

МЕХАНИКА

УДК 629.7.054.847

А.Б. Бовсуновский, В.Г. Бутов, А.А. Хвалько

АРХИТЕКТУРА ИНТЕГРИРОВАННОЙ СИСТЕМЫ ДЛЯ ПРОВЕДЕНИЯ МЕХАНИЧЕСКОГО АНАЛИЗА БОРТОВОЙ РАДИОАППАРАТУРЫ КОСМИЧЕСКИХ АППАРАТОВ¹

Рассмотрена архитектура расширяемой интегрированной системы для проведения инженерного анализа. Выделены основные принципы разработки архитектуры подобного программного комплекса. Функционал интегрированной системы разделен на три класса: подготовка исходных данных, проведение расчетов по заданным методикам, анализ результатов. Предложена модульная архитектура комплекса, позволяющая совместить преимущества использования расчетных модулей сторонних CAD/CAE-систем наряду с реализацией собственных методических и теоретических разработок. Описана модель взаимодействия модулей комплекса на основе унифицированных программных интерфейсов. Описан минимальный набор программных интерфейсов и модулей, необходимых для построения комплекса по проведению механического анализа бортовой радиоаппаратуры космических аппаратов.

Ключевые слова: *CAD/CAE-система; программный комплекс; модульная архитектура; механический анализ.*

На сегодняшний день существует множество систем проектирования и инженерного анализа (так называемых CAD/CAE-систем) в той или иной области разработки изделий и симуляции проходящих физических процессов. Бортвая радиоэлектронная аппаратура (РЭА) космических аппаратов (КА) в этом смысле является неким эталоном, демонстрирующим самые передовые возможности широкого применения данных систем. Ее конструктивная сложность, многоплановость схемотехнических решений, обилие применяемой специальной элементной базы, необходимость противостоять большому спектру внешних воздействующих факторов в земных условиях и условиях космического пространства делают необходимым применение разноплановых систем автоматизированного проектирования (САПР) [1]. Эти системы отличаются спектром решаемых задач и дополнительным функционалом подготовки исходных данных и обработки результатов. Кроме того, требуется дополнительная адаптация функционала этих систем к специфике того или иного производства, организация взаимодействия, автоматизация

¹ Работа выполняется в порядке реализации Постановления № 218 Правительства РФ от 09.04.2010 г. «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства», и договора № 13.G25.31.0017 от 07.09.2010 между ОАО «ИСС» им. акад. М. Ф. Решетнева» и Минобрнауки РФ.

решения типичных задач и выполнения типовых процедур. Возникающие при этом трудности удается частично решить за счет встроенных в системы средств автоматизации и обмена данными, но возникает и целый ряд важных проблем:

- а) высокая стоимость итогового программного комплекса;
- б) сложность или невозможность использования собственных методических и теоретических наработок;
- в) ограниченная или отсутствующая возможность гибкой настройки комплекса для решения новых специфичных задач.

В данной статье рассматривается архитектура интегрированной системы (ИС) для проведения специализированных инженерных расчетов, в которой могут быть решены обозначенные проблемы.

Гибкая настройка, расширение и изменение функционала ИС под различные задачи естественным образом может быть обеспечена применением модульной архитектуры. Её использование подразумевает выделение определенного функционала (комплекса смежных функций) в отдельные программные модули, для которых возможна независимая разработка, интеграция и поддержка. Модули могут быть выполнены как в виде программных библиотек, динамически загружаемых в процессе работы (DLL-модулей), так и в виде независимых исполняемых программ (EXE-модулей).

Явные признаки применения модульной архитектуры можно встретить и у CAD/CAE-систем от ведущих мировых производителей в этой области. В таких программных комплексах, как ANSYS и NASTRAN инструментарий разработки и анализа изделий разделен на три функциональных класса: так называемые «препроцессор», «решатель» и «постпроцессор». К первому классу можно отнести подготовку и обработку геометрических данных, построение расчетной модели, задание начальных и граничных условий. Ко второму классу относятся реализации различных методик (разностных, конечно-элементных и т.д.) по решению задач моделирования определенных физических процессов. К третьему классу относятся построители графиков и отчетов, анализаторы критических значений рассчитанных параметров.

Опыт использования различных CAD/CAE-систем показывает, что хранение исходных данных и результатов анализа в «нейтральных» форматах обеспечивает значительно большую свободу в выборе сторонних программных комплексов для работы в единой ИС. Например, для хранения и обмена геометрических данных удобно использование форматов IGES и STEP, а для сеточного разбиения конечно-элементной модели удобно использовать формат SAT. Наконец, в качестве важного связующего звена ИС удобно использовать единую базу данных, которая хранит исходные сведения для проведения расчетов (в том числе геометрические модели), данные самих расчетов и результаты их обработки. Оптимальной формой организации такой базы является сочетание реляционной базы данных табличного типа и файловой библиотеки дополнительных материалов.

Для осуществления поставленных задач предлагается построить ИС по следующим принципам:

- а) использовать модульную архитектуру, где комплексы функций схожего назначения выделены в отдельные программные модули;
- б) обозначить три класса модулей: «препроцессоры», «решатели», «постпроцессоры»;
- в) взаимную связь между модулями ИС организовать на основе унифицированных программных интерфейсов (API).

Для обеспечения работы составных частей препроцессора и постпроцессора в едином информационном (адресном) пространстве ИС данные классы модулей предлагается выполнять в виде DLL-модулей, поддерживающих унифицированные API. Такой подход позволяет избежать процедур дополнительного копирования и преобразования данных, используемых различными программными компонентами, а также открывает широкие перспективы гибкой настройки ИС к новым специфичным задачам путем выпуска и интеграции дополнительных модулей автоматизации подготовки и анализа данных.

Модули из класса решателей предлагается разрабатывать в виде таких же DLL-модулей, но имеющих определенную специфику. В случае реализации собственных методик моделирования физических процессов такой модуль может включать в себя полный функционал решателя, а в случае использования расчетного EXE-модуля стороннего производителя – быть, фактически, лишь надстройкой над ним и автоматизировать его запуск/остановку с заданными параметрами командной строки. В частности, задействование ядра CAE-системы ANSYS может быть организовано запуском соответствующего EXE-модуля с указанием входного и выходного файла, а в качестве входного файла средствами препроцессора ИС может быть подготовлен (сверстан) набор команд построения или загрузки модели, запуска расчета, динамической корректировки параметров [2]. Использование потенциала расчетных модулей сторонних CAD/CAE-систем во многих случаях имеет преимущество перед разработкой собственных аналогичных средств как по времени разработки, так и по скорости и качеству самих расчетов.

Изложенные принципы построения ИС нашли применение при проектировании аппаратно-программного комплекса для проведения механического анализа унифицированных электронных модулей бортовой радиоаппаратуры космических аппаратов. В качестве основного решателя используется ядро CAE-системы ANSYS, графическая оболочка комплекса разработана с использованием кросс-платформных библиотек QT и OpenCascade, а связь с реляционной базой данных в операционной системе Windows организована через драйвер ODBC.

Загрузка и работа с DLL-модулями комплекса производится по единой схеме и отчасти напоминает технологию Microsoft Component Object Model (COM):

- 1) загружается DLL-файл, в котором определяется адрес функции «фабрики» объектов;
- 2) запуск указанной функции создает экземпляр функционального объекта модуля и возвращает ссылку на его основной программный интерфейс;
- 3) через основной программный интерфейс приложение получает доступ ко всем дополнительным интерфейсам, поддерживаемым модулем.
- 4) удаление объекта производится вызовом соответствующей функции основного программного интерфейса.

Для связывания составных частей комплекса разработаны два программных интерфейса: IAppHook и его наследник ISubModule. При проектировании DLL-модулей комплекса в каждый из них закладывается поддержка ISubModule в качестве основного интерфейса, тогда как основное приложение (оболочка комплекса) поддерживает только IAppHook. Функциональный состав данных интерфейсов приведен в табл. 1 и 2 соответственно.

Интерфейс IAppHook имеет две пары симметричных функций и обслуживает межмодульный обмен ссылками на объекты графической оболочки на базе QT и на программные интерфейсы. Таким образом, разработчик модуля может

Таблица 1

Функциональный состав интерфейса IAppHook

Функции API	Описание
long GetObject (long objID, QObject **ppObj)	Посредством данной функции основное приложение запрашивает у модуля ссылки на объекты платформы QT, которые можно встроить в общий графический интерфейс комплекса (главное меню, панели инструментов, окна). За каждым из объектов закреплен уникальный числовой идентификатор
long SetObject (long objID, QObject *pObj)	Посредством данной функции основное приложение передает модулю ссылку на основное графическое окно. Эта ссылка используется при создании дочерних окон модуля, которые впоследствии запрашиваются командой GetObject()
long GetInterface (long IID, void **ppObj)	Посредством данной функции основное приложение запрашивает у модуля ссылку на программный интерфейс. Также и модуль может воспользоваться этой функцией основного приложения для получения ссылки на интерфейс смежного модуля. За каждым из интерфейсов закреплен уникальный числовой идентификатор
long SetInterface (long IID, void *pObj)	Посредством данной функции основное приложение передает модулю ссылку на собственный интерфейс IAppHook. Впоследствии эта ссылка используется модулем для ретрансляции запроса дополнительных интерфейсов у смежных модулей комплекса

Таблица 2

Функциональный состав интерфейса ISubModule

Функции API	Описание
char * GetModuleName ()	Посредством данной функции основное приложение получает имя модуля, которое впоследствии используется для показа пользователю (заголовок окна описания и надпись команды активации модуля)
char * GetModuleDescription ()	Посредством данной функции основное приложение получает расширенное описание модуля, которое впоследствии используется для показа пользователю (содержательная часть окна описания модуля)
long GetModuleId ()	Посредством данной функции основное приложение получает числовой идентификатор модуля, который впоследствии используется для пересылки уведомлений. За каждым из модулей закреплен уникальный числовой идентификатор
long GetModuleVersion (long *pMajor, long *pMinor, long *pBuild)	Посредством данной функции основное приложение получает сведения о версии модуля, которые впоследствии используются для отслеживания совместимости компонент программного комплекса
bool isActive ()	Посредством данной функции основное приложение проверяет статус активности модуля
long Initialize ()	Посредством данной функции основное приложение инициализирует загруженный модуль. При инициализации модулей допускается их связывание между собой, поэтому основное приложение сначала загружает все доступные модули, а потом инициализирует их
long Activate (bool bNewStatus)	Посредством данной функции основное приложение управляет активностью модуля. Соответствующая команда встраивается в графическую оболочку комплекса
void dispose ()	Посредством данной функции основное приложение выгружает модуль из памяти компьютера по окончании работы

интегрировать в оболочку основного приложения полный спектр необходимых элементов управления (пункты главного меню, панели инструментов, окна и др.). Кроме того, любой из модулей через основное приложение может запрашивать непосредственные ссылки на интерфейсы смежных модулей для доступа к необходимому функционалу. При этом основное приложение играет роль ретранслятора запроса и опрашивает модули по очереди, пока какой-либо из них не предоставит требуемый интерфейс.

Интерфейс ISubModule расширяет IAppHook функциями, специфичными только для DLL-модулей комплекса. Среди них можно выделить два набора: первые 4 функции позволяют получить описание модуля, а вторые 4 – управлять его жизнедеятельностью. Важной особенностью является управление состоянием активности модуля. В активном состоянии DLL-модуль может получать уведомления о действиях пользователя или о готовности результатов работы других модулей (при условии, что он поддерживает интерфейсы работы с уведомлениями). Таким образом, активные модули могут динамически отслеживать и реагировать на изменения в рабочей области (вставка дополнительных команд при вызове контекстных меню, автоматический пересчет параметров при изменении свойств моделируемого объекта и др.). В неактивном состоянии модуль не реагирует на уведомления и делает свои элементы управления недоступными пользователю.

Для обеспечения непосредственного взаимодействия между модулями потребовалась разработка дополнительных программных интерфейсов (приводятся без детализации функционального состава):

- IDbUser – функции доступа к таблицам, записям и полям базы данных;
- IChangesTrack – функции отслеживания изменений в параметрах расчетной модели;
- IContextTrack – функции отслеживания вызова контекстного меню для элементов расчетной модели;
- ISettingsManager – функции работы с настройками модулей, хранимыми в базе данных;
- IProgressHook – функции отслеживания и управления процессом расчета.

Базовый функционал комплекса для проведения механического анализа может быть обеспечен следующим набором модулей:

- База Данных – модуль связи с реляционной базой данных и файловой библиотекой;
- Импорт-Экспорт – модуль автоматизации импорта и экспорта геометрии и прочей информации из сторонних CAD/CAE-систем и баз данных;
- 3D-модель – модуль визуализации пространственной геометрии расчетной модели;
- Механализ – модуль автоматизации работы с ядром CAE-системы ANSYS;
- Отчет – модуль автоматизации обработки данных расчета.

Дополнительной спецификой аппаратно-программного комплекса для проведения механического анализа унифицированных электронных модулей бортовой радиоаппаратуры космических аппаратов являются организация связи с CAD/CAM/CAE-системами и базой данных, используемой на предприятии, а также методика построения конечно-элементных моделей для проведения различных видов анализа. Все это реализуется средствами вспомогательных модулей пре-процессора и постпроцессора ИС.

ЛИТЕРАТУРА

1. *Руководство по системному инжинирингу NASA. NASA SYSTEMS ENGINEERING HANDBOOK (REV. 1). NASA/SP-2007-6105. URL: <http://education.ksc.nasa.gov/esmdspacegrant/Documents/NASA%20SP-2007-6105%20Rev%201%20Final%2031Dec2007.pdf>*
2. *Каплун А.Б., Морозов Е.М., Олферьева М.А. ANSYS в руках инженера. Практическое руководство. М.: Едиториал УРСС, 2003. С. 194–269.*

Статья поступила 20.09.2011 г.

Bovsunovskii A.B., Butov V.G., Khval'ko A.A. ARCHITECTURE OF THE INTEGRATED SYSTEM FOR MECHANICAL ANALYSIS OF SPACECRAFT ON-BOARD ELECTRONIC EQUIPMENT. The architecture of the expandable integrated system for engineering analysis is discussed. The main principles of design for such system are emphasized. The functionality of the integrated system is divided onto three classes: preparing the initial data, performing calculations based on certain methods, and analyzing the results. The module architecture which comprises the advantages of both the facilities of external CAD/CAE-systems and own methodical and theoretical implementations, is presented. The model of cross-module interoperability based on unified program interfaces is described. The minimal required set of program interfaces and modules for building the integrated system for mechanical analysis of the spacecraft on-board electronic equipment is mentioned.

Keywords: CAD/CAE-system; program complex; module architecture; mechanical analysis.

BOVSUNOVSKIY Aleksandr Borisovich (Tomsk State University)

E-mail: Alexander.Bovsunovsky@niipmm.tsu.ru

BUTOV Vladimir Grigor'evich (Tomsk State University)

E-mail: bvg@niipmm.tsu.ru

HVALKO Aleksandr Aleksandrovich

(The JSC "Academician M.F. Reshetnev "Information Satellite Systems")

E-mail: hvalko@iss-reshetnev.ru