

ТЮГАШЕВ Андрей Александрович

**СИНТЕЗ И ВЕРИФИКАЦИЯ
УПРАВЛЯЮЩИХ АЛГОРИТМОВ РЕАЛЬНОГО ВРЕМЕНИ
ДЛЯ БОРТОВЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ КОСМИЧЕСКИХ АППАРАТОВ**

по специальности
05.13.12 – Системы автоматизации проектирования
(в машиностроении)

А В Т О Р Е Ф Е Р А Т
диссертации на соискание ученой степени
доктора технических наук

Работа выполнена на кафедре компьютерных систем ГОУ ВПО «Самарский государственный аэрокосмический университет имени академика С.П. Королева»

Научный консультант

доктор технических наук, профессор

А.А. Калентьев

Официальные оппоненты:

доктор технических наук, профессор Н.А. Северцев

доктор технических наук, профессор Б.Е. Федунев

доктор технических наук, профессор А.Н. Коварцев

Ведущая организация: ФГУП «Научно-производственное объединение имени Лавочкина»

Защита состоится 9 ноября 2007 г. в ____ часов на заседании диссертационного совета Д 212.215.05 в конференц-зале ГОУ ВПО «Самарский государственный аэрокосмический университет имени академика С.П. Королева» по адресу: г. Самара, Московское шоссе, 34.

С диссертацией можно ознакомиться в библиотеке ГОУ ВПО «Самарский государственный аэрокосмический университет имени академика С.П. Королева».

Автореферат разослан «__» _____ 2007 г.

Ученый секретарь диссертационного совета, д.т.н., профессор

А.А. Калентьев

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы

Ключевую роль при управлении современными космическими аппаратами (КА) играют бортовые вычислительные системы (БВС), в состав которых входят одна или несколько бортовых цифровых вычислительных машин (БЦВМ). На них возлагаются задачи контроля работоспособности бортовой аппаратуры (БА), управления движением КА и навигации, выдачи управляющих воздействий на БА при решении КА целевых задач и др. Функции управления реализуются при этом бортовым программным обеспечением (БПО). Расширение спектра выполняемых на борту КА задач привело к созданию сложных структурированных комплексов БПО объемом в сотни тысяч и миллионы команд. В настоящее время создание и отладка БПО является одним из наиболее трудоемких и длительных видов работ среди всех видов деятельности, связанных с созданием КА.

К надежности БПО предъявляются чрезвычайно высокие требования, поскольку ошибки в его функционировании могут приводить к катастрофическим последствиям - потере дорогостоящих КА, срыву важнейших исследований, готовившихся на протяжении нескольких лет международными коллективами ученых. При управлении изделиями космической техники одной из важнейших является также проблема обеспечения безопасности, напрямую связанная с надежностью и качеством БПО.

Понятия качества и надежности программных средств (ПС) определяются международным (ISO 9126:1-4) и российскими (ГОСТ 28195-89, ГОСТ 28806-90) стандартами, где они задаются набором общих и детальных характеристик. Качество ПС определяется как его важнейшее свойство, характеризующееся несколькими группами показателей. При этом надежность рассматривается как одна из важнейших характеристик качества ПС наряду с корректностью (соответствием спецификации), эффективностью (временной эффективностью и использованием ресурсов), сопровождаемостью и переносимостью. Надежность, в свою очередь, определяется «уровнем завершенности (отсутствия ошибок)», устойчивостью и восстанавливаемостью.

Среди ошибок БПО значительное количество приходится на сбои синхронизации и согласования логики управления БА при одновременном функционировании ряда бортовых систем и программ БПО в рамках решения КА целевых задач (ошибки в управляющих алгоритмах реального времени - УА РВ).

Основными используемыми в настоящее время методами обеспечения надежности УА РВ для БВС КА являются тестирование и отладка, которые, однако, не могут гарантировать отсутствия ошибок. При этом, в связи с необходимостью отработки взаимодействия с БА при всех возможных ситуациях на борту КА (в т.ч. нештатных), их трудоемкость является наибольшей среди этапов жизненного цикла (ЖЦ) БПО и составляет, по экспертным оценкам, около 57% общей трудоемкости. Вследствие этого проблемы обеспечения надежности и сокращения сроков и трудоемкости разработки БПО оказываются неразрывно связанными.

Одним из наиболее эффективных методов повышения качества и надежности БПО является создание средств автоматизации проектирования и отладки программ. Значимый вклад в разработку теоретических основ такой автоматизации применительно к ПО реального времени внесли такие ученые, как Е.А. Микрин, В.А. Крюков, В.В. Липаев, А.К. Петренко, Я.А. Мостовой, А.А. Калентьев, Р.Л. Смелянский, Г. Хольцманн и др.

На большинстве предприятий ракетно-космической отрасли в нашей стране (ракетно-космической корпорации «Энергия» имени С.П. Королева, ГНПРКЦ «ЦСКБ-Прогресс», НПО имени Лавочкина и др.) и за рубежом используются средства автоматизации отдельных этапов ЖЦ БПО. При этом с целью снижения количества ошибок в программах и повышения производительности труда разработчиков практикуется применение языков программирования высокого уровня и специально разрабатываемых проблемно-ориентированных языков. Так, при создании программного обеспечения космической системы Space Shuttle был разработан язык программирования высокого уровня HAL/S, имеющий в своем составе, помимо иных специальных возможностей, средства для реализации управления в реальном времени. В Институте прикладной математики имени Келдыша

при проектировании БПО многоразового космического корабля «Буран» были разработаны специализированные языки ПРОЛ2 и СИПРОЛ. Кроме того, используется комплексная отладка программ БПО совместно с программами моделей бортовой аппаратуры и факторов космического пространства, на специальном испытательном стенде.

Однако, перечисленные подходы ориентированы на верификацию уже созданных в рамках той или иной технологии программ и оставляют «за скобками» решение задач обеспечения корректности исходных требований к БПО и формальной верификации (доказательства) УА РВ.

В значительной степени повысить уровень надежности УА РВ для БВС КА могут позволить *аналитические методы верификации*, которые на основе логически строгого доказательства способны определять наличие или отсутствие у управляющего алгоритма желаемых свойств. Аналитические методы также могут облегчить проведение верификации традиционными средствами за счет автоматизированной генерации полного набора тестов, покрывающего все варианты исполнения УА РВ.

Еще больший интерес представляет разработка методов автоматизированного *синтеза* УА РВ, которые при этом гарантировали бы его корректность (соответствие спецификации). Важным преимуществом метода автоматизированного синтеза является также его инвариантность относительно используемого языка программирования и архитектуры БВС, что обеспечивает возможность быстрого перехода на новые БЦВМ и средства программирования при разработке перспективных КА.

Для спецификации и формальной верификации динамических управляющих систем в мире применяются подходы, основанные на темпоральной логике (в этой области известны работы А. Пнеули, З. Манна, А. Морценти, Д.В. Царькова, А. Эмерсона и др.), а также алгебры процессов (ССS Р. Милнера, CSP Э.Хоара, ACP Я. Бергстры и В. Клопа и др.), и графовые модели (в основном представленные различными разновидностями сетей Петри). В качестве модели семантики в темпоральных логиках обычно применяются модель Крипке или таймированные автоматы, основанные, как и сети Петри, на концепции состояний. Их использование применительно к исследованию реальных управляющих систем затруднено в связи с проблемой взрывного роста числа состояний, подлежащих моделированию. Говоря об известных алгебрах процессов, можно отметить, что имеющиеся в них операции в недостаточной степени соответствуют требованиям согласования работы бортовой аппаратуры во времени и с точки зрения логики управления. Кроме того, алгебры процессов, как правило, не ориентированы на решение задачи автоматизированной генерации управляющей программы.

Общим вопросам автоматизации различных этапов жизненного цикла ПО (CASE-средств) посвящены работы Г. Буча, Э. Хоара, Д. Румбаха, В.М. Глушкова, А.П. Ершова, С.С. Лаврова, В.В. Липаева, И.В. Вельбицкого, Г.Н. Калянова, А.Н. Коварцева, А.М. Вендрова и других ученых. В индустрии ИТ разработаны и широко используются комплексные методологии разработки программных средств (IBM Rational Unified Process, основанный на языке UML – Unified Modeling Language, Borland ALM, Microsoft Solutions Framework). При спецификации требований к телекоммуникационным системам используются специализированные языки SDL, LOTOS и др. Однако, возможности данных средств недостаточны для описания специфических свойств УА РВ для БВС КА. При этом даже в новейших расширениях упомянутых методов, имеющих некоторые возможности описания систем реального времени (UML2.0), построение логической структуры программы возлагается на разработчика, т.е. синтез УА РВ не предусматривается.

Таким образом, создание методов и средств автоматизации синтеза и верификации УА РВ для БВС КА является весьма актуальной задачей. Верификация означает проверку УА РВ на наличие у него требуемых свойств, которые при этом должны быть точно специфицированы. Поскольку спецификация должна быть непротиворечивой (а для случая управляющих алгоритмов, согласовывающих работу большого числа приборов и систем КА, обеспечение этого является самостоятельной непростой задачей), весьма актуальна и разработка методов верификации самих спецификаций. Решение задачи синтеза УА РВ также связано с необходимостью точной

спецификации требований. Для этого требуется однозначное определение семантики УА РВ. Следует отметить, что при синтезе УА РВ одна и та же семантика может быть реализована алгоритмами с разной логико-временной структурой (схемой), и различными управляющими программами, что создает пространство выбора проектных решений, варьируя которыми можно управлять эффективностью получаемой программы. Таким образом, цепочка преобразований УА РВ при его автоматизированном синтезе выглядит как:

спецификация → семантика УА РВ → логико-временная схема алгоритма → управляющая программа.

В настоящее время существует ряд формализаций семантики программных средств, в основном относящихся к одному из трех направлений - операционному, аксиоматическому (деривационному) и денотационному. Существенный вклад в их развитие внесли такие ученые, как Дж. Мак-Карти, Г. Бекик, М. Маркотти, Д. Кнут, Р. Флойд, Э. Хоар, Д. Скотт, Я. де Баккер, С.С. Лавров и др. Однако, упомянутые формализации семантики в основном ориентированы на вычислительные программы. При этом они обычно интерпретируются в рамках фон-неймановской императивной модели вычислений. В связи с этим данные модели не могут адекватно описать УА РВ для БВС КА, основным содержанием которых является выдача в требуемое время команд управления на БА и запуск функциональных программ БПО в соответствии с требованиями целевой задачи и складывающейся на борту ситуацией.

Все вышперечисленное обуславливает актуальность выбранной тематики диссертации, посвященной разработке формального математического аппарата для описания УА РВ и на этой основе – методов и средств их автоматизированного синтеза и верификации.

Объектом исследования в диссертационной работе являются управляющие алгоритмы реального времени для БВС КА, координирующие комплексное (согласованное во времени и с точки зрения логики управления БА) функционирование систем бортовой аппаратуры и программ комплекса БПО космического аппарата при решении им целевых задач.

Целью диссертационной работы является сокращение сроков и трудоемкости разработки, повышение надежности и качества управляющих алгоритмов реального времени для БВС КА на основе построения формального математического аппарата, методов и средств синтеза и верификации УА РВ, в том числе специального программного комплекса, обеспечивающего эффективную инструментальную поддержку процессов их проектирования и разработки.

Задачи исследования вытекают из поставленной цели и включают:

1. Исследование существующих подходов к синтезу и верификации динамических управляющих систем.
2. Построение формального математического аппарата для представления УА РВ, адекватно описывающего их как сложные объекты, обладающие временными и логическими аспектами.
3. Разработку средств формальной спецификации УА РВ для БВС КА.
4. Создание методов и средств синтеза УА РВ, включая построение логико-временной схемы алгоритма и генерацию управляющей программы на параметрически задаваемом целевом языке программирования.
5. Разработку и анализ эффективности методов выбора проектных решений при автоматизированном синтезе УА РВ.
6. Разработку методов и средств верификации УА РВ на различных этапах жизненного цикла.
7. Построение на основе разработанных моделей, методов и средств инструментального программного комплекса, его апробацию и исследование эффективности применительно к бортовым управляющим алгоритмам и программам моделей бортовой аппаратуры КА.

Научная новизна (личный вклад автора) работы заключается в следующем:

1. Построена модель семантики УА РВ, адекватно отражающая их временные и логические аспекты, что отличает ее от известных моделей семантики программ, ориентированных на описание преобразований данных.
2. Построена многоосновная алгебраическая система УА РВ, обеспечивающая описание отношений на множестве УА РВ и конструктивный подход к построению УА РВ на базе функциональных задач.
3. Построена формальная теория управляющих алгоритмов, позволяющая осуществлять спецификацию УА РВ, проводить их исследование и верификацию методами логического вывода и проверки на модели (model checking).
4. Разработан метод автоматизированного построения логико-временной схемы УА РВ, обеспечивающий гарантированное соответствие алгоритма заданной спецификации.
5. Разработаны методы и средства генерации управляющей программы, реализующей УА РВ, на параметрически задаваемом языке программирования.
6. Предложены методы структурной оптимизации УА РВ при его автоматизированном синтезе, обеспечивающие повышение его эффективности за счет уменьшения требований к ресурсам БВС.
7. На основе разработанных моделей, методов и средств создана новая автоматизированная технология синтеза и верификации управляющих алгоритмов реального времени для БВС космических аппаратов.

Практическая ценность работы состоит в том, что:

1. Разработанные модели, методы и инструментальный программный комплекс позволяют снизить трудоемкость и сократить сроки разработки УА РВ для БВС КА на 20-30%, повысить их надежность и качество.
2. Практическую значимость имеют следующие методы решения частных проектных задач:
 - метод визуального конструирования семантики УА РВ;
 - метод визуального конструирования логико-временной схемы алгоритма;
 - методы и средства автоматизированного синтеза УА РВ и генерации управляющей программы;
 - методы и средства построения таблицы вариантов УА РВ и генерации отладочных заданий для каждого варианта.
3. Разработанная технология применима не только для синтеза и верификации бортовых управляющих алгоритмов, но и при создании программ моделей систем БА, используемых на этапе комплексных испытаний КА.

Апробация работы проводилась в ходе выступлений и обсуждений на 31 научно-технической конференции, семинарах и симпозиумах, включая Всесоюзные Туполевские чтения (г. Казань, 1990 г.), Гагаринские чтения (г. Москва, 1991, 1992 и 2001 гг.), Всероссийскую научную школу по компьютерной алгебре, логике и интеллектуальному управлению (г. Иркутск, 1994 г.), 5-ю межвузовскую научную конференцию по математическому моделированию и крайним задачам (г. Самара, 1995 г.), 1-ю Поволжскую научно-техническую конференцию по научно-исследовательским разработкам и высоким технологиям двойного применения, VII, VIII, IX, X, XI и XII Всероссийские семинары по управлению движением и навигации летательных аппаратов (г. Самара, 1995, 1997, 1999, 2001, 2003, 2005 гг.), Международные научно-технические конференции и симпозиумы «Актуальные проблемы анализа и обеспечения надежности и качества приборов, устройств и систем» и «Надежность и качество» (г. Пенза, 1995, 1996, 1999, 2001, 2003, 2004, 2007 гг.), научную конференцию по перспективным информационным технологиям, посвященную 20-летию факультета информатики СГАУ (г. Самара, 1995 г.), Международную космическую конференцию, посвященную 40-летию первого полета человека в космос «Космос без оружия – арена мирного сотрудничества в XXI веке»

(г. Москва, 2001 г.), 8-ю, 9-ю 10-ю и 11-ю Международные научно-технические конференции «Системный анализ и управление аэрокосмическими комплексами» (г. Евпатория, 2003-2006 гг.).

Реализация результатов связана с использованием:

- Технологии автоматизации синтеза и верификации бортовых управляющих алгоритмов в ходе работ над перспективными космическими аппаратами в Государственном научно-производственном ракетно-космическом центре «ЦСКБ-Прогресс». Работа выполнялась в рамках хоздоговоров с ЦСКБ (1994-2005 гг.), ее результаты отражены в 14 научно-технических отчетах, подготовленных либо автором лично, либо при его непосредственном участии. Результаты внедрены на предприятии, что отражено в Приложении к диссертации.

- Методов разработанной автоматизированной технологии при построении программы модели бортовой телеметрической системы, используемой при комплексных испытаниях на наземном отладочном стенде, в ГНПРКЦ «ЦСКБ-Прогресс».

- Разработанных моделей, концепций и методик в учебном процессе Самарского государственного аэрокосмического университета имени академика С.П. Королева в курсах «Математические модели объектов авиационно-технической техники», «Языки программирования», в курсовом и дипломном проектировании, учебно-исследовательской работе студентов.

Научные результаты диссертационной работы являются обобщением научно-производственной деятельности диссертанта в период 1988-2007 гг.

Публикации. По материалам проведенных исследований опубликовано 57 работ, включая монографию, 35 статей в центральных и региональных научных журналах и сборниках научных трудов (в том числе девять публикаций в изданиях, рекомендуемых ВАК), а также тезисы докладов научно-технических конференций. 23 работы выполнены автором единолично.

Структура и объем диссертации. Диссертация состоит из введения, пяти глав, заключения и приложений. Общий объем работы 312 страниц машинописного текста, включая 59 рисунков, список использованных источников содержит 215 наименований.

На защиту выносятся:

1. Математический аппарат для описания управляющих алгоритмов реального времени, в том числе модель семантики УА РВ, логико-временная схема и многовариантная модель УА РВ.
2. Алгебраический подход к УА РВ, обеспечивающий конструктивное построение и исследование УА РВ с использованием операций и предикатов многоосновной алгебраической системы УА РВ.
3. Формальная теория УА РВ и ее применение в качестве средства спецификации и верификации важнейших свойств УА РВ.
4. Методы и средства автоматизированного синтеза логико-временной схемы УА РВ и параметрической генерации текста управляющей программы на заданном языке.
5. Методы выбора проектных решений при синтезе УА РВ для БВС КА.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы, дан краткий анализ текущего состояния проблемы, сформулированы цель и задачи исследований, показана научная новизна и значимость, описана практическая значимость работы, перечисляются основные положения, выносимые на защиту.

Первая глава посвящена обзору современных подходов к синтезу и верификации динамических управляющих систем, и постановке проблемы, решаемой в диссертации.

На современных космических аппаратах функции управления и контроля работоспособности БА возлагаются на бортовое программное обеспечение. При решении КА целевых задач на одном отрезке времени согласованно должны функционировать различные системы, приборы и агрегаты бортовой аппаратуры. Поскольку особую важность при этом имеет синхронизация, т.е. корректное согласование работы приборов БА и программ БПО во времени, в состав БВС и бортовой

операционной системы включаются соответствующие механизмы. Таким образом, УА РВ для БВС КА представляют собой динамическую управляющую систему реального времени.

В диссертации представлен обзор современного состояния моделей подобных систем, используемых при решении задач их спецификации и верификации. В настоящее время наиболее широко применяются подходы, основанные на *временной (темпоральной) логике, алгебрах процессов* (CCS, CSP, ACP), и сетях Петри. Рассмотрено развитие данных методов, начиная с их зарождения в середине 1960-х годов XX века, проведен анализ присущих им особенностей и ограничений. На основе анализа трудов Х.Б. Керри, Г. Бекика, В.М. Глушкова, Р. Милнера, Э. Хоара, К. Петри, Я. Бергстры, А. Пнуели, З. Манны проведена классификация существующих подходов. Показано их несоответствие без соответствующей модификации задачам синтеза и верификации управляющих алгоритмов реального времени для БВС КА.

Создание комплекса программ БПО для современных космических аппаратов представляет собой сложный процесс с большим количеством участников. Организационный сетевой график работ включает в себя сотни ключевых вершин, часто в нем содержится критический путь графика работ по созданию КА в целом. В разработке задействуются десятки алгоритмистов и программистов, как создающих программы управления конкретной бортовой аппаратурой, так и разрабатывающих алгоритмы комплексного функционирования, обеспечивающие согласованную работу ряда систем и подсистем КА в рамках решения важных целевых задач (УА РВ).

При традиционном методе в качестве исходных данных для построения УА РВ используется описание логики управления подсистемами КА и требуемых временных соотношений на естественном языке, например: «При проведении операции O_1 система C_1 должна быть приведена в исходное состояние выдачей команд K_1, \dots, K_5 . Перед пуском КА необходимо после выполнения программы P_2 проверить условия A_1, A_2 . В случае ложности условия A_1 необходимо осуществить переход в режим штатной работы вызовом программы P_{11} . В случае ложности A_2 необходимо повторно вызвать программу P_2 для формирования массива X . Непосредственно перед моментом t_0 необходимо обеспечить функционирование P_1 . За 360 сек до t_0 требуется включить режим R_0 системы C_1 . При этом необходимо обеспечить выполнение ограничений: интервал между выдачей команд K_2, \dots, K_5 должен быть не менее 1 сек. Интервал между K_1 и K_2 составляет не менее 2 мин». На этой основе разработчик УА РВ формирует таблицу условий (глобальных переменных), определяющих необходимость выполнения алгоритмом тех ли иных действий, и затем вручную строит временную диаграмму алгоритма (документ, отражающий логико-временную схему УА). Затем он осуществляет написание и отладку управляющей программы, реализующей временную диаграмму, на заданном языке программирования. При этом ответственность за непротиворечивость исходных требований, за соответствие формируемой логико-временной схемы алгоритма спецификации и за корректность управляющей программы ложится на человека.

В настоящее время срок разработки комплекса БПО для нового космического аппарата составляет от двух до четырех лет (из них 44% приходится на этапы анализа требований и проектирования). Самым трудоемким является обеспечение требуемого уровня надежности бортовых программ, в основе которого лежит многоэтапное тестирование.

На основании этого в качестве актуальных задач, возникающих при разработке управляющих программ, выделены автоматизация процесса формирования требований (*спецификации*), разработка методов формальной *верификации*, позволяющих, в отличие от тестирования и отладки, строго доказывать соответствие программы требованиям аналитическими средствами, создание методов автоматизированного *синтеза* УА РВ, гарантированно соответствующего спецификации. Решение перечисленных задач, в свою очередь, требует построения модели семантики УА РВ.

Приведен обзор существующих методов описания семантики программ. Наиболее общий характер, абстрагированный от деталей реализации, имеет аксиоматическая модель семантики (Флойд, Хоар). Поэтому с точки зрения описания семантики УА РВ наибольший интерес представляет именно это направление. Однако в своем традиционном виде аксиоматическая семантика направлена на описание последовательных вычислительных программ, что не соответствует предметной области

УА РВ для БВС КА, где основным содержанием алгоритма являются не вычисления, а своевременная выдача управляющих воздействий на бортовую аппаратуру.

Делается вывод о том, что решение задач синтеза и верификации УА РВ для БВС КА возможно на основе построения модели семантики УА РВ и ее использования совместно с соответствующим образом адаптированными и интегрированными принципами алгебраического и логического подходов.

Во второй главе строится формальный математический аппарат для описания УА РВ, и на этой основе - методы и средства синтеза и верификации УА РВ для БВС КА.

В отличие от вычислительных программ, применительно к УА РВ для БВС КА, с одной стороны, неправильно вести речь об исходных данных, не претерпевающих изменений в процессе исполнения программы. С другой стороны, о корректности работы УА РВ недостаточно судить по состоянию программной памяти в момент его завершения. Основной функцией УА РВ является реализация требуемых управляющих воздействий в заданные моменты времени, в зависимости от решаемой целевой задачи и текущего состояния КА и внешней среды.

Модель семантики УА РВ может быть построена как набор кортежей (четверок) Φ_i :

$$\text{УА РВ} = \{ \Phi_i \}, \Phi_i = \langle f_i, t_i, \tau_i, \bar{l}_i \rangle, i = \overline{1, N}, \quad (1)$$

где f_i – идентификатор функциональной задачи (ФЗ); t_i – момент начала выполнения ФЗ (целое неотрицательное число); τ_i – длительность ФЗ (целое неотрицательное число); \bar{l}_i – логический вектор, обуславливающий ФЗ.

Каждый кортеж Φ_i описывает одно действие (функциональную задачу), производимое управляющим алгоритмом. Φ_i обычно подразумевает работу какого-то прибора или агрегата, входящего в состав БА, или выполнение функциональной программы из комплекса БПО. При этом функциональная задача может выполняться не мгновенно, а на протяжении интервала времени τ_i , начиная с момента t_i . Осуществление тех или иных действий не носит безусловного характера, а должно соответствовать текущей ситуации на борту КА, которая описывается набором значений логических переменных $\langle \alpha_1, \dots, \alpha_k \rangle$, формирующих логический вектор. Таким образом, выполнению ФЗ в момент времени t_i сопоставляется логический вектор, обуславливающий данное действие. Значение каждой из логических переменных (ЛП), обуславливающих выполнение ФЗ, принадлежит множеству $\{1, 0, N\}$. Здесь 1 обозначает ИСТИНУ, 0 – ЛОЖЬ, N в соответствующей компоненте логического вектора подразумевает, что выполнение ФЗ не зависит от значения данной логической переменной. По своему смыслу логические переменные α_k , входящие в состав логического вектора либо отражают показания некоторого датчика, входящего в состав БА, либо значение «флага» в памяти БЦВМ, устанавливаемого программой БПО, либо сложное условие, формируемое относительно текущей информации в заданной области программной памяти.

Предлагаемая модель семантики УА РВ является альтернативной по отношению к известным моделям, используемым в темпоральных логиках. К ним, прежде всего, относятся таймированные системы переходов и таймированные автоматы. Данные модели семантики схожи между собой и базируются на концепции состояний. Однако, при их использовании на практике встает проблема быстро растущей размерности, т.к. количество состояний системы определяется числом возможных комбинаций значений логических переменных в каждый моделируемый момент времени и в случае достаточно малой длительности такта системных часов по отношению к продолжительности работы алгоритма быстро становится необозримым.

Предложенная в диссертации модель семантики позволяет избежать этой проблемы, поскольку не требует детальной имитации всех состояний системы и конкретной реализации переходов между ними, а позволяет проводить анализ свойства УА РВ сразу на всем временном отрезке его функционирования. Кроме того, принятый подход позволяет затем рассматривать реализации одной

и той же семантики УА РВ различными логико-временными схемами, за счет чего открывается возможность выбора предпочтительных с точки зрения эффективности проектных решений при синтезе УА РВ.

Затем в диссертации строится многоосновная алгебраическая система УА РВ:

$$АУА = \langle \{UA, LV, Z_{\geq 0}\}, \Omega_F, \Omega_P, \rangle,$$

основными множествами (носителями) которой являются:

UA - множество УА РВ в виде наборов кортежей (1),

LV - множество логических переменных $\{\alpha_k\}$,

$Z_{\geq 0}$ – множество моментов времени (неотрицательные целые числа, соответствующие возможным показаниям дискретного бортового датчика времени КА).

Множество операций $\Omega_F = \Omega_{FT} \cup \Omega_{FL}$ включает операции во временном пространстве $\Omega_{FT} = \{CH, CK, \rightarrow\}$ и логические операции $\Omega_{FL} = \{\neg, \Rightarrow\}$.

Множество отношений, определенных на элементах множества UA, включает в себя $\Omega_P = \{\ll, \langle, \langle / \rangle, CH, CK, \rightarrow\}$. Символы отношений отличаются от одноименных символов операций гарнитурой и курсивным начертанием шрифта.

Отношение *CH* имеет место, если два УА начинаются в одно и то же время. Отношение *CK* имеет место, если два УА РВ завершаются в одно и то же время, отношение \rightarrow - если непосредственно в момент завершения выполнения некоторого алгоритма начинается выполнение другого управляющего алгоритма.

Иными словами, с точки зрения интерпретации параметров УА РВ на временной оси:

$$T_1 \text{ CH } T_2 \Leftrightarrow t_{T1} = t_{T2}$$

$$T_1 \text{ CK } \Leftrightarrow t_{T1} + \tau_{T1} = t_{T2} + \tau_{T2}$$

$$T_1 \rightarrow T_2 \Leftrightarrow t_{T1} + \tau_{T1} = t_{T2}$$

Здесь t_{T1} и t_{T2} – моменты начала выполнения алгоритмов T_1 и T_2 соответственно, τ_{T1} и τ_{T2} – длительности их исполнения. Кроме того, в рассмотрение вводятся дополнительные отношения:

- простого временного предшествования \ll : ($T_1 \ll T_2$) $\Leftrightarrow t_{T1} < t_{T2}$

- «строгого» временного предшествования $\ll\ll$: ($T_1 \ll\ll T_2$) $\Leftrightarrow t_{T1} + \tau_{T1} < t_{T2}$

- наложения во времени *H*: ($T_1 \text{ H } T_2$) $\Leftrightarrow (t_{T1} + \tau_{T1} \leq t_{T2}) \vee (t_{T2} + \tau_{T2} \leq t_{T1})$

- несовместности $\langle \rangle$: ($T_1 \langle \rangle T_2$) $\Leftrightarrow (t_{T1} + \tau_{T1} < t_{T2}) \vee (t_{T2} + \tau_{T2} < t_{T1})$, может быть определен как отрицание предиката *H*

- несовместности по логике \langle / \rangle : ($T_1 \langle / \rangle T_2$) $\Leftrightarrow l_{T1} \# l_{T1}$, т.е. логический вектор, обуславливающий выполнение T_1 , несовместен с логическим вектором, обуславливающим выполнение T_2 . Понятие логической несовместности логических векторов означает, что в них в качестве значений одной и той же ЛП α_k присутствуют в одном ИСТИНА, в другом - ЛОЖЬ.

К основным операциям над УА РВ во времени относятся:

- операция совмещения по времени начала выполнения (*CH*),

- совмещения по концу (*CK*),

- непосредственного следования (\rightarrow).

Логическая операция $(\alpha_1) \Rightarrow T_1$ означает, что в логическом векторе, обуславливающим выполнение алгоритма T_1 компонента, соответствующая ЛП α_1 принимает значение 1, операция $(\neg\alpha_1) \Rightarrow T_1$ – значение 0.

Добавление в описание множества неотрицательных целых чисел $Z_{\geq 0}$ позволяет расширить алгебраическую систему УА РВ операциями, аргументами которых будут, с одной стороны, элементы множества УА РВ, а с другой - числа из $Z_{\geq 0}$:

$H(T_1, T_2, x)$ - операция над алгоритмами и числом x , означающая наложение выполнения T_2 на выполнение T_1 , со сдвигом начала выполнения T_2 по отношению к T_1 , равным x .

$ZA(T_1, T_2, x)$ – операция, означающая образование нового УА с началом выполнения T_2 в момент времени, отстоящий на интервал x от момента окончания выполнения алгоритма T_1 .

@ (T, x) – операция привязки старта алгоритма T к абсолютному значению времени x.

Свойства объектов алгебраической системы УА РВ могут записываться и исследоваться с помощью формальной теории УА РВ.

Теория УА РВ позволяет производить спецификацию управляющих алгоритмов реального времени с точки зрения их свойств (временных и логических), и осуществлять логические рассуждения (выводы) о них. Это дает возможность проводить формальную верификацию спецификаций УА РВ. Основным используемым при этом методом является дополнение базовых аксиом теории набором аксиом, соответствующих спецификации конкретного управляющего алгоритма, и поиск противоречий.

Алфавит теории включает предметные символы T_i для обозначения функциональных задач, входящих в состав УА РВ, предикатные символы $CH, CK, \rightarrow, H, <, <<, <>, </>$, символ отрицания \sim , круглые скобки () . Правильно построенные формулы имеют вид $T_i \$ T_j$ или $\sim(T_i \$ T_j)$, где $\$$ используется для обозначения произвольного символа отношения. Одновременная выводимость формулы Φ и ее отрицания, т.е. $\sim(\Phi)$, считается недопустимой. Это позволяет проводить анализ спецификаций УА РВ, записанных на языке формальной теории, на непротиворечивость.

Формальная теория УА РВ содержит следующие аксиомы (схемы аксиом, т.е. вместо символов T_i, T_j и T_k могут быть подставлены любые предметные символы, обозначающие ФЗ) и правила вывода (схемы правил вывода):

Аксиомы

A01. $T_i CH T_i$ A02. $T_i CK T_i$ A03. $\sim(T_i \rightarrow T_i)$ A04. $T_i H T_i$

A05. $\sim(T_i < T_i)$ A06. $\sim(T_i << T_i)$ A07. $\sim(T_i <> T_i)$ A08. $\sim(T_i </> T_i)$

Правила вывода

П1. $\frac{T_i CH T_j}{T_j CH T_i}$ П2. $\frac{T_i CK T_j}{T_j CK T_i}$ П3. $\frac{T_i CH T_j, T_j CH T_k}{T_i CH T_k}$ П4. $\frac{T_i CK T_j, T_j CK T_k}{T_i CK T_k}$

П5. $\frac{T_i CH T_j}{\sim(T_i \rightarrow T_j)}$ П6. $\frac{T_i CK T_j}{\sim(T_i \rightarrow T_j)}$ П7. $\frac{T_i \rightarrow T_j}{\sim(T_i CH T_j)}$ П8. $\frac{T_i \rightarrow T_j}{\sim(T_i CK T_j)}$

П9. $\frac{T_i CH T_j}{T_i H T_j}$ П10. $\frac{T_i CK T_j}{T_i H T_j}$ П11. $\frac{T_i \rightarrow T_j}{\sim(T_j H T_i)}$ П12. $\frac{T_i \rightarrow T_j}{T_i < T_j}$

П14. $\frac{T_i << T_j}{T_i < T_j}$ П15. $\frac{T_i H T_j}{T_j H T_i}$ П16. $\frac{T_i H T_j}{\sim(T_i <> T_j)}$ П17. $\frac{T_i << T_j}{T_i <> T_j}$

П18. $\frac{T_i <> T_j}{T_j <> T_i}$ П19. $\frac{T_i </> T_j}{T_j </> T_i}$ П20. $\frac{T_i <> T_j}{\sim(T_i H T_j)}$ П21. $\frac{T_i H T_j}{\sim(T_i \rightarrow T_j)}$

П22. $\frac{T_i << T_j, T_j << T_k}{T_i << T_k}$ П23. $\frac{T_i < T_j, T_j < T_k}{T_i < T_k}$ П24. $\frac{(\alpha_k) \Rightarrow T_i}{\sim((\neg \alpha_k) \Rightarrow T_i)}$

Помимо приведенного варианта, в диссертации приводится формулировка теории УА РВ как прикладного исчисления предикатов первого порядка. Благодаря этому, теория «погружается» в среду языка Пролог, что дает возможность использования встроеного в него механизма логического вывода для исследования свойств управляющих алгоритмов. Этот вопрос подробно освещается в специальном подразделе второй главы, посвященном использованию Пролога для решения задач верификации УА РВ.

Формулы теории УА РВ интерпретируются в рамках многоосновной алгебраической системы УА РВ. Так, формула $f_i \text{ CH } f_j$ будет означать, что в кортежах, описывающих ФЗ f_i и f_j в данном управляющем алгоритме, $t_i = t_j$, т.е. моменты запуска соответствующих функциональных задач совпадают. Формула $f_i \text{ CK } f_j$ будет означать, что в четверках, описывающих ФЗ f_i и f_j в данном УА, $t_i + \tau_i = t_j + \tau_j$, т.е. времена окончания выполнения функциональных задач совпадают. Формула $f_i \rightarrow f_j$ означает, что в четверках, описывающих вхождения функциональных задач f_i и f_j в данном УА РВ, $t_i + \tau_i = t_j$. Формула $(\alpha_k) \Rightarrow f_i$ означает, что соответствующая компонента логического вектора, входящего в кортеж, описывающий вхождение ФЗ f_i в данный УА РВ, должна иметь значение 1 (ИСТИНА).

В диссертации доказана теорема о том, что алгебраическая система УА РВ является моделью формальной теории УА РВ.

Набор формул может быть интерпретирован на некотором управляющем алгоритме, имеющем конкретные значения длительностей для каждой из функциональных задач. В образующемся таким образом «возможном мире» каждая из формул теории УА РВ принимает истинностное значение. Аксиомы при этом сохраняют свою общезначимость. Это дает возможность проводить верификацию свойств УА РВ, записанных с помощью формул теории УА РВ, не только путем логического вывода, но и путем проверки на модели (метод model checking).

Интерпретация, кроме того, позволяет проверять реализуемость (выполнимость) имеющегося набора формул (спецификации управляющего алгоритма) на заданном базисе. Выполнимость будет иметь место в том случае, если все формулы спецификации одновременно могут являться истинными.

В соответствии с изложенным, становится возможной постановка задачи «разрешения спецификации», т.е. определения значений длительностей ФЗ, при которых заданная спецификация будет выполнима. Единственное решение эта задача будет иметь в том случае, если известны длительности и логические вектора всех ФЗ, кроме одной (неизвестной), и необходимо определить, при каких параметрах искомой ФЗ спецификация будет выполнимой. В остальных случаях имеется множество решений, тем не менее, возможно определение границ области решений (области выполнимости в пространстве параметров). Решение задачи производится путем перехода от формул теории УА РВ к уравнениям и неравенствам на множестве целых чисел. Для дальнейшего их решения можно применить любой известный метод, в том числе, осуществив описание семантики УА РВ средствами системы Пролог можно использовать для поиска значений неизвестных параметров ФЗ встроенную в Пролог машину логического вывода. Вопросам применения среды логического программирования Пролог при работе с представлениями УА РВ для БВС КА посвящен специальный подраздел главы.

Вторая глава содержит также подраздел, посвященный сравнению описываемого в диссертации подхода с существующими. С одной стороны, предложенный подход во многом подобен применению темпоральных логик и алгебр процессов, с другой стороны, у него имеется ряд принципиальных отличий. Так, он изначально предназначался для описания процессов реального времени, и в отличие, например, от интервальной логики Аллена описывает не только качественные, но и количественные характеристики соотношений между исполняемыми процессами. При этом построенная модель включает как относительное, так и абсолютное время за счет присутствия операции @. Применение описанной модели семантики УА РВ (фактически представляющей статическое описание динамического исполнения УА РВ) позволяет отказаться от пошагового моделирования состояний алгоритма при проверке формул теории УА РВ. Коренным отличием от подхода, присущего временным логикам является, кроме того, предоставление не только способов описания свойств существующего УА РВ, но и конструктивных возможностей для построения управляющих алгоритмов на некотором базисе элементарных ФЗ. Это роднит его с алгебрами процессов (BPA, ATP, ACP, CSP и др.). В то же время, предлагаемая система обладает более богатыми изобразительными возможностями, чем большинство алгебр процессов. При представлении различных вариантов поведения системы в них обычно имеется способ лишь самым общим образом описать альтернативную композицию двух процессов. В отличие от этого, предлагаемый подход

позволяет сопоставить каждой функциональной задаче, выполняемой управляющим алгоритмом, логический вектор, который может включать в свой состав целый набор логических переменных. Кроме того, при описании параллельного исполнения процессов обычно отсутствует возможность точной синхронизации моментов их начала и окончания, в то время как в рамках алгебраической системы УА РВ присутствуют специальные связки СН, СК и Н.

Во второй главе также содержится описание *многовариантной* модели и логико-временной схемы УА РВ.

Рассмотрим набор кортежей (1), задающий семантику УА РВ. Зафиксируем какое-то конкретное значение логического вектора. В результате кортеж из четырех значений преобразуется в тройку W_i :

$$MWM = \{ W_i \}, W_i = \langle f_i, t_i, \tau_i \rangle, i = \overline{1, N}.$$

Полученная модель описывает один из возможных вариантов исполнения управляющего алгоритма. При этом в отличие от операционно-логической истории программы указывается не просто последовательность шагов, а точные значения времени начала и длительности выполнения отдельных действий. Диссертация содержит процедуру преобразования данного представления в графовую модель, описывающую структуру конкретной реализации управляющего алгоритма. Под «*ходом*» при этом понимается включение (получение управления) управляющего алгоритма в некоторый момент времени t_i , в который он должен осуществить запуск некоторой функциональной задачи f_i (или нескольких функциональных, если в УА имеется несколько ФЗ с одинаковым временем запуска). С точки зрения программной реализации это соответствует одной из точек входа управляющей программы, по которой производятся некоторые действия, после чего осуществляется возврат в бортовую операционную систему. Список базовых действий включает занесение запросов на включение программ БПО по истечении заданного временного интервала, для чего используются специальные возможности БВС и бортовой операционной системы. В том числе возможно занесение запроса на включение следующего входа данного УА РВ, что формирует косвенную передачу управления.

Сопоставляя каждому включению (входу) управляющего алгоритма вершину графа и соединяя их дугами в случае наличия передачи управления между входами, мы получаем граф, соответствующий траектории во времени (включениям) одного варианта УА. Дугам графа при этом приписываются веса, соответствующие интервалам времени между входами.

Возможно также построение расширенной модели, отражающей все варианты исполнения алгоритма. В этом случае первый шаг (фиксация значения логического вектора и отбрасывание несоответствующих ему кортежей в модели семантики УА) не производится, и все входящие в алгоритм различающиеся моменты времени, в которые включаются функциональные задачи, отображаются во входы модели. Такая расширенная многоходовая модель фактически представляет собой *логико-временную схему* алгоритма. Логико-временная схема является аналогом управляющего графа последовательной программы, в котором, помимо дуг, обозначающих передачу управления от одного оператора программы другому, имеются особые дуги – дуги с приписанными им весами. Взвешенная дуга обозначает передачу управления между входами УА РВ не непосредственно, а по истечении интервала времени, соответствующего весу дуги. Вершинам сопоставляются элементарные действия (запуски функциональных задач и проверки значений логических переменных) УА РВ. Дуги без весов соответствуют простой передаче управления в традиционном алгоритмическом смысле (следование действий). В логико-временной схеме присутствуют, кроме того, «терминальные» вершины, соответствующие завершению работы УА РВ (передаче управления из управляющей программы программе-диспетчеру бортовой операционной системы).

Поскольку в описание УА РВ сразу несколько функциональных задач могут входить с одним и тем же временем запуска, они будут принадлежать одному входу. В связи с этим для каждого входа можно построить традиционный управляющий граф программы с вершинами, соответствующими операторам (запускам ФЗ) и вершинами, сопоставляемыми проверкам условий. Таким образом, вход

будет иметь собственную внутреннюю структуру (возможно, достаточно сложную). Кроме того, в общем случае каждый вход может передавать управление не одному, а нескольким входам.

В процессе исполнения алгоритма обеспечивается реализация некоторой семантики УА РВ. На рисунке 1 приведен пример соответствия между логико-временной схемой и семантикой УА РВ.

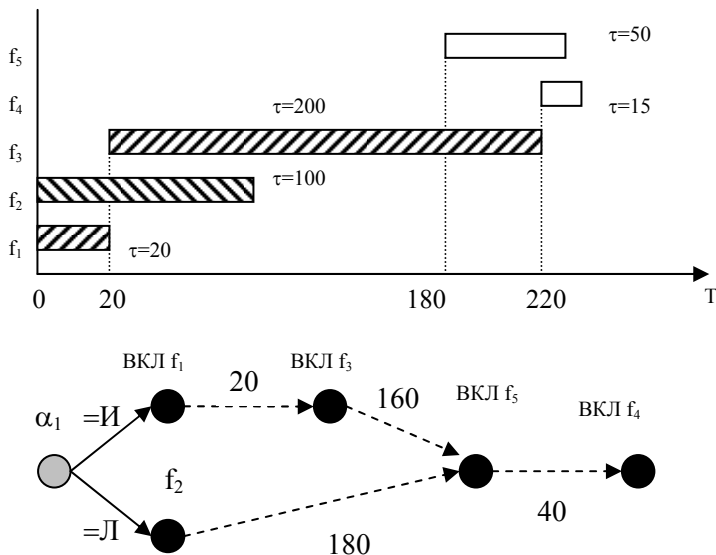


Рис. 1. Переход от семантики УА РВ к логико-временной схеме алгоритма

Логико-временные схемы, реализующие одну и ту же семантику УА РВ, являются *семантически эквивалентными*. В диссертации приводится алгоритм, позволяющий построить логико-временную схему управляющего алгоритма по его семантике, сгенерированной, в свою очередь, на основе спецификации на языке формальной теории УА РВ. Кроме того, становится возможной верификация логико-временной схемы путем выявления реализуемой ей семантики, и проверки выполнимости на ней формул спецификации УА РВ.

На основе модели семантики УА РВ можно построить также *многовариантную модель* УА РВ, играющую важную роль, в частности, при отладке управляющих программ, когда требуется точный набор возможных вариантов выполнения. Если зафиксировать в семантике УА РВ некоторый момент времени (вход), то получится набор:

$$МВМ = \{ \langle f_i, \tau_i, \overline{I_i} \rangle, i = \overline{1, N} \}$$

Данное представление включает в себя описание для каждой функциональной задачи обуславливающего ее выполнение в исследуемый момент времени логического вектора. Если устранить в ней все строки, в которых логические вектора совпадают, т.е. дубли, и отбросить первые два компонента каждой триады, то фактически получается таблица возможных вариантов исполнения данного входа. Если же применить операцию простого отбрасывания компоненты времени запуска в формирующих семантику УА РВ кортежах, то получается расширенная многовариантная модель всего управляющего алгоритма (на основе которой нетрудно построить таблицу возможных вариантов исполнения, имеющую важное значение с точки зрения исследования и тестирования УА РВ).

В таблице 1 приводится перечень основных описываемых в диссертации моделей УА РВ и задачи, при решении которых они используются.

Таблица 1.

<i>Разновидность модели</i>	<i>Использование</i>
Модель семантики УА РВ	Визуальное конструирование УА РВ, оптимизация плана управления, контроль соответствия (верификация) семантики УА РВ спецификации, проверка спецификации УА РВ на выполнимость
Многоосновная алгебраическая система УА РВ	Визуальное конструирование УА РВ, проверка спецификации УА РВ на выполнимость, построение множества возможных вариантов исполнения УА.
Формальная теория УА РВ	Спецификация требований к УА РВ. Верификация спецификаций УА РВ, проверка спецификации УА РВ на выполнимость.
Многовариантная модель УА РВ	Построение множества вариантов исполнения УА, генерация отладочных заданий для вариантов.
Логико-временная схема УА РВ	Синтез и верификация логико-временной схемы УА РВ, построение документации на УА РВ, генерация управляющей программы.

Третья глава посвящена описанию методов выбора проектных решений (структурной оптимизации) на различных этапах автоматизированного синтеза УА РВ для БВС КА.

На этапе *спецификации* аксиомы и правила вывода формальной теории УА РВ делают возможным проведение эквивалентных преобразований формул, в том числе – сокращающих общую длину формулы. Таким образом, задача оптимизации на этом этапе ставится как:

$$N_F(S) \rightarrow \min, \text{ при } S = \text{const},$$

где $N_F(S)$ - целевая функция, определяющая количество функциональных и предметных символов, содержащихся в спецификации, S – описываемая спецификацией семантика.

При этом, несмотря на изменение вида записи и значительное сокращение формулы (синтаксическая редукция), семантика описываемого ей управляющего алгоритма не претерпевает каких-либо изменений, т.е. подразумевается запуск на выполнение тех же функциональных программ, в те же моменты времени и при выполнении тех же логических условий. В диссертации приводится алгоритм проведения эквивалентных преобразований спецификаций УА РВ на языке формальной теории, использующий представление формул теории УА РВ с помощью бинарных деревьев.

Еще одной возможностью является оптимизация семантики УА. Пространство выбора проектных решений здесь образуется благодаря тому, что при спецификации требований к УА РВ с помощью набора формул формальной теории УА РВ может существовать не одна, а несколько удовлетворяющих спецификации семантик за счет присутствия таких связей, как $<$, $<<$, и H .

Наиболее практически обоснованными при этом будут два критерия:

1. Снижение количества одновременно выполняемых в каждый момент времени на борту КА процессов (оптимизация по критическому сечению).

2. Сокращение общей продолжительности исполнения УА РВ (оптимизация по длительности). В этом случае задача оптимизации при синтезе конкретного плана управления (семантики УА РВ) ставится как $T_2(УА) \rightarrow \min$, при условии выполнения всех формул спецификации,

где $T_2(YA)$ – общая продолжительность исполнения УА, или $N_{ФЗ}(t) \rightarrow \min, t \in [0, T]$, где $N_{ФЗ}(t)$ – количество параллельно исполняемых в момент времени t функциональных задач, $[0, T]$ – продолжительность исполнения УА РВ.

Поскольку приведенные критерии противоречат друг другу, оптимизация различных участков управляющего алгоритма может проводиться гибко, т.е. с выбором различных критериев для разных участков алгоритма с учётом контекста состояния БВС и решаемых на данном временном отрезке задач.

Следующим этапом синтеза УА РВ является генерация логико-временной схемы алгоритма, реализующей построенную семантику. На этом этапе открывается еще одна возможность оптимизации. В качестве целевой функции, значение которой подлежит минимизации, фигурирует число логических вершин графа логико-временной схемы (вершин, в которых происходит ветвление) N_L , и задача оптимизации ставится как:

$N_L(YA) \rightarrow \min$, при сохранении реализуемой семантики УА.

В диссертационной работе приводится описание двух типов структурной оптимизации при синтезе логико-временных схем УА РВ. Первый касается преобразований структуры отдельного входа УА. Вторым относится к оптимизации на основе учета связей между входами. Оптимизация внутренней логической структуры входа основывается на упорядочении логических переменных их «весу» - количеству зависимых от них ФЗ, и может давать достаточно существенный выигрыш по количеству логических ветвей в алгоритме. Идея заключается в том, что проверка переменных, от которых меньше всего зависит выполнение ФЗ, производится в последнюю очередь. При синтезе логико-временной схемы наихудший по количеству ветвей вариант оценивается величиной $2^M - 1$, где M – количество фигурирующих в управляющем алгоритме логических переменных. Число же ветвлений в наилучшем (оптимизированном) варианте будет стремиться к M . Таким образом, уже для случая семи переменных по сравнению с наихудшим случаем достигается выигрыш, более чем в десять раз. Потенциал оптимизации первого рода в зависимости от M иллюстрируется рисунком 2.

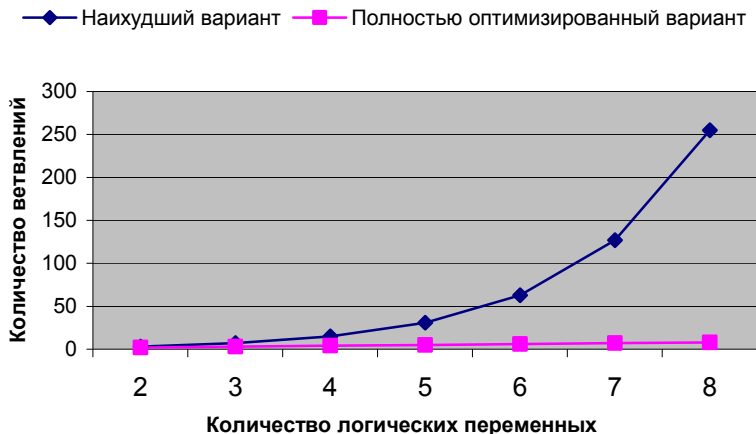


Рисунок 2. Эффективность оптимизации структуры входа УА РВ

На рисунке 3 показана сущность второго метода оптимизации, основанного на построении «цепочек» одинаково обусловленных входов УА. Представлена структура алгоритма, включающего несколько входов, между которыми заданы временные интервалы. Изображенные на рисунке логико-временные схемы УА могут быть описаны одной и той же семантикой УА (зададим для определенности длительности ФЗ $\tau_{f1}=100, \tau_{f2}=200, \tau_{f3}=50, \tau_{f4}=150$): $YA = \{ \langle f_1, 0, 100, (\alpha_1=0) \rangle, \langle f_2, 0, 200, (\alpha_1=1) \rangle, \langle f_3, 0, 10, (\alpha_1=0) \rangle, \langle f_4, 0, 210, (\alpha_1=1) \rangle \}$.

В то же время, во втором случае выполняется на две проверки логических условий меньше, что приводит к упрощению структуры. Это достигается за счет выноса операции занесения временной уставки на включение соответствующего входа, в одну из ветвей, расположенных после проверки значения логической переменной. В примере фигурирует только одна переменная – α_1 , но принцип оптимизации не меняется и для более сложных комбинаций проверок логических условий.

При проведении оптимизации данным методом важно учитывать, что логические условия, фигурирующие в логико-временной схеме УА, по времени их актуальности могут относиться к одному из трех типов (родов):

1. α_{t_0} , когда значение переменной может меняться, но актуальным является ее значение в некоторый момент времени t_0 (часто это момент начала исполнения УА), и последующие изменения не должны влиять на логику УА РВ;
2. $\alpha_{\text{стат}}$, когда значение логической переменной не претерпевает изменений в процессе исполнения УА.
3. $\alpha(t)$, когда требуется учитывать значение логической переменной в текущий момент времени t , и с течением времени значение ЛП может меняться.

Оптимизация такого типа применима для логических условий первого и второго рода (α_{t_0} и $\alpha_{\text{стат}}$), значения которых либо статичны, т.е. не изменяются в процессе исполнения алгоритма, либо должны учитываться на начальный момент времени.

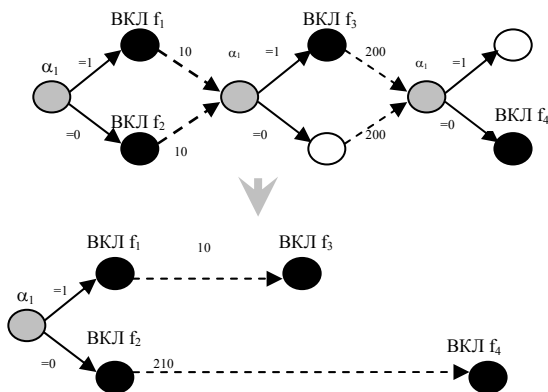


Рис.3. Оптимизирующее преобразование логико-временной схемы УА

В случае, если все логические переменные, используемые в алгоритме, будут относиться к логическим условиям третьего рода (т.е. их значение меняется в ходе исполнения алгоритма и именно это текущее значение подлежит проверке), от описанного вида оптимизации придется отказаться. В данной ситуации применимо представление УА так называемой *регулярной схемой*.

В регулярной схеме каждый вход запускает на выполнение ближайший по времени вход, тот, в свою очередь, ближайший себе, и так далее до самого последнего по времени входа. Внутри каждого входа осуществляется проверка всех значимых логических переменных. Диссертация содержит алгоритм построения регулярной схемы по заданной семантике УА РВ. Сравнение количества проверок в регулярной логико-временной схеме и в максимальной степени оптимизированной для различного количества входов приводится на рисунке 4.

Кроме описанных выше случаев оптимизации, при автоматизированном синтезе управляющего алгоритма возможна оптимизация на этапе генерации текста управляющей программы.

Задача оптимизации в этом случае ставится как задача сокращения общего количества строк в синтезируемой программе при сохранении ее семантики:

$$N_k(\text{УП}) \rightarrow \min, S = \text{const},$$

где $N_k(\text{УП})$ – общее количество строк в генерируемой управляющей программе, S – семантика УА РВ. В диссертации описывается машинно-зависимая оптимизация, опирающаяся на особенности реализации конкретной архитектуры бортовой вычислительной системы и синтаксис команд ассемблера БЦВМ, задаваемый с помощью специальных параметрических файлов формата XML.

В частности, осуществляется оптимизация меток. На первом проходе генерируется текст управляющей программы. При этом возможен случай, когда несколько логических ветвей алгоритма «сходятся» в одном и том же месте, и строка программы помечается сразу несколькими метками.

Повторный проход производит удаление излишних меток и их замену во всех переходах, в которых они встречаются, на эквивалентные метки. Кроме того, осуществляется сокращение «лишних» команд переходов.

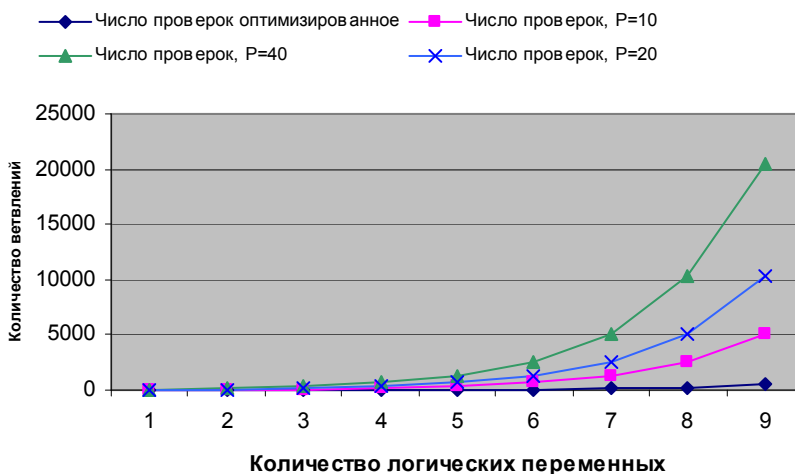


Рис. 4. Число проверок логических условий с оптимизацией и без оптимизации

В рамках оптимизации на данном этапе в тексте управляющей программы также выявляются одинаковые фрагменты сегментов данных, и в тексте программы оставляется лишь один из них, ссылка на который подставляется во соответствующих обращениях к данным.

В качестве выводов третьей главы констатируется, что применение описанных методов выбора предпочтительных проектных решений при синтезе УА РВ на различных стадиях жизненного цикла может давать существенный выигрыш в эффективности алгоритма с точки зрения требуемых ресурсов (времени, памяти и производительности БВС).

Четвертая глава посвящена описанию разработанной автоматизированной технологии и поддерживающего ее инструментального программного комплекса. Его применение обеспечивает автоматизацию практически всех этапов жизненного цикла УА РВ для БВС КА, от спецификации до генерации отладочных заданий и внесения изменений в УА РВ на этапе эксплуатации и сопровождения.

Можно выделить следующие основные задачи, решаемые в рамках разработанной автоматизированной технологии:

- Формальная *спецификация* требований к УА РВ с использованием проблемно-ориентированного языка, основанного на языке формальной теории УА РВ или визуального интерактивного языка
- Построение семантики управляющего алгоритма, соответствующей спецификации.
- Построение многовариантной модели УА РВ, т.е. выделение всех значимых при исполнении управляющей программы комбинаций значений логических переменных.
- Построение логико-временной схемы УА РВ.
- Генерация технической документации на УА РВ в составе временной диаграммы и блок-схемы алгоритма.
- Генерация управляющей программы БЦВМ, реализующей управляющий алгоритм, на параметрически задаваемом языке программирования.
- Генерация отладочных заданий (на языке технологической среды отладки БПО) для каждого возможного варианта исполнения УА РВ.
- Генерация таблиц входных и выходных информационных и управляющих связей программы.

Проектировщики и разработчики УА РВ взаимодействуют при этом с несколькими интерфейсными модулями. Это в первую очередь - интегрированная среда, являющаяся аналогом IDE современных систем программирования Borland C++ Builder, MS Visual Studio, Eclipse, IDEA. Она объединяет все модули программного комплекса в единое целое и позволяет получить из единого центра доступ к любой функции системы: графическому конструктору управляющих алгоритмов, графическому редактору временных диаграмм, генераторам управляющей программы и отладочных заданий, редактору описаний ФЗ и ЛП. Благодаря этому, интегрированная оболочка инструментального программного комплекса может в некоторой степени рассматриваться, как аналог PDM-средства, интегрирующего различные электронные модели изделия в традиционных пакетах САПР. Как известно, интерфейс пользователя в PDM-системе обычно предусматривает возможность одновременного просмотра разных представлений проектных данных в нескольких окнах. На рисунке 5 представлен экран компьютера при работе с одновременно открытыми окнами, содержащими различными виды представления управляющего алгоритма, что иллюстрирует наличие подобных возможностей разработанного программного комплекса.

Модель семантики УА РВ допускает наглядную графическую интерпретацию в виде циклограммы. Циклограмма является одним из основных документов, используемых при описании функционирования КА, что облегчает понимание и использование построенной модели семантики разработчиками УА РВ. В состав инструментального программного комплекса входит специальный графический конструктор, позволяющий просматривать и конструировать семантику УА РВ визуально в виде циклограммы. Таким образом, проектировщик имеет выбор: либо осуществить ввод спецификации алгоритма на проблемно-ориентированном языке во встроенном текстовом редакторе, либо провести интерактивное построение.

Графический конструктор УА РВ может работать в прямом и обратном режимах. В прямом режиме он принимает на вход описание базиса, на котором формируется управляющий алгоритм, т.е. логических переменных и функциональных задач, а выходом служит спецификация УА РВ на проблемно-ориентированном языке. В обратном режиме модуль позволяет считывать спецификацию из текстового файла и наглядно отображать графический образ соответствующей семантики УА РВ. Кроме того, возможна проверка выполнения (истинности) тех или иных формул теории УА РВ при интерпретации спецификации на заданном базисе ФЗ, т.е. поддержка *верификации* спецификации. После этого спецификация УА РВ на проблемно-ориентированном языке транслируется во внутреннее представление. В результате работы транслятора прежде всего строится многовариантная модель, отражающая многообразие вариантов исполнения УА РВ. На этой основе генерируется модель семантики алгоритма.

Далее строится логико-временная схема УА РВ, которая используется в качестве основы для генерации управляющей программы на целевом языке программирования и технической документации, включающей временную диаграмму управляющего алгоритма с отражением логики функционирования и блок-схему программы.

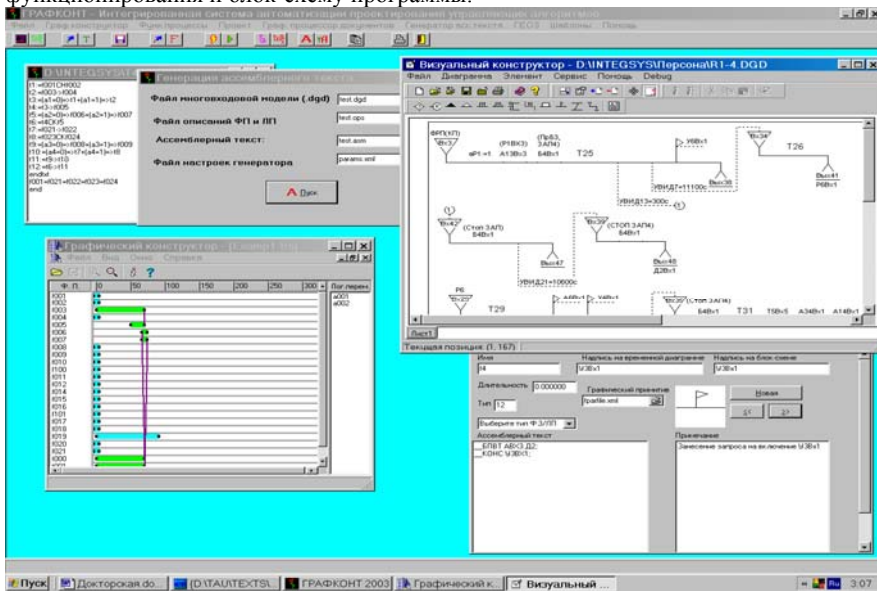


Рис. 5. Многооконный интерфейс инструментального программного комплекса

При генерации управляющей программы в качестве исходных данных, помимо логико-временной схемы, используется описание базиса функциональных задач и логических переменных, которое может быть представлено следующим образом:

$$OF = \{ \langle f_i, \tau_i, Tx_i, Pr_i, Tur_i, GP_i \rangle \},$$

$$LP = \{ \langle \alpha_i, Tx_i, Pr_i, Tur_i, GP_i \rangle \},$$

где f_i – идентификатор ФЗ, α_i – имя ЛП, τ_i – длительность исполнения ФЗ; Tur_i – тип функциональной задачи или логической переменной, $Tur_i \in \{Tur_k\} = \{ \langle \text{«выдать команду на БА»}, \langle \text{«занести запрос на включение программы БПО»}, \langle \text{«открыть канал»}, \langle \text{«спросить сигнал»}, \langle \text{«проверить переменную»} \dots \rangle \}$ - тип действия, каждому типу соответствует свой шаблон фрагмента управляющей программы; Tx_i – параметры конкретной ФЗ или ЛП; Pr_i - текстовое описание ФЗ (примечание); GP_i - код графического примитива, отражающего ФЗ на временной диаграмме УА РВ.

Описание ФЗ и ЛП проводится с помощью специального модуля программного комплекса. Параметрическая генерация управляющей программы в рамках автоматизированной технологии позволяет значительно ускорить процесс программирования и облегчить переход от одних целевых языков программирования и архитектур БВС к другим.

Помимо автоматизированного синтеза логико-временной схемы УА РВ по его спецификации, система также позволяет начинать описание управляющего алгоритма с построения его временной диаграммы в специальном визуальном конструкторе. При этом происходит оперирование графическими примитивами, соответствующими предметной области УА БВС КА (деревья входов, логические условия, типовые ФЗ), а не простейшими графическими элементами, такими как точки и отрезки, как при использовании неспециализированной САПР (например, AutoCAD), что обеспечивает значительное повышение производительности труда. В процессе визуального конструирования в памяти компьютера «на лету» строится соответствующая логико-временная схема

УА РВ. Визуальный конструктор может применяться и для ручной правки логико-временной схемы, ранее построенной на основе семантики алгоритма. Эта возможность весьма значима с точки зрения таких характеристик качества БПО, как восстанавливаемость и сопровождаемость. Особенно актуальна она может быть в случае нештатной ситуации, когда оперативное внесение изменений и передача исправленного ПО на борт КА по радиоканалу позволяет в ряде случаев, несмотря на имеющуюся проблему, выполнить поставленную перед КА задачу. Следует подчеркнуть, что при этом, как и в случае необходимости штатных доработок БПО при создании модифицированных КА, предлагаемая технология предоставляет возможность быстрого и удобного внесения изменений в УА РВ с соответствующим автоматическим изменением технической документации и генерируемой управляющей программы.

В диссертации приводится алгоритм обратного преобразования, позволяющий получать по логико-временной схеме, сконструированной пользователем-проектировщиком УА в визуальном конструкторе, семантику реализуемого ей управляющего алгоритма. На этой основе становится возможной *верификация* построенной в визуальном конструкторе логико-временной схемы алгоритма, т.е. проверка ее соответствия спецификации.

Несмотря на использование перечисленных аналитических методов верификации УА РВ, в том числе формальных и методом *model checking*, полный отказ от тестирования и отладки при разработке БПО не представляется целесообразным (в частности, ошибка может возникнуть вследствие пробела в исходных материалах по логике управления подсистемами КА). В связи с этим созданная автоматизированная технология обеспечивает поддержку верификации УА РВ традиционными средствами, т.е. путем тестирования и отладки. Подсистема генерации отладочных заданий позволяет на базе внутренних структур данных системы сформировать отладочное задание для автономной отладки управляющего алгоритма на специальном технологическом языке. При этом автоматически выявляются все возможные варианты исполнения («маршруты») алгоритма и осуществляется генерация отладочного задания для каждого варианта, что позволяет достичь наиболее эффективного уровня тестирования – С2 по существующей классификации уровней тестирования программных средств. Кроме того, система позволяет на основании текста управляющей программы и описаний ФЗ выявлять имеющиеся в алгоритме информационные и управляющие связи с другими программами и формировать их таблицу, которая используется на этапах совместной и комплексной отладки.

Структура инструментального программного комплекса, поддерживающего разработанную автоматизированную технологию, представлена на рисунке 6.

В соответствующих подразделах четвертой главы приводятся разработанные автором алгоритмы решения частных задач автоматизации: задачи генерации семантики УА РВ, задачи построения многовариантной модели, задачи

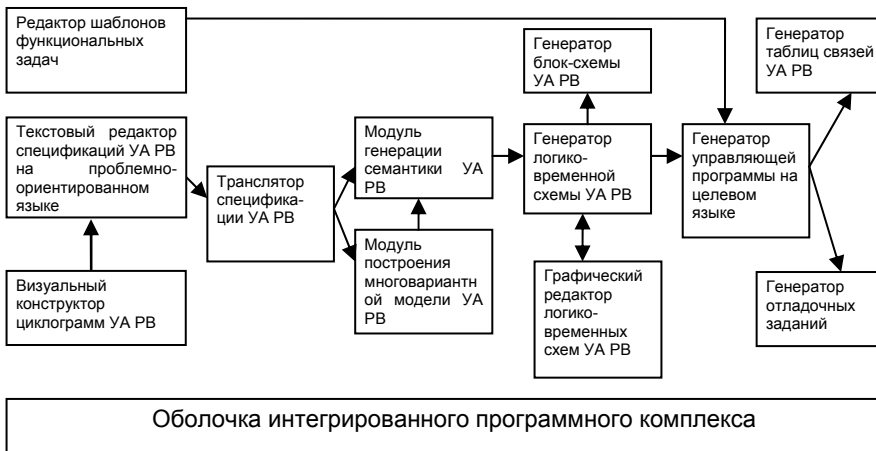


Рис. 6. Структура инструментального программного комплекса

построения логико-временной схемы алгоритма, задачи генерации управляющей программы, задачи генерации отладочных заданий.

В четвертой главе приводится также оценка эффективности разработанной автоматизированной технологии. С использованием функционального моделирования процессов ЖЦ УА РВ было проведено исследование на примере программы управления комплексным функционированием БА при приведении КА в ориентированное положение. Оно показало (см. рисунок 7), что наибольшую эффективность с точки зрения снижения трудоемкости (до 75%-80%) технология имеет на этапах программирования и автономной отладки и на этапе подготовки технической документации на алгоритм. В целом же применение описанных методов и средств позволяет сократить длительность жизненного цикла приблизительно на 20% при общем снижении трудоемкости разработки УА РВ на величину до 30%.

Применение автоматизированной технологии обеспечивает повышение надежности и качества управляющих алгоритмов по следующим причинам.

Использование строгих и регулярных методик описания УА РВ на различных этапах ЖЦ понижает значимость человеческого фактора. За счет использования разработанного языка формальной спецификации удастся уменьшить число ошибок на этапе формирования требований, в первую очередь это касается наиболее сложно выявляемых и трудноустраняемых на последующих этапах семантических ошибок. Средства теории УА РВ позволяют проводить формальную верификацию управляющих алгоритмов. При автоматизированном синтезе УА РВ гарантируется его соответствие заданной спецификации, т.е. соблюдение содержащихся в ней требований к согласованию исполнения функциональных задач по времени и с точки зрения логики управления. За счет проведения оптимизирующих преобразований при синтезе управляющего алгоритма становится возможным снижение потребления ресурсов БВС, т.е. повышение эффективности УА РВ. Упрощается этап сопровождения комплекса программ БПО ввиду удобства предложенных моделей для анализа, простоты

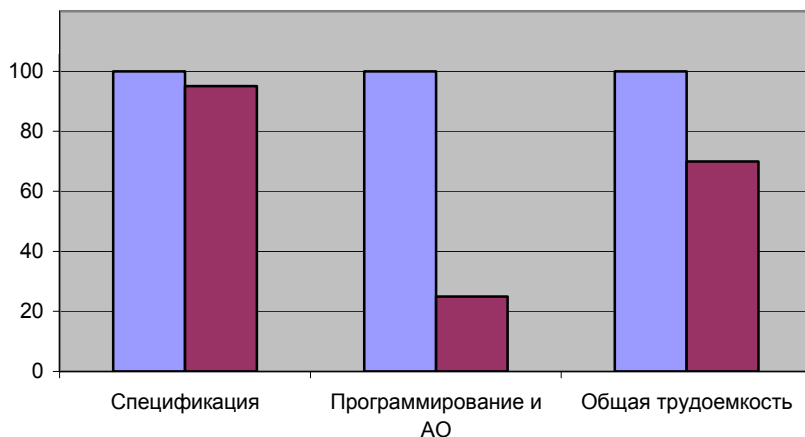


Рис. 7. Снижение трудоемкости при применении автоматизированной технологии

внесения изменений и автоматизации тестирования. Облегчается переносимость комплекса управляющих программ БПО, в том числе на языки программирования высокого уровня и другие БЦВМ на основе параметрической генерации управляющей программы, реализующей алгоритм. Все это позволяет говорить о повышении качества управляющей программы на основе улучшения перечисленных характеристик, входящих в стандартное определение качества программных средств.

Возможность автоматизированной генерации отладочных заданий (тестов), гарантирующих проверку всех ветвей дерева управляющего алгоритма, также является значимой с точки зрения повышения надежности УА РВ.

Пятая глава посвящена описанию практического использования разработанной автоматизированной технологии и программного комплекса.

Приводится сквозной пример разработки - от спецификации до получения текста управляющей программы и отладочных заданий, одного из бортовых управляющих алгоритмов – алгоритма приведения КА в ориентированное положение.

Кроме того, демонстрируется применимость автоматизированной технологии к задачам построения программ моделей бортовой аппаратуры КА на примере бортовой информационно-телеметрической системы (БИТС).

Важным с точки зрения анализа жизненного цикла КА и его БПО является то, что при создании комплекса БПО, помимо собственно бортовых управляющих программ, необходимо разработать комплекс программ моделей бортовой аппаратуры, движения КА и факторов внешней среды. Программы моделей используются в процессе комплексной отладки БПО и испытаний КА в качестве замены натуральных приборов и агрегатов, что позволяет сократить сроки разработки. При этом программа модели проходит практически те же стадии жизненного цикла, что и УА РВ, а ее суть состоит в реализации циклограммы функционирования бортовой аппаратуры в различных режимах.

Предлагаемая технология позволяет автоматизировать этапы жизненного цикла программ моделей БА на основе тех же подходов, структур данных и программных средств, что используются для бортовых алгоритмов управления.

Таким образом, применение описанного в диссертационной работе подхода дает возможность использовать так называемое совмещенное проектирование (concurrent design), являющееся характерной отличительной чертой современных CALS-технологий в промышленности и

позволяющее сократить временные затраты путем перекрытия по времени фаз жизненного цикла (операций, процедур или этапов), при традиционном подходе выполняемых последовательно.

В главе поставлена задача построения программы модели бортовой информационно-телеметрической системы (БИТС) КА. Рассказывается об основных функциях и структуре системы телеметрирования, приводятся сведения о режимах работы. Описывается использование разработанной автоматизированной технологии для решения данной задачи, начиная с интерактивного конструирования временной диаграммы (циклограммы) работы и выделения функциональных задач, исполняемых программой модели.

Показано, как по результатам графического конструирования автоматизированно формируется текст спецификации алгоритма модели на проблемно-ориентированном языке, а затем генерируется программа на языке программирования Си. Таким образом, демонстрируется практическая применимость предлагаемого подхода для программ моделей бортовой аппаратуры, используемых при комплексной отладке БПО. При этом переход от генерации управляющей программы на языке ассемблера БЦВМ к генерации программы модели на языке Си, исполняемой на наземном отладочном комплексе, осуществляется простой заменой настроечных файлов.

В заключении приводятся основные выводы:

1. Исследованы существующие подходы к синтезу и верификации динамических управляющих систем. В основе методов решения этих задач на современном этапе лежат логические (темпоральная логика), алгебраические (алгебры процессов), и графовые (сети Петри) модели. Однако, они не в полной мере соответствуют особенностям УА РВ для БВС КА, что вызвало необходимость построения специализированного математического аппарата.

2. Построен формальный математический аппарат для описания УА РВ, в том числе впервые построена модель семантики УА РВ с адекватным отражением их временных и логических аспектов в виде набора кортежей, описывающих выполнение функциональных задач в заданные моменты времени: $УА РВ = \{ \Phi_i \}, \Phi_i = \langle f_i, t_i, \tau_i, \bar{I}_i \rangle, i = \bar{1}, \bar{N}$.

3. Разработаны языки и средства формальной спецификации УА РВ, в том числе с применением визуального конструирования.

4. Разработаны методы и средства синтеза УА РВ для БВС КА, включая:

- алгоритм генерации семантики УА РВ, соответствующей спецификации;
- алгоритм построения логико-временной схемы (структуры) УА РВ;
- методы и средства параметрической генерации управляющей программы, реализующей требуемую семантику УА РВ, на целевом языке программирования (включая ассемблер БЦВМ и язык Си).

5. Сформулированы критерии, предложены методы выбора проектных решений при синтезе УА РВ (структурной оптимизации) и проведено исследование их эффективности, в том числе:

- методов эквивалентных преобразований спецификаций УА РВ;
- методов оптимизации семантики УА РВ;
- методов оптимизации логико-временной схемы управляющего алгоритма;
- методов оптимизации при генерации управляющей программы, реализующей управляющий алгоритм.

6. Разработаны методы и средства верификации УА РВ, в том числе:

- проверки спецификации УА РВ на непротиворечивость;
- проверки выполнимости спецификации УА РВ на заданном базисе функциональных задач и поиска разрешающего базиса, в том числе с применением машины логического вывода среды программирования Пролог;

- верификации логико-временной схемы управляющего алгоритма (проверки соответствия реализуемой логико-временной схемой семантики спецификации УА РВ);
- автоматизированного построения таблицы возможных вариантов УА РВ и генерации отладочного задания на автономную отладку для каждого варианта.
- автоматизированного построения таблицы информационных и управляющих связей УА РВ с другими программами БПО.

7. На основе разработанного математического аппарата, методов и средств синтеза и верификации УА РВ построен инструментальный программный комплекс. Проведенное исследование эффективности его применения при создании бортовых управляющих алгоритмов и программ моделей БА показало возможность достижения существенного (на 20-30%) снижения общей трудоемкости этапов жизненного цикла УА РВ для БВС КА.

Таким образом, в диссертации решена **актуальная научная проблема, имеющая важное народнохозяйственное значение** - проблема сокращения сроков и трудоемкости разработки, повышения надежности и качества УА РВ для БВС КА на основе разработки формального математического аппарата, методов и средств синтеза и верификации УА РВ в рамках автоматизированной технологии, поддерживаемой инструментальным программным комплексом.

Разработанные в диссертации модели, методы синтеза и верификации УА РВ, инструментальный программный комплекс могут использоваться в различных отраслях народного хозяйства, в которых встает задача управления сложными техническими системами в реальном времени - таких, как телекоммуникации, транспорт, энергетика, химические и нефтехимические производства, и т.п.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ ДИССЕРТАЦИИ ОПУБЛИКОВАНЫ В СЛЕДУЮЩИХ РАБОТАХ:

Монография:

1. Калентьев А.А., Тюгашев А.А. ИПИ/CALS технологии в жизненном цикле комплексных программ управления. - Самара: СНЦ РАН, 2006. - 285 с.: ил.- 1000 экз. - ISBN 5-934-24257-1.

Статьи в изданиях, рекомендованных ВАК для публикации результатов докторских диссертаций:

2. Тюгашев, А.А. Интегрированная среда для проектирования управляющих алгоритмов реального времени // Известия РАН. Теория и системы управления. -2001.- №5.-, С. 128-141.

3. Калентьев А.А., Тюгашев А.А. Использование технологии ГРАФКОНТ при автоматизированном проектировании управляющих программ реального времени для БВС КА. // Всероссийский аэрокосмический журнал «Полет». – 2005. - №12. - С. 41-47.

4. Тюгашев А.А. Автоматизированная спецификации, верификация и синтез управляющих программ на основе логического и алгебраического подходов // Вестник СГАУ. – 2007.- №1(12). – С. 171-179.

5. Тюгашев А.А. Автоматизированное оценивание временных характеристик программ управления комплексного функционирования в технологии ГРАФКОНТ // Вестник СГТУ. Серия «Физико-математические науки». – 2007. - №1 (14). – С. 134-137.

6. Тюгашев А.А. Автоматизация спецификации, синтеза и верификации управляющих программ реального времени с применением логического и алгебраического подходов // Мехатроника, автоматизация, управление.- 2007. - №7. - С. 46-50.

7. Тюгашев А.А. Методология и программный комплекс для автоматизированного проектирования управляющих алгоритмов реального времени // Вестник СНЦ РАН. Том 7.- 2005. - №1. - С.131-137.

8. Калентьев А.А., Тюгашев А.А. CALS-технологии для бортовых алгоритмов управления космическими аппаратами // Вестник СНЦ РАН. Том 7. - 2005.- №1. - С.124-130.

9. Тюгашев А.А. Алгебраические модели управляющих алгоритмов и программ реального времени для космических аппаратов. // Вестник СГТУ. Серия «Физико-математические науки».- 2005. - выпуск 38. – С. 19-25.

10. Тюгашев А.А. Технология проектирования надежных управляющих алгоритмов реального времени для космических аппаратов. // Вестник СГАУ. - 2004. - №1. – С.124-131.

Статьи в сборниках научных трудов:

11. Калентьев А.А., Тюгашев А.А. Оптимизация логико-временной схемы управляющего алгоритма при его автоматизированном синтезе. // Международный симпозиум «Надежность и качество 2007»:сб. научн. тр. / ПГУ. – Пенза, 2007. -С.117-118.

12. Калентьев А.А., Тюгашев А.А. Математическое моделирование управляющих алгоритмов и программ реального времени. // Международный симпозиум «Надежность и качество 2005»:сб. научн. тр. / ПГУ. – Пенза, 2005. ЧП.-С.251-257.

13. Калентьев А.А., Тюгашев А.А. Технология проектирования надежных и качественных алгоритмов управления реального времени. // Труды Международного симпозиум «Надежность и качество 2004»: сб. научн. тр. / ПГУ.- Пенза, 2004. – С.249-253.

14. Калентьев А.А., Тюгашев А.А. Методология проектирования надежных алгоритмов управления для космических аппаратов // 11-й Всероссийский научно-технический семинар по управлению движением и навигации летательных аппаратов: сб. научн. тр. / СГАУ.– Самара, 2003. - С.115-119.

15. Тюгашев А.А. Технология проектирования качественных и надежных управляющих программ реального времени для БВС КА // Международный симпозиум «Надежность и качество 2003»: сб. научн. тр. / ПГУ.- Пенза, 2003. -С.201-203.

16. Тюгашев А.А. Автоматизация оценивания временных характеристик программ управления комплексного функционирования в технологии ГРАФКОНТ // Международный симпозиум «Надежность и качество 2003»: сб. научн. тр. / ПГУ. – Пенза, 2003. – С.198-201.
17. Калентьев А.А., Тюгашев А.А. Развитие системы автоматизации проектирования бортовых управляющих алгоритмов реального времени ГРАФКОНТ // Международный симпозиум «Надежность и качество 2002»: сб. научн. тр. / ПГУ. – Пенза, 2002. – С.123-129.
18. Тюгашев А.А. Использование функции выполнимости для описания алгоритмов управления дискретными асинхронными системами // Международный симпозиум «Надежность и качество 2002»: сб. научн. тр. / ПГУ. – Пенза, 2002. – С.129-132.
19. Калентьев А.А., Мостовой Я.А., Тюгашев А.А. Технология ГРАФКОНТ для автоматизированного проектирования комплексных управляющих программ. // Сб. научно-технических статей по ракетно-космической тематике. / ГНПРКЦ «ЦСКБ-Прогресс».- Самара,2001. - С.165-167.
20. Калентьев А.А., Мостовой Я.А., Тюгашев А.А. Проблема неоднозначности при порождении синтаксических структур над семантикой управляющих алгоритмов реального времени. // Сб. научно-технических статей по ракетно-космической тематике. / ГНПРКЦ «ЦСКБ-Прогресс».- Самара,2001. - С.162-165.
21. Тюгашев А.А. Построение улучшенного транслятора проблемно-ориентированного языка описания управляющих алгоритмов реального времени в системе ГРАФКОНТ // Международный симпозиум «Надежность и качество 2001»: сб. научн. тр. / ПГУ. – Пенза, 2001. – С.33-35.
22. Калентьев А.А., Тюгашев А.А. Проблема автоматизированного формирования отладочных заданий в рамках информационной технологии автоматизированного проектирования бортовых алгоритмов управления // Международный симпозиум «Надежность и качество 2001»: сб. научн. тр. / ПГУ. – Пенза, 2001. – С.35-37.
23. Тюгашев А.А. Неоднозначность порождения логической схемы управляющего алгоритма реального времени // Перспективные информационные технологии в научных исследованиях, проектировании и обучении: сб. научн. тр. / СГАУ. - Самара, 2001.-С.29-35.
24. Калентьев А.А., Тюгашев А.А. Математическая модель алгоритмов управления системами с множеством дискретных состояний // Перспективные информационные технологии в научных исследованиях, проектировании и обучении: сб. научн. тр. / СГАУ. - Самара, 2001.-С.29-35.
25. Калентьев А.А., Тюгашев А.А. Задача согласования логики и времени исполнения функциональных управляющих программ в системе ГРАФКОНТ // Управление движением и навигация летательных аппаратов: сб. научн. тр./ Самарский филиал академии космонавтики. – Самара, 2000. - С.90-95.
26. Калентьев А.А., Мостовой Я.А., Платонов С.Н., Тюгашев А.А., Ендуткина Л.И., Николаев Ю.А. Система автоматизированного синтеза бортовых алгоритмов управления // Управление движением и навигация летательных аппаратов: сб. научн. тр./ Самарский филиал академии космонавтики. – Самара, 2000. - С.96-97.
27. Калентьев А.А., Мостовой Я.А., Тюгашев А.А. Параметризованная генерация кода ассемблерного текста в CASE-технологии ГРАФКОНТ // Международный симпозиум «Надежность и качество»: кн.докл. / ПГУ– Пенза, 1999.С.186-187.
28. Калентьев А.А., Мостовой Я.А., Тюгашев А.А. Проблема переполнения при кодировании векторов трехзначной логики в системе ГРАФКОНТ // Международный симпозиум «Надежность и качество»: кн.докл. / ПГУ– Пенза, 1999.С.185-186.
29. Калентьев А.А., Тюгашев А.А. Графический редактор в системе автоматизации проектирования управляющих алгоритмов // Международная научно-техническая конференция «Актуальные проблемы анализа и обеспечения надежности и качества приборов, устройств и систем»: сб. докладов. - Пенза, 1998., - С. 158-159.
30. Калентьев А.А., Тюгашев А.А. Автоматизированный синтез управляющих программ реального времени на языке ассемблера // Международная научно-техническая конференция

«Актуальные проблемы анализа и обеспечения надежности и качества приборов, устройств и систем»: сб. докладов . - Пенза, 1998., - С. 156-157.

31. Тюгашев А.А. Построение текста управляющего алгоритма на базе многоходовой модели. // Управление движением и навигация летательных аппаратов: сб. научн. тр./ Самарский филиал академии космонавтики. – Самара, 1998. - С.71-74.

32. Калентьев А.А., Тюгашев А.А. Проектирование управляющих алгоритмов средствами объектно-ориентированного программирования. // Управление движением и навигация летательных аппаратов: сб. научн. тр./ Самарский филиал академии космонавтики. – Самара, 1998. - С.68-71.

33. Калентьев А.А., Тюгашев А.А. Алгебраические модели и программные средства для проектирования алгоритмов управления бортовой аппаратурой. // Цифровые модели в проектировании и производстве РЭС: межвузовский сб. научн. тр.. / ПГТУ. - Пенза, 1996. – С.30-35.

34. Калентьев А.А., Тюгашев А.А. Языковые средства системы автоматизированного синтеза алгоритмов управления реального времени // Международная научно-техническая конференция «Актуальные проблемы анализа и обеспечения надежности и качества приборов, устройств и систем»: сб. докл.. / ПГТУ.- Пенза, 1996. С.55-57.

35. Калентьев А.А., Тюгашев А.А. Многовариантный синтез алгоритмов управления реального времени // VII-й Всероссийский семинара с межд. участием по управлению движением и навигации летательных аппаратов : сб. докладов / СГАУ. - Самара, 1995. –С.19-25.

36. Тюгашев А.А. Процессор выходных документов в системе автоматизации проектирования управляющих алгоритмов // VII-й Всероссийский семинара с межд. участием по управлению движением и навигации летательных аппаратов : сб. докладов / СГАУ. - Самара, 1995. –С.17-19.

37. Калентьев А.А., Тюгашев А.А. Разработка подсистемы синтеза управляющих алгоритмов на базе исчисления УА // Всероссийская научная школа «Компьютерная алгебра, логика и интеллектуальное управление. Проблемы анализа стратегической стабильности»: сб. трудов / ИрВЦ СО РАН. - Иркутск, 1994.

Тезисы докладов:

38. Калентьев А.А., Тюгашев А.А. Алгебраические и логические подходы к спецификации, верификации и синтезу управляющих программ для КА // 11-я Международная научно-техническая конференция «Системный анализ, управление и навигация»: тез. докл. / МАИ.-М., 2006. – С.171.

39. Калентьев А.А., Тюгашев А.А. Автоматизированное оценивание характеристик управляющих программ БВС. // 10-я Международная научно-техническая конференция «Системный анализ, управление и навигация»: тез. докл. / МАИ.-М., 2005. – С.101.

40. Калентьев А.А., Тюгашев А.А. Методы повышения качества при проектировании управляющих программ для космических аппаратов. // 9-ая Международная конференция «Системный анализ и управление»: тез. докл. / МАИ.-М., 2005. – С.56.

41. Калентьев А.А., Тюгашев А.А. Технология проектирования надежных бортовых алгоритмов реального времени // 8-ая Международная конференция «Системный анализ и управление»: тез. докл. / МАИ.-М., 2003. - С. 55.

42. Калентьев А.А., Тюгашев А.А. Автоматизация проектирования и отладки бортовых управляющих алгоритмов CASE-средствами // 7-ая Международная конференция «Системный анализ и управление»: тез. докл. / МАИ.-М., 2002. - С. 51-52.

43. Калентьев А.А., Тюгашев А.А. Система автоматизации проектирования алгоритмов управления летательными аппаратами // 6-я Международная конференция "Системный анализ и управление аэрокосмическими комплексами»: тез. докл. / МАИ.-М., 2001 - С. 99.

44. Калентьев А.А., Тюгашев А.А. Моделирование алгоритмов управления асинхронными системами с дискретными состояниями // Тезисы докладов Международная космическая конференции, посвященной 40-летию первого полета человека в космос «Космос без оружия – арена мирного сотрудничества в XXI веке»: тез. докл. / МАИ. -Москва, 2001.С.50.

45. Тюгашев А.А. Технология ГРАФКОНТ автоматизированного создания алгоритмов управления космическими аппаратами // XXVII н.т.к., посвященная 40-летию первого полета человека в космос «Гагаринские чтения»: тез. докл. / МАТИ. - Москва, 2001.С.19-20.

46. Тюгашев А.А. Параметрическая генерация текста управляющей программы реального времени // XXVII н.т.к., посвященная 40-летию первого полета человека в космос «Гагаринские чтения»: тез. докл. / МАТИ. - Москва, 2001. - С.21.

47. Калентьев А.А., Тюгашев А.А. Использование многовходовой модели для представления управляющего алгоритма // Научно-техническая конференция «Перспективные информационные технологии в научных исследованиях, проектировании и обучении»: тез. докл./ СГАУ.- Самара, 1995. - С.65.

48. Калентьев А.А., Тюгашев А.А. Применение CASE-технологии для проектирования управляющих алгоритмов реального времени // Научно-техническая конференция «Перспективные информационные технологии в научных исследованиях, проектировании и обучении»: тез. докл./ СГАУ.- Самара, 1995. - С.64.

49. Калентьев А.А., Тюгашев А.А. Алгебраическая система синтеза УА // 1-ая Поволжская научно-техническая конференция «Научно-исследовательские разработки и высокие технологии двойного применения»: тез. докл. / СГАУ. - Самара, 1995.С.51-52.

50. Тюгашев А.А. Автоматизация проектирования алгоритмов управления реального времени на основе алгебраических моделей // V-я научная межвузовская конференции «Математическое моделирование и краевые задачи»: тез. докл. / СамГТУ. - Самара, 1995.С.18.