

Федеральное государственное автономное образовательное учреждение высшего образования «Сибирский федеральный университет»

На правах рукописи



Космынина Наталья Александровна

**ЯЗЫКОВЫЕ И ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА СОЗДАНИЯ И
ИСПОЛНЕНИЯ СЦЕНАРИЕВ УПРАВЛЕНИЯ КОСМИЧЕСКИМИ
АППАРАТАМИ**

Специальность 05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Легалов Александр Иванович

Новосибирск – 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1 ЯЗЫКОВЫЕ СРЕДСТВА УПРАВЛЕНИЯ КОСМИЧЕСКИМИ АППАРАТАМИ	11
1.1 Применение языковых средств на этапах разработки и эксплуатации космических аппаратов.....	11
1.2 ПО управления космическими аппаратами	13
1.2.1 Состав ПО управления.....	13
1.2.2 Принципы построения ПО управления.....	14
1.2.3 Примеры ПО управления.....	16
1.3 Языковые средства управления космическими аппаратами	28
1.3.1 Возможности языков управления	28
1.3.2 Примеры языков управления	28
1.4 Требования, предъявляемые к языковым средствам управления	39
Выводы по главе 1	39
2 ОРГАНИЗАЦИЯ ПРОЦЕССА УПРАВЛЕНИЯ.....	41
2.1 Основные задачи управления.....	41
2.2 Автоматизированная система управления.....	42
2.3 Режимы управления	43
2.3.1 Автономный режим управления.....	43
2.3.2 Оперативный режим управления.....	44
2.4 Состав наземного комплекса управления.....	45
2.5 Технологический цикл управления	46
2.6 Командный метод управления космическими аппаратами	46
2.7 Этапы эксплуатации космических аппаратов	47
2.7.1 Подготовка к вводу в эксплуатацию	47
2.7.2 Штатная эксплуатация	50
2.7.3 Заключительные операции	53
2.8 Задачи управления на этапе штатной эксплуатации	55
2.9 Сценарии управления.....	56
2.10 Библиотека сценариев управления.....	58
2.11 Модель предметной области	60
Выводы по главе 2	62

3 ПРЕДМЕТНО-ОРИЕНТИРОВАННЫЙ ЯЗЫК УПРАВЛЕНИЯ КОСМИЧЕСКИМИ АППАРАТАМИ	64
3.1 Требования к языку управления	64
3.2 Язык управления космическими аппаратами «Дельта».....	65
3.2.1 Общие характеристики языка управления.....	65
3.2.2 Синтаксис	66
3.2.3 Структура сценария управления	67
3.2.4 Операторы языка управления	68
3.3 Поддержка англоязычного варианта языка управления	79
3.4 Оценка соответствия языка «Дельта» предъявляемым требованиям	82
3.5 Семантический контроль.....	84
Выводы по главе 3	85
4 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА СОЗДАНИЯ И ИСПОЛНЕНИЯ СЦЕНАРИЕВ УПРАВЛЕНИЯ КОСМИЧЕСКИМИ АППАРАТАМИ.....	87
4.1 Структура инструментальных средств.....	87
4.1.1 Интерпретатор языка «Дельта».....	88
4.1.2 Редактор сценариев управления	91
4.1.3 Конвертер	91
4.1.4 Библиотека сценариев управления	97
4.2 Реализация.....	98
4.3 Интерфейс пользователя.....	98
4.4 Сравнение размерности и времени заведения сценариев на языках ЯОТР и «Дельта»	102
Выводы по главе 4.....	107
ЗАКЛЮЧЕНИЕ	109
СПИСОК ЛИТЕРАТУРЫ.....	111
СПИСОК СОКРАЩЕНИЙ.....	121
ПРИЛОЖЕНИЕ А СИНТАКСИС ЯЗЫКА «ДЕЛЬТА»	123
ПРИЛОЖЕНИЕ Б АКТ О ВВОДЕ В ЭКСПЛУАТАЦИЮ	127

ВВЕДЕНИЕ

Актуальность темы. В настоящее время космические аппараты (КА) применяются в самых разных областях [1, 2]: связь, навигация, геодезия, телевидение и другие. Для выполнения КА целевых функций необходимо обеспечить решение задач управления [3]: удержание КА на заданной орбите, обеспечение живучести КА, включение/отключение различных режимов работы полезной нагрузки, выполнение регламентных работ по подсистемам КА.

Командный метод управления КА [3, 4] обеспечивает решение задач управления КА выдачей соответствующих команд с Земли, с учетом текущего состояния, анализируемого на основе непрерывно получаемой телеметрии. Программные средства управления КА, помимо автоматизации выдачи команд и анализа телеметрии, выполняют также и сервисные задачи: сбор и сохранение статистики о выданных командах управления КА и средствами наземного комплекса управления (НКУ), планирование работ с КА на произвольный промежуток времени и другие.

Последовательность команд, обеспечивающую решение каждой отдельной задачи управления, принято называть сценарием управления. Для описания таких сценариев широко используются языковые средства управления КА, позволяющие описать логику выдачи команд. Примеры отечественных и зарубежных предметно-ориентированных языков [5, 6, 7, 8]: ЯОТР (АО «ИСС»), Python (GMV), PLUTO (ESA), SCL (Interface & Control Systems), STOL (NASA), TAO (Kratos Integral Systems International), CSTOL (LASP), CCL (Harris), JAS (L-3 Telemetry-West), Cecil (Raytheon Company), CIL (Jet Propulsion Laboratory), TOPE/tcl (ESA), UCL (Airbus Defence & Space), Elisa (Astrium), PIL (Astrium).

Функциональные возможности языков управления связаны, в первую очередь, с необходимостью обеспечения автоматизации процесса управления, что обеспечивается языковой поддержкой выполнения следующих задач: выдачи команд, анализа текущих значений телеметрических параметров, выдачи сообщений оператору, пауз между командами. Применение такой технологии

ведет к снижению суммарного времени, затрачиваемого на поиск и выдачу отдельных команд управления, и повышению надежности часто повторяемых «ручных» операций управления полетом.

В связи с расширением и усложнением спектра задач, выполняемых современными КА, усложняется и процесс их управления. В связи с этим, актуальной является задача повышения эффективности языковых и инструментальных средств управления КА.

Работа посвящена повышению эффективности и надежности процессов подготовки сценариев управления космическими аппаратами, в частности, исследованию, проектированию и разработке языковых и инструментальных средств создания и исполнения сценариев управления космическими аппаратами.

Цель работы – повышение эффективности и надежности процесса подготовки сценариев управления космическими аппаратами за счет новых языковых и инструментальных средств подготовки и исполнения сценариев управления КА.

Задачи работы:

- анализ специфики командного метода управления КА;
- формирование требований, определяемых предметной областью, к языковой и инструментальной поддержке командного метода управления КА;
- создание языковых и инструментальных средств, обеспечивающих повышение эффективности процесса подготовки сценариев управления КА.

Объектом исследования является процесс управления КА.

Предметом исследования являются языковые и инструментальные средства создания и исполнения сценариев управления КА.

Методы исследования.

В диссертационной работе использовались теория языков и формальных грамматик, теории разработки трансляторов. В качестве средства моделирования предметной области применялись UML-диаграммы. Для описания синтаксиса языка программирования использовались расширенные формы Бэкуса-Наура (РБНФ).

В экспериментальной части применялись методы синтаксического анализа и компиляции, методы объектно-ориентированного и событийно-ориентированного программирования.

Научная новизна:

– на основе анализа предметной области предложена модель предметной области, включающая структуру командного метода управления КА, и определяющая функциональный состав языковых и инструментальных средств управления КА;

– разработан предметно-ориентированный язык программирования, повышающий эффективность разработки сценариев управления КА;

– разработаны инструментальные средства подготовки и выполнения сценариев управления КА, обеспечивающие повышение надежности и эффективности поддержки командного метода управления КА.

Положения, выносимые на защиту:

– модель предметной области, на основе которой были сформулированы требования к языковым и инструментальным средствам управления КА;

– предметно-ориентированный язык создания сценариев управления КА;

– инструментальные средства, обеспечивающие повышение эффективности подготовки и выполнения сценариев управления.

Практическая ценность результатов. В рамках выполнения исследования был спроектирован язык управления КА «Дельта». Помимо этого, были разработаны следующие программные продукты:

– интерпретатор, поддерживающий выполнение сценариев управления в реальном времени в ручном и автоматизированном режимах во время сеанса связи с КА, что обеспечивает практическое использование полученных результатов;

– библиотека стандартных сценариев управления КА, для создания наборов сценариев управления перспективными КА;

– программная система, обеспечивающая инструментальную поддержку процессов создания и выполнения сценариев управления КА, а также конвертер, в

автоматизированном режиме преобразующий эксплуатационную документацию по управлению КА в набор сценариев управления.

Опыт эксплуатации разработанных языковых и инструментальных средств создания и исполнения сценариев управления КА показал, что их применение минимизирует количество ошибок в сценариях, так как исключает ввод текстов сценариев управления вручную. Также снижается суммарное время, затрачиваемое на подготовку сценариев управления КА.

Предложенный подход к организации языковых и инструментальных средств создания и исполнения сценариев управления КА применим для автоматизации процесса управления КА различного назначения для предприятий ракетно-космической отрасли.

Апробация и внедрение результатов. Основные положения диссертационной работы докладывались и обсуждались на следующих конференциях:

– XIX Научно-практическая конференция на тему «Научно-практические аспекты совершенствования управления КА и информационного обеспечения запусков КА», г. Краснознаменск, 2009 (диплом за лучший доклад, представленный на конференции);

– Научно-техническая конференция молодых специалистов ОАО ИСС «Разработка, производство, испытания и эксплуатация космических аппаратов и систем», г. Железногорск, 2011 (диплом за лучший доклад, представленный на конференции);

– V Общероссийская молодежная научно-техническая конференция «Молодежь. Техника. Космос», Балтийский государственный технический университет «ВОЕНМЕХ» имени Д. Ф. Устинова, г. Санкт-Петербург, 2013г. (диплом за лучший доклад на секции «Системы управления и информационные системы»);

– III Научно-техническая конференция молодых специалистов ОАО «ИСС», г. Железногорск, 2014 (диплом за I место на секции «Средства выведения, управления и эксплуатация космических аппаратов и систем»);

– Международная научная конференция «Решетневские чтения», ФГБОУ ВПО «Сибирский государственный аэрокосмический университет имени академика М.Ф. Решетнева», г. Красноярск, 2015;

– IX ежегодная научно-техническая конференция молодых ученых и специалистов, Центр управления полетами ФГУП ЦНИИмаш, г. Королев, 2019 (диплом лауреата II степени).

Практическое применение разработанных программных продуктов подтверждено актом о внедрении (приложение Б).

Публикации. По теме диссертационной работы опубликовано 14 работ, из них 3 в изданиях Перечня ВАК. Получено свидетельство о государственной регистрации программы для ЭВМ.

Структура и объем диссертации.

Диссертационная работа состоит из введения, 4 глав, заключения, списка литературы, включающего 81 наименование, списка сокращений и двух приложений. Работа изложена на 127 листах машинописного текста, содержит 32 рисунка, 10 таблиц.

Краткое содержание работы.

Во введении представлена актуальность работы, цель и задачи исследования, раскрыта научная новизна, теоретическая и практическая значимость исследования и приведено краткое содержание работы.

В первой главе проведен анализ специфики предметно-ориентированных языков управления КА, выделены общие функции, обеспечивающие языковую поддержку процесса управления КА.

Выделены недостатки существующих программных средств и языков управления в плане применения, обоснована необходимость в разработке нового языка и инструментальных средств управления.

На основании анализа достоинств и недостатков существующих средств управления КА были сформулированы требования, предъявляемые к перспективным языковым и инструментальным средствам управления КА, подразумевающие использование предметно-ориентированного языка управления

в качестве средства автоматизации выдачи команд на КА из центра управления полетами (ЦУП).

Во второй главе проводится анализ процесса управления КА, на основе которого предложена структура командного метода управления КА, а также модель предметной области, описывающая организацию сценария управления для поддержки командного метода управления КА. Формирование такой модели позволило определить основные требования к предметно-ориентированному языку программирования и его инструментальной поддержке. Сформирован набор сценариев, с помощью которых осуществляется библиотечная поддержка предложенной модели процесса управления.

В третьей главе предложен предметно-ориентированный язык управления КА «Дельта», синтаксис которого описан с помощью РБНФ.

Разработанный язык управления поддерживает следующие функции: выдачу команд управления КА, проверку выполнения заданного условия, паузы, печать текстовых сообщений оператору; комментарии, содержащие пояснения к тексту программы, циклы, смену декодера. Многие ключевые слова, используемые в языке, совпадают с терминами предметной области.

В «Дельте» поддерживаются как русский, так и английский варианты написания операторов языка для использования в отечественных и международных проектах. Показано соответствие разработанного языка ранее выделенному перечню требований к языкам такого типа.

В четвертой главе предложена инструментальная система создания и исполнения сценариев управления, а также система автоматизации создания сценариев управления на основе анализа эксплуатационной документации. Представлен интерфейс пользователя разработанной системы, приведена структура разработанных программ, особенности реализации предложенных алгоритмов.

Приведены результаты сравнения характеристик существующих и предложенных языковых и инструментальных средств управления КА, применяемых на базе АО «ИСС» по следующим критериям: время создания

наборов сценариев управления, количество строк в итоговых сценариях управления, количество символов в итоговых сценариях управления для разработанных и существующих программных средств.

1 ЯЗЫКОВЫЕ СРЕДСТВА УПРАВЛЕНИЯ КОСМИЧЕСКИМИ АППАРАТАМИ

1.1 Применение языковых средств на этапах разработки и эксплуатации космических аппаратов

Очевидные преимущества применения языков программирования для автоматизации различных процессов управления определили широту их использования на предприятиях ракетно-космической отрасли как на этапе разработки, так и на этапе эксплуатации КА.

На этапе разработки языковые средства управления применяются для автоматизации проведения наземных испытаний КА.

На этапе эксплуатации языки управления применяются как для выдачи команд на КА непосредственно с Земли, так и для составления полетных заданий для исполнения вычислительным комплексом, расположенным на борту КА.

Проблема выбора языковых средств решается несколькими способами.

1. По типу используемого языка программирования можно выделить два подхода:

- использование универсального языка программирования;
- использование предметно-ориентированного языка программирования.

2. По количеству языков автоматизации также можно выделить два подхода:

- использование единственного языка программирования для проведения испытаний и управления КА в полете;
- использование нескольких различных языков программирования для решения задач испытаний и управления КА.

Примеры использования указанных технологий автоматизации:

- в NASA используется язык GOAL [9] для наземных испытаний КА, и язык STOL [6] для управления КА на орбите;

- в ESA для испытаний применяются языки ETOL и TCL/TK [9], для управления – язык PLUTO [6];
- в компании Integral Systems для управления и испытаний КА используется единственный язык STOL [9];
- в компании Thales Alenia Space используется язык SCOPE FCP DSL [10] для проведения испытаний, управления КА на орбите, а также создания полетных заданий для КА;
- в компании GMV применяется универсальный язык программирования Python в качестве языка наземных испытаний и языка управления КА из ЦУП [11 – 17];
- в РКК «Энергия» используется язык наземных испытаний «Диполь» [18], а также семейство предметно-ориентированных языков составления полетных заданий для исполнения на борту КА [19];
- в АО «Информационные спутниковые системы» применяется предметно-ориентированный язык испытаний «Диполь-6» [20], а также предметно-ориентированный язык управления КА в полете из ЦУП ЯОТР [21, 22].

Подход, при котором используются несколько предметно-ориентированных языков для решения различных задач испытаний и управления КА в полете обладает следующими преимуществами:

- наличие специализированных инструментальных и языковых средств, разработанных непосредственно для выполнения целевой задачи;
- хорошая читаемость текста сценариев управления вследствие применения терминологии предметной области в качестве ключевых слов языка;
- ориентированность на специалистов предметной области, не обладающих навыками программирования;
- быстрое освоение языковых и инструментальных средств специалистами предметной области;
- отсутствие зависимости от программного обеспечения, разрабатываемого третьей стороной.

Данные преимущества делают подход, при котором используются несколько предметно-ориентированных языков для решения задач производства и эксплуатации КА в полете, более перспективным.

1.2 ПО управления космическими аппаратами

1.2.1 Состав ПО управления

Состав программного обеспечения (ПО) управления КА стандартно включает в себя следующий набор компонент (рисунок 1.1):

- общесистемное программное обеспечение (ОСПО);
- специальное программное обеспечение (СПО).

ОСПО управления КА включает в себя следующие компоненты:

- операционные системы;
- системы управления базами данных;
- антивирусные пакеты.

СПО управления КА включает в себя следующие компоненты [22]:

- баллистическое программное обеспечение, автоматизирующее баллистические и навигационные расчеты;
- телеметрическое программное обеспечение, автоматизирующее распознавание, обработку и отображение поступающей телеметрии;
- программное обеспечение управления КА, автоматизирующее процесс выдачи управляющих воздействий на КА;
- программный имитатор КА, имитирующий логику работы КА и средств НКУ, предназначенный как для отладки всего программного обеспечения управления, так и для обучения персонала процессу управления КА;
- программное обеспечение управления средствами НКУ, позволяющее производить настройки КИС: параметры радиолинии (применяемые, в том числе для защиты передаваемых данных от несанкционированного доступа),

целеуказания для наведения антенн на КА и его сопровождения при полете по орбите, выбираемый комплект приборного состава.

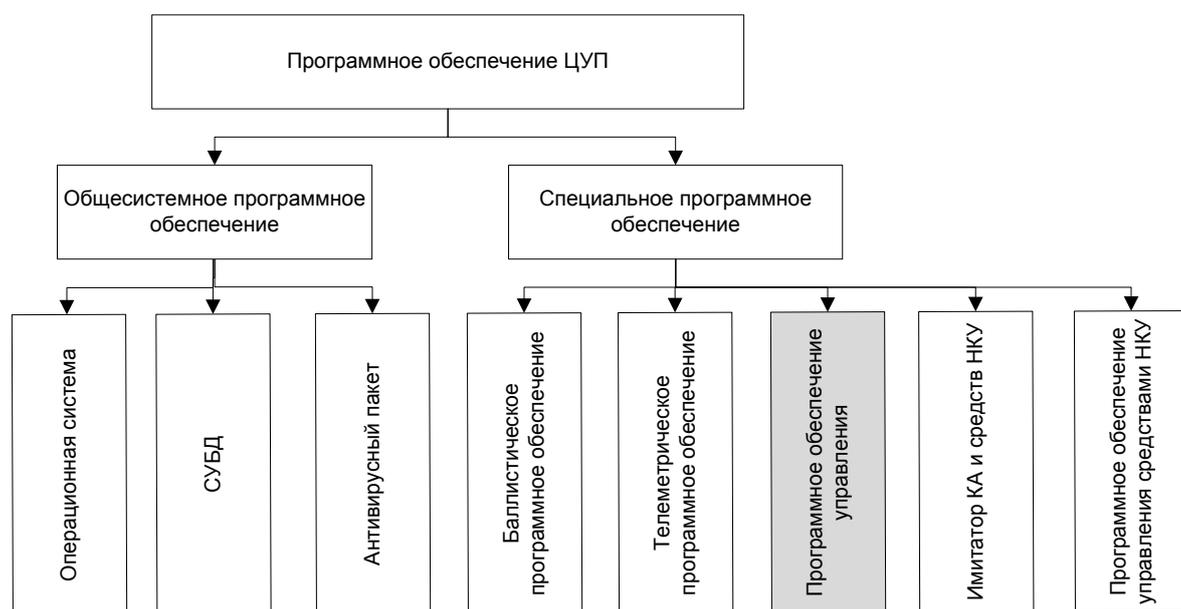


Рисунок 1.1 – Структура ПО управления КА

В некоторых случаях функционально разные (из перечисленных) задачи управления КА могут выполняться одним ПО.

1.2.2 Принципы построения ПО управления

Примеры ПО управления КА [5, 6, 8]: СПО ПКПО (АО «ИСС»), hifly (GMV), SCOS-2000 (ESA), ICS (Interface & Control Systems), EPOCH (Kratos Integral Systems International), OASIS_CC (LASP), OS-Comet (Harris), InControl-NG (L-3 Telemetry-West), Eclipse (Raytheon Company), SuperMOCA (Jet Propulsion Laboratory), CGS (Airbus Defence & Space), Astrium (Astrium), SMACS (NASDA) и другие.

Подробные технические описания вышеуказанных систем в открытом доступе отсутствуют, однако, на основе данных, являющихся доступными, можно сделать выводы об общих чертах программного обеспечения управления КА:

1. Сходная функциональность, связанная с необходимостью выдачи команд, отображения и анализа поступающей телеметрии, управления средствами наземного комплекса управления (НКУ):

- планирование действий по управлению КА и средствами НКУ;
- формирование и выдача команд управления КА и средствами НКУ в шифрованном/открытом виде;
- протоколирование выданных управляющих воздействий;
- обработка данных, поступающих от средств НКУ и бортовой аппаратуры спутника в составе ТМИ;
- репликация данных (синхронизация данных, содержащихся в БД ЦУП и резервного ЦУП, при наличии такового).

2. Клиент-серверная архитектура. Программное обеспечение управления непосредственно взаимодействует с программным обеспечением средств НКУ – позволяет отправлять команды в специализированном формате, получать и анализировать телеметрические данные.

3. В качестве средства хранения исходных данных, а также статистики управления используется СУБД.

4. Модульность, с настраиваемым числом модулей, в зависимости от особенностей конкретного проекта и пожеланий заказчика.

5. Кроссплатформенность (поддержка выполнения на базе различных операционных систем, например, Unix, Linux, Windows).

6. Открытость (предоставление заказчику различных API для доработок и расширения предоставляемого ПО).

7. Поддержка стандартов, существующих в космической области: семейства стандартов CCSDS (Consultative Committee for Space Data Systems, Международного Консультативного Комитета по космическим системам передачи данных), семейства стандартов ESA PSS.

8. Автоматизированное обнаружение и поддержка обработки нештатных ситуаций в процессе управления КА.

9. Применение различных языков управления для автоматизации выполнения типовых процедур управления (сценариев сеансов управления, описанных на специализированном скриптовом языке). Режимы выполнения процедур управления:

– «автоматический» (процедура выполняется полностью автоматически за исключением ситуаций возникновения ошибок, сбоев, или запроса данных у пользователя);

– «полуавтоматический» – выполнение процедуры приостанавливается после выполнения каждого действия до подтверждения пользователем продолжения выполнения;

– «ручной» – подтверждение оператора требуется для выполнения каждой отдельной строки сценария управления (или оператора языка, или выдачи команды управления).

1.2.3 Примеры ПО управления

1.2.3.1 ЕРОСН

ПО управления КА разработки Kratos Integral Systems [23, 24, 25]. Применяется на стадиях испытаний и эксплуатации КА. Предназначено для работы под ОС семейства Unix, Linux. Поддерживает стандарты CCSDS и TDM. Автоматизация процесса управления производится с помощью процедурного языка STOL, а также языков управления производителя КА, например, СЕСІL.

Структура основных компонентов системы ЕРОСН представлена на рисунке 1.2. Конфигурация системы может быть дополнительно расширена модулями различного назначения.

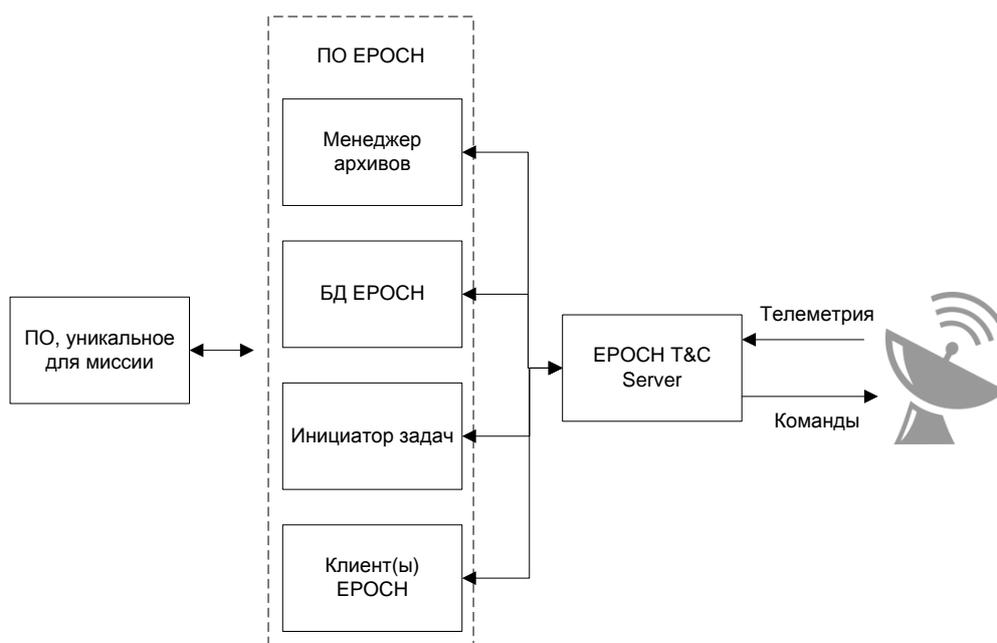


Рисунок 1.2 – Структура программного обеспечения ЕРОСН

ЕРОСН T&C Server обеспечивает обработку телеметрии, получаемой с КА, управление КА, а также управление наземным оборудованием.

«Инициатор задач» (Task Initiator) обеспечивает автоматизацию запуска различных повторяющихся задач в процессе управления КА. Задачи могут быть запущены на основе плана или по наступлению некоторого события. Примеры автоматизируемых задач: регулярное измерение дальности, синхронизация БД, управление файлами отчетов программы, управление средствами НКУ.

«Клиенты ЕРОСН» выполняют функции отображения текущего состояния, выдачи команд управления и управления памятью КА и средствами НКУ. ПО поддерживает возможность управления несколькими КА одновременно. Позволяет формировать отчеты в программах Word и Excel. Предоставляются API для взаимодействия с ПО разработки третьей стороны.

«БД ЕРОСН» – приложение, обеспечивающее доступ к БД, и обеспечивающее ввод и редактирование команд, определений телеметрических данных, данных о правах доступа пользователей системы.

Менеджер архивов выполняет задачи архивирования и просмотра данных различного происхождения, полученных в ПО управления.

1.2.3.2 OS/COMET

OS/COMET – ПО управления КА разработки компании Harris [26, 27, 28]. Структура OS/COMET проиллюстрирована рисунком 1.3, назначение отдельных модулей системы описано далее.

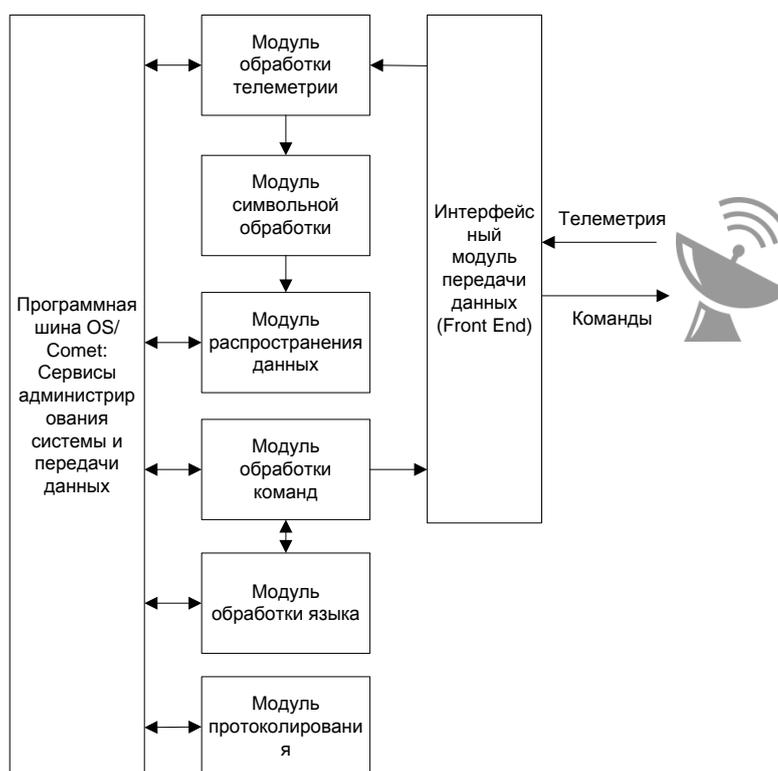


Рисунок 1.3 – Структура программного обеспечения OS/COMET

«Программная шина» OS/COMET выполняет следующие функции:

- поддерживает актуальные данные обо всех процессах и ресурсах OS/COMET;
- предоставляет возможность обмена сообщениями между процессами OS/COMET (системными и разработанными пользователем) на одном узле и между различными узлами.

Модуль символьной обработки предназначен для работы с текстовой информацией, например, описанием команд управления, телеметрическими данными и другими параметрами системы.

Модуль распространения данных поддерживает актуальность данных в системе, включая серверную и клиентские части.

Модуль обработки телеметрии предназначен для получения отдельных ТМ-параметров из телеметрических данных. Форматы парсинга телеметрии определяются в специализированном файле.

Модуль обработки команд предназначен для передачи команд на целевое устройство. Перечень команд для конкретного типа оборудования описан в файле определенной структуры.

Модуль обработки языка предназначен для разбора выражений языка OS/COMET Control Language (CCL), а также других специализированных текстовых файлов приложения.

Модуль протоколирования обеспечивает механизм для записи необработанных телеметрических данных и логирование значимых событий.

Все компоненты OS/COMET взаимодействуют между собой через программную шину OS/COMET.

Дополнительные модули OS/COMET, поставляемые компанией Harris:

- FA/Tool – модуль распознавания и автоматической обработки сбоев;
- OA/Tool – интегрированный автоматизированный план мероприятий ЦУП;
- SIM/Tool – модуль имитации любых данных, использующихся в ходе разработки ЦУП;
- GC/Tool – модуль мониторинга и управления наземным оборудованием;
- MEM/Tool – модуль управления бортовой памятью КА.

1.2.3.3 hifly

ПО hifly [13, 14] – система мониторинга и управления группировками КА разработки компании GMV. Включает в себя клиентскую и серверную части. Построена вокруг ядра, обеспечивающего базовые возможности мониторинга и

управления КА, и переменного числа компонентов, расширяющих данные возможности.

ПО hifly состоит из следующих компонент (рисунок 1.4):

- Views – открытый клиент просмотра телеметрии. Обеспечивает гибкий доступ к телеметрии и конфигурируемое представление данных. Запускается под ОС Windows, Linux и Unix.

- Archiva – база данных обработанной телеметрии, содержащая данные за весь срок эксплуатации КА.

- SatDB – модуль управления БД, поддерживает редактирование и проверку корректности данных.

- autofly – специализированный компонент, автоматизирующий выполнение и планирование процедур управления КА.

Дополнительные компоненты:

- DPWizard – модуль вычисляемых параметров (Derived Parameter), обеспечивает поддержку синтезированных/вычисляемых параметров, что ведет к уменьшению затрат на генерацию вычисляемых параметров и увеличение безопасности эксплуатации.

- SyncBridge – модуль синхронизации данных управления между различными площадками. Также может быть использован для передачи данных управления в нерабочую среду, например, для обучения, или испытаний ПО.

- FleetMan – компонента, обеспечивающая расширенные возможности управления полетами для средних и больших группировок КА, централизует обработку событий и внештатных ситуаций для всей системы, обеспечивает общий обзор текущего состояния системы.

- eIBook – электронный журнал учета, обеспечивающий централизованную обработку и хранение всех событий в системе.

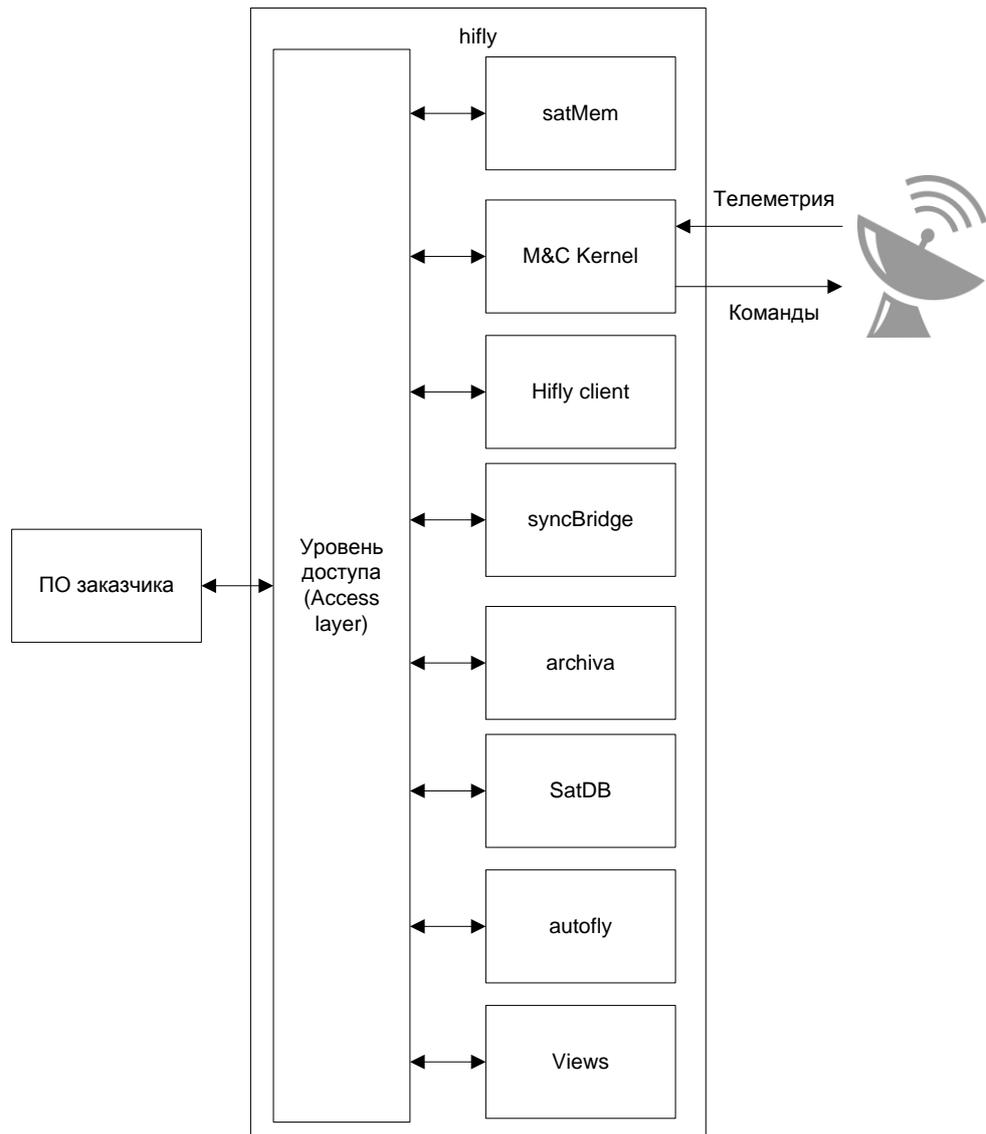


Рисунок 1.4 – Структура программного обеспечения hifly

MIDAS (Mission Data Analysis System) – система, обеспечивающая анализ и обработку данных миссии в оффлайн режиме. Гарантирует полноту и целостность хранилища данных.

E2EE – модуль, имитирующий функционирование элементов космической системы (КА, КИС, ЦУП, и др.). Поддерживает существующие стандарты и интерфейсы (CCSDS, NCTRS, и т.д.).

TCRbridge – модуль управления связью между ЦУП и приемными/передающими антеннами. Поддерживает различные протоколы взаимодействия.

1.2.3.4 InControl

Комплекс InControl – кроссплатформенное программное обеспечение управления и мониторинга КА и средств НКУ во время заводских испытаний и эксплуатации. Разработано компанией L-3 Telemetry-West [29, 30]. Может выполняться на базе ОС Windows, Solaris, Linux.

Программный комплекс InControl разработан с соблюдением принципов открытой архитектуры, обеспечивающих передачу данных клиентам различными способами: специализированные API, специализированные функции, написанные пользователем, открытые форматы архивов. Поддерживается обеспечение доступа к функциям клиента через веб-браузер: к управлению конфигурацией программного обеспечения, к архивам данных.

ПО InControl поддерживает выполнение сценариев на нескольких скриптовых языках управления, например, языке JAS, и обладает возможностью добавлять другие языки.

1.2.3.5 SMACS

ПО SMACS (Spacecraft Management and Control System) [31] разработки NASDA (National Space Development Agency of Japan), с 2003г. вошедшей в состав JAXA (Japan Aerospace Exploration Agency) поддерживает выполнение следующих задач: планирование, мониторинг, управление, анализ текущего состояния КА. Структура ПО SMACS представлена на рисунке 1.5.

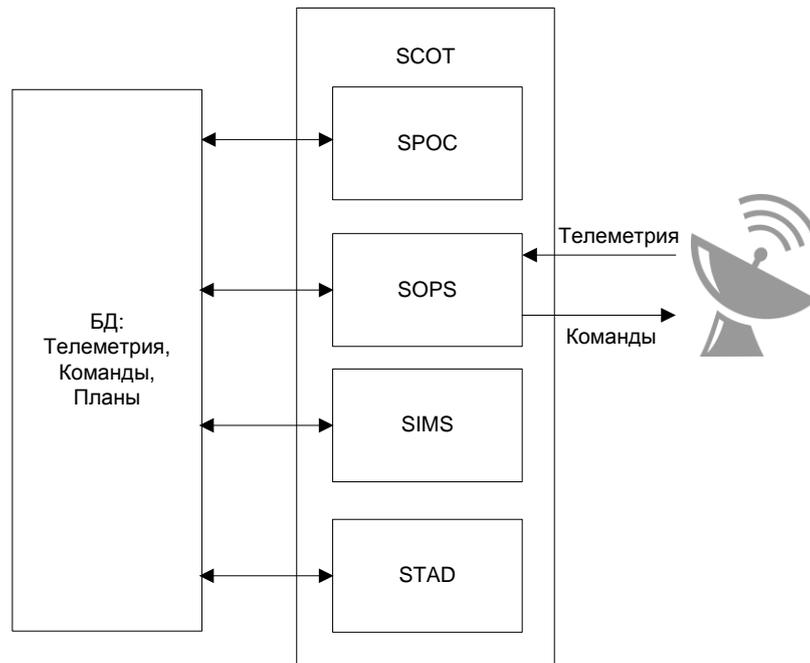


Рисунок 1.5 – Структура программного обеспечения SMACS

ПО «SMACS» состоит из следующих модулей:

- SIMS (Spacecraft Information Management Subsystem) – обеспечивает формирование и проверку данных в БД (например, процедур управления и телеметрических данных) для других модулей;
- SOPS (Spacecraft Operation Planning Subsystem) – обеспечивает планирование операций по эксплуатации КА и наземной системы слежения;
- SPOC (Spacecraft Processing and Operation Control subsystem) – обеспечивает обработку телеметрии и данных управления;
- STAD (Spacecraft Telemetry data Analyzing and Drawing subsystem) – обеспечивает архивацию, анализ и отображение телеметрии и других данных управления;
- SCOT (Spacecraft Control and Operation Terminal subsystem) – предназначен для мониторинга и управления КА;
- БД, содержащая системную информацию (план работы с КА, тип КА и другие) и оперативную информацию (телеметрия, протоколы выдачи команд и другие).

1.2.3.7 Mois

Mois – ПО управления КА разработки компании Rhea, Бельгия [32].

Структура программного обеспечения Mois (рисунок 1.6):

- Writer – модуль, позволяющий создавать и редактировать процедуры управления и планы работы;
- Flowcharter – модуль, позволяющий создавать, редактировать и отображать структуру процедур управления в виде блок-схем;
- DB – модуль, отвечающий за создание, импорт и редактирование базы исходных данных по управлению КА;
- Function Editor – модуль, который обеспечивает дружественный интерфейс пользователя для создания и поддержки функций и директив для использования в процедурах управления;
- Supervisor – модуль управления средой выполнения процедур управления;
- Validator – модуль тестирования процедур управления, а также хранения результатов тестирования;
- Test Harness – эмулятор системы управления/имитатор для тестирования процедур управления;
- Scheduler – модуль планирования работ по управлению КА и наземными станциями;
- Big board – модуль отображения текущих событий из планов работы с КА, применяется как до, так и после запуска КА;

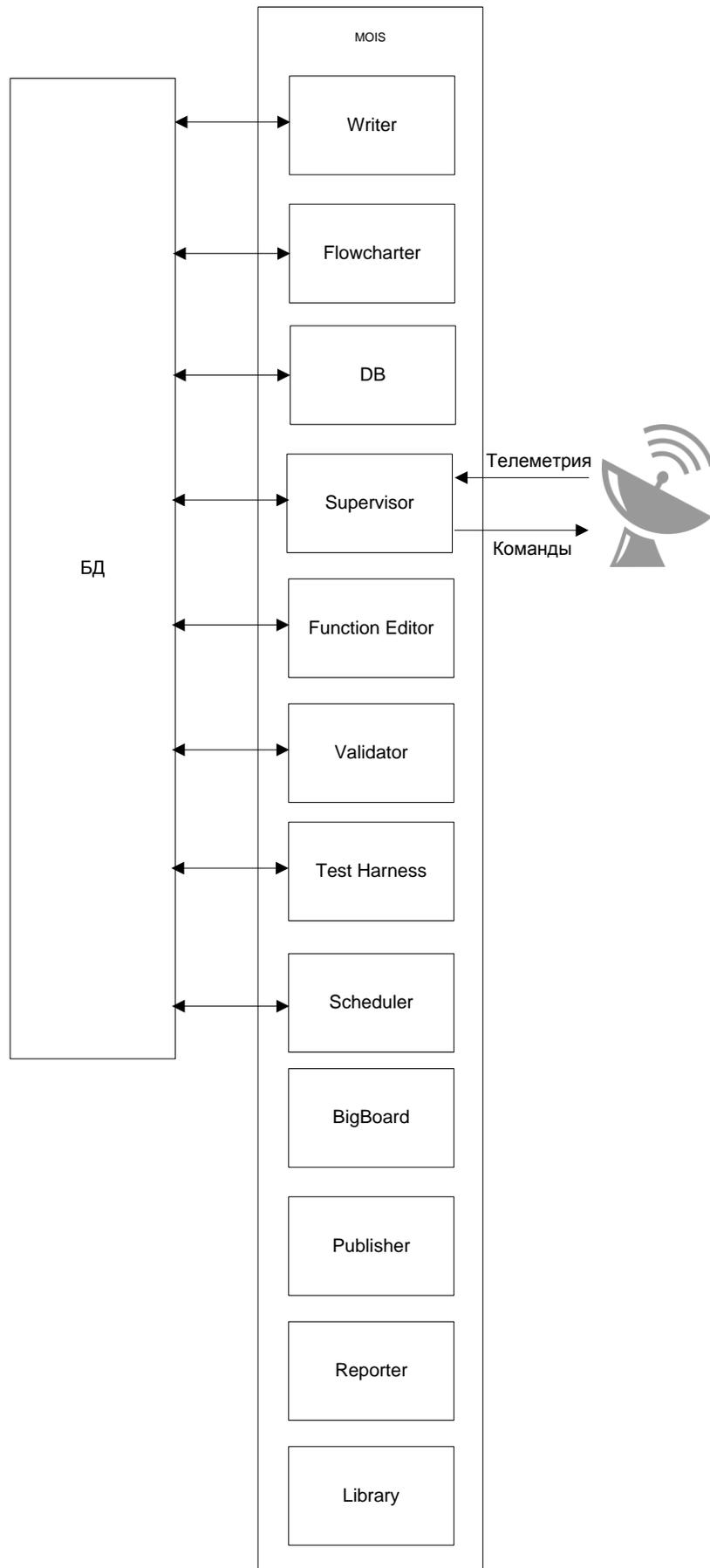


Рисунок 1.6 – Структура программного обеспечения MoIS

- Publisher – модуль разработки и редактирования документации по управлению КА;
- Reporter – модуль, последовательно сохраняющий описание, анализ и решение возникающих проблем; также выполняет проверку структуры и анализ набора процедур управления;
- Library – модуль, отвечающий за обзор, администрирование и управление конфигурацией библиотек.

1.2.3.8 ПО управления космическими аппаратами АО «Информационные Спутниковые Системы»

Управление КА средствами АО «ИСС» реализуется набором следующего СПО (рисунок 1.7): СПО ПКПО, СПО ИТО, СПО САО [33].

СПО ИТО (Специальное Программное Обеспечение Информационно-Телеметрического Обмена) предназначено для приёма, оперативной оценки и анализа ТМИ, её отображения, документирования и архивирования. Реализует послесеансную обработку и анализ архивированной ТМИ.

СПО САО (Специальное Программное Обеспечение Системы Автоматизированного Обмена) обеспечивает сетевое взаимодействие программ управления КА.

СПО ПКПО (Специальное Программное Обеспечение Планирования и Командно-Программного Обеспечения) – кроссплатформенное ПО управления КА. СПО ПКПО обеспечивает автоматизированный расчёт, формирование, регистрацию, документирование и выдачу технологических данных, команд и программ управления на КА и средства НКУ.

СПО ПКПО [34] состоит из следующих компонент (рисунок 1.7):

- КП ПДС предназначен для планирования работ по управлению КА на произвольный интервал времени;
- КП ФСИ служит для заведения и корректировки заданий КПИ в БД;

– КП ПСУ обеспечивает проведение оперативного сеанса управления КА, включая отработку как заранее подготовленных сценариев управления, так и управляющих воздействий, вводимых оператором в реальном масштабе времени, регистрацию и обработку квитанций, поступающих от средств НКУ и бортовой аппаратуры спутника, протоколирование действий оператора, выдаваемых управляющих воздействий, квитанций на них, возникающих нештатных ситуаций;

– КП ИСС обеспечивает отображение и фильтрацию статистической информации о проведенных сеансах управления за весь срок эксплуатации КА;

– КП ПИД обеспечивает заведение исходных данных для расчета программ (последовательности команд) управления спутником.

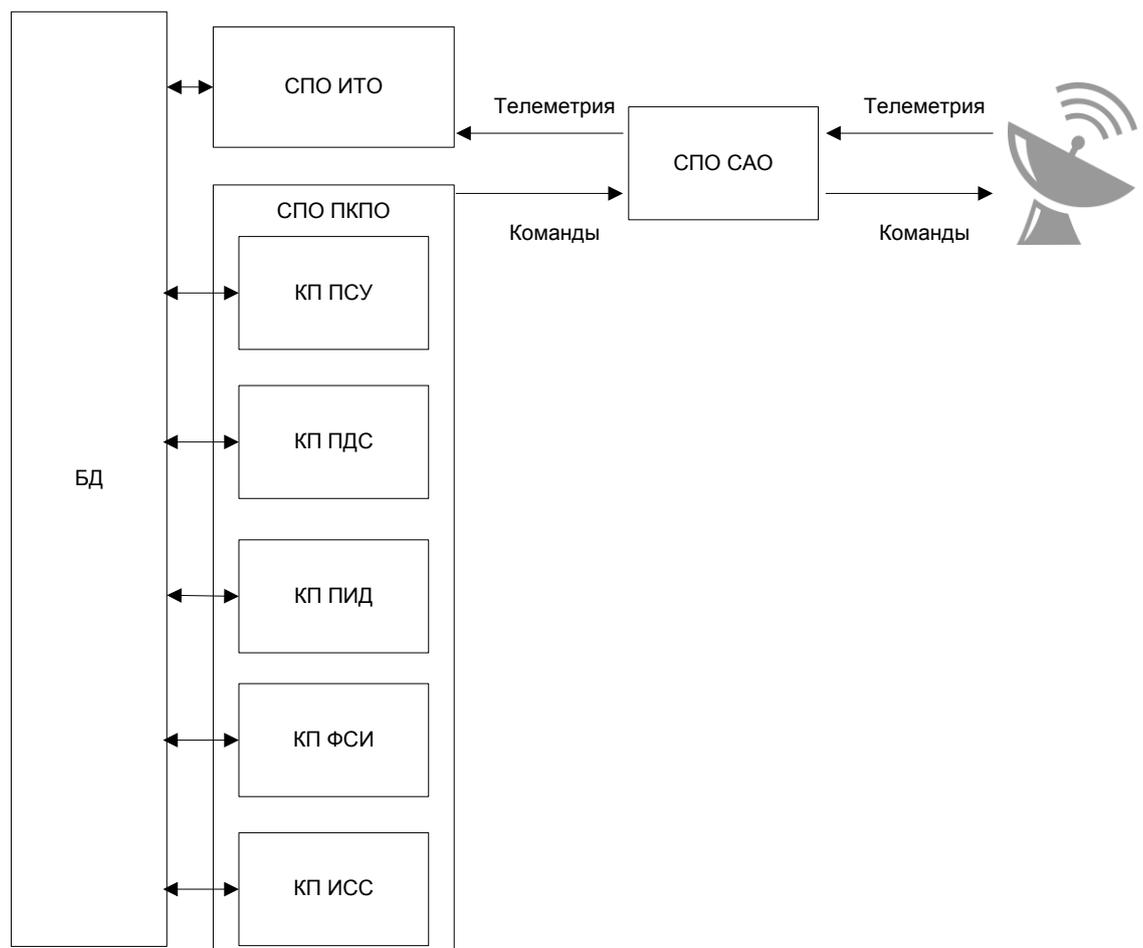


Рисунок 1.7 – Структура программного обеспечения управления КА АО «ИСС»

Исходные данные управления (перечень команд управления, сценариев управления, ТМ-параметров, и другие), а также статистика управления хранятся в БД, с которой взаимодействуют компоненты СПО управления КА.

1.3 Языковые средства управления космическими аппаратами

1.3.1 Возможности языков управления

Функциональные возможности языков управления КА обоснованы необходимостью решения задач управления КА с Земли, путем выдачи команд из ЦУП с использованием наземных радиосредств, и включают, но не ограничиваются, следующим набором возможностей:

- выдача команд управления на КА;
- анализ текущего состояния подсистем КА на основе анализа поступающих по радиоканалу с борта КА телеметрических параметров;
- выдача сообщений оператору согласно логике, заложенной в сценарии;
- паузы между командами;
- вызов другого сценария управления из текущего сценария;
- комментарии к тексту сценария.

1.3.2 Примеры языков управления

Примеры языков автоматизации управления КА в ходе сеанса связи с КА из ЦУП [5, 6, 8]: ЯОТР (АО «ИСС»), Python (GMV), PLUTO (ESA), SCL (Interface & Control Systems), STOL (NASA), TAO (Kratos Integral Systems International), CSTOL (LASP), CCL (Harris), JAS (L-3 Telemetry-West), Cecil (Raytheon Company), CIL (Jet Propulsion Laboratory), TOPE/tcl (ESA), UCL (Airbus Defence & Space), Elisa (Astrium), PIL (Astrium).

В качестве формы представления языков управления КА применяются следующие: графическая, табличная и текстовая.

Графическая форма представления (блок-схемы различного типа) используется в системе Rhea [32].

Табличная форма представления (сценарий представлен в виде таблицы со столбцами фиксированного назначения) применяется в следующих языках: Arges [35, 36], язык управления компании NAZDA [31].

Текстовая форма (сценарий управления КА представляется в виде набора символов некоторого языка, написанный по определенным правилам) наиболее широко распространена. Примеры текстовых языков: Python (GMV), PLUTO (ESA), SCL (Interface & Control Systems), STOL (NASA), TAO (Kratos Integral Systems International), CSTOL (LASP), CCL (Harris), JAS (L-3 Telemetry-West), Cecil (Raytheon Company), CIL (Jet Propulsion Laboratory), TOPE/tcl (ESA), UCL (Airbus Defence & Space), Elisa (Astrium), PIL (Astrium).

Для текстовой формы представления характерно использование как языков программирования общего назначения (Python), так и специализированных языков, разработанных специально для автоматизации процесса управления КА (STOL, TAO, CSTOL, CCL, JAS, ICL, Btscript, Cecil, CIL, TOPE/tcl, UCL, Elisa, PIL).

1.3.2.1 Python

Язык Python – переносимый, высокоуровневый, объектно-ориентированный универсальный язык программирования, применяемый для записи сценариев управления КА, выполняемых в ходе проведения сеанса связи. Применяется для автоматизации управления КА с помощью ПО hifly [11 – 17], разработанного компанией GMV.

Достоинства использования языка Python в качестве языка управления КА:

- поддержка параллельного выполнения сценариев управления КА;
- широкие возможности языка (структуры данных, управление памятью и другие);

- наличие библиотек для выполнения различных задач: математических расчетов, работы с графикой и других;
- наличие множества готовых инструментальных средств разработки, тестирования, и других.

Ограничения применения языка Python для решения задач управления КА:

- отсутствие предметной ориентации делает язык слишком сложным для изучения специалистами, не обладающими навыками программирования;
- читаемость кода хуже, чем у предметно-ориентированных языков;
- отсутствие русскоязычного синтаксиса;
- зависимость от инструментальных средств, разработанных третьей стороной.

1.3.2.2 PLUTO

Язык PLUTO, применяющийся в ПО SCOS Mission Control System, разработки компании SciSys (Англия, Германия) [37, 38], а также в ПО STEPS, разработанным институтом INPE, Бразилия [42] – высокоуровневый язык программирования, поддерживающий циклы и арифметические операции. Процедура управления, созданная с использованием данного языка, должна быть разделена на следующие части: раздел объявлений; раздел условий старта выполнения процедуры (по наступлению конкретного времени или выполнению ряда условий); тело процедуры; раздел проверок оборудования (если в ходе выполнения данного раздела произошел сбой, то выполнение процедуры останавливается); раздел пост-условий (проверка того, что процедура была выполнена правильно). Пример процедуры управления представлен на рисунке 1.8.

```

procedure Sun_Vector_Determination
Declare
  variable real X_Axis := 0.0 deg;
  variable real Y_Axis := 0.0 deg;
  variable real Z_Axis := 0.0 deg;
  variable real analog_value_out1 := 0.0 volts;
  variable real analog_value_out2 := 0.0 volts;
  variable real analog_value_out3 := 0.0 volts;
  variable real analog_value_out4 := 0.0 volts;
end declare
preconditions
  wait until switch_on of ACDH_COMPUTER = true;
end preconditions
main
  Simulate Environment Analog_Value_Out_1 of SUN_SENSOR_A of ACDH_SUBSYSTEM of SATELLITE;
  Simulate Environment Analog_Value_Out_2 of SUN_SENSOR_B of ACDH_SUBSYSTEM of SATELLITE;
  Simulate Environment Analog_Value_Out_3 of SUN_SENSOR_C of ACDH_SUBSYSTEM of SATELLITE;
  Simulate Environment Analog_Value_Out_4 of SUN_SENSOR_D of ACDH_SUBSYSTEM of SATELLITE;
  analog_value_out1 := Analog_Value_Out_1 of SUN_SENSOR_A of ACDH_SUBSYSTEM of SATELLITE;
  analog_value_out2 := Analog_Value_Out_2 of SUN_SENSOR_B of ACDH_SUBSYSTEM of SATELLITE;
  analog_value_out3 := Analog_Value_Out_3 of SUN_SENSOR_C of ACDH_SUBSYSTEM of SATELLITE;
  analog_value_out4 := Analog_Value_Out_4 of SUN_SENSOR_D of ACDH_SUBSYSTEM of SATELLITE;
  wait for 0.5 s;
  X_Axis := Sun_Vector_X_Axis of SUN_VECTOR_ALG of ACDH_COMPUTER of ACDH_SUBSYSTEM of SATELLITE;
  inform user "X axis result: " X_Axis;
  Y_Axis := Sun_Vector_Y_Axis of SUN_VECTOR_ALG of ACDH_COMPUTER of ACDH_SUBSYSTEM of SATELLITE;
  inform user "Y axis result: " Y_Axis;
  Z_Axis := Sun_Vector_Z_Axis of SUN_VECTOR_ALG of ACDH_COMPUTER of ACDH_SUBSYSTEM of SATELLITE;
  inform user "Z axis result: " Z_Axis;
end main
end

```

Рисунок 1.8 – Пример процедуры на языке PLUTO

Достоинства языка: применение терминологии предметной области в качестве ключевых слов языка управления; хорошая читаемость текста сценария управления; структура процедуры управления, позволяющая предопределить условия запуска процедуры на выполнение, и условия, когда выполнение процедуры считается успешным.

Ограничения языка: отсутствие русскоязычного синтаксиса, закрытый исходный код средств инструментальной поддержки и выполнения сценариев управления КА.

1.3.2.3 SCL

Язык SCL [39, 40, 41] применяется для автоматизации выдачи команд управления КА из ЦУП в ПО ASPEN, компания Jet Propulsion Laboratory, США, а также в ПО SCL, компания Interface & Control Systems, США. На SCL можно создавать как сценарии управления, так и правила управления. Правила предназначены для мониторинга внештатных ситуаций в подсистемах КА, при появлении которых предусматривается автоматический запуск соответствующего сценария. Сценарии управления предназначены для исполнения в ходе сеанса связи с КА для выполнения какой-либо целевой задачи. Сценарии могут выполняться параллельно, на основе приоритетов, назначаемых при создании сценария. Кроме функций, общих для всех языков управления, SCL поддерживает циклы и выражения присваивания. Примеры сценария и правила управления представлены на рисунках 1.9 и 1.10.

```
rule Detect_Second_Change
  subsystem none
  category pwr
  priority 21
  activation yes
  continuous yes

  if change (SGPCFIDA2) then

    msg "The beginning time is " & ticks

    spawn UpdateTwo now

  end if

end Detect_Second_Change
```

Рисунок 1.9 – Пример правила управления на языке SCL

```

script UpdateTwo

spawn BFSDataDispMajMode with SGPCAREA2, V98U2408C1 now
spawn BFSDataDispMajMode with SGPCAREA2, V98J2299C1 now
spawn BFSDataDispMajMode with SGPCAREA2, V98J2258C1 now

msg "Ending time for Update for UpdateTwo is " & ticks

end UpdateTwo

```

Рисунок 1.10 – Пример сценария управления на языке SCL

Достоинства языка: хорошая читаемость текста сценария управления; наличие системы приоритетов; возможность параллельного выполнения сценариев управления; возможность создания правил управления, при выполнении которых запускается определенный сценарий управления.

Ограничения языка: закрытый синтаксис, отсутствие поддержки русскоязычного синтаксиса, закрытый исходный код средств инструментальной поддержки и выполнения сценариев управления КА.

1.3.2.4 SMACS

Компанией NASDA, Япония разработан язык управления КА, предназначенный для составления и исполнения сценариев управления КА из ЦУП в составе ПО SMACS [31]. Помимо общих функций языка управления данный язык поддерживает обработку следующих событий: изменение конфигурации сети, смену используемой для управления КА станции. Процедура также может производить запись телеметрических данных и сигнализировать о возникших аномалиях. Пример процедуры управления представлен на рисунке 1.11.

STEP	TIME	PROCEDURE	OPERATING DATA	VALUE	RESULT
1		Setting System			
		1)connect station			
	2003-01-01 00:00:00		CONNECT:xx		OK/NG
2		Spacecraft Check			
		1)confirm telemetry			
	2003-01-01 00:01:00		TLM aaa > 10.0		OK/NG
			TLM zzz = "A ON"		OK/NG
3		Selective step			
3-1		TLM yyy>5			
		1)send command			
			CMD:a		OK/NG
3-2		TLM yyy>7			
		1)send command			
			CMD:b		OK/NG
3-3		TLM yyy<0			
		1)telemetry confirm			
			TLM zzz="B on"		OK/NG

Рисунок 1.11 – Пример процедуры, поддерживаемой ПО «SMACS», Япония

Достоинства языка: понятная форма представления, соответствие ключевых слов языка терминам предметной области.

Ограничения языка: отсутствие русскоязычного синтаксиса, процедура управления визуально занимает большую площадь, закрытый исходный код средств инструментальной поддержки и выполнения сценариев управления КА.

1.3.2.5 Ares

Компанией Kratos Integral Systems International, США, разработан язык управления КА, применяющийся в составе ПО ЕРОСН [6, 35, 36].

Сценарий управления представляет собой текстовый файл в виде таблицы с использованием символов табуляции в качестве разделителей, со следующими столбцами: номер шага, инструкция, команда, телеметрический параметр, метка, вызов функции. Позволяет вызывать функции из предоставляемой библиотеки функций. Помимо общих функций языка управления дополнительно поддерживается загрузка файлов с аргументами команд.

Достоинства языка: наличие библиотеки функций, поддержка загрузки файлов с аргументами команд.

Ограничения языка: закрытый исходный код средств инструментальной поддержки и выполнения сценариев управления КА.

1.3.2.6 ЯОТР

В 1983 г. на базе Института Прикладной математики им. Келдыша совместно с РКК «Энергия» был разработан предметно-ориентированный табличный язык «Диполь» для проведения наземных испытаний проекта «Энергия-Буран». В 1987 г. язык «Диполь-4» был внедрен в НПО ПМ для написания программ наземных испытаний КА различного назначения и в 1993 г. был изменен и доработан до версии «Диполь-5». Затем, в 1999 г. на основе языка «Диполь-5» был разработан язык ЯОТР путем исключения команд, не применимых для управления КА; табличная структура языка, а также написание отдельных операторов было сохранено. Язык применяется для управления и эксплуатации КА разработки АО «ИСС». В 2004 г. на основе языка «Диполь-5» была разработана очередная версия языка «Диполь-6» [20], которая применяется для комплексных электрических испытаний космических аппаратов производства АО «ИСС». Также язык «Диполь» применяется для электрических испытаний на базе РКК «Энергия» [18, 43].

Язык ЯОТР [44], применяемый для управления КА, представляет собой интерпретируемый скриптовый процедурный язык с текстовой нотацией. Сценарий управления представляет собой текстовый файл в виде таблицы с использованием символов псевдографики «|», со следующими столбцами: номер строки, тип строки, оператор языка, аргументы. Пример реального сценария из состава исходных данных по управлению КА представлен на рисунке 1.12.

○		1				ПРОГРАМ					
○		2				ВЫЗВАТЬ				_PK	
		3				40		&1			
○		4				ЕСЛИТО					
		5				#ГТС_ВЛ				0	
○		6				ПЕЧАТЬ					
Φ		7		Счетчик фраз		обнулен					
○		8				ИНАЧЕ					
○		9				ПЕЧАТЬ					
Φ		10		Счетчик фраз		не был изменен					
○		11				КЕСЛИТО					
○		12				КПРОГРАМ					

Рисунок 1.12 – Пример сценария управления на языке ЯОТР

Многолетний опыт эксплуатации языка ЯОТР в качестве языка управления КА выявил наличие в нем ряда ограничений.

1) Использование текстово-табличной нотации крайне неудобно, так как требует запоминания правил заполнения каждого отдельного поля, и различается от одного оператора к другому, что особенно неудобно, например, в условиях, когда необходимо создать сценарий управления за ограниченный промежуток времени (например, для обработки нештатной ситуации).

2) Не поддерживается английский язык написания сценариев управления, что является одним из требований международных контрактов.

3) Отсутствуют возможности для автоматизации выполнения отдельных функций, необходимых для проведения процесса управления современными КА.

Последний недостаток следует уточнить:

- невозможно использовать номер КА в качестве параметра, по значению которого меняется логика сценария управления для систем управления группировками КА;

- для ситуации множественного выбора не предусмотрено обработки случая, когда ни один из предложенных вариантов не подошел;

- не поддерживается использование телеметрических параметров в качестве аргументов операций (например, сравнения, ожидания, выбора из нескольких вариантов);

- отсутствует обработка ТМ-параметров из заранее сохраненного архива (например, для восстановления состояния бортового ретрансляционного комплекса в некоторое исходное состояние);
- невозможно использование дробных чисел в качестве параметров для проверки различных условий (например, принадлежности ТМ-параметра некоторому интервалу);
- не поддерживается возможность выдачи команды смены декодера средствами языка;
- использование лексических единиц, не соответствующих терминологии предметной области, что усложняет процесс запоминания правил написания сценариев управления.

1.3.2.7 Сопоставление языков управления космическими аппаратами

Результаты сопоставления перечисленных языков управления КА по общим характеристикам языков программирования (тип языка, используемая парадигма, возможность параллельного выполнения сценариев, используемая нотация) представлены в таблице 1.1. Анализ полученных данных показывает, что, несмотря на сходные функциональные задачи, связанные с выдачей команд на КА, реализация языков управления КА сильно различается:

- языки управления могут быть как общего назначения, так и специализированными (разработанными специально для автоматизации процесса управления КА);
- возможно использование процедурного и объектно-ориентированного стиля программирования;
- отдельные языки поддерживают только последовательное выполнение сценариев управления, другие – как последовательное, так и параллельное;
- для написания сценариев может применяться текстовое, табличное и графическое представление.

1.4 Требования, предъявляемые к языковым средствам управления

На основании достоинств и недостатков существующих языковых средств управления КА можно сформулировать следующий перечень требований к перспективным языковым и инструментальным средствам управления КА.

Требования к перспективным языковым средствам управления:

- перспективный язык управления должен относиться к классу специализированных языков, предназначенных для целей автоматизации процесса управления КА, что позволит упростить процесс обучения конечных пользователей;

- текстовая форма представления, что обеспечит хорошую читаемость даже объемных сценариев управления с нелинейной логикой выполнения, строгость, компактность записи, простоту хранения;

- использование лексических единиц, соответствующих терминологии предметной области, что облегчит процесс создания сценариев управления;

- поддержка как русскоязычного, так и англоязычного синтаксиса для соответствия требованиям международных контрактов.

Требования к перспективным инструментальным средствам управления КА:

- обеспечение проверки синтаксиса сценариев управления КА;

- ручной и автоматизированный режимы выполнения сценариев управления, что позволит выполнять как заранее созданные последовательности команд управления в соответствии с логикой, заложенной в сценарии управления, так и произвольно менять ее, выдавая команды управления в ручном режиме;

- кроссплатформенность, что обеспечит переносимость разрабатываемых инструментальных средств на базу операционных систем различного типа.

Выводы по главе 1

Проведенный анализ существующих предметно-ориентированных языков управления КА выявил, что они обладают сходной функциональностью,

связанной с необходимостью выдачи команд управления КА, анализом условных выражений, обработкой поступающей телеметрии, печатью сообщений оператору, и других. Также была выявлена значительная разница в реализации: языки управления могут быть как общего назначения, так и специализированными, процедурными и объектно-ориентированными, текстовыми, табличными и графическими, поддерживать только последовательное, и последовательное и параллельное выполнение.

Помимо этого были выявлены недостатки в языке управления КА, эксплуатируемом на базе АО «ИСС», связанные с тем, что язык не поддерживает изменившиеся требования предметной области.

Применение зарубежных аналогов языков управления является нецелесообразным ввиду несовпадения специфики решаемых задач управления КА, а также ориентированности на технологии управления КА, специфичные для организации-разработчика. Данные причины вызвали необходимость создания нового предметно-ориентированного языка управления.

На основании анализа достоинств и недостатков существующих средств управления КА были сформулированы требования, предъявляемые к перспективным языковым и инструментальным средствам управления КА, подразумевающие использование предметно-ориентированного языка управления в качестве средства автоматизации выдачи команд на КА из ЦУП.

2 ОРГАНИЗАЦИЯ ПРОЦЕССА УПРАВЛЕНИЯ

Для обеспечения автоматизации процесса управления КА предлагается использовать предметно-ориентированный язык, который должен оперировать набором понятий предметной области, с учетом взаимоотношений между ними. Для определения такого набора был рассмотрен процесс управления КА, и составлена его модель, позволившая в дальнейшем определить требования к разрабатываемому языку управления.

2.1 Основные задачи управления

Управление КА [1, 3] – комплекс мер организационного и технического характера, направленных на выполнение запланированной программы полета на всех этапах жизненного цикла КА:

- выведение КА на целевую орбиту;
- начальные операции по приведению подсистем КА в рабочее состояние;
- проведение проверок, летных или приемосдаточных испытаний;
- штатная эксплуатация на орбите согласно целевому назначению;
- проведение заключительных операций с КА.

При управлении и контроле КА решаются следующие задачи:

- управление системами, целевой и служебной аппаратурой в оперативном режиме или автономно, в соответствии с программой работ, закладываемой на КА с НКУ;
- передача отчетной и телеметрической информации на НКУ для оперативного контроля;
- контроль и автономное управление бортовых подсистем и КА в целом;
- коррекция программы работы КА, алгоритмов автономного управления и контроля бортовых подсистем и КА в целом;
- организация вызова НКУ.

Структура процесса управления полетом одинакова для отечественных и зарубежных КА различного назначения, и включает в себя следующие мероприятия:

- планирование проведения работ по управлению КА;
- навигационные измерения и баллистические расчеты;
- контроль параметров состояния КА;
- выдача на КА управляющих воздействий (УВ);
- реагирование на обнаруженные нештатные ситуации.

2.2 Автоматизированная система управления

Автоматизированная система управления космическими аппаратами (рисунок 2.1) представляет собой совокупность технических и программных средств наземного комплекса управления (НКУ) и бортового комплекса управления (БКУ), предназначенную для управления КА на всех этапах жизненного цикла [45, 46].

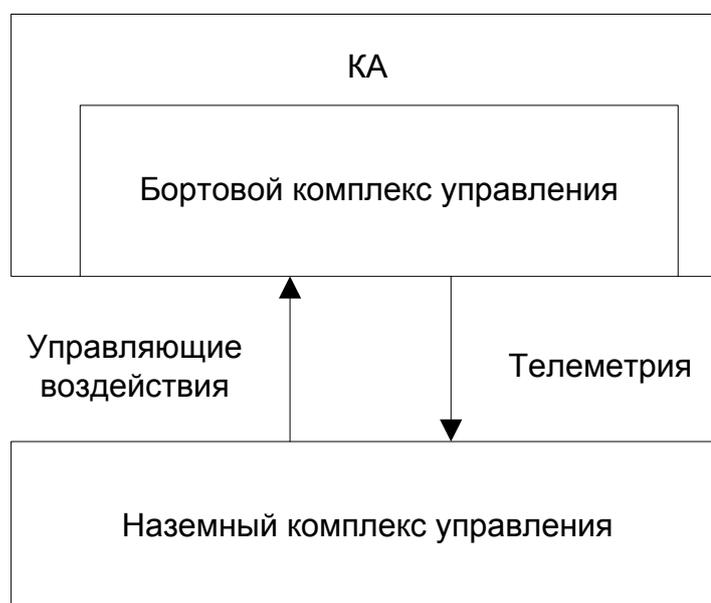


Рисунок 2.1 – Структура АСУ КА

БКУ предназначен для выполнения следующих задач:

- обеспечение приведения элементов конструкции и бортовых систем КА в рабочее состояние;
- управление работой бортовой аппаратуры КА в автономном и оперативном режимах;
- контроль и диагностика технического состояния бортовых систем;
- организация работы бортового программного обеспечения;
- приём от НКУ управляющих воздействий (команд управления);
- передача в НКУ полученной с приборов и подсистем КА телеметрической информации.

Наземный комплекс управления предназначен для непрерывного выполнения задач управления, контроля и поддержания заданных технических и баллистических характеристик КА и орбитальной группировки в целом.

2.3 Режимы управления

Существуют следующие режимы управления КА:

- автономный (задачи управления решаются внутренним контуром управления с использованием средств программно-аппаратных средств БКУ);
- оперативный (задачи управления и контроля подсистемами КА решаются внешним контуром управления с привлечением средств НКУ).

При проведении сеанса управления допускается совмещение оперативного и автономного режимов управления.

2.3.1 Автономный режим управления

В случае использования автономного режима управления решение задач управления обеспечивается бортовыми аппаратно-программными средствами с использованием внутреннего контура управления при непрерывном сбросе на Землю информации в виде совокупности ТМ-параметров. Автономный режим

используется на участке выведения КА, после отделения от носителя, а также в течение срока эксплуатации КА для решения следующих задач:

- выдача команд и сигналов в бортовые системы с привязкой ко времени, в соответствии с программой, заложенной с Земли;
- выдача команд и сигналов в бортовые подсистемы для перевода КА в режимы обеспечения живучести;
- контроль и диагностика технического состояния КА, парирование выявленных неисправностей, включая переход на резервные комплекты БА с формированием вызова НКУ по заранее разработанным алгоритмам;
- формирование диагностической и отчетной информации для НКУ.

2.3.2 Оперативный режим управления

В оперативном (транзитном) режиме управляющие воздействия выдаются на КА из центра управления полетами (ЦУП) с использованием наземного измерительного пункта (НИП) в реальном масштабе времени (рисунок 2.2).

ТМ-информация, снимаемая с борта КА, передается в ЦУП через станцию постоянного приема телеметрической информации, совмещенную с задействованной командно-измерительной станцией (КИС) в реальном масштабе времени.

Контроль исполнения управляющих воздействий в бортовой аппаратуре (БА) КА осуществляется оператором ЦУП.

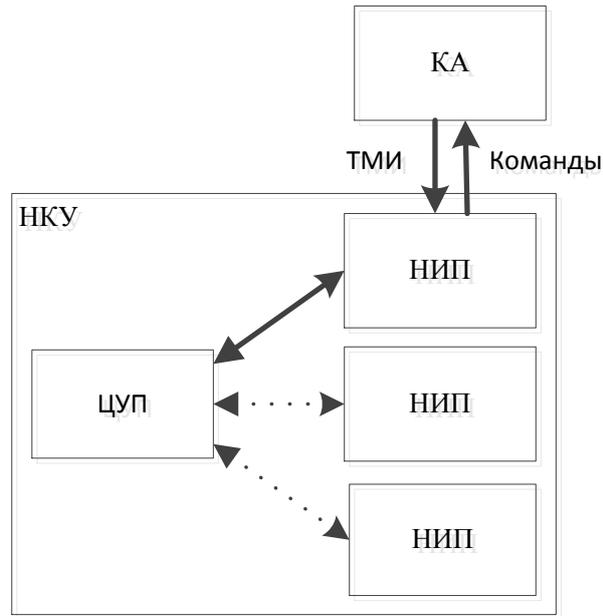


Рисунок 2.2 – Схема оперативного режима управления КА

2.4 Состав наземного комплекса управления

В состав НКУ (рисунок 2.3) входят следующие элементы [3, 46]:

1. ЦУП, предназначенный для обеспечения контроля и управления КА, начиная с момента отделения от средств выведения на начальном этапе полета, на этапах установки в заданную орбитальную позицию, летных испытаний КА и ввода его в штатную эксплуатацию, штатной эксплуатации, перевода КА в другую орбитальную позицию и эксплуатации КА в этой орбитальной позиции, вывода с орбиты.

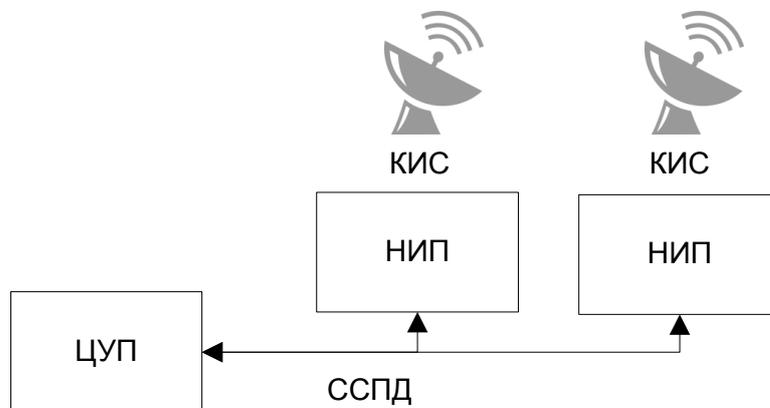


Рисунок 2.3 – Состав НКУ

2. Несколько НИП, разнесенных территориально для максимального охвата орбиты КА зонами радиовидимости. В состав НИП входят средства организации радиолиний управления и контроля КА (КИС), которые обеспечивают вхождение в связь с КА, передачу УВ, прием ТМИ, измерение дальности до КА и его углового положения, криптозащиту и имитозащиту радиолинии.

3. Система связи и передачи данных.

2.5 Технологический цикл управления

Управление КА реализуется отработкой технологического цикла управления. Технологический цикл управления включает в себя последовательное проведение следующих операций:

- планирование работы с КА и задействование наземных средств управления;
- формирование программы управления для запланированного сеанса;
- проведение сеанса управления в запланированное время;
- контроль и оценка технического состояния КА;
- оценка результатов выполнения программы.

2.6 Командный метод управления космическими аппаратами

В ходе проведения запланированных сеансов управления проводится анализ состояния КА на основе непрерывно поступающей телеметрии, контроль работоспособности бортовых и наземных средств управления, выработка решения о выдаче команд управления и их выдача в реальном масштабе времени.

В число задач управления КА, выполняемых с помощью командного метода, входят следующие:

- обеспечение движения КА по требуемой орбите;
- поддержание работоспособности систем КА в целом;
- обеспечение выполнения целевой задачи;

– обнаружение и парирование аномальных ситуаций, препятствующих решению перечисленных задач, ликвидация их последствий.

Командный метод управления КА применяется на следующих этапах жизненного цикла КА (рисунок 2.4): подготовка к вводу в эксплуатацию, штатная эксплуатация, заключительные операции.

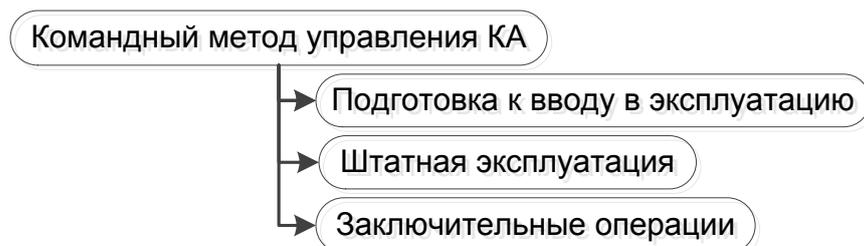


Рисунок 2.4 – Этапы жизненного цикла КА

2.7 Этапы эксплуатации космических аппаратов

2.7.1 Подготовка к вводу в эксплуатацию

Этап подготовки к вводу КА в эксплуатацию отсчитывается с момента контакта подъема КА (рисунок 2.5). На протяжении всего участка выведения, до момента контакта отделения, должен производиться контроль технического состояния КА. Затем выполняются стадии режима начальной подготовки и ориентации. Далее должны быть проверены и приведены в рабочее состояние бортовые системы КА, после чего КА должен быть переведен на рабочую орбиту.

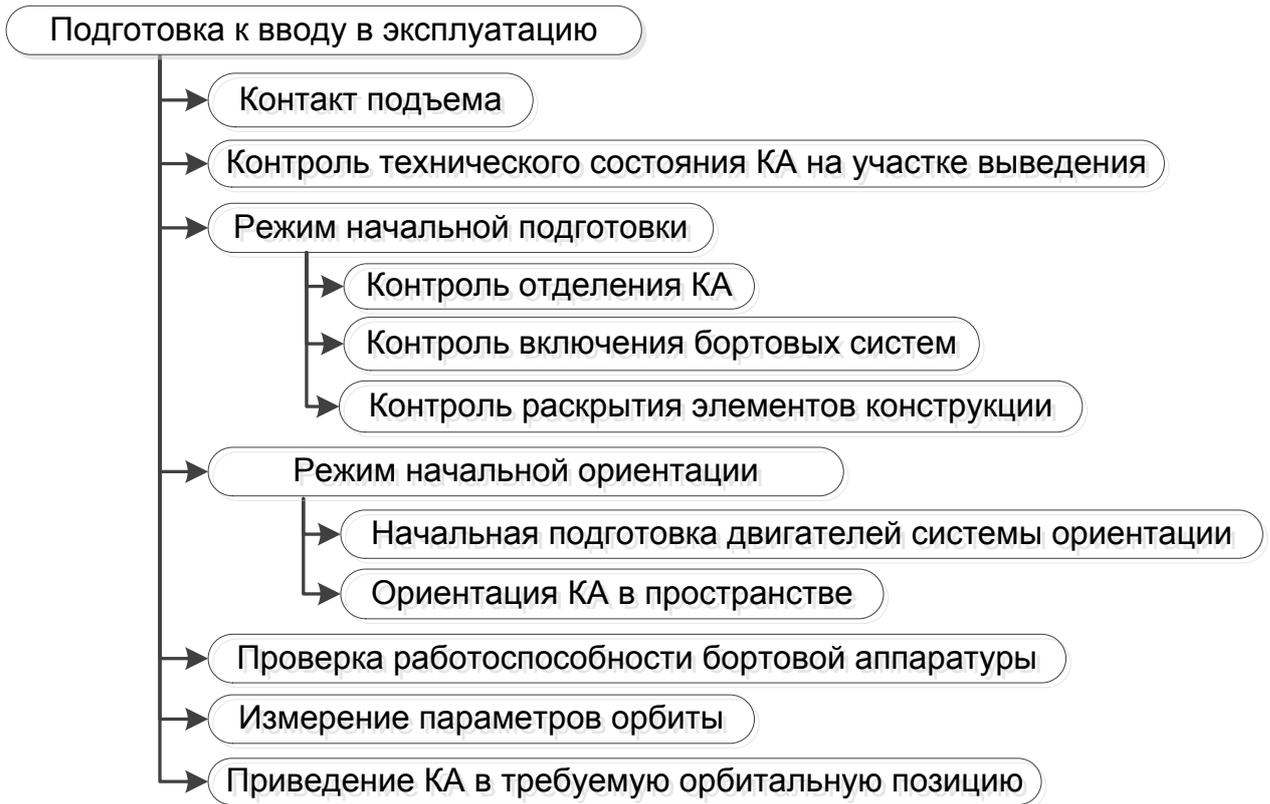


Рисунок 2.5 – Этап подготовки к вводу КА в эксплуатацию

Режим начальной подготовки КА заключается в следующем:

- контроль появления сигнала бортового передатчика КИС и ТМИ до момента контроля отделения;
- контроль отделения аппарата от средства выведения;
- контроль включения бортовых систем по сигналу КО от средств выведения (по данным ТМИ);
- контроль раскрытия элементов конструкции (панелей солнечных батарей, бортовых антенн, защитных экранов, астродатчиков и пр.). Алгоритм раскрытия элементов конструкции реализован в блоке управления БКУ и представляет собой строго определенную последовательность выдачи управляющих сигналов на пиропатроны расчехловки элементов конструкции, реализующие раскрытие пакета солнечных батарей и антенных систем.

Режим начальной ориентации КА заключается в следующем:

- начальная подготовка двигателей системы ориентации;

– ориентация КА в пространстве, осуществляемая следующей последовательностью операций: успокоение и стабилизация аппарата в пространстве в произвольном положении (режим успокоения), ориентация солнечной батареи на Солнце (режим начальной ориентации на Солнце), разворот аппарата до попадания Земли в поле зрения датчика ориентации на Землю (режим начальной ориентации на Землю), режим точной стабилизации по трем осям;

– проверка работоспособности БА, режимов и алгоритмов управления;

– измерение параметров орбиты. Данная операция выполняется для оценки параметров орбиты аппарата, полученных после его выведения. Измерения выполняются после достижения режима начальной ориентации на Солнце (РНОС). Полученные результаты используются для нужд бортового навигационного программного обеспечения, программного обеспечения системы ориентации и для расчета стратегии приведения аппарата в заданную область требуемой орбитальной позиции. Измерения параметров орбиты проводятся дальномерно-угломерной системой, входящей в состав наземного комплекса управления.

Операция приведения КА в требуемую орбитальную позицию начинается с момента определения параметров орбиты выведения и готовности двигательной установки коррекции. После получения окончательных параметров орбиты выведения баллистический центр рассчитывает стратегию выполнения коррекции орбиты, целью которой является не только приведение аппарата в заданную орбитальную позицию, но и исправление ошибок выведения, например ошибок по эксцентриситету или наклонению.

2.7.2 Штатная эксплуатация

В процессе штатной эксплуатации КА по целевому назначению выполняются следующие основные задачи (рисунок 2.6).

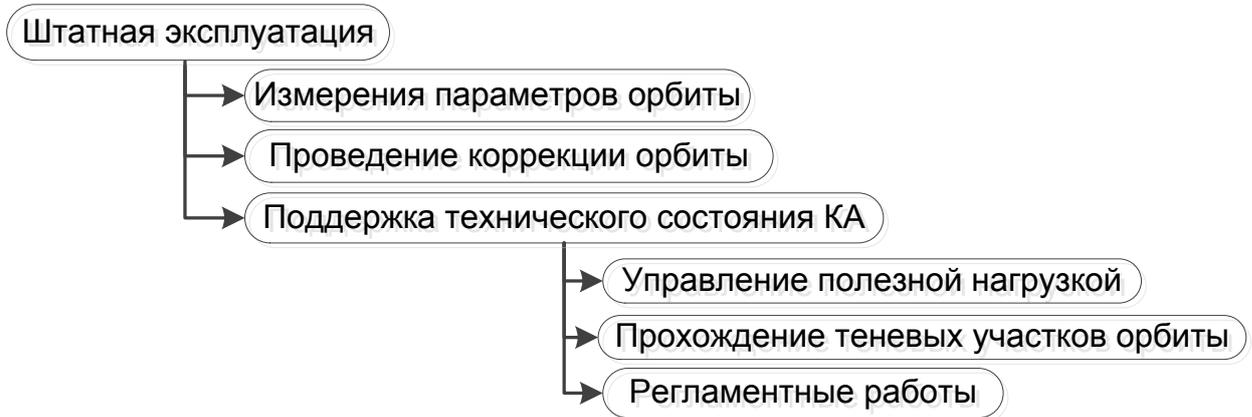


Рисунок 2.6 – Процесс штатной эксплуатации КА

1. Измерения текущих параметров орбиты. Для обеспечения штатной эксплуатации КА на орбите периодически должна выполняться процедура измерения текущих навигационных параметров.

2. Плановые включения двигателей системы коррекции с целью удержания аппарата в заданной области орбитальной позиции по долготе и наклонению (для геостационарных КА). Работы по проведению коррекции орбиты включают в себя следующие составляющие:

а) автономный режим коррекции удержания:

- определение параметров орбиты;
- расчет плана коррекции;
- проведение коррекции удержания.

б) полуавтономный режим коррекции по плану, сформированному в НКУ:

- закладка плана коррекций;
- закладка исходных данных для проведения коррекции;
- проведение коррекции.

3. Поддержка технического состояния КА.

3.1 Управление полезной нагрузкой, в частности, обеспечение требуемых рабочих режимов и распределение ресурсов полезной нагрузки, выполняемых по заявкам и указаниям потребителей. Работы по управлению полезной нагрузкой сильно различаются от одного КА к другому в зависимости от конструкции и назначения КА.

3.2 Прохождение теневых участков орбиты. По результатам анализа энергопотребления с учетом фактической длительности теневых участков определяется стратегия прохождения данных теневых участков с возможностью перехода на режим обеспечения максимальной степени заряженности аккумуляторной батареи и возможными ограничениями на работу полезной нагрузки. При разработке стратегии прохождения теней от Луны обязательно учитывается степень (фаза) затенения. Работы по обеспечению прохождения теневых участков орбиты включают в себя следующие составляющие:

- включение программы РТУЗ;
- отключение программы РТУЗ;
- прохождение теневого участка (ТУ) от Луны, сдвоенных ТУ, ТУ от Земли с НКУ.

3.3 Регламентные работы по подсистемам. Регламентные работы, проводимые в ходе штатной эксплуатации КА, включают в себя периодическое выполнение следующих задач (рисунок 2.7): регламентные работы системы электропитания; регламентные работы системы терморегулирования; регламентные работы бортовой аппаратуры телесигнализации; регламентные работы системы ориентации и стабилизации.

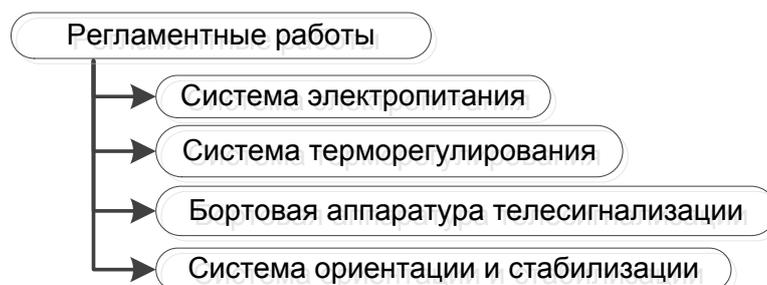


Рисунок 2.7 – Регламентные работы, проводимые в ходе штатной эксплуатации КА

3.3.1 Регламентные работы системы электропитания. Обеспечение штатного функционирования системы электропитания (СЭП), в частности, следующие работы:

– за 10-15 суток до начала сезона теней проводится подготовка СЭП к прохождению теневых участков; за 3 суток до завершения сезона ТУЗ начинается подготовка СЭП к работе на солнечных участках орбиты;

– контроль технического состояния солнечной батареи. Данная операция проводится для принятия решений по ограничениям функционирования СЭП при прохождении ТУ от Земли или Луны, а также для определения степени деградации тока солнечной батареи;

– контроль периодичности включения зарядного устройства аккумуляторной батареи. Проводится в периоды максимальной и минимальной освещенности солнечной батареи с целью определения ее влияния на периодичность цикла заряда аккумуляторной батареи. Целью данной работы является оценка степени саморазряда аккумуляторной батареи и работы зарядно-разрядного устройства блока стабилизации напряжения системы электропитания;

– профилактический разряд аккумуляторной батареи 1 (АБ1), приведение АБ1 в режим обеспечения максимальной степени заряженности;

– профилактический разряд аккумуляторной батареи 2 (АБ2), приведение АБ2 в режим обеспечения максимальной степени заряженности.

3.3.2 Регламентные работы системы терморегулирования. Регламентные работы системы терморегулирования состоят в регулярных проверках резервного гидроэлектронасосного агрегата.

3.3.3 Регламентные работы бортовой аппаратуры телесигнализации. При штатной эксплуатации КА проверки работоспособности аппаратуры телесигнализации проводятся не реже одного раза в год и включают в себя принудительное включение резервных комплектов аппаратуры.

3.3.4 Регламентные работы системы ориентации и стабилизации. Включают в себя выполнение следующих работ:

- проверка резервных комплектов аппаратуры системы ориентации и стабилизации (СОС);
- проверка режима аппаратной солнечной ориентации (РАСО).

2.7.3 Заключительные операции

На этапе заключительных операций выполняются следующие мероприятия: отключение полезной нагрузки; уточнение параметров орбиты; маневр увода КА со штатной орбиты (на орбиту захоронения); измерение параметров орбиты КА после маневра увода; отключение бортовых систем КА; передача КА службе контроля космического пространства. Работы, проводимы на этапе заключительных операций, представлены на рисунке 2.8.

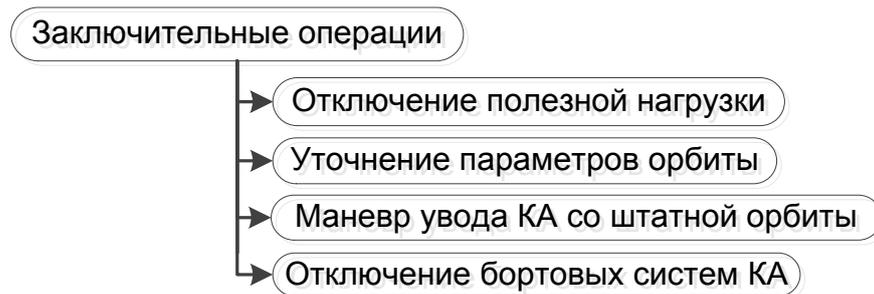


Рисунок 2.8 – Этап заключительных операций

Структура процесса управления КА, применимая для КА различного класса и назначения, представлена на рисунке 2.9.



Рисунок 2.9 – Структура процесса управления КА

2.8 Задачи управления на этапе штатной эксплуатации

В состав минимального набора задач, выполняемых в процессе управления КА, входит перечень работ, представленный далее. Указанный перечень дополняется и изменяется в зависимости от структуры и назначения каждого отдельного изделия.

Измерение текущих навигационных параметров (ИТНП) – определение текущих параметров орбиты КА.

Закладка плана коррекций. Коррекция орбиты КА проводится с целью удержания КА относительно штатной точки стояния по широте и долготе с требуемой точностью. Расчет параметров коррекции удержания (время включения, длительность и номер двигателя коррекции) проводится по результатам ИТНП. Параметры коррекции закладываются на борт в виде плана коррекции. Времена включения и длительность работы двигателей коррекции рассчитываются, исходя из необходимой величины изменения параметров орбиты с учетом ограничений, накладываемых системой электропитания.

Проведение коррекции. Выполнение данного сценария обеспечивает выполнение коррекции орбиты КА в соответствии с ранее заложенным планом.

Включение программы РТУЗ. Выполнение данного сценария обеспечивает выдачу серии команд на подготовку подсистем спутника к работе на теневой орбите и последующим контролем прохождения ТУ.

Отключение программы РТУЗ. После окончания сезона ТУ от Земли производится выключение программы РТУЗ.

Прохождение ТУ от Луны, сдвоенных ТУ, ТУ Земли с НКУ. Выполнение данного сценария обеспечивает выдачу серии команд на подготовку подсистем КА к работе на ТУ от Луны, сдвоенных ТУ, ТУЗ с контролем начала ТУ. Под сдвоенным теневым участком понимается один интервал времени, на котором наступают тени от Земли и тени от Луны, если они совмещаются или интервал между ними менее 30 минут.

Профилактический разряд аккумуляторной батареи (АБ) 1. Выполнение данного сценария обеспечивает выполнение профилактического разряда АБ1, приведение АБ1 в режим обеспечения максимальной степени заряженности. Программа проводится не раньше, чем за 15 суток и не позже, чем за 10 суток до начала ТУ.

Профилактический разряд АБ2. Выполнение данного сценария обеспечивает выполнение профилактического разряда АБ2, приведение АБ2 в режим обеспечения максимальной степени заряженности. Программа проводится не раньше, чем за 15 суток, и не позже чем за 10 суток до начала ТУ.

Проверка резервного гидроэлектронасосного агрегата. В процессе штатной эксплуатации один раз в два года проводится проверка электронасосных агрегатов выдачей соответствующих команд управления.

Переключение резервных комплектов бортовой аппаратуры телесигнализации (БАТС). При штатной эксплуатации КА переход с комплекта на комплект БАТС осуществляется один раз в 6 месяцев.

Проверка резервных комплектов аппаратуры системы ориентации и стабилизации. Регламентные работы по СОС заключаются во включении резервного комплекта оборудования СОС на 24 часа 1 раз в год.

Проверка режима аппаратной солнечной ориентации. Выполнение данного сценария обеспечивает проверку функционирования первого и второго комплектов датчиков наличия Солнца.

2.9 Сценарии управления

Указанные выше стандартные задачи предметной области по управлению КА должны описываться в виде сценария управления.

Сценарий управления представляет собой логически связанную последовательность команд управления, выдаваемых с Земли на бортовые системы КА для реализации единичной задачи управления. Выдача каждой следующей команды зависит от некоторого набора условий, например:

- параметров текущего сеанса, например, наличия или отсутствия дополнительного шифрования (открытый или закрытый режим) выдачи команд;
- текущего значения телеметрических параметров, характеризующего состояние бортовых систем;
- состояния предыдущей команды (один из нескольких вариантов: дошла до программного обеспечения системы автоматизированного обмена, куда выданные команды попадают перед тем, как быть отправленными на наземную станцию, выдана наземной станцией, выполнена на КА);
- значения ТМ-параметров, сохраненных оператором на какой-либо момент времени (необходимо, например, для восстановления состояния приборов полезной нагрузки до определенной конфигурации).

Темп выдачи команд, если это не оговорено специально, определяется темпом получения с НИП подтверждения, свидетельствующего об исполнении команды на борту КА.

Если состояние какого-либо признака не допускает выдачи очередной команды, алгоритм выполнения операции имеет ответвление, обеспечивающее либо продолжение операции по другой ветви алгоритма, либо ее прерывание, регламентированное нужным образом.

Типовые сценарии управления имеют стандартный характер, т. е. при каждом повторении должны выполняться по одним и тем же шаблонам и методикам.

Приблизительное количество сценариев управления для КА различных космических систем, а также суммарное количество строк сценариев управления КА представлено в таблице 2.1. В ходе эксплуатации количество сценариев может изменяться (могут создаваться новые сценарии, редактироваться и удаляться уже существующие).

Таблица 2.1 – Количество и размерность сценариев управления для различных КА разработки АО «ИСС»

КА	Количество сценариев управления	Суммарное количество строк в сценариях управления
Экспресс-АМ	1357	56267
Экспресс-АТ	359	12424
Ямал 300	434	14931
Ямал 401	646	45456
Луч-5А	256	11591

2.10 Библиотека сценариев управления

На основе анализа предметной области (задач по управлению КА) был предложен набор наиболее востребованных сценариев по управлению КА. На основании данного набора разработана библиотека сценариев по управлению КА, фрагмент которой представлен в таблице 2.2. Библиотека может использоваться при создании исходных данных управления для вновь создаваемых КА, с учетом расширения за счет специализированных сценариев, отражающих специфику процесса управления разрабатываемого КА.

Таблица 2.2 – Фрагмент библиотеки сценариев управления КА

Задача		Название	Назначение
Измерение параметров орбиты		ИТНП	Измерение текущих навигационных параметров
Проведение коррекции орбиты		ЗКПИ_ПКУ	Закладка плана коррекций (автономный режим работы)
		A_КОР	Проведение коррекции в автономном режиме включения двигателей коррекции
		ЗКПИ_ИД_К	Закладка ИД для проведения коррекции в полуавтономном режиме работы
		ПА_КОР	Проведение коррекции в полуавтономном режиме включения двигателей коррекции
		З_БИ	Используется для закладки (обновления) с НКУ баллистической информации
		ОБНОВ_НУ	Используется для закладки (обновления) с НКУ орбиты КА
		ВОЗМ_УСК	Используется для обновления значений возмущающих ускорений от двигателей коррекции после проведения уточнения (калибровки) двигателей коррекции
Прохождение теневых участков		ВКЛ_РТУЗ	Включение программы РТУЗ
		ОТКЛ_РТУЗ	Отключение программы РТУЗ
		ТУЛ	Прохождение ТУ от Луны, сдвоенных ТУ, ТУЗ с НКУ
Регламентные работы	Регламентные работы системы электропитания	ПРФ_АБ1	Профилактический разряд АБ1, приведение АБ1 в режим обеспечения максимальной степени заряженности
		ПРФ_АБ2	Профилактический разряд АБ2, приведение АБ2 в режим обеспечения максимальной степени заряженности

Продолжение таблицы 2.2

Регламентные работы	Регламентные работы системы терморегулирования	ПРОВ_ГН_ЭНА	Проверка резервного гидроэлектронасосного агрегата
	Регламентные работы бортовой аппаратуры телесигнализации	РЕГ_БАТС	Переключение комплектов БАТС
	Регламентные работы системы ориентации и стабилизации	ПРОВ_РЕЗ_КОМПЛ	Проверка резервных комплектов аппаратуры СОС
		ПРОВ_РАСО	Проверка РАСО
		НАР_СОС	Наработка приборов СОС

2.11 Модель предметной области

Для рассматриваемой предметной области значимым является следующий набор понятий:

– «Сценарий управления» – последовательность команд, обеспечивающая решение отдельных задач управления.

– «Декодер» – оборудование, предназначенное для кодирования и передачи команд по радиоканалу на борт КА. У декодера есть имя, которое может принимать значение «А» или «В».

– «Команда» – команды различных форматов и назначения, для выдачи на КА. Каждая команда характеризуется типом команды и номером команды. Типы команд: РК (команды управления для исполнения аппаратурой КА), ПК (программные команды для исполнения бортовым программным обеспечением), и некоторые другие типы команд, отличающихся для различных систем. Номер команды – целое число. Назначение диапазона номеров команд определяется при разработке КА.

– «Сообщение» – текстовое сообщение оператору, выдаваемое в ходе выполнения сценария управления, различной степени важности, например, «Пользовательское», «Диагностическое», «Критическое», и другие. Важность

сообщения указывается с помощью цвета, которым оно будет подсвечено при выводе на экран.

– «Пауза» – временная задержка продолжения выполнения процедуры управления.

– «Проверка» – сравнение параметров, указанных в тексте сценария, с их реальными значениями. В качестве сравниваемых параметров могут использоваться следующие параметры: «ТМ-параметр», «Сохраненный ТМ-параметр», «Состояние команды», «Параметр сеанса», «Номер КА».

– «ТМ-параметр» – поименованный элемент данных, отображающий некоторую характеристику функционирования КА в реальном времени (например, температуры, или состояния отдельно взятого прибора: включен или выключен).

– «Сохраненный ТМ-параметр» – значение ТМ-параметра, принудительно сохраненное на определенный момент времени. В данный «архив ТМ-параметров» сохраняется полный набор ТМ-параметров, характеризующий состояние КА на определенный момент времени.

– «Параметр сеанса» – набор характеристик текущего сеанса управления (например, необходимость дополнительного специализированного шифрования выдаваемых команд управления). Набор параметров сеанса известен заранее, до начала сеанса управления, и неизменен в его процессе.

– «Состояние команды» – состояние предыдущей выданной на КА команды. В случае нештатного выполнения команды управления процесс управления должен быть приостановлен до выяснения и устранения причин нештатной ситуации. Данный параметр может принимать несколько значений: «ДоСАО» – команда дошла до программного обеспечения сетевого обмена, «ДоЗС» – команда дошла до земной станции выдачи команд, «Исп» – команда успешно выполнена бортовыми системами КА.

– «Номер КА» – номер КА, управление которым производится с помощью данного сценария управления. Применяется для создания универсальных

сценариев управления КА различных группировок, содержащих несколько КА с различными номерами.

Модель предметной области, разработанная с учетом основных понятий, значимых для автоматизируемой предметной области, определяющая функциональный состав языковых средств поддержки командного метода управления КА, представлена на рисунке 2.10.



Рисунок 2.10 – Модель предметной области

Предложенная модель предметной области позволила в дальнейшем выделить основные операторы языка управления, обеспечивающие поддержку процесса управления КА, что обеспечивает предметную ориентированность разрабатываемых языковых средств.

Выводы по главе 2

Проведен анализ командного метода управления КА, выделена его структура.

На основании анализа процесса управления КА был выделен ряд основных сценариев управления, которые целесообразно включить в библиотеку сценариев. Указанная библиотека может использоваться при создании наборов исходных данных управления для вновь создаваемых КА, с учетом расширения за счет

специализированных сценариев, отражающих специфику процесса управления разрабатываемого КА.

На основе проведенного анализа предложена модель предметной области, определяющая основные понятия, значимые для автоматизируемой предметной области. Выделенный набор понятий позволил в дальнейшем выделить основные операторы языка управления, обеспечивающие поддержку процесса управления КА, что обеспечивает предметную ориентированность разрабатываемых языковых средств.

3 ПРЕДМЕТНО-ОРИЕНТИРОВАННЫЙ ЯЗЫК УПРАВЛЕНИЯ КОСМИЧЕСКИМИ АППАРАТАМИ

3.1 Требования к языку управления

Для обеспечения выполнения целевой задачи (автоматизации процесса управления КА) разрабатываемый язык должен соответствовать модели предметной области, и при этом включать в себя управляющие конструкции, определяющие последовательность выполнения отдельных предметно-ориентированных операторов языка.

1. Требования, определяемые моделью предметной области:

1.1 Язык должен оперировать набором сущностей, достаточных для написания сценариев по управлению КА, в следующем составе: «Команда», «Сообщение», «Пауза», «Проверка», «Декодер» и их параметрами.

1.2 Язык должен поддерживать остановку выполнения сценария управления, для прерывания стандартного хода выполнения сценария управления, в случае, когда необходимо вмешательство оператора для анализа сложившейся ситуации.

2. В языке должны присутствовать управляющие конструкции, определяющие последовательность выполнения отдельных предметно-ориентированных операторов языка.

2.1 Язык должен обеспечивать проверку многовариантного ветвления, включая случай, когда ни один из предложенных вариантов не подошел.

2.2 Язык должен поддерживать проверку выполнения нескольких заданных логических условий «И», «ИЛИ», «НЕ».

2.3 Язык должен обеспечивать возможность включения в текст сценария управления комментариев, содержащих пояснения к тексту программы.

2.4 Язык должен поддерживать вызов других сценариев управления из данного с возможностью передачи некоторых параметров, что обеспечивает

повторное использование уже созданных и отлаженных сценариев, например, путем вызова на выполнение одного сценария из другого.

2.5 Язык должен поддерживать циклы, необходимые для выполнения повторяющихся действий.

3.2 Язык управления космическими аппаратами «Дельта»

3.2.1 Общие характеристики языка управления

Рассмотренные выше требования были реализованы в предметно-ориентированном языке управления КА «Дельта» [47]. Основные особенности языков такого класса представлены в [48 – 56]. Многие ключевые слова, используемые в языке, совпадают с терминами предметной области. Некоторые ключевые слова совпадают с ключевыми словами языка ЯОТР для обеспечения преемственности.

Сценарий управления является процедурой, допускающей использование формальных параметров, хранящейся в базе данных в текстовом виде, и обладающей уникальным именем. Сценарий можно вызвать на исполнение в реальном времени в ходе сеанса управления. Также его можно вызывать и из других сценариев, для чего в состав сценария могут входить формальные параметры, которые при вызове должны быть заменены на фактические. В случае отсутствия такой замены выдается сообщение об ошибке.

Разработанный язык управления КА «Дельта» поддерживает следующие функции:

- выдачу команд управления КА (оператор «ВЫДАТЬ»);
- проверку выполнения заданного условия, например, попадания текущего значения указанного телеметрического параметра в допустимый интервал (оператор «ЕСЛИ»);
- проверку многовариантного ветвления, включая случай, когда ни один вариант из предложенных не подошел (оператор «ВЫБОР»);

- проверку выполнения нескольких заданных логических условий (операторы «И», «ИЛИ», «НЕ»);
- задание паузы на заранее заданный промежуток времени (оператор «ПАУЗА»);
- печать текстовых сообщений оператору (оператор «ПЕЧАТЬ»);
- комментарии, содержащие пояснения к тексту программы (оператор «К»);
- вызов других сценариев управления из данного с возможностью передачи некоторых параметров, что обеспечивает повторное использование уже созданных и отлаженных сценариев (оператор «ВЫЗВАТЬ»);
- остановку выполнения типовой работы, что применяется для обработки нештатной ситуации в управлении, прерывания стандартного хода выполнения сценария управления, в случае, когда необходимо вмешательство оператора для анализа сложившейся ситуации (оператор «СТОП»);
- циклы, необходимые для выполнения повторяющихся действий (оператор «ЦИКЛ»).

В Приложении А приведен синтаксис языка «Дельта».

3.2.2 Синтаксис

Синтаксис языка – набор правил, определяющих допустимые цепочки входных символов, описывается с помощью Расширенных Бэкуса-Наура Форм (РБНФ).

РБНФ разрешает использовать следующие упрощения:

- нетерминалы записываются как отдельные слова;
- терминалы записываются в кавычках;
- необязательные элементы синтаксической конструкции заключаются в квадратные скобки «[» и «]»;
- элементы синтаксической конструкции, повторяющиеся нуль и более раз, заключаются в фигурные скобки «{» и «}»;

- альтернативные варианты разделяются знаком «|»;
- круглые скобки "(" и ")" используются для ограничения альтернативных конструкций;
- сочетание фигурных скобок и косой черты "{/" и "/}" используется для обозначения повторения один и более раз;
- нетерминальные символы изображаются словами, выражающими их интуитивный смысл и написанными на русском языке.

Нетерминальные символы начинаются с заглавной буквы (например, Оператор). Терминальные символы записываются прописными буквами (например, «ПРОГРАМ»).

Операторы языка должны разделяться символом «перенос строки» – каждый следующий оператор сценария управления должен быть позиционирован в начале следующей строки. Отступы в начале строки для лучшей читаемости кода сценариев допустимы. Один оператор должен располагаться на одной строке сценария управления. Операторы языка могут находиться только в начале строки. Пробелы и концы строк не должны встречаться внутри слов (исключая комментарии). Заглавные и строчные буквы считаются различными. Количество символов табуляции между лексемами (пробелы, табуляции) не регламентировано.

3.2.3 Структура сценария управления

В начале и окончании сценария управления должны быть расположены ключевые слова «ПРОГРАМ», «КПРОГРАМ». Их наличие является обязательным (по ним автоматически определяется язык обрабатываемого сценария управления – русский или английский, и тип синтаксиса – ЯОТР или «Дельта»).

Сценарий управления может содержать набор операторов, описывающих требуемую логику выдачи команд управления КА, в частности управляющие операторы: условные операторы и оператор цикла.

Сценарий_управления = "ПРОГРАМ"

```
{Оператор}
"КПРОГРАМ" .
```

3.2.4 Операторы языка управления

Перечень операторов языка можно разделить на две группы: операторы описания предметной области и операторы управления ходом выполнения сценария.

К операторам описания предметной области относятся действия, выполняемые системой в процессе управления КА: выдача команды, печать сообщения, остановка выполнения сценария, смена декодера, пауза. К операторам управления ходом выполнения сценария относятся проверка условий, выбор альтернативы из различных вариантов, цикл, вызов другого сценария управления из данного.

```
Оператор =
Оператор_выдать          |
Оператор_печать          |
Оператор_пауза           |
Оператор_стоп            |
Оператор_декодер         |
Оператор_если            |
Оператор_выбор           |
Оператор_цикл            |
Оператор_вызвать         |
Оператор_комментарий .
```

3.2.4.1 Оператор «ВЫДАТЬ»

Оператор предназначен для выдачи управляющих воздействий на КА. Применяемое ключевое слово для обозначения оператора – «ВЫДАТЬ». В качестве аргументов оператора должны быть использованы тип команды и номер команды. Допускаются следующие типы команд:

- разовые команды (РК) для исполнения аппаратурой КА;

- программные команды (ПК) для исполнения бортовым программным обеспечением;
- некоторые другие типы команд, зависящие от конкретной реализации космической системы.

Оператор_выдать = "ВЫДАТЬ" Тип_команды Номер_команды.

Тип_команды = "РК" | "ПК".

Номер_команды = Целое.

Числа – целые или вещественные (без знака) константы. Вещественное число должно содержать десятичную точку.

Число = Цифра [{Цифра}] ["." Цифра].

Цифра = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9".

Примеры:

18

2.5

Пример использования:

ВЫДАТЬ РК 40

Выполнение данного оператора языка вызовет выдачу команды типа «РК» номер 40 на наземные радиосредства, и далее на борт КА.

3.2.4.2 Оператор «ПЕЧАТЬ»

Оператор выдачи произвольных сообщений на экран для пояснения производимых в сценарии управления действий по управлению КА. Ключевое слово для обозначения оператора – «ПЕЧАТЬ». Цвет выводимого сообщения можно указать, включив в сообщение наименование цвета в формате «#ЦВЕТ», например, «#КРАСНЫЙ», или «#ЗЕЛЕНый».

Значение по умолчанию (пустое поле) соответствует черному тексту.

По умолчанию все сообщения, выдаваемые из сценария управления, попадают в общее окно лог-файла наряду с другими диагностическими

сообщениями. При необходимости привлечения особого внимания к сообщению есть возможность вызывать отдельное всплывающее окно с текстом сообщения, для чего необходимо указать ключевое слово «#ОКНО» в тексте сообщения. Исполнение сценария продолжится только после закрытия окна всплывающего сообщения.

Оператор_печать = "ПЕЧАТЬ" [Цвет] ["#ОКНО"] Текст .

**Цвет = "#КРАСНЫЙ" |
"#ЗЕЛЕНый" |
"#СИНИЙ" |
"#ОРАНЖЕВый" |
"#РОЗОВый" |
"#ФИОЛЕТОВый" |
"#ЖЕЛТый" .**

Текст = (Буква|Цифра) {Пробел|Буква|Символ|Цифра} .

Пример использования:

ПЕЧАТЬ #КРАСНЫЙ Выполнение процедуры остановлено

3.2.4.3 Оператор «ПАУЗА»

Оператор остановки выполнения сценария управления на указанный интервал времени. Предназначен для обеспечения ожидания завершения процессов на борту КА. Ключевое слово – «ПАУЗА». Операция обеспечивает выдержку нужного интервала времени между предшествующей и последующей операциями.

Оператор_пауза = "ПАУЗА" Время .

Запись времени – запись временных интервалов в общепринятом формате, часы, минуты и секунды задержки указываются через двоеточия «:».

Время = чч:мм:сс .

**чч = "00" | "01" | "02" | "03" | "04" | "05" | "06" ...
"99" .**

**мм = "00" | "01" | "02" | "03" | "04" | "05" | "06" ...
"59" .**

сс = "00" | "01" | "02" | "03" | "04" | "05" | "06" ...
"59".

Пример использования:

ПАУЗА 00:01:00

3.2.4.4 Оператор «СТОП»

Оператор принудительной остановки выполнения сценария управления (например, в случае, когда требуемые ТМ-параметры не приняли необходимое значение). Ключевое слово «СТОП» останавливает автоматическую обработку сценария. После исполнения данного оператора ни одна команда, описанная далее по тексту сценария, не будет выполнена.

При необходимости выполнение сценария управления можно продолжить средствами программного обеспечения управления.

Оператор_стоп = "СТОП".

Пример использования:

СТОП

3.2.4.5 Оператор «ДЕКОДЕР»

Оператор смены декодера (комплекта наземного оборудования, предназначенного для кодирования и передачи команд по радиоканалу на борт КА). Ключевое слово – «ДЕКОДЕР», затем идет номер активируемого декодера, «А» или «В».

Оператор_ДЕКОДЕР = "ДЕКОДЕР" Номер_декодера.

Номер_декодера = "А" | "В".

Пример использования:

ДЕКОДЕР А

Помимо целевых операторов, описывающих действия по управлению КА, в язык введены управляющие конструкции, обеспечивающие возможность нелинейного выполнения сценария управления.

3.2.4.6 Операции «ЕСЛИ», «ИНАЧЕ», «КЕСЛИ»

Данные условные операторы позволяют выполнять определенную последовательность действий в зависимости от выполнения некоторого логического выражения. В случае необходимости можно указать время, в течение которого будет выполняться ожидание выполнения указанного условия – например, предусмотренное на завершение протекания некоторого физического процесса на борту КА, что отразится в поступающей телеметрии. В случае, если условие выполнилось, исполняется указанная далее последовательность операций.

Приоритет (порядок выполнения) логических операций внутри условного выражения следующий:

1. Выполняются отрицания над отдельными переменными («НЕ»);
2. Вычисляются конъюнкции («И»);
3. Вычисляются дизъюнкции («ИЛИ»).

Внутри операции «ЕСЛИ» допускается использовать оператор «ИНАЧЕ», который задает альтернативную последовательность действий на тот случай, когда условие, заданное в дополнительных строках оператора «ЕСЛИ», не выполнено. В любом случае оператор «ЕСЛИ» должен быть закрыт оператором «КЕСЛИ».

Оператор_если =

"ЕСЛИ" Список_условий ["ЗА" Время]

{Оператор}

["ИНАЧЕ"

{Оператор}]

"КЕСЛИ" .

Список_условий = Условие {"И" Условие | "ИЛИ" Условие} .

Условие = ["НЕ"] (Идентиф Оператор_сравнения Число |
Идентиф "В" Интервал) .

Интервал = (" ("|"[") Число ";" Число (")"|"]") .

Оператор_сравнения = "=" |
"<" |
">" |
"<=" |
">=" |
"!=" .

Идентификаторы – буквенно-цифровые обозначения, применяющиеся для обработки значений следующих параметров: ТМ-параметров, ТМ-параметров из сохраненного архива, состояние выданной ранее команды управления, номера космического аппарата, параметров сеанса управления.

Идентиф = **ТМ_параметр**
| **Сохраненный_ТМ_параметр**
| **Состояние**
| **Ном_КА**
| **Параметр_сеанса** .

ТМ-параметр (**ТМ_параметр**) – элемент данных, получаемых по радиоканалу с борта КА (телеметрии) о состоянии бортовой аппаратуры и наиболее важных эксплуатационных параметрах подсистем КА. Имя ТМ-параметра представляет собой буквенно-цифровую последовательность. Первый символ должен быть буквой. Имя ТМ-параметра не должно совпадать с ключевыми словами языка. Значения ТМ-параметров – вещественные числа.

ТМ_параметр = Буква{Буква|Цифра} .
Буква = "А" | "Б" | "В" | ... "Я" | "а" | "б" | "в" | ...
"я" .

ТМ-параметр из сохраненного архива (**Сохраненный_ТМ_параметр**) – один из ТМ-параметров системы, текущее значение которого на определенный момент

времени сохранено в архив. Для использования в тексте сценария значений ТМ-параметров из числа сохраненных необходимо указать их имена в формате «#ТМ_параметр».

Сохраненный_ТМ_параметр = #ТМ_параметр.

Управляющее воздействие (команда) в ходе выдачи с помощью наземных радиосредств и исполнения бортовыми системами КА проходит несколько «контрольных точек», что отображается в созданном языке управления:

- команда выдана из ПО управления КА, и дошла до ПО системы автоматизированного обмена (САО), куда выданные команды попадают перед тем, как быть отправленными на наземную станцию,
- выдана наземной станцией,
- выполнена на КА.

Состояние = "@доСАО" | "@доЗС" | "@ИСП".

Данные переменные принимают значение «1» когда выданная команда прошла соответствующую «контрольную точку», и остаются равными «0», в противоположном случае. В случае сбоя выдачи и исполнения команды управления данные параметры позволяют отследить этап, на котором произошел сбой.

Номер КА (**Ном_КА**) – применяется для систем из нескольких КА, для управления которых применим один и тот же сценарий управления. При запуске сценария на выполнение в ходе сеанса связи с КА будут выполнены только действия, соответствующие указанному номеру КА. Номер КА может принимать значение от 1 до 1000.

Параметры сеанса – характеристики проводимого сеанса управления, например, необходимости специализированного шифрования выдаваемых команд (открытый или закрытый режим управления). Должны указываться в формате «INFx», где x – число от 1 до 15. Каждый параметр содержит в себе целое число в интервале от 0 до 100.

```

Параметр_сеанса = "INF1" | "INF2" | "INF3" | "INF4" |
"INF5" ... "INF15".

```

Пример использования:

```

ЕСЛИ ТМ1 > 8 ЗА 00:01:00
ПЕЧАТЬ ТМ1 > 8
ИНАЧЕ
СТОП
КЕСЛИ

```

Данный пример иллюстрирует ситуацию ожидания в течение одной минуты того, чтобы значение ТМ-параметра ТМ1 стало больше восьми – в этом случае будет напечатано сообщение «ТМ1 > 8», иначе выполнение сценария управления будет остановлено (будет выполнен оператор «СТОП»). Если данный признак примет требуемое значение за меньшее время, чем указано, выполнение сценария управления продолжится.

3.2.4.7 Операторы «ВЫБОР», «ВАРИАНТ», «КВЫБОР»

Образуют конструкцию многовариантного ветвления – аналог оператора «switch» в языках высокого уровня. Конкретная операция «ВАРИАНТ» выбирается для исполнения в том случае, если текущее проверяемое значение переменной оказалось в указанном диапазоне. Внутри операции «ВЫБОР» может содержаться операция «ОСТАЛЬНОЕ», которая будет выполняться, если ни один из предложенных вариантов не исполнился.

Между ключевыми словами «ВЫБОР» и «ВАРИАНТ» не допускается использовать никакие другие операции.

```

Оператор_выбор = "ВЫБОР" Идентиф
{Вариант}
["ОСТАЛЬНОЕ"
{Оператор}]
"КВЫБОР".
Вариант =

```

"ВАРИАНТ" "(" Число ")" | Интервал
 [{Оператор}]

Пример использования:

```

ВЫБОР ТМ1
ВАРИАНТ (0; 8)
ПЕЧАТЬ Параметр ТМ1 < 8
ВАРИАНТ 8
ПЕЧАТЬ Параметр ТМ1 = 8
ВАРИАНТ (8;1000)
ПЕЧАТЬ Параметр ТМ1 > 8
КВЫБОР
  
```

3.2.4.8 Оператор «ЦИКЛ»

Операция обеспечивает повторение выбранных действий по управлению указанное количество раз.

```

Оператор_цикл = "ЦИКЛ" "(" целое ")"
{Оператор}
"КЦИКЛ" .
  
```

Пример использования:

```

ЦИКЛ (8)
ВЫДАТЬ РК 40
ПАУЗА 00:01:00
КЦИКЛ
  
```

3.2.4.9 Оператор «ВЫЗВАТЬ»

Оператор «ВЫЗВАТЬ» инициирует выполнение указанного сценария управления из текущего. Возвращение значений в исходную процедуру не предусмотрено. Вызов сценария может содержать список фактических параметров, которые заменяют соответствующие формальные параметры, использованные в тексте вызываемого сценария. Передаваемый параметр не может содержать символа «,». Чаще всего в качестве параметров передаются номера команд для выдачи, время ожидания исполнения команды, имена ТМ-параметров для проверки. Рекурсивный вызов сценариев управления не поддерживается.

В тексте вызываемого сценария должны быть указаны параметры для замены в формате «&1», «&2», которые будут заменены переданными при вызове значениями. В случае отсутствия замены параметра ситуация считается ошибочной, о чем выдается соответствующее сообщение.

```
Оператор_вызвать = "ВЫЗВАТЬ" Имя_сценария [ "(" "Параметр{" , "
Параметр} " ) " ] .
```

```
Имя_сценария = {Буква|Цифра} .
```

```
Параметр = Идентиф | Число .
```

Примеры использования:

```
ВЫЗВАТЬ _РК(40)
```

```
ВЫЗВАТЬ БРК_ШАГ
```

В первом примере в вызываемый сценарий передается параметр «&1», равный 40. Это значит, что в вызываемом сценарии все параметры «&1» будут заменены на «40».

Во втором примере показан вызов сценария «БРК_ШАГ» без передачи фактических параметров. Случай, когда в вызываемом сценарии были использованы формальные параметры, но они не были заменены на фактические, считается ошибочным.

```
ПЕЧАТЬ Выдача ПК &1
```

ВЫДАТЬ &1

В третьем и четвертом примерах показано использование параметра внутри вызываемого сценария управления.

3.2.4.10 Комментарии

Язык предусматривает возможность включения в текст сценария произвольных текстовых пояснений, игнорируемых транслятором. Для включения комментария должно использоваться ключевое слово «К».

Оператор_комментарий = "К" Текст

Пример использования:

К Дата создания сценария – 2015.04.29

3.2.4.11 Пример сценария управления

Пример сценария управления на разработанном языке управления Дельта с вызовом другого сценария из данного:

```

ПРОГРАМ
ПЕЧАТЬ Выдача ПК &1
ВЫДАТЬ ПК &1
ПЕЧАТЬ Выдача ПК &2
ВЫДАТЬ ПК &2
КПРОГРАМ

```

Пример сценария управления на разработанном языке управления Дельта:

ПРОГРАМ К Пример написания сценария управления

```

ВЫЗВАТЬ РТР_Вкл_1
ВЫЗВАТЬ РТР_АКТ(40, 5)
ПАУЗА 00:00:10

```

```

ЕСЛИ АТМ1 = 1 ЗА 00:00:20
    ПЕЧАТЬ #КРАСНЫЙ Параметр АТМ1 = 1
    СТОП
КЕСЛИ

```

ПАУЗА 00:00:10

ЕСЛИ АТМ1 = 1 **И** АТМ2 В (-5,1) **И** АТМ3 В [-5,1] **ЗА** 00:00:20

```

ПЕЧАТЬ ТМ-параметры в норме
СТОП
КЕСЛИ

ДЕКОДЕР А

ЦИКЛ (8)
  ВЫДАТЬ РК 40
  ПАУЗА 00:01:00
КЦИКЛ

ЕСЛИ #ИСП = 0
  ПЕЧАТЬ #КРАСНЫЙ Команда не прошла!
  СТОП
КЕСЛИ

ВЫБОР АТМ2
ВАРИАНТ (-5;1)
  ПЕЧАТЬ Параметр АТМ2 находится в границах (-5,1)
ВАРИАНТ (1,8)
  ПЕЧАТЬ Параметр АТМ2 находится в границах (1,8)
ОСТАЛЬНОЕ
  ПЕЧАТЬ #КРАСНЫЙ Параметр АТМ2 не принял требуемых
значений
  СТОП
КВЫБОР
КПРОГРАМ

```

3.3 Поддержка англоязычного варианта языка управления

Для соответствия требованиям международных контрактов была реализована поддержка англоязычной версии языка управления КА «Дельта». Синтаксис языка для англоязычной и русскоязычной версий не различается. Соответствие ключевых слов различных вариантов языка представлено в таблице 3.1. Примеры сценариев управления с использованием англоязычной версии представлены на рисунках 3.3 и 3.4.

В рамках одного сценария поддерживается использование только одного языка. Язык вызываемого сценария управления может не совпадать с исходным (например, из русскоязычных сценариев можно вызывать англоязычные).

Таблица 3.1 – Соответствие русскоязычной и англоязычной версий ключевых слов разработанного языка

Ключевое слово		Комментарий
Русскоязычная версия	Англоязычная версия	
ПРОГРАМ	PROGRAM	Операция начала сценария
КПРОГРАМ	EPROGRAM	Операция завершения сценария
ВЫДАТЬ	SEND	Операция выдачи команды управления
ЕСЛИ	IF	Операция начала условия
ИНАЧЕ	ELSE	Операция несовпадения условия
КЕСЛИ	EIF	Операция завершения условия
ВЫБОР	SELECT	Операция выбора из нескольких вариантов
ВАРИАНТ	CASE	Операция выбора указанного варианта
ОСТАЛЬНОЕ	REST	Операция выбора по умолчанию
КВЫБОР	ESELECT	Операция завершения выбора
ПАУЗА	PAUSE	Операция задержки выполнения сценария
ПЕЧАТЬ	PRINT	Операция вывода сообщения на экран
К	C	Комментарий к тексту сценария
ВЫЗВАТЬ	CALL	Операция вызова указанного сценария управления
ДЕКОДЕР	DECODER	Операция смены декодера
СТОП	STOP	Операция остановки выполнения сценария
ЦИКЛ	CYCLE	Операция начала цикла
КЦИКЛ	ECYCLE	Операция завершения цикла
РК	RK	Разовая команда
ПК	PK	Программная команда
И	AND	Операция конъюнкции
ИЛИ	OR	Операция дизъюнкции
НЕ	NOT	Операция отрицания
ЗА	FOR	Указание периода ожидания
@доСАО	@toFES	Состояние выданной команды – дошла до САО
@доЗС	@toTTC	Состояние выданной команды – дошла до ЗС
@ИСП	@DONE	Состояние выданной команды – исполнена на КА
#ОКНО	#WINDOW	Параметр вывода сообщения во всплывающее окно
#КРАСНЫЙ	#RED	Цвет выводимого на экран сообщения – красный
#ЗЕЛЕНЬ	#GREEN	Цвет выводимого на экран сообщения – зеленый
#СИНИЙ	#BLUE	Цвет выводимого на экран сообщения – синий
#ОРАНЖЕВЫЙ	#ORANGE	Цвет выводимого на экран сообщения – оранжевый
#РОЗОВЫЙ	#PINK	Цвет выводимого на экран сообщения – розовый
#ФИОЛЕТОВЫЙ	#PURPLE	Цвет выводимого на экран сообщения – фиолетовый
#ЖЕЛТЫЙ	#YELLOW	Цвет выводимого на экран сообщения – желтый
А	A	Имя декодера – А
В	B	Имя декодера – В
INF1...INF15	INF1...INF15	Параметры сеанса связи

Пример англоязычного сценария управления КА:

```
PROGRAM
PRINT Sending of PK &1
SEND ПК &1
PRINT Sending of PK&2
SEND ПК &2
EPROGRAM
```

Дополнительный пример сценария управления:

```
PROGRAM
  K Control Script example

  CALL PL_ON_1
  CALL PL_ACT(40, F300:1)
  PAUSE 00:00:10

  IF ATM1 = 1 FOR 00:00:20
    PRINT #RED TM-parameter ATM1 = 1
    STOP
  EIF

  PAUSE 00:00:10

  IF ATM1 = 1 AND ATM2 IN (-5,1) AND ATM3 IN [-5,1]
FOR 00:00:20
    PRINT TM-parameters are in limits
    STOP
  EIF

  DECODER A

  CYCLE (8)
    SEND RK 40
    PAUSE 00:01:00
  ECYCLE

  IF #ИСП = 0
    PRINT #RED Command was not executed
    STOP
  EIF

  SELECT ATM2
  CASE (-5,1)
```

```

PRINT Parameter ATM2 are in limits (-5,1)
CASE (1,8)
PRINT Parameter ATM2 are in limits (1,8)
REST
PRINT #RED TM-parameter ATM2 did not take
required values
STOP
EPROGRAM

```

3.4 Оценка соответствия языка «Дельта» предъявляемым требованиям

Оценка степени соответствия разработанного языка управления КА предъявляемым требованиям представлена в таблице 3.2.

Таблица 3.2 – Оценка степени соответствия разработанного языка предъявляемым требованиям

Требование	Реализация в разработанных языковых средствах
Текстовая форма представления, что обеспечит хорошую читаемость даже объемных сценариев управления с нелинейной логикой выполнения, строгость, компактность записи, простоту хранения	Сценарий управления представляется в виде последовательности символов
Перспективный язык управления должен относиться к классу специализированных языков, предназначенных для целей автоматизации процесса управления КА, что обеспечит большую предметно-ориентированность разрабатываемых средств, что позволит упростить процесс обучения конечных пользователей	Язык управления разработан с использованием терминологии предметной области

Продолжение таблицы 3.2

Использование лексических единиц, соответствующих терминологии предметной области, что облегчит процесс создания сценариев управления	Наименования команд «РК», «ПК», операторы «ВЫДАТЬ», «ПАУЗА», «ДЕКОДЕР», «ПЕЧАТЬ», цвета строк печати – «КРАСНЫЙ», «ЖЕЛТЫЙ»
Поддержка как русскоязычного, так и англоязычного синтаксиса для соответствия требованиям международных контрактов	Каждому оператору языка соответствует англоязычный вариант
Язык должен оперировать набором сущностей, достаточных для написания сценариев по управлению КА, в следующем составе: «Команда», «Сообщение», «Пауза», «Проверка», «Декодер» и их параметрами	Отражено в наборе операторов языка: «ВЫДАТЬ», «ПЕЧАТЬ», «ПАУЗА», и др.
Язык должен поддерживать остановку выполнения сценария управления, для прерывания стандартного хода выполнения сценария управления, в случае, когда необходимо вмешательство оператора для анализа сложившейся ситуации	Оператор «СТОП»
Язык должен обеспечивать проверку многовариантного ветвления, включая случай, когда ни один из предложенных вариантов не подошел	Операторы «ВЫБОР», «ВАРИАНТ», «ОСТАЛЬНОЕ», «КВЫБОР»
Язык должен поддерживать проверку выполнения нескольких заданных логических условий «И», «ИЛИ», «НЕ»	Ключевые слова «И», «ИЛИ», «НЕ»

Продолжение таблицы 3.2

Язык должен обеспечивать возможность включения в текст сценария управления комментарием, содержащих пояснения к тексту программы	Оператор «К»
Язык должен поддерживать вызов других сценариев управления из данного с возможностью передачи некоторых параметров, что обеспечивает повторное использование уже созданных и отлаженных сценариев, например, путем вызова на выполнение одного сценария из другого	Оператор «ВЫЗВАТЬ», а также библиотека сценариев
Язык должен поддерживать циклы, необходимые для выполнения повторяющихся действий	Оператор «ЦИКЛ»

На основе данных анализа можно сделать вывод о соответствии разработанных языковых средств выдвинутым ранее требованиям.

3.5 Семантический контроль

Надежный язык программирования позволяет выявлять большинство ошибок во время трансляции программы, а не во время ее выполнения.

Для обеспечения надежности созданного языка, помимо проверки текстов сценария управления на соответствие разработанному синтаксису, были приняты следующие меры:

- проверка отсутствия скрытой рекурсии (в тексте сценария вызывается другой сценарий, который, возможно, через еще несколько вложенных сценариев, вызывает первый);

- проверка номеров команд управления на соответствие допустимому диапазону;

- проверка наличия проверяемых в тексте сценария ТМ-параметров в составе исходных данных на КА в БД;
- проверка наличия текстов вызываемых сценариев в БД;
- проверка совпадения количества и состава параметров в исходном и вызываемом сценариях управления;
- в случае вызова сценария управления с передачей параметров проверка синтаксиса вызываемого сценария после подстановки параметров осуществляется дополнительно;
- перед началом выполнения в ходе сеанса связи сценарий управления принудительно проверяется на наличие ошибок. При их наличии исполнение сценария блокируется.

Помимо этого, надежность созданного языка управления обеспечивается читабельностью создаваемых сценариев управления за счет использования системы понятий предметной области.

Контроль выдачи опасных команд осуществляется средствами программного обеспечения управления: выдача потенциально опасных команд требует двойного-тройного подтверждения, а наиболее опасные команды не входят в состав команд, доступных для выдачи (для их использования необходимо сначала произвести дополнительные действия по вводу команд в состав доступных).

Все сценарии управления отрабатываются на программном имитаторе КА, получающем и обрабатывающем команды управления, и в ответ выдающем поток телеметрии, идентичному потоку реальной ТМИ с КА в соответствии с заложенной логикой, что дополнительно обеспечивает надежность создаваемых сценариев.

Выводы по главе 3

Разработан предметно-ориентированный язык управления КА «Дельта», синтаксис которого описан с помощью РБНФ.

Разработанный язык управления поддерживает следующие функции: выдачу команд управления КА, проверку выполнения заданного условия, паузы, печать текстовых сообщений оператору; комментарии, содержащие пояснения к тексту программы, циклы, смену декодера.

В «Дельте» поддерживаются как русский, так и английский варианты написания операторов языка для использования в отечественных и международных проектах.

Показано соответствие разработанного языка ранее выделенному перечню требований к языкам такого типа.

4 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА СОЗДАНИЯ И ИСПОЛНЕНИЯ СЦЕНАРИЕВ УПРАВЛЕНИЯ КОСМИЧЕСКИМИ АППАРАТАМИ

4.1 Структура инструментальных средств

Для поддержки ранее представленных требований, помимо языка программирования сценариев управления, необходима разработка инструментальных средств, обеспечивающих технологический цикл создания и выполнения сценариев управления КА.

Структура разработанных инструментальных средств (рисунок 4.1):

- «Интерпретатор» отвечает за проверку синтаксиса сценариев управления, а также за их выполнение в реальном времени в ходе сеанса связи с КА;
- «Помощник» по вводу сценариев предназначен для упрощения ввода текстов сценариев управления – осуществляет подстановку введенных пользователем данных в соответствующие конструкции языка и добавление этих конструкций к тексту редактируемого сценария;
- «Текстовый редактор» отвечает за отображение, редактирование и последующее сохранение текстов сценариев;
- «Библиотека процедур управления» содержит стандартные, наиболее применимые процедуры управления, подлежащие редактированию, изменению и дополнению;
- «Конвертер» отвечает за анализ выбранных пользователем документов и формирование текстов сценариев на основе полученных данных.

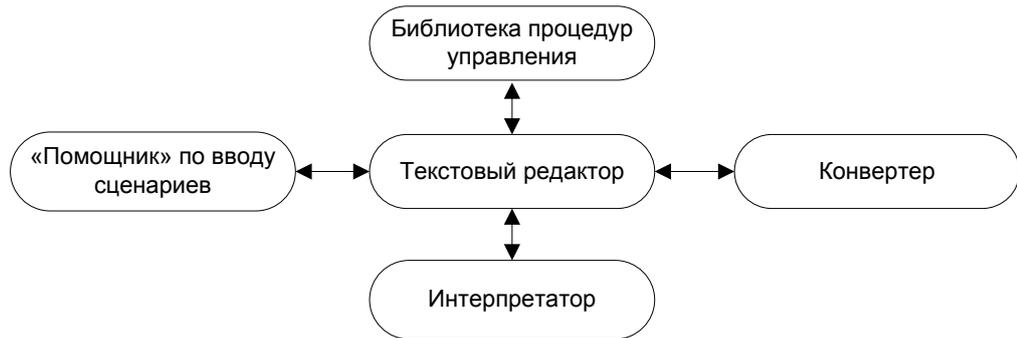


Рисунок 4.1 – Структура инструментальных средств

Разработанные инструментальные средства создания и исполнения сценариев управления КА были интегрированы в состав ПО управления КА разработки АО «ИСС» (рисунок 4.2). Получено свидетельство о регистрации программы для ЭВМ [55].

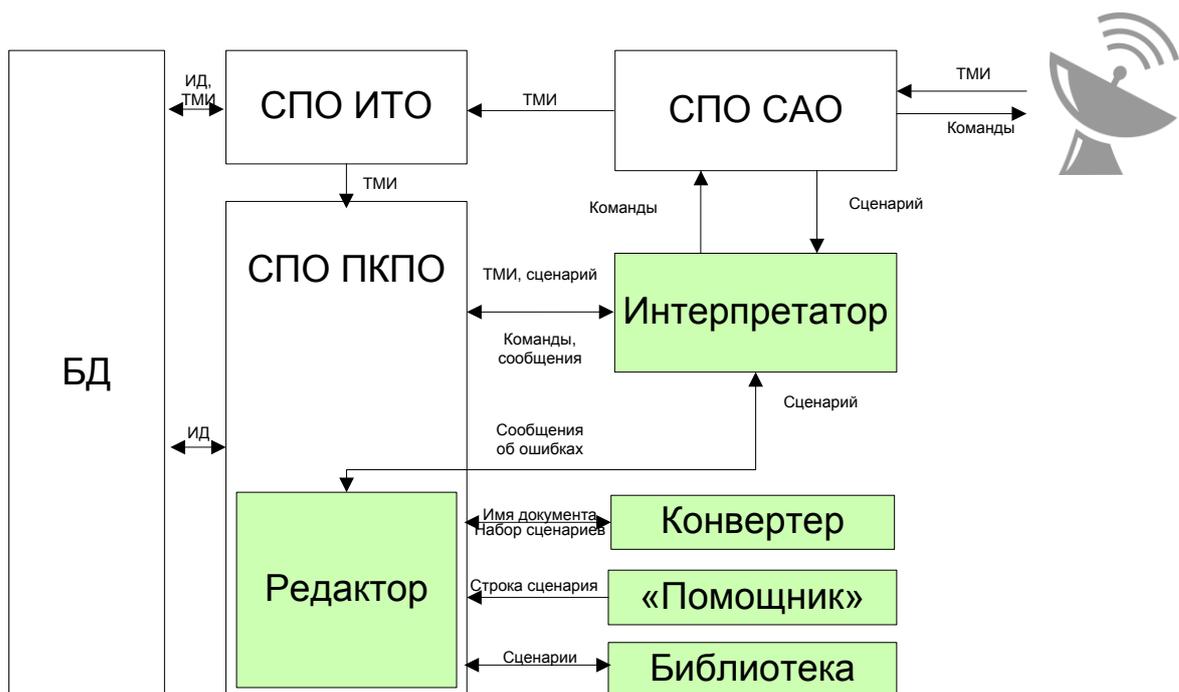


Рисунок 4.2 – Взаимодействие разработанных средств с существующим программным обеспечением управления КА

4.1.1 Интерпретатор языка «Дельта»

Программное обеспечение было спроектировано в соответствии с основными принципами построения трансляторов [56 – 73].

Интерпретатор реализован в виде динамической библиотеки [21, 74 – 77], содержащей набор целевых функций.

Интерпретатор взаимодействует со следующими программными компонентами: графической оболочкой разработанного ПО в части проверки создаваемых сценариев управления; СПО ПКПО в части выполнения сценариев управления в реальном времени в ручном и автоматизированном режимах; СПО САО для отработки сценариев управления в автоматическом режиме по наступлению назначенного времени. Для дальнейшего унифицированного описания назовем компоненту, с которой в данном случае взаимодействует интерпретатор как «Оболочка».

Оболочкой реализуются следующие целевые функции: получение текущих значений указанных телеметрических параметров, формирование команд, отправка команд по сети, и другие, поэтому основная задача реализуемой компоненты – обрабатывать тексты сценариев управления и рекомендовать оператору следующую команду для выдачи.

Поддерживается следующий алгоритм выполнения (рисунок 4.3):

1. Текст сценария управления попадает в лексический анализатор (Lexer), который определяет язык обрабатываемого сценария управления, тип синтаксиса (существующий или вновь разработанный), разделяет его на отдельные лексемы (минимальные неделимые частицы языка, например, числа, или операторы).

2. Массив лексем передаётся в модуль проверки синтаксиса (Checker), где идёт проверка правильности написания типовой работы в соответствии с назначенными языком («Дельта» или ЯОТР) и типом синтаксиса. Грамматика разработанного языка относится к классу контекстно-свободных, и для нее реализован разбор методом рекурсивного спуска (для каждого нетерминального символа грамматики разработана процедура, задача которой – найти подцепочку, выводящуюся из этого нетерминала). Согласно общепринятой классификации, синтаксический анализатор является нисходящим, использует последовательность разбора слева направо с просмотром вперед на 1 символ без возвратов. Реализованы только диагностика и нейтрализация ошибок.

Исправление ошибок не производится. Сообщения о найденных ошибках записываются в список и возвращаются в оболочку.

3. Компоновщик (Finder) просматривает сценарий на наличие других вызываемых сценариев, и возвращает их имена в «оболочку», которая, обращаясь к БД, подгружает тексты вызываемых сценариев управления.

4. Парсер (GetModel) формирует промежуточное представление сценария, которое передается в «оболочку», где оно отображается в удобочитаемом для пользователя виде. Человек-оператор может начать выполнение сценария с произвольного места: как с начала, так и с середины сценария. На этом цикл обработки типовой работы может быть остановлен.

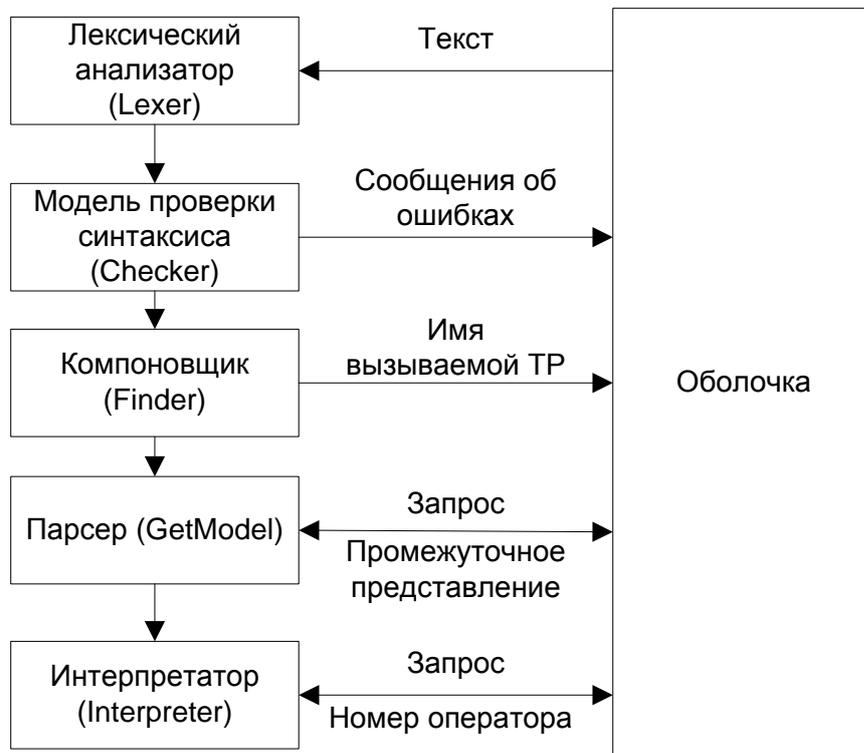


Рисунок 4.3 – Структура интерпретатора сценариев управления

Формат промежуточного представления – список структур, заполненных данными о последовательности операторов и их аргументах, в следующем формате:

```

struct TRLine
{

```

```
int nNumOper; // Порядковый номер оператора
int nLevelOper; // Уровень вложенности оператора
                // в условных блоках
QString arg1; // Аргумент 1
QString arg2; // Аргумент 2
QString arg3; // Аргумент 3
};
```

Правила заполнения аргументов различаются от одного оператора языка к другому.

5. После команды оператора на выполнение сценария управления оболочка последовательно вызывает интерпретатор («Interpreter»), получает от него номер следующей команды на исполнение, выполняет её, вновь вызывает интерпретатор, и так до тех пор, пока интерпретатор не вернёт пустой ответ. Данный процесс возможен как в ручном (выдача каждой команды подтверждается оператором) так и в автоматизированном (все возможные команды из данного сценария выдаются в автоматизированном режиме). Также возможна выдача только части команд из содержащихся в сценарии (оператор выбирает необходимые из списка).

4.1.2 Редактор сценариев управления

Текстовый редактор отвечает за отображение, редактирование и последующее сохранение текстов сценариев управления в БД и в текстовые файлы.

4.1.3 Конвертер

Описания отдельных сценариев управления (последовательность команд, обеспечивающих выполнение операции, а также методику выполнения и характеристики операции) включают в эксплуатационную документацию по управлению КА.

Сценарии управления создаются на основе эксплуатационной документации вручную. Данный процесс занимает значительное время, а также подвержен

влиянию человеческого фактора (итоговые сценарии управления могут содержать ошибки). Данный факт вызвал необходимость в разработке алгоритма автоматизированного заведения сценариев управления на основе анализа эксплуатационной документации.

Пример описания сценария управления, входящего в состав ЭД по управлению КА представлен в таблице 4.1. Описание сценария представлено в виде таблицы, содержащей набор выдаваемых команд, описание действия оператора, время выдачи команды, а также перечень ТМ-параметров, контроль которых производится для подтверждения успешности прохождения текущего шага.

Таблица 4.1 – Пример сценария управления

Номер шага	Действие оператора	Время	Команда	Контроль
1	Подрыв основного и резервного ПП в узле зачековки 1		1	ПП171=1; ПП172=1
2	Подрыв основного и резервного ПП в узле зачековки 2	00:00:10	2	ПП181=1; ПП182=1
3	Подрыв основного и резервного ПП в узле зачековки 3		3	ПП191=1; ПП192=1

Столбцы в данном формате описания сценариев управления имеют следующее назначение:

- «Номер шага» – порядковый номер выполняемого шага;
- «Действие оператора» – текстовое описание выполняемого действия;
- «Время» – задержка между выполнением предыдущего шага сценария управления и текущего;
- «Операция управления» – столбец, который может содержать следующие данные: номер выдаваемой команды или имя вызываемого сценария управления;

– «Контроль» – перечень ТМ-параметров, меняющих свое состояние в случае успешного выполнения операции, и их значений. Если в графе «Контроль» ТМ-параметр отсутствует, то контроль данного действия не производится.

Структура документа из состава эксплуатационной документации по управлению КА с помощью РБНФ описана далее.

Документ представляет собой набор фрагментов текста и таблиц.

Документ = { Текст | Таблица }

Текст представляет собой последовательность символов.

Текст = {буква | цифра | знак_препинания | разделитель}

**Буква = "А" | "Б" | "В" | ... "Я" | "а" | "б" | "в" | ...
"я" | "А" | "В" | "С" | ... "Z" | "а" | "b" | "с" | ... "z".**

**Цифра = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8"
| "9".**

Знак_препинания = "." | "," | ":" | "-" | "?" | "!" | ";"

Разделитель = " " | '\r\n'

Таблица представляет из себя набор строк, в свою очередь, состоящих из набора ячеек. Некоторые из таблиц, содержащихся в документе, содержат описание сценариев управления.

Таблица = Произвольная_таблица | Таблица_сценария

Произвольные таблицы содержат произвольное количество строк, состоящих из нерегламентированного числа ячеек, содержащих текстовые фрагменты, содержимое которых не анализируется.

Произвольная_таблица = {Произвольная_строка}

Произвольная_строка = {Произвольная_ячейка}

Произвольная_ячейка = текст

Таблицы, содержащие описание сценариев управления, имеют определенный формат: содержат пять столбцов и минимум две строки: одну строку заголовка, и минимум одну строку содержания. Строка заголовка может содержать произвольные символы и при использовании данного алгоритма не анализируется.

Таблица_сценария = Строка_заголовка {Строка_сценария}

Строка_заголовка = Произвольная_ячейка Произвольная_ячейка
 Произвольная_ячейка Произвольная_ячейка Произвольная_ячейка
 Строка_сценария = Ячейка_номера_шага Ячейка_описания
 Ячейка_паузы Ячейка_команды Ячейка_контроля

Каждая ячейка таблицы содержит данные определенного формата:
 содержит числовые, текстовые данные или данные о времени.

Ячейка_номера_шага = целое

Ячейка_описания = текст

Ячейка_паузы = <чч>:<мм>:<сс>

чч = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" |
 "9" ... "99".

мм = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" |
 "9" ... "59".

сс = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" |
 "9" ... "59".

Ячейка_команды = целое

Ячейка_контроля = условие {, условие }

Условие = ТМ_параметр < значение |

ТМ_параметр > значение |

ТМ_параметр = значение |

Значение < ТМ_параметр < Значение

ТМ_параметр = Буква{Буква|Цифра}.

Был разработан алгоритм подготовки сценариев управления с помощью автоматизированного анализа эксплуатационной документации [79, 80, 81], заключающийся в следующем: вместо ручного набора сценариев управления создавать их в автоматизированном режиме, на основе данных, содержащихся в эксплуатационной документации путем подстановки в соответствующие синтаксические конструкции языка.

Разработанный алгоритм заключается в следующем (рисунок 4.4): проанализировать указанный пользователем документ на наличие таблиц определенного формата (имеющих пять столбцов, с заголовком первого столбца «Номер шага»). Далее, последовательно переходя от одной таблицы указанного формата к другой, и от строки к строке в текущей обрабатываемой таблице, формировать текст сценария управления в соответствии с синтаксическими правилами языка управления, с учетом данных, содержащихся в соответствующих ячейках. Также предусмотрена возможность последующей корректировки полученного набора сценариев управления КА.

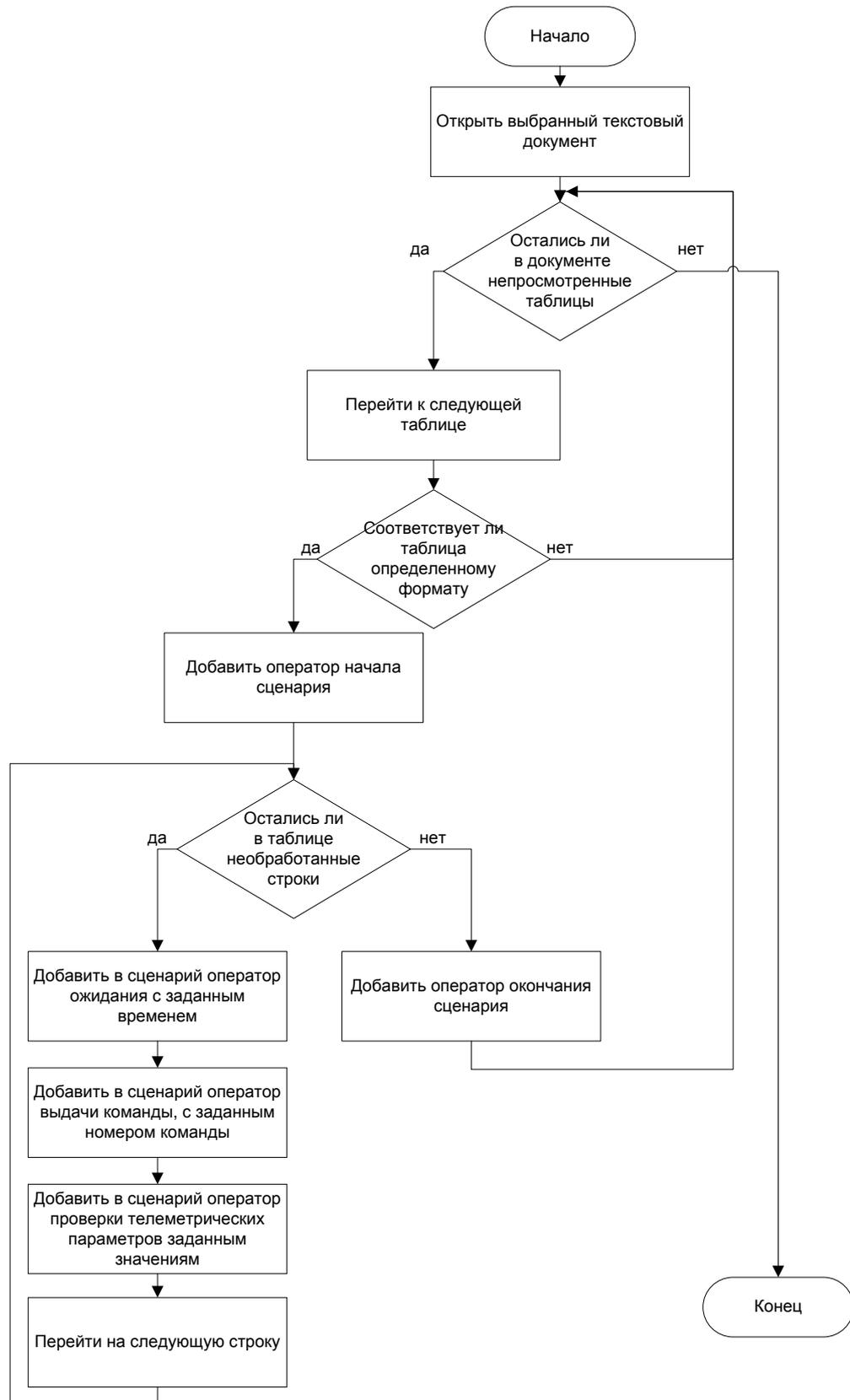


Рисунок 4.4 – Блок-схема алгоритма автоматизированного формирования сценариев управления на основе анализа ЭД

Сценарий управления, сгенерированный на основании данных, содержащихся в таблице 4.1, представлен далее.

ПРОГРАМ

ПЕЧАТЬ Подрыв основного и резервного ПП в узле зачековки 1

ВЫЗВАТЬ _ПК(1)

ЕСЛИ ПП171=1 И ПП172=1 ЗА 00:20:00

ИНАЧЕ

ПЕЧАТЬ ТМ параметры ПП171=1 И ПП172=1 не приняли заданное значение

СТОП

КЕСЛИ

ПАУЗА 00:00:10

ПЕЧАТЬ Подрыв основного и резервного ПП в узле зачековки 2

ВЫЗВАТЬ _ПК(2)

ЕСЛИ ПП181=1 И ПП182=1 ЗА 00:20:00

ИНАЧЕ

ПЕЧАТЬ ТМ параметры ПП181=1 И ПП182=1 не приняли заданное значение

СТОП

КЕСЛИ

ПЕЧАТЬ Подрыв основного и резервного ПП в узле зачековки 3

ВЫЗВАТЬ _ПК(3)

ЕСЛИ ПП191=1 И ПП192=1 ЗА 00:20:00

ИНАЧЕ

ПЕЧАТЬ ТМ параметры ПП191=1 И ПП192=1 не приняли заданное значение

СТОП

КЕСЛИ

КПРОГРАМ

Разработанный алгоритм был реализован в программной системе анализа ЭД [78]: пользователь выбирает документ, содержащий описание команд для выдачи на КА, после чего происходит автоматизированное формирование сценариев управления на основе полученных данных. Предусмотрена возможность последующей корректировки полученных сценариев управления.

4.1.4 Библиотека сценариев управления

Представляет собой набор наиболее используемых сценариев управления, подлежащих обязательной проверке и редактированию перед включением в набор исходных данных управления вновь создаваемыми КА. Фрагмент перечня сценариев представлен в таблице 2.2.

4.2 Реализация

Предложенный инструментарий был реализован с использованием языка C++ и библиотеки Qt.

Для реализации разработанного алгоритма был выбран объектно-ориентированный подход, как метод, обеспечивающий наименее затратную модификацию программы и обновление её функциональности.

В качестве системы хранения сценариев управления применяется существующая схема БД, содержащая исходные данные управления, а также статистику управления.

4.3 Интерфейс пользователя

4.3.1 Главное окно программы

Внешний вид редактора исходных текстов сценариев управления КА представлен на рисунке 4.5.

Левая часть содержит набор сценариев управления, полученных в результате анализа выбранного документа из состава эксплуатационной документации по управлению КА. Номер страницы с таблицей, на основе которой сформирован сценарий управления, указан справа от его имени. Красный значок рядом с работой – сценарий управления содержит ошибки, зеленый – не содержит. Перечень содержащихся в тексте сценария управления ошибок приведен в нижней части окна. Строки редактируемого сценария управления, содержащие ошибки, подкрашиваются красным цветом.

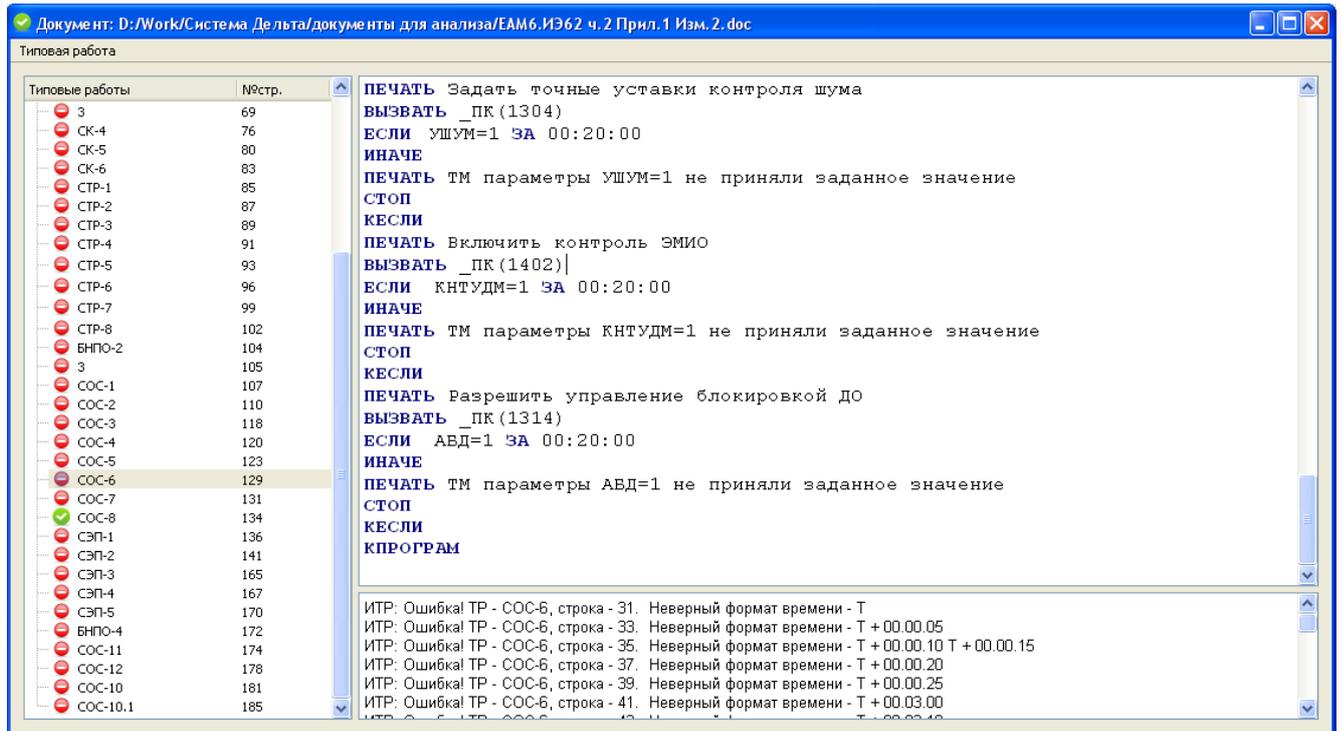


Рисунок 4.5 – Интерфейс пользователя редактора исходных текстов

Полученные тексты сценариев управления являются редактируемыми, внесенные в их текст изменения можно сохранять, кликнув на другой сценарий в дереве. Возможен перевод ключевых слов, содержащихся в сценарии с русского на английский язык, и в обратном направлении. Также возможна конвертация текста сценария в направлении старый – новый синтаксис (поддерживается для операторов, содержащихся в обоих вариантах языка).

Ключевые слова языка управления КА подкрашиваются синим цветом.

Тексты сценариев управления возможно сохранять в указанную БД определенной структуры или в текстовые файлы.

4.3.2 Конвертер, преобразующий эксплуатационную документацию в набор сценариев управления

Окно настройки параметров запуска конвертера представлено на рисунке 4.6.

Поддерживается автоматизированное создание сценариев управления как на существующем языке ЯОТР, так и на вновь разработанном языке «Дельта»; как на основе русскоязычного синтаксиса, так и англоязычного.

Помимо этого, можно выбирать и состав итоговых сценариев управления КА: исключать из анализа или, наоборот, включать в анализ данные, соответствующие следующим операторам: «Пауза», «Сообщение», «Команда», «Контроль».

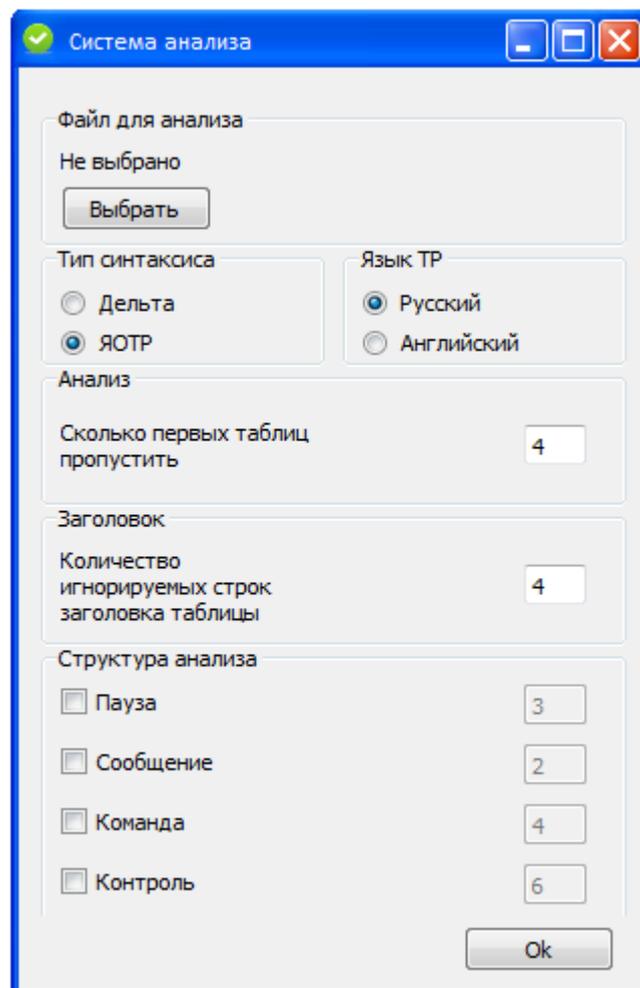


Рисунок 4.6 – Окно настроек анализа эксплуатационной документации

Заголовок таблицы описания сценария управления может содержать несколько строк незначимой информации, поэтому предусмотрена возможность исключения их из рассмотрения. Число игнорируемых строк является настраиваемой из интерфейса пользователя величиной.

После завершения анализа список сформированных сценариев управления отображается в главном окне программы. Сценарии управления, содержащие синтаксические ошибки, будут выделены красным цветом.

4.3.3 «Помощник» по заведению текстов сценариев управления

Интерфейс пользователя «Помощника» по заведению сценариев управления КА представлен на рисунке 4.7.

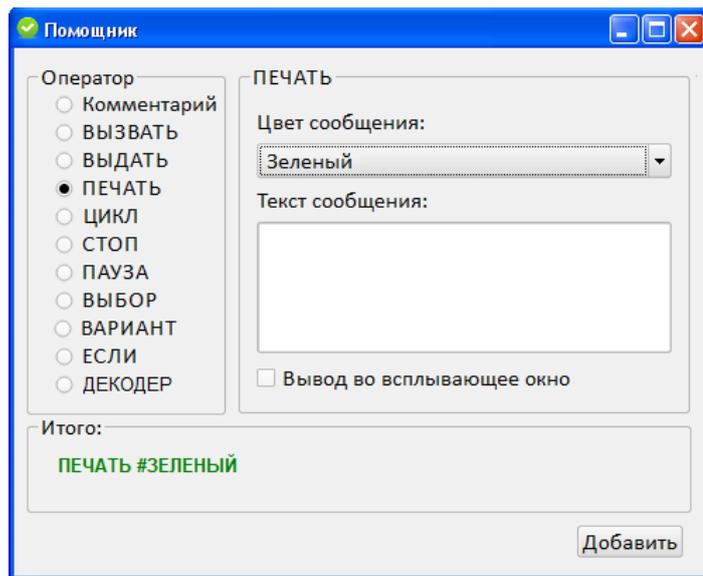


Рисунок 4.7 – Интерфейс пользователя «Помощника» по заведению сценариев управления КА

Пользователю необходимо выбрать требуемый оператор из списка и заполнить появившиеся поля. После нажатия кнопки «Добавить» выбранный оператор языка вставляется в текущую позицию курсора.

Поддерживаются как русскоязычный, так и англоязычный синтаксис создаваемых сценариев.

4.4 Сравнение размерности и времени заведения сценариев на языках ЯОТР и «Дельта»

Для анализа разработанных языка и алгоритмов заведения сценариев управления в качестве критериев сравнения были выбраны следующие характеристики:

- время, затрачиваемое на заведение сценариев управления с использованием существующего (ручного) и разработанного (автоматизированного) методов заведения сценариев;
- количество символов в итоговых сценариях управления, составленных с использованием эксплуатируемого языка ЯОТР и разработанного языка «Дельта»;
- количество строк в итоговых сценариях управления, составленных с использованием эксплуатируемого языка ЯОТР и разработанного языка «Дельта».

В качестве материала для анализа случайным образом было выбрано несколько документов, относящихся к различным проектам АО «Информационные Спутниковые Системы». Перечень документов, а также их характеристики (число страниц, число содержащихся в них описаний сценариев управления) указаны в таблице 4.2.

Полученные экспериментальные данные сравнения представлены в таблицах 4.3, 4.4, 4.5 и на рисунках 4.8, 4.9, 4.10.

Таблица 4.2 – Перечень документов, использованных для оценок характеристик созданных алгоритмов

Название документа	Дальнейшее обозначение по тексту	Число страниц	Число сценариев управления, содержащихся в документе
Экспресс-АТ1. Инструкция по разработке программы полета. Часть вторая. Типовая программа полета Приложение 2. Программа полета на этапе эксплуатации	Документ 1	44	16
Экспресс-АТ1. Инструкция по разработке программы полета. Часть вторая. Типовая программа полета Приложение 1 Программа ввода в эксплуатацию	Документ 2	169	40
Спутник «KazSat-3». Справочник по эксплуатации Часть 1 Управление спутником	Документ 3	210	61
Спутник «KazSat-3» Программа испытаний на орбите	Документ 4	129	41
Космический аппарат «Луч-5Б». Инструкция по подготовке исходных данных для управления в полете. Часть вторая. Инструкция по разработке программы полета. Приложение 1 Программа прохождения начальных режимов и проверок на функционирование обеспечивающих систем КА	Документ 5	140	46
Космический аппарат «Луч-5А» Инструкция по подготовке исходных данных для управления в полете Часть вторая Инструкция по разработке программы полета Приложение 2 Типовые программы управления КА при штатной эксплуатации	Документ 6	107	100

Таблица 4.3 – Сравнение времени, затраченного на заведение сценариев управления с использованием существующего (ручного) и разработанного (автоматизированного) методов заведения сценариев

Обозначение документа	Время, затраченное при ручном вводе сценариев управления (приблизит.), сек.	Время, затраченное при автоматизированном вводе сценариев управления, сек.	Время, затраченное при автоматизированном вводе сценариев управления, с учетом корректировки, сек.
Документ 1	28800	3	1200
Документ 2	144000	16	7200
Документ 3	72000	13	7200
Документ 4	144000	16	7200
Документ 5	72000	10	7200
Документ 6	72000	10	7200

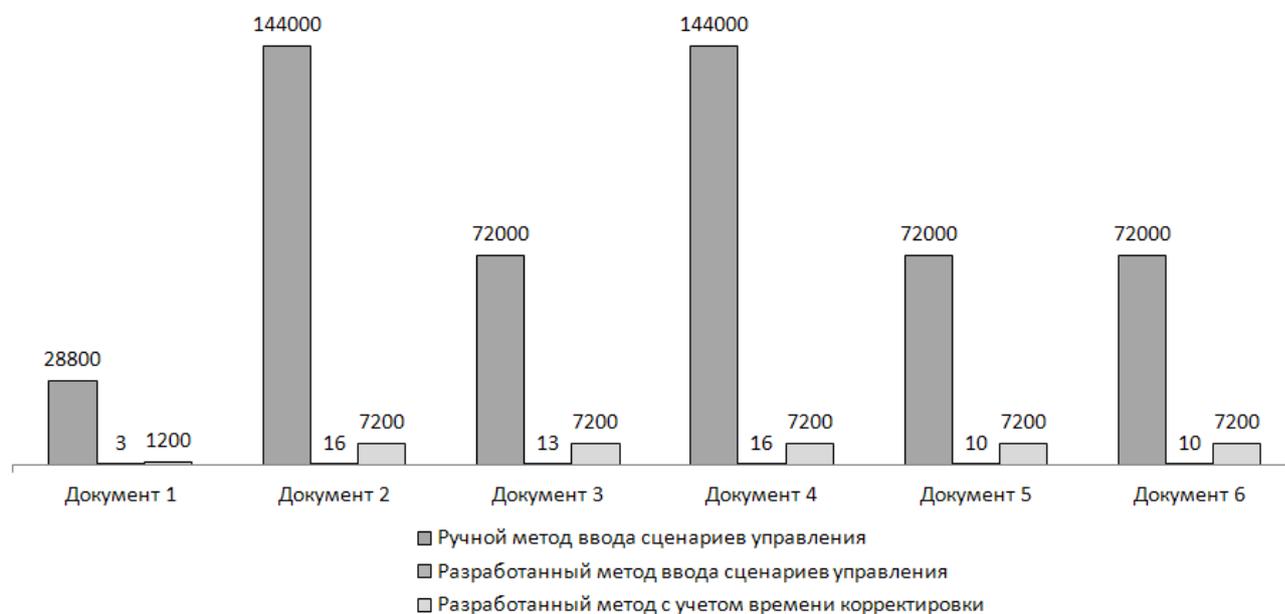


Рисунок 4.8 – Сравнение времени, затрачиваемого на составление сценариев управления по существующим эксплуатационным документам

Таблица 4.4 – Сравнение количества символов в итоговых сценариях управления, составленных с использованием эксплуатируемого языка ЯОТР и разработанного языка «Дельта»

Обозначение документа	Количество символов в сценариях управления, созданных с помощью языка ЯОТР, симв.	Количество символов в сценариях управления, созданных с помощью языка «Дельта», симв.
Документ 1	31557	3000
Документ 2	318567	29405
Документ 3	166451	17110
Документ 4	359588	35911
Документ 5	177300	17127
Документ 6	172195	16643

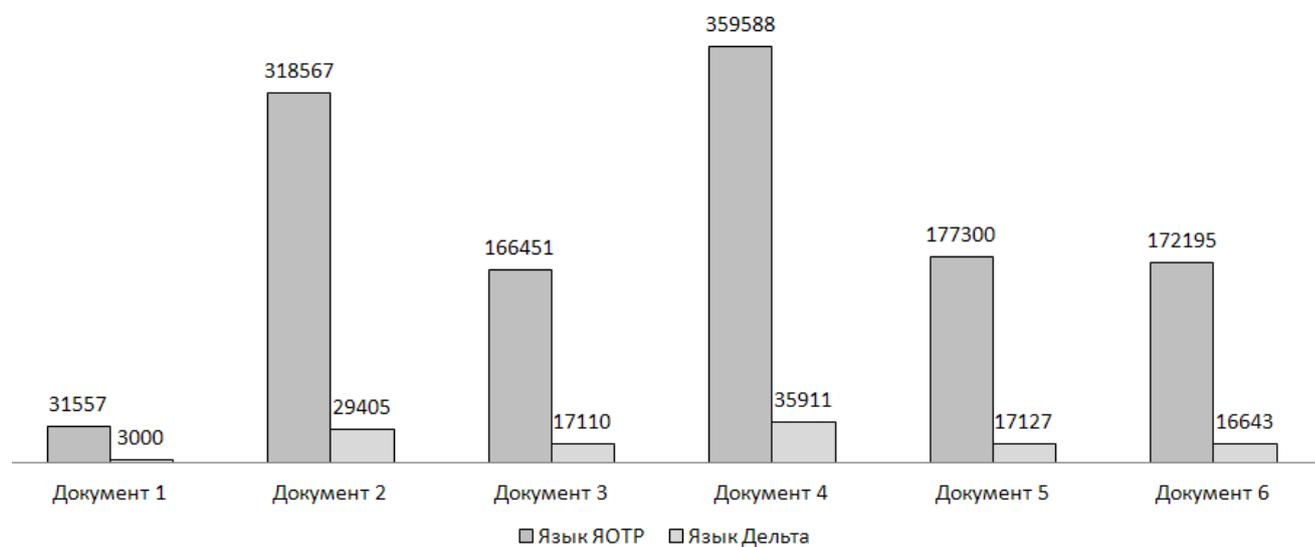


Рисунок 4.9 – Сравнение количества символов в итоговых сценариях управления, составленных с использованием эксплуатируемого языка ЯОТР и разработанного языка «Дельта»

Таблица 4.5 – Сравнение количества строк в итоговых сценариях управления, составленных с использованием эксплуатируемого языка ЯОТР и разработанного языка «Дельта»

Обозначение документа	Число строк в сценариях управления, созданных с помощью языка ЯОТР	Число строк в сценариях управления, созданных с помощью языка «Дельта»
Документ 1	668	288
Документ 2	6771	2576
Документ 3	3241	1674
Документ 4	6714	3860
Документ 5	3685	1530
Документ 6	3578	1489

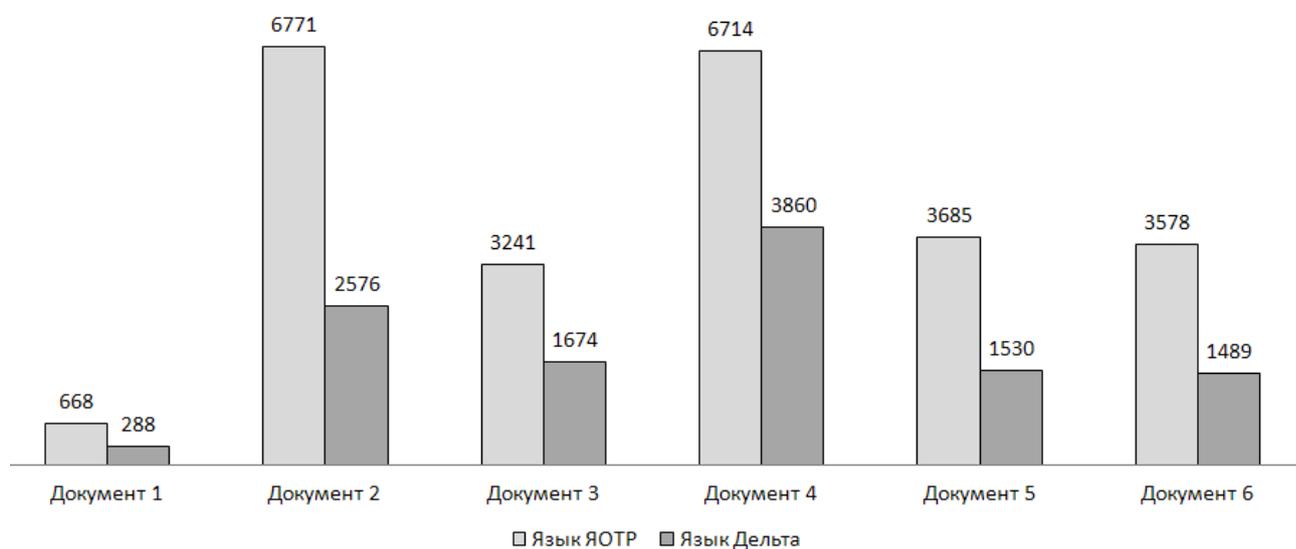


Рисунок 4.10 – Сравнение количества строк в итоговых сценариях управления, составленных с использованием эксплуатируемого языка ЯОТР и разработанного языка «Дельта»

Анализ полученных данных [81] показывает увеличение эффективности подготовки исходных данных управления КА: время, затрачиваемое на создание сценариев управления КА сократилось приблизительно в 10 раз. Сократилась также и размерность сценариев управления: количество строк сократилось приблизительно в два раза, количество символов сократилось приблизительно в десять раз, за счет удаления не значащих пробелов, без потери информационной составляющей.

Выводы по главе 4

Для осуществления инструментальной поддержки создания сценариев управления на разработанном языке управления КА «Дельта» были созданы инструментальные средства создания и исполнения сценариев управления КА следующей структуры:

- интерпретатор языка «Дельта», осуществляющий проверку, а также выполнение сценариев управления;
- «Помощник» по вводу сценариев управления – предназначен для упрощения ввода текстов сценариев управления – осуществляет подстановку введенных пользователем данных в соответствующие конструкции языка и добавление этих конструкций к тексту редактируемого сценария;
- редактор сценариев управления, отвечающий за отображение, редактирование и последующее сохранение текстов сценариев;
- конвертер, в автоматизированном режиме преобразующий эксплуатационную документацию по управлению КА в набор сценариев управления.

Сравнение разработанного и существующего методов заведения сценариев управления показывает значительное уменьшение времени создания сценариев управления КА в случае использования спроектированного метода автоматизированного анализа документации, что является неоспоримым преимуществом разработанной системы.

Сравнение разработанного и существующего языков управления КА показывает уменьшение размерности итоговых сценариев управления, что является одним из преимуществ разработанного языка.

Сравнение разработанного и эксплуатируемого программного обеспечения показывает расширение функциональных возможностей созданных инструментальных средств.

ЗАКЛЮЧЕНИЕ

Работа посвящена повышению эффективности и надежности процессов подготовки сценариев управления космическими аппаратами, в частности, исследованию, проектированию и разработке языковых и инструментальных средств создания и исполнения сценариев управления космическими аппаратами.

Научная новизна проведенного исследования заключается в следующем:

- на основе анализа предметной области предложена модель предметной области, включающая структуру командного метода управления КА, и определяющая функциональный состав языковых и инструментальных средств управления КА;

- разработан предметно-ориентированный язык программирования, повышающий эффективность разработки сценариев управления КА;

- разработаны инструментальные средства подготовки и выполнения сценариев управления КА, обеспечивающие повышение надежности и эффективности поддержки командного метода управления КА.

Разработаны программные продукты, осуществляющие поддержку процесса создания и исполнения сценариев управления КА:

- интерпретатор, поддерживающий выполнение сценариев управления в реальном времени в ручном и автоматизированном режимах во время сеанса связи с КА, что обеспечивает практическое использование полученных результатов;

- библиотека стандартных сценариев управления КА, для создания наборов сценариев управления перспективными КА;

- программная система, обеспечивающая инструментальную поддержку процессов создания и выполнения сценариев управления КА, а также конвертер, в автоматизированном режиме преобразующий эксплуатационную документацию по управлению КА в набор сценариев управления.

Опыт эксплуатации разработанных языковых и инструментальных средств создания и исполнения сценариев управления КА показал, что их применение

минимизирует факт появления ошибок, так как исключает ввод текстов сценариев управления вручную. Помимо этого, снижается размерность итоговых сценариев управления, а также суммарное время, затрачиваемое на подготовку сценариев.

Предложенный подход к организации языковых и инструментальных средств создания и исполнения сценариев управления КА применим для автоматизации процесса управления КА различного назначения для предприятий ракетно-космической отрасли.

СПИСОК ЛИТЕРАТУРЫ

- 1 Соловьев, В.А. Управление космическими полетами: учебное пособие: в 2 ч. / В.А.Соловьев, Л.Н.Лысенко, В.Е.Любинский. – М.: Изд-во МГТУ им.Н.Э.Баумана, 2009. – 902 с.
- 2 Управление спутниками в полете [Электронный ресурс] Режим доступа: <http://www.iss-reshetnev.ru/capabilities>.
- 3 Чеботарев, В.Е. Основы проектирования космических аппаратов информационного обеспечения / В.Е.Чеботарев, В.Е.Косенко. – Красноярск: Сиб. гос. аэрокосмич. ун-т, 2011. – 488 с.
- 4 Зюзина, В.И. Методы управления космическими аппаратами / В.И.Зюзина, Н.А.Космынина // Материалы XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых «Молодежь и современные информационные технологии» – 2015. – С. 32-33.
- 5 Chaudhri, G. A Model for a Spacecraft Operations Language [Электронный ресурс] / G.Chaudhri, J.Cater, B.Kizzort // Режим доступа: <http://arc.aiaa.org/doi/pdf/10.2514/6.2006-5708>.
- 6 Chaudhri, G. Ground Systems. The Need for Standardization [Электронный ресурс] / G.Chaudhri, S.Hollander // Режим доступа: <http://arc.aiaa.org/doi/pdf/10.2514/6.2004-333-178>.
- 7 Kepler MOC. Architecture and Design Notebook. Flight Operations Subsystem OASIS-CC Software [Электронный ресурс] Режим доступа: http://www.stsci.edu/old/old/vhost/keplerdmc/GSCDR/MOC/Architecture_Design_Notes/MOC_ADN_5_Flight_Operations/MOC_ADN_5_FO_OASIS-CC.pdf.
- 8 Космынина, Н.А. Языки управления космическими аппаратами / Н.А.Космынина // Электронный журнал «Труды МАИ». – 2015. – № 81.
- 9 Dan, Y.U. Design and Implementation of Spacecraft Automatic Test Language [Электронный ресурс] / Y.U.Dan, M.A.Shilong // Режим доступа: <https://www.sciencedirect.com/science/article/pii/S1000936111600344>.

- 10 Bracco, C. SCOPE product: the Thales Alenia Space Unified Operations Preparation Environment [Электронный ресурс] / C.Bracco, N.Richasse, D.Portilla // Режим доступа: <https://arc.aiaa.org/doi/pdf/10.2514/6.2018-2313>.
- 11 Garcia, G. Use of Python as a satellite operations and testing automation language [Электронный ресурс] / G.Garcia // Режим доступа: <http://csse.usc.edu/GSAW/gsaw2008/s6/garcia.pdf>.
- 12 Palsson, M. 21st century operational procedure automation with Python and autofly. Lessons learned from migrating legacy systems and developing new ones. [Электронный ресурс] / M.Palsson, G.Garcia, T.Morel // Режим доступа: https://info.aiaa.org/tac/SMG/SOSTC/Workshop%20Documents/2010/Palsson_GMV_Operations_Automation_Python%5B1%5D.pdf.
- 13 Hifly. The complete solutions for satellite operations [Электронный ресурс] Режим доступа: <https://www.gmv.com/en/Products/hifly/components/>.
- 14 The complete solution for satellite fleet operations [Электронный ресурс] Режим доступа: https://www.gmv.com/DocumentosPDF/hifly/hifly_brochure.pdf.
- 15 Noguero, J. Limitless Automation: ANY CONOPS. [Электронный ресурс] / J.Noguero, E.Fraga, L.Blanco, A.Pizarro // Режим доступа: <http://www.spaceops2012.org/proceedings/documents/id1272487-Paper-001.pdf>.
- 16 Garcia, G. Migration of the Operations of a Legacy Multi-Satellite System to a state-of-the-art Ground System. [Электронный ресурс] / G.Garcia, M.Palsson, T.Morel, T.Beech, P.Honold, O.Fuente, A.Cerviño // Режим доступа: <https://arc.aiaa.org/doi/abs/10.2514/6.2009-6622>.
- 17 GMV 2012 User's Conference GMV Portfolio [Электронный ресурс] Режим доступа: http://www.gmv.com/export/sites/gmv/DocumentosPDF/focusSuite/GUC2012_Portfolio_GMV.pdf.
- 18 Баранова, Т.П. Автоматизированная испытательная система [Электронный ресурс] / Т.П.Баранова, В.Г.Буликов, В.Ю.Вершубский, С.А.Гайфулин,

- В.В.Луцикович, Г.Ю.Молчанова, Т.В.Семенова, М.Р.Шура-Бура// Режим доступа: http://www.keldysh.ru/papers/2008/prep29/prep2008_29.html.
- 19 Брега, А.Н. Командно-программное управление полетом российского сегмента МКС / А.Н.Брега, А.А.Коваленко // Космическая техника и технологии – 2016. – № 2. – С. 90-104.
 - 20 Барков, А.В. Структура проблемно-ориентированного языка испытаний космических аппаратов / А.В.Барков // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2006. – № 5. – С. 15-18.
 - 21 Космынина, Н.А. Разработка программного обеспечения автоматизации управления космическими аппаратами / Н.А.Космынина // Сборник трудов VIII Всероссийской научно-практической конференции студентов, аспирантов и молодых учёных «Молодёжь и современные информационные технологии. – 2010. – С.15-17.
 - 22 Спутник «AMOS-5» Руководство по эксплуатации ЦУП. Часть 2. Планирование и командно-программное обеспечение. – ОАО «ИСС». – 2011. – 212 с.
 - 23 Kratos Integral Systems International Products [Электронный ресурс] Режим доступа: <http://www.integ.com/Products.html>.
 - 24 EPOCH T&C Server. Real-Time Telemetry and Command Processing [Электронный ресурс] Режим доступа: <http://www.integ.com/ProductBriefPDF/EPOCH%20TC%20Server%20Datasheet.pdf>.
 - 25 Byrne, J. Commanding in EPOCH for the Metop Spacecraft [Электронный ресурс] / J.Byrne, J.Consiglio // Режим доступа: <http://arc.aiaa.org/doi/pdf/10.2514/6.2006-5591>.
 - 26 Harris OS/COMET. Satellite Command and Control [Электронный ресурс] Режим доступа: http://itservices.harris.com/media/OSmain_tcm40-22359.pdf.

- 27 Harris Technical Services Corporation. OS/COMET System Overview [Электронный ресурс] Режим доступа: http://itservices.harris.com/media/Systems-Overview_tcm40-22365.pdf.
- 28 OS/COMET. The industry's leading product for satellite command and control [Электронный ресурс] Режим доступа: <http://itservices.harris.com/oscomet.aspx>.
- 29 InControl. Satellite Command and Control Software for On-Orbit, Factory Test and Ground System Monitor and Control [Электронный ресурс] Режим доступа: http://www2.l-3com.com/tw/pdf/datasheets/InControl_Brochure_2010.pdf.
- 30 Allen, D. InControl-NextGeneration: An Ops Concept Driven Architecture for Satellite Command and Control [Электронный ресурс] / D.Allen // Режим доступа: <http://sunset.usc.edu/GSAW/gsaw2002/s10d/allen.pdf>.
- 31 Konishi, H. Spacecraft control system Development. [Электронный ресурс] / H.Konishi, M.Miyoshi, M.Sugawara // Режим доступа: <https://arc.aiaa.org/doi/10.2514/6.2002-T3-27>.
- 32 Rhea Group. Contributes to the operations of more than 80 space missions [Электронный ресурс] Режим доступа: <http://www.rheagroup.com/products/mois/test-harness/>.
- 33 Савенко, И.И. Метод автоматизации процесса реконфигурации бортового ретранслятора спутника связи / И.И.Савенко, М.С.Суходоев, Н.А.Космынина, Д.Н.Рыжков, С.Г.Цапко // Глобальный научный потенциал. –2013. – № 11 (32) – С. 94- 98.
- 34 Модернизированный унифицированный космический комплекс «Луч-М» Наземный комплекс управления Специальное программное обеспечение управления (СПО-У) Специальное программное обеспечение планирования и командно-программного обеспечения (СПО ПКПО) Комплекс программ проведения сеанса управления КА и средствами НКУ (КП ПСУ) Руководство оператора. – ОАО «ИСС». – 2012. – 55 с.

- 35 Ares. Procedure Development and Execution Suite [Электронный ресурс] Режим доступа: <http://www.kratoscomms.com/-/media/kisi/pdf/datasheet-ares.pdf>.
- 36 Using ARES to Improve Efficiency in Satellite Operations [Электронный ресурс] Режим доступа: http://www.powershow.com/view/1c44e-NjEwM/Using_ARES_to_Improve_Efficiency_in_Satellite_Operations_powerpoint_ppt_presentation.
- 37 Seymour, M.A. / The PLUTO operations procedure language and its use for RADARSAT-2 mission operations [Электронный ресурс] / M.A.Seymour // Режим доступа: <http://arc.aiaa.org/doi/pdf/10.2514/6.2004-538-333>.
- 38 Recher, S. Lessons learned from 10 years building ground M&C Systems [Электронный ресурс] / S.Recher, C.Stöcker, M.Brüggen // Режим доступа: <http://www.spaceops2012.org/proceedings/documents/id1291466-Paper-003.pdf>.
- 39 Ground Software. SCL for Ground Workstations and Servers [Электронный ресурс] Режим доступа : www.sra.com/scl/ground/.
- 40 Use of Spacecraft Command Language for Advanced Command and Control Applications. [Электронный ресурс]. Режим доступа: <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110015644.pdf>.
- 41 Spacecraft Command Language [Электронный ресурс] Режим доступа: <http://www.sw-ctrl.com/spacecraft-command-language-scl>.
- 42 Pereira, T.D. / Using the PLUTO language on functional tests of a Brazilian Satellite's On-Board Data Handling Computer [Электронный ресурс] / T.D.Pereira, M.G.V.Ferreira, F.N.Kucinskis // Режим доступа: <http://www.spaceops2012.org/proceedings/documents/id1259165-Paper-002.pdf>.
- 43 Баранова, Т.П. Автоматизированная испытательная система. Имитаторы / [Электронный ресурс] / Т.П.Баранова // Режим доступа: <http://library.keldysh.ru/preprint.asp?id=2011-19>.
- 44 Модернизированный унифицированный космический комплекс «Луч-М» Наземный комплекс управления Специальное программное обеспечение

- управления (СПО-У) Специальное программное обеспечение планирования и командно-программного обеспечения (СПО ПКПО) Комплекс программ подготовки и исходных данных (КП ПИД) Руководство оператора. – ОАО «ИСС». – 2012. – 22 с.
- 45 Иванов, Д.С. Управление полетом космического аппарата [Электронный ресурс] / Д.С.Иванов // Режим доступа: <http://www.cosmos-journal.ru/articles/936/>.
- 46 Некрасов, М.В. Проблемы обработки телеметрической информации в контуре автоматизированной системы управления космическими аппаратами/ М.В.Некрасов, Д.Н.Пакман, А.Н.Антамошкин // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2009. – № 1. – С. 4-9.
- 47 Космынина, Н.А. Разработка языка управления космическими аппаратами / Н.А.Космынина, А.И.Легалов, А.В.Барков, А.А.Лапин // Информационно-управляющие системы / Санкт-Петербург, 2015, №5(78) – С. 82 – 90.
- 48 Фаулер, М. Предметно-ориентированные языки программирования / М.Фаулер. – М.; СПб.; Киев: Вильямс, 2011. – 576 с.
- 49 Фаулер, М. Языковой инструментарий: новая жизнь языков предметной области [Электронный ресурс] / М.Фаулер // Режим доступа: <http://www.maxkir.com/sd/languageWorkbenches.html>.
- 50 Anderson, C. DSL Tools: [Электронный ресурс] / C.Anderson // Режим доступа: <http://www.codeproject.com/KB/cs/DSLTools.aspx>.
- 51 Creating Domain-Specific Languages [Электронный ресурс] Режим доступа: [http://msdn2.microsoft.com/en-us/library/bb126259\(VS.90\).aspx](http://msdn2.microsoft.com/en-us/library/bb126259(VS.90).aspx).
- 52 Dmitriev, S. Language Oriented Programming: The Next Programming Paradigm: [Электронный ресурс] / S.Dmitriev / Режим доступа: <http://www.onboard.jetbrains.com/is1/articles/04/10/lop/index.html>.
- 53 Ботов, Д.С. Обзор современных средств создания и поддержки предметно-ориентированных языков программирования / Д.С.Ботов // Вестник Южно-

- Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. – 2013. – Т. 13. № 1. – С. 10 - 15.
- 54 Ботов, Д.С. Возможности и проблемы использования современного языкового инструментария в разработке на предметно-ориентированных языках программирования / Д.С.Ботов // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. – 2013. – Т. 13. № 2. – С. 128 -130.
- 55 Космынина, Н.А. Специальное программное обеспечение планирования и командно-программного обеспечения / А.А.Лапин, А.В.Петров, Н.А.Космынина и др. // Свидетельство о государственной регистрации программы для ЭВМ № №2014619652. – М.: Федеральная служба по интеллектуальной собственности, патентам и товарным знакам. – 2014.
- 56 Васильев, С.С. Об использовании в программировании проблемно-ориентированных языков / С.С.Васильев, В.Б.Новосельцев // Известия Томского политехнического университета. – 2008. – Т. 313. № 5. – С. 68-72.
- 57 Легалов, А.И. Курс лекций по предмету «Основы разработки трансляторов» / А.И.Легалов // [Электронный ресурс] Режим доступа: www.softcraft.ru.
- 58 Брежнев, А.М. Конспект лекций по дисциплине «Системное программное обеспечение». Тема: «Трансляторы» / А.М.Брежнев // [Электронный ресурс] Режим доступа: <http://www.kit.kulichki.ru/>.
- 59 Серебряков, В.А. Основы конструирования компиляторов / В.А.Серебряков, М.П.Галочкин // [Электронный ресурс]. Режим доступа: <http://www.citforum.ru/programming/theory/serebryakov>.
- 60 Ахо, А.В. Компиляторы: принципы, технологии и инструменты / А.В.Ахо, Р.Сети, Д.Ульман – Москва, 2004 – 783 с.
- 61 Волкова, И.А. Формальные грамматики и языки. Элементы теории трансляции: Учебное пособие / И.А.Волкова, Т.В.Руденко. – М.: МГУ, 1999. – 783 с.

- 62 Ахо, А. Теория синтаксического анализа, перевода и компиляции. Т 1,2/ А.Ахо, Дж.Ульман. – М.: Мир, 1979. – 552 с.
- 63 Мартыненко, Б.К. Языки и трансляции / Б.К.Мартыненко. – СПб.: Издательство С.-Петербургского университета, 2003. – 235 с.
- 64 Мозговой, М. Классика программирования: алгоритмы, языки, автоматы, компиляторы / М.Мозговой. – М.: Наука и техника, 2006. – 316 с.
- 65 Молчанов, А.Ю. Системное программное обеспечение: Учебник для вузов/ А.Ю.Молчанов. – СПб.: Питер, 2010. – 400 с.
- 66 Свердлов, С.З. Языки программирования и методы трансляции: Учебное пособие / С.З.Свердлов – СПб.: Питер, 2007. – 638 с.
- 67 Опалева, Э.А. Языки программирования и методы трансляции / Э.А.Опалева В.П.Самойленко. – СПб.: БХВ-Петербург, 2005. – 480 с.
- 68 Алёшин, А.В. Теория языков программирования и методы трансляции / А.В.Алёшин. – Краснодар: ИМСИТ, 2005. – 19 с.
- 69 Хопкрофт, Дж. Введение в теорию автоматов, языков и вычислений / Дж.Хопкрофт, Р.Мотвани, Дж.Ульман. – Вильямс, 2002. – 528 с.
- 70 Пратт, Т. Языки программирования: разработка и реализация/ Т.Пратт, М.Зелковиц. – СПб.: Питер, 2002. – 688 с.
- 71 Карпов, Ю.Г. Теория и технология программирования. Основы построения трансляторов / Ю.Г.Карпов. – СПб.: БХВ-Петербург, 2005. – 271 с.
- 72 Абельсон, Х. Структура и интерпретация компьютерных программ / Х.Абельсон, Д.Д.Сассман. – Добросвет, 2004. – 608 с.
- 73 Рейуорд-Смит, В.Дж. Теория формальных языков / В.Дж.Рейуорд-Смит. – М.: Радио и связь, 2003. – 128 с.
- 74 Космынина, Н.А. Разработка интерпретатора командных скриптов по управлению КА как части специального программного обеспечения центра управления полётами / Н.А.Космынина, А.А.Лапин // Материалы научно-технической конференции молодых специалистов ОАО «Информационные Спутниковые Системы» имени академика М.Ф.Решетнева» «Разработка,

- производство, испытания и эксплуатация космических аппаратов и систем». – 2011. – С. 22-23.
- 75 Космынина, Н.А. Разработка интерпретатора командных скриптов для обеспечения автоматизированного управления космическим аппаратом / Н.А.Космынина, А.И.Легалов // Теоретические и прикладные проблемы науки и образования в 21 веке. – 2012. – С.57-58.
- 76 Космынина, Н.А. Разработка интерпретатора скриптов управления космическим аппаратом / Н.А.Космынина, А.И.Легалов // Решетневские чтения: материалы XVI Международной научной конференции, посвященной памяти генерального конструктора ракетно-космических систем акад. М.Ф. Решетнева. – 2012. Ч. 2 – С. 572-573.
- 77 Космынина, Н.А. Разработка интерпретатора скриптов управления космическим аппаратом / Н.А.Космынина, А.А.Лапин, А.И.Легалов // Общероссийская молодежная научно-техническая конференция «Молодежь. Техника. Космос». – 2013. – С.221-222.
- 78 Космынина, Н.А. Разработка метода автоматизации подготовки исходных данных и эксплуатационной документации по управлению КА / Н.А.Космынина, А.А.Лапин // III Научно-техническая конференция молодых специалистов «Разработка, производство, испытания и эксплуатация космических аппаратов и систем». – 2014. – С. 56-57.
- 79 Космынина, Н.А. Метод автоматизации подготовки типовых работ по управлению космическими аппаратами / Н.А.Космынина // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2015. – С.183-187
- 80 Космынина, Н.А. Разработка языка и инструментальных средств управления КА / Н.А.Космынина, А.А.Лапин, А.И.Легалов // Актуальные проблемы космонавтики: Труды XXXIX академических чтений по космонавтике, посвященных памяти академика С.П. Королева и других выдающихся

отечественных ученых-пионеров освоения космического пространства. – 2015. – С. 305.

- 81 Космынина, Н.А. Разработка средств автоматизации управления космическими аппаратами / Н.А.Космынина // Решетневские чтения; материалы XIX Международной Научно-практической конференции, посвященной 55-летию Сибирского Государственного Аэрокосмического Университета им. акад. М.Ф. Решетнева. – 2015. – С.180-181.

СПИСОК СОКРАЩЕНИЙ

API	– Application Programming Interface;
CCSDS	– Consultative Committee for Space Data Systems, Международный Консультативный Комитет по космическим системам передачи данных;
ESA	– European Space Agency;
ESA PSS	– European Space Agency Procedures, Specifications & Standards;
GMV	– GMV Space Systems Inc.;
LASP	– Laboratory for Atmospheric and Space Physics;
NASA	– National Aeronautics and Space Administration;
NASDA	– National Space Development Agency of Japan;
АБ	– аккумуляторные батареи;
АО «ИСС»	– Акционерное Общество «Информационные спутниковые системы» имени академика М.Ф. Решетнёва»;
БА	– бортовая аппаратура;
БАТС	– бортовая аппаратура телесигнализации;
БД	– база данных;
БК	– бортовой компьютер;
БКУ	– бортовой комплекс управления;
БПО	– бортовое программное обеспечение;
БУ БКУ	– блок управления бортового комплекса управления;
ЖЦ	– жизненный цикл;
ИТНП	– измерение текущих навигационных параметров;
ИТО	– информационно-телеметрическое обеспечение;
КА	– космический аппарат;
КИС	– командно-измерительная система;
КУ	– команды управления наземными средствами;

МКПИ	– массив командно-программной информации;
НИП	– наземный измерительный пункт;
НКУ	– наземный комплекс управления.
НС КИС	– наземная станция командно-измерительной системы;
ОС	– операционная система;
ОСПО	– общесистемное программное обеспечение;
ПК	– программная команда;
ПН	– полезная нагрузка;
ПУ	– программа управления;
РБ	– разгонный блок;
РК	– разовая команда;
РКК «Энергия»	– Ракетно-космическая корпорация «Энергия» имени С.П. Королёва;
РН	– ракета-носитель;
САО	– система автоматизированного обмена;
САС	– срок активного существования;
СК	– система коррекции;
СОС	– система ориентации и стабилизации;
СПО	– специальное программное обеспечение;
СЭП	– система электропитания;
ПКПО	– планирование и командно-программное обеспечение;
РАСО	– режим аппаратной солнечной ориентации;
СТР	– система терморегулирования;
ТМИ	– телеметрическая информация;
ТУ	– теневой участок;
УВ	– управляющее воздействие;
ЦУП	– центр управления полетами;
ЯОТР	– язык описания типовых работ.

ПРИЛОЖЕНИЕ А СИНТАКСИС ЯЗЫКА «ДЕЛЬТА»

Синтаксис языка – набор правил, определяющих допустимые цепочки входных символов (текстовая, табличная или графическая нотации), описывается с помощью Расширенных Бэкуса-Наура Форм (РБНФ).

РБНФ разрешает использовать следующие упрощения:

- нетерминалы записываются как отдельные слова;
- терминалы записываются в кавычках;
- необязательные элементы синтаксической конструкции заключаются в квадратные скобки «[» и «]»;
- элементы синтаксической конструкции, повторяющиеся ноль и более раз, заключаются в фигурные скобки «{» и «}»;
- альтернативные варианты разделяются знаком «|»;
- круглые скобки «(» и «)» используются для ограничения альтернативных конструкций;
- сочетание фигурных скобок и косой черты «{/» и «/}» используется для обозначения повторения один и более раз;
- нетерминальные символы изображаются словами, выражающими их интуитивный смысл и написанными на русском языке.

Сценарий_управления = "ПРОГРАМ" {Оператор} "КПРОГРАМ".

Оператор =

Оператор_выдать	
Оператор_печатать	
Оператор_пауза	
Оператор_стоп	
Оператор_декодер	
Оператор_если	
Оператор_выбор	
Оператор_цикл	
Оператор_вызвать	
Оператор_комментарий.	

Оператор_выдать = "ВЫДАТЬ" Тип_команды Номер_команды.
Тип_команды = "РК" | "ПК".
Номер_команды = Целое.

```

Целое =          Цифра {Цифра}.
Цифра =          "0" | "1" | "2" | "3" | "4" | "5" |
                  "6" | "7" | "8" | "9".

Оператор_печать = "ПЕЧАТЬ" [Цвет] ["#ОКНО"] Текст.

Цвет =           "#КРАСНЫЙ"      |
                  "#ЗЕЛЕНый"     |
                  "#СИНИЙ"       |
                  "#ОРАНЖЕВый"   |
                  "#РОЗОВый"     |
                  "#ФИОЛЕТОВый"  |
                  "#ЖЕЛТый".

Текст =          (Буква | Цифра)
                  { Пробел | Буква | Символ | Цифра } .

Пробел =         " ".
Буква =          "А" | "Б" | "В" | ... "Ю" | "Я" |
                  "а" | "б" | "в" | ... "ю" | "я".

Символ =         "!" | "@" | "#" | "$" | "%" |
                  "&" | "*" | "(" | ")" | "|" |
                  "\" | "/" | "<" | ">" | "," |
                  "." | ":" | ";" | "-" | "№" |
                  "' | "?" | "\"" | "{" | "}".

Оператор_пауза = "ПАУЗА" время.
время =        чч:"мм":"сс.

чч =            "00" | "01" | "02" | "03" | "04" |
                  "05" | "06" | "07" | "08" | "09"
                  ... "99".

мм =            "00" | "01" | "02" | "03" | "04" |
                  "05" | "06" | "07" | "08" | "09"
                  ... "59".

сс =            "00" | "01" | "02" | "03" | "04" |
                  "05" | "06" | "07" | "08" | "09"
                  ... "59".

Оператор_стоп = "СТОП".

Оператор_декодер = "ДЕКОДЕР" Номер_декодера.
Номер_декодера = "А" | "В".

Оператор_если = "ЕСЛИ" Список_условий ["ЗА" Время]
                  { Оператор }
                  ["ИНАЧЕ"

```

```

{Оператор}]
"КЕСЛИ".

Список_условий =      Условие {"И" Условие |
                        "ИЛИ" Условие}.

Условие =              ([ "НЕ" ] Идентиф
                        Оператор_сравнения
                        Число |
                        Идентиф "В" Интервал) .

Оператор_сравнения =  "=" | "<" | ">" |
                        "<=" | ">=" | "!=".

Идентиф =              ТМ_параметр |
                        Сохр_ТМ_параметр |
                        Состояние |
                        Ном_КА |
                        Параметр_сеанса.

ТМ_параметр =          Буква{Буква|Цифра}.
Сохр_ТМ_параметр =    "#"ТМ_параметр.

Состояние =           "@доСАО" |
                        "@доЗС" |
                        "@ИСП".

Параметр_сеанса =     "INF1" | "INF2" | "INF3" | "INF4" |
                        "INF5" ... "INF15".

Интервал =             (" ("|"[") Число ";" Число (")"|"]").

Число =                Цифра [{Цифра}] [ "." Цифра].

Оператор_выбор =      "ВЫБОР" Идентиф
                        {Оператор_вариант}
                        ["ОСТАЛЬНОЕ"
                        {Оператор}]
                        "КВЫБОР".

Оператор_вариант =    "ВАРИАНТ" "(" Число ")" | Интервал
                        {Оператор}.

Оператор_цикл =       "ЦИКЛ" "(" Целое ")"
                        {Оператор}
                        "КЦИКЛ".

```

Оператор_вызвать = "ВЫЗВАТЬ" Имя_сценария ["(" Параметр {"," Параметр} ")"].

Имя_сценария = {Буква|Цифра}.

Параметр = Идентиф | Число.

Оператор_комментарий = "К" Текст.

ПРИЛОЖЕНИЕ Б АКТ О ВВОДЕ В ЭКСПЛУАТАЦИЮ

УТВЕРЖДАЮ

Главный конструктор разработки космических комплексов (систем) координатно-метрического назначения, наземных комплексов управления и баллистического обеспечения



С.В.Сторожев

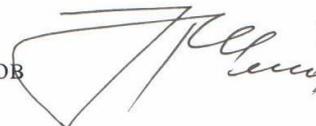
АКТ О ВВОДЕ В ЭКСПЛУАТАЦИЮ

результатов научной работы Космыниной Н.А.

Настоящий акт составлен о том, что результаты научной работы «Языковые и инструментальные средства создания и исполнения сценариев управления космическими аппаратами», аспиранта Космыниной Н.А. входят в состав Специального программного обеспечения планирования и командно-программного обеспечения управления космическими аппаратами (Свидетельство №2014619652), которое применяется в процессе управления большинством КА разработки АО «ИСС», включая, но не ограничиваясь, следующими КА: «Луч-5А», «Луч-5Б», «Луч-5В», «Ямал-300К», «Ямал-401», «Экспресс-АМ5», «Экспресс-АМ6», «Экспресс-АТ1», «Экспресс-АТ2».

Разработанные программные решения позволили упростить процесс подготовки исходных данных управления космическими аппаратами, а также повысить эффективность принятия решений при управлении космическими аппаратами.

Начальник отдела проектирования
автоматизированных систем управления
и разработки наземных комплексов
управления

 К.Б. Шмик
03.2017