

А. В. Барков

СТРУКТУРА ПРОБЛЕМНО-ОРИЕНТИРОВАННОГО ЯЗЫКА ИСПЫТАНИЙ КОСМИЧЕСКИХ АППАРАТОВ

Рассмотрены требования к проблемно-ориентированному языку испытаний космических аппаратов. Разработана структура языка, соответствующая предъявляемым требованиям. Описаны возможности этого языка.

В настоящее время для проведения комплексных электрических испытаний космических аппаратов используется автоматизированный испытательный комплекс. Целью этих испытаний является контроль функционирования как изделия в целом, так и его отдельных систем. Идеология их проведения заключается в выдаче управляющих воздействий на объект контроля и анализе параметров функционирования характеризующих состояние и поведение отдельных систем и космического аппарата в целом.

Состояние космического аппарата может характеризоваться тысячами параметров. Управление изделием также включает тысячи команд. Динамика переходных процессов различных систем космического аппарата колеблется от долей секунд до суток, а в некоторых случаях скорость принятия решений по управлению изделием должна быть мгновенной. Таким образом, проблема автоматизации задач контроля и управления при проведении комплексных электрических испытаний является весьма актуальной.

Для автоматизированного проведения испытаний проектанты систем и компонентов космического аппарата разрабатывают алгоритмы (циклограммы) работы на специальном прикладном языке Диполь-5К [1] с помощью редактора циклограмм. В ходе проведения испытаний эти алгоритмы исполняются интерпретатором языка.

Однако все возрастающая сложность систем и компонентов новых изделий, логики их функционирования предъявляют новые требования к средствам автоматизации испытаний. Поэтому в ходе модернизации существующего аппаратно-программного комплекса одной из задач являлась разработка нового проблемно-ориентированного языка испытаний, получившего название Диполь-6.

Для решения этой задачи необходимо определить требования к языку испытаний и разработать его структуру, а также множество базовых понятий этого языка.

Требования к языку. Формирование языка испытаний основано на анализе проблемной области и принципах структурного программирования. С учетом этого выделим требованиями к языку.

Во-первых, это необходимость проблемной ориентации языка как в его синтаксисе, так и в структуре. Очень важным является требование максимальной простоты разработки и анализа циклограмм, так как их созданием занимаются не программисты, а специалисты по конкретным системам, компонентам и блокам космического аппарата. Иными словами, необходимо использо-

вать простую русскоязычную техническую лексику, близкую к терминологии, принятой в данной области. Обеспечение проблемной ориентации языка также служат формализованные языковые средства контроля и управления КА и его системами. Следует использовать конструкции, которые учитывают специфику объекта контроля.

Во-вторых, в составе языка необходимо ввести операции и конструкции, характерные для структурных языков высокого уровня. Как известно, любую структурную программу можно написать, используя только три структуры: следования, ветвления и повторения. Обычно структура следования считается встроенной в язык, и это означает, что все операции языка исполняются последовательно, одна за другой. Имея в своем составе конструкции ветвления и цикла, прикладной язык позволит испытателям разрабатывать сложные алгоритмы испытаний. Это также повышает степень автоматизации работы.

В-третьих, существует еще одна особенность разрабатываемого прикладного языка. Анализ проблемной области и предыдущих разработок показывает, что необходимо разработать по единой технологии универсальный язык испытаний, позволяющий проводить работы на различных аппаратно-программных комплексах. Фактически это требование говорит о необходимости разработки универсального языка определенного уровня абстракции. Кроме того, это требование подразумевает наличие определенного механизма для изменения функциональности языка в случае необходимости.

В-четвертых, очень важным требованием является преемственность с используемым в настоящее время прикладным языком, за счет чего достигается максимальное использование существующего задела и во многом упрощается восприятие нового языка пользователями. Это означает, что идентификаторы и назначение основных операций нового языка должны остаться такими же, как и существующие в уже используемом языке.

Таким образом, необходимо разработать язык испытаний, отвечающий представленным выше требованиям.

Синтаксис языка. В теории языков известны следующие определения, относящиеся к математическому описанию синтаксиса языков [2].

Алфавит языка – это конечное множество символов.

Предложение (строка, слово, лексема) есть любая цепочка конечной длины, составленная из

символов алфавита. Таким образом, язык есть любое множество предложений над некоторым алфавитом.

Алфавит языка испытаний будет содержать прописные и строчные буквы русского и латинского алфавитов, цифры, специальные символы, используемые в качестве ограничителей и разделителей, а также ключевые слова – имена операций.

Далее в классической теории программирования [3] для построения синтаксического и семантического трансляторов определяется грамматика языка. Однако метод построения системы «редактор – интерпретатор циклограмм», метод построения структурного редактора и метод построения интерпретатора циклограмм позволяют исключить задачу анализа синтаксиса и, соответственно, проблемы синтеза и разработки анализатора, рассмотрение которых выходит за рамки данной статьи. Поэтому мы можем сразу перейти к разработке структуры языка [4].

Структура языка. Начнем с разработки иерархии конструкций языка (рис. 1).

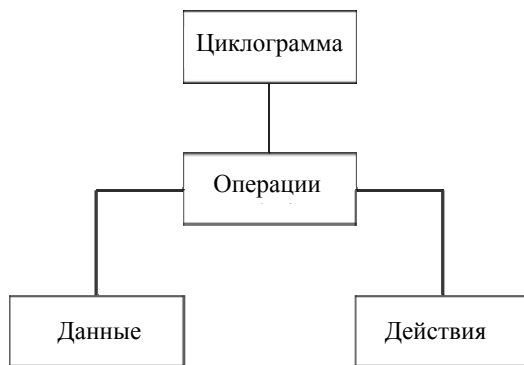


Рис. 1. Иерархия конструкций языка

Верхний уровень образует сама циклограмма, которая является основной единицей выполнения. Ниже следуют операции языка – единицы интерпретации. Операции строятся из данных и действий. Под действиями мы понимаем действия как по обработке данных, так и действия по организации выполнения. Таким образом, операции и циклограммы являются основой построения программ на языке испытаний.

Для записи конструкций, лежащих на нижнем уровне иерархии (см. рис. 1) будем использовать цепочки литер из алфавита языка, называемые лексемами. Выделим следующие группы лексем:

- *идентификаторы* – лексемы, состоящие из букв и цифр. Введено ограничение на длину идентификатора управляющих воздействий и телеметрических параметров, связанных с объектом контроля, – 16 символов. Запрещено использовать ключевые слова и специальные символы #, &, ^;
- *ключевые слова*. В данном случае это лексемы из множества идентификаторов операций языка испытаний.

В качестве базовых будем использовать следующие типы данных:

- *целый тип* – некоторый отрезок всех целых чисел;

– *вещественный тип* – конечное подмножество внутри определенного диапазона вещественных чисел;

– *литерный тип* – конечное множество символов.

Заметим, что на основе базовых типов данных возможно создание производных типов, соответствующих проблемной области. Например, тип *сигнальный параметр функционирования* основан на целом типе и использует отрезок [0, 1].

Введем понятие *окружение языка*. В качестве окружения языка испытаний будет выступать система дежурного контроля. Эта система используется для контроля состояния объекта контроля, его подсистем и компонентов по ключевым параметрам. Дежурный контроль проверяет нахождение значений требуемых параметров в указанных допусках и в случае выхода значения какого-либо параметра из допуска выполняет предусмотренную реакцию. В качестве реакции может быть запуск аварийной циклограммы, сообщение оператору, останов или завершение выполнения либо другое действие.

Для описания сложных семантических конструкций языка воспользуемся *понятиями*. Под ними будем понимать элементы языка, каждому вхождению которых в программу правила языка однозначно сопоставляют их синтаксические значения. При этом одному и тому же понятию может соответствовать целое множество значений вхождений, и для любых двух вхождений понятий их значения не имеют общих цепочек лексической свертки или одно значение входит полностью в другое [4].

В качестве понятий выделим следующие:

- операция языка испытаний;
- телеметрический параметр объекта контроля;
- команда управления объектом контроля;
- переменные;
- допуск (на параметр);
- реакция (на параметр);
- алгоритм дежурного контроля

Правила языка задают для каждого понятия набор характеристик, называемых *атрибутами*. Атрибуты отражают общность и различие семантических объектов. Для каждого понятия были разработаны присущие также им атрибуты.

Разработаем модульную структуру области действия. Она характеризуется тем, что четко разграничивает области действия по модулям построения структуры испытаний. Эти ограничения вместе с программными единицами образуют иерархическую структуру (рис. 2). Программными единицами будут являться циклограмма, заявка, процесс управления.

Процесс управления – отдельный логический процесс, организующий интерпретацию циклограмм в соответствии с их приоритетом.

Заявка – отдельная циклограмма являющаяся головной для запущенных из нее циклограмм, или стек циклограмм времени выполнения.

Циклограмма – отдельная циклограмма, основная единица выполнения.

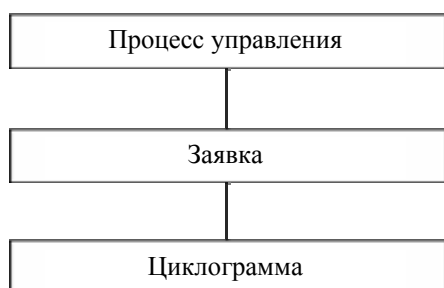


Рис. 2. Иерархическая структура программных единиц

В языке четко выделены несколько областей действия: глобальная, процессная и локальная. Эти области применяются для разграничения действий переменных (рис. 3).



Рис. 3. Области действия переменных

Глобальная область соответствует всему контексту выполнения всех циклограмм, т. е. объекты, определенные в глобальной области, доступны из всех циклограмм. *Процессная область*, соответствует контексту выполнения отдельного процесса управления, т. е. объекты, определенные в процессной области доступны из всех циклограмм, выполняемых в этом процессе управления. И, наконец, *локальная область* соответствует контексту выполнения отдельной циклограммы.

Для поддержания иерархической макроструктуры языка разработан механизм *подпрограмм*. Этот механизм является многоуровневым. Он имеет отличие в организации взаимодействия на следующих уровнях: между процессами управления; между процессом управления и заявками; между заявками и их дочерними циклограммами.

Разработаем верхний уровень взаимодействия – уровень взаимодействия между процессами управления (рис. 4). Работа процессов управления организована псевдопараллельно, т. е. после завершения отдельного шага выполнения в первом процессе управления передается второму, чтобы в нем исполнился следующий шаг. После завершения

этого шага управление передается в первый процесс для исполнения следующего шага и т. д. Таким образом достигается псевдопараллельная работа процессов управления.

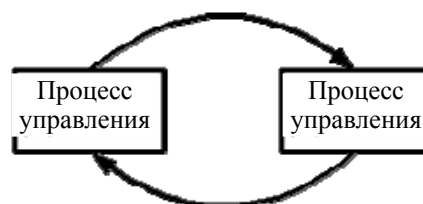


Рис. 4. Взаимодействие процессов управления

Разработаем средний уровень взаимодействия – между процессом управления и заявками. В процессе управления для организации выполнения циклограмм используется система приоритетов. Приоритет задается каждой заявке при начале ее выполнения, и затем он может быть изменен программно или испытателем. При возникновении заявки с большим приоритетом выполнение циклограммы с меньшим приоритетом приостанавливается. Заявки с меньшим приоритетом образуют очередь выполнения (рис. 5).



Рис. 5. Взаимодействие заявок

В заключение разработаем нижний уровень взаимодействия – на уровне заявок и циклограмм. Для организации вызова дочерних циклограмм из заявки используется принцип стека (первым пришел, последним ушел), т. е. при вызове дочерней циклограммы образуется стек, в основании которого лежит головная циклограмма (заявка). Управление передается дочерней циклограмме (на первую операцию), и пока она не завершит работу родительская циклограмма исполняться не будет. Возврат управления из дочерней циклограммы в родительскую осуществляется на операции, следующей за операцией вызова дочерней (рис. 6).



Рис. 6. Взаимодействие заявки с дочерними циклограммами

В языке имеется возможность передачи параметров в циклограмму. Параметры циклограмм указываются при проектировании и при выполнении инициализируются автоматически или испытателем. Правила передачи параметров разрабо-

таны в соответствии с понятиями и типами данных. В качестве параметров могут быть телеметрические параметры, команды управления, значения вещественного или символьного типов и др.

Для поддержания структурности разрабатываемых циклограмм и расширения прикладного языка до уровня структурных языков в языке предусмотрены конструкции с единственным и множественным выбором. Так же поддерживается конструкция повторения.

Базовые множества. На основе созданной структуры языка необходимо разработать множества базовых понятий, которые послужат основой формируемого языка.

В соответствии с разработанными понятиями в базовое множество понятий языка необходимо включить базовое множество операций языка B^O , базовое множество переменных B^S , базовое множество реакций B^R и базовое множество алгоритмов дежурного контроля B^A . Следует заметить, что во множество базовых понятий не включены множества, образуемые понятиями, связанными и зависимыми от объекта контроля. Это позволяет разработать действительно универсальное ядро языка испытаний, независимое от параметров и воздействий, присущих конкретным объектам контроля.

Таким образом, базовое множество понятий языка представляет собой совокупность множеств:

$$B = B^O \cup B^S \cup B^R \cup B^A.$$

Рассмотрим, например, базовое множество операций языка. Как уже отмечалось, данное множество должно содержать операции, выполняющие базовые функции языка, т. е. выдачу команд и контроль параметров. Кроме того, в базовый набор операций необходимо включить операции, организующие общий ход выполнения, и структурные операции. Аналогичным образом разработаны базовые множества остальных понятий языка.

Итак, мы получили основу языка испытаний – множества базовых понятий, благодаря чему получена возможность построения специализированных языков испытаний, предназначенных для конкретных аппаратно-программных комплексов, за счет расширения множеств функциональных операций языка, алгоритмов дежурного контроля и системных переменных в рамках разработанной структуры языка.

Таким образом, представленные выше требования позволяют создать прикладной язык испытаний космических аппаратов с глубоким интегрированием в проблемную область и средствами

структурного программирования, а использование существующего задела обеспечивает легкость перехода пользователей со старого на новый прикладной язык.

Разработанная структура наделяет язык следующими возможностями. Максимальная проблемная ориентация языка как в области синтаксиса, так и в области структуры делает написание циклограмм испытаний максимально простым. Пользователи оперируют терминами проблемной области и применяют обычные структурные конструкции для описания сложных алгоритмов испытаний. Кроме того, язык позволяет использовать для разработки циклограмм простые средства визуального представления (структурный редактор с древовидным представлением содержимого [5]). Выделим следующие основные возможности языка:

- анализ штатной и технологической телеметрической информации;
- выдача штатных и технологических команд и закладка массивов;
- чтение дампов памяти из бортовой цифровой вычислительной машины;
- управление ходом функционирования системы испытаний;
- выдача сообщений и директив испытателю;
- управление интерпретатором языка испытаний.

В настоящее время данный язык испытаний применяется в Научно-производственном объединении прикладной механики имени академика М. Ф. Решетнева на ряде аппаратно-программных комплексах. А за счет расширения множества базовых понятий он может использоваться и на других рабочих местах, для различных объектов контроля.

Библиографический список

1. Описание ПРИС. Программное обеспечение УИК. Комплекс проведения испытаний : руководство оператора / Науч.-произв. об-ние прикл. механики. Железнодорожск, 1993. 104 с.
2. Ахо, А. Компиляторы: принципы, технологии и инструменты / А. Ахо, Р. Сети, Дж. Ульман. М. : Вильямс, 2001. 768 с.
3. Пратт, Т. Языки программирования: разработка и реализация : пер. с англ. / Т. Пратт, М. Зелковиц. 4-е изд. СПб. : Питер, 2002. 688 с.
4. Касьянов, В. Н. Методы построения трансляторов / В. Н. Касьянов, И. В. Поттосин. Новосибирск : Наука. Сиб. отд-ние. 1986. 347 с.
5. Барков, А. В. Структурный редактор прикладного языка / А. В. Барков // Решетневские чтения : материалы IX Междунар. науч. конф. / Сиб. гос. аэрокосмич. ун-т. Красноярск, 2005. С. 291.

A. V. Barkov

STRUCTURE OF THE PROBLEM-ORIENTED LANGUAGE OF SPACECRAFTS TESTS

Requirements to the problem-oriented language of spacecrafts tests are considered. The language structure, which answer the requirements, is worked out. It is described the opportunities of language.