Система информационной поддержки разработки макропрограмм автономного управления космическим аппаратом

А.А. Тюгашев
кафедра КОТ
Университет ИТМО
Санкт-Петербург, Россия
e-mail: tau797@mail.ru

А.А. Насекин
магистрант
СГАУ имени С.П. Королева
Самара, Россия
e-mail: zzzzlzzzz@yandex.ru

***Аннотация* — В докладе описывается ход работ по созданию инструментальных программных средств информационной поддержки процессов жизненного цикла разновидности бортового программного обеспечения космических аппаратов – «макропрограмм интегрального управления». Работы выполняются для конкретного предприятия-заказчика российской космической отрасли.**

Ключевые слова — макропрограмма интегрального управления; космический аппарат; бортовое программное обеспечение; визуализация; проектирование ПО; верификация программ;

I. Введение

Управление современными изделиями ракетно-космической техники (включая нано- и даже пикоспутники) осуществляется с применением бортовых цифровых вычислительных машин (БЦВМ). При этом реализация логики управления бортовыми системами как в штатных, так и в нештатных ситуациях осуществляется программным обеспечением [1]. Ошибка в бортовом программном обеспечении (БПО) может привести к катастрофическим последствиям, включая потерю дорогостоящих космических аппаратов (КА) – зачастую результатов труда многотысячных коллективов предприятий-участников разработки на протяжении ряда лет. Это касается и международных проектов, аварии и катастрофы при реализации которых вызывают значительный общественный резонанс [2, 3].

Неудивительно, что проектированию и отработке БПО уделяется большое внимание. БПО проходит многостадийную отладку с использованием имитационных моделей бортовой аппаратуры на специальных стендах [2]. Эти процессы весьма трудоемки и занимают продолжительное время. При этом по оценке специалистов ЦНИИМАШ [3], стоимость создания комплекса программного обеспечения системы управления изделием РКТ на порядок превосходит стоимость создания аппаратных средств БЦВМ. По времени разработка бортового программного обеспечения может являться критическим путем на сетевом графике работ по созданию авиационного или космического комплекса в целом [1].

Под руководством А.А. Тюгашева с выполнением существенной части работ студентами, магистрантами (в том числе А.А. Насекиным) и аспирантами, в настоящее время по заказу АО «ИСС», г. Железногорск Красноярского края, реализуется проект создания инструментального программного комплекса СИПР МП (может быть расшифровано как «система информационной поддержки разработки макропрограмм автономного управления КА), описываемый в докладе.

Это дает возможность преподавания основ программной инженерии систем реального времени на реальном нетривиальном примере со всеми его особенностями, активного привлечения обучающихся к решению практически значимых, сложных и интересных задач.

II. Описание предмета исследования

Основным объектом (предметом) проектирования, верификации и документирования в СИПР МП являются так называемые макропрограммы интегрального управления КА производства АО «Информационные спутниковые системы». АО «ИСС» является крупнейшим производителем спутниковых систем в нашей стране – 3/4 активно действующей на орбите отечественной группировки произведено в Железногорске. БПО КА производства АО «ИСС» представляет собой хорошо структурированный комплекс с выделением разновидностей программ как по назначению, так и по «уровню» [4]. Термин «макропрограммы интегрального автономного управления» принадлежит Заказчику и подразумевает особый слой БПО, реализующий функции координатора («дирижера»), задающего, какие модули ПО должны запускаться на выполнение, в какое время и в зависимости от каких логических условий [5]. Помимо управления вызовами модулей ПО функционального назначения, макропрограммы реализуют непосредственное управление бортовой аппаратурой путем выдачи на нее команд управления. В связи со

сказанным, в наименовании фигурирует слово «интегральное управление». Автономность управления подразумевает, что основные решения, призванные в том числе сохранить работоспособность КА в случае развития нештатных ситуаций, принимаются на борту без вмешательства специалистов с Земли на основе логики, заложенной в макропрограмму.

Нетрудно понять важность макропрограмм в общей структуре БПО – это классический пример «критически важного» программного обеспечения.

При этом важно отметить следующее. АО «ИСС» в настоящее время уже располагает достаточно развитой технологией производства и отработки БПО, обеспечивающей хорошие показатели качества получаемой программной продукции.

В рамках технологии Заказчика макропрограммы не формируются как «программы» в классическом понимании этого слова, записываемые в виде текстов на языке низкого, высокого уровня (или даже проблемно-ориентированном языке, что сейчас называют Domain Specific Language и что было известно и применялось в СССР еще в 1970-е). Фактически, «макропрограммы» интегрального автономного управления представляют собой массивы исходных данных (в табличной форме) для специальной разновидности БПО – бортовых интерпретаторов. Подобный подход обеспечивает значительную гибкость – при необходимости внесения изменений в логику управлении КА нет необходимости каждый раз удаленно «перепрошивать» ПЗУ бортовой цифровой вычислительной машины. В то же время, подобное обстоятельство в значительной мере обуславливает ограниченность возможности применения известных методов проектирования и верификации ПО. Формат таблиц – закрытый, принадлежит Заказчику, более того, в этой части присутствует специфика отрасли с набором режимных ограничений.

В настоящее время в рамках технологии производства БПО АО «ИСС» подготовка (проектирование и документирование) исходных данных макропрограмм осуществляется самостоятельно разработанными специалистами предприятия инструментальными программными средствами на основе таблиц. При этом существующие формы представления не обеспечивают наглядности. Тестирование хотя и является тщательным и многоэтапным, проходит, в том числе, с применением специальных автоматизированных стендов, все равно в рамках существующей технологии предполагает, что подготовка и запуск тестов возлагаются на человека.

В теории и практике программной инженерии известно достаточно большое число работ, связанных с автоматизацией жизненного цикла программ реального времени [6-14]. Однако, они либо представляют больше теоретически, нежели конкретно практически значимые для предприятий космической отрасли разработки [8,10-14], либо на создание программ на языках Ада, Си, C++, и пр. Специфика макропрограмм Заказчика и их представление не в виде исходных текстов и объектного кода, а в виде фактически данных закрытого формата, интерпретируемых бортовым интерпретатором, исключает прямое использование известных средств в данном случае. Еще одной важнейшей особенностью служит то, что элементарными «акторами» макропрограммы являются не привычные операторы присваивания, а вызовы «команд управления спутником», которые, в свою очередь, интерпретируются другим бортовым интерпретатором. Команды управления имеют достаточно сложную внутреннюю структуру, в частности – могут в ходе своего выполнения инициировать занесение запроса на включение (постановку в очередь диспетчера бортовой операционной системы) той или иной макропрограммы.

Особенностью разрабатываемой СИПР МП является также комплексный подход – ставится задача автоматизировать проектирование (используются визуальные модели), тестирование, создание программной документации и управление версиями. Написание программы как таковое предполагаемой технологией исключается.

III. Назначение и составные части СИПР МП

Назначением системы является снижение трудоемкости разработки одной из ключевых частей БПО – макропрограмм автономного управления КА, повышение надежности и качества программ за счет достижения лучшего взаимопонимания в коллективе разработчиков, автоматизации процессов тестирования на наземном отладочном комплексе и автоматизации генерации программной документации.

СИПР МП состоит из следующих основных частей:

- Интегрирующей оболочки.
- Средств визуализации и графического конструирования макропрограмм.
- Средств генерации тестов.
- Средств исполнения тестов.
- Средств документирования.
- Системы контроля версий программной документации и ее соответствия версиям макропрограмм.

Средства визуализации и графического конструирования обеспечивают представление создаваемых макропрограмм в наглядной эргономичной форме, включая внутреннюю структуру программы и межпрограммные связи.

Средства генерации тестов обеспечивают с использованием подхода «белого ящика» автоматическое построение набора тестов с заданной степенью покрытия (в настоящее время реализованы покрытие всех ветвей, покрытие всех путей и покрытие сложных условий).

Средства документирования на основе гибко настраиваемых шаблонов документов позволяют

автоматически генерировать актуальную программную документацию с поддержкой контроля версий и автоматической проверкой актуальности документации на маропрограмму (соответствия версии документа версии программы).

IV. Средства визуализации и графического конструирования

Одной из возможных причин возникновения ошибок в критическом ПО можно считать сложность самого процесса разработки бортового программного обеспечения. На начальном этапе логику программы разрабатывают проектировщики (алгоритмисты, системные аналитики), которые затем передают эту информацию программистам. Недопонимание или неточность при взаимодействии может привести (и, увы, приводит) к ошибкам. Для минимизации ошибок в программах, создаваемых для исполнения на бортовых вычислительных системах, был предложен метод визуального контроля ранее созданных макропрограмм и графического конструирования новых, описываемый ниже. Таким образом возможно уменьшить степень недопонимания и предоставить процесс разработки макропрограмм проектантам системной логики управления изделием, исключив программистов (раз человеку-программисту свойственно ошибаться, наиболее радикальным решением с позиций обеспечения надлежащего качества и надежности является устранение человека из процесса!).

A. *«Группы логических последовательностей»*

Макропрограмма интегрального автономного управления состоит из так называемых «групп логических последовательностей» (термин Заказчика). Каждая логическая последовательность имеет главное («спусковое, охраняющее» - *guard*) условие, в случае истинности которого будут выполнена заданная последовательность команд управления спутником, возможно – с указанием временного интервала между командами (блок команд с привязкой к таймеру).

Суть модели легко понять по визуальному представлению, показанному на рисунке 1.

Разрабатываемое инструментальное программное обеспечение должно «без швов» интегрироваться с существующим действующим программным обеспечением Заказчика и взаимодействовать с уже существующей базой данных с ее форматом представления макропрограмм. Необходимо обеспечить возможность графического редактирования и сохранения всех изменений в базе данных макропрограмм.

B. *Состояние разработки*

На данный момент был создан инструмент, который позволяет визуализировать и конструировать группы логических последовательностей. При этом для каждого элемента группы существует отдельный редактор, с помощью которого можно настраивать его свойства, и тем самым влиять на функционирование макропрограммы (рисунок 1). В связи с тем, что в структуре макропрограмм существуют команды и для перехода из одной группы логических последовательностей в другую, был также разработан визуализатор связей между группами. Макропрограмма представляется графом, в котором множество вершин есть множество групп логических последовательностей, а множество ребер – отражает связи, когда в логической последовательности присутствует отдается команда на заявку в очередь исполнения другой группы.

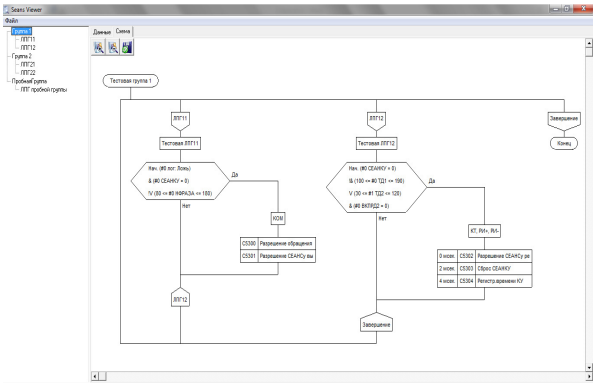

Рис. 1. Экран средств визуализации и графического конструирования

Внешний вид программы визуализации связей групп представлен на рисунке 2.

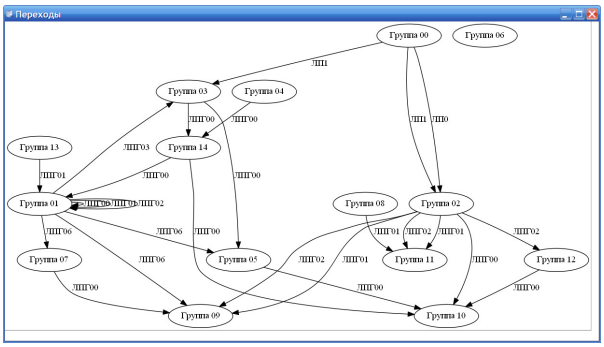

Рис. 2. Визуализация связей между группами ЛП

V. Средства генерации тестов

В СИПР МП применяются следующие базовые подходы к автоматизации тестирования.

A. *Степень покрытия*

В силу критической важности макропрограмм, весьма важно убедиться, что тесты максимально полно покрывают исходную программу. Критерий покрытия операторов (всех действий, возможных в макропрограмме) в данном случае недостаточен. В СИПР МП реализуется по выбору пользователя либо покрытие ветвей, либо – маршрутов на управляющем графе программы. Задача покрытия маршрутов в данном случае

вполне разрешима на практике вследствие природы макропрограмм интегрального управления, в которых циклы в классическом виде отсутствуют.

В действительности применяется модифицированный критерий покрытия ветвей/условий [15], в соответствии с которым обеспечивается достаточный уровень покрытия, поскольку он проверяет влияние каждой составляющей сложных условий и проходит по всем ветвям управляющего графа. При этом используется допущение о взаимной независимости логических условий (до некоторой степени обоснованное тем, что макропрограммой интегрального управления проверяются в основном значения телеметрические параметры, множество которых проектируется так, чтобы они в максимальной степени были независимыми).

B. Дополнительные требования

По желанию Заказчика, каждый тест сопровождается графическим представлением «трассы исполнения», на которой наглядно отображается последовательность действий, предусматриваемых макропрограммой для данной ситуации.

C. Корректные и некорректные исходные данные

Тесты генерируются не только для случая корректных, но и некорректных исходных данных. Используются различные виды проверяемых в условиях параметров. Типы условий: "значение имеет допустимые границы", "логическая переменная (ДА/НЕТ)", и "вызов функции". Для каждого типа параметров применяется свой подход к генерации корректных и некорректных входных значений.

D. Тестирование и макропрограммы в целом, и частей

Необходимо проверить и каждую «группу логических последовательностей» в отдельности, и всю сложную взаимосвязанную структуру, определяемую командами передачи управления между группами логических последовательностей.

E. Реализация на данный момент

Для генерации тестов используется алгоритм обхода графа связей между группами логических последовательностей с указанием начальной группы для каждой макропрограммы (при входе в группу затем происходит обход дерева группы слева направо, при прохождении разрешающего условия строится набор значений параметров, позволяющий пройти по ветви «да» или «нет», с учетом всех компонент сложного условия).

На настоящий момент построен прототип, позволяющий:

- Подключаться к существующей базе макропрограмм, разработанной у Заказчика ранее.
- Находить все возможные корректные и некорректные комбинации проверяемых параметров.
- Генерировать набор покрывающих тестов по группам и для всей макропрограммы.
- Визуализировать исполнение выбранного варианта.

Копия экрана средств генерации тестов представлена на рисунке 3.

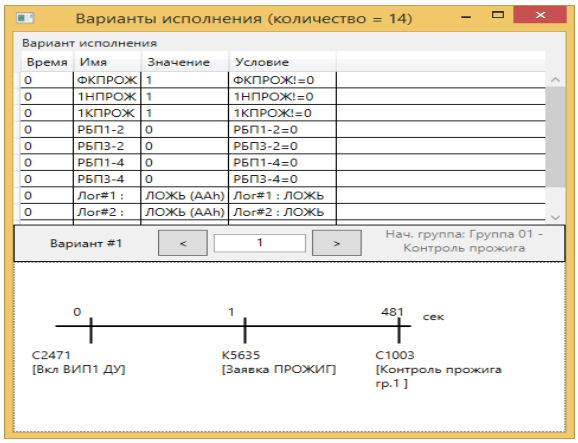

Рис. 3. Экран прототипа средств генерации тестов

VI. Заключение

В докладе представлен ход разработки набора инструментальных программных средств специального назначения, разрабатываемый для конкретного Заказчика из российской космической отрасли.

С целью улучшения взаимопонимания в коллективе разработчиков, снижения числа ошибок предлагается подход, основанный на использовании визуального языка для представления макропрограммы (фактически – визуального программирования).

Средства комплекса СИПР МП должны позволить повысить качество и надежность критически важной составляющей бортового программного обеспечения КА – «макропрограмм интегрального автономного управления» за счет реализации следующих возможностей:

- Наглядного просмотра структуры ранее созданных макропрограмм.
- Средств графического конструирования вновь создаваемых макропрограмм.
- Средств автоматической генерации тестов с заданным уровнем покрытия.
- Средств автоматизированного исполнения тестов.
- Средств документирования, обеспечивающих автоматизированное формирование заданной документации на макропрограмму с гарантией соответствия версий документов версиям программ.

На настоящий момент успешно сдан заказчику первый этап - прототип создаваемого комплекса СИПР МП,

завершен опытный образец в соответствии с техническим заданием и дополнениями к нему.

Список литературы

[1] Авиастроение. Том 6 (Итоги науки и техники, ВИНИТИ АН СССР). М., 1978

[2] Есюнин В.В. Основные принципы построения испытательного стенда бортового комплекса управления для перспективных изделий ОАО «ИСС». Вестник СибГАУ, 2010, Т. 2, С. 93-96.

[3] А.А. Тюгашев, И.А. Ильин, И.Е. Ермаков. Пути повышения надежности и качества программного обеспечения в космической отрасли // Управление большими системами. Сборник трудов. Инс т проблем управления им. В.А.Трапезникова РАН, Вып.39 2012, с. 288-299.

[4] Антамошкин А.Н., Колташев А.А. Технологические аспекты создания бортового программного обеспечения спутников связи. Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2005. № 3. С. 93-95.

[5] Кочура Е.В. Разработка макропрограмм интегрального управления космическими аппаратами. Вестник СибГАУ, 2011, Т.1, С 105-107.

[6] Бахмуров А.Г., Смелянский Р.Л. Проблемы инструментальной поддержки разработки распределенных встроенных систем реального времени // Программирование. 2013. Т. 39. № 5. С. 5-21.

[7] Смелянский Р.Л., Бахмуров А.Г., Волканов Д.Ю., Чемерицкий Е.В. Интегрированная среда для анализа и разработки распределенных встроенных вычислительных систем реального времени // Программирование. 2013. Т. 39. № 5. С. 35-52.

[8] Шаров О.Г., Афанасьев А.Н. Методы и средства трансляции графических диаграмм // Программирование Т.37 № 3 2011 С. 65-75

[9] Буздалов Д.В., Зеленов С.В., Корныхин Е.В., Петренко А.К., Страх А.В., Угненко А.А., Хорошилов А.В. Инструментальные средства проектирования систем интегрированной модульной авионики // Труды Института системного программирования РАН. 2014. Т. 26. № 1. С. 201-230.

[10] Swarup Mohalik, Ambar A. Gadkari, Anand Yeolekar, K.C. Shashidhar, S. Ramesh. Automatic test case generation from Simulink/Stateflow models using model checking. Software Testing, Verification and Reliability, Vol. 24, Iss. 2, pp. 155–180, March 2014

[11] Maximiliano Cristiá, Pablo Albertengo, Claudia Frydman, Brian Plüss, Pablo Rodríguez Monetti. Tool support for the Test Template Framework. Software Testing, Verification and Reliability, Vol. 24, Iss. 1, pp. 3–37, January 2014

[12] Adilson Luiz Bonifácio, Arnaldo Vieira Moura. A new method for testing timed systems. Software Testing, Verification and Reliability. Vol. 23, Iss. 2, pp. 91–117, March 2013

[13] Abdeslam En-Nouaary, A test purpose-based approach for testing timed input output automata Software Testing, Verification and Reliability, Vol. 23, Iss. 1, pp. 53–76, January 2013

[14] Mercedes G. Merayo, Manuel Núñez and Ismael Rodríguez. A formal framework to test soft and hard deadlines in timed systems. Software Testing, Verification and Reliability, Vol. 22, Iss. 8, pp. 583–608, December 2012

[15] Покрытие программного кода. ИНТУИТ – Интернет-университет информационныхтехнологий. http://www.intuit.ru/studies/courses/1040/209/lecture/5410.