

УДК 629.78.018

И.Б. ТУРКИН, П.А. ЛУЧШЕВ

Национальный аэрокосмический университет им. Н.Е. Жуковского “ХАИ”, Украина

МОДЕЛИ И СПОСОБЫ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ИСПЫТАНИЯМИ СИСТЕМ КОСМИЧЕСКОГО АППАРАТА

Выполнена разработка комплекса моделей, способов и информационных средств для автоматизации технологических процессов испытаний на основе системы энергоснабжения (СЭС) космических аппаратов (КА). Рассмотрены некоторые аспекты применения методов и средств автоматизированного конструирования информационных систем для разработки ПО автоматизации технологических процессов испытаний сложных технических систем (СТС).

испытания, технологический процесс испытаний, формальные модели, модель испытаний

Введение

Цель статьи заключается в разработке комплекса моделей, способов и информационных средств для автоматизации технологических процессов испытаний сложных технических систем.

Общая проблема заключается в большом разнообразии видов испытаний СТС и схем обработки данных, когда сотни и тысячи информационных сигналов и команд управления связаны с жесткими ограничениями на время выполнения операций при наличии внешнего контура управления, включающего человека-оператора. При этом, например, экспериментальная отработка космического аппарата и его систем сопровождается практически весь его жизненный цикл и по некоторым оценкам стоимость исследовательских и экспериментально-отрабочных испытаний достигает 80% стоимости создания комплексов КА [1].

Актуальность разработки специализированных моделей и способов автоматизированного управления испытаниями СТС базируется на том, что современные методы автоматизированного конструирования информационных систем, основанные на концептуальных и функциональных моделях, являются скорее философией, а не технологией компьютерной инженерии [2, 3].

1. Результаты исследований

Основные понятия. Для формализации модели испытаний воспользуемся определением испытаний

по ГОСТ 16504-81.

Тогда для определения свойств объекта необходимо получить оценку в виде множества результатов **Results**, полученных в результате множества воздействий **Actions** на объект испытаний множеством проверок **Checks** при его функционировании на множестве режимов **Modes**.

Таким образом, можно сказать, что испытание проведено, если в результате некоторых функциональных преобразований F_{test} , определенных на множестве временных отметок **T** и зависящих от режимов и проверок, произведена оценка и получено множество результатов.

На практике, испытания в зависимости от цели классифицируются по видам.

Это позволяет распределить весь процесс испытания по этапам и определить каждый этап, как отдельное испытание со своими подмножествами режимов, проверок и результатов:

$$\mathbf{Modes}_{test} \subseteq \mathbf{Modes}; \quad (1)$$

$$\mathbf{Checks}_{test} \subseteq \mathbf{Checks}; \quad (2)$$

$$\mathbf{Result}_{test} \in \mathbf{Results}. \quad (3)$$

Синтез системы управления испытаниями СТС. Общий вид системы испытаний можно представить в виде модели взаимодействия объекта управления, объекта испытаний, устройства управления испытаниями и оператора (рис. 1).



Рис. 1. Общий вид модели системы испытаний СТС

Каждый технологический процесс испытаний явно или неявно предполагает существование некоторого семейства допустимых наборов исходных данных и некоторого множества результатов. Таким образом, каждое испытание можно считать функциональным преобразованием, реализующим решение конкретной прикладной задачи. Чтобы описать такое решение, необходимо охарактеризовать выполняемое функциональное преобразование.

Из приведенной на рисунке модели следует, что устройство управления испытаниями должно выполнять функциональное преобразование F_{test} по управлению, текущей и общей оценке испытаний:

$$F_{test} : \mathbf{Actions} \times \mathbf{Sub}_{out} \times \mathbf{T} \rightarrow \langle \mathbf{Sub}_{in}, \mathbf{Messages}_{test}, \mathbf{Result}_{test} \rangle. \quad (4)$$

В качестве входных данных для устройства управления выступает дискретно поступающая информация о состоянии объекта испытаний и действиях оператора. Выходными сигналами являются дискретные управляющие воздействия и текущая оценка функционирования в виде сообщений о ходе выполнения программы испытаний. После завершения испытаний устройство управления также должно сформировать интегральную оценку испытаний.

2. Архитектура программных средств автоматизации технологических процессов испытаний

Программные средства автоматизации технологических процессов испытаний состоят из взаимодействующих компонентов различного назначения.

При этом необходима конструкция, обеспечивающая прозрачный для человека переход от элементарных команд управления и действий с ОИ, до высокоуровневых последовательностей. Поэтому, несмотря на кажущуюся простоту (4) техническая реализация может столкнуться с разноплановыми требованиями. В такой ситуации будет разумно провести декомпозицию общей задачи испытаний на функциональные блоки для внешнего управления F_{hmi} , автоматического F_{fp} и получения результата F_{res} :

$$F_{test} = F_{hmi} \times F_{fp} \times F_{res}. \quad (5)$$

Кроме этого необходим функциональный блок F_{io} для организации информационного обмена с аппаратурой и блок лингвистической обработки F_{lng} для реализации основной идеи по исключению разработчиков программы из процесса создания сценариев ТП испытаний. В этом случае программные средства автоматизации технологических процессов испытаний можно изобразить в виде взаимодействующих компонентов, представленных на рис. 2.

Процессор ввода-вывода. Объект испытаний, как сложная техническая система состоит из нескольких составных частей, модулей или устройств, которые зачастую создаются различными разработчиками, работа которых координируется разработчиком всей системы. Под процессором ввода/вывода будем понимать совокупность управляющей программы-дешифратора и некоторой платформенно-ориентированной библиотеки подпрограмм, обеспечивающих собственно взаимодействие F_{io} с внешними устройствами (6):

$$F_{io} : \langle \mathbf{Output} \times \mathbf{T}, \mathbf{Sub}_{in} \rangle \rightarrow \langle \mathbf{Sub}_{out}, \mathbf{Input} \rangle. \quad (6)$$

Основными функциями такого процессора являются преобразования для получения информации

$$\mathbf{Output} \times \mathbf{T} \rightarrow \mathbf{Sub}_{out}$$

и выдачи команд $\mathbf{Sub}_{in} \rightarrow \mathbf{Input}$.

Таким образом, процессор ввода-вывода (рис. 3) позволяет абстрагироваться от особенностей взаимодействия с внешними устройствами, предоставляя на внешних интерфейсах множества сигналов \mathbf{Sub}_{in} , \mathbf{Sub}_{out} . Каждый элемент данных множеств характеризуется текущей величиной, признаком качества и отметкой времени.

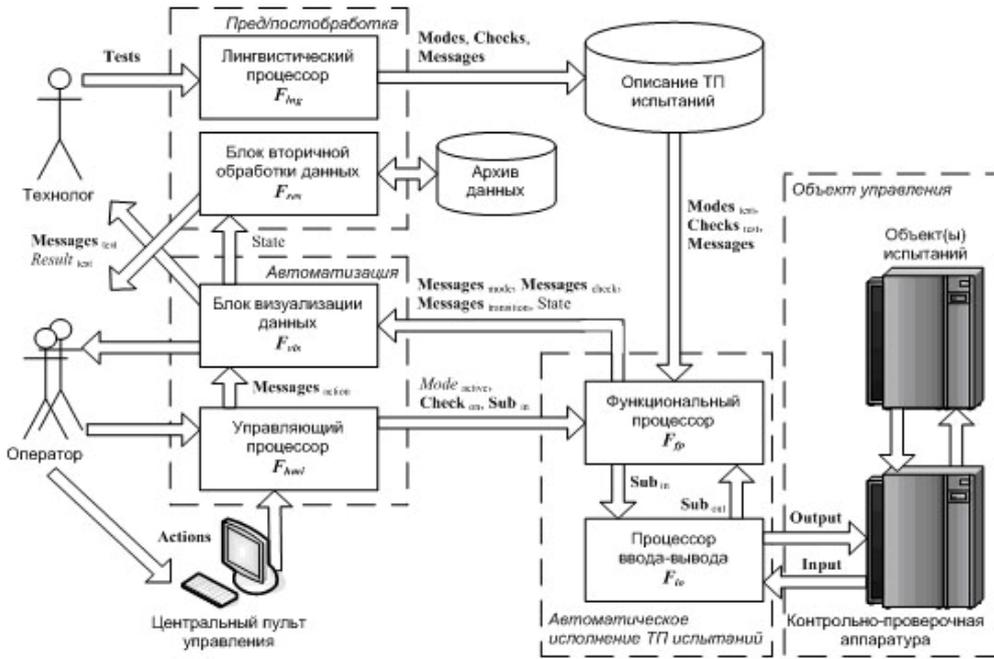


Рис. 2. Архитектура ПО автоматизации ТП испытаний

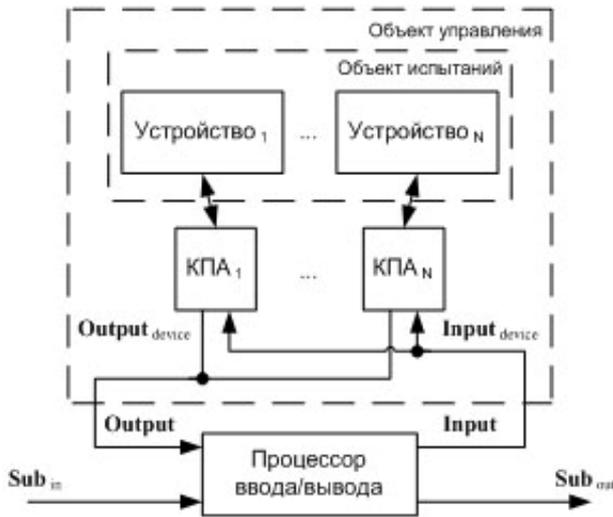


Рис. 3. Модель взаимодействия с объектом испытаний

Для сложных технических систем как входные **Input**, так и выходные **Output** сигналы можно представить как непересекающиеся множества входов **Input_{device}** и выходов **Output_{device}** более простых устройств, из которых состоит объект испытания:

$$\text{Input} = \bigcup_{\text{device}} \text{Input}_{\text{device}},$$

$$\forall i, j (i \neq j) \text{Input}_i \cap \text{Input}_j \equiv \emptyset; \quad (7)$$

$$\text{Output} = \bigcup_{\text{device}} \text{Output}_{\text{device}},$$

$$\forall i, j (i \neq j) \text{Output}_i \cap \text{Output}_j \equiv \emptyset. \quad (8)$$

Модель устройства управления испытаниями.

Система управления испытаниями может быть полностью реализована в виде программных средств автоматизации ТП испытаний, модель которых представлена на рис. 4. Данная модель позволяет охарактеризовать выполняемую работу как функцию проведения испытания в соответствии с (4).

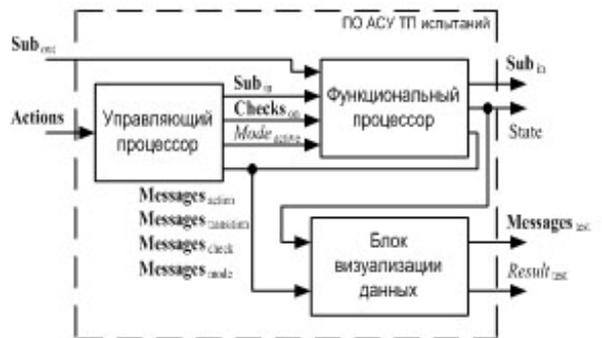


Рис. 4. Модель программного устройства управления

Действия оператора могут быть декомпозированы по функциональной сложности на множество единичных команд **Actions_{command}**, множество действий по безусловной активации указанного режима работы ОИ **Actions_{mode}** и принудительного управления состоянием проверок **Actions_{check}**:

$$\text{Actions} = \text{Actions}_{\text{command}} \cup \text{Actions}_{\text{mode}} \cup \text{Actions}_{\text{check}}. \quad (9)$$

Управляющий процессор и блок накопления и обработки данных относятся к средствам взаимодействия. Каждый из этих блоков должен быть рассчитан на работу с оператором, в качестве которого может выступать как человек в случае автономных испытаний, так и другой компьютер в случае комплексных испытаний. Функции управляющего процессора F_{hmi} сводятся к преобразованию требуемых действий **Actions**, задаваемых на входе, к низкоуровневому воздействию на объект испытаний или активации требуемого режима работы или активации/деактивации некоторого набора проверок:

$$F_{hmi}: \mathbf{Actions} \rightarrow \langle \mathbf{Messages}_{action}, \mathbf{Action} \rangle,$$

$$\text{где } \mathbf{Action} = \begin{cases} \mathbf{Sub}_{in}, & \mathbf{Action} \in \mathbf{Actions}_{command}; \\ \mathbf{Checks}_{on}, & \mathbf{Action} \in \mathbf{Actions}_{check}; \\ \mathbf{Mode}_{active}, & \mathbf{Action} \in \mathbf{Actions}_{mode}. \end{cases} \quad (10)$$

К функциям блока накопления и обработки данных относится мониторинг поступающих сообщений о ходе выполнения программы испытания и преобразования внутреннего формата сообщений в приемлемый для человека вид:

$$F_{res}: \mathbf{Messages}_{test} \rightarrow \mathbf{Result}_{test}. \quad (11)$$

Так как все сообщения разделены на категории по типу (отказы, предупреждения, норма, отладка) и по месту возникновения:

$$\begin{aligned} \mathbf{Messages} &= \mathbf{Messages}_{fault} \cup \mathbf{Messages}_{alarm} \\ &\cup \mathbf{Messages}_{info} \cup \mathbf{Messages}_{debug} \\ &= \mathbf{Messages}_{mode} \cup \mathbf{Messages}_{check} \\ &\cup \mathbf{Messages}_{actions} \cup \mathbf{Messages}_{transition}. \end{aligned} \quad (12)$$

то результат испытания \mathbf{Result}_{test} можно определить следующим образом:

$$\mathbf{Result}_{test} = \begin{cases} \mathbf{Correct}, & \begin{cases} \mathbf{Messages}_{test} \cap \\ \mathbf{Messages}_{fault} \cap \\ \mathbf{Messages}_{alarm} = \emptyset \end{cases} \\ \mathbf{Warnings}, & \begin{cases} (\mathbf{Messages}_{test} \cap \\ \mathbf{Messages}_{fault} = \emptyset) \\ \wedge (\mathbf{Messages}_{test} \cap \\ \mathbf{Messages}_{alarm} \neq \emptyset) \end{cases} \\ \mathbf{Fault}, & \begin{cases} \mathbf{Messages}_{test} \cap \\ \mathbf{Messages}_{fault} \neq \emptyset \end{cases} \end{cases} \quad (13)$$

В данной ситуации *Correct* соответствует успешному завершению испытания, *Warnings* – успешное

завершение с замечаниями и *Fault* – в процессе испытания выявлены несоответствия техническим требованиям. Кроме этого возможно получение дополнительной информации в процессе испытаний. Корректность действий оператора, например, можно оценить по пересечению множества сообщений, полученных в процессе выполнения испытания, с предопределенным множеством предупреждений $\mathbf{Messages}_{alarm} \cap \mathbf{Messages}_{test}$ или оценить корректность работы программного обеспечения по множеству отладочных сообщений $\mathbf{Messages}_{debug} \cap \mathbf{Messages}_{test}$.

Функциональный процессор (рис. 5) является самой сложной подсистемой обработки информации, которая фактически отвечает за автоматизированный режим проведения испытаний.

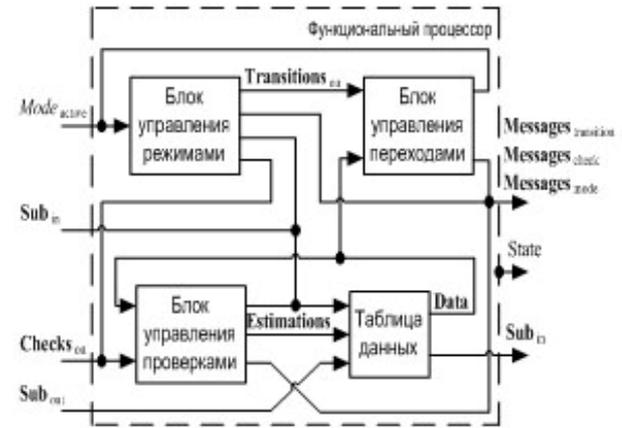


Рис. 5. Модель функционального процессора автоматизации испытаний

Текущее состояние функционального процессора можно охарактеризовать кортежем *State*, который включает номер текущего режима \mathbf{Mode}_{active} , множество активированных проверок \mathbf{Checks}_{on} , множество активированных переходов $\mathbf{Transitions}_{on}$ и состоянием таблицы данных *Data*:

$$\mathbf{State} = \langle \mathbf{Mode}_{active}, \mathbf{Checks}_{on}, \mathbf{Transitions}_{on}, \mathbf{Data} \rangle,$$

где $\mathbf{Mode}_{active} \in \mathbf{Modes}_{test}$, $\mathbf{Checks}_{on} \subseteq \mathbf{Checks}_{test}$,

$$\mathbf{Transitions}_{on} \subseteq \mathbf{Transitions}_{test}. \quad (13)$$

В этом случае $\mathbf{Transitions}_{on}$ определяет множество активированных переходов, которые определены для текущего режима и позволяют изменить активный режим работы, а множество $\mathbf{Transitions}_{test}$ определяет все возможные условия переходов для текущего испытания.

Таблица данных представляет собой множество **Data**, которое объединяет всю регистрируемую во время проведения испытания информацию, в том числе множество входных сигналов **Sub_{in}**, выходных **Sub_{out}** и накапливаемого множества оценок требуемых характеристик объекта испытаний **Estimations**:

$$\mathbf{Data} = \mathbf{Sub}_{in} \cup \mathbf{Sub}_{out} \cup \mathbf{Estimations}. \quad (14)$$

Кроме этого, таблица данных активно взаимодействует с процессором ввода/вывода и выполняет функцию оповещения потребителей информации об изменениях в данных.

Любое действие начинается с установки активного режима и, несмотря на то, что объект испытаний может состоять из нескольких устройств, активный режим для всей системы может быть только один, следовательно, функциональное преобразование, выполняемое множеством всех режимов эквивалентно преобразованию активного режима:

$$F_{mode} : \text{Mode}_{active} \times \mathbf{T} \rightarrow \langle \mathbf{Sub}_{in}, \mathbf{Checks}_{on}, \mathbf{Transitions}_{on}, \mathbf{Messages}_{mode} \rangle. \quad (15)$$

Результатом активации режима является выдача стартовой последовательности команд на вход объекта испытаний $\text{Mode}_{active} \times \mathbf{T} \rightarrow \mathbf{Sub}_{in}$, активации/деактивации набора проверок, заданных множеством $\text{Mode}_{active} \times \mathbf{T} \rightarrow \mathbf{Checks}_{on}$, а так же активации переходов и постоянного отслеживания условий их выполнения, которые определены в $\text{Mode}_{active} \rightarrow \mathbf{Transition}_{on}$ и выдачей сообщений $\text{Mode}_{active} \times \mathbf{T} \rightarrow \mathbf{Messages}_{mode}$.

Кроме режимов на объект испытаний оказывают воздействие активированные проверки, которые могут выполняться параллельно и независимо друг от друга, синхронизируясь при необходимости через таблицу данных **Data**.

При этом каждая проверка представляет собой зависимую от времени систему из активных воздействий на объект испытаний $\mathbf{T} \rightarrow \mathbf{Sub}_{in}$ и оценки результата проверки $\mathbf{T} \times \mathbf{Data} \rightarrow \mathbf{Estimation}_i$ и **Estimations** $\times \mathbf{T} \rightarrow \mathbf{Messages}_{check}$ в зависимости от получаемой информации:

$$F_{check} : \langle \mathbf{Checks}_{on} \times \mathbf{T}, \mathbf{Data} \times \mathbf{T} \rangle \rightarrow \langle \mathbf{Sub}_{in}, \mathbf{Estimations}, \mathbf{Messages}_{check} \rangle. \quad (16)$$

Аналогично проверкам происходит параллельное отслеживание условий в данных испытания. В случае выполнения одного из условий происходит блокировка проверки других условий и на основе полученной информации выполняется переход на новый режим:

$$F_{transition} : \langle \mathbf{Transitions}_{on} \times \mathbf{T}, \mathbf{Data} \times \mathbf{T} \rangle \rightarrow \langle \text{Mode}_{active}, \mathbf{Messages}_{transition} \rangle. \quad (17)$$

3. Результаты практического применения

В процессе итеративной разработки приведенной выше модели автоматизированной системы испытаний сложных технических систем и решения практических задач были разработаны программные комплексы для проведения испытаний подсистемы энергоснабжения для спутников MC2-8 и Egyptsat-1.

Это позволило успешно провести следующие виды испытаний системы энергоснабжения: лабораторные, автономные стендовые, комплексные стендовые, производственные, полигонные.

При этом некоторые сценарии технологического процесса испытаний содержат более 50 тыс. действий в виде команд управления и проверок параметров испытуемой системы.

Литература

1. Баженов В.И., Иванов С.А., Наговкин В.Ф. Об оптимальном планировании экспериментальной отработки ЛА // Труды ОНЧ. – М.: Наука, 1980. – С.38-43.
2. Лачинов В.М., Поляков А.О. Информодинамика или Путь к Миру открытых систем. – СПб.: СПбГТУ, 1999. – 280 с.
3. Кузин С.Г. Конструирование унифицированных компонентов прикладной программы на основании ролевого графа // Вестник ННГУ. – Н. Новгород: ННГУ, 1999. – Вып. 1 (20). – С. 226-237.

Поступила в редакцию 20.02.2007

Рецензент: д-р техн. наук, проф. Э.Г. Петров, Харьковский национальный университет радиоэлектроники, Харьков.