

ПОДСИСТЕМА УПРАВЛЕНИЯ РАСЧЕТНЫМИ ДАННЫМИ ИНТЕГРАТОРА ЛОГОС МИП

Е. А. Ескова, А. Д. Черевань, А. Г. Надуев, Д. А. Кожяев

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

Введение

Модульная интеграционная платформа (МИП), входящая в состав пакета программ ЛОГОС, используется для ряда задач, в том числе для подготовки и проведения:

- связанных расчетов комплексных мультидисциплинарных задач математического моделирования;
- задач оптимизации и параметрических исследований.

В состав модульной интеграционной платформы входит Интегратор, который предоставляет графический интерфейс пользователя и выполняет функцию связующей программы, позволяя взаимодействовать в единой логически связанной цепочке всем необходимым модулям. Расчетные данные, используемые Интегратором, могут взаимодействовать с файловыми объектами, и организованы в виде древовидной структуры. Для управления этой структурой данных и отображения ее в Интеграторе, была реализована подсистема «Управления расчетными данными», которая состоит из следующих модулей:

- древовидная структура данных, которая опеределает типы объектов и их взаимосвязи;
- контроллер распределенной файловой структуры, который обеспечивает управление расчетными файловыми объектами в локальной и удаленной файловых системах;
- графическое дерево объектов, которое используется для отображения пользователям структуры данных задачи и возможности взаимодействия с отдельными объектами структуры.

Древовидная структура данных

Структура данных, определяет типы объектов и их взаимосвязи. Структура статическая, то есть объекты, которые могут быть в ней созданы, заранее определены и достаточны для решения расчетных задач в Интеграторе ЛОГОС МИП.

Структура содержит следующие элементы:

- типы данных;
- контейнер элементов;
- контейнер связей.

Структура данных храниться в формате xml и должна быть представлена в единственном экземпляре. Для обеспечения её уникальности, структура

реализована с использованием шаблона проектирования SingleTone (Одиночка) [1, 130].

Типы данных

Тип расчетных данных, в рамках Интегратора ЛОГОС МИП, представляет собой класс C++, унаследованный от базового класса BaseElement. Такой тип наследования, один базовый класс для всех типов данных, позволяет обеспечить интерфейс для доступа к производному классу, через указатель на базовый класс, а также хранение и обход разнотипных данных. Для гибкого интерфейса копирования/клонирования структура типов данных организована с помощью порождающего шаблона проектирования «Прототип» [1, 121].

В Интеграторе реализованы следующие типы расчетных данных:

- геометрическая модель;
- расчетная модель;
- параметры геометрической модели;
- параметры расчетной модели;
- задача параметрических исследований;
- задача оптимизации;
- расчетная задача;
- результаты расчетной задачи.

Контейнер элементов

Контейнер элементов содержит в себе объекты, созданные в процессе подготовки и расчета задачи. Каждый объект имеет свой уникальный идентификатор в рамках проекта, а так же связь с файловыми объектами. Контейнер элементов выполняет следующие функции:

- хранение объектов;
- добавление/удаление новых элементов;
- поиск элементов данных с указанным идентификатором или типом.

Контейнер связей

Содержит описание связей между объектами, что позволяет выстроить иерархию расчетных данных. Контейнер выполняет следующие функции:

- хранение связей между объектами;
- добавление/удаление связей.

Набор вспомогательных методов для поиска и навигации

Обход и составление структуры дерева объектов реализованы с помощью паттерна проектирования «Визитер» [1, 314]. Использование паттерна Визитер (Visitor) позволяет единообразно обойти набор элементов с разнородными интерфейсами, а также позволяет добавить новый метод (функцию) в класс объекта, при этом, не изменяя сам класс этого объекта. В подсистеме управления расчетными данными интегратора ЛОГОС МИП были реализованы следующие виды визитеров:

- Визитер прямого обхода. На вход подается узел, визитер, перемещаясь вниз по дереву объектов, находит все узлы, заданного типа, а также их взаимосвязи.

- Визитер обратного обхода. На вход подается узел, визитер, перемещаясь вверх по дереву объектов, составляет путь до корневого элемента.

На базе данных визитеров реализованы различные виды поиска, например, поиск элементов по типу и идентификатору.

Контроллер распределенной файловой структуры

Модуль «Контроллер распределенной файловой структуры» предназначен для взаимодействия интегратора с распределенной структурой данных проекта.

Модуль выполняет следующие основные задачи:

- формирует и контролирует структуру данных в локальном и удаленном каталоге проекта;

- контролирует занятость проекта при помощи сигнальных файлов, используемых для оценки статуса расчетов;

- обмен данными между локальным и удаленным каталогами проектов, осуществляемый по запросу пользователя.

Общая структура взаимодействия модуля ProjectDirectory с каталогами проекта представлена на рис. 1.

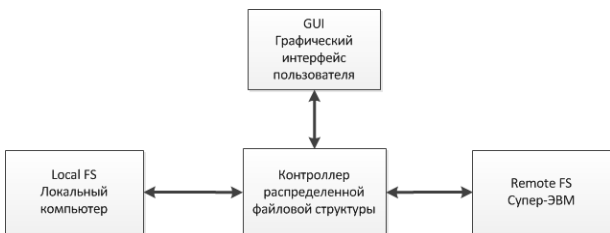


Рис. 1. Общая структура взаимодействия ProjectDirectory с каталогами проекта

Каталоги проекта, взаимодействующие с модулем ProjectDirectory:

- Local – каталог текущего проекта на локальном компьютере. Содержит данные для работы с САПР, ППП и оптимизатором;

- Remote – каталог текущего проекта на супер-ЭВМ. Содержит данные, для расчета оптимизационных задач и результаты этих расчетов.

Модуль ProjectDirectory осуществляет взаимодействие с каталогом текущего проекта на локальном компьютере с использованием локальных файловых операций, а с каталогом текущего проекта на супер-ЭВМ с использованием сетевого протокола SSH.

Алгоритм взаимодействия модуля ProjectDirectory с каталогами проекта на примере подготовки и решения расчетной задачи

Подготовка задачи с вызовом САПР и ППП из интегратора и дальнейшая оптимизация геометрии выполняется на локальном компьютере. При этом выполняется следующий алгоритм:

- создается новый проект, представляющий собой каталог в файловой системе локального компьютера;

- запускается САПР в каталоге проекта;

- производится подготовка геометрической модели;

- задаются параметры геометрической оптимизации;

- запускается ППП, в который загружается файл проекта, созданный в САПР;

- формируется расчетная сетка изделия;

- задаются сеточные величины и граничные условия;

- выбираются параметры оптимизации и выходные параметры;

- создается расчетная модель;

- формируется файл, который содержит:

- параметры оптимизации, диапазоны и начальные значения;

- целевые функции для выходных параметров и их желаемое значение.

Графический интерфейс пользователя представлен в виде дерева проекта, в соответствии с рис. 2.

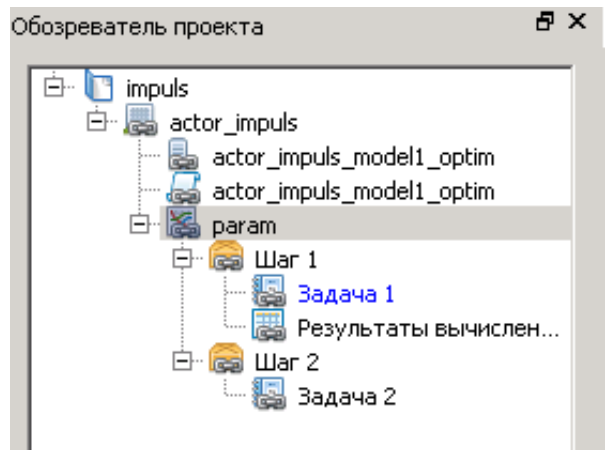


Рис. 2. Дерево проекта в графическом интерфейсе пользователя

Дерево проекта в файловой системе локального компьютера представлено на рис. 3.

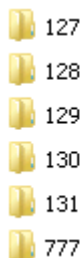


Рис. 3. Дерево проекта в файловой системе локального компьютера

Дерево проекта в файловой системе локального компьютера содержит директории, обозначенные 0, 1, 2, 3 и т. д. – это уникальные идентификаторы, которые присваиваются каждому логическому объекту в структуре данных дерева проекта.

Модуль ProjectDirectory копирует файлы в каталог проекта и устанавливает связь между скопированным файлом и элементом дерева, хранит в себе описание дерева (главная информация – на какие файлы ссылается тот или иной пункт дерева проекта) и служебную информацию о проекте (в том числе, диапазоны изменения оптимизируемых параметров, пути к решателям).

Необходимо отметить, что ProjectDirectory позволяет получить при помощи сигнальных файлов статус текущего проекта на локальном компьютере – свободен или занят проект другим интегратором.

Для запуска задачи выполняется следующий алгоритм:

- копируется на супер-ЭВМ файл с описанием интегральных параметров;
- рассчитывается совокупность наборов параметров оптимизации:
- геометрические параметры передаются в САПР,
- параметрические – передаются в ППП;
- создается новая расчетная модель в САПР;
- рассчитывается вариант задачи в ППП на основе:
 - новых значений оптимизационных параметров,
 - новой расчетной модели,
 - оригинальной модели;
- обрабатывается следующий набор параметров оптимизации, создается новая расчетная модель и т. д. пока не будут обработаны все наборы параметров оптимизации;
- копируется на супер-ЭВМ вариант расчетной задачи.

Каждая поставленная в очередь исполнения задача выполняется, а набор искомых параметров сохраняется в отдельный для каждой задачи выходной файл.

Дерево проекта в файловой системе супер-ЭВМ представлено на рис. 4.

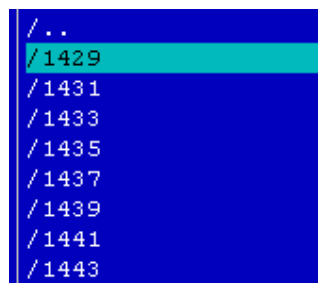


Рис. 4. Дерево проекта в файловой системе супер-ЭВМ

Модуль ProjectDirectory анализирует каталог каждой поставленной в очередь исполнения задачи на супер-ЭВМ и при появлении выходного файла с набором интегральных величин копирует его на локальную машину в каталог проекта.

Алгоритм построения геометрической и расчетной модели, а также алгоритм построения задачи при решении параметрических задач аналогичны алгоритмам при решении оптимизационных задач.

Графическое дерево объектов

Графическое дерево объектов предоставляет интерфейс для взаимодействия со структурой данных, начиная от геометрии (структуры) изделия, его параметров, результатов расчетов каждого оптимизационного шага и кончая результатами расчетов. Пример графического дерева объектов изображен на рис. 5.

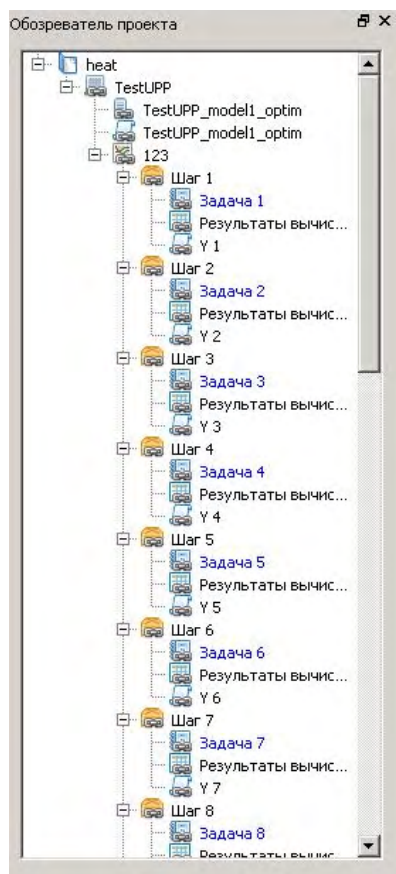


Рис. 5. Графическое дерево объектов

Реализованный интерфейс позволяет выполнять следующие операции:

- добавление/редактирование/удаление объекта в структуре данных;
- клонирование/копирование объектов в структуре данных;
- запуск сторонних приложений для определенных типов данных;
- доступ к конкретному элементу структуры;
- просмотр результатов задачи, включая результаты на каждом расчетном шаге.

Графическое дерево объектов реализовано на базе Qt с использованием QStandardItemModel и QTreeView.

Заключение

В докладе описана подсистема управления расчетными данными Интегратора ЛОГОС МИП, алгоритмы работы и принципы взаимодействия модулей, входящих в ее состав. Реализованная подсистема предоставляет:

эффективную организацию и хранение расчетных данных задач, решаемых в Интеграторе ЛОГОС МИП;

удобное для пользователей взаимодействие с расчетными данными в процессе расчета задачи.

Литература

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно-ориентированного проектирования. СПб.: Питер. 2012.