

Секция «Вычислительная математика и кибернетика»

Среда языково-ориентированного программирования с инструментарием поддержки понимания программ

Мальцев Иван Юрьевич

Аспирант

Национальный исследовательский ядерный университет «МИФИ», Факультет кибернетики, Москва, Россия

E-mail: imaltsev87@gmail.com

На сегодняшний день проблемы проектирования и разработки сложных программных систем являются хорошо изученными. Ключевым и одним из фундаментальных факторов, затрудняющих разработку программного обеспечения, является сложность. Сложность самой программы можно интерпретировать как сложность понимания программного кода.

В последнее время активно развиваются две схожие парадигмы, направленные на преодоление сложности разработки программных систем: языково-ориентированное программирование [2] и фабрики программного обеспечения [1]. Оба принципа предусматривают упрощение процесса написания программ при помощи разработки предметно-ориентированных языков (domain-specific language, DSL) [2]. DSL позволяет описывать решаемую проблему с помощью высокоуровневых абстракций предметной области и привлекать к разработке системы специалистов, не имеющих обширных знаний языков программирования общего назначения.

Перечисленные выше вопросы давно волнуют как сообщество разработчиков, так и компании, занимающиеся выпуском средств разработки. Тем не менее комплексные решения вышеуказанных проблем стали появляться относительно недавно. Класс программно-инструментальных средств поддержки языково-ориентированного программирования, получил название языковых инструментариев [4].

Из всех разработок в этой области наиболее совершенным и новаторским продуктом является Meta Programming System (MPS) [5] компании JetBrains. MPS – свободно-распространяемая программная система, большинство исходного кода которой доступно под лицензией Apache 2.0.

Внутренняя модель DSL в MPS состоит из нескольких языков [3], которые сами являются DSL.

1. Язык структуры используется для задания абстрактного синтаксиса программы.
2. Язык редактора задает конкретный синтаксис, а также расположение вложенных синтаксических блоков, составляющих модель редактора программы.
3. Язык трансформации описывает механизм перевода программы из исходного языка DSL в целевой или внутренний язык `baseLanguage`, т.е. компиляцию DSL.

Использование такого подхода при создании модели DSL позволяет получить преимущества, предоставляемые интеллектуальными средами разработки для языков программирования общего назначения, например: автодополнение кода, рефакторинг, подсветка синтаксиса и пр.

Однако по причине сложной структуры модели DSL эта система имеет ряд недостатков:

- порождает зависимость проекта от среды разработки;
- имеет ограниченную поддержку языков программирования общего назначения;
- отсутствие интеграции с инструментами, ориентированными на текстовое представление языков.

В рамках проекта ReWorker, основанного на исходном коде MPS, производится разработка среды языково-ориентированного программирования, лишённой описанных выше недостатков и включающей в себя инструменты поддержки понимания программ.

В среду ReWorker была добавлена поддержка языка C# помимо существовавшего в MPS Java. В результате анализа и сопоставления абстрактного синтаксиса двух языков были реализованы структура, редактор и транслятор для языка `abstractLanguage`. Таким образом, можно говорить, что `abstractLanguage` является сужающим синтаксическим отображением [3] одновременно для языков Java и C#. Транслятор для `abstractLanguage` назовём абстрактным вычислителем программ, написанных на этом языке.

Проведённые в среде эксперименты показывают возможность трансляции одной и той же программы или её частей между различными языками программирования. Это позволяет снизить зависимость ПО от языков программирования общего назначения.

Для устранения зависимости проекта от среды разработки проводится реализация инструментария описания языков при помощи КС-грамматик. Этот функционал позволит упростить импорт/экспорт моделей языков из среды ReWorker.

Чтобы понять семантику программы разработчику зачастую требуется провести определённые преобразования, направленные на её упрощение. Эти преобразования могут изменять поведение ПО, поэтому должны быть обратимыми и выполняться над специальной моделью программы. В качестве такой модели можно использовать редактор языка `abstractLanguage`, в который транслируется код программы. Чтобы упростить задачу преобразования модели необходимо разработать DSL, метапрограммы которого будут использовать редактор `abstractLanguage` в качестве объекта вычислений.

Литература

1. Гринфилд Дж., Шорт К., Кук Ст., Кент Ст., Крупи Дж. Фабрики разработки программ (Software Factories): потоковая сборка типовых приложений, моделирование, структуры и инструменты. М.: Диалектика, 2006.
2. Дмитриев С. Языково-ориентированное программирование: следующая парадигма // RSDN Magazine. 2005, №5.
3. Соломатин Д.И. Синтаксически расширяемые языки программирования как средство реализации языков предметной области // Вестник Воронежского государственного университета. 2008, №1.
4. Fowler M. Language Workbenches: The Killer-App for Domain Specific Languages? 2005: <http://www.martinfowler.com/articles/languageWorkbench.html>
5. Meta Programming System: <http://www.jetbrains.com/mps/>