

МЕХАНИЗМ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ МОДУЛЬНОЙ ИНТЕГРАЦИОННОЙ ПЛАТФОРМЫ ЛОГОС

А. А. Тюндина, Н. А. Побуринная, Д. А. Кожаев

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

Введение

Пользовательская функция (ПФ) – это функция обратного вызова, разработанная пользователем, вызываемая расчетным модулем в процессе расчета физической задачи. Механизм ПФ предоставляет возможность изменять поведение существующего расчетного модуля (solver).

Модульная интеграционная платформа ЛОГОС (ЛОГОС-МИП) предназначена для решения ряда задач в рамках пакета программ «ЛОГОС», в том числе для динамического подключения пользовательских функций (ПФ) для расчетных задач и для внесения изменений в стандартные параметры задачи в процессе моделирования.

В основе механизма ПФ ЛОГОС-МИП лежит механизм функций обратного вызова – в расчетном модуле реализуются коннекторы ПФ, которые инициализируются в процессе расчета ПФ из динамических библиотек, после чего вызываются в определенных местах расчетного модуля.

Коннектор ПФ – это указатель на функцию, с помощью которого производится вызов пользовательской функции, в которой может быть сохранен адрес любой пользовательской функции с соответствующим интерфейсом.

Механизм ПФ ЛОГОС-МИП обладает следующими особенностями:

- настройка подключения ПФ происходит с помощью графического интерфейса ЛОГОС-МИП;
- отсутствие ограничения по наименованию динамических библиотек;
- отсутствие ограничения по наименованию ПФ и их размещению в динамических библиотеках;

– загрузка динамических библиотек в момент расчета и связывание ПФ с расчетным модулем происходит с помощью модуля сопряжения ЛОГОС-МИП.

Описание механизма пользовательских функций ЛОГОС-МИП

Общая схема механизма ПФ ЛОГОС-МИП представлена на рис. 1.

Для использования механизма ПФ ЛОГОС-МИП необходимо:

- определить API коннекторов ПФ;
 - создать коннекторы ПФ;
 - определить места вызовов ПФ;
 - вызвать ПФ, используя коннекторы ПФ;
 - описать API коннекторов ПФ с помощью тегов Doxygen [1] в исходных кодах расчетного модуля;
 - создать SPEC-файл (файл спецификации XML-формата [2]) с описаниями API коннекторов ПФ с помощью специальной утилиты, входящей в состав ЛОГОС-МИП;
 - разработать ПФ в отдельной динамической библиотеке, с учетом API коннекторов ПФ расчетного модуля;
 - описать API ПФ с помощью тегов Doxygen в исходных кодах ПФ;
 - создать SPEC-файл с описаниями API ПФ с помощью специальной утилиты, входящей в состав ЛОГОС-МИП.
- Настройка подключения ПФ к задачам расчетного модуля происходит с помощью графического интерфейса Интегратора, входящего в состав ЛОГОС-МИП.

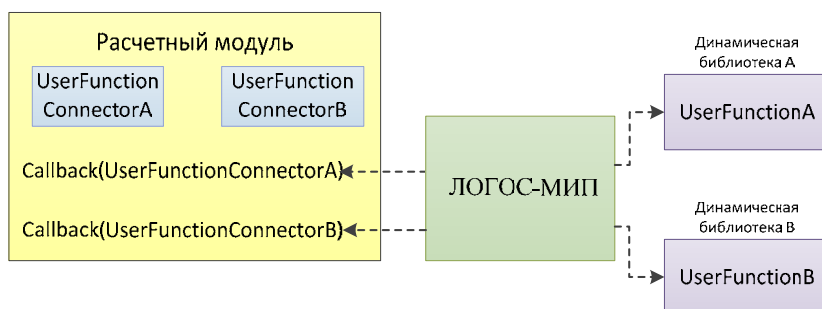


Рис. 1. Общая схема ПФ ЛОГОС-МИП

Для настройки подключения ПФ к задачам расчетного модуля необходимо передать в Интегратор SPEC-файлы коннекторов ПФ расчетного модуля и SPEC-файлы используемых ПФ, как показано на рис. 2.

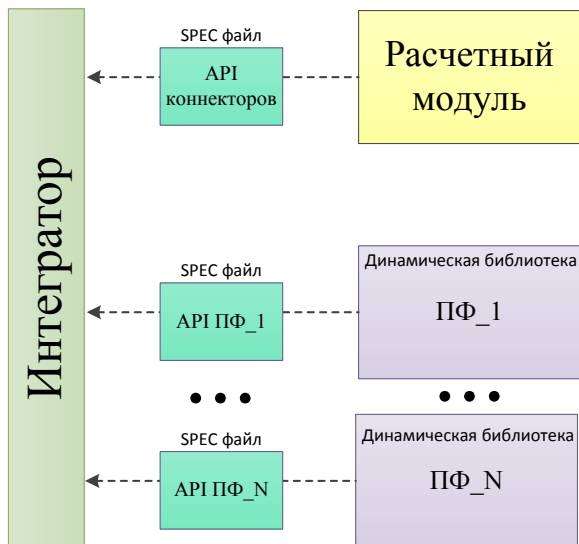


Рис. 2. Передача SPEC-файлов в Интегратор

Используя доступные SPEC-файлы, Интегратор предоставляет пользователю возможность конфигурировать соединение расчетного модуля и ПФ. На основе настройки параметров взаимодействия расчетного модуля с ПФ, Интегратор создает YAML-файл [3] конфигурации `scf.yaml` с описанием связей коннекторов и соответствующих ПФ.

Общая схема конфигурирования ПФ ЛОГОС-МИП представлена на рис. 3.

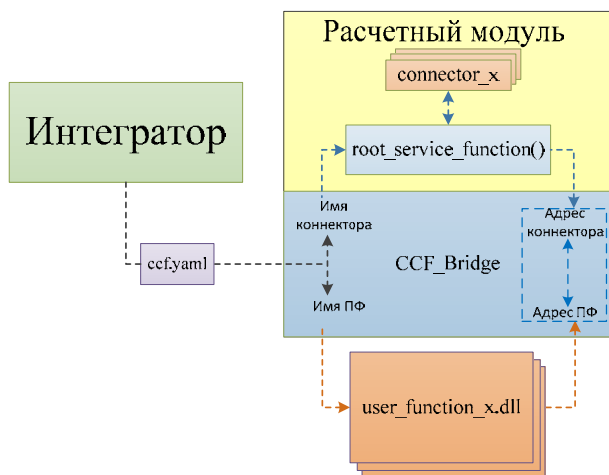


Рис. 3. Общая схема конфигурирования ПФ ЛОГОС-МИП

Интегратор передает файл конфигурации `scf.yaml` в модуль `CCF_Bridge`. При запуске приложения расчетного модуля инициализируется модуль `CCF_Bridge` и ему передается указатель на функцию `root_service_function`.

Модуль `CCF_Bridge` передает в функцию `root_service_function` имя используемого коннектора ПФ, и получает из этой функции указатель на соответствующий коннектор.

Модуль `CCF_Bridge` загружает нужные динамические библиотеки с ПФ, и инициализирует указатели коннекторов ПФ адресами соответствующих ПФ. Во время расчета задачи расчетный модуль вызывает ПФ, обращаясь к проинициализированным коннекторам ПФ.

Создание SPEC-файлов

SPEC-файл – это файл спецификации XML формата, используемый Интегратором для получения информации о различных модулях.

Для формирования SPEC-файлов исходные коды ПФ и коннекторов ПФ должны быть описаны с помощью тегов `Doxygen`. Для описания ПФ и коннекторов ПФ были созданы теги `user_function` и `user_function_connector`, соответственно. С помощью тегов `Doxygen` можно дать описание функции и ее интерфейса.

SPEC-файл с описанием API ПФ создается с помощью специальной утилиты `specgen`, входящей в состав ЛОГОС-МИП. Алгоритм подготовки SPEC-файлов представлен на рис. 4.

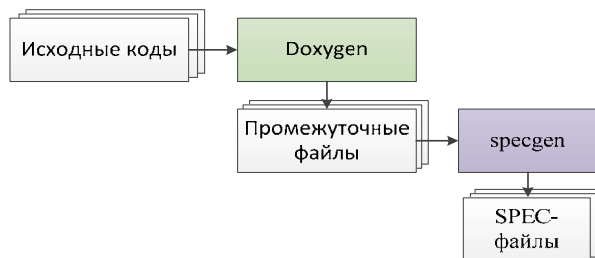


Рис. 4. Алгоритм подготовки SPEC-файлJD

Исходные коды на первом этапе поступают на вход в программу `Doxygen`. В результате работы `Doxygen` формирует промежуточные файлы, которые передаются в утилиту `specgen`. В результате работы утилита `specgen` создает SPEC-файлы.

Для формирования SPEC-файлов исходные коды ПФ и коннекторов ПФ должны быть описаны с помощью тегов `Doxygen`. Для описания ПФ и коннекторов ПФ были созданы теги `user_function` и `user_function_connector`, соответственно. С помощью тегов `Doxygen` можно дать описание функции и ее интерфейса.

В SPEC-файл ПФ записывается следующая информация:

- имя библиотеки, в которой находится ПФ;
- имя ПФ;
- описание ПФ;
- интерфейс ПФ и описание к нему.

В SPEC-файл расчетного модуля записывается следующая информация о коннекторах ПФ:

- имя коннектора ПФ;
- описание коннектора ПФ;
- интерфейс коннектора ПФ и описание к нему.

Графический интерфейс настройки подключения ПФ

Настройка подключения ПФ к задачам расчетного модуля происходит с помощью графического интерфейса Интегратора.

Используя доступные SPEC-файлы, Интегратор предоставляет пользователю возможность конфигурировать для расчетной задачи соединение расчетного модуля и ПФ.

Конфигурирование соединения коннекторов ПФ расчетного модуля и ПФ в Интеграторе происходит в специальном диалоге настройки ПФ, в котором пользователь из предложенного списка может выбрать соответствие коннектора ПФ расчетного модуля и доступной ПФ.

Список доступных коннекторов ПФ и ПФ формируется исходя из информации, полученной из спецификаций доступных расчетных модулей и ПФ.

Для каждой задачи пользователь выбирает соответствие коннекторов ПФ и ПФ. Пример графического интерфейса настройки подключения пользовательских функций к расчетному модулю показан на рис. 5.

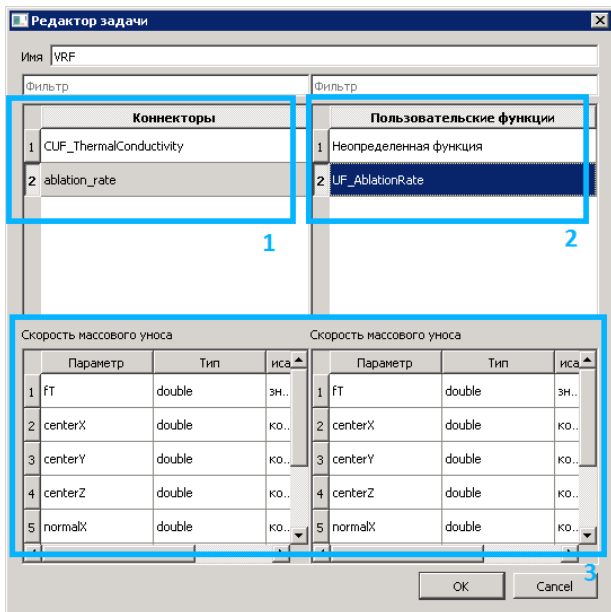


Рис. 5. Внешний вид диалога настройки пользовательских функций

Диалог настройки пользовательских функций и состоит из:

- таблицы коннекторов ПФ (1), которая формируется «Интегратором» на основе спецификации расчетного модуля решаемой задачи;

- таблицы подключаемых к коннекторам ПФ (2), которая формируется «Интегратором» для каждого коннектора на основе спецификации пользовательских функций;

- таблиц описания коннектора/ПФ (3).

Чтобы настроить связь «коннектор-ПФ», пользователю необходимо выбрать коннектор и ПФ, соответствующую ему по интерфейсу. «Интегратор» запоминает выбранный элемент. Для каждого коннектора возможен выбор только одной пользовательской функции из списка доступных ПФ.

После завершения конфигурации, Интегратором создается список связей, типа «коннектор – ПФ», который сохраняется в структуры данных «Интегратора».

На основе настройки параметров взаимодействия расчетного модуля с ПФ, Интегратор создает файл конфигурации `scf.yaml`, который является неотъемлемой частью подготовленной задачи.

В файле `scf.yaml` для каждой пары соответствия коннектора и ПФ Интегратором формируются следующие поля:

- `dll_name` – имя динамической библиотеки, в которой расположена ПФ;
- `connector_name` – имя коннектора ПФ;
- `function_name` – имя ПФ.

Подготовленный файл конфигурации передается модулю `CCF_Bridge` и используется им на стадии инициализации расчета.

Модуль сопряжения CCF_Bridge

Связывание ПФ и расчетного модуля обеспечивает модуль `CCF_Bridge`. Модуль `CCF_Bridge` реализован в виде статической библиотеки, которая связывается (линкуется) с исполняемым файлом расчетного модуля на стадии сборки приложения.

Основными задачами модуля сопряжения в рамках использования механизма ПФ функций являются:

- загрузка YAML-файла с информацией о динамических библиотеках подключаемых ПФ и связей коннекторов с соответствующими ПФ;
- загрузка динамических библиотек с подключаемыми ПФ;
- получение из расчетного модуля указателей на коннекторы и связывание их с соответствующими ПФ.

Заключение

Реализованный механизм ПФ ЛОГОС-МИП обладает следующими особенностями:

- настройка механизма ПФ производится с помощью графического интерфейса ЛОГОС-МИП;
- отсутствие ограничения по наименованию ПФ и их размещению в динамических библиотеках;

– загрузка динамических библиотек и связывание ПФ с расчетным модулем происходит с помощью модуля CCF_Bridge.

Применение механизма ПФ ЛОГОС-МИП позволяет расширить возможности расчетного модуля, например:

– позволяет инициализировать различные параметры в момент начала расчета (поля скоростей, давления, температур и т. д.);

– позволяет изменять значения различных параметров в процессе расчета (значения плотности, коэффициента теплопроводности, скорости массового уноса материала и т. д.).

Литература

1. Doxygen [Electronic resource]: standard tool for generating documentation / Dimitri van Heesch. [1997–2016]. Mode of access: <http://www.stack.nl/~dimitri/doxygen/manual/xmlcmds.html>.

2. XML [Electronic resource]: standard of XML document / World Wide Web Consortium. [2000–2007]. Mode of access: <https://www.w3.org/XML>.

3. YAML Ain't Makeup Language [Electronic resource]: data serialization standard / Evans Clark. [S.l.], YAML 10, 2009. Mode of access: <http://yaml.org/spec/1.2/spec.html>