

УДК 004.434

## **Языки управления космическими аппаратами**

**Космынина Н. А.**

*Предприятие «Информационные спутниковые системы»*

*имени академика М.Ф. Решетнева», ул. Ленина 52, Железногорск, Красноярский край 662972, Россия*

*e-mail: n.kosmynina@list.ru*

### **Аннотация**

Рассматривается способ автоматизации управления космическими аппаратами с помощью проблемно-ориентированных скриптовых языков, функции таких языков, стандарты ECSS-E-ST-70-32C: Test and Operations Procedure Language и OMG SOLM на космические языки. Приводится краткое описание языков Python, PLUTO, SCL, без названия (ПО «SMACS»), без названия (ПО управления «Ares»), ЯОТР, перечислены другие языки управления.

**Ключевые слова:** управление космическим аппаратом, языки управления космическим аппаратом.

### **Введение**

Современные космические аппараты (КА) являются сложными системами, выполняющими разнообразные целевые задачи: связи, навигации, геодезии, и др.

Управление КА представляет собой процесс выдачи на борт КА управляющих воздействий, обеспечивающих эксплуатацию КА на орбите и решение им указанных

целевых задач. Контроль исполнения управляющих воздействий осуществляется по ТМ информации. Логическая совокупность управляющих воздействий, выдаваемая на борт КА для реализации единичной задачи управления, называется процедурой управления.

Для автоматизации процесса управления широко применяются проблемно-ориентированные языки, что позволяет описать процедуру управления в виде скрипта, и выполнять его в ходе сеанса связи с КА неограниченное количество раз. Скрипты хранятся в некотором внешнем хранилище (базе данных или файловой системе) и выполняются с помощью программного обеспечения управления КА, которое в автоматизированном режиме выполняет действия, записанные в скрипте.

Применение такой технологии ведет к снижению суммарного времени, затрачиваемого на поиск и применение элементарных управляющих воздействий, и повышению надежности часто повторяемых «ручных» операций управления полетом.

Языки управления КА получили широкое распространение в зарубежных системах управления КА, и практически не применяются в отечественных, что требует обобщения и анализа опыта применения данной технологии.

В работе приводится описание основных возможностей языка управления КА, существующих стандартов на языки управления, а также обзор некоторых из существующих языков управления КА.

## **1 Возможности проблемно-ориентированных языков управления КА**

Весь процесс управления КА можно разделить на следующие составляющие:

- выдачу команд управления КА предусмотренных типов;
- проверку выполнения некоторых условий (например, принял ли телеметрический параметр заданное значение, или находится ли он в требуемом интервале);
- ожидание заранее заданного времени, например, для установления выбранного режима на КА.

Все вышеперечисленные требования должны реализовываться в любом языке управления КА, который может содержать и дополнительные возможности, предназначенные для удобства выполнения процедур (например, печать сообщений оператору) или для большего соответствия применяемой технологии управления.

## **2 Стандарты, применяемые к языкам управления КА**

В связи с большим разнообразием существующих языков управления производятся попытки стандартизировать выполняемые ими функции, для чего были разработаны следующие документы: стандарт ECSS-E-ST-70-32C: Test and Operations Procedure Language (PLUTO); спецификация OMG SOLM.

1. Стандарт ECSS-E-ST-70-32C: Test and Operations Procedure Language (PLUTO)[2]. Стандарт Европейской кооперации по стандартизации в области космической техники (European Cooperation for Space Standardization) на язык описания процедур испытаний и управления КА. Стандарт определяет ряд требований, которым должен удовлетворять любой язык, используемый для разработки процедур наземных испытаний КА и управления в ходе эксплуатации:

- 1) простое условное ветвление (например. if ... then ... else ...);

2) множественное условное ветвление, где выбранная на исполнение ветка соответствует значению некоторого параметра;

3) повтор выполнения выражения (или группы выражений) определенное количество раз; выполнение, пока некоторое условие все еще является истинным; выполнение до момента, пока некоторое условие не станет истинным;

4) ожидание достижения некоторого абсолютного времени;

5) ожидание истечения некоторого определенного интервала времени;

6) ожидание выполнения некоторого условия;

7) ожидание наступления некоторого события;

8) возможность использовать локальные переменные строгих типов, которые должны быть инициализированы, а их значение должно быть изменяемым в ходе выполнения процедуры;

9) должны поддерживаться некоторые математические, временные и строковые функции;

10) должна быть предусмотрена возможность включать комментарии в текст процедуры;

11) должна быть реализована возможность выдавать сообщение с подтверждением пользователя, а также сообщение, выводимое в лог-файл программы;

12) возможность запрашивать ввод некоторых входных данных у пользователя;

13) возможность инициировать локальные события внутри процедуры (например, запускать обработку внештатной ситуации);

14) возможность определять локальные события, как часть описания процедуры или шага;

15) должна быть реализована поддержка использования единиц измерения;

16) должно быть реализовано присваивание единиц измерения константам;

17) должна поддерживаться возможность смешивать совместимые единицы измерения в одном выражении;

18) должно поддерживаться автоматическое преобразование между различными, но совместимыми единицами измерения;

19) должна поддерживаться возможность конструировать выражения, которые оперируют константами, параметрами космической системы, изменяемыми аргументами (чьё значение поддерживается во время выполнения) и локальными переменными.

Также стандарт ECSS-E-ST-70-32C содержит описание языка, соответствующего всем вышеприведенным требованиям (PLUTO).

2. OMG SOLM: Spacecraft Operations Language Metamodel[3]. Спецификация Object Management Group на проблемно-ориентированные языки управления КА. Описывает метамодель (модель для создания других моделей) языка управления, представляющую собой иерархию объектов-составляющих процедуры управления КА, с краткой характеристикой каждого.

В составе метамодели в качестве поддерживаемых действий предусматриваются следующие (рисунок 1):

- вызов другой процедуры («Invoke»);

- ожидание («Wait»);

- запись параметра («ParameterWrite»);
- чтение параметра («ParameterRead»);
- сравнение («Verify»);
- запрос параметра у оператора («Query»).

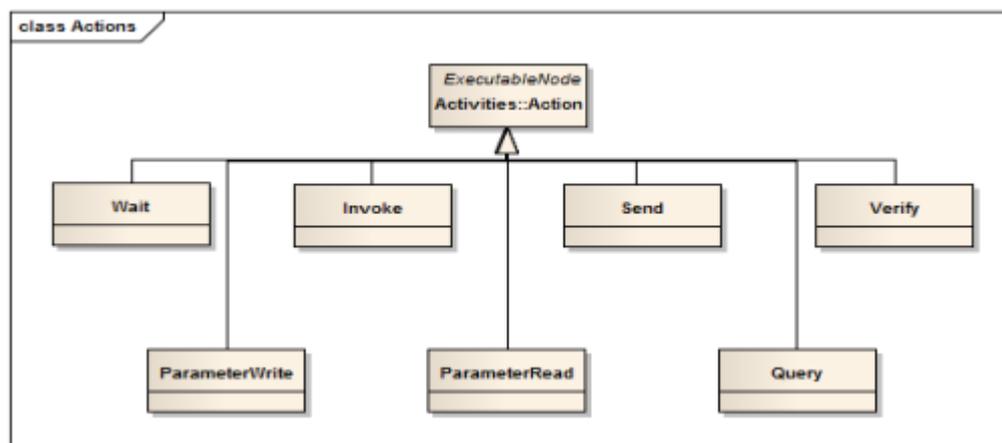


Рисунок 1 – Действия-составляющие процедуры управления КА согласно спецификации OMG SOLM

В составе спецификации производится сопоставление языков SpacePython и CCL на соответствие вышеописанной модели.

### 3 Существующие проблемно-ориентированные языки управления КА

В качестве языков управления могут использоваться как языки программирования общего назначения (Python и др.), применяемые для решения более широкого круга задач, и так и специализированные языки, разработанные специально для задач управления КА (PLUTO, CSL, ЯОТР, и др.).

Язык Python[4] – переносимый, высокоуровневый, объектно-ориентированный язык управления КА, применяемый в СПО-У «GSEOS», разработанным компанией «I&T», США. Помимо этого, язык Python применяется для автоматизации

управления КА с помощью СПО-У «hifly»[5], разработанным компанией «GMV», Испания.

Язык PLUTO[6] - высокоуровневый язык программирования, поддерживающий циклы, арифметические операции, планирование выполнения. Процедура управления, созданная с использованием данного языка, должна быть разделена на следующие разделы:

- 1) раздел объявлений;
- 2) раздел условий старта выполнения процедуры (по наступлению конкретного времени или выполнению ряда условий);
- 3) тело процедуры;
- 4) раздел проверок оборудования (если в ходе выполнения данного раздела произошел сбой, то выполнение процедуры останавливается);
- 5) раздел пост-условий (проверка того, что процедура была выполнена правильно).

Пример процедуры на языке PLUTO приведен на рисунке 2.

Язык PLUTO применяется в СПО-У «SCOS Mission Control System», разработанным компанией «SciSys» (Англия, Германия), а также в СПО-У «STEPS», разработанным институтом «INPE», Бразилия.

```

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 2 - Пример процедуры на языке PLUTO

Язык SCL - ПО «ASPEN», компания «Jet Propulsion Laboratory»[7], США, а также ПО «SCL»[8], 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 могут выполняться параллельно, на основе приоритетов, назначаемых при создании скрипта. Кроме функций, общих для всех языков управления, SCL поддерживает циклы и выражения присваивания. Пример процедуры управления и правила управления на языке SCL представлены на рисунке 3.

<pre> script UpdateTwo  spawn BFSDDataDispMajMode with SGPCAREA2, V98U2408C1 now spawn BFSDDataDispMajMode with SGPCAREA2, V98J2299C1 now spawn BFSDDataDispMajMode with SGPCAREA2, V98J2258C1 now  msg "Ending time for Update for UpdateTwo is " &amp; ticks  end UpdateTwo </pre>	<pre> rule Detect_Second_Change subsystem none category pwr priority 21 activation yes continuous yes  if change (SGPCFIDA2) then      msg "The beginning time is " &amp; ticks      spawn UpdateTwo now  end if  end Detect_Second_Change </pre>
--	---

Рисунок 3 – Пример процедуры (слева) и правила (справа) на языке SCL

Язык без названия[9] - предназначен для составления и исполнения процедур управления в составе СПО-У «SMACS», разработанным компанией «NASDA», Япония. Пример процедуры представлен на рисунке 4. 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.

Язык без названия[10] – применяется в составе СПО-У «Ares», разработанным компанией «Kratos Integral Systems International», США. Скрипт 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.

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 4 – Пример процедуры, поддерживаемой ПО «SMACS», Япония

Язык ЯОТР[11] - СПО «ПКПО», ОАО «ИСС», Россия. Скрипт представляет собой текстовый файл в виде таблицы с использованием символов псевдографики “|”, со следующими столбцами: тип строки, номер строки, оператор языка, аргументы. Поддерживается как русскоязычный, так и англоязычный синтаксис.

Пример процедуры представлен на рисунке 5.

```

O|1| |ПРОГРАМ| | | | |
O|2| |ВЫЗВАТЬ| | | |_PK| PK| |
|3| |40| |&1| | | | |
O|4| |ЕСЛИТО| | | | | | |
|5| |#ГШТАТ| | | |0| | | |
O|6| |ПИСАТЬ| |НОМ_ФРАЗЫ| =| |0| | | |
O|7| |ПЕЧАТЬ| |#ГПП| | | |4| | |
Ф|8| |Счетчик фраз КПИ сброшен| | | | | | |
O|9| |ИНАЧЕ| | | | | | | |
O|10| |ПЕЧАТЬ| | | | |2| | | |
Ф|11| |Счетчик фраз не изменен| | | | | | | |
O|12| |КЕСЛИТО| | | | | | | |
O|13| |КПРОГРАМ| | | | | | | |

```

Рисунок 5 – Пример процедуры на языке ЯОТР

Помимо вышеперечисленных, в различных системах управления КА применяются следующие скриптовые языки[12]: STOL, TAO, CSTOL, CCL, JAS, ICL, Btscript, Cecil, CIL, TOPE/tcl, UCL, Elisa, PIL, Excel, и др.

Большое разнообразие существующих языков управления вызвало проблему переносимости, когда уже разработанный и отлаженный набор процедур нельзя использовать для управления КА при передаче управления из одного центра управления в другой без установки соответствующего программного обеспечения, и необходимости изучать как ПО управления, так и новый язык. В качестве способов решения данной проблемы применяются следующие:

а) трансляторы с одного языка управления на другой встраиваются в уже созданные программные системы управления[5];

б) разработано ПО, позволяющее на основе анализа выбранной процедуры, генерировать процедуру на любом другом известном системе скриптовом языке управления КА[13].

### **Заключение**

Использование скриптовых проблемно-ориентированных языков – фактически сложившаяся мировая практика управления КА, применение которой обеспечивает надежность управления, одновременно сохраняя его гибкость.

В настоящее время используется большое число различных языков управления, которые значительно отличаются друг от друга, как по синтаксису, так и по реализуемому функционалу. Языки управления могут быть как общего назначения, так и специализированными; как объектно-ориентированными, так и процедурными; скрипты управления могут выполняться как последовательно, так и

параллельно, на основе приоритетов; запускаться как по команде оператора, так и автоматически, по наступлению заданного времени и условий; могут поддерживаться или не поддерживаться команды управления наземным оборудованием и его проверки.

Однако, во всех языках управления есть и общие черты, связанные с необходимостью выдачи команд на КА, обеспечением временных задержек, проверками условий различного типа.

Для обеспечения совместимости и переносимости программ управления для КА производства различных стран и организаций разработаны следующие стандарты: спецификация OMG SOLM, и стандарт ECSS-E-ST-70-32C, которым рекомендуется следовать при разработке нового языка управления КА; разрабатывается программное обеспечение, позволяющее транслировать тексты процедур управления с одного языка управления КА на другой.

Также можно отметить большой опыт применения языков управления КА в зарубежных программных системах управления, и ограниченное их распространение в отечественных.

### **Библиографический список**

1. Соловьёв В.А., Лысенко Л.Н., Любинский В.Е. Управление космическими полётами. - М.: МГТУ им. Н.Э. Баумана, 2009. 902 с.
2. Стандарт Европейского космического агентства ECSS-E-ST-70-32C Space engineering Test and operations procedure language Режим доступа:

[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 23.09.2014)

3. Satellite Operations Language Metamodel (SOLM). Режим доступа: <http://www.omg.org/spec/SOLM/1.0/PDF/> (дата обращения 23.09.2014)
4. Use of the ground support equipment operating system (GSEOS) software on the messenger mission: a case study. Режим доступа: <http://www.satellite-evolution.com/digital/Satellite-Evolution-Asia-Sept-Oct-2009/PDFs/Command.pdf> (дата обращения 23.09.2014)
5. 21st century operational procedure automation with Python and autofly. Lessons learned from migrating legacy systems and developing new ones. Режим доступа: <http://download.improve2.org/u/use-of-python-as-a-satellite-operations-and-testing-automation-book-w728/> (дата обращения 23.09.2014)
6. Mark A. Seymour / The PLUTO operations procedure language and its use for RADARSAT-2 mission operations. Режим доступа: <http://arc.aiaa.org/doi/pdf/10.2514/6.2004-538-333> (дата обращения 23.09.2014)
7. Use of Spacecraft Command Language for Advanced Command and Control Applications. Режим доступа: <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110015644.pdf> (дата обращения 23.09.2014)
8. Flight Software - SCL for Flight Systems. Режим доступа: <http://www.sra.com/scl/flight/> (дата обращения 23.09.2014)
9. Spacecraft control system Development. Режим доступа: <http://Arc.Aiaa.Org/Doi/Pdf/10.2514/6.2002-T3-27> (дата обращения 23.09.2014)

10. Using ARES to Improve Efficiency in Satellite Operations - PowerPoint PPT Presentation. Режим доступа: [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 23.09.2014)
11. AMOS5-PR-75000-ISS-00239 Спутник «AMOS-5». Руководство по эксплуатации ЦУП. Часть 2. Планирование и командно-программное обеспечение, Железногорск, ОАО «ИСС», 2011. – 212 с.
12. A Model for a Spacecraft Operations Language. Режим доступа: <http://arc.aiaa.org/doi/pdf/10.2514/6.2006-5708> (дата обращения 23.09.2014)
13. European Technology Harmonisation on Command Procedures GSAW 2010 Simon Reid, Rhea System S.A. Режим доступа: <http://csse.usc.edu/GSAW/gsaw2010/s2/reid.pdf> (дата обращения 23.09.2014)