

Компонентная модель программного обеспечения для испытаний бортовых систем космического аппарата

Игорь Туркин, Павел Лучшев, Иван Перекопский, Игорь Мосиенко

В статье изложены проблемные вопросы, свойственные автоматизации испытаний сложных технических систем. Рассматриваются задачи представления и обработки информации о технологических процессах испытаний такого рода систем. Приведены апробированные на практике решения указанных задач.

ВВЕДЕНИЕ

Современный космический аппарат (КА) средней сложности (рис. 1) включает в себя 20...25 систем различных наименований и назначений (системы управления, телеметрии, ориентации, терморегулирования, антенно-фидерная и др.), которые обеспечиваются электроэнергией от системы электроснабжения (СЭС) [1]. Главной задачей СЭС является поддержание необходимого уровня потребляемых электрических мощностей по определённой программе в течение заданного периода эксплуатации [2]. Система электроснабжения обладает полным набором признаков, позволяю-

щих отнести её к классу сложных технических систем. В состав солнечной СЭС входят следующие устройства: батарея фотоэлектрическая (БФ), батарея химическая (БХ), аппаратура управления, регулирования и контроля (АУРК). Такая СЭС может нормально функционировать только при условии информационного и энергетического взаимодействия с другими системами КА: ориентации, командной, телеметрической, управления, терморегулирования. Её главные электроэнергетические показатели зависят от характеристик внешней среды, с которой СЭС находится в непрерывно-циклическом энергетическом

обмене. БФ преобразует энергию солнечного излучения в электрическую, которая в зависимости от режима потребления бортовыми системами может накапливаться в БХ. Избыточная часть энергии сбрасывается в космическое пространство.

В процессе испытаний систем КА тщательно проверяются не только их конструкции, но и логика работы, а также комплексы, образующиеся при взаимодействии исследуемой системы с внешней космической средой, смежными системами КА, центром управления полётами. Особенно трудоёмкими и продолжительными являются исследовательские и экспериментально-отрабочные испытания. При этом объектом изучения и проверок является не только испытываемая система, но и контрольно-измерительная, контрольно-проверочная аппаратура (КИА и КПА), технологические процессы испытаний и проверок, эксплуатационная документация.

Высокий уровень компьютеризации объекта проверок (и в бортовых системах, и в технологической проверочной аппаратуре) используются микроЭВМ для реализации внутренней логики работы, при этом дополнительно в бортовой аппаратуре эти микроЭВМ резервированы для обеспечения необходимой надёжности) требует применения специальных методов испытаний, которые во многом схожи с мето-

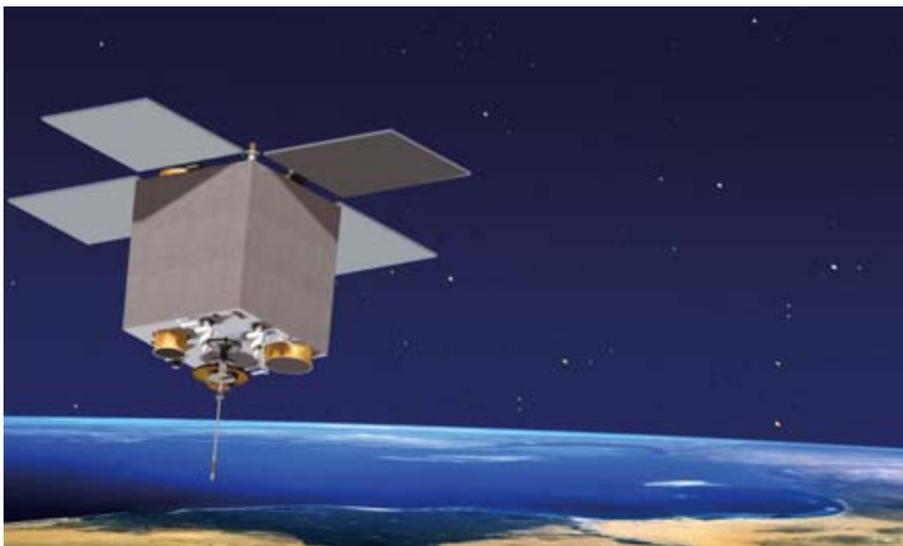
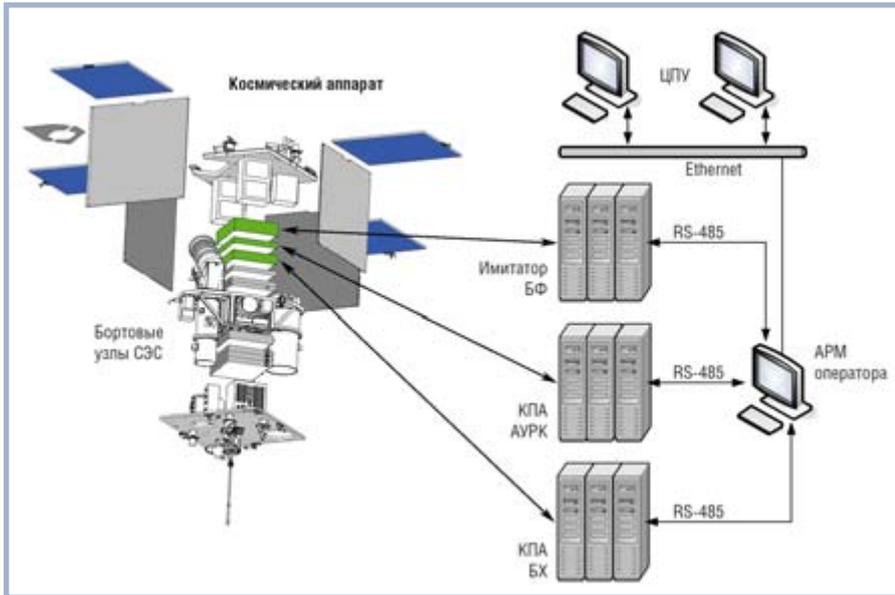


Рис. 1. Космический аппарат EGYPTSAT-1 (разработка ГKB «Южное», г. Днепропетровск)



Условные обозначения:

СЭС — система энергоснабжения; БФ — батарея фотоэлектрическая; КПА — контрольно-проверочная аппаратура; АУРК — аппаратура управления, регулирования и контроля; БХ — батарея химическая; ЦПУ — центральный пульт управления.

Рис. 2. Схема взаимодействия аппаратных средств во время комплексных испытаний

дами тестирования, верификации и валидации, применяемыми в отношении программных средств. Вместе с тем и в периодической печати, и в фундаментальной литературе вопросы создания эффективных программных средств для автоматизации испытаний аппаратуры, содержащей в своём составе развитую компьютеризованную составляющую, практически не рассмотрены. Авторы данной статьи предлагают апробированное решение задачи представления и обработки информации о технологических процессах испытаний сложных технических систем.

ОБЪЕКТ ИСПЫТАНИЙ

Технологические процессы испытаний реализуются в системе, которая погружена во внешнюю среду, основными составляющими которой являются человек-оператор, объект испытаний, его КИА и КПА.

На рис. 2 приведена схема взаимодействия аппаратных средств во время комплексных испытаний системы СЭС КА. Бортовые узлы СЭС связаны с имитатором БФ и КПА БХ (разработчик — Институт физики полупроводников НАН Украины), а также с КПА АУРК (разработчик — НПП «Хартрон-ЮКОМ»). Имитатор и КПА через двухпортовую интерфейсную плату с гальванической изоляцией PCI-1602B (фирма Advantech) соединены последовательными полевыми

шинами RS-485 с ПЭВМ в промышленном исполнении (панельный компьютер PPC-153T фирмы Advantech — табл. 1) из состава АРМ оператора (рис. 3). КПА АУРК построена на основе модулей, выполненных в формате MicroPC:

- микроконтроллера CPU188-5MX фирмы Fastwel для реализации алгоритмов АУРК по командам АРМ оператора;
- универсального модуля ввода-вывода UNIO96-1 (Fastwel) для приёма 39 и выдачи 47 дискретных сигналов;
- 8-канального модуля последовательной связи 5558 фирмы Octagon Systems для расширения количества каналов последовательной связи (скорость обмена программируется от 300 до 115200 бит/с);
- двух модулей аналогового ввода-вывода с гальванической изоляцией



Рис. 3. КПА АУРК, КПА БХ, имитатор БХ и ПЭВМ для испытаний СЭС КА



Рис. 4. Каркас с модулями MicroPC в составе аппаратуры КПА АУРК

- AI16-5A-1 (Fastwel) для преобразования 24 аналоговых сигналов диапазона 0...10 В в 13-разрядный код (модули подключены к шине параллельно) и формирования 4 выходных аналоговых сигналов;
- трёх модулей аналогового вывода с гальванической изоляцией AO16-V8 (Fastwel) для формирования 18 выходных аналоговых сигналов.

Все эти модули установлены в объединительный монтажный каркас ICC19101 (Fastwel) с источником электропитания от первичной сети 5105 (Octagon Systems). Каркас с модулями MicroPC в составе аппаратуры КПА АУРК показан на рис. 4.

Таблица 1

Основные характеристики панельного компьютера Advantech PPC-153T

Процессор	Intel Pentium III 850 МГц
ОЗУ	256 Мбайт SDRAM
Дисплей	15" TFT
Накопитель	IDE HDD Fujitsu MHT2060 AT
Ethernet	10/100Base-T
Расширение	Плата PCI-1602B
Установленное программное обеспечение	Windows 2000 с программой Hyper Terminal
	OPC-сервер фирмы Fastwel для сети PLCNet
	OPC-сервер UniOPC фирмы Fastwel
	Advantech PCI-1602B Driver for Windows 2000
	ПО разработчиков аппаратуры и др.

Автономной отработкой СЭС проверяются и подтверждаются её энергетические и командно-информационные характеристики, которые проявляются только в процессе взаимодействия составных частей системы между собой. До автономной отработки составные части СЭС проходят собственный цикл испытаний, в частности, проводятся такие испытания, которые затруднительно (иногда и невозможно) либо экономически нецелесообразно проводить на СЭС в собранном виде.

Во время комплексных испытаний в систему добавляются дополнительные командно-информационные связи с центральным пультом управления (ЦПУ), реализованные через стандартную сеть Ethernet с помощью протоколов TCP, FTP. В этом случае для СЭС главными становятся задачи снабжения бортовых систем КА электроэнергией, а программное обеспечение (ПО) испытаний СЭС обеспечивает контроль текущего состояния системы и проверку «связок» (взаимовлияния, взаимодействия) с другими бортовыми системами. ПО испытаний СЭС становится ведомым по отношению к ПО ЦПУ, подчиняясь его командам и периодически по запросам обмениваясь с ним информацией.

Постановка задачи

Используемые в процессе испытаний программные средства относятся к классу систем реального времени, то есть систем, в которых правильность работы зависит не только от логических результатов вычислений, но и от времени, затраченного на получение этих результатов. В качестве количественных показателей, характеризующих сложность задачи автоматизации испытаний систем КА, отметим следующие:

- на канальном уровне скорость обмена информацией составляет единицы кбайт/с;
- количество сигналов и команд управления — порядка $10^2 \dots 10^3$;
- на мнемосхеме одновременно должно быть визуализировано текущее состояние 30–80 сигналов, для которых дополнительно требуются графические средства анализа их предыстории;
- сообщения о событиях и тревогах, которые должны быть зарегистрированы для последующего анализа, формируются с частотой $10^{-2} \dots 10^2$ 1/с;

- требуемая периодичность циклического выполнения алгоритмов диагностирования составляет 0,1 с;
- количество алгоритмов диагностирования — десятки;
- количество технологических процессов и форм протоколов испытаний — десятки;
- описание типового технологического процесса включает несколько тысяч действий (наборов операций), выполняемых в последовательностях, находящихся в сложных причинно-следственных связях.

Все перечисленные показатели, кроме последнего, обеспечиваются функциональными возможностями большинства известных SCADA-систем. Такие системы, как правило, имеют встроенные языки высокого уровня: чаще всего это языки программирования контроллеров (стандарт IEC 61131-3) либо языки, подобные VisualBasic, позволяющие генерировать адекватную реакцию на события, связанные с изменением значения переменной, с выполнением некоторого логического условия и т.д. Стандарт IEC 61131-3 описывает три графических языка: «Диаграмма цепей» (LD), «Диаграмма функциональных блоков» (FBD) и «Схема последовательных функций» (SFC) — и два текстовых: «Список команд» (IL) и «Структурированный текст» (ST). При этом части прикладной программы могут быть запрограммированы на любом языке и скомпонованы в единую исполняемую программу.

Вместе с тем функциональности SCADA-систем часто оказывается недостаточно для эффективной автоматизации технологических процессов испытаний систем КА по причинам, которые приводятся далее.

- Разработка комплектующих бортовой системы, их наземной КИИА и КПА, определение технологических режимов испытаний ведутся практически параллельно, причём аппаратура и ПО испытаний должны быть введены в эксплуатацию одновременно. В ходе этой разработки общей продолжительностью 2–3 года постепенно конкретизируется структура и функциональность объекта управления, согласовываются его интерфейсы взаимодействия с ПО испытаний. При этом требования к программным средствам не становятся полностью определёнными после завершения вы-

страивания цепочки «объект испытаний, его интерфейсы, технологические процессы испытаний», так как звенья этой цепочки проверяются и модифицируются в ходе экспериментальной отработки. Следствием этого является требование избыточной функциональности ПО для реализации незапланированных расширений первоначальной спецификации.

- Реализуемые программными средствами технологические процессы испытаний являются достаточно разнообразными по своей сути и объёмными по содержанию. Разумеется, можно на основе языков высокого уровня SCADA-систем описать последовательности из нескольких тысяч элементарных действий, но экспериментальная отработка этих последовательностей потребует постоянного участия разработчика ПО в экспериментальной отработке бортовой системы, и во время этих многомесячных работ многократно придётся искать ответ на вопрос: «Кто виноват в том, что система работает неправильно: разработчики комплектующих бортовой системы, которые не выполнили требования технического задания, разработчик системы, сформулировавший эти требования недостаточно чётко либо неправильно задавший технологические режимы испытаний, разработчик программных средств, допустивший ошибки при алгоритмизации технологического процесса испытаний?».

- Проблемную ситуацию, описанную в предыдущем пункте, в некоторой степени могла бы предотвратить предварительная отработка технологического процесса испытаний на имитационной математической модели, тем более что и задача экономного расходования ресурса объекта испытаний весьма актуальна, но и в этой части SCADA-системы не предлагают готовых решений.

Исходя из анализа возможностей SCADA-систем и причин их недостаточной функциональности в разрешении определённых проблем, возникающих в ходе испытаний, можно перечислить основные требования, которым должно соответствовать разрабатываемое ПО для автоматизации испытаний систем КА.

- Описание технологического процесса (ТП) испытаний необходимо

представить формальной спецификацией (сценарием) испытаний, которая должна:

- быть простой для восприятия и понимания инженером-технологом;
- поддаваться синтаксическому и семантическому контролю, в том числе в автоматическом режиме;
- допускать использование абстракций, позволяющих представить описание ТП в виде набора взаимосвязанных элементов, каждый из которых может быть рассмотрен независимо от остальных (в противном случае такое описание может стать «необозримым» для инженера-технолога из-за своей объёмности).
- Документируемость ТП должна обеспечиваться за счёт возможности автоматического построения:
 - описательных графических иллюстраций, позволяющих оператору и разработчику ТП получить обобщённое представление о сути выполняемых действий;
 - исчерпывающего текстового описания всех действий, подлежащих выполнению.
- Интерпретатор сценария должен удовлетворять жёстким временным ограничениям в процессе автоматической работы при испытаниях.
- Повторно используемыми должны быть как программные средства для работы со сценарием испытаний, так и собственно описания ТП испытаний; при этом разработанные программные средства должны быть совместимыми с наиболее распространёнными SCADA-системами, например GENESIS32.
- Необходимо обеспечить возможность работы с имитационной моделью объекта испытаний и внешней среды. Это полезно не только для тестирования системы, но и для уточнения и развития требований, изучения свойств системы, понимания логики взаимодействия с внешней средой и т.д. Обращение к модели необходимо как на ранних этапах разработки, когда объект испытаний, КИА и КПА ещё отсутствуют, так и на поздних, когда актуальными становятся вопросы сохранения ресурса, экономии времени и трудозатрат. Важной является реализация способности имитационной модели работать как в ускоренном, так и в замедленном модельном времени.

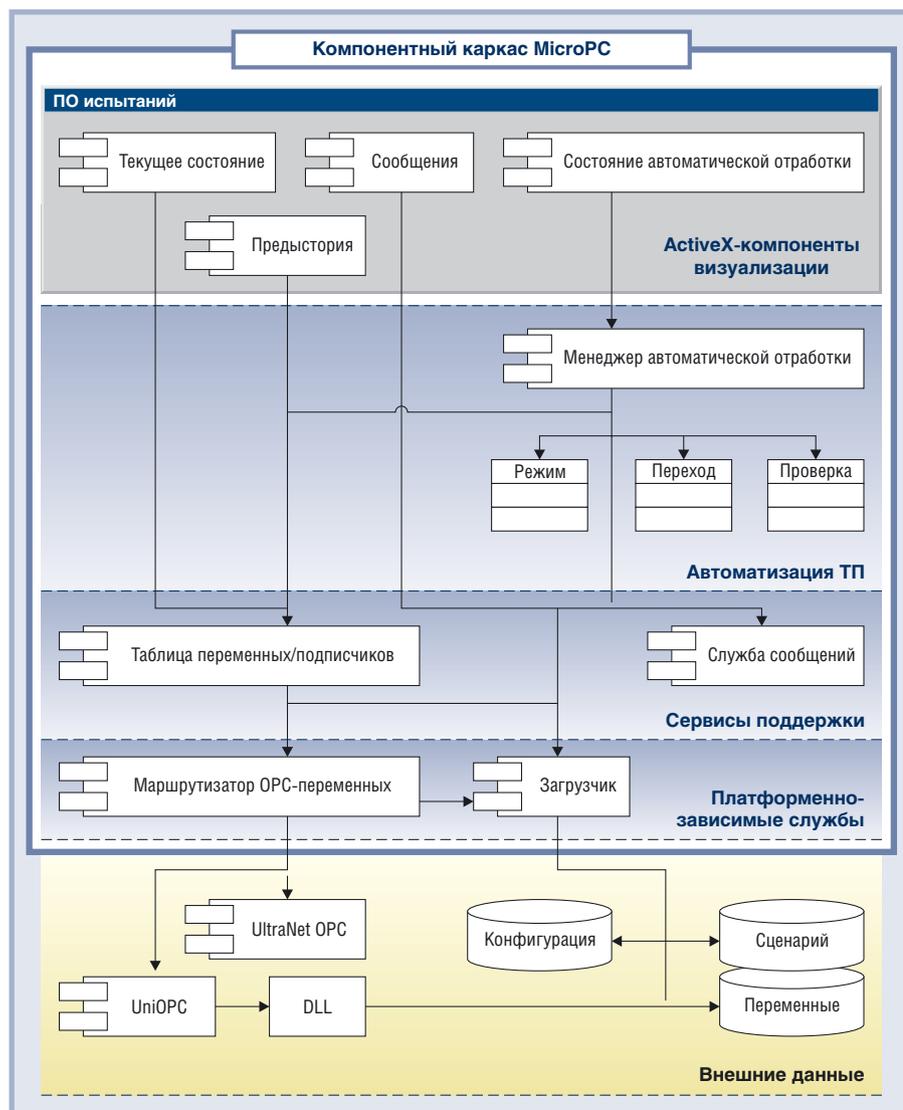


Рис. 5. Архитектурная модель ПО для испытаний

АРХИТЕКТУРНАЯ МОДЕЛЬ ПРОГРАММНЫХ СРЕДСТВ

Основой архитектурной модели (рис. 5) является технология COM (Component Object Model — модель составных объектов), которая определяет общую схему взаимодействия компонентов программного обеспечения в среде Windows и предоставляет стандартную инфраструктуру, позволяющую объектам обмениваться данными и функциями между прикладными программами.

Уровень внешних данных

На уровне внешних данных присутствует информация двух видов: статическая и динамическая.

Статическая (конфигурационная) информация в зависимости от выбранного технологического процесса испытаний описывает состав физических переменных, необходимых для работы, собственно содержание ТП (сценария), режимы визуализации.

Источниками динамической информации являются программные серверы PLCNet из состава UltraNet OPC и UniOPC (Fastwel), использующие технологию связывания и внедрения объектов для систем промышленной автоматизации OPC (OLE for Process Control). PLCNet OPC-сервер обеспечивает связь с КПА АУРК по протоколу сети PLCNet, а универсальный UniOPC-сервер — с имитатором БФ и КПА БХ по протоколу полевой шины ModBus.

UniOPC-сервер предназначен для поддержки быстрой разработки OPC-серверов различных устройств, в том числе нестандартных. UniOPC-сервер функционирует в паре с динамической библиотекой (DLL), разрабатываемой пользователем. Эта библиотека содержит ту часть кода программы, которая специфична для данного устройства, то есть осуществляет считывание и запись данных, а также периодическое обновление массивов значе-

ний переменных (тегов), видимых OPC-серверу. OPC-сервер обеспечивает «публикацию» этих тегов, открывая их для доступа со стороны клиентов. В ПО для испытаний систем КА динамическая библиотека в зависимости от режима работы выполняет двойную функцию:

- в штатном (нормальном) режиме обеспечивает обмен данными и командами с имитатором БФ и КПА БХ через последовательный порт;
- в режиме «Модель» для отладки технологических процессов проведения испытаний имитирует работу системы, её КИА и КПА, поддерживая, в том числе, и возможность работы в режиме модельного времени (в данном режиме библиотека дополнительно воспроизводит пространство переменных PLCNet OPC-сервера, а перенаправление вызовов осуществляется маршрутизацией OPC-переменных).

Конфигурирование динамической библиотеки осуществляется в момент старта программы путём считывания информации из базы данных, содержащей информацию о физических переменных, правилах формирования запросов к внешней аппаратуре для их получения, способах их нормирования и масштабирования.

Уровень платформенно-зависимых служб

На уровне платформенно-зависимых служб находятся компоненты, основное назначение которых — ретранслировать внешние данные во внутреннее адресное пространство, структуры и объекты программы.

Загрузчик работает в моменты изменения оператором технологического процесса испытаний (загрузки нового сценария), обновляя конфигурационную информацию для всех вышестоящих программных компонентов на уровнях сервиса поддержки, автоматизации ТП и визуализации.

Маршрутизатор поддерживает синхронные и асинхронные методы взаимодействия с внепроцессными OPC-серверами, решая попутно ряд вспомогательных задач, например:

- в зависимости от выбранного режима (штатный или «Модель») выбирает OPC-сервер, содержащий необходимые переменные;
- вместо внутренних для программы имён переменных осуществляет подстановку их внешних псевдонимов,

под которыми эти переменные зарегистрированы в ОРС-серверах.

Сервисы поддержки логики программы

Хранилище данных (таблица переменные/подписчики) обеспечивает логически непротиворечивую и надёжную работу алгоритмов, состав которых может изменяться во время исполнения основной программы. Хранилище данных обеспечивает целостность информации о множестве объектов двух ключевых базовых классов: «Переменная» и «Подписчик». «Переменные» обеспечивают хранение информации о состоянии внешних (физических) сигналов, внутренних общепрограммных переменных, необходимых для синхронизации основных программных компонентов. «Подписчики» отвечают за логическую обработку информации, их работа инициируется изменением «Переменных» в хранилище данных.

Именно на основе классов «Переменная» и «Подписчик» в программе решена проблема разбиения системы на относительно простые, параллельно работающие, независимые компо-

ненты. Передача информации между ними осуществляется по единому механизму.

- Объект «Переменная» в случае изменения его значения и обнаружения хотя бы одного «Подписчика» на оповещение об этом изменении вызывает соответствующие методы обработки информации.
- «Подписчик»-преобразователь информации выполняет некоторые вычисления, результаты которых может опубликовать как новые значения «Переменных» в хранилище данных. В этом случае рекурсивно повторяется пункт 1 и возможна последовательная обработка информации цепочкой «Подписчиков».
- Рекурсия завершается работой «Подписчиков», не публикующих результаты своей работы в хранилище данных, ответственных, например, за протоколирование результатов работы в файле либо за визуализацию текущего состояния системы или предыстории процесса.

После загрузки нового сценария состав «Переменных», отображающих состояние внешних (физических) сигналов, остаётся неизменным, а вари-

ативен состав «Подписчиков» и внутренних «Переменных».

Служба сообщений решает задачи централизованного хранения, обработки и архивирования сообщений о событиях и тревогах.

Уровень автоматической отработки технологического процесса испытаний

Автоматическая отработка ТП испытаний осуществляется менеджером автоматической работы, который после загрузки файла сценария испытаний и соответствующей команды оператора, запускающей ТП на выполнение, формирует управляющие воздействия (команды) на систему и КПА, выполняет необходимые диагностические проверки, а также через службу сообщений регистрирует происходящие события и выполняемые действия.

В основе описания ТП испытаний, а следовательно, и в архитектуре программно реализованных классов менеджера автоматической работы находятся такие понятия, как *режим, переход, последовательность действий, проверка*.

Режим — это текущее состояние выполняемого технологического процесса. В начале любого из режимов менеджер обрабатывает стартовую последовательность действий, после чего изменяет состояние активных проверок. В этом состоянии менеджер будет находиться до тех пор, пока не будут выполнены условия перехода на иной режим.

Переход из одного режима в другой инициируется, если будут выполнены все условия соответствующего списка, заранее приписанного данному переходу.

Последовательность действий (в частном случае может быть пустой) выполняется в начале нового режима и в процессе проверки. Она представляет собой упорядоченный список элементарных действий, в том числе пауз для отсчёта интервалов времени, сообщений оператору, изменений в таблице переменных (если переменной во внутренней таблице соответствует переменная в пространстве имён OPC-сервера, то фактически это изменение приведёт к передаче управляющего воздействия во внешнее устройство).

Проверка расширяет функциональность последовательности. Проверки работают параллельно, ограничение количества одновременно выполняемых проверок отсутствует. Среди возможных элементарных действий в проверке дополнительно реализовано действие «Контроль списка условий». В отличие от перехода, где контроль списка условий осуществляется в следующем режиме, в проверке такое действие выполняется только в моменты времени, предписанные последовательностью. По итогам этого контроля

вычисляется результат выполнения проверки, который автоматически помещается в хранилище данных как новое значение уникальной переменной, соответствующей определённой проверке; при этом автоматически генерируется сообщение оператору, если таковое предусмотрено описанием проверки.

Функциональности описанного механизма оказалось более чем достаточно для практической реализации ПО автоматизации технологических процессов автономных и комплексных испытаний ЭЭС. Была также реализована, но на практике оказалась невостребованной дополнительная поддержка в проверках программных фрагментов, написанных на VBScript. Эта поддержка основана на использовании ActiveX-интерпретатора Microsoft Script Control, который входит в комплект поставки Windows, начиная с Windows 98 (2000), а для предыдущих версий доступен в виде свободно распространяемого дистрибутива.

Уровень ActiveX-компонентов визуализации

Элементы управления ActiveX — это объекты, которые представляют собой внутрипроцессные COM-серверы. Большинство SCADA-систем имеют в своём составе контейнеры, которые позволяют загружать ActiveX-компоненты и получать уведомления о произошедших событиях. В этом случае любые объекты ActiveX могут быть загружены в среду разработки и использованы при создании прикладных программ. Управление ими осуществляется с помощью данных, методов и событийных функ-

ций, свойственных выбранному объекту.

Применение технологии ActiveX помимо решения задачи взаимодействия с пользователем обеспечило переносимость разработки и совместимость со SCADA-системами.

На рис. 6, 7 приведены примеры использования ActiveX-элементов в мнемосхемах, обеспечивающих визуализацию текущего состояния объекта испытаний и режимов автоматической обработки ТП испытаний.

РЕЗУЛЬТАТЫ

В основе реализованного подхода лежит идея полного переноса описания выполняемых действий, необходимых для проведения ТП испытаний, из программной реализации во внешний источник информации (сценарий испытаний), представленный в формате реляционной базы данных. Чтобы осуществить эту идею, потребовалось помимо собственно реляционного представления найти текстово-графические формы описания ТП, ориентированные на человека (технолога и/или оператора), и решить задачу, какими программными конструкциями обеспечить требования реального времени при автоматической обработке сценария.

На наш взгляд, именно решение о ключевой роли человеко-ориентированного представления информации в общей цепочке её передачи «технолог — база данных — реальное время — оператор» позволило получить главный эффект — уменьшить затраты на подготовку и проведение ТП испытаний.

Исторически к предлагаемой реализации мы пришли далеко не сразу.

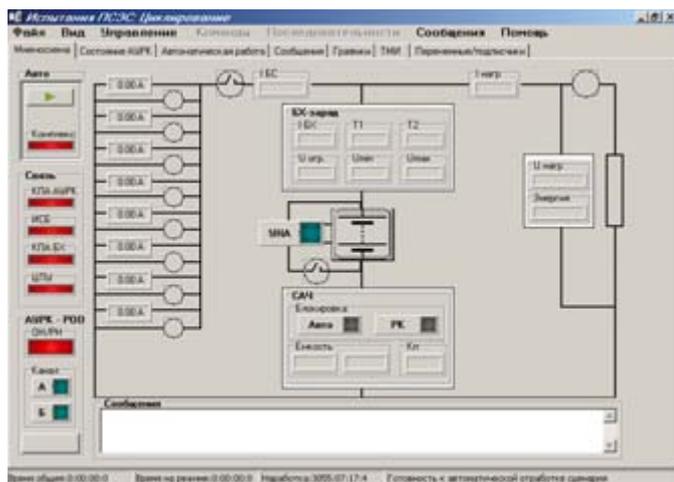


Рис. 6. Мнемосхема для отображения текущего состояния объекта испытаний

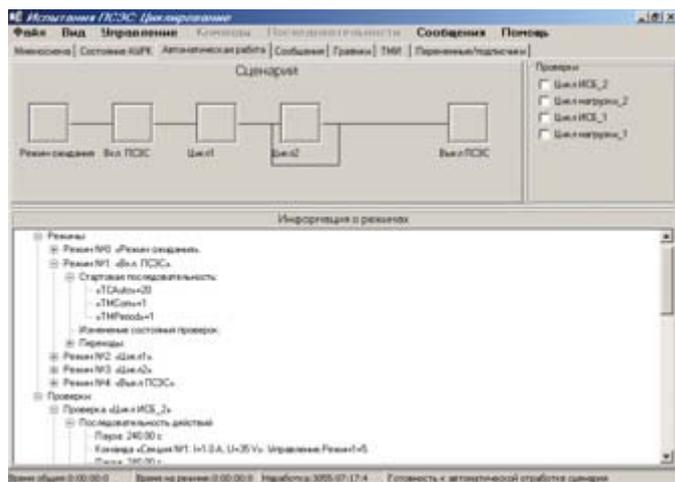


Рис. 7. Мнемосхема автоматической обработки технологического процесса

Вначале была попытка ограничиться только текстовым представлением на основе специально сконструированного проблемно-ориентированного языка, затем — описывать технологический процесс FBD-диаграммами. После работы с прототипами и соответствующих неудачных попыток приучить заказчика «самовыражаться» с помощью данных средств стало понятно, что уровень сложности задачи именно такой, что только путём взаимодополняющего совмещения графического и текстового описаний ТП испытаний станет возможным:

- создать модель ТП испытаний в виде иерархической структуры таким образом, чтобы для работы с любой из областей такой структуры было достаточно лишь нескольких информационных компонентов;
- при необходимости спрятать несущественные детали и выделить принципиально важные ракурсы рассмотрения;
- обеспечить хорошее понимание, строгость, простоту обработки и т.д.;
- реализовать средства документирования ТП.

Таким образом, в специфической и вместе с тем достаточно разнообразной области задач автоматизации ТП испытаний сложных технических систем предложены оригинальные решения, которые позволили успешно справиться с проблемами, перечисленными в начале статьи.

ЗАКЛЮЧЕНИЕ

Предлагаемый подход был практически апробирован и внедрён в рабо-

ты Государственного КБ «Южное» (г. Днепропетровск). Спецификация ТП испытаний в виде сценариев и разработанные программные средства для редактирования этих сценариев позволили инженеру-технологу проверять и по сути «на лету» корректировать операции ТП, что объективно в немалой степени способствовало повышению эффективности экспериментальной отработки системы.

Созданное ПО обеспечило проведение всех видов испытаний систем электроснабжения двух типов КА. Первым из них был микроспутник дистанционного зондирования Земли КС5МФ2, который, будучи экспериментальным изделием, служил в основном для отработки универсальной космической платформы. Этот спутник при массе 66 кг оснащён малогабаритной бортовой телевизионной камерой видимого диапазона с передачей информации на Землю по радиоканалу. Запуск осуществлён в декабре 2004 года совместно со спутником «Січ-1М» ракетой-носителем «Циклон-3» с космодрома Плесецк. Второй КА — малогабаритный спутник высокого разрешения Egyptsat-1 массой 135 кг. Его комплекс оптико-электронной аппаратуры предназначен для решения ряда практических и научных задач регионального и локального уровня по мониторингу кризисных ситуаций, растительных и почвенных покровов суши, созданию цифровых карт местности, управлению ресурсами и планированию в урбанизированных и прибрежных зонах и др. Запуск планируется в начале

2006 года ракетой-носителем «Днепр-1» с космодрома Байконур.

Некоторые практически важные вопросы, например использование в качестве графического редактора пакета векторной графики Visio корпорации Microsoft в виде СОМ-сервера, интегрированного в состав разработанного редактора сценариев испытаний, либо организация информации о сценарии в базе данных, остались не освещёнными в данной статье, прежде всего потому, что эти вопросы достойны самостоятельного рассмотрения. ●

ЛИТЕРАТУРА

1. Технология сборки и испытаний космических аппаратов: Учебник для высших учебных заведений / И.Т. Белякова, И.А. Зернов, Е.Г. Антонов и др. / Под общ. ред. И.Т. Белякова и И.А. Зернова. — М.: Машиностроение, 1990. — 352 с.
2. Солнечные энергосистемы космических аппаратов. Физическое и математическое моделирование / К.В. Безручко, Н.В. Белан, И.Б. Туркин / Под ред. акад. НАН Украины С.Н. Конюхова. — Харьков: Гос. аэрокосмический ун-т «Харьковский авиационный институт», 2000. — 516 с.

**Авторы — сотрудники
Национального аэрокосмического
университета им. Н.Е. Жуковского
«ХАИ»,
телефон: (+38-057) 707-4574,
факс: (+38-057) 707-4402,
и ГKB «Южное»,
телефон: (+38-0562) 92-1713,
факс: (+38-056) 770-0430**