A*B = A.$$

Это происходит, если любой из входов схемы «И» также является прямым входом со схемой «ИЛИ».

Схемы «И» проверяются на предмет наличия каких-либо входов по схеме «ИЛИ», которые могут быть удалены, потому что все ее входы приведут к не минимальным наборам сечений, когда объединяются с другими входами схемы «И». Это происходит, если любой из входов схемы «ИЛИ» также оказывается прямым входом схемы «И». Это выполняется последовательным приложением Булевых законов:

$$A*A = A, \quad \text{и} \quad A + A*B = A$$

1.3.7.5. Обработка модулей

Модули в дереве отказов находят, применяя следующие три шага:

- Поиск логических операторов, которые имеют в качестве входов только основные события или другие, уже найденный, модули. Если все эти входы встречаются в дереве отказов только однажды, то этот логический элемент является модулем. Поиск модулей делается снизу (от основания дерева) вверх.
- Поиск таких логических операторов среди схем «ИЛИ» и «И», которые имеют два или больше исходных событий, которые происходят только однажды и являются основными событиями или уже найденным модулем. Если такие события находятся ниже логического элемента G1, создается новый модуль М (логический элемент, помеченный как модуль) с этими событиями в качестве входов. Входы, которые включены под модулем М удаляются

из логического элемента G1 и заменяются модулем M. Тип (ИЛИ или И) модуля M - тот же, что и у логического элемента G1.

- Поиск среди схем ИЛИ - и «И» таких комбинации исходных событий, которые всегда происходят вместе под логическими элементами одного типа. Если такие комбинации событий найдены, они могут быть объединены, чтобы сформировать новые модули, подобные модулям, описанным выше в пункте 2.

1.3.7.6. Генерация множеств минимальных сечений

После реструктуризации и построения ДО из модулей и окончания расчетов неготовности для всех основных событий и модулей начинается генерация МСО. В основном генерация наборов сечений делается согласно нисходящему алгоритму, который начинается созданием набора сечений с верхним логическим элементом, как единственным. Этот верхний логический элемент затем заменяется его входами. Если логический элемент - схема «И», то все ее входы добавляются к набору сечений, увеличивая число элементов в наборе. Если логический элемент - схема «ИЛИ», то создаются новые наборы сечений, по одному для каждого входа к схеме «ИЛИ». Эти новые наборы сечений подобны первоначальному, но со схемой «ИЛИ», замененной одним из ее входов. Этот процесс замены выполняется до тех пор, пока все элементы набора сечений не станут или основными событиями или модулями.

Основные принципы этого алгоритма известны и понятны, были описаны во многих книгах и статьях по анализу ДО. В первоначальной форме этот алгоритм очень неэффективен для ДО большого размера. Причиной является то, что он генерирует **все** МСО ДО (в форме функций булевой алгебры). Дело в алгоритме, который увеличивает размер набора сечений для схем «И» и увеличивает число наборов сечений для схем «ИЛИ», как объяснено выше. Это ведет к большому количеству генерируемых наборов сечений.

Значительно эффективнее другой способ: число наборов сечений рассчитывается, начиная от основания дерева, для каждого логического элемента с прибавлением числа наборов сечений для всех входов для схем «ИЛИ» и умножением числа наборов сечений для всех входов для схем «И». Основные события (или модули) рассчитываются как 1. Число, полученное для верхнего логического элемента таким образом, обычно упоминается как число Булевых обозначенных наборов сечений (BICS) и быстро становится чрезвычайно высоким в больших и сложных деревьях отказов (типично в пределах от нескольких миллионов до $1E+500$ (!) в исследованиях безопасности АЭС).

Наборы сечений, сгенерированные этим алгоритмом - не конечный продукт, потому что они **не минимальны**. В дереве отказов, содержащем повторные события, этот алгоритм обычно производит много не минимальных наборов сечений. Процесс устранения этих не минимальных наборов сечений может отнимать очень много времени.

Хотя основные принципы генерации набора сечений в Risk Spectrum такие же, как вышеописанные, программа имеет много особенностей, который, вместе взятые, чрезвычайно эффективно генерируют наборы минимальных сечений. Не минимальные наборы сечений генерируются в намного меньшей степени, чем в первоначальном алгоритме и если они сгенерированы, то процесс устранения очень эффективен.

Булевы законы, используемые при генерации набора сечений

В руководстве по теории не раскрывается ноу-хау алгоритма генерации МСО. Вместо этого описаны правила использования при этом законов алгебры логики. Эти правила достаточно просты и применяются на трех этапах генерации МСО:

Булевы законы, применяемые к каждому набору сечений

При каждом изменении набора сечений (сложение / замена нового события) применяются следующие Булевы законы:

- a) $A * A = A$
- b) $A * /A = 0$ ($/A = \text{NOT } A$)

Кроме того не сохраняются в наборах сечений дополненные события. Их единственная польза - действовать по логике в соответствии с Булевым законом b) выше. Результат этого - наборы сечений, которые содержат и событие и его дополнение устраняются.

Проверка того, является ли набор сечений минимальным, когда его генерация закончена

Когда все элементы набора сечений являются или основными событиями или модулями из множества МСО удаляется какое-либо событие. Затем этот новый набор событий проверяется, чтобы выяснить, происходит верхнее событие или нет. Если верхнее событие происходит, значит набор сечений не минимален (потому, что одно событие было удалено из набора, а верхнее событие все еще происходит). Если верхнее событие не происходит, тот же самый процесс повторяется, но теперь со вторым событием в удаленном наборе сечений. Это продолжается, пока набор сечений не станет не минимальным или пока все события не будут удалены по одному, и в каждом случае верхнее событие не будет происходить. В последнем случае, набор сечений минимален и сохраняется.

Проверка того, что набор минимальных сечений - не дубликат

Если набор сечений выявлен как минимальный, согласно шагу 2, он может все еще быть дубликатом, то есть тот же самый набор минимальных сечений уже был сгенерирован. Это проверяется путем сравнения только что сгенерированного МСО со всеми предварительно сгенерированными МСО. Если он не найден в существующем списке МСО, то он добавляется к списку.

Генерация МСО для верхних событий последовательностей

Для верхних событий последовательностей выполняются некоторые дополнительные шаги, включенные в процесс генерации МСО (и в процесс квантификации этих МСО).

- Программа сначала определяет, имеются ли в последовательностях какие-либо успешные функциональные события. Это делается путем определения местонахождения верхнего логического элемента успеха (NOR-логический элемент со всеми успешными функциональными событиями в последовательности как входами).

- МСО генерируются для верхнего логического элемента последовательности точно тем же путем, как для любого другого верхнего события. Из-за верхнего логического элемента успеха (NOR-логический элемент со всеми успешными функциональными событиями в последовательности как входами), список МСО, который генерируется, не содержит никаких МСО, которые являются логически противоречивыми. Другими словами, программа устраняет все МСО, которые не могут происходить, потому что некоторые функциональные события успешны.

- Если не имеется никаких успешных функциональных событий в последовательности, то генерация МСО заканчивается. Если верхний логический элемент успеха найден, программа генерирует МСО для этого верхнего логического элемента успеха. В этом процессе логический элемент обрабатывается как схема «ИЛИ». МСО, которые произведены - комбинации, которые приводят к отказу одного или большего количества успешных функциональных событий. Нас интересует дополнение этого верхнего события, так что квантификация (более поздний шаг) фактически вычисляет вероятность события дополнения.

- Список успешных МСО максимизируется до 1000 МСО. Кроме того при производстве этих МСО используются специальные значения отсечения (см. раздел «Процедура Отсечения»).

- МСО для верхнего логического элемента успеха сохраняются отдельно. Они также сохраняются в двоичном файле МСО (расширение .CSB) который используется во всех других типах исследований (неопределенность, значимость, зависимость от времени). Это означает, что успешные МСО определены количественно в этих исследованиях также.

Процедура отсечения

Для уменьшения объема вычислений при генерации МСО применяется процедура отсечения (усечения) МСО. В Risk Spectrum используются два основных способа отсечения:

- отсечение в соответствии с заданной величиной вероятности реализации сечения (абсолютным значением отсечения ACUTM или относительным значением отсечения RCUTM).

- отсечение в соответствии с порядком сечений. В этом случае все наборы сечений, содержащие большее количество событий, чем указанные этим значением отсечения, будут отброшены.

Вероятностное отсечение

Вероятностные значения отсечения можно задавать двумя способами:

- задавая величину абсолютного значения отсечения (ACUTM), например, 1.0 E-15.

- задавая величину относительного значения отсечения (RCUTM), например, 1.0 E-4.

Оба эти значения могут лежать в диапазоне от 0.0 до 0.999. Может быть использована любая комбинация абсолютных и относительных значений отсечения.

Задание абсолютного значения отсечения приводит к тому, что все наборы сечений, имеющие величину Кнг (или частоты) ниже, чем это значение отсечения отбрасываются при генерации множества МСО.

Относительное значение отсечения RCUTM используется следующим образом: После окончания генерации МСО программа вычисляет неготовность / частоту для каждого набора минимальных сечений, которые найдены. Поскольку МСО найдены и сохранены, может быть рассчитана сумма их неготовностей / частот. Эта сумма, QSUM, является приближенной верхней границей для неготовности / частоты верхнего события. Относительное значение отсечения RCUTM используется, чтобы вычислить новое (абсолютное) значение отсечения, которое связано с QSUM:

$$\text{Значение отсечения} = \text{RCUTM} \times \text{QSUM}$$

В качестве величины значения отсечения, используемом при усечении МСО используется самое большое значение из ACUTM или (RCUTM x QSUM).

Вероятностное отсечение используется в следующих случаях:

- Числовые данные и модели вычисления должны существовать для всех основных событий в дереве отказов
- Числовые вычисления верхнего события выполняются
- Список наборов минимальных сечений в файле результатов сортируется согласно неготовности / частоты для каждого набора сечений

Отсечение при генерации успешных МСО

При производстве МСО для верхнего логического элемента успеха в верхних событиях последовательности используется специальный набор значений отсечения:

$$\text{ACUTM} = 1.0\text{E-}6$$

$$\text{RCUTM} = 1.0\text{E-}3$$

Это - довольно сильное отсечение, но нужно помнить, что цель производства этих МСО состоит в нахождении вероятности успеха, когда он незначительно меньше 1. Другими словами, вероятность отказа (на котором МСО и отсечение основаны) - достаточно высока (по крайней мере относительно 1.0E-2), иначе это не важно.

Отсечение в соответствии с порядком сечений

Отсечение в соответствии с порядком используется, если значение отсечения $ACUTM \geq 1$. В этом случае все наборы сечений, содержащие большее количество событий, чем указанное этим значением отсечения $ACUTM$, будут отброшены.

Отсечение в соответствии с порядком МСО используется в следующих случаях:

- Числовые данные и модели вычисления не нужны для основных событий
- Числовые вычисления верхнего события не выполняются
- Список наборов минимальных сечений в файле результата сортируется согласно порядку набора сечений (длины набора сечений) для каждого набора сечений

Отсечение в соответствии с порядком фактически осуществляется как специальный тип вероятностного отсечения следующим способом: Всем основным событиям назначается вероятность отказа, равная 0.1, и значение отсечения тогда устанавливается внутри программы $1E-1$ для порядка 1, $1E-2$ для порядка 2, $1E-3$ для порядка 3 и т.д. Если выбрано отсечение в соответствии с порядком МСО, то квантификация верхнего события не выполняется.

Окно задания порога отсечения сечений при генерации МСО представлено на рис.3.6

Вычисление ошибки отсечения

Вычисление ошибки отсечения выполняется только при использовании вероятностного отсечения. При применении процедуры отсечения конечный список МСО не является полным отражением логики дерева отказов. Некоторые из наборов сечений отбрасываются (обрезаются) вследствие отсечения. В результате появляется «ошибка отсечения» (ошибка отбрасывания), связанная с квантификацией наборов сечений. Верхняя граница для этой ошибки отсечения оценивается Risk Spectrum.

Оценка ошибки отсечения делается следующим способом: - Программа всегда обновляет текущую неготовность для набора сечений, с которым работает в данный момент. В определенной стадии разработки, этот набор сечений может состоять из смеси основных событий, модулей и логических элементов. Неготовность основных событий и модулей известна, тогда как неготовность логических элементов не известна. Текущая неготовность набора сечений рассчитывается как произведение неготовностей основных событий и модулей. Неготовность логических элементов, таким образом, предполагается равной 1. Это конечно консервативно и обычно - очень завышенная оценка.

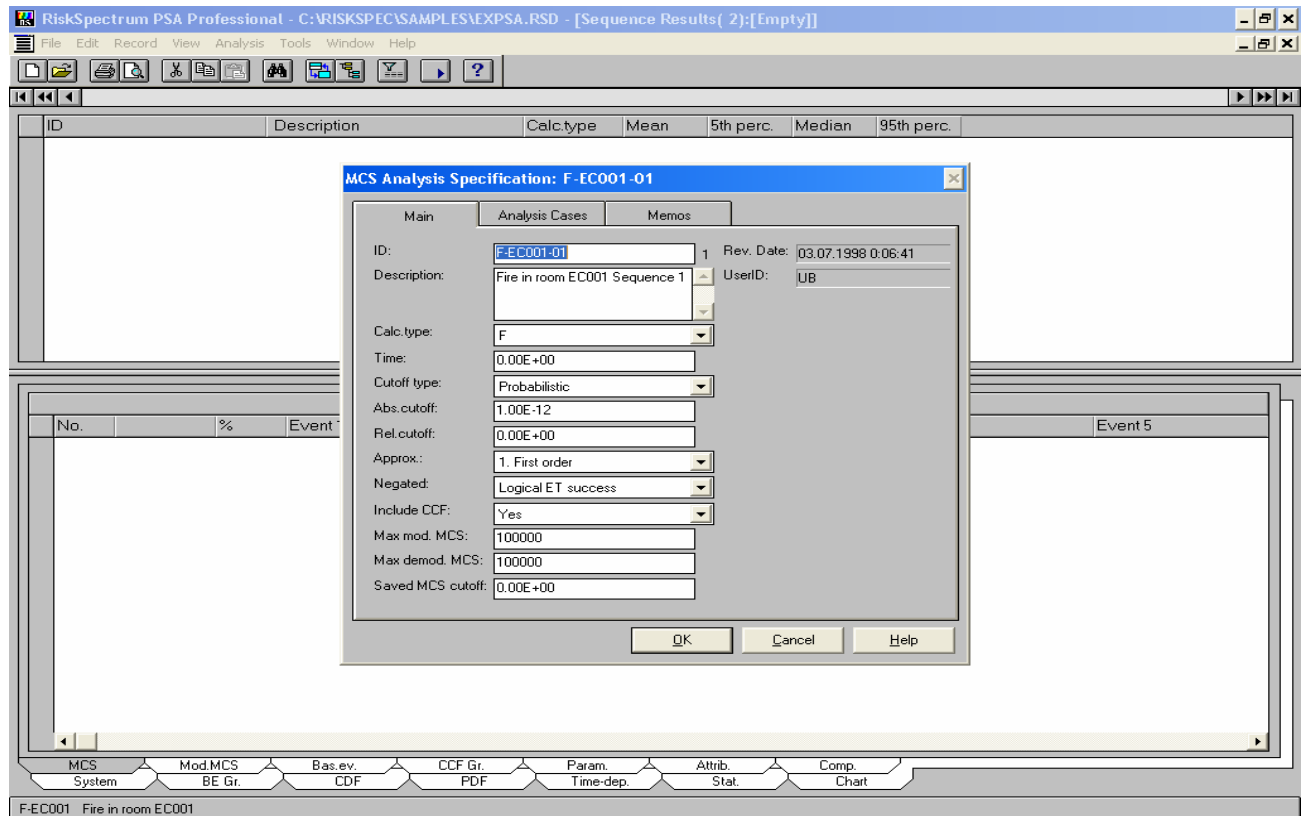


Рис. 3.6. Окно задания порога отсеечения сечений при генерации МСО

- Если текущая неготовность набора сечений, как описано выше, является более низкой чем значение отсеечения, набор сечений отбрасывается. Это может значительно сократить объем вычислений, потому что разработка оставшихся логических элементов в наборе сечений уже не будет выполняться.

- При отказе от набора сечений, как описано выше, текущая неготовность добавляется к переменной, которая является оценкой верхней границы ошибки отсеечения, то есть суммы неготовностей всех отвергнутых наборов сечений.

Величина ошибки отсеечения консервативна по двум причинам. Первая: логические элементы, для которых принята неготовность, равная 1, обычно имеют реальную неготовность, которая намного ниже 1. Вторая: отвергнутые наборы сечений не обязательно минимальны (большое количество их, вероятно, не минимально). Не минимальные наборы сечений действительно не должны быть включены в ошибку отсеечения.

В большинстве случаев значение отсеечения может быть откорректировано так, чтобы верхняя граница ошибки отсеечения была только малой долей неготовности верхнего события. Когда дерево отказов чрезвычайно большое и/или сложное, значение отсеечения приходится устанавливать таким, что ошибка отсеечения становится достаточно большой (в некоторых случаях сравнимой или даже выше, чем неготовность верхнего события). Это при-
 скорбно, но к сожалению очень трудно преодолимо. С известными в настоящее время алгоритмами не остается ничего делать с этим, кроме как увеличить число наборов сечений, ко-

торые могут быть обработаны программой (которая учтет более низкие значения отсечения). Это, с другой стороны, может вызвать проблемы с памятью на персональных компьютерах. Будущие версии смогут обеспечить лучшую характеристику в этом отношении.

Нужно помнить, однако, что ошибка отсечения обычно **чрезвычайно** переоценивается, во многих случаях в 10-100 раз и, возможно, даже больше. Не имеется, к сожалению, точного метода, чтобы предсказать насколько завышена оценка. Это должно делаться из раза в раз аналитиком, например, путем изучения списка МСО. Если, скажем, первые 1000 наборов сечений напечатаны, и первые 100 явно доминируют, а остальная часть напечатанного множества МСО дает очень малый вклад, вероятно, что ошибка отсечения на самом деле является малой.

Автоматическое регулирование значения отсечения

Если в ходе генерации наборов сечений число построенных из модулей наборов сечений превышает указанный пользователем максимум или если ПК вышел за пределы памяти, значение отсечения автоматически увеличивается программой. Эта процедура регулирования отсечения работает следующим образом:

1. Вероятностное отсечение

Если текущее значение отсечения меньше, чем $1.0E-6$ от суммы неготовностей (или частот) всех МСО, сгенерированных таким образом (QSUM), значение отсечения корректируется, чтобы быть $1.0E-6 \times QSUM$

Во втором случае значение отсечения устанавливается так, чтобы быть на коэффициент $10^{1/2}$ (квадратный корень 10) выше чем текущее значение отсечения. Это означает, что две последовательных операции приведут к увеличению значения отсечения в 10 раз.

2. Отсечение по порядку набора сечений

Порядок отсечения уменьшается на один.

Процесс регулирования отсечения будет повторяться, пока результат не будет получен. Таким образом, выполнение программы никогда не заканчивает из-за «слишком большого количества сечений».

Каждый раз при регулировании значения отсечения в файл журнала анализа пишется сообщение в виде «значение отсечения увеличилось до ...». Конечное значение отсечения также пишется в файл результата.

Применяется один и тот же тип процедуры независимо от введения абсолютных или относительных значений отсечения (или любой комбинации). Всегда используется абсолютное значение отсечения. Разница только в том, что когда задано относительное отсечение изменяется величина порога абсолютного отсечения, поскольку большее количество МСО будет найдено.

В ходе процесса демодуляризации число МСО может также превысить максимальный предел. Если это имеет место, то применяется тот же самый тип регулирования отсечения. В этом случае, однако, отсечение не может иметь никакого влияния на числовые результаты (так как квантификация уже была сделана на основе построенных из модулей МСО). Информация относительно этого типа регулирования отсечения также пишется в файл журнала.

Рекомендации для выбора порога отсечения

Для вероятностного отсечения может быть использована любая комбинация абсолютного отсечения ACUTM и относительного отсечения RCUTM. Вообще, хорошо подобранное абсолютное отсечение более эффективно, чем относительное отсечение (то есть ведет к более короткому времени выполнения). Однако, когда деревья отказов очень сложны, не просто выбрать «оптимальное» абсолютное значение отсечения, которое ведет и к короткому времени выполнения и к приемлемо низким ошибкам отсечения. В таких случаях может быть легче использовать относительную ошибку отсечения. Как правило, относительные значения отсечения $1E-5$ - $1E-4$ дадут разумно низкие ошибки отсечения.

В крупномасштабном ВАБ обычно рекомендуется использовать следующие значения:

Абсолютное значение отсечения, ACUTM: $1.0E-12$ - $1.0E-15$

Относительное значение отсечения, RCUTM: $1.0E-3$ - $1.0E-5$

Эти параметры порога отсечения будут обеспечивать хорошее равновесие между точностью результатов (низкая ошибка отсечения) и временем выполнения программы. Более высокие значения отсечения рекомендуются в главной рабочей фазе ВАБ, в то время как модель развивается и выполняется много расчетов, чтобы проверить модель и получить предварительные результаты. В этой стадии более важна быстрота, чем низкие ошибки отсечения. Более низкие значения отсечения рекомендуются для вычисления конечных результатов.

Преимущество комбинации низкого абсолютного значения отсечения и немного «более резкого» относительного значения отсечения - то, что та же самая установка отсечения может быть применена ко всем верхним событиям с хорошими результатами.

1.3.8. Квантификация МСО

Квантификация верхнего события выполняется в ходе следующих этапов расчета:

Вычисление неготовности каждого основного события

Вычисление неготовности событий ООП

Вычисление неготовности модулей

Вычисление неготовности каждого МСО

Вычисление неготовности / частоты верхнего события на основе формулы, которая «подводит итог» неготовностям / частотам всем МСО

Шаги 1, 2 и 3 фактически делаются перед началом процесса генерации набора сечений, потому что соответствующие числовые значения необходимы при реализации отсечения.

При выполнении шага 5 могут быть применены несколько различных методов. В Risk Spectrum могут быть выбраны три различных уровня аппроксимации, называемые аппроксимацией *первого*, *второго*, и *третьего* порядка. Названия даны в соответствии с тем, что различные аппроксимации рассматривают члены *первого* порядка, *второго* порядка, и *третьего* порядка, соответственно.

Квантификация основных событий

Неготовности для основных событий рассчитываются в соответствии с формулами, описанными выше. Способ вычислений определяется заданием *типа вычислений* в технических требованиях анализа МСО:

Таблица 3.10.

| Тип вычисления неготовности | Обозначение |
|-----------------------------|-------------------|
| Среднее значение | Q_{mean} |
| Функция времени | $Q(T)$ |
| Частота | Q_{mean} |

Квантификация ООП

Квантификация событий CCF делается согласно вышеприведенным формулам. «Полная неготовность компонента» Q_{tot} , использованная в этих формулах, является или зависимой от времени неготовностью $Q(T)$ или средней неготовностью Q_{mean} , в зависимости от выбора типа вычисления в технических требованиях МСо-анализа:

Таблица 3.11.

| Тип вычисления неготовности | Обозначение |
|-----------------------------|------------------------------------|
| Среднее значение | $Q_{\text{tot}} = Q_{\text{mean}}$ |
| Функция времени | $Q_{\text{tot}} = Q(T)$ |
| Частота | $Q_{\text{tot}} = Q_{\text{mean}}$ |

Когда рассчитывается неготовность $Q(T)$ для ООП при задании периодически проверяемой модели надежности, параметры времени до первой проверки обрабатываются так, как описано в разделе «Квантификация CCF для шахматного испытания».

Квантификация модулей

После расчета характеристик надежности основных событий и ООП могут быть рассчитаны неготовности модулей.

Модуль может состоять из основных событий, событий ООП, других модулей или любой смеси из этих элементов, как прямых входов. Модули рассчитываются в «восходящем» порядке. Модуляризация делается таким образом, что базисные события типа «частоты»

та» (тип 5) не могут входить в модули, так как модульные вычисления не предназначены для таких расчетов.

В формулах для модульной квантификации использована следующая система обозначений:

Таблица 3.12.

| | |
|-------|----------------------------------------------------------------------------|
| Q_i | Неготовность входа i , (основное событие, событие ООП или другой модуль) |
| k | Число входов в модуль |

Порядок расчетов определяется выбором *типа вычисления* в технических требованиях МСО-анализа:

Таблица 3.13.

| Тип вычисления неготовности | Обозначение |
|-----------------------------|-------------------------|
| Среднее значение | $Q_i = Q_{\text{mean}}$ |
| Функция времени | $Q_i = Q(T)$ |
| Частота | $Q_i = Q_{\text{mean}}$ |

Вычисление неготовности для ИЛИ-модулей производится по формуле:

$$Q_{\text{mod}} = 1 - \prod_{i=1}^k (1 - Q_i) \quad (24)$$

Вычисление неготовности для И-модулей производится по формуле:

$$Q_{\text{mod}} = \prod_{i=1}^k Q_i \quad (25)$$

Эти формулы дают **точные** оценки величины неготовности (без аппроксимации) для всех ИЛИ-модулей и И-модулей за следующим исключением: Если модуль был первоначально К/Н-модулем, то он преобразуется в эквивалентную комбинацию ИЛИ- и И-схем. Это ведет к неточной оценке величины неготовности для К/Н-модулей, но очень хорошо аппроксимирована, если неготовности входов в К/Н-модуль низки (< 0.1).

Вычисление для К/Н-модулей опишем с помощью примера.

Предположим, что имеется логический элемент (G1 - 2/3) со входами А, В и С. Входы А, В, и С используются в дереве отказов только по одному разу, так что логический элемент G1 действительно модуль. Этот модуль преобразуем в OR-модуль с тремя И-модулями в качестве входов:

| | | | | |
|----|-----|----|----|----|
| G1 | ИЛИ | M1 | M2 | M3 |
| M1 | И | A | B | |
| M2 | И | A | C | |
| M3 | И | B | C | |

После этого преобразования рассчитывается величина неготовности модулей путем использования формул для ИЛИ-модулей и И-модулей. Примем, что неготовность Q каждого события (A, B, C) равна 0.1 и получим следующие результаты:

$$Q(M1) = Q(M2) = Q(M3) = Q(A) * Q(B) = 0.1 * 0.1 = 0.01$$

$$Q(G1) = 1 - (1 - Q(M1)) * (1 - Q(M2)) * (1 - Q(M3)) = 0.297$$

Результат неточен, потому что после преобразования «модули» M1, M2 и M3 уже не являются модулями в строгом смысле слова вследствие наличия зависимостей между модулями M1, M2 и M3 (события A, B, C входят более чем в один модуль).

Квантификация наборов минимальных сечений

После расчетов показателей надежности основных событий, ООП и модулей может быть рассчитана величина неготовности или частоты каждого МСО.

Сечение может состоять из основных событий, событий ООП, модулей или любой смеси из них. В формулах для квантификации МСО использована следующая система обозначений:

Таблица 3.14.

| | |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Q_i | Неготовность элемента i в МСО (основного события, события ООП или другого модуля) |
| F_1 | Частота основного события. Применяется только когда задан <i>тип вычисления – частота</i> . Имеется всегда точно одно событие с типом «частота» в каждом МСО для этих типов вычисления. |
| k | Число элементов (основные события, события ООП или модули) в МСО |

Порядок расчетов определяется выбором *типа вычисления* в технических требованиях МСО-анализа:

Таблица 3.15.

| Тип вычисления неготовности | Обозначение |
|-----------------------------|-------------------------|
| Среднее значение | $Q_i = Q_{\text{mean}}$ |
| Функция времени | $Q_i = Q(T)$ |
| Частота | $Q_i = Q_{\text{mean}}$ |

Вычисление неготовности МСО производится по формуле:

$$Q_{MCS} = \prod_{i=1}^k Q_i \quad (26)$$

Вычисление частоты МСО производится по формуле:

$$F_{MCS} = F_1 \prod_{i=2}^k Q_i \quad (27)$$

Квантификация с минимальной верхней границей сечения (задается по умолчанию)

Нормальная аппроксимация первого порядка часто называется «аппроксимацией редких событий», потому что она дает неплохие результаты, если вероятности отказов малы.

В применении к МСО неготовность верхнего события, согласно аппроксимации редких событий, это просто сумма неготовностей МСО:

$$Q_{TOP,1} = \sum_{i=1}^n Q_{MCS,i} \quad (28)$$

Программа также вычисляет другую аппроксимацию, которая также является аппроксимацией «первого порядка», но дает несколько лучшие результаты, чем аппроксимация редких событий. Эта аппроксимация часто называется *минимальной верхней границей сечения*. Ее значение вычисляется по следующему выражению:

$$Q_{TOP,MCUB} = 1 - \prod_{i=1}^n (1 - Q_{MCS,i}) \quad (29)$$

В обеих формулах $Q_{MCS,i}$ относится к неготовности MCS_i и эта неготовность является или средней неготовностью или функцией времени T , в зависимости от выбора типа вычислений в технических требованиях МСО-анализа.

Аппроксимация второго и третьего порядка

Вычисление аппроксимаций второго и третьего порядка основано на «Принципе включения-исключения». При использовании этого метода последовательно получают верхние и нижние границы вероятности верхнего события, вначале добавлением суммы всех членов первого уровня, затем вычитанием суммы всех членов со вторым порядком, затем добавлением всех членов с третьим порядком и так далее.

Поясним как работает метод на простом примере.

Пусть имеется множество МСО для системы «3 из 4». Обозначим компоненты системы буквами А, В, С, D. Имеем следующие МСО: А В С - А В D - А С D - В С D

К членам первого порядка относятся: $P(1) = P(ABC) + P(ABD) + P(ACD) + P(BCD)$

Примечание:

Эта формула использована для членов первого порядка только при использовании аппроксимации второго или третьего порядка. При выборе аппроксимация первого порядка программа использует минимальную верхнюю границу сечения.

Кроме того, если имеется два или больше МСО содержащих только по одному событию, то они перед выполнением вычислений комбинируются в своего рода «модуль». Для этого «модуля», минимальная верхняя граница сечения **точно** и «модуль» может быть обработан при дальнейших вычислениях так, как будто это отдельное базисное событие. Данное утверждение основано на допущении о независимости базисных событий..

Члены второго порядка:

$P(2) = P(MCS_1 \& MCS_2) + P(MCS_1 \& MCS_3) + P(MCS_1 \& MCS_4) + P(MCS_2 \& MCS_3) + P(MCS_2 \& MCS_4) + P(MCS_3 \& MCS_4) = P(ABCD) + P(ABCD) + P(ABCD) + P(ABCD) + P(ABCD) + P(ABCD) + P(ABCD)$

Члены третьего порядка:

$$P(3) = P(MCO1 \& MCO2 \& MCO3) + P(MCO1 \& MCO2 \& MCO4) + P(MCO1 \& MCO3 \& MCO4) + P(MCO2 \& MCO3 \& MCO4) = (ABCD) + P(ABCD) + P(ABCD) + P(ABCD)$$

Наконец, имеем различные аппроксимации для вероятности верхнего события согласно оценкам членов первого, второго и третьего порядка по принципу исключения-включения:

$$\text{Аппроксимация первого порядка (верхняя граница)} = P(\text{TOP1}) = P(1)$$

$$\text{Аппроксимация второго порядка (нижняя граница)} = P(\text{TOP2}) = P(1) - P(2)$$

$$\text{Аппроксимация третьего порядка (верхняя граница)} = P(\text{TOP3}) = P(1) - P(2) + P(3)$$

Рассчитаем вероятности верхнего события, приняв соответственно для всех компонентов равные вероятности отказа: 0.01; 0.1; 0.5 и 0.9. Результаты расчетов сведем в таблицу 3.16

Таблица 3.16.

| №№ п/п | Pi | P(TOP1) | P(TOP2) | P(TOP3) | Точное значение | Min верхняя граница |
|-----------|------|----------|----------|----------|-----------------|---------------------|
| 1 | 0.01 | 4.000E-6 | 3.940E-6 | 3.980E-6 | 3.970E-6 | 4.00E-6 |
| 2 | 0.1 | 4.000E-3 | 3.400E-3 | 3.800E-3 | 3.700E-3 | 3.99E-3 |
| 3 | 0.5 | 0.50000 | 0.12500 | 0.37500 | 0.3125 | 0.4138 |
| 4 | 0.9 | 2.9160 | -1.0206 | 1.6038 | 0.9477 | 0.9946 |

Как можно заметить из данных таблицы, последовательные верхние и нижние границы не обязательно монотонно сходятся. Это явление может обнаруживаться в Risk Spectrum, когда имеется более трех МСО (МСО-модулей) и когда вероятности реализации сечений очень высоки, т.о. аппроксимации первого, второго и третьего порядка не сходятся к истинному значению. Напротив, значения расходятся и иногда могут полностью не соответствовать истине (например, отрицательная аппроксимация второго порядка). Это плохо, но это - не признак ошибки в алгоритме Risk Spectrum; это - свойство данного метода.

Как можно заметить, все аппроксимации достаточно хороши, когда вероятности отказов малы (менее 0.01). Верхние и нижние границы довольно близки, когда вероятности - 0.1, давая довольно хорошую аппроксимацию. Границы все еще дают разумные результаты, когда вероятности отказа равны 0.5, даже при том, что аппроксимация довольно плоха в этом случае. Однако, когда вероятности отказа равны 0.9 аппроксимации дают полностью не соответствующие действительности числа. Достоинство этого метода в том, что можно быть уверенным, что результат первого порядка является верхней границей, результат второго порядка - нижней границей, и результат третьего порядка - верхней границей истинного значения. Это очень важно и дает, фактически, единственное правило, которое нужно помнить, когда приступаешь к принятию решений относительно достоверности результатов:

Если получены верхние и нижние границы, которые близки друг к другу, аппроксимация хороша; если границы далеки друг от друга, аппроксимация плоха. (Близость границ может меняться от случая к случаю).

Когда программа выполняет аппроксимацию второго и третьего порядка, она фактически применяет процедуру отсечения, чтобы ускорить вычисления. Дело в том, что число членов логической функции возрастает очень быстро. Приблизительно число членов второго порядка есть квадрат количества МСО, а число членов третьего порядка - куб числа МСО. Другими словами, для 1000 МСО число членов логической функции при аппроксимации третьего порядка составляет около $1 \text{ E}+9$. Это может приводить к весьма существенному увеличению времени расчетов. Дело в том, при генерации каждого нового члена логической функции используются законы алгебры логики для обработки наборов сечений, и лишь затем выполняются вычисления с плавающей точкой (расчет числовых показателей).

Процедура отсечения, используемая программой, применяется в целях уменьшения числа членов функции, которые учитываются программой. Работает данная процедура следующим образом:

Сначала рассчитывается и сохраняется число $M = 10 * N$, где N – количество МСО.

Затем рассчитываются значения порогов отсечения: $C1 = QTOP/M$ и $C2 = C1/M$, где $QTOP$ - минимальная верхняя граница сечения.

При дальнейших вычислениях каждое отдельное МСО объединяется со всеми другими МСО, создавая таким образом до $N-1$ членов второго порядка (первое МСО генерирует $N-1$ членов второго порядка, второе - $N-2$ новых членов и т.д.). Если вероятность реализации очередного МСО окажется ниже, чем значение отсечения $C1$, члены второго и третьего порядка, включающие комбинации с этим МСО, не рассчитываются. Максимальная теоретически возможная ошибка, получаемая при пренебрежении этими членами составляет $QTOP/M$, а максимальная ошибка в случае, если отсечение было бы применено ко всем МСО - меньше 10 % от $QTOP$.

Когда получен член второго порядка он объединяется с другими МСО, чтобы генерировать члены третьего порядка. Перед генерацией членов третьего порядка выполняется другая проверка на отсечение. В этом случае члены третьего порядка не рассматриваются, если текущий член второго порядка имеет вероятность, более низкую, чем $C2$. В самом плохом (теоретически) случае ошибка, полученная при этом, даже более низка, чем максимальная ошибка для членов второго порядка.

Разработчики ПК утверждают, что при решении практических задач описанная выше процедура отсечения не влияет на точность результатов. Однако ее влияние на время выполнения может быть весьма значительным.

Квантификация верхних событий последовательности

Как указывалось выше, для верхних событий последовательности программа генерирует МСО для так называемого верхнего логического элемента **успеха**. Список «успешных» МСО определяется путем использования минимальной верхней границы сечения. Если Q – неготовность, то вероятность успеха является обратной величиной:

$$Q(\text{успех}) = 1 - Q$$

Данный результат используется при вычислении частоты последовательности:

$$Q_{TOP} = Q(\text{отказ}) * Q(\text{успех})$$

При этой квантификации потенциальными зависимостями между «неудачными» МСО и «успешными» МСО пренебрегают. Это аппроксимация (приближение). Правильная квантификация должна фактически вычислить условное $Q(\text{успеха})$ возникновения «неудачного» верхнего логического элемента. Ошибка, полученная путем пренебрежения этой потенциальной зависимостью, дает консервативный результат. Это потому, что вероятность неуспеха Q (см. выше) недооценена (завышена), следовательно вероятность $Q(\text{успех})$ завышена.

1.3.9. Анализ неопределенности

При анализе неопределенности вычисляется распределение вероятности для результатов событий (в противоположность точечному значению, рассчитываемому в МСО-анализе). Анализ неопределенности основан на использовании метода Монте-Карло.

При анализе неопределенности для верхнего события выполняются следующие шаги:

Чтение технических требований к анализу неопределенности из записи параметров анализа.

Чтение МСО-модулей из двоичного файла результатов (с расширением .RSR).

Чтение из базы данных проекта данных для основных событий и параметров.

Методом Монте-Карло вычисляется неготовность верхнего события NSIM раз (NSIM - число итераций, заданное в технических требованиях). Каждый раз методом Монте-Карло рассчитываются все параметра надежности.

Распечатка в файл результатов полученного распределения неопределенности.

1.3.9.1. Технические требования анализа неопределенности

При задании технических требований анализа неопределенности определяются значения следующих параметров:

Таблица 3.17.

| | |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NSIM | Число итераций в процессе моделирования методом Монте-Карло. Рекомендуется задавать по крайней мере 1000 итераций. |
| Ав- то/Ручное начальное число | Моделирование Монте-Карло основано на использовании генератора случайных чисел, который нуждается в начальном (стартовом) значении. Если «Ав-то» выбрано, программа выбирает значение начального числа автоматически (случайное начальное число, расчетное для текущего времени). Если выбрано «Ручное», пользователь должен задать значение начального числа сам (см. ниже). |
| Началь- ное число | Ручное значение начального числа, если выбрано задание его вручную. Если используется одно и то же значение начального числа, то можно воспроизвести идентичные результаты в двух анализах неопределенности, потому что конкретное начальное число приводит к установленной последовательности псевдослучайных чисел. |

1.3.9.2. Моделирование значений параметров

В процессе моделирования результат верхнего события рассчитывается NSIM раз. В каждой из итераций все параметры зациклены. Для каждого параметра моделируется распределения неопределенности значения параметра.

Моделирование выполняется следующим способом:

В ходе анализа генерируется случайное значение между 0 и 1, которое является моделируемой процентилью, то есть значением от ЧПЗ. Значение, используемое из распределения, получают по различным опубликованным алгоритмами для различных типов распределения.

1.3.9.3. Вычисление результатов верхнего события в процессе моделирования

Когда смоделированы все значения параметров рассчитывается значение вероятности верхнего события, основанное на этих значениях.

Все вычисления для основных событий, событий ООП, модулей и МСО согласно описанным выше правилам. Квантификация МСО всегда делается при использовании *аппроксимации минимальной верхней границы сечения* (аппроксимации первого уровня).

Вычисление неготовности для основных событий выполняется в соответствии с *типом вычислений*, выбранного в технических требованиях анализа МСО (среднее, функция времени, частота).

1.3.9.4. Сбор распределения неопределенности

Все NSIM смоделированных результатов оценки заданного параметра верхнего события сохраняются в специальном массиве программы. Этот массив затем сортируется и производится вычисление значений процентили и средние значения.

При этом используется следующая система обозначений:

Таблица 3.18.

| | |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Q(i) | Число примеров неготовности верхнего события i (в массиве всех NSIM смоделированных значений, то есть. $i = 1, 2, \dots, NSIM$) |
| Q_M | Среднее значение распределения значений Q |
| Q_X | Процентиль X распределения значений Q |

Среднее значение рассчитывается как среднее арифметическое всех имитируемых неготовностей верхнего события:

$$Q_M = \sum_{i=1}^{NSIM} \frac{Q(i)}{NSIM} \quad (30)$$

Значения процентиля рассчитываются следующим способом (при этом массив Q(i) должен сортироваться в порядке возрастания):

$$Q_x = Q\left(\frac{X}{100} NSIM\right) \quad (50)$$

Чтобы получить график совокупной плотности распределения (CDF) используются следующие значения в файле результатов:

«Процентиль» / 100 как Y-значение

«Значение процентиля» как X-значение

Чтобы получить график функции плотности вероятности столбец «Плотность вероятности» дается в файле результатов. Это значение рассчитывается следующим способом:

$$Prob. density = \frac{P_i - P_{i-1}}{PV_i - PV_{i-1}} \quad (51)$$

где P_i – Процентиль i ,

PV_i - значение процентиля i в таблице.

После расчета всех значений PDF согласно вышеприведенной формулы, они нормируются так, чтобы их сумма была равна 1.

Значения PDF могут быть использованы для составления графика PDF (гистограммы) следующим способом:

PDF = 0 для X-значений ниже первого значения процентиля (процентиль 0)

PDF = второе значение PDF в X-интервале от первого значения процентиля до третьего значения процентиля

и т.д.

1.3.10. Анализ значимости и чувствительности

Назначение анализа значимости состоит в вычислении значимостей и специального типа меры чувствительности для:

отдельных базисных событий

групп основных событий, характеризуемых одним и тем же атрибутом

групп базисных событий, которые принадлежат одному компоненту

групп базисных событий, которые принадлежат одной системе

групп базисных событий, определенных в специальных группах основных событий, заданных в Risk Spectrum

групп ООП

отдельных параметров

Для каждой из этих категорий рассчитываются три различные меры значимости:

Дробный вклад

Значимости по Фусселю-Везели (только для базисных событий и событий ООП)

Фактор уменьшения риска (также известный как стоимость снижения риска)

Фактор увеличения риска (также известный как стоимость достижения риска)

Все вычисления для основных событий, событий ССF, модулей и МСО выполняются в соответствии с описанными ранее правилами. Квантификация МСО всегда выполняется при использовании *аппроксимацию минимальной верхней границы сечения* (первого уровня аппроксимации).

Вычисление неготовности для основных событий выполняется в соответствии с заданным *типом вычислений* (среднее, функция времени, частота).

Во всех таблицах значимости и чувствительности перечисляются (выводятся на распечатку) N наиболее важных элементов. Значение N задается в технических требованиях анализа значимости для отдельных базисных событий, атрибутов и параметров. Для групп базисных событий все заданные группы включаются в распечатку. Необходимо отметить, что всегда вычисляются **все** значения значимости каждого типа, N влияет только на объем распечатки.

1.3.10.1. Значимость для базисных событий

Значимость по Фусселю-Везели

Значимость по Фусселю-Везели (FV) для базисного события рассчитывается следующим образом:

Вычисляется неготовность верхнего события на основе только тех МСО, которые содержат данное событие (это аналогично установке неготовности, равной нулю, для всех МСО, *не* содержащих данное событие)

Вычисляется FV значимость как частное от деления неготовности, рассчитанной в соответствии с пунктом 1 и номинальной неготовностью верхнего события:

$$I_i^{FV} = \frac{Q_{TOP}(MCS \text{ including } i)}{Q_{TOP}} \quad (52)$$

Другие показатели значимости

Все вычисления значимости для базисных событий (индивидуальных или групп) выполняются, в основном, тем же способом:

неготовность для базисного события или для всех базисных событий в группе задается равной $Q = 0$, то есть принимается, что компонент абсолютно надежен. Для основного события с 5 типом модели надежности устанавливается частота $f=0$.

Вычисляется новый результат верхнего события (неготовность или частота, в зависимости от заданного типа вычислений). Этот новый, меньший, результат обозначается Q_{TOP} ($Q_i=0$).

Теперь могут быть рассчитаны:

- фактор уменьшения риска:

$$I_i^R = \frac{Q_{TOP}}{Q_{TOP}(Q_i = 0)} \quad (53)$$

- дробный вклад:

$$I_i^F = I - \frac{I}{I_i^R} \quad (54)$$

неготовность для базисного события или для всех базисных событий в группе задается равной $Q_i = 1$, то есть принимается, что компонент достоверно отказал. Эта операция делается для основных событий с 5 типом модели надежности.

Вычисляется новый результат верхнего события (неготовность или частота, в зависимости от заданного типа вычислений). Этот новый результат обозначен $Q_{TOP}(Q_i=1)$.

Теперь может быть рассчитан фактор увеличения риска:

$$I_i^I = \frac{Q_{TOP}(Q_i = 1)}{Q_{TOP}} \quad (55)$$

1.3.10.2. Значимость для групп основных событий

Значимость для групп основных событий рассчитывается таким же образом, как и для отдельных основных событий, но неготовности задаются равными $Q = 0$ и $Q = 1$ для **всех** событий, которые принадлежат группе на шагах 1 и 3.

1.3.10.3. Значимость для атрибутов

Значимость для атрибута это значимость группы базисных событий, которым присвоен признак этого атрибута. Поэтому, вычисление значимости для атрибутов выполняется так же образом как и для групп основных событий.

1.3.10.4. Значимость для параметров

Вычисление значимости для параметров производится в соответствии с той же процедурой, что и для основных событий. В отдельных случаях показатели значимости либо не могут быть определены, либо будут бессмысленными. Это относится к таким параметрам, как времена до первой проверки (T_1), фактор увеличения риска для частоты (f).

Для вычисления значимости параметра необходимо:

Установить значение параметра «лучший из всех теоретически возможных» (во всех случаях это $X=0$). Это относится ко всем типам параметров, кроме параметра времени до первой проверки (TF).

Вычислить новый результат вероятности осуществления верхнего события (неготовность или частота, в зависимости от заданного типа вычисления). Этот новый, меньший, результат обозначим как $Q_{TOP}(X=0)$.

Теперь фактор уменьшения риска может быть рассчитан с помощью формулы вида:

$$I_i^R = \frac{Q_{TOP}}{Q_{TOP}(X_i = 0)} \quad (56)$$

а дробный вклад - с помощью формулы вида:

$$I_i^F = I - \frac{I}{I_i^R} \quad (57)$$

Установить значение параметра «наихудший теоретически возможный». Это значение зависит от типа параметра, для вероятностных параметров это $q=1$ и $X=\infty$ для всех других параметров. Это неприменимо для параметров частоты, потому что бесконечное значение частоты подразумевало бы частоту верхнего события, равную бесконечности.

Вычислить новый результат верхнего события (неготовность или частота, в зависимости от заданного типа вычислений). Этот новый, больший, результат обозначим Q_{TOP} ($q_i=1$). Тогда фактор увеличения риска может быть рассчитан следующим образом.

а) для вероятностных параметров:

$$I_i^I = \frac{Q_{TOP}(q_i = 1)}{Q_{TOP}} \quad (58)$$

б) для всех других типов параметра X (кроме параметров частоты (f) и времени до первой проверки (TP), для которых вычисления не делаются):

$$I_i^I = \frac{Q_{TOP}(X_i = \infty)}{Q_{TOP}} \quad (59)$$

1.3.10.5. Вычисления показателей чувствительности

Все вычисления показателей чувствительности (для основных событий, для групп событий и для параметров) выполняются в основном одним и тем же способом:

Рассматриваемое значение ($я$) (неготовность, частота или параметр) устанавливается равным номинальному значению, деленному на фактор чувствительности. Фактором чувствительности может быть любое число большее 1. По умолчанию задается значение равное 10. Для групп основных событий номинальные неготовности делятся на фактор чувствительности для всех основных событий в группе.

Вычисляется новый результат верхнего события (неготовность или частота, в зависимости от заданного типа вычислений) - $Q_{TOP,L}$.

Рассматриваемое значение ($я$) (неготовность, частота или индивидуальное значение параметра) устанавливается равным номинальному значению, умноженному на фактор чувствительности. Для групп основных событий номинальные неготовности умножаются на 10 для всех основных событий в группе. Если какое-либо значение после умножения находится вне дозванного диапазона, то оно усекается до максимально допустимого значения. Вероятности, например, усекаются до 1.

Вычисляется новый результат верхнего события (неготовность или частота, в зависимости от заданного типа вычислений) - $Q_{TOP,U}$.

Вычисляется значение параметра, которое определяется как показатель чувствительности:

$$S = \frac{Q_{TOP,U}}{Q_{TOP,L}} \quad (60)$$

После этого программа записывает значения $Q_{TOP,L}$, $Q_{TOP,U}$ и чувствительности S . Таблица результатов сортируется по величине S .

1.3.11. Выполнение анализа показателей надежности и безопасности как функции времени.

При анализе показателей надежности и безопасности как функций времени вычисляется распределение вероятности для исхода верхнего события.

При этом выполняются следующие этапы моделирования и расчета:

Из записи параметров верхнего события считываются технические требования анализа;

Из двоичного файла результатов (с расширением .RSR) считываются параметры МСО-модулей;

Из базы данных проекта считываются данные для основных событий и параметров; Создается массив всех моментов времени, которые будут использованы при дальнейших вычислениях. Способ создания массива зависит от выбора способа расчета промежуточных моментов времени в технических требованиях анализа: «Авто» или «Ручного».

Вычисления выполняются в цикле для каждого промежуточного момента времени. Так создается таблица результатов, в которой каждая строка соответствует определенному заданному моменту времени.

Вычисляются значения тех параметров, для которых необходимо интегрирование промежуточных значений для моментов времени, например, средние значения. Результаты печатаются в файл результатов.

1.3.11.1. Технические требования анализа показателей надежности и безопасности в функции времени

К техническим требованиям анализа зависимости от времени относятся:

Таблица 3.19.

| | |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TE | Максимальный рассматриваемый промежуток времени, используемый при вычислениях |
| Авто / Ручные | «Авто» означает, что программа автоматически вычисляет и рассматриваемый промежуток времени TE и моменты (интервалы) времени, для которых выполняются вычисления |
| NI | Число интервалов. Определяет моменты времени, для которых выполняется анализ |

1.3.11.2. Расчет интервалов времени

При «ручном» задании параметров TE и NI моменты времени для расчетов определяются по формуле:

$$T_i = i \frac{TE}{NI} \quad i = 0, 1, 2, \dots, NI \quad (61)$$

При задании автоматического определения моментов времени это выполняется путем анализа заданных параметров базисных событий, включенных в МСО-анализ:

В цикле проверяются все базисные события 1-го и 2-го типа. Самое большое значение времени восстановления (TR) для событий 1-го типа умножается на пять и сохраняется как TE1. Самый большое значение интервала между проверками (TI) среди компонентов 2-го типа сохраняется как TE2. Больше из значений TE1 и TE2 присваивается параметру TE, «полным» интервалом. Если значение параметра TE в технических требованиях анализа превышает величину любого из параметров TE1 и TE2, то величина TE приравнивается к нему. Интервал TE делится на 100 подинтервалов.

В цикле анализируются все основные события 2-го типа. Для каждого из них прибавляются следующие моменты времени (если они еще не заданы):

- пять моментов времени для каждого цикла интервала между проверками до TE: Каждый интервал между проверками разделен на 4 подинтервала. Так генерируется 5 моментов времени, включая начальную и конечную точки интервала между проверками. Точка отсчета определяется так, чтобы она была сразу после испытания, а конечная точка - перед испытанием.
- если для базисного события заданы параметры TR И TF (время до первой проверки), то «местоположения» момента времени «перемещаются» к более высоким значениям, соответствующим значениям TR и TF. Кроме того, при задании TF специальный цикл интервала первой проверки определяется с 5 моментами времени так же, как было указано выше.

Максимальное число моментов времени - 32767.

1.3.11.3. Вычисление результатов верхнего события для каждого момента времени

Для каждого момента времени T рассчитываются следующие параметры верхнего события:

Таблица 3.20.

| | |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Q(T) | Неготовность в момент времени T. Вероятность того, что моделируемая система (верхнее событие) находится в состоянии отказа в момент времени T. |
| W(T) | Частота (безусловная интенсивность отказа) в момент времени T. Ожидаемое число отказов в единицу времени (обычно час) в момент времени T. |
| L(T) | Интенсивность (условная интенсивность отказа) в момент времени T. Ожидаемое число отказов в единицу времени (обычно час), упри условии что система находится в работоспособном состоянии непосредственно перед временем T. |
| E(0,T) | Ожидаемое число отказов на интервале 0 - T |

| | |
|-------------|-----------------------------------------------------------------------------------------|
| F(T) | Вероятность по крайней мере одного отказа системы (верхнего события) на интервале 0 - T |
|-------------|-----------------------------------------------------------------------------------------|

Вычисление неготовности

Все вычисления неготовности для основных событий, событий ООП, модулей и МСО выполняются в соответствии с описанными ранее правилами. Квантификация МСО всегда делается при использовании *аппроксимации минимальной верхней границы сечения (первого порядка)*. Все вычисления выполняются для каждого из заданных моментов времени.

Вычисление частоты (безусловной интенсивности отказов)

Вычисление частоты (безусловной интенсивности отказов) $W(T)$ делается в той же последовательности, что и вычисление неготовности:

Вычисляется $W(T)$ для всех основных событий

Вычисляется $W(T)$ для всех событий ООП

Вычисляется $W(T)$ для всех модулей

Вычисляется $W(T)$ для всех МСО

Вычисляется $W(T)$ для верхнего события

Формулы для вычисления частоты $W(T)$ были приведены при описании моделей надежности базисных событий.

Вычисление частоты для событий ООП производится по формулам, описанным ранее, но значение Q_{tot} заменяется на W_{tot} , где W_{tot} рассчитывается так же, как определено для основных событий при использовании модели надежности группы ООП.

Как уже говорилось, модуль может содержать в своем составе основные события, события ООП, другие модули или любое сочетание этих элементов. Модули рассчитываются после расчета параметров всех основных событий и событий ООП последовательно в восходящем порядке. Модуляризация делается таким образом, что основные события моделью типа «частота» (тип 5) не могут быть частью модулей, так что модульные вычисления не должны иметь дело с такими событиями.

Вычисление частоты для модулей делается по следующим формулам:

- частота модуля ИЛИ (безусловная интенсивность отказов):

$$W_{mod}(T) = \sum_{i=1}^k W_i(T) \prod_{\substack{j=1 \\ j \neq i}}^k (1 - Q_j(T)) \quad (62)$$

частота модуля И (безусловная интенсивность отказов):

$$W_{mod}(T) = \sum_{i=1}^k W_i(T) \prod_{\substack{j=1 \\ j \neq i}}^k Q_j(T) \quad (63)$$

где:

- $Q_i(T)$ - неготовность в момент времени T входа i (основное событие, событие ООП или другой модуль) в модуль
- $W_i(T)$ - частота в момент времени T входа i (основное событие, событие ООП или другой модуль) в модуль

Частота набора МСО рассчитывается после расчета параметров основных событий, событий ООП и модулей с помощью следующего выражения:

$$W_{MCS}(T) = \sum_{i=1}^k W_i(T) \prod_{\substack{j=1 \\ j \neq i}}^k Q_j(T) \quad (64)$$

где:

- $Q_i(T)$ - неготовность в момент времени T элемента i в МСО (i - базисное событие, событие ООП или другой модуль)
- $W_i(T)$ - частота в момент времени T элемента i в МСО
- k - число элементов в МСО

Вычисление результатов верхнего события

Результаты верхнего события рассчитываются на основе результатов, полученных после вычислений неготовности и частоты для МСО.

Неготовность в момент времени T для верхнего события рассчитывается согласно *аппроксимации минимальной верхней границы сечения* по формуле:

$$Q_{TOP}(T) = 1 - \prod_{i=1}^n (1 - Q_{MCS,i}(T)) \quad (65)$$

Частота (безусловная интенсивность отказа) для верхнего события - ожидаемое число отказов в единицу времени в период $T + dt$ рассчитывается по формуле:

$$W_{TOP}(T) = \sum_{i=1}^k W_{MCS,i}(T) \prod_{\substack{j=1 \\ j \neq i}}^k (1 - Q_{MCS,j}(T)) \quad (66)$$

где:

- $Q_{MCS,i}(T)$ - неготовность набора минимальных сечений i в момент времени T
- $W_{MCS,i}(T)$ - частота набора минимальных сечений i в момент времени T
- n - число наборов минимальных сечений

Интенсивность (условная интенсивность отказа) для верхнего события - ожидаемое число отказов в единицу времени в период $T + dt$ при условии, что непосредственно перед моментом времени T отказа нет рассчитывается по формуле:

$$L_{TOP}(T) = \frac{W_{TOP}(T)}{1 - Q_{TOP}(T)} \quad (67)$$

Ожидаемое число отказов на интервале от 0 до T получается интегрированием $W(T)$:

$$E_{TOP}(0, T) = \int_0^T W_{TOP}(t) dt \quad (68)$$

Интегрирование при вычислении $E_{TOP}(0, T)$ выполняется численно путем использования линейной аппроксимации между моментами времени, представленными в распечатке.

Вероятность по крайней мере одного отказа (ненадежность) на интервале от 0 до T рассчитывается по формуле:

$$F_{TOP}(T) = 1 - \exp\left(-\int_0^T L_{TOP}(t) dt\right) \quad (69)$$

Интегрирование при вычислении $F_{TOP}(0, T)$ выполняется численно путем использования линейной аппроксимации между моментами времени, представленными в распечатке.

Средняя неготовность рассчитывается интегрированием и усреднением по времени $Q(T)$:

$$Q_{TOP, mean} = \frac{1}{T} \int_0^T Q_{TOP}(t) dt \quad (70)$$

Интегрирование при вычислении $Q_{TOP, mean}(0, T)$ выполняется численно путем использования линейной аппроксимации между моментами времени, представленными в распечатке.

Если для верхних событий (системы) предусмотрено восстановление рассчитывается величина MTBF (средняя наработка между отказами):

$$MTBF_{TOP} = \frac{1}{W_{TOP}(t = \infty)} \quad (71)$$

Если для верхних событий (системы) восстановление не предусмотрено, то рассчитывается MTTF (средняя наработка до отказа):

$$MTTF_{TOP} = \int_0^{\infty} (1 - Q_{TOP}(t)) dt \quad (72)$$

Интегрирование в формуле для MTTF выполняется численно путем вычисления $Q_{TOP}(t)$ для большого количества моментов времени с коротким шагом и с линейной аппроксимацией между точками.

MTTR верхнего события (среднее время простоя (ремонта) при возникновении верхнего события) рассчитывается только если предусмотрено восстановление верхнего события (системы). Для этого используется следующая формула:

$$MTTR_{TOP} = \frac{Q_{TOP}(t = \infty)}{W_{TOP}(t = \infty)} \quad (73)$$

Примечание:

Все интегрирования, включенные в вычисление $E_{TOP}(0,T)$, $F_{TOP}(T)$ и $Q_{TOP, mean}$ выполняются численно, используя линейную аппроксимацию между моментами времени. Моменты времени, используемые в этих интегрированиях - те же самые моменты времени, используемые во всех других вычислениях, то есть моменты времени, представленные в распечатке.

Если эти моменты времени далеки друг от друга, то они могут давать очень плохое представление кривой. Численное интегрирование, выполненное программой, будет тогда плохой аппроксимацией.

Обычно рекомендуется использовать автоматический расчет моментов времени для всех случаев, где имеются базисные события 2-го типа (периодически проверяемые), чтобы удостовериться что программа использует моменты времени, дающие хорошее представление кривой. В тех случаях, когда любое из совокупных свойств важно, рекомендуется использовать ручное задание моментов времени со многими моментами времени.

Вычисление ошибки отсечения

Вычисление ошибки отсечения выполняется только при использовании вероятностного отсечения. При применении процедуры отсечения конечный список МСО не является полным отражением логики дерева отказов. Некоторые из наборов сечений отбрасываются (обрезаются) вследствие отсечения. В результате появляется «ошибка отсечения» (ошибка отбрасывания), связанная с квантификацией наборов сечений. Верхняя граница для этой ошибки отсечения оценивается Risk Spectrum.

Оценка ошибки отсечения делается следующим способом:

- Программа всегда обновляет текущую неготовность для набора сечений, с которым работает в данный момент. В определенной стадии разработки, этот набор сечений может состоять из смеси основных событий, модулей и логических элементов. Неготовность основных событий и модулей известна, тогда как неготовность логических элементов не известна. Текущая неготовность набора сечений рассчитывается как произведение неготовностей основных событий и модулей. Неготовность логических элементов, таким образом, предполагается равной 1. Это конечно консервативно и обычно - очень завышенная оценка.