

© 2023 г. В.В. КУЛЬБА, д-р техн. наук (kulba@ipu.ru),  
Е.А. МИКРИН, академик РАН,  
Б.В. ПАВЛОВ, д-р техн. наук (pavlov@ipu.ru),  
С.К. СОМОВ, канд. техн. наук (ssomov2016@ipu.ru)  
(Институт проблем управления им. В.А. Трапезникова РАН, Москва)

## ТЕХНОЛОГИЯ КОМПЛЕКСНОЙ ОТРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ БОРТОВЫХ СИСТЕМ КОСМИЧЕСКИХ АППАРАТОВ

В работе представлены основные положения концепции комплексной отработки программного обеспечения бортового комплекса управления космическими аппаратами. Выполнена постановка и предложен метод решения задачи выбора оптимальной стратегии комплексной отработки бортового комплекса программ. Предложены методы отработки программного обеспечения по показателям функциональной корректности.

*Ключевые слова:* космический аппарат, программное обеспечение, бортовой комплекс управления, комплексная отработка программного обеспечения.

DOI: 10.31857/S0005231023100057, EDN: YEDKZJ

### 1. Введение

Широкое разнообразие задач, решаемых при эксплуатации КА, экстремальные условия их функционирования, а также различные технологии и протоколы информационного взаимодействия между элементами бортовой аппаратуры и программным обеспечением, включая различные датчики и индикаторы определяют необходимость создания новых подходов, методов и технологий поддержки научных и технологических разработок в области перспективной космической техники. Проблемам разработки и внедрения методов моделирования в аэрокосмической отрасли посвящены работы [1–5]. В этих работах важное место занимают вопросы цифрового моделирования — актуального метода проведения исследований различных аспектов работы бортовых комплексов управления космическими аппаратами (БКУ КА), включая проектирование, создание и отработку их программного обеспечения (ПО БКУ КА) [6, 7]. Анализ опыта применения современных организационных, методических и технических решений, используемых для отработки ПО БКУ КА, выявил необходимость разработки основных положений методологии комплексной отработки ПО БКУ КА [6]. Решение данной задачи остро необходимо для эффективной разработки и отработки ПО БКУ КА за счет использования прототипов ПО, а также ранней функциональной интеграции и итеративной проверки выполнения требований к ПО. Данная методология отработки ПО БКУ КА используется для оптимизации процесса комплексной отработки ПО с использованием стоимостных и временных критериев с учетом разнообразных технологических ограничений.

## 2. Выбор оптимальной стратегии комплексной отработки

В формальных постановках задач выбора оптимальной стратегии комплексной отработки программного обеспечения БКУ КА обычно используются два критерия оптимальности. Первый критерий — минимум времени, второй — минимум стоимости проведения отработки. Цель общей задачи выбора оптимальной стратегии комплексной отработки заключается в определении: 1) оптимального разбиения структуры КП на отдельные части, 2) множества необходимых подпрограмм — «заглушек» и подпрограмм — «драйверов», 3) сценария отработки выделенных ранее отдельных частей КП. При постановке задачи используются ограничения, которые определяют допустимые варианты разбиения и объединения специального графа  $\Gamma$ , вершины которого соответствуют программным модулям ПО, а дуги — это связи по управлению между вершинами. При разработке тестов и при локализации ошибок используются графовые модели, которые детализируют граф  $\Gamma$ . Данные графовые модели применяются для формализации детальных блок-схем КП. Кроме того, данные графовые модели применяются и для формализации детальных блок-схем отдельных программных модулей (ПМ).

Множество различных стратегий комплексной отработки ПО БКУ определяется следующим образом. На начальном этапе необходимо определить множество всех допустимых способов разбиения графа  $\Gamma$  на подграфы. Для каждого полученного подграфа проводится автономное тестирование. Далее для полученных подграфов определяется множество всех допустимых вариантов их объединения. Эти варианты объединения подграфов используются для связного тестирования ПО. Каждая из множества стратегий комплексной отработки определяется следующим образом. Во-первых, это множество подграфов  $p^m = \{p_1, \dots, p_l, \dots, p_M\}$ , полученных при разбиении графа  $\Gamma$ . Во-вторых, очередностью объединения данных подграфов. Объединение подграфов в соответствии с данной очередностью позволяет получить исходную графовую структуру  $\tilde{p}^{mn} = \{\tilde{p}_1^{mn}, \dots, \tilde{p}_k^{mn}, \dots, \tilde{p}_N^{mn}\}$ . Здесь  $\tilde{p}_N^{mn}$  совпадает с графом  $\Gamma$ .

Цель решения задачи выбора оптимальной стратегии комплексной отработки заключается в поиске такого разбиения  $p^{m*}$  графа  $\Gamma$  и такой последовательности объединения подграфов  $\tilde{p}^{mn*}$ , которые при совместном использовании позволяют получить такой сценарий комплексной отработки, который обеспечит получение оптимальных значений временных и стоимостных характеристик стратегий отработки.

В случае, если процесс отработки использует  $mn$ -стратегию, то затраты времени и стоимости на комплексную отладку будут состоять из двух следующих компонент:  $\bar{T}_{mn}^p(\bar{C}_{mn}^p), \bar{T}_{mn}^o(\bar{C}_{mn}^o)$ . Первая компонента  $\bar{T}_{mn}^p(\bar{C}_{mn}^p)$  — это время  $\bar{T}_{mn}^p$  и (стоимость  $\bar{C}_{mn}^p$ ) автономной отработки подграфов графа  $\Gamma$ , полученных при разбиении графа для  $mn$ -стратегии. Вторая компонента  $\bar{T}_{mn}^o(\bar{C}_{mn}^o)$  — время  $\bar{T}_{mn}^o$  и (стоимость  $\bar{C}_{mn}^o$ ) реализации этапов объединения подграфов и последующей связной отработкой подграфов для  $mn$ -стратегии. Величина стоимостных затрат и времени автономной отработки подграфов, полученных при разбиении графа  $\Gamma$ , рассчитывается при помощи

следующих двух выражений:

$$\bar{T}_{mn}^p = \sum_{\nu} t_{\nu mn}; \quad \bar{C}_{mn}^p = \sum_{\nu} C_{\nu mn},$$

где  $t_{\nu mn}$  и  $C_{\nu mn}$  — время и стоимость автономной отработки  $\nu$ -го подграфа разбиения графа  $\Gamma$ .

При объединении подграфов связанная отработка будет иметь характеристики времени и стоимости, значения которых определяются выражениями:

$$\bar{T}_{mn}^o = \sum_k b_{kmn}; \quad \bar{C}_{mn}^o = \sum_k S_{kmn}.$$

Здесь  $b_{kmn}$  и  $S_{kmn}$  — время и стоимость связанной отработки на  $k$ -м этапе объединения подграфов.

Задача определения оптимальной стратегии с использованием временного критерия в общем виде имеет следующую формулировку: необходимо найти минимум выражения:

$$\sum_{mn} (\bar{T}_{mn}^p + \bar{T}_{mn}^o) x_{mn}.$$

С учетом ограничения на стоимость отработки

$$\sum_{mn} (\bar{C}_{mn}^p + \bar{C}_{mn}^o) x_{mn} \leq C.$$

В данном ограничении используется переменная  $x_{mn}$ :

$$x_{mn} = \begin{cases} 1, & \text{если выбрана } mn\text{-стратегия комплексной отработки;} \\ 0 & \text{в противном случае.} \end{cases}$$

В ограничении также используется константа  $C$ , определяющая максимально допустимые затраты на проведение КП.

В процессе решения данной задачи поиск вариантов разбиения на подграфы графа  $\Gamma$  состоит в выборе состава  $V$  групп программных модулей КП, где  $V$  — число программных модулей в составе КП БКУ. При решении задачи требуется соблюдать ограничения на допустимые комбинации программных модулей для каждой из  $V$  групп.

В ходе выбора варианта объединения подграфов из множества подграфов  $p^m = \{p_1, \dots, p_m, \dots, p_M\}$  до исходной структуры графа  $\Gamma$  требуется определить перечень этапов объединения  $V^*$  непустых подграфов в исходную структуру графа  $\Gamma$ . Максимальное число таких этапов должно быть равно числу непустых подграфов  $V^*$ .

Однако если количество программных модулей в КПО (компонент программного обеспечения) велико, то тогда и множество возможных вариантов стратегий системной отработки становится слишком большим. Из-за чего оценка значений временных и стоимостных характеристик стратегий становится чрезвычайно ресурсоемкой задачей, требующей много времени. Для

решения данной проблемы предлагается решать более частные задачи поиска оптимальной стратегии системной обработки ПО, так как на практике именно такие задачи возникают чаще всего.

Определим множество  $\overline{P}^p$  вариантов разбиения графа  $\Gamma$  на подграфы таким образом:

$$\overline{P}^p = \{P^m\}, \quad m = \overline{1, M}.$$

Здесь  $P^m = \{p_1^m, \dots, p_\nu^m, \dots, p_{D_m}^m\}$  —  $m$ -й вариант разбиения,  $p_\nu^m$  —  $\nu$ -й подграф,  $D_m$  — это количество подграфов при  $m$ -м варианте разбиения.

Множество  $\overline{P}^o = \{\tilde{P}^{mn}\}$ , ( $n = \overline{1, N_m}$ ,  $m = \overline{1, M}$ ) задает варианты объединения графа  $\Gamma$ . Элемент  $\tilde{P}^{mn} = \{\tilde{p}_1^{mn}, \dots, \tilde{p}_k^{mn}, \dots, \tilde{p}_{F_{mn}}^{mn}\}$  множества  $\overline{P}^o$  — это  $n$ -й вариант объединения при  $m$ -м варианте разбиения графа  $\Gamma$ . Величины  $N_m$  и  $F_{mn}$  — это число полученных вариантов объединения подграфов и число этапов объединения для  $m$ -го варианта разбиения графа  $\Gamma$ .

Для  $n$ -го варианта объединения элемент  $\tilde{p}_k^{mn}$  определяется следующим образом:

$$\tilde{p}_k^{mn} = \bigcup_{v \in R1_k^{mn}} p_v^m \bigcup_{i \in R2_k^{mn}} p_i^m.$$

Здесь  $R1_k^{mn}$  — множество индексов подграфов из  $P^m$ , а  $R2_k^{mn}$  — множество индексов подграфов из  $\tilde{P}^{mn}$ , включенных в  $k$ -й этап связной обработки для  $m$ -го варианта разбиения при  $n$ -м варианте объединения графа  $\Gamma$ .

Стратегия комплексной обработки, с одной стороны, определяется вариантом разбиения графа  $\Gamma$  на подграфы  $P^m \in \overline{P}^p$ , а с другой стороны — вариантом объединения полученных подграфов  $\tilde{P}^{mn} \in \overline{P}^o$  в исходную графовую структуру.

Время  $t_\nu$  и стоимость  $C_\nu$  автономной обработки каждого  $\nu$ -го подграфа разбиения состоят из трех компонент: времени и стоимости ( $t_\nu^n, C_\nu^n$ ) подготовки данных для тестирования, времени и стоимости ( $t_\nu^p, C_\nu^p$ ) процесса тестирования и времени и стоимости ( $t_\nu^r, C_\nu^r$ ) локализации ошибок, обнаруженных при тестировании подграфа:

$$t_\nu = t_\nu^n + t_\nu^p + t_\nu^r, \quad C_\nu = C_\nu^n + C_\nu^p + C_\nu^r,$$

где

$$t_\nu^n = t_\nu^r + t_\nu^3 + t_\nu^d, \quad C_\nu^n = C_\nu^r + C_\nu^3 + C_\nu^d.$$

В приведенных выше формулах использованы следующие обозначения:  $t_\nu^r$  — время, а  $C_\nu^r$  — стоимость генерации тестовых данных для  $\nu$ -го подграфа;  $t_\nu^3$  — время и  $C_\nu^3$  — стоимость создания подпрограмм «заглушек», требуемых для организации обработки  $\nu$ -го подграфа;  $t_\nu^d$  — время и  $C_\nu^d$  — стоимость создания подпрограмм «драйвера», необходимых для обработки  $\nu$ -го подграфа;  $t_\nu^p$  — время, а  $C_\nu^p$  — стоимость выполнения тестов для  $\nu$ -го подграфа;  $t_\nu^r$  — время, а  $C_\nu^r$  — стоимость локализации ошибок, обнаруженных при тестировании  $\nu$ -го подграфа.

Задача поиска оптимальной стратегии системной обработки может быть сведена к двум шагам. На первом шаге производится выбор варианта разбиения графа  $\Gamma$  на подграфы  $P^m \in \overline{P}^p$  для их автономной обработки. На

втором шаге выполняется выбор варианта объединения этих подграфов из множества  $\bar{P}^o$  для связной отработки. Эти два шага обеспечивают выполнение комплексной отработки при минимальных затратах времени и стоимости при ограничениях на стоимость и время отработки.

Определим переменную  $y_{mn}$  для ее использования в постановке задачи выбора оптимальной стратегии отработки следующим образом:

$$y_{mn} = \begin{cases} 1, & \text{если для } m\text{-го варианта разбиения графа } \Gamma \\ & \text{выбирается } n\text{-й вариант объединения;} \\ 0, & \text{в противном случае.} \end{cases}$$

При решении данной задачи используются следующие исходные данные:

- 1) Множества  $\bar{P}^p = \{P^m\}$ ,  $m = \overline{1, M}$ ,  $\bar{P}^o = \{\tilde{P}^{mn}\}$ ,  $n = \overline{1, N_m}$ ,  $m = \overline{1, M}$ .
- 2) Временные и стоимостные характеристики автономной, а также и связной отработки.

При этом время и стоимость для комплексной отработки задаются следующими выражениями:

$$\bar{T}^k = \bar{T}_m^p + \bar{T}_{mn}^o, \quad \bar{C}^k = \bar{C}_m^p + \bar{C}_{mn}^o,$$

где  $\bar{T}_m^p$  – время, а  $\bar{C}_m^p$  – стоимость автономной отработки в случае  $m$ -го варианта разбиения графа  $\Gamma$ ;  $\bar{T}_{mn}^o$  – время, а  $\bar{C}_{mn}^o$  – стоимость этапов связной отработки в случае  $m$ -го варианта разбиения графа  $\Gamma$  и  $n$ -го варианта объединения графа  $\Gamma$ .

Время  $\bar{T}_m^p$  автономной отработки и ее стоимость  $\bar{C}_m^p$  определяются с использованием следующих формул:

$$\begin{aligned} \bar{T}_m^p &= \sum_{\nu} (t_{\nu m}^r + t_{\nu m}^s + t_{\nu m}^d + t_{\nu m}^l), \\ \bar{C}_m^p &= \sum_{\nu} (c_{\nu m}^r + c_{\nu m}^s + c_{\nu m}^d + c_{\nu m}^l). \end{aligned}$$

Если для подграфов  $p_{\nu}^m \in P^m$  определены наборы тестов, используемые при их отладке, а также для них известны временные и стоимостные характеристики, то вычисление временных и стоимостных характеристик автономной отработки выполняется с использованием следующих соотношений:

$$\begin{aligned} t_{\nu m}^r &= \sum_{j=1}^{J_{\nu m}} t_{j\nu} m^r, & c_{\nu m}^r &= \sum_{j=1}^{J_{\nu m}} \hat{c}_{j\nu} m^r, \\ t_{\nu m}^{p^l} &= \sum_{j=1}^{J_{\nu m}} t_{j\nu} m^{p^l}, & c_{\nu m}^{p^l} &= \sum_{j=1}^{J_{\nu m}} \hat{c}_{j\nu} m^{p^l}, \\ t_{\nu m}^l &= \sum_{j=1}^{J_{\nu m}} t_{j\nu} m^l \rho, & c_{\nu m}^l &= \sum_{j=1}^{J_{\nu m}} \hat{c}_{j\nu} m^l \rho. \end{aligned}$$

В данных соотношениях используется величина  $J_{\nu m}$ , которая обозначает множество тестов, требуемых для проверки подграфа  $p_{\nu}^m$ .

Переменные  $t_{\nu m}^3$ ,  $c_{\nu m}^3$  задают соответственно время и стоимость создания всех подпрограмм «заглушек», необходимых для проверки подграфа  $p_{\nu}^m$ , т.е.

$$t_{\nu m}^3 = \sum_{i=1}^{I_{\nu m}} \hat{t}_{i\nu}^3; \quad c_{\nu m}^3 = \sum_{i=1}^{I_{\nu m}} \hat{c}_{i\nu}^3.$$

В данных выражениях величина  $I_{\nu m}$  определяет количество подпрограмм «заглушек», которые требуются при отладке подграфа  $p_{\nu}^m$ .

Затраты времени  $\bar{T}^o$  и стоимости  $\bar{C}^o$  выполнения этапов связной обработки для  $m$ -го варианта разбиения и  $n$ -го варианта объединения графа  $\Gamma$  определяются выражениями:

$$\bar{T}^o = \sum_{k=1}^{F_{mn}} (b_{kmn}^n + b_{kmn}^{p\pi} + b_{kmn}^{\pi});$$

$$\bar{C}^o = \sum_{k=1}^{F_{mn}} (S_{kmn}^n + S_{kmn}^{p\pi} + S_{kmn}^{\pi}).$$

Предположим, что для проверки подграфов  $\tilde{p}_k^{mn} \in \tilde{P}^{mn}$  заданы множества тестов, для которых определены их характеристики времени и стоимости. Тогда для определения величины затрат времени и стоимости на выполнение  $k$ -го этапа связной обработки для  $m$ -го варианта разбиения и  $n$ -го варианта объединения графа  $\Gamma$  используются следующие формулы:

$$b_{kmn}^n = \sum_{j=1}^{J_{kmn}} \hat{b}_{jkmn}^n; \quad b_{kmn}^{p\pi} = \sum_{j=1}^{J_{kmn}} \hat{b}_{jkmn}^{p\pi}; \quad b_{kmn}^{\pi} = \sum_{j=1}^{J_{kmn}} \hat{b}_{jkmn}^{\pi} \rho,$$

$$S_{kmn}^n = \sum_{j=1}^{J_{kmn}} \hat{S}_{jkmn}^n; \quad S_{kmn}^{p\pi} = \sum_{j=1}^{J_{kmn}} \hat{S}_{jkmn}^{p\pi}; \quad S_{kmn}^{\pi} = \sum_{j=1}^{J_{kmn}} \hat{S}_{jkmn}^{\pi} \rho.$$

Используя приведенные выше выражения для определения значений временных и стоимостных характеристик процесса обработки ПО, можно дать формальную постановку оптимизационной задачи поиска оптимальной стратегии реализации сценария комплексной обработки при использовании минимума общего времени обработки в качестве критерия оптимизации:

$$\sum_m \left( \sum_m \bar{T}_m^p \sum_{n=1}^{N_m} y_{mn} + \sum_{n=1}^{N_m} \bar{T}_m^o y_{mn} \right) \rightarrow \min.$$

При решении задачи используются следующие ограничения:

— ограничение на стоимость реализации обработки,

$$\sum_m \left( \bar{C}_m^p \sum_m y_{mn} + \sum_{n=1}^{N_m} \bar{C}_m^o y_{mn} \right) \leq C,$$

— множество  $M$  следующих ограничений на переменные  $y_{mn}$ :

$$\sum_{n=1}^{N_m} y_{mn} = 1; \quad m = \overline{1, M}.$$

Аналогично формулируется задача поиска оптимальной стратегии комплексной обработки с использованием стоимостного критерия оптимизации:

$$\sum_m \left( \bar{C}_m^p \sum_n y_{mn} + \sum_{n=1}^{N_m} \bar{C}_m^o y_{mn} \right) \rightarrow \min$$

при использовании ограничения на время, затраченное на проведение отладочных работ:

$$\sum_m \left( \sum_n \bar{T}_m^p \sum_{n=1}^{N_m} y_{mn} + \sum_{n=1}^{N_m} \bar{T}_m^o y_{mn} \right) \leq T$$

с учетом множества  $M$  ограничений на переменные  $y_{mn}$ .

Сформулированные выше задачи относятся к классу широко используемых задач линейного математического программирования.

### 3. Методы обработки ПО БКУ КА с использованием показателей функциональной корректности

На стадии ранней функциональной интеграции компонент БКУ КА используются показатели функциональной корректности для оценки правильности реализации функций комплекса программ БКУ КА. Каждый функционал КП БКУ КА реализуется на использовании некоторого множества маршрутов обработки данных. При следовании по данным маршрутам входная область определения функции преобразуется в один выходной результат данной функции или во множество выходных результатов. С целью полной проверки корректности работы любой функции необходимо проверить все множество маршрутов обработки данных, используемых данной функцией, при заданном множестве входных параметров функции. Для подтверждения корректности работы функций необходимо, чтобы выходные результаты этих функций полностью совпадали с эталонными результатами, определенными в спецификациях комплекса программ. Проверка корректности работы функций на полном множестве всех входных данных и на всех маршрутах их обработки — это задача очень большой сложности. Поэтому возникает необходимость выбора ограниченного подмножества маршрутов обработки данных для их проверки. Это подмножество маршрутов должно позволить проверить реализацию основных функций КП.

На данный момент для проверки корректности функционирования ПО используются два подхода: функциональный или структурный. Функциональный подход использует представление программного обеспечения в виде “черного ящика”. Структурный подход основан на проверке правильности выполнения маршрутов обработки данных и при подготовке тестов учитывает особенности структуры отдельных модулей КП и особенности межмодульного взаимодействия в рамках комплекса программ. Как функциональный, так и структурный подходы имеют существенные недостатки с позиции эффективности реализации проверки программного обеспечения [2].

С учетом данного обстоятельства предлагается метод, который использует положительные свойства обоих подходов. Данный метод предполагает

выбор такого множества тестов показателей функциональной корректности комплекса программ, которое необходимо для проверки корректности работы комплекса. Оценка качества работы КП делается на основе результатов выполнения множества отобранных тестов.

Метод подразумевает отбор из множества  $F$  комплекса программ, реализующих все основные и вспомогательные функции, такого подмножества  $\overline{F} < F$  функций, подлежащих проверке, в результате корректной работы которых будут получены требуемые значения показателей функциональной корректности КП.

Для комплекса программ задана область  $\overline{E}$  входных данных. Для каждой функции  $F_j \in \overline{F}$  определено соответствующее подмножество  $E_j \in \overline{E}$  области входных данных КП. Каждая из таких функций выполняет определенное преобразование данных из входной области  $E_j \in \overline{E}$  в соответствующие данные выходной области  $y_j \in \overline{Y}$ . Здесь множество  $y_j$  содержит все возможные значения выходных данных для функции  $F_j$  ( $j = \overline{1, J}$ ).

Выходные результаты  $y_{kj} \in Y$  комплекса программ получаются в ходе реализации множеств маршрутов  $M_{jk}$  ( $j = \overline{1, J}; k = \overline{1, K}$ ), при прохождении которых происходит обработка данных. Следовательно, можно сделать вывод о том, что для проверки множества функций  $\overline{F}$  комплекса программ необходимо проверить корректность работы множества маршрутов обработки данных. В результате реализации этих маршрутов должны быть получены необходимые выходные результаты  $y_{kj}$  соответственно для каждой функции  $F_j$  из множества  $\overline{F}$  с использованием подмножеств входных данных  $E_j \in \overline{E}$ .

Функция  $F_j$  комплекса программ будет считаться проверенной, если для всех выходных результатов  $y_{kj} \in Y_j$  этой функции успешно проверена корректность прохождения множества  $M_{jk}$  ( $j = \overline{1, J}; k = \overline{1, K}$ ) для всех тех маршрутов обработки данных, прохождение которых обеспечило получение выходных результатов для функции  $F_j$ . Множества  $M_{jk} \in \overline{M}_j$ ,  $k = \overline{1, K}$  таких маршрутов будут считаться множествами  $\overline{M}_j$  магистральных путей для функции  $F_j$ . Показатель  $N_{kj}$  будет использоваться для определения корректности получения результата  $j$ -й функции. Его значение равно отношению:

$$N_{kj} = \frac{n_{kj}^{\text{np}}}{n_{kj}^{\text{общ}}}.$$

В данном отношении  $n_{kj}^{\text{np}}$  – это количество проверенных магистральных путей, а  $n_{kj}^{\text{общ}}$  – общее количество магистральных путей, формирующих результаты  $y_{kj} \in Y_j$ . Значение общего количества магистральных путей равно мощности множества  $M_{kj}$ .

Будем использовать магистральный путь в качестве основной проверяемой единицы при оценке показателя функциональной корректности ПО и графовую модель  $\Gamma(V, C)$  укрупненной блок-схемы КП при выполнении сценария отработки и определения магистральных путей для функций множества  $\overline{F}$ .

В графовой модели  $V$  – множество вершин графа  $\Gamma$ , которое соответствует множеству блоков укрупненной блок-схемы КП, а  $C$  – множество дуг графа.



Дуги  $C$  отображают передачу управления между блоками блок-схемы КП. Блоки блок-схемы – отдельные процедуры или их совокупности либо программные модули КП. Дуга между блоками  $i$  и  $j$  означает переход управления от блока  $i$  к блоку  $j$ . В рассматриваемой модели вершине  $v_i \in V$  графа  $\Gamma(V, C)$  соответствуют как множества ее аргументов  $A_i = \{a_{in}\}$ , так и множества ее результатов  $R_i = \{r_{ij}\}$ .

Маршрут обработки информации  $m$  на графе  $\Gamma(V, C)$  – это последовательность вершин и дуг  $(v_0, c_0, v_1, c_1, \dots, c_{I-1}, v_I)$ . В данной последовательности  $v_i$  ( $0 \leq i \leq I$ ) это вершина графа  $\Gamma(V, C)$ , а  $c_i$  ( $1 \leq i \leq I - 1$ ) это дуга связи от вершины  $v_i$  к вершине  $v_{i+1}$ . Последовательность  $(v_0, \dots, v_I)$  вершин, в свою очередь, соответствует тем преобразованиям, которые реализуются по мере следования маршрута  $m$  обработки данных. Такая последовательность называется преобразователем маршрута  $m$ , а последовательность  $(c_0, \dots, c_{I-1})$  дуг соответствует условиям, которые при прохождении маршрута  $m$  должны быть выполнены, и называется условием маршрута  $m$ .

Такая последовательность называется преобразователем маршрута  $m$ , а последовательность  $(c_0, \dots, c_{I-1})$  дуг соответствует условиям, которые при прохождении маршрута  $m$  должны быть выполнены, и называется условием маршрута  $m$ .

Магистральным путем  $m_{jk}$  для результата  $y_{jk} \in Y_j$  функции  $F_j \in \bar{F}$  будем считать такой маршрут, преобразователь маршрута  $(v_0, \dots, v_i)$  которого включает в себя как минимум одну из возможных последовательностей внешних и внутренних информационных связей. При этом данные внешние и внутренние связи должны начинаться в вершине  $v_0$  и заканчиваться в вершине  $v_i$  получения результата  $y_{jk}$ .

#### 4. Организация отработки программного обеспечения БКУ РС МКС

В данной главе представлен пример использования изложенной выше концепции для отработки элементов конфигурации программного обеспечения (ЭКПО) российского сегмента МКС (БКУ РС МКС) [3].

В ходе отработки ЭКПО поэтапно выполнено:

- 1) автономное тестирование КПО;
- 2) комплексная отработка ЭКПО на стенде наземного комплекса отработки (НКО);
- 3) отработка ПО совместно с американской системой C&C MDM (мультиплексор-демультиплексор бортовой центральной вычислительной машины (БЦВС) американского сегмента и МКС в целом);
- 4) квалификационные формальные тесты ЭКПО.

Перечисленные этапы отработки ЭКПО решают задачи по обнаружению, локализации и устранению ошибок, возникающих в процессе отработки ПО, задачи по подтверждению работоспособности ПО и по оценке соответствия функционирования ПО техническому заданию.

**Автономное тестирование** программного обеспечения производится на базе автономного рабочего места (АРМ) персонального компьютера и на ком-

плексе “SDDF” (средства разработки проекта программного обеспечения). Тестирование выполняется в соответствии с методикой, в которой используются: описание процедуры проверки, начальные условия тестирования, а также контрольные примеры проверок. После того, как тестирование ПО закончено, оно передается группе конфигурационного контроля, которая интегрирует протестированное ПО в состав ЭКПО БЦВМ.

**Комплексная отработка ЭКПО** выполняется по специально разработанному сценарию и решает следующие задачи:

- 1) проверка качества работы операционной системы;
- 2) сборка ПО БКУ и комплексная отработка ПО в соответствии с планом полета и режимами работы российского сегмента и служебного модуля одновременно с контролем обеспечения безопасности полета (т.е. проверка корректности реализации подграфов графа  $\Gamma$ , так и самого графа  $\Gamma$  в целом);
- 3) выборочные проверки тех магистральных путей, которые соответствуют наиболее вероятным нештатным ситуациям, а также локализация нештатных ситуаций и их устранение;
- 4) проверка соответствия функционирования ПО БКУ документам (ICD SSP 50 097);
- 5) контроль процесса распределения таких ресурсов, как память, процессорное время, работа каналов ввода/вывода.

**Совместные тесты с американской системой C&C MDM** выполнялись на стендах “SITE-C”, “EGSE” и “SVF” по специальным сценариям реализации тестов. В процессе испытаний использовались бортовое ПО БКУ АС и ПО БКУ РС, модельное ПО БС Американского сегмента и модельное ПО БС Российского сегмента.

**Формальные квалификационные тесты или приемо-сдаточные испытания (ПСИ) и стыковочные испытания** — это процесс, в ходе которого проверяется соответствие ЭКПО ЦВМ требованиям, указанным в ТЗ и ICD.

Из множества тестов, которые реализуются на наземном комплексе НКО, выбирается определенное подмножество тестов, которые используются для проверки корректности реализации выбранного множества магистральных путей. По завершении формального квалификационного тестирования оформляется и подписывается «Заказчиком» заключение о том, что ЭКПО ЦВМ готов к проведению стыковочных испытаний.

Стыковочные испытания проводились в соответствии с методикой, разработанной для таких испытаний. Аппаратно-программные средства БЦВС (АПС БЦВС) проходят стыковочные испытания с реальными аппаратными средствами или с ее аналогами на базе наземного комплекса отработки НКО-2. Стыковочные испытания АПС БЦВС с АПС БЦВС Американского сегмента проводились на базе наземного комплекса отработки НКО-1. Эти испытания проводились в соответствии с планом совместной отработки NASA-RSA Phase 2-3 Bilateral Integration and Verification Plan – SSP50101. Испытания аппаратно-программных средств БЦВС в составе служебного мо-

дуля «Звезда» (индекс изделия 17КСМ) проводились на комплексном стенде № 24008 и на контрольно-испытательной станции в необходимом объеме.

## 5. Заключение

В данной работе представлен существующий опыт, организационные, методические и технические решения по отработке программного обеспечения БКУ для космических аппаратов различного назначения. Приведены основные положения технологии комплексной отработки ПО БКУ КА, которая обеспечивает эффективную разработку ПО, его отладку на основе использования прототипов ПО, итеративную верификацию требований и раннюю функциональную интеграцию. Предложенная технология реализована в рамках автоматизированной системы разработки и отработки ПО БКУ КА, что позволило существенно снизить общее число ошибок в процессе разработки и отработки программного обеспечения российского сегмента МКС.

## СПИСОК ЛИТЕРАТУРЫ

1. *Микрин Е.А., Кульба В.В., Павлов Б.В.* Разработка моделей и методов проектирования информационно-управляющих систем космических аппаратов // *АиТ*. 2013. № 3. С. 38–50.
2. *Микрин Е.А., Кульба В.В., Косяченко С.А., Сомов Д.С., Гладков Ю.М.* Комплексная отработка программного обеспечения бортового комплекса управления космическими аппаратами и имитационные модели функционирования бортовых систем и внешней среды. Препринт. М.: ИПУ РАН, 2011.
3. *Кульба В.В., Микрин Е.А., Павлов Б.В., Платонов В.Н.* Теоретические основы проектирования информационно-управляющих систем космических аппаратов. М.: Наука, 2006.
4. *Куренков В.И., Кучеров А.С.* Методы исследования эффективности ракетно-космических систем. Проблемно-ориентированные системы автоматизированного проектирования. Изд-во Самар. гос. аэрокосм. ун-та, 2012.
5. *Зеленцов В.А., Ковалев А.П., Охтилев М.Ю., Соколов Б.В., Юсупов Р.М.* Методология создания и применения интеллектуальных информационных технологий наземно-космического мониторинга сложных объектов / *Труды СПИИРАН*. 2013. Вып. 5(28).
6. *Микрин Е.А.* Бортовые комплексы управления космическими аппаратами и проектирование их программного обеспечения М.: МГТУ им. Н.Э. Баумана, 2003. 336 с.
7. *Микрин Е.А.* Принципы построения бортовых комплексов управления автоматических космических аппаратов / *Е.А. Микрин, Н.А. Суханов, В.Н. Платонов и др.* // *Проблемы управления*. 2004. № 3. С. 62–66.

*Статья представлена к публикации членом редколлегии В.М. Глузовым.*

Поступила в редакцию 19.06.2023

После доработки 18.07.2023

Принята к публикации 02.08.2023