

## РАЗРАБОТКА ЯЗЫКА УПРАВЛЕНИЯ КОСМИЧЕСКИМИ АППАРАТАМИ

Космынина Н. А.<sup>а</sup>, инженер

Легалов А. И.<sup>б</sup>, доктор техн. наук, профессор

Барков А. В.<sup>а</sup>, канд. техн. наук, начальник сектора

Лапин А. А.<sup>а</sup>, канд. техн. наук, начальник сектора

<sup>а</sup>АО «Информационные спутниковые системы» им. академика М. Ф. Решетнёва», Железногорск, Красноярский край, РФ

<sup>б</sup>Сибирский федеральный университет, Красноярск, РФ

**Постановка проблемы:** в настоящее время возможности языка управления космическими аппаратами, эксплуатируемого на базе акционерного общества «Информационные спутниковые системы» им. академика М. Ф. Решетнёва», перестали соответствовать требованиям предметной области: язык не поддерживает новых возникающих функциональных задач и слишком сложен для быстрого освоения, что вызвало необходимость разработки нового языка управления. **Результаты:** на основании анализа ограничений применяемого языка управления, требований предметной области и существующих аналогов был спроектирован и разработан проблемно-ориентированный язык управления «Дельта», обладающий простым синтаксисом, с применением терминов предметной области в качестве ключевых слов, поддерживающий как русский, так и английский варианты написания операторов языка. В языке поддерживаются основные функции управления: выдача управляющих воздействий, анализ поступающей телеметрии, паузы, печать сообщений, условные операторы и др. **Практическая значимость:** данная разработка предназначена для управления большинством космических аппаратов производства акционерного общества «Информационные спутниковые системы» как в отечественных, так и в международных проектах.

**Ключевые слова** — предметно-ориентированный язык, проблемно-ориентированный язык, управление космическими аппаратами, процедура управления.

### Введение

В настоящее время возможности, предоставляемые использованием космических систем, применяются в различных областях: связи, навигации, геодезии, телевидении и др. Для поддержания выполнения системой указанных целевых функций необходимо обеспечить решение задач управления космическими аппаратами (КА): удержание на заданной орбите, включение/отключение различных режимов работы, изменение алгоритма функционирования полезной нагрузки и др., — что решается выдачей соответствующих команд с Земли на КА [1, 2]. Логическая совокупность команд, выдаваемая на борт КА для реализации единичной задачи управления, называется процедурой управления. Проблемно-ориентированные языки позволяют описать процедуру управления в виде сценария и выполнять его в ходе сеансов связи с КА в автоматизированном режиме неограниченное количество раз. Примеры таких языков: Python, PLUTO, CSL, STOL, TAO, CSTOL, CCL, JAS, ICL, Btscript, Cecil, CIL, TOPE/tcl, UCL, Elisa, PIL, ЯОТР и др.

### Обзор и сопоставление ряда языков управления КА

Функциональные возможности языков управления КА обоснованы необходимостью решения

задач управления КА и включают, но не ограничиваются, следующим набором:

- выдача команд управления на КА;
- анализ текущих значений телеметрических (ТМ) параметров на соответствие допустимому интервалу значений;
- сообщение оператору;
- паузы между командами;
- вызов другой процедуры управления из данной;
- комментарии к тексту процедуры.

В связи с большим разнообразием существующих языков управления [3] производятся попытки унификации, для чего разработаны следующие документы: стандарт ECSS-E-ST-70-32C: Test and Operations Procedure Language (PLUTO) [4]; спецификация OMG SOLM [5], — содержащие описание возможностей, которыми, по мнению разработчиков, должен удовлетворять любой язык управления для обеспечения совместности различных систем управления.

Рассмотрим некоторые применяемые языки.

Язык Python [6] — переносимый, высокоуровневый, объектно-ориентированный универсальный язык, поддерживающий параллельное выполнение сценариев управления КА. Применяется в специальном программном обеспечении управления (СПО-У) GSEOS, разработанном компанией I&T, США. Помимо этого, язык Python применяется для автоматизации управления КА

с помощью СПО-У «hifly» [7], разработанном компанией GMV, Испания.

Язык PLUTO [8], применяющийся в СПО-У SCOS Mission Control System, разработанном компанией SciSys (Англия, Германия), а также в СПО-У STEPS, разработанном институтом INPE, Бразилия, — высокоуровневый язык программирования, поддерживающий циклы, арифметические операции, планирование выполнения. Процедура управления, созданная с использованием данного языка, должна содержать следующие разделы: 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 постусловий (проверка того, что процедура была выполнена правильно).

Язык SCL применяется для управления КА в СПО-У ASPEN, компания Jet Propulsion Laboratory [9], США, а также в СПО-У SCL [10], компания Interface & Control Systems, Inc, США. На SCL можно создавать как скрипты управления, так и правила управления. Правила предназначены для мониторинга внештатных ситуаций в подсистемах КА, при появлении которых предусматривается автоматический запуск соответствующего скрипта. Скрипты управления предназначены для исполнения в ходе сеанса связи с КА для выполнения какой-либо целевой задачи. Скрипты могут выполняться параллельно, на основе приоритетов, назначаемых при создании скрипта. Кроме функций, общих для всех языков управления, SCL поддерживает циклы и выражения присваивания.

Компанией NASDA, Япония [11], разработан язык управления КА, предназначенный для формирования и исполнения процедур управления в составе СПО-У SMACS. 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.

Компанией Kratos Integral Systems International, США, разработан язык управления КА [12], применяющийся в составе СПО-У Ares. Скрипт представляет собой текстовый файл в виде таблицы с использованием символов табуляции в качестве разделителей со следующими столбцами: номер шага, инструкция, команда, ТМ-параметр, метка, вызов функции. Позволяет вызывать функции из предоставляемой библиотеки функций. Помимо общих функций языка управления, дополнительно поддерживается загрузка файлов с аргументами команд.

Язык описания типовых работ — ЯОТР — разработан на базе акционерного общества «Информационные спутниковые системы» им. академика М. Ф. Решетнёва (АО «ИСС»), Россия.

В 1983 г. на базе Института прикладной математики им. Келдыша совместно с РКК «Энергия» был разработан проблемно-ориентированный табличный язык «Диполь» для проведения наземных испытаний проекта «Энергия-Буран». В 1987 г. язык «Диполь-4» был внедрен в Научно-производственное объединение прикладной механики им. академика М. Ф. Решетнёва 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 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 базе языка «Диполь-5» была разработана очередная версия языка «Диполь-6» [13], которая применяется для комплексных электрических испытаний КА производства АО «ИСС».

Язык описания типовых работ, применяемый для управления КА, — это интерпретируемый скриптовый процедурный язык с текстовой нотацией. Типовая работа (ТР) представляет собой текстовый файл в виде таблицы с использованием символов псевдографики “|” со следующими столбцами: номер строки, тип строки, оператор языка, аргументы. Пример ТР управления КА показан на рис. 1.

Результаты сопоставления перечисленных языков управления КА по общим характеристикам языков программирования (тип языка, используемая парадигма, возможность параллельного выполнения сценариев, используемая нотация) представлены в табл. 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 (разра-

|   |  |    |  |              |  |                |  |    |  |     |  |  |
|---|--|----|--|--------------|--|----------------|--|----|--|-----|--|--|
| О |  | 1  |  |              |  | ПРОГРАМ        |  |    |  |     |  |  |
| О |  | 2  |  |              |  | ВЫЗВАТЬ        |  |    |  | _PK |  |  |
| О |  | 3  |  |              |  | 40             |  | &1 |  |     |  |  |
| О |  | 4  |  |              |  | ЕСЛИТО         |  |    |  |     |  |  |
| О |  | 5  |  |              |  | #ГТС_БЛ        |  |    |  | 0   |  |  |
| О |  | 6  |  |              |  | ПЕЧАТЬ         |  |    |  |     |  |  |
| Ф |  | 7  |  | Счетчик фраз |  | обнулен        |  |    |  |     |  |  |
| О |  | 8  |  |              |  | ИНАЧЕ          |  |    |  |     |  |  |
| О |  | 9  |  |              |  | ПЕЧАТЬ         |  |    |  |     |  |  |
| Ф |  | 10 |  | Счетчик фраз |  | не был изменен |  |    |  |     |  |  |
| О |  | 11 |  |              |  | КЕСЛИТО        |  |    |  |     |  |  |
| О |  | 12 |  |              |  | КПРОГРАМ       |  |    |  |     |  |  |

■ Рис. 1. Пример ТР на ЯОТР

■ Таблица 1. Сопоставление реализации ряда языков управления КА

| Критерий сопоставления                                 | Python                   | Pluto                      | SCL                | Без названия, СПО-У SMACS | Без названия, СПО-У Ares   | ЯОТР   |
|--|--------------------------|----------------------------|--------------------|---------------------------|--|--|
| Страна разработки СПО-У                                | США, Испания             | Англия, Германия, Бразилия | США                | Япония                    | США  | Россия   |
| Тип языка  | Общего назначения        | Специализированный         | Специализированный | Специализированный        | Специализированный   | Специализированный   |
| Используемая парадигма                                 | Объектно-ориентированный | Процедурный                | Процедурный        | Процедурный               | Процедурный  | Процедурный  |
| Поддержка параллельного выполнения скриптов управления | +                        | -                          | +                  | -                         | -  | -  |
| Табличная/текстовая нотация                            | Текстовая                | Текстовая                  | Текстовая          | Табличная                 | Текстовая в виде таблицы с использованием символов псевдографики | Текстовая в виде таблицы с использованием символов псевдографики |

ботанными специально для автоматизации процесса управления КА);

— возможно использование как процедурно-го, так и объектно-ориентированного стилей программирования;

— отдельные языки поддерживают только последовательное выполнение сценариев управления, другие — как последовательное, так и параллельное;

— для написания сценариев может применяться как текстовое, так и табличное представление.

### Требования к создаваемому языку

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

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

2. Не поддерживается английский язык написания ТР, что является несоблюдением одного из требований международных контрактов, заключаемых АО «ИСС».

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 ТР для систем управления группировками КА;

— для ситуации множественного выбора не предусмотрена обработка случая, когда ни один из предложенных вариантов не подошел;

— не поддерживается использование ТМ-параметров в качестве аргументов операций (сравнение, ожидание, выбор из нескольких вариантов и т. д.);

— отсутствует обработка ТМ-параметров из заранее сохраненного архива (например, для восстановления состояния бортового ретрансляционного комплекса в исходное состояние после какого-либо сбоя);

— невозможно использовать дробные числа в качестве параметров для проверки различных условий (например, принадлежности ТМ-параметра некоторому интервалу и т. д.);

— не поддерживается возможность выдачи команды смены декодера средствами языка.

Применение зарубежных аналогов в качестве языка управления является нецелесообразным из-за проблем интеграции с существующими средствами управления КА, отсутствия детального описания языков в открытом доступе и пр. Это привело к необходимости разработки нового проблемно-ориентированного языка управления КА.

Для соответствия набору предъявляемых требований разрабатываемый язык должен обладать следующими характеристиками:

— быть императивным (описывать логику программы в виде последовательности инструкций);

— иметь простой синтаксис (т. е. содержать минимум синтаксических конструкций, что упрощает процесс обучения конечных пользователей, увеличивает скорость разработки сценариев, и вследствие простоты получаемых программ управления уменьшает количество потенциальных ошибок);

— использовать термины, близкие к предметной области, с сохранением преемственности с существующим языком управления, что упрощает восприятие нового языка пользователями;

— содержать как русский, так и английский варианты написания операторов языка для использования в отечественных и международных проектах АО «ИСС».

Язык управления КА должен поддерживать следующие функции:

— выдачу управляющих воздействий, например команд управления КА и массивов командно-программной информации;

— проверку выполнения заданного условия, например попадания текущего значения указанного ТМ-параметра в допустимый интервал;

— проверку многовариантного ветвления, включая случай, когда ни один вариант из предложенных не подошел;

— проверку выполнения нескольких заданных логических условий «И», «ИЛИ», «НЕ»;

— задание паузы на указанный промежуток времени;

— печать текстовых сообщений оператору;

— комментарии, содержащие пояснения к тексту программы;

— вызов других ТР из данной с возможностью передачи некоторых параметров, что обеспечивает повторное использование уже созданных и отлаженных сценариев, например путем вызова на выполнение одного сценария из другого;

— остановку выполнения ТР, что применяется для обработки нештатной ситуации в управлении, прерывания стандартного хода выполнения типовой процедуры в случае, когда необходимо вмешательство оператора для анализа сложившейся ситуации;

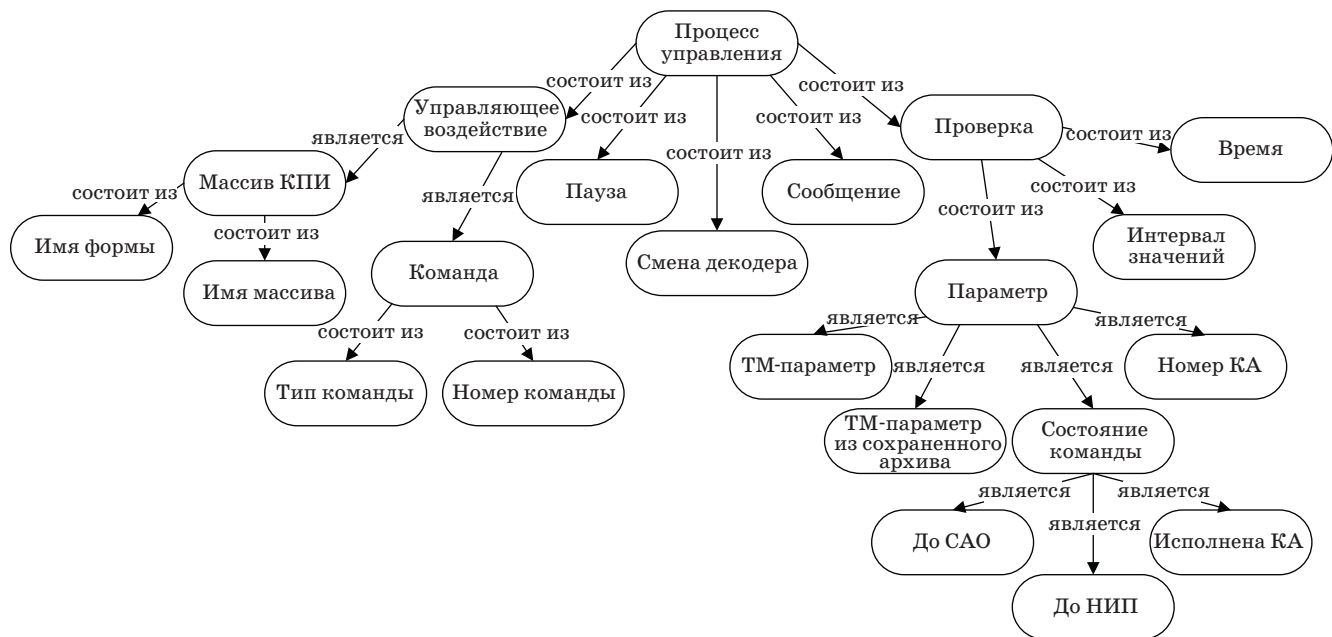
— циклы, необходимые для выполнения повторяющихся действий.

### Разработка языка управления КА

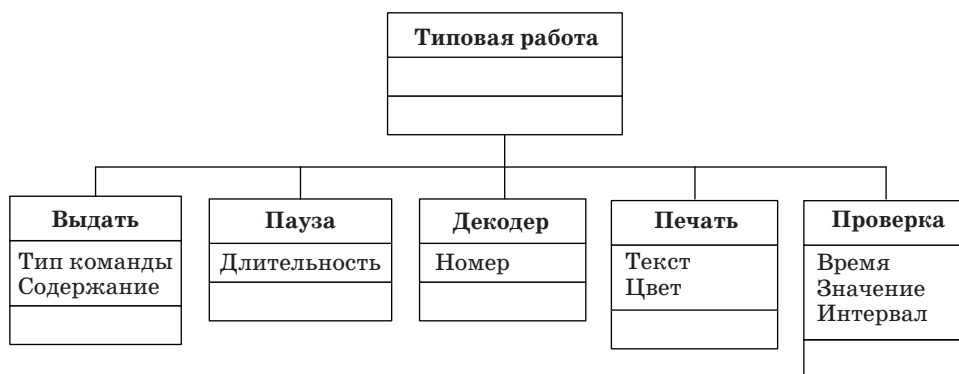
Рассмотренные выше требования были реализованы в языке управления КА «Дельта», который относится к классу предметно-ориентированных языков.

В соответствии с работой [14] последовательность разработки внешнего проблемно-ориентированного языка состоит из трех шагов: определение семантической модели; определение синтаксической модели (абстрактный и конкретный синтаксис); определение правил трансформации (правила, по которым абстрактное представление транслируется в исполнимое). Третий этап подразумевает разработку транслятора языка, описание которого выходит за рамки данной статьи.

Семантическую модель предметной области разрабатываемого языка представим в виде семантической сети (рис. 2).



■ Рис. 2. Семантическая модель предметной области: КПИ — командно-программная информация; САО — система автоматизированного обмена; НИП — наземный измерительный пункт



■ Рис. 3. Метамоделер разрабатываемого языка «Дельта»

*Абстрактный синтаксис* — описание синтаксиса, не зависящее от способа представления данных (от текстового представления команд или пиктограмм, используемых для визуализации объектов). Метамоделер разрабатываемого языка, созданная на основе анализа семантической модели предметной области, представлена с помощью диаграммы классов UML на рис. 3.

Данная метамоделер несколько отличается от существующих универсальных метамоделер языка управления КА [5, 12], что объясняется отличиями в автоматизируемых технологиях управления.

*Конкретный синтаксис* — набор правил, определяющих допустимые цепочки входных символов. Из возможных форм представления (текстовой, табличной, графической) выбрано текстовое представление. Это связано с тем, что автоматизируемая логика управления КА является линейной, т. е. относительно простой для восприятия в виде текста.

### Общие характеристики

Сценарий является текстовым файлом, хранящимся в базе данных и обладающим уникальным именем, по которому его можно вызвать на исполнение в ходе сеанса управления. Также его можно вызывать 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.

Операторы сценария выполняются последовательно, исключая случаи обработки управляющих конструкций (операторов ветвления).

### Переменные

— Имена ТМ-параметров (идентификаторы которых представляют собой произвольную последовательность символов и могут меняться от одной космической системы к другой). Текущие значения ТМ-параметров — числа с плавающей точкой.

— ТМ-параметр из сохраненного архива. При необходимости оператор может сохранить текущие значения ТМ-параметров на определенный момент времени в некоторый архив. Для использования сохраненных значений необходимо указать их в формате «#имя\_тм\_параметра».

— Состояние выданной команды (булева переменная) — на каком этапе прохождения она находится: дошла до программного обеспечения САО, куда выданные команды попадают перед тем, как быть отправленными на наземную станцию («#доСАО»), выдана наземной станцией («#доНИП»), выполнена на КА («#ИСП»).

— Номер КА — применяется для систем из нескольких КА, для управления которых применима одна и та же ТР. Номер КА должен быть указан в формате «#КА».

### Операторы

Перечень операторов языка «Дельта» создан на основе разработанной метамоделер (см. рис. 3). Многие ключевые слова, используемые в языке, совпадают с терминами предметной области.

*Оператор выдачи управляющих воздействий на КА.* В качестве аргументов оператора должны быть использованы тип и номер команды. Допускаются следующие типы команд: разовые команды (РК), программные команды (ПК) и некоторые другие, зависящие от конкретной реализации космической системы. Помимо этого можно выдать массив КПИ, также обладающий уникальным именем и хранящийся в специализированной таблице баз данных. Применяемое ключевое слово для обозначения оператора — «ВЫДАТЬ»:

```
<оператор_выдать> ::= ВЫДАТЬ <тип_команды>
<номер_команды>
```

Пример использования:

```
ВЫДАТЬ РК 40
```

*Оператор выдачи произвольных сообщений на экран.* Ключевое слово для обозначения оператора — «ПЕЧАТЬ». Цвет выводимого сообщения можно указать, включив в сообщение

наименование цвета в формате «#ЦВЕТ», например «#КРАСНЫЙ» или «#ЗЕЛЕНый»:

```
<оператор_печать> ::= ПЕЧАТЬ (<цвет>)<сообщение>
<цвет> ::= #КРАСНЫЙ | #ЗЕЛЕНый | #СИНИЙ |
#ОРАНЖЕВый | #РОЗОВый | #ФИОЛЕТОВый |
#ЖЕЛТый
```

**Пример использования:**

ПЕЧАТЬ Выполнение ТР остановлено #КРАСНЫЙ.

*Оператор остановки выполнения программы на указанный интервал времени.* Ключевое слово — «ПАУЗА»:

```
<оператор_пауза> ::= ПАУЗА <время>
<время> ::= <чч>:<мм>:<сс>
```

**Пример использования:**

ПАУЗА 00:01:00

*Оператор принудительной остановки процесса выполнения ТР.* Ключевое слово — «СТОП». При необходимости выполнение ТР можно продолжить СПО-У:

```
<опер_стоп> ::= СТОП
```

**Пример использования:**

СТОП

*Оператор смены декодера.* Ключевое слово — «ДЕКОДЕР», затем идет номер активируемого декодера, «А» или «В»:

```
<опер_декодер> ::= ДЕКОДЕР <номер_декодера>
```

**Пример использования:**

ДЕКОДЕР А

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

*Условные операторы.* Операции «ЕСЛИ», «ИНАЧЕ», «КЕСЛИ» позволяют выполнять определенную последовательность действий в зависимости от выполнения некоторого условия. В случае если условие выполнилось, исполняется указанная далее последовательность операций. 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>)
<последовательность_операторов>
ИНАЧЕ
<последовательность_операторов>
КЕСЛИ
<время> ::= <чч>:<мм>:<сс>
<список_условий> ::= <условие>(И<список_условий> | ИЛИ <список_условий>
<условие> ::= (НЕ) <параметр> <оператор> <значение> | <параметр> В <интервал>
<оператор> ::= < | > | <= | >=
```

```
<интервал> ::= (<значение>; <значение>) |
[<значение>; <значение>] | [<значение>; <значение>]
| (<значение>; <значение>)
```

**Пример использования:**

ЕСЛИ(ТМ1>8 ЗА 00:01:00)

ПЕЧАТЬ ТМ1 > 8

ИНАЧЕ

СТОП

КЕСЛИ

*Конструкции выбора.* Операторы «ВЫБОР», «ВАРИАНТ», «ОСТАЛЬНОЕ» «КВЫБОР» образуют конструкцию многовариантного ветвления в ТР — 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.

```
<оператор_выбор> ::= ВЫБОР (<параметр>)
<список_вариантов>
КВЫБОР
<список_вариантов> ::=
ВАРИАНТ(<значение>|<интервал>)
<последовательность_операторов>|<список_вариантов>
<интервал> ::= (<значение>; <значение>) |
[<значение>; <значение>] | [<значение>; <значение>]
| (<значение>; <значение>)
```

**Пример использования:**

ВЫБОР (ТМ1)

ВАРИАНТ (., 8)

ПЕЧАТЬ Параметр ТМ1 < 8

ВАРИАНТ 8

ПЕЧАТЬ Параметр ТМ1 = 8

ВАРИАНТ (8, ..)

ПЕЧАТЬ Параметр ТМ1 > 8

КВЫБОР

*Циклы.* Операция обеспечивает повторение выбранных операций по управлению указанное количество раз:

```
<оператор_цикл> ::= ЦИКЛ (<число раз>)
<последовательность_операторов>
КЦИКЛ
```

**Пример использования:**

ЦИКЛ (8)

ВЫДАТЬ РК 40

ПАУЗА 00:01:00

КЦИКЛ

*Конструкции включения.* Оператор «ВЫЗВАТЬ» инициирует выполнение указанной ТР из текущей. При необходимости можно передать в вызываемую ТР значения некоторых параметров. В тексте вызываемой ТР 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:

```
<опер _ вызвать> ::= ВЫЗВАТЬ <имя _ работы>
(<список _ параметров>)
<список _ параметров> ::= <значение> (<список _
параметров>)
```

**Пример использования:**

```
ВЫЗВАТЬ _ РК(40)
```

**Комментарии.** Должна быть предусмотрена возможность включения в текст процедуры произвольных пояснений к особенностям ее написания, не обрабатываемых интерпретатором языка. Для включения комментария должно использоваться ключевое слово «К»:

```
<опер _ комментар> ::= К <текст _ комментария>
```

**Пример использования:**

```
К Дата создания ТР - 2015.04.29
```

Для международных проектов АО «ИСС» поддерживается также вариант на английском языке, сохраняющий все особенности разработанного синтаксиса, за исключением написания ключевых слов. Соответствие русскоязычных и англоязычных слов представлено в табл. 2.

■ **Таблица 2.** Соответствие ключевых слов в русском и английском вариантах разработанного языка управления

| Русский вариант | Английский вариант |
|-----------------|--------------------|
| ВЫДАТЬ          | SEND               |
| ЕСЛИ            | IF                 |
| ЗА              | IN                 |
| КЕСЛИ           | EIF                |
| ВЫБОР           | SELECT             |
| ВАРИАНТ         | CASE               |
| ОСТАЛЬНОЕ       | REST               |
| КВЫБОР          | ESELECT            |
| И               | AND                |
| ИЛИ             | OR                 |
| НЕ              | NOT                |
| ПАУЗА           | PAUSE              |
| ПЕЧАТЬ          | PRINT              |
| К               | K                  |
| ВЫЗВАТЬ         | CALL               |
| ДЕКОДЕР         | DECODER            |
| СТОП            | STOP               |
| ЦИКЛ            | CYCLE              |
| КЦИКЛ           | ECYCLE             |
| #НКА            | #SC                |
| #доСАО          | #toFES             |
| #доЗС           | #toST              |
| #ИСП            | #DONE              |
| #КРАСНЫЙ        | #RED               |
| #ЗЕЛЕНый        | #GREEN             |
| #СИНИЙ          | #BLUE              |
| #ОРАНЖЕВый      | #ORANGE            |
| #РОЗОВый        | #PINK              |
| #ФИОЛЕТОВый     | #MAGENTA           |
| #ЖЕЛТый         | #YELLOW            |

К Пример написания типовой работы

```
ВЫЗВАТЬ РТР _Вкл_1
ВЫЗВАТЬ РТР _АКТ(40,ф 300:1)
ПАУЗА 00:00:10
```

```
ЕСЛИ (АТМ 1 = 1 ЗА 00:00:20)
ПЕЧАТЬ #КРАСНЫЙ Параметр АТМ 1 = 1
СТОП
КЕСЛИ
```

```
ПАУЗА 00:00:10
```

```
ЕСЛИ (АТМ 1 = 1 И АТМ 2 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
```

■ **Рис. 4.** Пример ТР на языке управления «Дельта»

Пример ТР на разработанном языке управления «Дельта» показан на рис. 4.

Стоит отметить, что разработанный язык «Дельта» поддерживает и расширяет функциональные возможности ранее эксплуатируемого ЯОТР для обеспечения переносимости управления в уже существующих системах управления КА.

### Заключение

В ходе выполнения исследования были рассмотрены существующие зарубежные аналоги — языки управления космическими аппаратами, выявлена схожесть выполняемых ими функциональных задач и отсутствие общих тенденций в реализации.

Проанализирован эксплуатируемый на АО «Информационные спутниковые системы» язык управления космическими аппаратами — ЯОТР, сформулированы его недостатки. Выявлены требования к новому языку управления КА: импера-

тивность; простой синтаксис; использование терминов, близких предметной области; наличие как русского, так и английского вариантов написания операторов языка; необходимость в расширении существующих функциональных возможностей.

В соответствии со сформулированными требованиями был разработан новый язык управления КА «Дельта», удовлетворяющий текущим функциональным задачам предметной области и обеспечивающий лаконичность и простоту соз-

даваемых сценариев управления. Язык является более простым в освоении для специалистов по управлению КА, чем предыдущий вариант языка управления, что увеличивает скорость разработки сценариев управления и уменьшает количество потенциальных ошибок.

Данная разработка предназначена для управления большинством КА производства АО «ИСС» как в отечественных, так и в международных проектах.

## Литература

1. Соловьев В. А., Лысенко Л. Н., Любинский В. Е. Управление космическими полетами. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2010. — 432 с.
2. Иванов Д. С. Управление полетом космического аппарата. <http://www.cosmos-journal.ru/articles/936/> (дата обращения: 03.06.2015).
3. Chaudhri G., Carter J., Kizzort B. A Model for a Spacecraft Operations Language. <http://arc.aiaa.org/doi/pdf/10.2514/6.2006-5708> (дата обращения: 03.06.2015)
4. ECSS-E-ST-70-32C Space Engineering Test and Operations Procedure Language. [ftp://ftp.heanet.ie/disk1/sourceforge/o/op/open-hades/Documentation/ECSS-E-ST-70-32C%20\(31July2008\).pdf](ftp://ftp.heanet.ie/disk1/sourceforge/o/op/open-hades/Documentation/ECSS-E-ST-70-32C%20(31July2008).pdf) (дата обращения: 03.06.2015).
5. Satellite Operations Language Metamodel (SOLM). <http://www.omg.org/spec/SOLM/1.0/PDF/> (дата обращения: 03.06.2015).
6. Hauck T. F. Use of the Ground Support Equipment Operating System (GSEOS) Software on the Messenger Mission: a Case Study. <http://descanso.jpl.nasa.gov/RCSGSO/Paper/A0013Paper.pdf> (дата обращения: 03.06.2015).
7. Garcia G., et al. Migration of the Operations of a Legacy Multi-Satellite System to a State-of-the-art Ground System. <http://www.enu.kz/repository/2009/AIAA-2009-6622.pdf> (дата обращения: 03.06.2015).
8. Seymour M. A. The PLUTO Operations Procedure Language and its Use for RADARSAT-2 Mission Operations. <http://arc.aiaa.org/doi/pdf/10.2514/6.2004-538-333> (дата обращения: 03.06.2015).
9. Mims T. L. Use of Spacecraft Command Language for Advanced Command and Control Applications. <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110015644.pdf> (дата обращения: 03.06.2015).
10. Flight Software — SCL for Flight Systems. <http://www.sra.com/scl/flight/> (дата обращения: 03.06.2015).
11. Konishi M., Miyoshi M., Sugawara M. Spacecraft Control System Development. <http://arc.aiaa.org/doi/pdf/10.2514/6.2002-T3-27> (дата обращения: 03.06.2015).
12. Cruickshank D. M., Aparicio A. J., Mistry A. A., Poynton B. R. Using ARES to Improve Efficiency in Satellite Operations. [http://www.powershow.com/view/1c44e-NjEwM/Using\\_ARES\\_to\\_Improve\\_Efficiency\\_in\\_Satellite\\_Operations\\_powerpoint\\_ppt\\_presentation](http://www.powershow.com/view/1c44e-NjEwM/Using_ARES_to_Improve_Efficiency_in_Satellite_Operations_powerpoint_ppt_presentation) (дата обращения: 03.06.2015).
13. Барков А. В. Структура проблемно-ориентированного языка испытаний космических аппаратов // Вестник Сибирского государственного аэрокосмического университета им. академика М. Ф. Решетнёва. 2006. № 5. С. 15–18.
14. Ботов Д. С. Обзор современных средств создания и поддержки предметно-ориентированных языков программирования // Вестник Южно-Уральского государственного университета. 2013. № 1. С. 10–15.

UDC 004.434

doi:10.15217/issn1684-8853.2015.5.82

### Spacecraft Control Language Development

Kosmynina N. A.<sup>a</sup>, Engineer, [kosmynina@iss-reshetnev.ru](mailto:kosmynina@iss-reshetnev.ru)

Legalov A. I.<sup>b</sup>, Dr. Sc., Tech., Professor, [legalov@mail.ru](mailto:legalov@mail.ru)

Barkov A. V.<sup>a</sup>, PhD, Tech., Head of Section, [barkov@iss-reshetnev.ru](mailto:barkov@iss-reshetnev.ru)

Lapin A. A.<sup>a</sup>, PhD, Tech., Head of Section, [lapin@iss-reshetnev.ru](mailto:lapin@iss-reshetnev.ru)

<sup>a</sup>Joint-stock Company «Academician M. F. Reshetnev «Information Satellite Systems», 52, Lenin St., 662972, Zheleznogorsk, Krasnoyarsk region, Russian Federation

<sup>b</sup>Siberian Federal University, 79, Svobodny Pr., 660041, Krasnoyarsk, Russian Federation

**Purpose:** Nowadays, the capabilities of the control language used in “JSC Academician M.F. Reshetnev Information Satellite Systems” do not meet the requirements of the object domain: the language does not support new functional tasks, and it is too sophisticated



to learn it quickly. A new control language needs to be developed. **Results:** We have analyzed the limitations of the current control language, the object domain requirements and the existing analogs. On the base of this analysis, a new control language named “Delta” was designed and developed. It has simple syntax, using the object domain terms as key words and supporting both English and Russian syntax in spelling its operators. “Delta” supports basic control functions: sending commands, incoming telemetry analysis, pauses, messages, conditional operators, etc. **Practical relevance:** “Delta” is intended to be used to control the majority of “JSC Academician M. F. Reshetnev Information Satellite Systems” satellites, in both domestic and foreign projects.

**Keywords** — Problem-Oriented Language, Domain Specific Language, Satellite Control, Control Script.

## References

1. Solov'ev V. A., Lysenko L. N., Liubinskii V. E. *Upravlenie kosmicheskimi poletami* [Satellites Flight Control]. Moscow, MGTU im. N. E. Bauman Publ., 2010. 432 p. (In Russian).
2. Ivanov D. S. *Upravlenie poletom kosmicheskogo apparata* [Spacecraft Flight Control]. Available at: <http://www.cosmos-journal.ru/articles/936/> (accessed 03 June 2015).
3. Chaudhri G., Carter J., Kizzort B. *A Model for a Spacecraft Operations Language*. Available at: <http://arc.aiaa.org/doi/pdf/10.2514/6.2006-5708> (accessed 03 June 2015).
4. *ECSS-E-ST-70-32C Space Engineering Test and Operations Procedure Language*. Available at: [ftp://ftp.heanet.ie/disk1/sourceforge/o/op/open-hades/Documentation/ECSS-E-ST-70-32C%20\(31July2008\).pdf](ftp://ftp.heanet.ie/disk1/sourceforge/o/op/open-hades/Documentation/ECSS-E-ST-70-32C%20(31July2008).pdf) (accessed 03 June 2015).
5. *Satellite Operations Language Metamodel (SOLM)*. Available at: <http://www.omg.org/spec/SOLM/1.0/PDF/> (accessed 03 June 2015).
6. Hauck T. F. *Use of the Ground Support equipment Operating System (GSEOS) Software on the Messenger Mission: a Case Study*. Available at: <http://descanso.jpl.nasa.gov/RCSGSO/Paper/A0013Paper.pdf> (accessed 03 June 2015).
7. Garcia G., Palsson M., Morel T., Beech T., Honold P., Fuente O., Cervino A., Cunha C., Florido L., Hilden V., Silva R. *Migration of the Operations of a Legacy Multi-Satellite System to a State-of-the-art Ground System*. <http://www.enu.kz/repository/2009/AIAA-2009-6622.pdf> (accessed 03 June 2015).
8. Seymour A. M. *The PLUTO Operations Procedure Language and its Use for RADARSAT-2 Mission Operations*. Available at: <http://arc.aiaa.org/doi/pdf/10.2514/6.2004-538-333> (accessed 03 June 2015).
9. Mims T. L. *Use of Spacecraft Command Language for Advanced Command and Control Applications*. Available at: <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110015644.pdf> (accessed 03 June 2015).
10. *Flight Software – SCL for Flight Systems*. Available at: <http://www.sra.com/scl/flight/> (accessed 03 June 2015).
11. Konishi M., Miyoshi M., Sugawara M. *Spacecraft Control System Development*. <http://arc.aiaa.org/doi/pdf/10.2514/6.2002-T3-27> (accessed 03 June 2015).
12. Cruickshank D. M., Aparicio A. J., Mistry A. A., Poynton B. R. *Using ARES to Improve Efficiency in Satellite Operations*. [http://www.powershow.com/view/1c44e-NjEwM/Using\\_ARES\\_to\\_Improve\\_Efficiency\\_in\\_Satellite\\_Operations\\_powerpoint\\_ppt\\_presentation](http://www.powershow.com/view/1c44e-NjEwM/Using_ARES_to_Improve_Efficiency_in_Satellite_Operations_powerpoint_ppt_presentation) (accessed 03 June 2015).
13. Barkov A. V. Structure of the Problem-oriented Language of Spacecrafts Tests. *Vestnik Sibirskogo gosudarstvennogo aerokosmicheskogo universiteta im. akademika M. F. Reshetneva*, 2006, no. 5, pp. 15–18 (In Russian).
14. Botov D. S. Review of Modern Development and Support Tools for Domain-specific Programming Language. *Vestnik Iuzhno-Ural'skogo gosudarstvennogo universiteta*, 2013, no. 1, pp. 10–15 (In Russian).

## УВАЖАЕМЫЕ АВТОРЫ!

Научные базы данных, включая SCOPUS и Web of Science, обрабатывают данные автоматически. С одной стороны, это ускоряет процесс обработки данных, с другой — различия в транслитерации ФИО, неточные данные о месте работы, области научного знания и т. д. приводят к тому, что в базах оказывается несколько авторских страниц для одного и того же человека. В результате для всех по отдельности считаются индексы цитирования, снижая рейтинг ученого.

Для идентификации авторов в сетях Thomson Reuters проводит регистрацию с присвоением уникального индекса (ID) для каждого из авторов научных публикаций.

Процедура получения ID бесплатна и очень проста: входите на страницу <http://www.researcherid.com>, слева под надписью «New to ResearcherID?» нажимаете на синюю кнопку «Join Now It's Free» и заполняете короткую анкету. По указанному электронному адресу получаете сообщение с предложением по ссылке заполнить полную регистрационную форму на ORCID. Получаете ID.