
СИСТЕМА СОДЕРЖАТЕЛЬНОГО АНАЛИЗА ПРОГРАММ, ПРЕДНАЗНАЧЕННАЯ ДЛЯ ИНФОРМАТИЗАЦИИ ОБУЧЕНИЯ БУДУЩИХ ПРОГРАММИСТОВ И КВАЛИФИЦИРОВАННЫХ ПОЛЬЗОВАТЕЛЕЙ

М.Ю. Чернышов

Президиум Иркутского научного центра СО РАН
ул. Лермонтова, 134, Иркутск, Россия, 664033

Обсуждается государственная постановка вопроса информатизации образования и некоторые дидактические аспекты этой задачи. Предложен программный комплекс, обеспечивающий эффективную поддержку процесса обучения будущих программистов и квалифицированных пользователей приемам практического анализа содержания программ и наглядность каждой аналитической операции.

Ключевые слова: информатизация образования, информационные технологии, программный продукт, обучение.

Дидактические аспекты информатизации образования

В начале второго десятилетия XXI в. едва ли нужно доказывать кому-либо необходимость информатизации образования. Информатизация является потребностью, диктуемой временем. Уже сегодня подавляющее большинство вузов и школ оснащены самыми современными компьютерными классами. Во многих передовых вузах и школах введены системы компьютеризированного контроля обучения, контроля успеваемости, а также системы ведения отчетности преподавателей. Компьютерные системы повсеместно используются для автоматизированной охраны учебных заведений.

С учетом степени уже существующей компьютеризации есть возможность широко внедрять информационные и телекоммуникационные технологии. В последние годы они все шире используются в образовании, в сфере научных исследований, провидимых в вузах и средних технических учебных заведениях. Не случайно в Послании Президента РФ Д.А. Медведева Федеральному собранию Российской Федерации (2009) главная дидактическая цель обучения определена как «раскрытие способностей обучаемого, воспитание личности, готовой к жизни в высокотехнологичном, конкурентном мире».

Обучение должно способствовать личностному росту так, чтобы выпускники смогли самостоятельно ставить серьезные цели и достигать их, уметь реагировать на различные жизненные ситуации [3]. Такая постановка цели и задач представляется правильной с различных точек зрения. Во-первых, интеллектуальный потенциал общества, служащий фактором его научного прогресса, экономического и социального развития, в значительной степени зависит от качества образования в отечественных школах, профессиональных училищах и вузах. Во-вторых, будущее общества зависит от наличия в нем хорошо воспитанных и обученных граждан, которым предстоит жить в мире, где степень сложности технологий будет

лишь возрастать со временем, а к обучаемым и технологиям обучения будут предъявляться все более высокие требования.

Существенную роль в подготовке специалистов высокой квалификации в отечественных вузах играет наличие качественного передового программного обеспечения. Это приобретает особое значение при подготовке будущих программистов и специалистов в других областях знания, которые станут, кроме прочего, квалифицированными пользователями программ и программных комплексов, необходимых в их исследованиях. Но сегодня на рынке практически нет ни отечественных, ни зарубежных программных комплексов, на которых можно было бы обучать и тренировать программистов, например, в умении анализировать существующие программные модули и программы или в синтезе (построении) новых программ и программных комплексов.

Решение задачи, связанной с созданием таких технологий разработки программных комплексов (и, как следствие, создание эффективных комплексов), тормозится, как это ни парадоксально, из-за нерешенности задач, в числе которых — общий дефицит эффективных технологий создания программных продуктов, которые характеризовались бы высокой надежностью и гибкостью с точки зрения потребности их оперативной адаптации, например к изменившимся потребностям пользователей. Для того, чтобы обеспечить эти важнейшие характеристики, нужно, чтобы создаваемые программы строились на неких единых и универсальных принципах. Но пока это не так. Поэтому анализ программ составляет проблему. И обучение анализу программ — тоже проблема.

На сегодняшний день в мире практически отсутствуют программные комплексы, предназначенные для анализа программ, синтеза программ или какие-либо иные программные системы, которые были бы построены на основе гибких логико-смысловых моделей и, следовательно, могли бы быть легко (оптимально по времени) изменены или полностью перестроены с учетом возникших новых потребностей и задач.

Программный комплекс, предназначенный для поддержки обучения приемам практического анализа содержания программ

В рамках большого проекта РАН была поставлена задача создания гибких аналитических программных комплексов высокой производительности, которые могли бы быть использованы как в практическом анализе программ и программных систем, так и в обучении будущих программистов.

Очевидно, что для решения поставленной задачи необходимо добиваться того, чтобы создаваемые программы строились на неких единых и универсальных принципах. Согласно сложившимся на сегодня представлениям, это могли бы быть единые принципы модельной логико-смысловой организации программ или, иначе говоря, принципы технологии программирующего моделирования. При таком подходе строится прозрачная программная система, и каждая из ее подпрограмм строится как органичный и легко вычленимый логико-смысловой элемент в единой логико-смысловой модели целостной программной системы.

Но пока для анализа множества уже существующих программ и программных систем нужен был эффективный инструмент автоматизированного анализа программ и программных систем, который мог бы быть также эффективным средством обучения будущих программистов. В целях обучения программистов и квалифицированных пользователей этот инструмент можно было бы применять в анализе доступных программ из государственного банка алгоритмов и программ (БАП).

С учетом постановки задач нами был разработан подход, который предполагал следующее. Во-первых, для решения поставленной задачи была использована созданная нами оригинальная инструментальная среда программирования ЗИРУС [1]. На ее базе был разработан универсальный метод, который предполагал семантическую интерпретацию анализируемых компьютерных программ в виде моделей данных и наборов отношений [5]. Он был основан на общих принципах теории построения баз данных и рассчитан на формирование и перестройку внешних текстовых форм программ и получение различных их графических отображений. Применение теории графов в качестве аппарата модельного отображения построенной программной системы позволяет, как выяснилось, существенно упростить интерпретацию сложных структур и алгоритмов анализируемых программ [2]. Это имеет большое значение с точки зрения обучения.

На основе указанного метода, а также с помощью теории графов сначала был разработан программный комплекс, позволявший осуществлять наглядный анализ программ, даже если они имеют большой размер. Такой анализ осуществлялся путем построения графа функциональных связей в анализируемой программе по ее исходному коду, что уже обеспечивало наглядную (т.е. удобную при обучении) и эффективную навигацию обучаемого аналитика по исходным файлам с выявлением средствами системы и наглядной демонстрацией основных свойств программы (типов, методов, функций, использованных в программе). Это также позволяло подобрать возможные способы корригирующего преобразования исходного текста. Общую постановку задачи по преобразованию исходных текстов программ можно представить в виде следующей схемы (рис. 1).

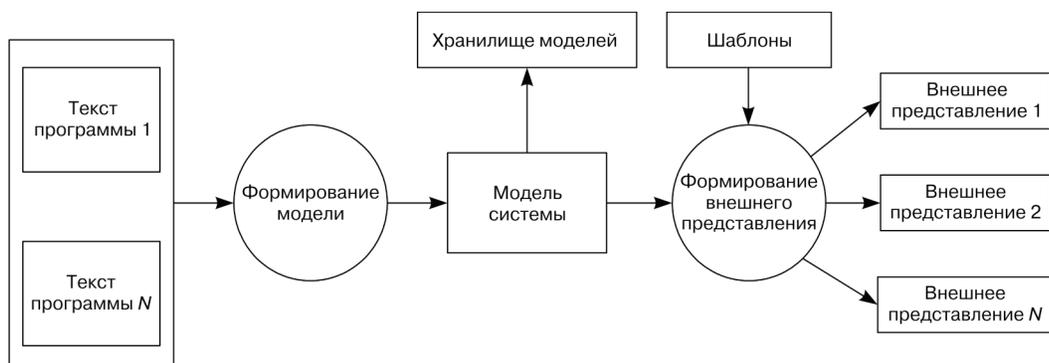


Рис. 1. Схема процесса анализа программной системы, построения ее модели и отображения этой модели в различные ее внешние представления

Разработанный программный комплекс с самого начала давал возможность формировать различные внешние представления исходного текста, а также различные графические представления анализируемой программы. Это исключительно удобно с точки зрения обучения. В дальнейшем был разработан программный комплекс, предназначенный для анализа широкого класса программных систем и их модулей. Такой анализ был ориентирован уже на оценку возможности их повторного использования во вновь создаваемых программных продуктах. И это была еще одна задача, важная с точки зрения обучения.

Важным в аспекте обучения программистов является то, что созданный программный комплекс предполагает возможность обращения в хранилище моделей ранее проанализированных программ и на основе этих моделей выполнить статический анализ исходного текста анализируемой программы и в итоге сформировать по задаваемым учебным шаблонам графы связей различного внешнего представления и различной степени детализации (рис. 2).

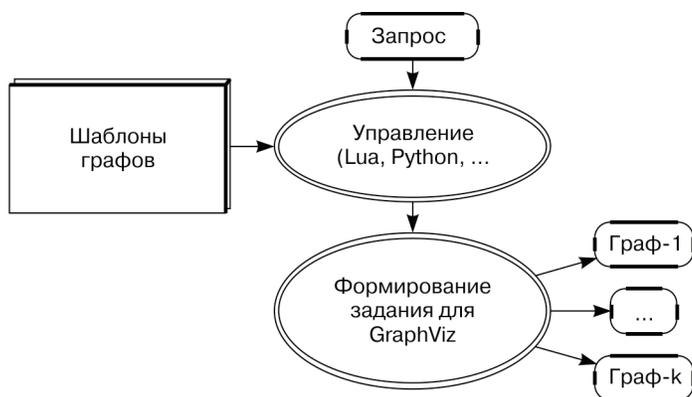


Рис. 2. Технологический принцип статического анализа исходных текстов программ (по существу это — технология формирования графов связей)

Рис. 2 детализирует общий технологический принцип анализа (см. рис. 1), осуществляемого в процессе функционирования программного комплекса, рассчитанного на статический анализ кода. На основе этого принципа и была реализована оригинальная технология автоматического формирования графов связей в текстах анализируемых программ и программных систем. Логический аппарат функционирования статического анализатора программного комплекса написан нами на скриптовых языках Lua и Python.

Необходимо пояснить, что означает утверждение о том, что разработанное программное средство использует статический анализ кода (САК). Заметим, что данный термин обычно применяют к анализу, производимому специальным программным обеспечением. САК производится без реального выполнения исследуемой программы, в большинстве случаев — над версией исходного кода, хотя в принципе анализу может быть подвергнут какой-нибудь один вид объектного кода.

Если оказывается возможной декомпозиция системы на основании некоторого принципа или правила (например, правила применения кластеризации к графу), то можно найти подсистемы, соответствующие этому принципу/правилу (кластеру). Такой подход к анализу архитектуры объектных программных систем дает возможность не только упростить понимание внутреннего устройства программной системы (ПС), но и осуществить ее рефакторинг, т.е. выполнить моделирование архитектуры программной системы с целью ее улучшения. Это позволяет, например, избежать неоправданного дублирования подсистем ПС, обнаружить в ней сильно связанные компоненты и т.п.

Возможности, предоставляемые статическим анализом исходного текста анализируемой программной системы

В последние годы статический анализ все чаще используется при исследовании и проверке надежности и свойств программного обеспечения, применяемого в современных вычислительных системах высокой производительности. Уже поэтому его роль в обучении может быть велика. Например, с точки зрения обучения важно то, что обучаемый может отслеживать процесс статического анализа и видеть, как выполняется исследование архитектуры объектной ПС, ее объектов и связей между объектами. Дело в том, что на основании анализа связей между объектами возможна декомпозиция ПС на подсистемы (она упрощает дальнейший анализ ПС за счет уменьшения количества объектов архитектуры и связей между ними на каждом конкретном этапе анализа системы или ее части). Кроме того, процесс анализа предполагает получение данных, необходимых для построения модели ПС в форме графа (причем последняя немедленно помещается в хранилище (базу) моделей как один из модельных эталонов, который позднее может потребоваться для сравнения как прецедент).

С точки зрения обучения важно и то, что информация, полученная в ходе анализа, и построенная модель используются для оперативного выявления мест в ПС, которые содержат ошибки (если таковые есть); определения основных свойств ПС и, главное, для определения функционально-смыслового назначения ПС или ее соответствия функционально-смысловому назначению, указанному в спецификации. Заметим, что в зависимости от целей и, как следствие, инструмента анализа, используемого в каждом конкретном случае, глубина анализа может варьировать от определения поведения (функций) отдельных операторов программы до анализа всего исходного кода.

Во время работы анализатора возможно применение предустановленных шаблонов различных представлений графов, а также дополнение графов новыми кластерами. Заключительный этап формирования графа заключается в создании текстового описания графа и передаче его компоненту Graphviz, который выполняет функции создания графического отображения графа.

В выступлении на Петербургском экономическом форуме-2010, а позднее в Послании Президента РФ Федеральному собранию (2010) Президент РФ Д.А. Медведев

дев существенно конкретизировал постановку задачи, связанной с инновациями: «Мы должны изменить нашу страну, мы должны внедрить новые технологии, мы должны поменять культуру производства... Мы должны создать технологический приоритет. Мы должны перевести нашу экономику на рельсы инноваций, т.е. высоких технологий» [4. С. 28].

В рамках реализованного нами проекта, задачей которого первоначально была лишь интерпретация исходного кода анализируемой программы, разработана технология комплексного анализа широкого класса программных систем и их модулей в целях проверки качества этих систем, их свойств и, главное, их функционально-смыслового (интенционального) назначения, что важно, если необходимо использовать уже написанные программы для разработки вновь создаваемых программных систем.

Создана модель обучающего комплекса, который позволит эффективно (за счет использования программного комплекса, обеспечивающего наглядность всех осуществляемых им аналитических операций) обучать будущих программистов и системных аналитиков приемам высокотехнологичного анализа программных систем.

Перечень наглядных аналитических операций, обеспечиваемых созданным нами программным комплексом и позволяющих быстро и качественно выполнять анализ программ большого размера посредством построения графов отношений, обеспечивает:

- быструю навигацию по исходным файлам;
- выявление свойств анализируемой программы (типов, методов, функций, используемых в ней);
- выявление ошибок в программе;
- определение функционально-смыслового назначения программы.

Программный комплекс рассчитан на формирование различных внешних форм представления текста программы и различных его графических отображений. Иными словами, он позволяет строить модели программной системы, получая при этом различные формы ее внешнего представления, что весьма удобно для обучения программистов и квалифицированных пользователей.

Заметим, что достоинства технологии, лежащей в основе созданного программного комплекса, не ограничиваются лишь возможностями анализа. Усовершенствованная технология и новое программное средство, создаваемое на ее основе, позволят также осуществлять автоматическую модификацию кода программы, т.е. реализовывать различные способы автоматического преобразования исходного текста программы.

ЛИТЕРАТУРА

- [1] *Абасов Н.В.* Основы универсальной среды программирования ЗИРУС // Вестник ИрГТУ. — 2006. — № 2 (26). — С. 62—68.
- [2] *Касьянов В.Н., Евстигнеев В.А.* Графы в программировании: обработка, визуализация и применение. — СПб.: БХВ-Петербург, 2003.

- [3] Послание Федеральному собранию РФ: Выступления и стенограммы. — URL: <http://www.kremlin.ru/transcripts/5979>
- [4] Послание Президента РФ Д.А. Медведева Федеральному собранию РФ // Вестник российской нации. — 2010. — № 5—6. — С. 10—29.
- [5] *Chernyshov M.Yu., Bychkov I.V.* On the Way to a Sense-Analyzing Intelligent System Oriented to Earth and Solar Sciences. Part 1: The Principles of Investigation and the Logic of Mapping // *Data Science Journal*. — 2009. — Issue. 2.

A COMPLEX OF SOFTWARE SEMANTIC ANALYSIS INTENDED FOR INFORMATIZATION OF TRAINING FUTURE PROGRAMMERS AND EXPERIENCED USERS

M.Yu. Chernyshov

Presidium of Irkutsk Scientific Center
Siberian Branch of RAS
Lermontov Str., 134, Irkutsk, Russia, 664033

The government statement of the issue of education informatization and some didactic aspects of this complicated problem are discussed. A software complex, which provides for training future programmers and experienced users in techniques of practical analysis of the content of programs and visualization of each analytical operation, is proposed.

Key words: informatization of education, information technology, software product, education.