

УДК (004.4'236:004.424.2):629.78.064.018

И.Б. ТУРКИН, Ю.А. КУЗНЕЦОВА

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

МОДЕЛЬ УПРАВЛЯЮЩИХ АЛГОРИТМОВ ИСПЫТАНИЙ БОРТОВЫХ СИСТЕМ КОСМИЧЕСКИХ АППАРАТОВ

Испытания и экспериментальная отработка таких сложных технических объектов, как бортовые системы космических аппаратов, во многом определяют стоимость и продолжительность всего цикла разработки космического аппарата. Автоматизация процессов испытаний невозможна без тщательной проработки управляющих алгоритмов, для тестирования и отладки которых требуются соответствующие средства визуализации, наглядно представляющие оператору и разработчику космического комплекса последовательность и содержание режимов и проверок. Показано, что из-за сложности и иерархичности управляющих алгоритмов существует необходимость в разработке модели управляющего алгоритма испытаний бортовых систем космических аппаратов, которая предназначена для верификации таких алгоритмов, а также для анализа результатов работы программного обеспечения испытаний.

Ключевые слова: космический аппарат, бортовые системы, испытания, управляющий алгоритм, визуализация, модель вычислений, управляемых данными.

Введение

Разработка систем космических аппаратов (КА) является одним из наиболее наукоёмких видов инженерной деятельности. Сложность процессов разработки объясняется особенностями создаваемых систем, повышенными требованиями к надёжности функционирования и долговечности космической техники, жёсткими условиями эксплуатации КА, невозможностью технического обслуживания.

Общей тенденцией развития бортовых систем КА является увеличение количества и расширение типов решаемых задач, что во многом достигается за счет повышения вычислительной мощности микропроцессоров, установленных в системах. Основными используемыми в настоящее время методами обеспечения надёжности для бортовых систем КА являются тестирование и отладка управляющих алгоритмов реального времени (УА РВ), которые не могут гарантировать отсутствия ошибок. При этом, в связи с необходимостью отработки взаимодействия с бортовой аппаратурой (БА) при всех возможных ситуациях на борту КА (в т.ч. нештатных), их трудоёмкость является наибольшей среди этапов жизненного цикла программного обеспечения (ПО) испытаний и составляет, по экспертным оценкам, около 57% общей трудоёмкости [1]. Учитывая сжатые сроки, необходимы более эффективные подходы и средства к разработке ПО, реализующего УА испытаний.

Одним из таких подходов является визуализация УА ПО, наглядно представляющая оператору и разработчику космического комплекса последова-

тельность и содержание режимов и проверок, предусмотренных документацией, для определения оптимального (по продолжительности и глубине проверок) алгоритма испытаний.

Визуализация также обусловлена необходимостью продолжительной адаптации ПО к условиям применения КА, так как необходимость в модификациях УА возникает на протяжении всей экспериментальной отработки КА, когда изменениям и доработкам подвергается и объект, и процессы испытаний.

1. Анализ исследований и публикаций

Под **экспериментальной отработкой** понимается процесс активного воздействия на исследуемый объект, проводимый с помощью материальных средств в задаваемых условиях в целях выявления или подтверждения свойств и характеристик объекта [2]. *Итог отработки* – установление фактических показателей функционирования систем КА и последующее сопоставление их с некоторыми априорными критериями, которые, в свою очередь, могут корректироваться по результатам испытаний.

Экспериментальная отработка КА охватывает такие направления работ, как:

1. Наземные испытания отдельных частей КА или КА в целом в условиях, максимально имитирующих натурные условия эксплуатации, исследование параметров и условий работы систем и агрегатов КА;

2. Экспериментальное определение лётно-технических характеристик КА в ходе лётных испытаний;

3. Доводочные работы по изменению конструкции КА и режимов его работы, выполняемые на основании анализа результатов указанных выше экспериментальных исследований.

Общие задачи экспериментальной обработки:

1. Оценка правильности основных конструктивных и схемных решений, положенных в основу проекта КА, и их корректировка в процессе отработки.

2. Проверка и отработка функционирования агрегатов КА, отдельных конструктивных узлов и приборов в эксплуатационных условиях, а также отработка их взаимодействия в общей конструктивной схеме.

3. Определение лётно-технических характеристик КА в полном диапазоне условий его применения.

4. Исследование и устранение в процессе отработки причин всех возможных неисправностей, которые могут привести к отказу или аварии при подготовке и пуске КА или головного блока с полезной нагрузкой при полёте с ракетой-носителем на активном участке траектории [3].

5. Исследование и устранение в процессе отработки причин неисправностей, которые могут привести к нарушению нормального полёта КА на орбитальном участке траектории.

6. Отработка технологии эксплуатации КА.

7. Доведение отработки до такого состояния, при котором предстартовая подготовка КА, проведенная в соответствии с эксплуатационной документацией, целиком определяет режим его полёта, гарантирует выведение КА на заданную орбиту, его безопасность, а также выполнение задачи полёта и возвращение КА при необходимости.

Учитывая определение, направления работ и задачи экспериментальной отработки, можно сделать вывод о том, что *испытания* представляют собой упорядоченную во времени и пространстве последовательность действий, работ, мероприятий и операций по подготовке, проведению и обработке результатов испытаний, а также совокупность средств и методов их выполнения.

Процесс испытаний протекает в рамках организационно-технической системы испытаний (СИ). СИ представляет собой совокупность средств, методов испытаний и исполнителей, взаимодействующих с объектом испытаний (рис. 1) [4].

Важным элементом, во многом определяющим эффективность, результативность и успешность испытаний, является ПО. Оно обеспечивает интеграцию бортовых систем и наземной контрольно-измерительной и контрольно-проверочной аппаратуры (КИА и КПА) в единый программно-информационный комплекс, а также решает следующие задачи:

– организацию процессов управления системами и подсистемами КА;

– организацию обмена командной и контрольной информацией между (КИА и КПА) и бортовыми системами КА;

– организацию приёма, логической обработки и выдачи команд управления (управляющих воздействий) на бортовые системы от наземного комплекса управления (НКУ) на этапе эксплуатации КА и наземного испытательного комплекса на этапе наземной отработки КА;

– формирование телеметрической информации (ТМИ) для НКУ;

– управление информационными потоками и распределение задач;

– сопряжение программно-алгоритмической составляющей бортовых систем с аппаратурой систем и агрегатов посредством приборных интерфейсов.

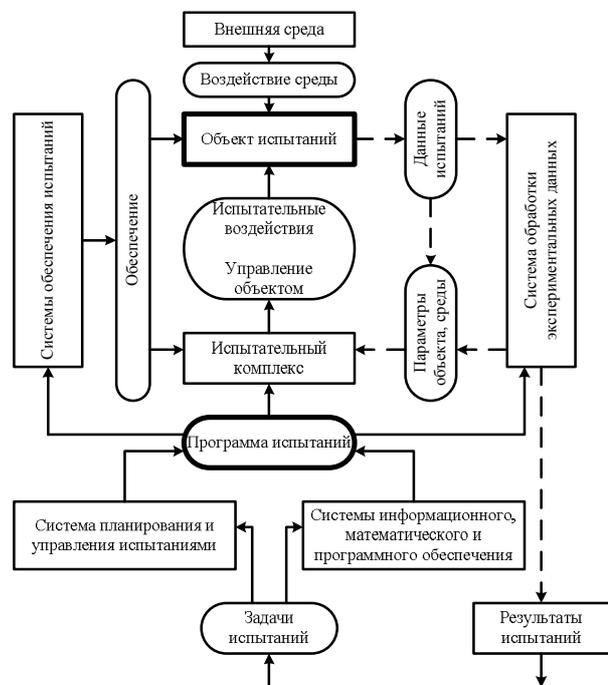


Рис. 1. Обобщённая структура системы испытаний

ПО СИ, реализующее управляющий алгоритм для автоматизации испытаний, в значительной мере определяет логику, динамику и точность функционирования объекта управления.

Для приложений данного типа характерной является обработка больших объёмов данных, поступающих от десятков тысяч датчиков различного назначения, а также иерархическая структура УА. Это обусловлено тем, что среди ошибок ПО испытаний (ПОИ) значительное количество приходится на сбои синхронизации и согласования логики управления БА при одновременном функциониро-

вании ряда бортовых систем и программ ПОИ в рамках решения КА целевых задач (ошибки в УА РВ) [5].

Наиболее значимая часть испытаний бортовых систем проводится в автоматическом режиме, а продолжительность отдельных режимов испытаний составляет часы. При этом сложная иерархическая структура УА усложняет понимание оператором как структуры УА в целом, так и отдельных составных частей, что не позволяет ускорить процесс обнаружения и устранения ошибок, которые могут привести к возникновению нештатной ситуации.

Основным направлением решения этой проблемы является использование интерактивной визуализации УА испытаний. Такой подход должен учитывать широкий круг различных параметров, в том числе психофизические характеристики пользователей, разнообразие и взаимосвязь решаемых пользователем задач и используемых при их решении данных, неравномерность, слабую предсказуемость и различную приоритетность поступающих запросов.

2. Постановка задачи

Из-за сложности и иерархичности УА, изменения последовательности и содержания режимов и проверок становится затруднительным описание правил представления алгоритмов сбора и преобра-

зований данных, формирования управляющих воздействий и выдачи сообщений оператору.

Поэтому целью данной работы является разработка модели управляющего алгоритма испытаний бортовых систем КА, которая предназначена для верификации таких алгоритмов и анализа результатов работы ПО СИ.

3. Модель управляющего алгоритма реального времени

Пусть обработка информации в ПО испытаний подсистем КА производится в соответствии с моделью вычислений, управляемых данными [6]:

$$MB_UD = \langle Data, Task, DT, TD \rangle, \quad (1)$$

где: $Data = \{Data_i\}$ – множество объектов данных, $i = \overline{1, N}$.

$Task = \{Task_j\}$ – множество элементарных задач-преобразователей информации, $j = \overline{1, M}$.

$DT = Data \times Task$ – матрица, характеризующая правила инициирования задач при изменении (обновлении) данных.

$TD = Task \times Data$ – матрица, характеризующая получение (порождение) новых данных в результате работы задач-преобразователей информации.

На рис. 2 представлена структурная схема взаимодействия элементов MB_УД (1).

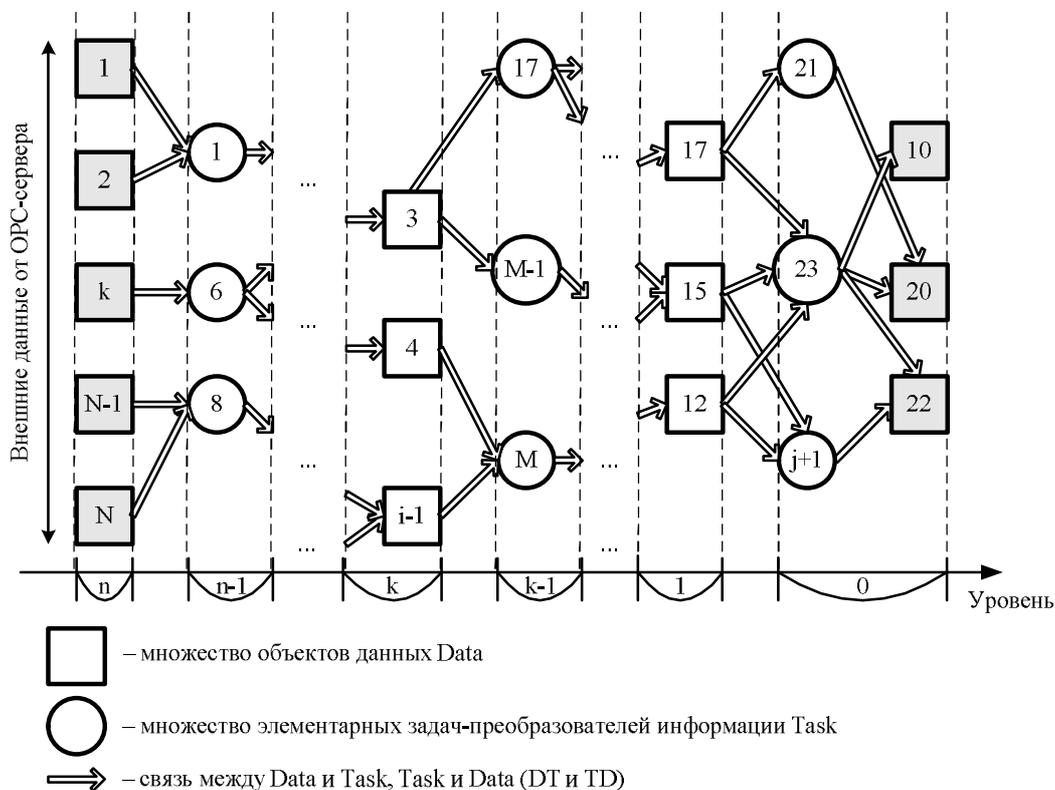


Рис. 2. Структурная схема взаимодействия элементов MB_УД

Данные. Формально объект данных $Data_i \in Data$ можно описать с помощью кортежа:

$$Data_i = \langle DataValue_i, TimeStamp_i, Quality_i \rangle,$$

где $DataValue_i$ – значение i -того объекта данных в некоторый $TimeStamp_i$ момент времени (отметка времени), $Quality_i$ – качество i -того объекта данных, $i = \overline{1, N}$ [7].

Задачи. В результате работы преобразователей информации $Task_j \in Tasks$ будут порождаться объекты данных – $Data_{outTask_j}$. Поэтому целесообразно рассматривать расширенное множество объектов данных как объединение двух непересекающихся подмножеств данных: сенсорных – $Data_{sensorTask}$, полученных из внешнего мира, и вычисляемых – $Data_{calculated}$, то есть порождаемых работой задач $Tasks$ [8].

Правила инициализации задач. Пусть $e \in DT$. Ребро e означает наличие связи, при этом вес ребра, характеризующий тип этой связи:

$$w_e = \begin{cases} 1 - \text{данные используются задачей } Task_j; \\ 2 - \text{по изменению } Data_i \text{ иницируется } Task_j; \\ 3 - \text{по обновлению } Data_i \text{ иницируется } Task_j; \\ i = \overline{1, N}, j = \overline{1, M}. \end{cases}$$

Управляющий алгоритм – алгоритм верхнего уровня, управляющий в реальном времени виртуальной машиной, в которой реализованы элементарные операции ввода/вывода и преобразования информации, а также выдачи в каждый заданный момент времени t_i (или на протяжении заданного промежутка времени Δt) корректного управляющего воздействия. Следовательно, УА представляет собой множество подграфов G_{DataCA_i} общего графа вычислений $G_{DataMB_УД}$, которые формируют выходные данные $Data_{outTask_j}$ как команды управления Y :

$$CA = \{G_{DataCA_i}\}, G_{DataCA_i} \subset G_{Data};$$

$$\forall G_{DataCA_i} \in G_{Data},$$

$$G_{DataCA_i} : (Data_{outTask_j}) \longrightarrow Y,$$

Следовательно, УА РВ целесообразно представить в виде кортежа:

$$CA_{RT} = \left\langle \begin{matrix} Data_{CA}, Task_{CA}, \\ Data_{CA} \times Task_{CA}, \\ Task_{CA} \times Data_{CA} \end{matrix} \right\rangle,$$

где: $Data_{CA} \subset Data$; $Task_{CA} \subset Task$;

$Data_{CA} \times Task_{CA} \subset DT$; $Task_{CA} \times Data_{CA} \subset TD$.

Пусть известно множество команд, формируемых УА: $Data_{\hat{E}O} \subset Data_{CA}$. Тогда УА как множество данных, задач и связей между ними (3) можно

определить рекурсивно:

1) определяется множество задач 0-го уровня $Task^0 = \{t_j\}$, $j = \overline{1, M}$ таким образом, что:

$$\forall t_j \in Task^0, \exists d_i \in Data^0,$$

$$(Data^0 \equiv Data_{\hat{E}O}) \wedge (td_{ji} \in TD);$$

2) определяется множество данных 1-го уровня $Data^1 = \{d_i\}$, $i = \overline{1, N}$ таким образом, что:

$$\forall d_i \in Data^1, \exists t_j \in Task^0, dt_{ij} \in DT;$$

3) множество задач k -го уровня ($k \geq 1$) $Task^{(k)}(t_j)$ определяется по известному множеству данных k -го уровня $Data^{(k)}(d_i)$ и известному множеству $TD(dt_{ji})$:

$$\forall t_j \in Task^k, \exists d_i \in Data^k, td_{ji} \in TD;$$

4) множество данных k -го уровня ($k \geq 1$) $Data^{(k)}(d_i)$ определяется по известному множеству задач $Task^{(k-1)}(t_j)$ ($k-1$ -уровня) и известному множеству $DT(dt_{i-1,j})$:

$$\forall d_i \in Data^k, \exists t_j \in Task^{k-1}, dt_{i-1,j} \in DT;$$

Вычисления (4)-(7) останавливаются, когда выполняется условие:

$$((\forall t_j \in Task^k) \wedge (\forall d_i \in Data^k)) \exists$$

$$\exists((dt_{ij} \in DT) \wedge (d_i \in Data_{in})),$$

где $Data_{in} = \{Data_{in}^k\}$; $Data_{in}^k \subset \{Data^k\}$, $k = 0 \dots (n-1)$.

Формулы (4)-(8) означают, что были последовательно по всем $(n-1)$ уровням определены все объекты данных d_i УА и все задачи-преобразователи t_j этих данных, при этом $i = \overline{1, N}$; $j = \overline{1, M}$; $k = 0 \dots (n-1)$ – уровень УА; $Data_{in} \equiv Data_{sensorTask}$.

Для того, чтобы упорядочить полученные при обратном проходе УА вершины $Data$ и $Task$ графа G_{DataCA_i} по $(n-1)$ уровням, необходимо выполнить сортировку этого графа при прямом проходе УА:

$$Data^{(n-1)} = Data^{(n-1)} - Data_{in};$$

$$Task^{(n-2)} = Task^{(n-2)} - Task^{(n-1)};$$

$$Data^{(k)} = Data^{(k)} - \bigcup_{i=k-1}^{n-1} Data^{(i)}, k = (n-2) \dots 1;$$

$$Task^{(k)} = Task^{(k)} - \bigcup_{i=k-1}^{n-1} Task^{(i)}, k = (n-2) \dots 1.$$

Заключение

Ключевые свойства ПО, реализующего УА испытаний бортовых систем КА, могут быть достигнуты при разработке модели УА РВ и метода его

визуализации, что в итоге способствует повышению наглядности такого ПО и снижению трудоёмкости процессов тестирования и отладки управляющих алгоритмов. Модель УА, представленная в статье, позволяет решать задачи визуализации управляющих алгоритмов и анализа результатов работы ПО СИ. Перспективой дальнейшей работы является разработка на основе предложенной модели метода визуализации УА РВ.

Литература

1. Тюгашев, А.А. *Технология проектирования надёжных управляющих алгоритмов реального времени для космических аппаратов* [Текст] / А.А. Тюгашев // *Вестник СГАУ*. – 2004. – №1. – С. 124 – 131.
2. *Солнечные энергосистемы космических аппаратов. Физическое и математическое моделирование* [Текст] / К.В. Безручко, Н.В. Белан, И.Т. Перекопский, И.Б. Туркин и др. – Х: НАУ «ХАИ», 2000. – 440 с.
3. *Ракеты и космические аппараты ГКБ «Южное» им. М.К. Янгеля* [Текст] / А.Н. Мащенко, В.Н. Паппо-Корыстин, В.А. Пащенко, В.Г. Васильев. – Д: ООО «КолорГраф», ООО РА «Тандем – У», 2001. – 240 с.
4. *Экспериментальная отработка космических летательных аппаратов* [Текст] / В.А. Афанасьев, В.С. Барсуков, М.Я. Гофин, Ю.В. Захаров и др. – М: МАИ, 1994. – 412 с.
5. *Теоретические основы проектирования информационно-управляющих систем космических аппаратов* [Текст] / В.В. Кульба, Е.А. Микрин, Б.В. Павлов, В.Н. Платонов. – М: Наука, 2006. – 579 с.
6. Туркин, И.Б. *Модель вычислений, управляемых данными, в программном обеспечении систем реального времени* [Текст] / И.Б. Туркин, Е.В. Соколова // *Радиоэлектронные и компьютерные системы*. – 2010. – № 6 (47). – С. 13–18.
7. *OPC Data Access Custom Interface: Industry Standard Specification, Version 3.00, Released* / OPC Foundation. – March 4, 2003. – 190 p.
8. *Практические аспекты реализации динамического планирования запросов в клиент-серверных системах на основе OPC* [Текст] / И.Б. Туркин, Е.В. Соколова, Ю.А. Шенетов, Т.С. Никитина // *Радиоэлектронные и компьютерные системы*. – 2008. – № 6 (33). – С. 58–61.

Поступила в редакцию 17.02.2012

Рецензент: д-р техн. наук, проф., зав. каф. информатики А.Ю. Соколов, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

МОДЕЛЬ КЕРУЮЧИХ АЛГОРИТМІВ ВИПРОБУВАНЬ БОРТОВИХ СИСТЕМ КОСМІЧНИХ АПАРАТІВ

І.Б. Туркін, Ю.А. Кузнецова

Випробування та експериментальне відпрацювання таких складних технічних об'єктів, як бортові системи космічних апаратів, багато в чому визначають вартість і тривалість усього циклу розробки космічного апарату. Автоматизація процесів випробувань неможлива без ретельного опрацювання керуючих алгоритмів, для тестування і налагодження яких потрібні відповідні засоби візуалізації, що наочно представляють оператору та розробнику космічного комплексу послідовність і зміст режимів та перевірок. Показано, що через складність і ієрархічність керуючих алгоритмів існує необхідність в розробці моделі керуючого алгоритму випробувань бортових систем космічних апаратів, яка призначена для верифікації таких алгоритмів, а також для аналізу результатів роботи програмного забезпечення випробувань.

Ключові слова: космічний апарат, бортові системи, випробування, керуючий алгоритм, візуалізація, модель обчислень, керованих даними.

THE TEST CONTROL ALGORITHM MODEL OF THE SPACECRAFT BOARD SYSTEMS

I.B. Turkin, Yu.A. Kuznetsova

The tests and experimental refining of such complex technical objects as spacecraft board systems largely determine the cost and the duration of whole cycle of spacecraft development. Automation of test processes is impossible without a thorough elaboration of control algorithms, which require the appropriate visualization tools for their testing and debugging, obviously representing the content and the sequence of modes and checks to the space complex developer and operator. It is shown that there is a need of creation of the test control algorithm model of the spacecraft board systems, which is used for verification of such algorithms, as well as for analysis results of test software due to the complexity and hierarchy of control algorithms.

Key words: spacecraft, board systems, tests, control algorithm, visualization, data-driven computing model.

Туркин Игорь Борисович – д-р техн. наук, проф., зав. каф. инженерии программного обеспечения, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: energy@d4.khai.edu.

Кузнецова Юлия Анатольевна – аспирант каф. инженерии программного обеспечения, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: JK-Sv@yandex.ru.