На правах рукописи

Даниленко Иван Николаевич

Функционально-структурные вероятностные модели в задачах анализа надежности микропроцессорных систем

Специальность 05.13.01 – системный анализ,
управление и обработка информации
(по отраслям: информация и информационные системы)

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Томск – 2004

Работа выполнена в Томском политехническом университете и в Сургутском государственном университете Ханты-Мансийского автономного округа.

Научный руководитель – доктор технических наук, профессор
Цапко Геннадий Павлович.

Официальные оппоненты – доктор технических наук, профессор,
Гончаров Валерий Иванович;
кандидат технических наук,
Казьмин Григорий Павлович.

Ведущая организация – Научно-производственное объединение
Прикладной механики имени академика
М.Ф. Решетнева, г. Железногорск.

Защита диссертации состоится «30» июня 2004 г. в ___ часов на заседании диссертационного совета Д 212.269.06 в Томском политехническом университете по адресу: г. Томск, ул. Советская, 84.

С диссертацией можно ознакомиться в библиотеке Томского политехнического университета.

Автореферат разослан «_____» _____ 2004 г.

Ученый секретарь
диссертационного совета
кандидат технических наук, доцент _____ Сонькин М.А.

Общая характеристика работы

Актуальность проблемы. Широкая автоматизация практически всех сфер человеческой деятельности с использованием вычислительной техники, робототехнических комплексов и высокопроизводительного прецизионного оборудования составляет одно из главных направлений научно-технического прогресса в настоящее время. Фундаментальной базой данной отрасли являются вычислительные средства различного назначения. Многолетний опыт разработки, производства и эксплуатации средств вычислительной техники привел к необходимости рассматривать в неразрывной связи вопросы конструирования средств вычислительной техники и их надежности. В настоящее время общепризнанно, что надежность является сложным неотъемлемым свойством любого промышленного изделия, а для таких устройств, которые участвуют в управлении технологическими процессами, объектами повышенной для человека опасности, дорогостоящими объектами, наземным, водным, воздушным транспортом и космическими аппаратами, надежность является наиважнейшим показателем.

Большинство вычислительных систем, управляющих процессами в особых ситуациях, обладают той или иной избыточностью для уменьшения или избежания последствий сбоев и отказов отдельных составляющих элементов таких систем. Эта избыточность заключается в использовании дополнительных аппаратных или программных средств, помимо функционально необходимых. Такие отказоустойчивые микропроцессорные системы (ОМС) обычно строятся на базе мультипроцессорных систем, состоящих из нескольких процессоров и модулей памяти и взаимодействующих посредством специализированного программно-аппаратного комплекса, обеспечивающего отказоустойчивость.

Создание и анализ такого рода систем представляет собой сложную задачу. В процессе проектирования должны быть отобраны те варианты архитектуры ОМС, которые наиболее полно удовлетворяют заданным критериям, зачастую противоречивым. И, несомненно, одними из наиболее важных параметров ОМС, которые обычно требуется оценить на этапе проектирования, причем с высокой степенью достоверности, являются показатели надежности.

Традиционными методами оценки показателей надежности систем являются комбинаторные методы и модели на основе марковских процессов. Однако эти подходы имеют ряд ограничений, которые сужают область их применения. Имитационное моделирование устраняет эти ограничения и позволяет отразить структурные свойства системы и ряд других более полно, по сравнению с аналитическими моделями. Большинство методов, направленных на решение задач анализа надежности средствами имитационного моделирования, представляют систему на низком уровне абстрагирования и ориентированы, как правило, на верификацию свойства отказоустойчивости ОМС, а не на анализ показателей надежности верхнего уровня, таких как, вероятность безотказной работы, наработка на отказ и т.п. Это подчеркивает необходимость разработки нового подхода к созданию имитационных моделей ОМС, которые бы отражали как процесс функционирования системы, так и ее состояние в смысле работоспособности, и, при этом, позволяли бы исследовать структурные и функциональные аспекты надежности.

Целью работы является: разработка нового функционально-структурного подхода к решению задач анализа надежности отказоустойчивых микропроцессорных систем, учи-

тывающего влияние возникновения отказов и сбоев и распространения их последствий на протекание вычислительного процесса в отказоустойчивой микропроцессорной системе, создание моделей, отражающих функциональные и структурные особенности отказоустойчивых микропроцессорных систем, и разработка программной среды функционально-структурного моделирования для решения задач создания, отладки, эксплуатации моделей отказоустойчивых микропроцессорных систем, проведения модельных экспериментов и обработки их результатов.

Методы исследования. Для решения поставленных задач в диссертационной работе применены методы теории моделирования, теории графов, теории надежности, теории вероятности и математической статистики, теории планирования экспериментов, структурного программирования, объектно-ориентированного проектирования. Верификация имитационных моделей отказоустойчивых микропроцессорных систем, включающих функционально-структурные вероятностные модели процессора, памяти, механизмов повышения отказоустойчивости и модель вычислительного процесса, проводилась экспериментально, с использованием разработанной программной среды имитационного моделирования.

Научная новизна работы:

1. Предложен новый функционально-структурный подход к решению задач анализа надежности вычислительных систем, заключающийся в учете влияния отказов и сбоев на протекание вычислительного процесса и позволяющий получать более адекватные оценки показателей надежности отказоустойчивой микропроцессорной системы на основе воспроизведения процесса функционирования системы и ее структурных и архитектурных особенностей.
2. Разработана оригинальная функционально-структурная имитационная вероятностная модель отказоустойчивой микропроцессорной системы, включающая функционально-структурные модели основных компонентов системы и механизмов повышения отказоустойчивости системы, интерпретация которых осуществляется в контексте модели вычислительного процесса, учитывающей характер его протекания в условиях возникновения отказов и сбоев и распространения их последствий.
3. Создана программная среда функционально-структурного имитационного моделирования, обеспечивающая создание, отладку, модификацию и анализ функционально-структурных моделей отказоустойчивых микропроцессорных систем с целью определения показателей надежности и позволяющая учитывать специфические факторы возникновения отказов и сбоев, распространения последствий этих неисправностей и работы механизмов повышения отказоустойчивости.

Результаты работы, выносимые на защиту

1. Функционально-структурный подход к решению задач анализа надежности вычислительных систем, заключающийся в учете воздействия отказов и сбоев и распространения их последствий на протекание вычислительного процесса.
2. Функционально-структурная имитационная вероятностная модель отказоустойчивой микропроцессорной системы, включающая функционально-структурные модели процессора, памяти и механизмов повышения отказоустойчивости, учитывающие воздействие отказов и сбоев, и модель вычислительного процесса, отражающая

взаимодействие основных компонентов отказоустойчивой системы и распространение последствий неисправностей между ними.

Практическая ценность работы

1. Разработанные функционально-структурные имитационные вероятностные модели основных компонентов отказоустойчивой микропроцессорной системы и механизмов повышения отказоустойчивости, отражающие процесс функционирования и работоспособность системы в условиях возникновения отказов и сбоев и распространения последствий этих неисправностей с учетом их влияния на протекание вычислительного процесса.
2. Разработанная программная среда функционально-структурного имитационного вероятностного моделирования, реализующая основные механизмы выполнения функционально-структурных моделей в составе общей модели отказоустойчивой микропроцессорной системы под управлением базовых средств организации вычислительного эксперимента и обработки его результатов.

Реализация результатов работы. Разработанные функционально-структурные модели и программные средства моделирования используются в Научно-производственном объединении Прикладной механики (НПО ПМ) имени академика М.Ф. Решетнева, г. Железногорск, для анализа показателей надежности отказоустойчивых бортовых комплексов управления (БКУ) космических аппаратов ретрансляции и связи в процессе их проектирования и последующей эксплуатации. Основные результаты диссертационной работы изложены в отчетах по НИР, выполненных институтом «Кибернетический центр» ТПУ в рамках хозяйственных договоров с НПО ПМ г. Железногорска и используются НПО ПМ при создании БКУ космических аппаратов ретрансляции и связи.

Апробация работы. Результаты диссертационной работы докладывались и обсуждались: на Третьем Российско-Корейском международном симпозиуме по науке и технологии KORUS’99, г. Новосибирск, Россия, 1999 г.; на Четвертом Российско-Корейском международном симпозиуме по науке и технологии KORUS’2000, г. Ульсан, Корея, 2000 г.; на Пятом Российско-Корейском международном симпозиуме по науке и технологии KORUS’2001, г. Томск, Россия, 2001 г.; Третьей окружной конференции молодых ученых Ханты-Мансийского автономного округа «Наука и инновации Ханты-Мансийского автономного округа», г. Сургут, 2002 г.

Публикации. По теме диссертационной работы опубликовано 15 статей и тезисов докладов, отчет о НИР.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав и заключения, изложенных на 114 страницах машинописного текста, включает 27 рисунков, 15 таблиц, а также содержит список литературы из 118 наименований и 3 приложения. Общий объем работы – 145 страниц.

Содержание работы

Во введении приводится обоснование актуальности темы, формулируется цель работы, указывается научная новизна и практическая ценность полученных результатов, описывается основное содержание работы.

В первой главе рассмотрены способы и средства реализации отказоустойчивости микропроцессорных систем, приводящие к существенному повышению сложности таких

систем вследствие усложнения структуры, алгоритмического и программного обеспечения. В то же время, именно для отказоустойчивых систем свойство надежности является одним из наиболее важных, поэтому оценка показателей надежности востребована на всех этапах проектирования аппаратного, алгоритмического и программного обеспечения.

Методы повышения надежности современных вычислительных систем по способу реализации отказоустойчивости делятся на две группы. Первый способ ориентирован на использование средств контроля, диагностирования, реконфигурации и восстановления вычислительного процесса (ВП) для реализации активной отказоустойчивости и опирается, по большей мере, на программные решения. Второй использует средства маскирования неисправностей для реализации пассивной отказоустойчивости при помощи аппаратных средств. В любом случае, независимо от вида вводимой избыточности – структурной, временной, алгоритмической или программной, – контроль, диагностирование и реконфигурация отказоустойчивой системы требует дополнительных ресурсов и приводит к усложнению. Наиболее распространенным видом избыточности является аппаратная избыточность, которая нередко является основой для использования ее других видов, например, использование *N*-версионного резервирования программного обеспечения требует дополнительной памяти для размещения версий. В силу специфических особенностей, время, необходимое для парирования неисправностей в системах, использующих активную отказоустойчивость, как правило, больше, чем в системах, основанных на пассивной, что позволяет выделить области применения отказоустойчивых вычислительных систем. Активная отказоустойчивость чаще находит применение в информационных системах, а пассивная – в системах управления, в том числе и в системах управления реального времени, для которых обеспечение безостановочной работы вычислительной системы является одним из важнейших требований.

На основе проведенного анализа существующих ОМС, в архитектуре отказоустойчивых систем, использующих как активный, так и пассивный способы реализации отказоустойчивости, выделено вычислительное ядро – мультипроцессорная система, – характерное для большинства подобных систем, в состав которого входит множество процессоров, множество модулей памяти и множество механизмов повышения отказоустойчивости (мажоритарные элементы, сторожевые таймеры, схемы сравнения). Акцентирование внимания на этой части отказоустойчивых вычислительных систем не случайно, ведь она является, с одной стороны, основным вычислительным ядром, и, с другой, представляет собой наиболее сложную ее часть с точки зрения количества элементарных логических элементов и организации их взаимодействия. Далее, под отказоустойчивой микропроцессорной системой будет пониматься та ее часть, которая представляет собой ее ядро.

Решение задач анализа надежности вычислительных систем, при сохранении общих черт, свойственных теории надежности в целом, имеет некоторые специфические особенности, обусловленные природой этих систем, структурными и информационными взаимосвязями компонентов. В первую очередь, это существенное влияние перемежающихся отказов, называемых сбоями, на адекватное протекание ВП и надежность вычислительной системы. Второй особенностью является наличие типовой схемы возникновения и обнаружения неисправности, ее парирования и восстановления вычислительного процесса. Каждый этап этого процесса имеет определенную длительность, которая может функционально зависеть от структуры и параметров системы или быть случайной величиной, рас-

пределенной по некоторому закону, но и в том, и в другом случае, она оказывает влияние на показатели надежности. Еще одним немаловажным фактором является вопрос о выборе математической модели, характеризующей зависимость надежности от времени. Несмотря на широкое распространение экспоненциального распределения в моделировании надежности вычислительных систем, у специалистов в теории надежности нет единого мнения о правомерности его использования. Таким образом, задача анализа надежности отказоустойчивых систем с учетом всех особенностей их функционирования представляет собой сложную задачу.

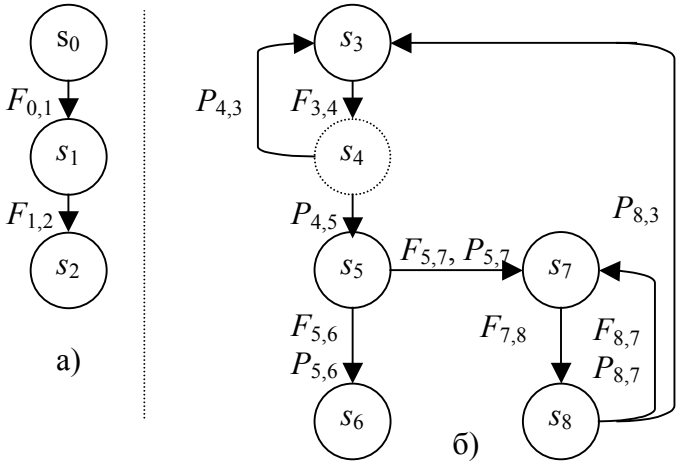
Для решения этой задачи применяются методы аналитического и имитационного моделирования. Аналитические методы характеризуются выводом аналитических зависимостей, связывающих показатели надежности с факторами, под влиянием которых формируются эти показатели. С развитием средств вычислительной техники широкое применение стал получать метод статистического моделирования и его модификации, так называемые аналитико-статистические методы, направленные на получение более достоверных результатов и ускорение статистических испытаний. Суть статистического моделирования заключается в формировании некоторой случайной величины, математическое ожидание которой является решением задачи, и, путем многократного проведения вычислительного эксперимента и обработки его результатов, отыскании оценки этого математического ожидания, являющегося приближенным решением задачи. Указанные методы направлены на получение оценок показателей надежности верхнего уровня (например, вероятность безотказной работы, среднее время наработки на отказ и т.д.) и рассматривают компоненты вычислительной системы на уровне «черных ящиков», предполагая два возможных состояния – исправное и неисправное. Метод имитационного моделирования близок по своей идее к методу статистического моделирования, но основной акцент при построении моделей производится на отражении процесса функционирования системы с выбранным уровнем абстрагирования. Такой метод получил широкое развитие в задачах оценки показателей надежности низкого уровня (например, длительность интервала проявления неисправности, область распространения ошибки и т.п.) и задачах тестирования и верификации алгоритмических, программных и аппаратных решений, используемых в отказоустойчивых микропроцессорных системах. В известных средствах, основанных на методе имитационного моделирования, используется физический или логический уровень представления микропроцессорной системы.

В заключение главы делается вывод о том, что для решения задач анализа отказоустойчивых микропроцессорных систем необходимо создание нового подхода к построению моделей, который позволит представлять мультипроцессорную систему на функциональном уровне – уровне основных информационных потоков, учитывать влияние неисправностей и последствий их распространения на состояние вычислительного процесса и оценивать показатели надежности различного уровня.

Во второй главе для решения поставленных задач предложен новый подход к построению моделей, основанный на представлении системы в виде иерархической структуры элементов. Средства, входящие в состав ОМС, могут быть представлены в виде множества элементов, поведение каждого из которых задано в виде графа состояний. Каждый переход характеризуется правилом перехода, включающем вероятность перехода и время перехода из состояния в состояние, которые, в общем случае, определяются текущим со-

стоянием системы и другими ее характеристиками и параметрами. Длительность перехода из состояния в состояние зависит от параметров системы, ее текущего состояния и времени и определяется либо некоторой функциональной зависимостью, либо является случайной величиной, распределенной по некоторому закону. Вектор состояния системы определяется текущим состоянием каждого элемента. Поведение системы отражается последовательностью пар, включающих вектор состояния системы и момент времени его наступления. При этом, можно определить подмножество значений вектора состояния, соответствующих неисправному состоянию. В этом случае, задача анализа надежности системы сводится к анализу последовательности смены состояний и оценке ее вероятностных характеристик на основе принадлежности текущего значения вектора состояния системы подмножеству значений, характеризующих неисправное состояние. Большинство методов анализа и расчета надежности основываются на следующих допущениях: элемент системы может находиться в двух состояниях – исправном и неисправном; элементы независимы; экспоненциальный характер наработки на отказ. Однако эти допущения при анализе отказоустойчивых микропроцессорных систем не всегда допустимы. Отказ от указанных выше допущений и соответствующая модификация модели анализа надежности вычислительной системы приводит к тому, что существующие аналитические методы становится практически не приемлемыми. Имитационные модели, построенные в соответствии с излагаемым подходом, должны отражать не только процесс функционирования, но и модель работоспособности, которая определяется, в первую очередь, структурой информационных взаимосвязей элементов, образующих систему. Функционально-структурные модели, основанные на данных принципах, учитывающие как факторы, вытекающие из архитектуры системы, так и характеристики ее основных информационных потоков, и, одновременно, воспроизводящие процесс возникновения и, возможно, устранения неисправностей, позволят получить более точную оценку показателей надежности отказоустойчивых микропроцессорных систем.

Функционально-структурная модель процессора, представленная на рис. 1, включает две подмодели, учитывающие влияние отказов и сбоев. Распределение $F_{0,1}$ характеризует наработку на отказ процессора. Подмодель отказов, в отличие от типовой модели, включает состояние «скрытый отказ», так как неисправность, после ее возникновения в процессоре, на некотором интервале времени не проявляется.



Обозначение	Состояние
s_0	исправное
s_1	скрытый отказ
s_2	отказ
s_3	исправное
s_4	потенциальный сбой
s_5	скрытый сбой
s_6	нарушение ВП
s_7	активный сбой
s_8	пассивный сбой

Рис. 1. Функционально-структурная модель процессора:
а – подмодель отказов; б – подмодель сбоев.

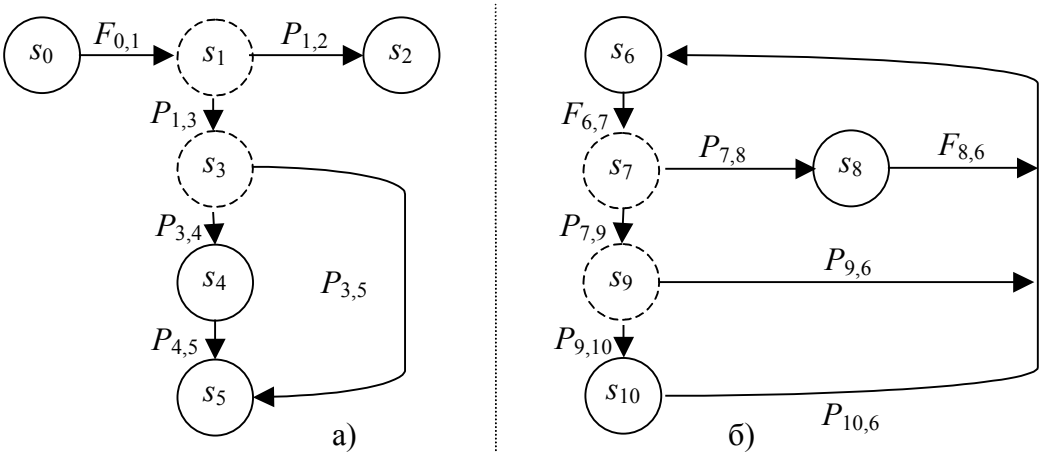
Подмодель сбоев включает ряд дополнительных состояний, учитывающих специфический характер этих неисправностей в том смысле, что время их существования невелико. Так же, как и отказы, сбои могут некоторое время находиться в скрытом состоянии и, в течение некоторого интервала времени, не оказывать влияния на протекание вычислительного процесса, более того, сбой, после возникновения, может устраниться прежде, чем он окажет какое-либо влияние на работу вычислительной системы. Данная ситуация моделируется путем включения в модель временного состояния – «потенциальный сбой» (состояние s_4 на рис. 1б). Из этого состояния процессор может перейти либо в исходное состояние (отсутствие неисправностей), либо в состояние «скрытый сбой», которое представляет ситуацию, когда возникший сбой окажет влияние на протекание вычислительного процесса, но только через некоторый интервал времени – интервал проявления. Из состояния «скрытый сбой» процессор может перейти либо в состояние «нарушение ВП», либо в состояние «активный сбой». Из состояния «нарушение ВП» процессор может выйти только после рестарта. Нахождение в состояние «активный сбой» приводит к искажению результатов выполнения на выходе и/или обращению к несоответствующей области памяти.

Из состояния «активный сбой» в течение одного такта процессор переходит в состояние «пассивный сбой», из которого он может либо вернуться в исправное состояние, либо в состояние «активный сбой». Переход из состояния «пассивный сбой» в состояние «активный сбой» отражает тот факт, что проявление одного сбоя на аппаратном уровне может повлечь за собой отклонение вычислительного процесса от нормального на протяжении некоторого интервала времени.

Функционально-структурная модель памяти так же, как и функционально-структурная модель процессора, включает две подмодели – отказов и сбоев (рис. 2). Особенность функционирования данного компонента микропроцессорной системы позволяет выделить две характерных области воздействия неисправностей – схему управления (СУ) и массив ячеек памяти (ЯП). Соотношение указанных видов неисправностей определяется соотношением количества элементарных логических элементов, реализующих соответствующие схемы памяти.

Исправная в исходном состоянии память под воздействием отказа переходит в состояние «потенциальный отказ», которое является промежуточным, из которого она переходит либо в состояние «отказ СУ», либо в промежуточное состояние «потенциальный отказ ЯП». В первом случае, последствия отказа проявляются незамедлительно и выражаются в том, что операции чтения/записи осуществляются по произвольным адресам. В состоянии «отказ СУ», память, при выполнении операций чтения, выдает произвольно искаженные слова данных (отличающиеся от правильных в произвольном количестве бит), а при выполнении операций записи, запись производится по произвольным (случайным) адресам, что приводит к искажению находящихся по этим адресам данных.

Развитие последствий неисправности во втором случае определяется следующим фактором. Так как отдельные ячейки памяти хранят двоичные значения – «0» или «1», то последствия отказа проявятся лишь в случае искажения исходного значения, хранящегося в ячейке, т.е. в случае, когда результатом чтения неисправной ЯП является неверное значение. Поэтому память будет находиться в состоянии «скрытый отказ ЯП», а последствия отказа скажутся либо незамедлительно и память перейдет в состояние «активный отказ ЯП», либо через некоторый интервал времени, когда произойдет перезапись содержимого



Обозначение	Состояние	Обозначение	Состояние
s_0	исправное	s_6	исправное
s_1	потенциальный отказ	s_7	потенциальный сбой
s_2	отказ СУ	s_8	сбой СУ
s_3	потенциальный отказ ЯП	s_9	потенциальный сбой ЯП
s_4	скрытый отказ ЯП	s_{10}	активный сбой ЯП
s_5	активный отказ ЯП		

Рис. 2. Функционально-структурная модель памяти:
а – подмодель отказов; б – подмодель сбоев.

отказавшей ячейки новым значением, несовпадающим (с некоторой вероятностью) со значением свойственным отказавшей ячейке, в результате чего состояние памяти изменится на «активный отказ ЯП». Следует отметить, что предлагаемая модель предполагает, что отказ в массиве ячеек не приводит к отказу всего модуля памяти, а сказывается лишь на достоверности информации, хранящейся в отказавших ячейках.

Подмодель сбоев также учитывает специфику функционирования памяти и кратковременный характер проявления этого вида неисправностей. В частности, из состояния «активный сбой СУ» память возвращается в исправное состояние уже при следующем обращении к ней, а из состояния «потенциальный сбой ЯП», память может перейти в исправное состояние, что моделирует ситуацию, когда сбой устраняется прежде, чем его последствия окажут влияние на содержимое ЯП. Кроме того, из состояния «активный сбой ЯП» память выходит в момент перезаписи соответствующей ЯП.

В рассматриваемом классе ОМС к механизмам повышения отказоустойчивости относят однонаправленные и двунаправленные мажоритарные элементы, компараторы, мультиплексоры, сторожевые таймеры и схемы контроля ошибок. В силу того, что эти элементы являются относительно простыми, соответствующая функционально-структурная модель включает небольшое количество состояний, однако и в этой модели учитываются как сбои, так и отказы. Использование данной модели предполагает, что сбой в работе механизма повышения отказоустойчивости приводит к искажению одного разряда в слове данных, а отказ влечет искажение всего слова данных.

Разработанная модель вычислительного процесса характеризует аспекты распределения обращений к памяти в пространстве и во времени. В формировании этой модели играют роль такие факторы и параметры ОМС, как локальность обращений к памяти, количество инструкций, вероятности чтения и записи при выполнении данной инструкции и

вероятность безуспешного обращения к кэш-памяти. Свойство локальности объясняется существующими технологиями разработки программного обеспечения и принципами структуризации данных. Временная локальность означает, что к ячейкам памяти, к которым недавно производилось обращение, с большой степенью вероятности будет обращение в ближайшем будущем. Пространственная локальность, означает, что обращения к памяти концентрируются, как правило, таким образом, что в случае обращения к некоторой ячейке памяти с большой вероятностью можно ожидать обращения к близлежащим ячейкам. В качестве модели распределения обращений к памяти выбрана модель, основанная на кусочно-линейной аппроксимации распределения Брэдфорда-Зипфа, которое адекватно отражает типовой вычислительный процесс. Выбранная модель локальности допускает эффективную реализацию в вычислительном эксперименте, так как не требует моделирования выполнения каждой инструкции. Для учета влияния и распространения последствий неисправностей в модели учитываются только те инструкции, на момент выполнения которых либо процессор находился в неисправном состоянии, либо из памяти считывается искаженное значение. Для учета последствий считывания искаженных значений из памяти, функционально-структурная модель процессора дополнена состояниями, отражающими возможность нарушения нормального протекания вычислительного процесса и аналогичными тем, что отражают развитие ситуации в случае возникновения сбоя.

Модель отказоустойчивой системы объединяет функционально-структурные модели процессора, памяти, механизмов повышения отказоустойчивости и модель ВП. Функционирование механизмов повышения отказоустойчивости имитируется с помощью программного модуля, реализованного в составе средств функционально-структурного моделирования.

Вычислительный эксперимент организуется путем имитирования поведения вычислительной системы и корректности протекания вычислительного процесса. При этом, исправная в исходном состоянии система функционирует до наступления отказа или до наступления времени окончания эксперимента. Количество опытов определяется требуемым значением доверительных интервалов исследуемых оценок показателей надежности системы. Время окончания эксперимента соответствует максимальному времени, в течение которого функционирование реальной системы имеет смысл, поэтому те опыты, в которых система не отказала к этому моменту времени, прекращаются. В ходе вычислительного эксперимента накапливаются статистические данные, необходимые для расчета выбранных показателей надежности.

Таким образом, разработанный подход, на основе функционально-структурных моделей процессора, памяти, механизмов повышения отказоустойчивости и модели вычислительного процесса, позволяет исследовать отказоустойчивые системы, использующие различные принципы резервирования и средства обеспечения отказоустойчивости, учитывая воздействие сбоев, отказов и последствия их распространения на протекание вычислительного процесса.

В третьей главе рассмотрены проектирование и реализация программной среды функционально-структурного моделирования (ПСФСМ). Исходя из назначения ПСФСМ, были выделены ее основные функции: реализация функционально-структурных моделей (процессора, памяти, механизмов повышения отказоустойчивости); реализация модели вычислительного процесса; имитация функциональности механизмов повышения отказо-

устойчивости; настройка параметров моделей отдельных составляющих, модели в целом и проводимого эксперимента; расчет статистических оценок выбранных показателей надежности; визуализация результатов моделирования. В ходе проведенного анализа возможных способов реализации ПСФСМ, было решено выполнить разработку среды на языке высокого уровня в силу того, что универсальные среды имитационного моделирования имеют, как правило, широкую направленность, и, в силу своей универсальности, не всегда эффективны при разработке и исследовании специализированных моделей, ориентированных на определенную предметную область. Другим немаловажным фактором, определившим выбора языка программирования высокого уровня, послужил тот факт, что такая программная реализация ПСФСМ позволяет обеспечить максимальную независимость этой среды от стороннего программного обеспечения.

В соответствии с существующими технологиями разработки программного обеспечения, было решено использовать объектно-ориентированный подход к проектированию ПСФСМ, объединяющий в себе процесс объектно-ориентированной декомпозиции, визуальную нотацию для описания логических, физических, статических и динамических представлений проектируемой системы. В качестве средства визуального описания проектных решений использовался унифицированный язык моделирования – Unified Modeling Language (UML). На рис. 3 приведена диаграмма классов ПСФСМ (внутреннее содержание классов не отображено, чтобы не загромождать диаграмму). Классы элементов моделируемой системы являются потомками классов TComplexObject и TSimpleObject, которые, вместе с классом-предком TRObject образуют так называемый паттерн «компоновщик». Такая структура позволяет классам-клиентам TSimulation и TEvent не различать простые и составные объекты и манипулировать ими по единым правилам. При этом класс TRObject объявляет интерфейс для простых и составных объектов, предоставляет реализацию операций по умолчанию, общую для всех классов потомков. Простые классы (потомки класса

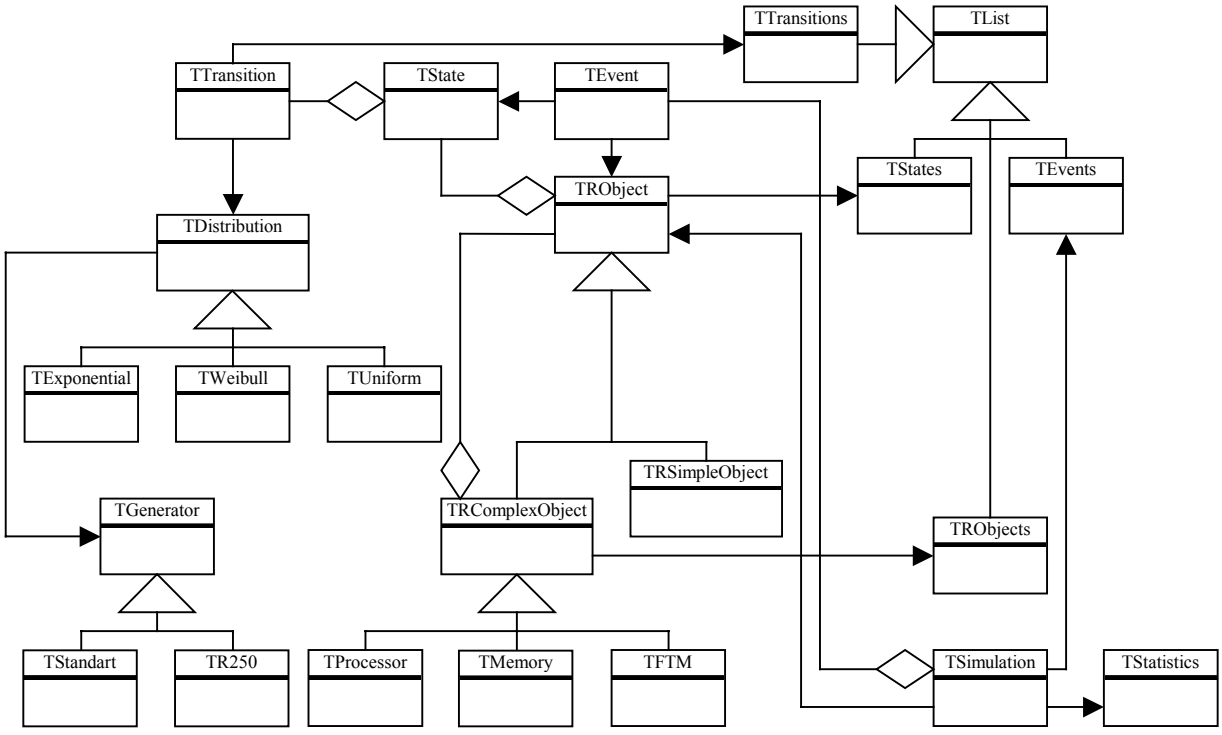


Рис. 3. Диаграмма классов ПСФСМ.

TSimpleObject) определяют поведение примитивных объектов, при этом состояние соответствующих моделируемых подобъектов однозначно определяется единственным текущим состоянием. Класс TComplexObject определяет поведение составных объектов, обеспечивает хранение подобъектов, входящих в его состав, и реализует относящиеся к управлению подобъектами операции интерфейса класса TRObject. Выполнение функций управления составным объектом осуществляется путем делегирования вызова каждому подобъекту, входящему в состав сложного, в независимости от того, является ли он простым или составным. Собственно хранение осуществляется с использованием экземпляра вспомогательного контейнерного класса, управление которым осуществляется классом TComplexObject.

Таким образом, классы-потомки TSimpleObject инкапсулируют поведение объектов, характеризующихся единственным текущим состоянием, а множество всех возможных его состояний агрегатирует контейнер TStates. Для составных объектов TStates представляет собой вектор текущих состояний всех подобъектов, входящих в его состав.

Для хранения в объекте ссылки на своего владельца предназначено поле Owner. Наличие такой ссылки позволяет упростить управление иерархической структурой объектов. Кроме того, ссылка на владельца помогает реализовать последовательное делегирование обязанностей в рамках так называемого паттерна «цепочка обязанностей». Поле Owner определено на уровне класса TRObject и наследуется всеми его потомками.

Класс TSimulation обеспечивает интерпретацию модели и непосредственно связан с классами, которые представляют моделируемую систему (TRObject), события (TEvent) и наблюдателя TStatistics, выполняющего роль обработчика статистики. Не смотря на то, что класс TSimulation ссылается на класс TRObject, на уровне объектов эта связь ориентирована на потомков класса TComplexObject, т.е., в конечном счете, на объект класса, представляющего моделируемую систему.

Последовательность изменения состояний объектов формируется в процессе интерпретации модели путем сбора возможных событий (моментов смены состояния и связанных с ними действий в рамках моделируемой системы), упорядочивания их по времени наступления и последующей выборкой и обработкой. Каждое событие несет информацию о моменте своего создания и интервале времени между моментами создания и перехода объекта в новое состояние. Таким образом, класс TEvent инкапсулирует всю необходимую информацию для осуществления перевода некоторого объекта из одного состояния в другое. В целях разграничения функциональных обязанностей, сбор и обработка статистики сосредоточены в классе TStatistics. Это позволяет его потомкам, унаследовав базовые методы, дополнить их какими-либо специфическими, связанными с обработкой результатов моделирования конкретной системы.

Класс TState отвечает за представление текущего состояния объекта, а также, с помощью класса TTransition, осуществляет представление графа состояний и переходов. Структура графа состояний формируется путем заполнения контейнера, входящего в состав класса TState, объектами переходов. Наличие переходов, их вид и параметры определяются предметной областью. Класс TTransition инкапсулирует такие параметры перехода, как ссылку на состояние, в которое осуществляется переход, вероятность этого перехода и его длительность, которая характеризуется распределением, заданным как экземпляр одного из классов-потомков TDistribution. Делегирование полномочий по формированию

длительности интервала перехода классу TDistribution и использование полиморфизма позволяют легко изменять вид и параметры законов распределения, которые могут быть как стохастическими, так и полностью детерминированными, а также могут представлять собой функцию от переменных модели или внешних переменных.

Известно, что в вычислительных экспериментах используются программные средства (так называемые генераторы) формирования псевдослучайных последовательностей, основанные на различных алгоритмах. Эти генераторы, как правило, формируют целочисленные значения, распределенные равномерно в некотором диапазоне, размер которого зависит от конкретной реализации. Для того чтобы иметь возможность использования в вычислительном эксперименте различных базовых генераторов – стандартных библиотечных и других, без каких-либо изменений в остальных частях рассматриваемых программных средств, класс TDistribution делегирует функцию формирования равномерно распределенной случайной величины классу TGenerator, что позволяет реализовать в виде производных от него классов различные генераторы псевдослучайных последовательностей. Ссылка на объект класса TGenerator будет унаследована экземплярами потомков TDistribution.

В целом, процесс интерпретации модели выглядит следующим образом (рис. 4). Для заполнения очереди событий экземпляр класса TSimulation обращается к объекту, представляющему моделируемую систему, с запросом CreateEvent. Если объект является составным, он делегирует запрос вниз по иерархии вплоть до передачи этого запроса на уровень простых объектов. Простой объект для создания события обращается к текущему состоянию TState с запросом CreateEvent. Для создания события, несущего информацию о состоянии, в которое должен быть осуществлен переход, и длительности перехода, производится вызов GetTransition, возвращающий необходимый объект перехода. Полученный объект TTransition вместе со ссылкой на объект-инициатор создания события передается в конструктор класса TEvent. Получив экземпляр класса TEvent, простой объект обращается к TSimulation, чтобы занести это событие в очередь, причем TSimulation фиксирует этот момент модельного времени в данных объекта TEvent.

В соответствии с реализуемым алгоритмом, TSimulation выбирает ближайшее по мере наступления событие и обращается к нему для обработки с вызовом Process. В свою очередь, событие обращается к объекту, состояние которого надо изменить, с запросом ChangeState и передает в качестве аргумента ссылку на себя. Так как управляемым объектом всегда является простой, он изменяет свое состояние и уведомляет своего владельца об этом изменении и передает информацию о предыдущем и новом состояниях посредством метода UpdateState. В свою очередь объект-владелец может уведомить своего владельца, если таковой имеется. Таким образом, информация об изменениях состояний на нижнем уровне иерархии объектов передается вверх, вплоть до корневого объекта. Так как источником новых событий может являться только объект, состояние которого изменилось последним, то объект события обращается к нему с запросом CreateEvent, и создание события повторяется по описанной ранее схеме. Процесс интерпретации модели может повторяться до тех пор, пока не будет достигнуто некоторое целевое состояние системы или некоторое значение модельного времени.

На основе полученных проектных решений были произведены кодирование ПСФСМ с использованием объектно-ориентированного языка C++ и ее последующая отладка. При

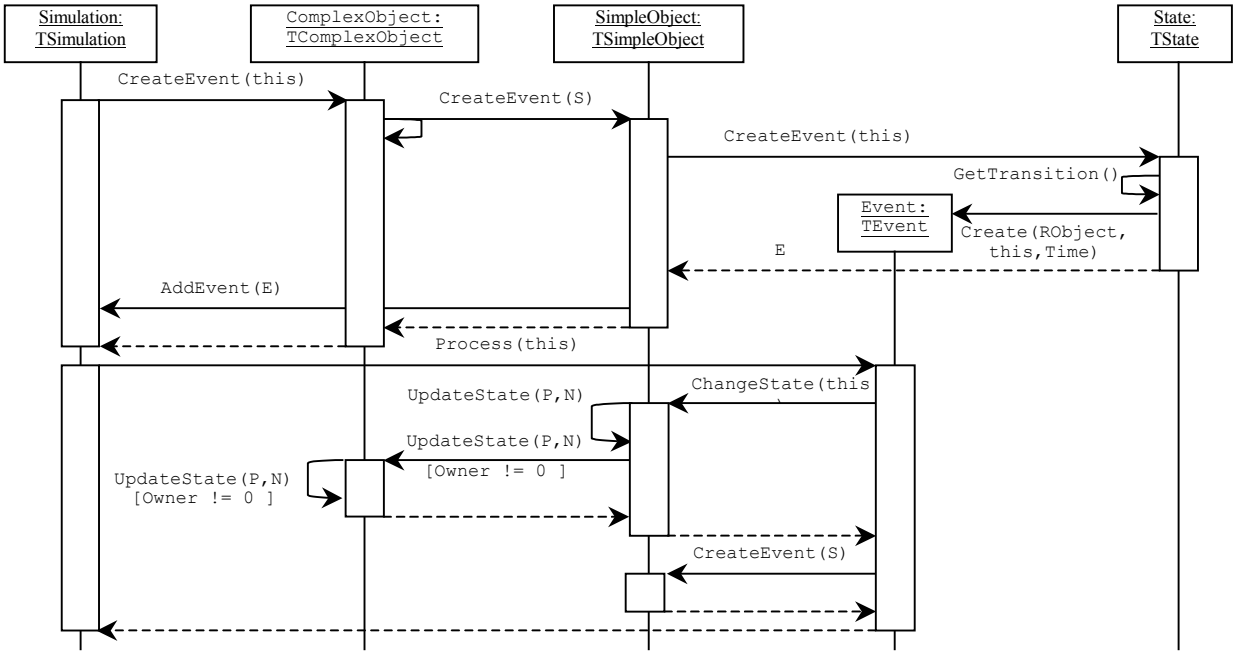


Рис. 4. Диаграмма взаимодействия: последовательность вызовов при передаче сообщений вниз и вверх по иерархии объектов, представляющих систему (S – объект TSimulation; P, S – объекты-состояния: предыдущее (previous) и следующее (next); E – объект-событие).

реализации программной среды было решено использовать механизм параметризованных классов для создания контейнерных классов, используемых для хранения коллекций целевых объектов. Другим интересным решением является обеспечение уникальности экземпляра класса TGenerator полученное с помощью, так называемого, паттерна «одиночка».

В заключительной части третей главы рассматривается вопрос верификации разработанной среды функционально-структурного моделирования. Проверка адекватности результатов, полученных в ходе функционально-структурного моделирования троированной системы, т.е. системы, состоящей из трехкратно резервированных процессоров, модулей памяти и мажоритарного элемента, проводилась путем сравнения со значениями, полученными по аналитическим выражениям, и показала, что результаты, полученные с помощью ПСФСМ, хорошо согласуются с результатами аналитических решений. В таблице приведены значения вероятности безотказной системы на момент времени $t_0 = 50$ [ч] (количество экспериментов $N_0 = 16 \cdot 10^6$, λ_P и λ_M – интенсивности отказов процессоров и памяти соответственно). Значения параметров функционально-структурных моделей были заданы таким образом, чтобы соответствовать аналитическим выражениям.

В четвертой главе рассматриваются задачи исследования вариантов построения ОМС с использованием средств функционально-структурного моделирования, демонстрирующие возможности предложенного подхода. Использование мажорирования данных в качестве механизма маскирования неисправностей приводит к появлению дополнительных задержек, связанных с распространением сигнала через схемы мажорирования и с синхронизацией данных, поступающих от резервированных устройств, что влечет существенное снижение производительности такого рода систем. Одним из способов повышения производительности, за счет снижения надежности, является использование однонаправленного голосования, т.е. мажорирования данных либо только при чтении из памяти, либо только при записи. Аналитические модели в обоих режимах однонаправленного голосования да-

Вероятность безотказной работы троированной системы на момент времени $t_0 = 50$ [ч]

Интенсивность отказов		Вероятность P		Доверительный интервал	
λ_P, [1/ч]	λ_M, [1/ч]	аналитическое решение	ПСФСМ		
10^{-3}	10^{-3}	0,986240	0,986204	0,986147	0,986261
10^{-3}	10^{-4}	0,993022	0,993020	0,992979	0,993061
10^{-4}	10^{-4}	0,999851	0,999850	0,999844	0,999856

ют идентичный результат. Проведенное функционально-структурное моделирование систем с двунаправленным и однонаправленным мажорированием данных при чтении из памяти и записи в память позволило сделать следующие выводы:

- последствия сбоев оказывают существенное влияние на надежность системы, уменьшить которое можно путем рестарта процессоров и восстановлением ВП;
- в случае, когда интенсивность отказов памяти выше, чем интенсивность отказов процессоров, использование кэш-памяти с прямой записью позволяет добиться более высокой надежности, чем использование кэш-памяти с обратной записью;
- распространение последствий неисправностей оказывает более существенное воздействие на системы, использующие голосование при записи данных в память, чем на системы, осуществляющие мажорирование при выборке данных из памяти;
- выигрыш в производительности за счет снижения надежности при использовании систем с однонаправленным голосованием более эффективен в случае, когда интенсивности отказов процессоров и памяти существенно отличаются;
- если интенсивность отказов памяти превышает интенсивность отказов процессоров, то большей надежности, в системах с однонаправленным голосованием, можно добиться при использовании мажорирования при чтении данных из памяти.

Другими важными вопросами, решаемыми при построении отказоустойчивых систем, являются вопросы минимизации массогабаритных и стоимостных показателей системы и уменьшения энергопотребления. В ряде случаев, например в бортовых системах управления космических аппаратов, эти требования могут носить принципиальный характер. В качестве одного из вариантов снижения вводимой в систему избыточности по сравнению с троированной системой, может рассматриваться уменьшение кратности внешнего резервирования памяти. Такой подход обоснован тем фактом, что память наиболее подвержена одиночным сбоям, парирование которых быть выполнено с использованием избыточного кодирования. В качестве исследуемых систем были выбраны следующие варианты архитектуры ОМС: память с исправлением одиночных ошибок без внешнего резервирования; с дублированием и сравнением результатов при чтении из памяти с обнаружением ошибок на основе контроля четности (1, 4, 8 разрядов) и с использованием кода Хэмминга, обнаруживающим две ошибки. Во всех случаях размер слова данных был принят равным 32. Данные, полученные в результате функционально-структурного моделирования рассмотренных вариантов организации работы памяти, позволяют сделать следующие выводы:

- режим работы кэш-памяти оказывает существенное влияние на надежность системы: в случае прямой записи, надежность исследуемых вариантов систем значительно ниже, чем при обратной записи;

- надежность систем с дублированием памяти, использующими 8 разрядов контроля четности и код Хэмминга во всех случаях оказывается выше, чем надежность системы с троированной памятью, причем избыточность таких систем ниже, чем троированной системы в 1,2 раза;
- более высокая интенсивность отказов процессоров нивелирует разницу между показателями надежности системы при различных вариантах повышения отказоустойчивости памяти.

В ходе проведенных исследований проводился не только анализ таких показателей, как вероятность безотказной работы и среднее время наработки на отказ, но и влияние отдельных видов неисправностей на надежность системы. При исследовании различных вариантов повышения надежности памяти, было установлено, что последствия сбоев существенно снижают надежность систем, использующих память без внешнего резервирования или память с дублированием и одним разрядом контроля четности. При использовании этих двух вариантов архитектуры, последствия сбоев приводили к отказу всей системы в 83,7% и 81,2% от общего числа отказов соответственно.

В заключении приведены выводы по диссертации, сформулированы полученные научные и практические результаты.

В приложении к диссертации приведены фрагменты исходных текстов разработанных средств функционально-структурного моделирования, графики вероятности безотказной работы отказоустойчивых микропроцессорных систем с двунаправленным и однонаправленным голосованием и различными методами повышения надежности памяти.

Основные результаты работы

В настоящей диссертации предложен новый функционально-структурный подход к решению задач анализа надежности отказоустойчивых микропроцессорных систем. На основе этого подхода разработана оригинальная модель отказоустойчивой микропроцессорной системы, включающая функционально-структурные имитационные вероятностные модели основных компонентов системы, модели механизмов повышения отказоустойчивости системы, модель вычислительного процесса, а также реализована программная среда имитационного моделирования, позволяющие повысить эффективность решения задач проектирования и анализа микропроцессорных систем, представляющих основу большинства современных систем управления объектами и процессами, требующими повышенной надежности.

Результаты диссертационной работы заключаются в следующем:

1. На основе анализа принципов построения отказоустойчивых микропроцессорных систем показано, что в сложной структуре отказоустойчивых вычислительных систем можно выделить вычислительное ядро, которое присуще большей части отказоустойчивых систем реального времени и представляет собой мультипроцессорную систему, включающую множество процессоров, множество модулей памяти и множество механизмов обеспечения отказоустойчивости.
2. Установлено, что существующие аналитические методы анализа надежности вычислительных систем опираются на ряд допущений, которые не гарантируют адекватность результатов анализа, но могут быть устранены при использовании имитационного мо-

делирования, тенденция к развитию которых характерны для современных исследований в теории надежности вычислительных систем.

3. Предложен новый функционально-структурный подход к решению задач анализа надежности отказоустойчивых микропроцессорных систем, снимающий ограничения, свойственные аналитическим методам, и учитывающий влияние сбоев и отказов и распространения их последствий на протекание вычислительного процесса в отказоустойчивой микропроцессорной системе.
4. Построены, в соответствии с предлагаемым подходом, оригинальные функционально-структурные модели основных компонентов системы, механизмов повышения отказоустойчивости и вычислительного процесса, входящие в состав модели отказоустойчивой микропроцессорной системы.
5. Разработана программная среда функционально-структурного имитационного моделирования, позволяющая производить создание, отладку, модификацию и анализ функционально-структурных моделей отказоустойчивых микропроцессорных систем с целью исследования надежности и позволяющая учитывать специфические факторы возникновения отказов и сбоев, распространения последствий этих неисправностей и работы механизмов повышения отказоустойчивости.
6. Проведен анализ архитектурных решений реализации отказоустойчивости, позволившие более детально, по сравнению с аналитическими моделями, оценить влияние различных факторов, включая процесс возникновения отказов и сбоев, распространения их последствий, средства и способы парирования неисправностей, на показатели надежности исследуемой отказоустойчивой системы.

Результаты диссертации могут быть использованы при анализе показателей надежности отказоустойчивых микропроцессорных систем как комплекса программно-аппаратных средств с учетом различных аспектов реализации как аппаратного, так и программного обеспечения, включая механизмы обеспечения отказоустойчивости. Предложенные модели, дополненные значениями параметров, характеризующих целевое алгоритмическое или программное обеспечение системы, позволяют проводить более детальное изучение надежности конкретных вариантов реализации как проектируемой, так и существующей отказоустойчивой системы и протекающего в ней вычислительного процесса.

Перечень работ по теме диссертации

1. Гришмановский, П.В. Паттерны проектирования в разработке систем имитационного моделирования / П.В. Гришмановский, И.Н. Даниленко // Сб. тр. Вып. 14. – Сургут: Изд-во СурГУ, 2004. – С. 37-45.
2. Гришмановский, П.В. Построение полиморфных структур данных в режиме проектирования средствами визуального проектирования / П.В. Гришмановский, И.Н. Даниленко // Сб. тр. Вып. 13. – Сургут: Изд-во СурГУ, 2003. – С.38-47.
3. Даниленко, И.Н. Алгоритм работы распределенного диспетчера отказоустойчивой бортовой вычислительной системы / И.Н. Даниленко, Е.А. Дмитриева, В.В. Хартов, Г.П. Цапко // Информационные средства и технологии: Сб. материалов междунар. конф. – М: Изд-во МЭИ, 1997. – С.122-126.
4. Даниленко, И.Н. Е-сетевое моделирование как средство определение надежности отказоустойчивых систем / И.Н. Даниленко, Г.П. Цапко // Информационные системы и тех-

нологии: Сб. материалов междунар. науч.-техн. конф. – Новосибирск: НГТУ, 2000. – Т.2. – С.330-333
5. Даниленко, И.Н. Применение Е-сетевого моделирования для анализа и разработки алгоритмов отказоустойчивых компьютерных систем / И.Н. Даниленко, Г.П. Цапко // Управляющие и вычислительные системы. Новые технологии: Сб. материалов межвузовской науч.-техн. конф. – Вологда: Изд-во ВоГТУ, 2000. – С.107-108.
6. Даниленко, И.Н. Принципы программной реализации обеспечения отказоустойчивости бортовых систем управления КА / И.Н. Даниленко, Е.А. Дмитриева, В.В. Хартов // Спутниковые системы связи и навигации: Сб. материалов междунар. конф. – Красноярск: Изд-во КГТУ, 1997.
7. Даниленко, И.Н. Система классов полиморфной коллекции иерархии средств визуального программирования в задачах моделирования и объектно-ориентированного проектирования / И.Н. Даниленко, П.В. Гришмановский // Наука и инновации ХМАО: Сб. тез. докл. Третьей окружной конф. молодых ученых ХМАО / Сургут. гос. ун-т. – Сургут: Изд-во СурГУ, 2002. – С.239-242.
8. Даниленко, И.Н. Функционально-структурные модели в имитационно-статистическом моделировании надежности отказоустойчивых микропроцессорных систем / И.Н. Даниленко // Информационные технологии моделирования и управления: Междунар. сб. науч. тр. – Под ред. О.Я. Кравца. Вып. 15. – Воронеж: Изд-во «Научная книга», 2004. – С. 24-34.
9. Даниленко, И.Н. Функционально-структурный подход к построению вероятностных моделей надежности отказоустойчивых микропроцессорных систем / И.Н. Даниленко // Сб. трудов. Вып. 14. – Сургут: Изд-во СурГУ, 2004. – С. 46-57.
10. Даниленко, И.Н. Энтропийная оценка алгоритмов согласования данных в распределенных системах / И.Н. Даниленко, Г.П. Цапко // Современные проблемы информатизации в непромышленной сфере и экономике: Сб. материалов V междунар. науч. конф. – Воронеж: Центрально-Черноземное книжное изд-во, 2000. – С.138-139
11. Создание новых принципов обеспечения отказоустойчивости БКУ КА навигации: Отчет о НИР (заключительный) / ТПУ; Руководитель Г. П. Цапко; И.Н. Даниленко, Е.А. Дмитриева – ГР № 01990011127; Инв. № 02200003836; – М., 2000. – 67 с.
12. Balakanov, A.R. Simulation Models at the Development of Distributed Data Processing Systems // A.R. Balakanov, I.N. Danilenko, P.V. Grishmanovsky, G.P. Tsapko // KORUS: Proceedings of The 4th Russian-Korean Inter. Symp. on Science and Technology. – Ulsan, 2000. – pp. 56-59.
13. Danilenko, I.N. Dependability Analysis of Fault-Tolerant Multiprocessor Systems by Probabilistic Simulation / I.N. Danilenko, E.A. Dmitrieva, G.P. Tsapko // KORUS: Proceedings of The 5th Russian-Korean Inter. Symp. on Science and Technology. – Tomsk, 2001, – pp.134-136.
14. Danilenko, I.N. Development of Fault-tolerant Computer System Algorithms / I.N. Danilenko, E.A. Dmitrieva, V.V. Khratov, G.P. Tsapko // KORUS: Abstracts of The 3rd Russian-Korean Inter. Symp. on Science and Technology. – Novosibirsk, 1999, – p.443.
15. Danilenko, I.N. Informational Foundation of the Reaching Agreement in Distributed Computing Systems / I.N. Danilenko, E.A. Dmitrieva, G.P. Tsapko // KORUS: Abstracts of The 4th Russian-Korean Inter. Symp. on Science and Technology. – Ulsan, 2000.
16. Danilenko, I.N. Object-Oriented Approach in E-Net Simulation / I.N. Danilenko, P.V. Grishmanovsky, G.P. Tsapko // KORUS: Proceedings of The 3rd Russian-Korean Inter. Symp. on Science and Technology. – Novosibirsk, 1999, – pp.227-230.