ИСПОЛЬЗОВАНИЕ СКРИПТОВОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ ДЛЯ СОЗДАНИЯ «ВНЕШНЕЙ» ФУНКЦИОНАЛЬНОСТИ РАСЧЕТНОГО КОДА

С.Л. Головков, А.В. Осипов

При разработке системного обеспечения расчетных кодов практически всегда возникает проблема создания «внешней» функциональности. Суть ее состоит в том, что конечным пользователям требуется возможность самостоятельно изменять (естественно, в определенных пределах) функциональность системы. Типичные примеры:

- если система подготовки данных осуществляет проверку (верификацию) исходных данных на предмет полноты, непротиворечивости, соответствия методики расчета и т.п., то пользователю может потребоваться изменение каких-то проверок, изменение порядка проверок, удаление ненужных проверок, добавление новых проверок и т.д.
- пользователю может требоваться программирование дополнительных пользовательских функций для задания нестандартных внешних полей, сторонних токов, источников и т.д.
- если система анализа результирующих данных вычисляет какие-либо функционалы, то пользователю может потребоваться изменение алгоритмов вычисления, добавление новых функционалов и т.д.

Конечно, все это можно сделать, модифицируя код системы. Однако это сделать не всегда просто:

- для модификации кода системы необходимо иметь полноценную среду разработки – компиляторы, отладчики и т.п.;
- код может быть весьма сложен для понимания и, тем более, для изменения, и квалификации конечного пользователя, как правило, недостаточно для модификации кода;
- модификация кода чревата внесением ошибок, способных «разрушить» всю систему.

Возможное решение состоит в выносе подобной функциональности «за рамки системы», иначе говоря, реализация этой функциональности с помощью скриптов на подходящем языке. Система должна уметь передавать в скрипты параметры и получать обратно результаты выполнения скриптов. Тогда пользователь получает возможность самостоятельно менять (конечно, в заданных пределах) поведение системы.

В статье описывается опыт использования языка Lua при создании системы формирования расчетных моделей (Конструктор) для кода TRIANA-4 [1,2,3], предназначенного для выполнения теплогидравлических расчетов реакторных установок с жидкометаллическим теплоносителем.

На этапе формирования расчетной модели требуется выполнить большой объем проверок параметров текущей модели на их допустимость и соответствие друг другу. Пример проверки: сумма произведений длин контрольных объемов на синусы угла наклона оси канала должны равняться нулю по всем замкнутым контурам.

Пользователям требуется периодически менять набор и содержание таких проверок в соответствии с методиками расчетов. Для того чтобы удовлетворить эти требования, проверки реализуются с помощью «внешней» функциональности, программируемой на скриптовом языке Lua. В самом же коде «зашиты» проверки только самых «тяжелых» ошибок, делающих невозможным представление схемы, и самых «легких», которые система способна исправить самостоятельно.

Скрипты, выполняющие проверки, представляют собой обычные текстовые файлы, которые не требуют компиляции и вызываются непосредственно из самой системы. Для изменения логики достаточно просто изменить соответствующую программу - текстовый файл. Причем такое изменение может осуществляться даже во время работы системы.

Скрипт, выполняющий некую проверку, можно рассматривать как записанное в процедурной форме утверждение о требуемом свойстве модели.

Следовательно, расширяемое множество подобных скриптов можно рассматривать как некую «базу знаний», аккумулирующую представления пользователей о требуемых свойствах моделей.

Lua - нетрудный в освоении скриптовый язык программирования, разработанный в Католическом университете Рио-де-Жанейро; является свободно распространяемым, с открытыми исходными текстами на языке Си. Выбор этого языка обусловлен следующими причинами: интерпретатор Lua работает быстрее и требует меньше памяти по сравнению с интерпретатором более распространенного языка Python, обладает более простым API между скриптом и C/C++ и, самое главное, Lua имеет более простой синтаксис, следовательно, конечным пользователям – конструкторам и физикам - будет легче пользоваться языком для самостоятельной разработки скриптов.

ЛИТЕРАТУРА

1. С.Л. Головков, А.В. Осипов. Разработка и реализация Конструктора расчетной модели для теплогидравлического кода «TRIANA» (версия 1.0). Руководство пользователя // Препринт ИПМ им. М.В.Келдыша РАН. 2011, № 28, 32 с., ISSN 2071-2898 URL: http://library.keldysh.ru/preprint.asp?id=2011-28
2. А.В. Осипов. Создание средств разработки расчетных моделей для моделирования теплогидравлических процессов в ядерных реакторах с жидкометаллическим теплоносителем // Научный сервис в сети Интернет: экзафлопсное будущее: Труды Международной суперкомпьютерной конференции (19-24 сентября 2011 г., г. Новороссийск). – М.: Изд-во МГУ, 2011. с. 603-614, ISBN 978-5-211-06229-0 URL: http://agora.guru.ru/abrau2011/pdf/603.pdf
3. А.В. Осипов. Создание средств разработки расчетных моделей для моделирования теплогидравлических процессов в ядерных реакторах с жидкометаллическим теплоносителем // Вычислительные методы и программирование. 2012. Т. 13. Раздел 2. Программирование, с.1-7. URL://http://num-meth.srcc.msu.su