

УДК [004.413+004.434]:629.78.064.018

И.Б. ТУРКИН, П.А. ЛУЧШЕВ

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

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ИСПЫТАНИЙ СИСТЕМ ЭЛЕКТРОСНАБЖЕНИЯ КОСМИЧЕСКИХ АППАРАТОВ

Рассмотрены проблемы испытаний космической техники как сложной технической системы. Определены основные требования к программному обеспечению для автоматизации испытаний подсистем космического аппарата. Предложен языково-ориентированный подход и разработана архитектура адаптируемого программного обеспечения автоматизации испытаний. Применение предлагаемого подхода, а также адаптируемого программного обеспечения позволяет эффективно автоматизировать испытания различных сложных технических систем. Рассмотрены примеры использования программного обеспечения для автоматизации испытаний подсистемы электроснабжения космических аппаратов.

Ключевые слова: автоматизация испытаний, адаптируемое программное обеспечение, языково-ориентированный подход, испытания подсистемы электроснабжения.

Введение

Космическая промышленность Украины, а это десятки крупных и очень крупных предприятий, конструкторских бюро и научно-исследовательских институтов, имеет свою шестидесятилетнюю историю, в которой отмечены даты и события, успехи и неудачи.

На момент распада СССР ракетно-космическая отрасль Украины была признанным мировым лидером в создании боевой ракетной техники, ракет-носителей (РН) и космических аппаратов (КА). После приобретения независимости положение отрасли радикально изменилось. Исчезли большие оборонные заказы, началась ускоренная эрозия кооперационных связей между предприятиями и организациями Украины и стран СНГ [1].

Завершение работ над базовой космической платформой нового поколения в 2004 году было одним из наиболее важных результатов в разработке новых технологий, которые дали значительный толчок научно-техническому развитию космической техники и ознаменовали возврат Украины в плеяду космических держав мира.

КА «Сич-1М», предназначенный для дистанционного зондирования Земли, был запущен 24 декабря 2004 года в 13.20 с космодрома «Плесецк» (Россия) с помощью ракеты-носителя «Циклон-3» вместе с микроспутником КС5МФ2.

Тремя годами ранее Государственное конструкторское бюро «Южное» (г. Днепропетровск) победило в международном тендере, который проводился правительством Египта на создание первого

египетского спутника дистанционного зондирования Земли Egyptosat-1. Кроме Украины в тендере принимали участие Великобритания, Россия, Корея, Италия. Работы по созданию космического аппарата Egyptosat-1 выполнялись по контракту на проектирование, изготовление, запуск, обучение и передачу технологии для спутника Egyptosat-1 между NARSS (National Authority for Remote Sensing and Space Sciences), Египет и ГКБ "Южное".

Разработка системы электроснабжения спутника Egyptosat-1 представляла собой сложную техническую задачу, решение которой требовало создания компактной высокоэффективной энергоустановки малой мощности, отвечающей высоким предъявляемым требованиям. К числу узловых технических вопросов относилось:

- необходимость обеспечения длительного ресурса в аппарате негерметичного исполнения, превышающего ресурс базового спутника в 1.7 раза;
- достижение повышенных удельных характеристик, увеличения среднесуточной мощности почти в 2.7 раза.

Такое существенное возрастание количественных показателей практически исключало возможность заимствования элементов подсистемы электроснабжения базового спутника и требовало применения ряда новых схемно-конструктивных решений и технических средств. Кроме того, прогресс в области космической электроэнергетики последних лет заставлял обновить ряд традиционных проектно-технических подходов.

Успешный запуск спутника Egyptosat-1 массой около 150 кг на солнечно-синхронную орбиту со-

стоялся 17 апреля 2007 года конверсионной ракетой-носителем «Днепр-1» с космодрома Байконур. В отличие от предыдущих договоров этот старт для преподавателей и студентов кафедры инженерии программного обеспечения ХАИ был промежуточным этапом, а не завершением работ по созданию программного обеспечения (ПО) для испытаний системы электроснабжения. Впереди оставались работы по обучению египетских специалистов и вводу в эксплуатацию инженерной модели спутника в Каире.

Космический аппарат и его системы как объект испытаний

Современный космический аппарат (КА) средней сложности включает в себя два – три десятка систем различного наименования и назначения (системы управления, телеметрии, ориентации, терморегулирования, антенно-фидерная и др.), работа которых полностью зависит от электроэнергии, получаемой от системы электроснабжения. В структуре спутника выделяют два основных сегмента: платформу, которая обеспечивает функционирование КА, и полезную нагрузку – несколько целевых систем, ради которых собственно и создают спутник.

Научно-технический прогресс приводит к улучшению удельных энергомассовых и объемных показателей, повышению ресурса, надежности и живучести, увеличению мобильности и удобства эксплуатации КА [2]. Все это невозможно без внедрения информационных технологий, обеспечивающих интеллектуализацию управления, регулирования и контроля. Как следствие, возрастает интенсивность информационного взаимодействия между целевыми и вспомогательными системами, КА и наземными центрами управления полетов.

Испытания занимают важное место в процессе создания ракетно-космической техники. Сложность задачи во многом определяется большим количеством информационных параметров и сигналов управления. Системы автоматизации таких испытаний имеют сотни и тысячи параметров измерения и управления, характеризующих электрические, химические, тепловые, пневматические и гидравлические явления. Интенсивность информационных потоков может достигать несколько мегабайт в секунду. Динамика переходных процессов различных систем космического аппарата колеблется от долей секунд до суток. В некоторых случаях скорость принятия решений по управлению изделием должна быть мгновенной.

Для проведения испытаний КА и его систем используется наземный сегмент, в стендовые средства которого входят имитаторы внешних условий, контрольно-измерительная и контрольно-про-

верочная аппаратура (КИА и КПА), что позволяет проводить как автономные, так и комплексные проверки.

Значительная часть отработки на устойчивость к дестабилизирующим воздействиям проводится с помощью автономных испытаний составных частей систем КА, и их результаты распространяются на систему в целом. Испытания могут проводиться либо на всем устройстве, либо на его составляющих. Этапы отработки составных частей систем включают лабораторно-отрабочные испытания, конструкторско-доводочные, чистовые доводочные испытания.

Автономные испытания систем КА подразделяются на лабораторные и стендовые. Первые проводятся на масштабированных физических моделях либо на полномасштабных образцах. Автономные испытания (АИ) включают проверку на воздействие всех эксплуатационных факторов, оговоренных техническим заданием: механических, климатических, радиационных, электромагнитных, вакуума, невесомости и т.д. [3]. Автономной отработкой проверяются и подтверждаются те характеристики, которые не могут быть аддитивно получены из параметров составных частей либо проявляются только в процессе их взаимодействия между собой.

Объектами испытаний при автономной экспериментальной отработке являются как сами системы КА, так и комплекс контрольно-проверочной аппаратуры, полный комплект схемной и эксплуатационной документации.

Комплексные испытания систем проводится в составе КА и охватывает все виды характеристик, указанные в [4]. В процессе комплексных испытаний в состав объектов контроля входят разнородные подсистемы, проектирование и разработку КИА и КПА для которых выполняют организации, специализирующиеся в соответствующих областях. Для интеграции нескольких подсистем в единый комплекс необходимо для каждой подсистемы разработать, согласовать и реализовать протокол взаимодействия как на аппаратном, так и на информационном уровне.

Цель **производственных испытаний**, проходящих при изготовлении штатных КА, является проверка соответствия основных характеристик систем данного комплекта требованиям документации на КА, в том числе подтверждение бесперебойного обеспечения потребителей электроэнергией требуемого качества и функционального взаимодействия и электромагнитной совместимости с данным комплексом бортового аппаратного комплекса КА.

К основным задачам **полигонных испытаний** относятся: проверка функционирования систем после хранения и транспортировки, отработка техно-

логии подготовки систем к работе в натуральных условиях, оценка качества и содержания эксплуатационной документации.

Летные испытания имеют целью проверку в условиях орбитального полета КА соответствия основных тактико-технических и эксплуатационных характеристик систем и их составных частей требованиям технического задания на разработку.

Испытания космической техники относятся к плохо формализованным и недостаточно структурированным динамическим процессам ввиду уникальности самого объекта испытаний, нестационарности процессов, нелинейности характера взаимосвязей, многоуровневой (иерархической) организации, многокритериальности решаемых задач.

Требования к программному обеспечению для автоматизации испытаний систем космических аппаратов

Разработка комплектующих бортовой системы, их наземной КИА и КПА, определение технологических режимов испытаний ведется практически параллельно, а введены в эксплуатацию аппаратура и программного обеспечения для автоматизации испытаний должны быть 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 определенными после завершения 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.

Из перечисленных особенностей следуют основные требования, которым должно соответствовать разрабатываемое программное обеспечение для автоматизации испытаний систем КА.

1. Описание технологического процесса (ТП) испытаний необходимо представить формальной спецификацией (сценарием) испытаний, которая должна:

- быть легкой для восприятия и прозрачной для понимания инженером-технологом, проектантом системы, что позволит повысить эффективность взаимодействия различных групп разработчиков;

- поддаваться синтаксическому и семантическому контролю, в том числе в автоматическом режиме;

- должна допускать использование иерархии абстракций, позволяющих представить описание ТП в виде набора взаимосвязанных элементов, каждый из которых может быть рассмотрен независимо от остальных.

2. Документируемость ТП должна обеспечиваться за счет возможности автоматического построения:

- описательных графических иллюстраций, позволяющих оператору и разработчику ТП получить обобщенное представление о сути выполняемых действий;

- исчерпывающего текстового описания всех действий, подлежащих выполнению.

3. Интерпретатор сценария должен удовлетворять жестким временным ограничениям во время автоматической работы при испытаниях.

4. Повторно используемыми должны быть как программные средства для работы со сценарием испытаний, так и собственно описания ТП испытаний.

5. Необходимо обеспечить возможность работы с имитационной моделью объекта испытаний и внешней среды. Такая модель полезна не только для тестирования системы, но и для уточнения и развития требований, изучения свойств системы, понимания логики взаимодействия с внешней средой и т.д. Модель необходима как на ранних этапах разработки, когда объект испытаний, контрольно-измерительная аппаратура ещё отсутствуют, так и на поздних, когда актуальными становятся вопросы сохранения ресурса и экономии времени и трудозатрат. Реализация имитационной модели должна предусматривать возможности ускорения и замедления модельного времени.

Языково-ориентированное проектирование адаптируемого программного обеспечения для автоматизации испытаний систем космических аппаратов

Программные системы в настоящее время стали столь большими и сложными, что для их производства требуется участие слаженных команд разработчиков различных специальностей и квалификаций. Вложенные в их производство и освоение средства должны окупаться, поэтому такие системы должны существовать и применяться долгие годы, развиваясь от версии к версии, претерпевая на своем жизненном пути множество изменений, улучшая существующие и добавляя новые функции, корректируя и устраняя дефекты и ошибки.

Длительный жизненный цикл предполагает способность программных систем адаптироваться

не только к изменению условий работы в новой среде, но и к изменяющимся требованиям пользователя.

Международные и отечественные стандарты, регламентирующие требования к качеству ПО (ISO/IEC 9126, серии ECSS-Q, СОУ-Н НКАУ 0012:2006 "Вимоги до якості програмного забезпечення програмно-технічних комплексів критичного призначення") определяют адаптируемость, как приспособленность программ и информации баз данных к модификации для эксплуатации в различных аппаратных и операционных средах без применения других действий или средств, чем те, что предназначены для этой цели при первичной разработке в исходной версии.

Адаптируемость является одной из субхарактеристик для двух показателей внешнего качества программного обеспечения.

1. Сопровождаемость (Maintainability) – способность ПО к модификации, которая включает исправление, улучшение или адаптацию ПО к изменению окружающей среды, требований или функциональных спецификаций;

2. Переносимость/Мобильность (Portability), под которой понимаются приспособленность программного обеспечения к:

- переносу программ и данных из одной аппаратной, операционной и пользовательской среды в иную по архитектуре и характеристикам среду с сохранением их целостности или небольшими изменениями функций системы;

- повторному использованию готовых программных компонентов и средств, а также информации баз данных возможно в пределах одной архитектуры аппаратной и операционной среды для расширения и изменения функций системы и программного продукта.

Мобильность (переносимость) определяется приспособленностью программного обеспечения к изменениям среды, простотой установки (инсталляцией) после переноса, замещаемостью компонентов при корректировках комплекса программ.

Приведенные выше определения термина «адаптируемость» соответствуют потребностям разработчика программных систем, но не конечного пользователя. Упоминания об адаптируемости конечным пользователем в стандартах ограничиваются масштабируемостью внутренних возможностей (например, экранных полей, размеров таблиц, объемов транзакций, форма-

тов отчетов), пригодностью для индивидуализации комплекса программ при изменениях внешней среды.

В основе языково-ориентированного проектирования лежит идея описания логики работы программы на проблемно-ориентированном языке (DSL – Domain-Specific programming Language) Заказчика (Пользователя) [5].

Сравним классический подход и языково-ориентированный (рис. 1).

Формирование концептуальной модели решения – необходимый этап проектирования в обоих случаях, после которого процесс языково-ориентированного проектирования отличается от традиционного.

Преимущество языково-ориентированного подхода очевидно: отсутствует звено кодирования, вместо него используется работа в концепциях и понятиях предметной области, для которой создается данный программный продукт. Язык, специфичный для предметной области разрабатывается заранее, возможно привлечение более опытных экспертов данной отрасли и программистов, что повышает качество программной системы.

Таким образом, для эффективного решения задач автоматизации ТП испытаний систем КА, обработки их результатов необходимо создание информационной технологии, основанной на применении моделей и методов и проблемно-ориентированного языка, который позволяет описать алгоритмы управления и проверок в терминах Заказчика (конечного пользователя).

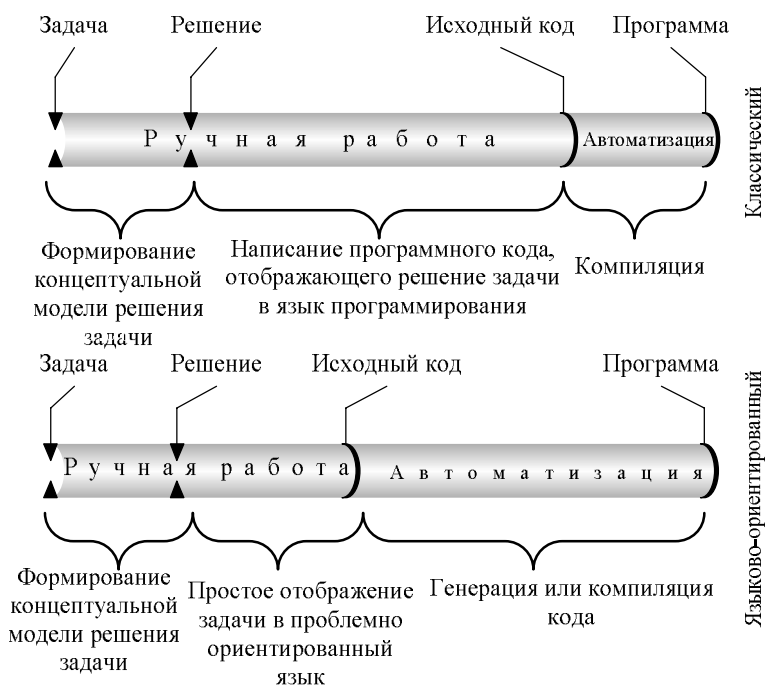


Рис. 1. Сравнение классического и языково-ориентированного подходов

Архитектурная модель программного обеспечения

Основой архитектурной модели 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 (рис. 2).

На уровне внешних данных присутствует информация двух видов:

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

2. Динамическая, источниками которой являются программные продукты фирмы Fastwel, США: PLCNET и UniOPC сервера, в основе которых тех-

нология связывания и внедрения объектов для систем промышленной автоматизации OPC (OLE for Process Control).

Стандарты OPC определяют универсальный механизм обмена данными между датчиками, исполнительными механизмами, контроллерами, устройствами связи с объектом и системами представления технологической информации. OPC устанавливает требования к классам объектов доступа к данным и их специализированным интерфейсам для использования разработчиками клиентских и серверных приложений.

Взаимодействия с контрольно-проверочной аппаратурой и бортовыми устройствами, разработанными в НПП «Хартрон-ЮКОМ», г. Запорожье и в Институте физики полупроводников НАН Украины, г. Киев осуществляется через два OPC-сервера.

1. PLCNET OPC-сервер обеспечивает обмен данными между узлами сети типа MicroPC, Adam-5510, разработанных с помощью системы программирования UltraLogik, и приложениями-клиентами OPC.

2. Универсальный UniOPC-сервер предназначен для поддержки быстрой разработки OPC серверов различных устройств, в том числе нестандартных. В программном обеспечении для испытаний систем КА UniOPC-сервер в зависимости от режима работы выполняет двойную функцию:

- в нормальном режиме обеспечивает обмен данными и командами с имитаторами и контрольно-проверочной аппаратурой через последовательный порт;

- в режиме «Модель» имитирует работу системы, чем и обеспечиваются возможности отладки и тестирования ПО, отработки технологических процессов проведения испытаний.

На уровне **платформозависимых 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 ин-

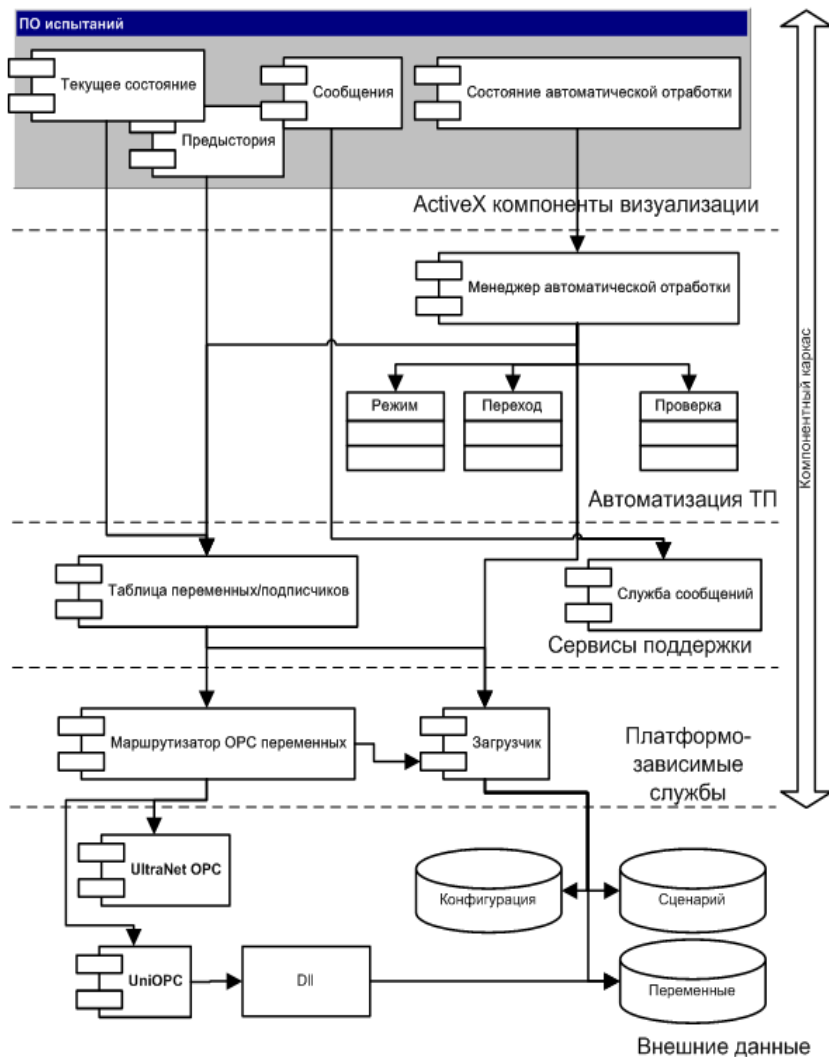


Рис. 2. Архитектурная модель программного обеспечения

формацию для всех вышестоящих программных компонентов.

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 имен переменных осуществляет подстановку их внешних псевдонимов, под которыми эти переменные зарегистрированы на OPC-серверах.

На уровне **сервисов поддержки логики программы** Хранилище данных (таблица переменные/подписчики) обеспечивает логически непротиворечивую и надежную работу алгоритмов, состав которых может изменяться во время исполнения основной программы. Хранилище данных обеспечивает целостность информации о множестве объектов 2 ключевых, базовых классов: «Переменная» и «Подписчик». «Переменные» обеспечивают хранение информации трех категорий о состоянии:

- внешних (физических) сигналов, необходимых для текущего технологического процесса;

- внутренних общепрограммных переменных, необходимых для синхронизации основных программных компонентов, например, «Текущий режим работы», «Время работы в автоматическом режиме», «Общая ресурсная наработка»;

- внутренних переменных, полученных в результате работы объектов «Подписчиков». «Подписчики» отвечают за логическую обработку информации, их работа инициируется изменением «Переменных» в хранилище данных.

Именно на основе классов «Переменная», «Подписчик» в программе решена проблема разбиения системы на относительно простые, параллельно работающие, независимые компоненты, передача информации между ними при этом осуществляется единым механизмом оповещения предварительно зарегистрированных «Подписчиков» об изменении значений «Переменных». «Подписчик», в свою очередь, является преобразователем информации и имеет право результаты своих действий опубликовать как новые значения «Переменных» в хранилище данных.

Служба сообщений решает задачи централизованного хранения, обработки и архивирование сообщений о событиях и тревогах, к основным типам которых отнесены:

- аварийные сообщения о неисправном состоянии аппаратуры во время автоматической работы по сценарию испытаний;

- сообщения о системных ошибках программного обеспечения.;

- предупредительные сообщения о неисправностях, выявленных в ходе проверок;

- информационные сообщения, к которым относятся сообщения о нормальном завершении проверок;

- сообщения о действиях оператора – командах управления внешней аппаратурой или режимами автоматической работы;

- отладочные сообщения.

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

Класс «Менеджер» отвечает за общее управление автоматической работой системы. В основе описания технологического процесса испытаний, а следовательно, и в архитектуре программно реализованных классов находятся понятия:

- *режим* – 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 пустая) выполняется в начале нового режима. Последовательность – это упорядоченный список элементарных действий, в том числе пауз для отсчета временных интервалов, сообщений оператору, изменений переменных в таблице переменных, выдачи управляющих сигналов на внешние устройства;

- *проверка* расширяет функциональность последовательности. Проверки работают параллельно, при этом отсутствует ограничение на количество одновременно функционирующих проверок. Среди возможных элементарных действий в проверке дополнительно реализовано действие «Контроль списка условий». В отличие от перехода, где контроль списка условий осуществляется в следящем режиме, в проверке это действие выполняется только в моменты времени, предписанные последовательностью. По результатам этого контроля вычисляется

результат выполнения проверки, который автоматически помещается в хранилище данных, как новое значение уникальной переменной, соответствующей данной проверке. Дополнительно, также автоматически генерируется сообщение оператору, если таковое предусмотрено описанием проверки.

Функциональности механизма, описанного выше, оказалось более чем достаточно для практической реализации программного обеспечения испыта-

ний систем электроснабжения (рис. 3, 4). 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.

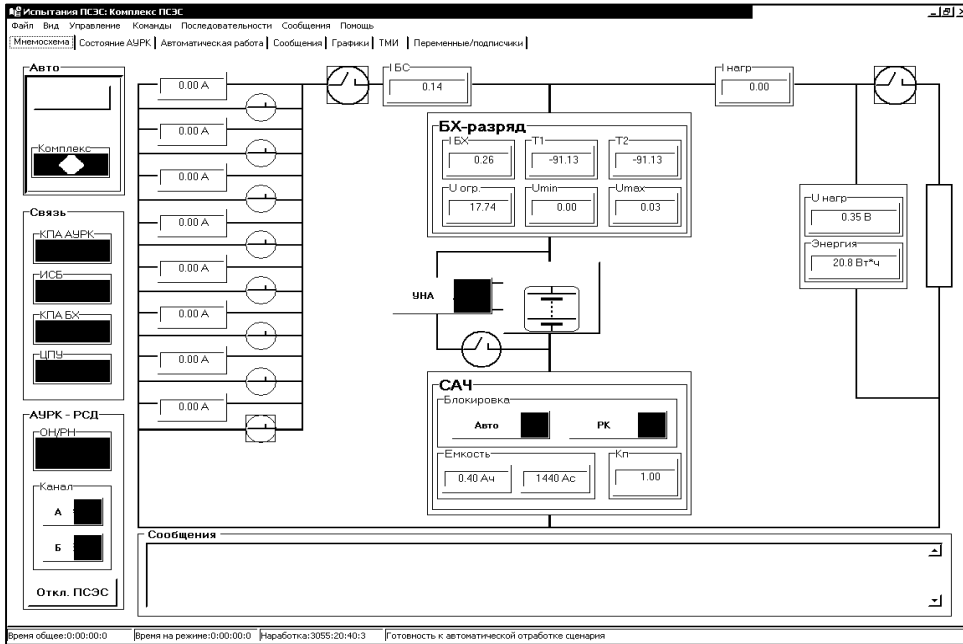


Рис. 3. Мнемосхема для визуализации текущего состояния объекта испытаний

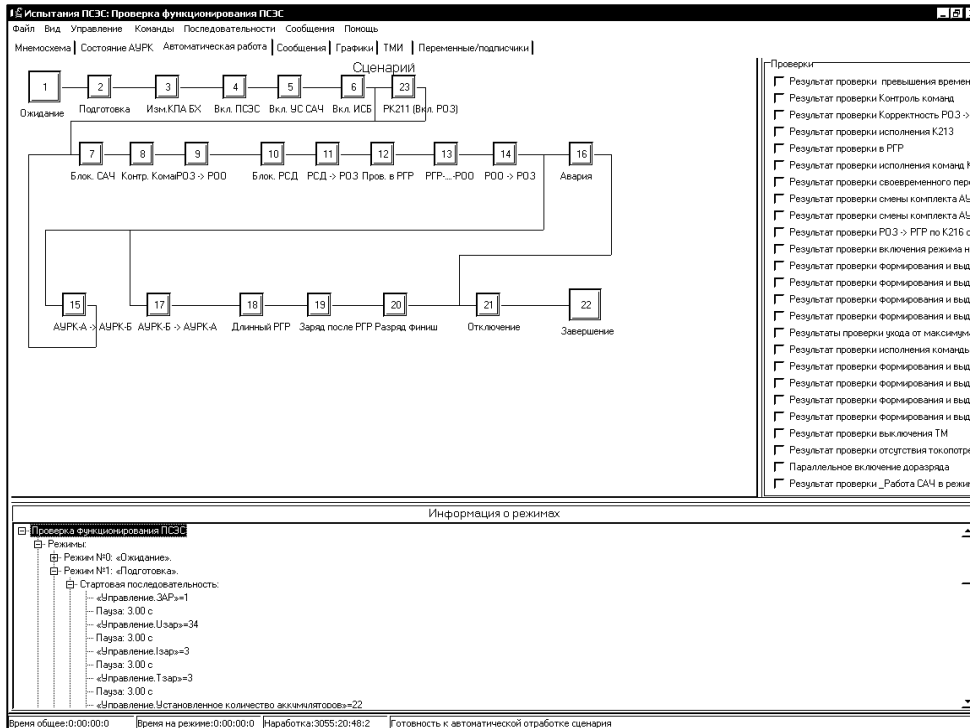


Рис. 4. Мнемосхема автоматической отработки технологического процесса

Визуализация реализована средствами ActiveX, которые представляют собой внутрипроцессные серверы, выполняющиеся в адресном пространстве приложения. Любые ActiveX объекты могут быть загружены в систему разработки большинства SCADA систем и использованы при создании прикладных программ. Управление ActiveX объектами осуществляется с помощью данных, методов и событийных функций, свойственных выбранному объекту. Применение технологии ActiveX помимо решения задачи взаимодействия с пользователем обеспечили переносимость разработки и совместимость со SCADA-системами.

Заключение

Эффективность разработанного программного обеспечения для автоматизации испытаний космической техники во многом определяется его высокой адаптируемостью. Для достижения этой адаптируемости реализованы следующие решения.

1. Компонентно-ориентированная архитектура приложения обеспечивает сопровождаемость и переносимость ПО.

2. Полный перенос описания выполняемых действий, необходимых для автоматизации технологически процессов испытаний, из программной реализации во внешний источник информации (сценарий испытаний), представленный в формате реляционной базы данных позволяет разделить сферы ответственности программиста и разработчика объекта испытаний.

3. Хорошая читаемость описания ТП и как следствие его безошибочность достигнута благода-



Рис. 5. Предполетная подготовка КА Egyptsat-1 в монтажно-испытательном корпусе космодрома «Байконур»

ря ориентации на языково-ориентированное проектирование. При этом применение взаимодополняющего совмещения графического и текстового описаний технологического процесса испытаний сделало возможным:

- представить 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 хорошую понимаемость, строгость, простоту обработки и т.д.

Приведенные примеры конкретных проектных решений уже использованы в Государственном КБ «Южное», г. Днепропетровск, НПП «Харьтон-Аркас», г. Харьков (рис. 5, 6). В настоящее время данная тематика остается актуальной: разработано ПО для автоматизации испытаний спутника МС2-8.

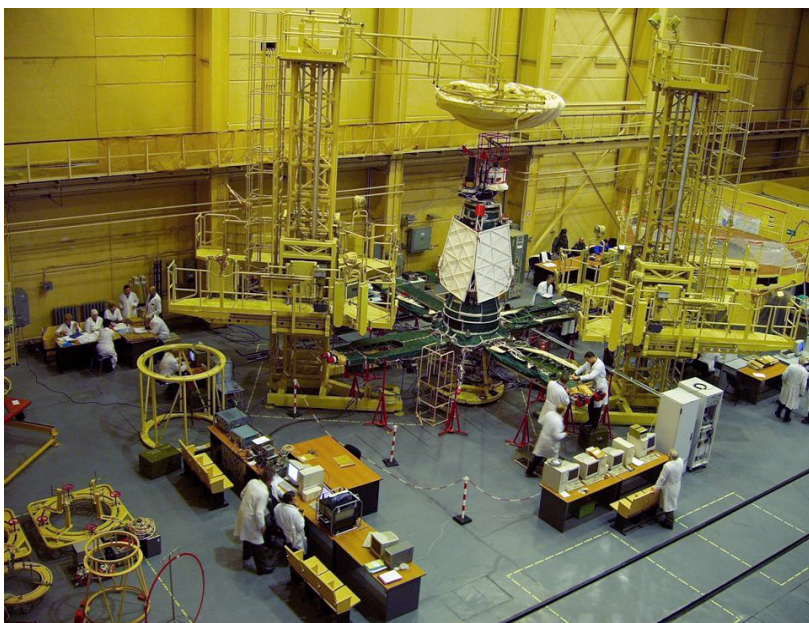


Рис. 6. Предполетная подготовка КА «Сич-1М» и микроспутника КС5МФ2 в монтажно-испытательном корпусе космодрома «Плесецк»

Література

1. Горбулін В.П. Збереження статусу ракетно-космічної держави - національне завдання України / В.П. Горбулін, А.І.Шевцов // Стратегічні пріоритети. – 2008. – № 1 (6). – С 144-152.

2. Конюхов С. Н. Научно-технические направления разработок космических аппаратов КБ "Южное" им. М. К. Янгеля / С.Н. Конюхов // Космічна наука і технологія. –1995. – № 1. – С. 12- 34.

3. Безручко К.В. Солнечные энергосистемы космических аппаратов. Физическое и математическое моделирование / К.В. Безручко, Н.В. Белан, Д.Г. Белов, С.В. Губин, В.И. Драновский, В.С. Кривцов, И.Т. Перекопский, И.Б. Туркин / Под ред. акад.

НАН Украины С.Н. Конюхова. – Харьков: Гос. аэрокосм. ун-т "Харьк. авиац. ин-т", 2000. – 516 с.

4. Белякова И.Т. Технология сборки и испытаний космических аппаратов: учеб. для вузов / 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.

5. Васин Ю.Г. Об одной технологии конструирования сложных программных комплексов / Ю.Г. Васин, М.В. Кошелев, С.Г. Кузин, А.Ф. Смирнов // Вестник Нижегородского университета: Математическое моделирование и оптимальное управление. – Н. Новгород: Изд-во Нижегородского университета. – 1998. – № 1 (18). – С. 213-226.

Поступила в редакцию 10.02.2011

Рецензент: д-р техн. наук, проф., зав. каф. економіко-математического моделювання В.М. Варталян, Національний аерокосмічний університет ім. Н.Е. Жуковського «ХАІ», Харків.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ ВИПРОБУВАНЬ СИСТЕМ ЕЛЕКТРОЗАБЕЗПЕЧЕННЯ КОСМІЧНИХ АПАРАТІВ

І.Б. Туркін, П.О. Лучшев

Розглянуто проблеми випробувань космічної техніки як складної технічної системи. Визначено основні вимоги до програмного забезпечення для автоматизації випробувань підсистем космічного апарату. Запропоновано мовно-орієнтований підхід і розроблена архітектура адаптуемого програмного забезпечення автоматизації випробувань. Застосування запропонованого підходу, а також адаптуемого програмного забезпечення дозволяє ефективно автоматизувати випробування різних складних технічних систем. Розглянуто приклади використання програмного забезпечення для автоматизації випробувань підсистеми електропостачання космічних апаратів.

Ключові слова: автоматизація випробувань, що адаптуються програмне забезпечення, мовно-орієнтований підхід, випробування підсистеми електропостачання.

SOFTWARE DEVELOPMENT FOR AUTOMATED TESTING SUBSYSTEM OF ELECTRICAL POWER SUPPLY OF SATELLITE

I.B. Turkin, P.A. Luchshev

The problems of testing of spacecraft as a complex technical system. The basic requirements for software to automate testing spacecraft subsystems is offered. A language-oriented approach is proposed and architecture of adaptive software for test automation are developed. Application of the proposed approach, as well as adaptive software allows you to effectively automate the testing of a variety of complex technical systems. The examples of the use of software to automate testing of the subsystem power supply of satellite.

Key words: automation testing, adaptable software, language-oriented approach, testing of the subsystem power supply.

Туркин Игорь Борисович – д-р техн. наук, проф., зав. каф. Інженерії програмного забезпечення, Національний аерокосмічний університет ім. Н.Е. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Лучшев Павел Александрович – канд. техн. наук, доцент каф. Інженерії програмного забезпечення, Національний аерокосмічний університет ім. Н.Е. Жуковського «Харківський авіаційний інститут», Харків, Україна.