

ПАРАМЕТРИЧЕСКИЙ ГЕНЕРАТОР УПРАВЛЯЮЩИХ ПРОГРАММ РЕАЛЬНОГО ВРЕМЕНИ

© 2014 Е.А.Мясникова, А.А.Тюгашев

Самарский государственный аэрокосмический университет имени академика С.П. Королёва
(национальный исследовательский университет)

Поступила в редакцию 17.12.2013

В статье описаны математическая модель и программная реализация параметрического генератора управляющих программ реального времени с возможностью получения текста на различных языках – Си, ассемблер, и т.д. Описан алгоритм генерации, который основан на многовходовой модели управляющей программы и использует описание стандартных конструкций целевого языка программирования. Рассмотрена структура программного управляющего алгоритма реального времени. Ключевые слова: управляющий алгоритм, управляющая программа реального времени, генерация программы, многовходовая модель.

При создании и эксплуатации современных космических аппаратов широко применяется вычислительная техника. Управление отдельными приборами и системами, входящими в комплекс бортовой аппаратуры, осуществляется при этом специальными функциональными программами. Схема управления подразумевает необходимость в некой координирующей программе, включающей в требуемые моменты времени в зависимости от ситуации на борту те или иные функциональные программы для достижения космическим аппаратом заданных целей. Такая программа, в частности, в ГНПРКЦ “ЦСКБ-Прогресс” называется “программой комплексного функционирования”, или *управляющим алгоритмом*. Разработка управляющих программ процесс долговременный, сложный, в программы человек может внести ошибки и опечатки, порождающие сбои в работе бортовых систем управления, что приводит к авариям и потере дорогостоящей космической техники.

В связи с этим актуальна задача создания настраиваемого на необходимый язык программирования автоматического генератора управляющих программ реального времени, позволяющего сделать процесс проще, ускорить ход работ и сократить число ошибок.

Основой любого генератора программ являются три ключевых компонента: метаданные (структура, которую мы пытаемся смоделировать в программе), шаблоны (образцы по которым будет создан код), и внутренние правила, которые определяют структуру и поведение ме-

таданных, шаблонов и их взаимодействие.

На выходе генератора получаем искомый программный код. Схема генерации представлена на рис. 1.



Рис. 1. Генерация программного кода

Метаданными является многовходовая модель управляющего алгоритма. Семантику управляющих алгоритмов реального времени (УА РВ) можно представить в виде набора четверок объектов:

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

где f_i – функциональная задача (программа); t_i – момент начала выполнения действия; τ_i – длительность; \bar{l}_i – логический вектор, обуславливающий действие [1]. Многовходовая модель это набор включений управляющего алгоритма в определенные моменты времени. Ее можно представить следующей парой:

$$MWM = \langle W, T \rangle, \quad (2)$$

где MWM – обозначение многовходовой модели, $W = \{W_1, W_2, \dots, W_n\}$ – множество входов управляющего алгоритма, $T = \{ \langle (W_i, W_j), \Delta T_{ij} \rangle \}$ –

отношение передачи управления, определённое на множестве W . То есть, T есть бинарное от-

*Мясникова Екатерина Андреевна, магистрант.
E-mail: myasnikovakate@gmail.com
Тюгашев Андрей Александрович, доктор технических наук,
профессор кафедры программных систем.
E-mail: tau797@mail.ru*

ношение на W , каждая пара входов в котором характеризуется временным интервалом ΔT_{ij} .

Если, например, пара $(W_i, W_j) \in T$, то это означает, что вход W_j запускается после выполнения входа W_i по прошествии временного интервала ΔT_{ij} . Таким образом, многовходовую модель можно представить в виде управляющего графа, где W – множество вершин, а T – множество взвешенных дуг. Пример графа многовходовой модели изображен на рис. 2.

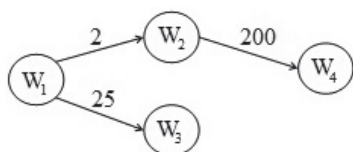


Рис. 2. Многовходовая модель

Каждый вход представляет собой логическую схему, т.е. совокупность линейных участков и ветвлений по результатам проверки логических условий.

$$W_i = \left\langle \left(LU^i, Y^i \right)_j \right\rangle, j = \overline{1, m}, \quad (3)$$

где $LU^i = (LU_1, LU_2, \dots, LU_n)$ – упорядоченная последовательность линейных участков, Y^i – вектор логических условий, при которых выполняется LU^i .

Каждый линейный участок можно описать в виде:

$$LU_i = \{ f_k \}, k = \overline{0, n}, \quad (4)$$

где $\{ f_k \}$ последовательность функциональных задач без проверок логических условий и передач управления [2].

Вход можно представить также в виде графа, где вершинами являются линейные участки, а дугами – передачи управления, которые обусловлены проверками условий логических переменных.

Пример графа входа приведен на рис. 3.

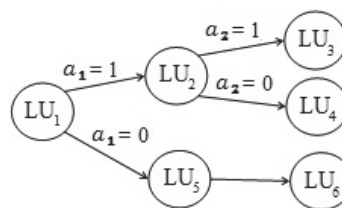


Рис. 3. Логическая схема входа

Шаблоны генератора это образцы стандартных конструкций языка программирования, на который настраивается генератор. В программе обязательно присутствуют стандартные начальный блок и завершающий блок, оформленные в соответствии с правилами конкретного целевого языка программирования. Поддержка множества входов реализуется с помощью меток и операторов безусловного перехода. Кроме этого необходима генерация условных операторов реализации ветвлений, а также описание выхода из программы в диспетчер бортовой операционной системы. В программах реального времени к этому необходимо добавить конструкцию, обеспечивающую необходимую задержку (уставку временного интервала). Структура программы представлена на рис. 4, в блоках текста должны быть описаны последовательности запусков функциональных задач.

Если настраиваемый язык не содержит оператора безусловного перехода, то его можно заменить комбинацией специальной переменной, цикла ПОКА и оператора выбора (переключателя).

Внутренние правила генератора определяют алгоритм генерации управляющей программы. Мы начинаем генерацию с вставки стандартного начала программы и обработки первого по времени исполнения входа. Обработка входа это рекурсивное построение дерева линейных участков. Повторы исключаются вставкой команды перехода на уже описанный участок. Каждая из функциональных задач будет заменена соответствующим фрагментом кода. Некоторые функциональные задачи запрашивают на включение

```

#include "opslib.h"
int main(int argc, char* argv[])
{
    M1: <блок кода>
    if(<логическая переменная>)
        goto M0;
    M2: <блок кода>
    goto M1;
    M0: return 0;
}

```

Метки

← Стандартное начало программы

← Оператор ветвления

← Оператор безусловного перехода

← Выход из программы

← Стандартное завершение программы

Рис. 4. Пример структуры программы

другие входы, исходя из этой информации, формируется очередь входов. После рассмотрения одного входа в основном алгоритме он удаляется из очереди. Повторение алгоритма продолжается до тех пор, пока очередь не опустеет. После этого мы завершаем генерируемую программу и основной алгоритм.

Таким образом, разработанный программный модуль параметрической генерации использует подход, основанный на преобразовании логико-временной схемы управляющего алгоритма над базовыми конструкциями требуемого языка программирования в программный код управляющей программы реального времени. Базовые конструкции языка программирования описываются в формате XML, логико-временная схема так же интерпретируется в XML-структуру. Это позволяет использовать технологию генерации с помощью XSLT (языка преобразования XML). XSLT имеет массу преимуществ: выполнение повторной генерации кода без перекомпиляции приложения (необходимо только изменить шаблон); возможность генерации программы на любом языке программирования; использование наглядного и широко распрост-

раненного XML формата [3].

Разработанный модуль генерации является частью программного комплекса ГРАФКОНТ, разрабатываемого по заказу ГНПРКЦ "ЦСКБ-ПРОГРЕСС" на кафедре программных систем СГАУ. Задача системы ГРАФКОНТ состоит в том, чтобы автоматизировать процессы проектирования, создания, документирования и тестирования бортовых управляющих алгоритмов и программ реального времени для космических аппаратов, обеспечить повышение качества и надежности программ, а так снизить трудоемкость и стоимость разработки.

СПИСОК ЛИТЕРАТУРЫ

1. *Калентьев А.А., Тогашев А.А.* ИПИ/CALS технологии в жизненном цикле комплексных программ управления. Самара: Изд-во Самарского научного центра РАН, 2006. 285 с.
2. *Трусов В.С.* Система визуального конструирования временных диаграмм управляющих алгоритмов беспилотных ЛА: дисс. ... канд. техн. наук. Самара, 2005. 116 с.
3. *Канжелев С.Ю., Шальто А.А.* Автоматическая генерация автоматного кода // Информационно-управляющие системы. 2006. № 6. С.35-42.

PARAMETRIC GENERATOR OF REAL-TIME CONTROL PROGRAMS

© 2014 Е.А.Мясникова, А.А.Тугашев

Samara State Aerospace University named after Academician S.P. Korolyov
(National Research University)

In this article we described mathematic model and program implementation of the parametric generator of real-time control programs with the possibility of obtaining code in different programming languages – C, assembler etc. We considered generation algorithm, based on the multi-input model of control program and used description basic structures of target programming language. Also we described structure of software real-time control algorithm.

Key words: control algorithm, real-time control program, code generation, multi-input model.

Ekaterina Myasnikova, Candidate for a Master's degree.

E-mail: myasnikovakate@gmail.com

Andrey Tyugashev, Doctor of Technics, Professor at the Program System Department. E-mail: tau797@mail.ru