

ВОЗМОЖНОСТИ И ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ СОВРЕМЕННОГО ЯЗЫКОВОГО ИНСТРУМЕНТАРИЯ В РАЗРАБОТКЕ НА ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ

Д.С. Ботов

В статье кратко рассматриваются общие для различных современных программных средств возможности, которые позволяют создавать и поддерживать предметно-ориентированные языки программирования. Среди возможностей рассматриваются возможности среды создания предметно-ориентированных языков и среды редактирования программ на таких языках. При рассмотрении возможностей сред редактирования основное внимание уделяется описанию проекционных сред редактирования, которые используются в современных языковых инструментальных средствах и позволяют привнести возможности интеллектуальных редакторов современных интегрированных сред разработки. В качестве вывода в данном кратком сообщении приводятся проблемы использования современных языковых инструментальных средств и разработки в языково-ориентированном стиле без их применения. Основное внимание в заключении уделено проблеме представления семантики предметно-ориентированных языков и отображения семантики на синтаксис. Делается предположение о возможности использования различных способов представления знаний для описания семантических моделей предметно-ориентированных языков.

Ключевые слова: предметно-ориентированный язык программирования, языковой инструментарий, языково-ориентированное программирование, семантическая модель, проекционный редактор.

Введение

Сегодня в промышленной разработке программного обеспечения крайне важную роль в повышении производительности труда программистов играют интеллектуальные интегрированные среды разработки. При активном применении в процессе разработки языково-ориентированного стиля программирования, который основывается на использовании предметно-ориентированных языков программирования (domain-specific language, DSL), эффективность работы разработчиков также будет зависеть от наличия программных средств для создания и поддержки предметно-ориентированных языков. Такие средства называют **языковым инструментарием** (языковыми инструментальными средствами, language workbench [1]).

Среди возможностей современных языковых средств, которые могут быть различными по особенностям реализации и предоставляемой функциональности, можно выделить следующие:

1. Возможности среды создания DSL:
 - a. Определение семантической модели.
 - b. Определение синтаксиса.
 - c. Определение правил отображения семантики на синтаксис.
2. Возможности среды редактирования программ на DSL (и возможности создания таких сред для разрабатываемых DSL).

Возможности среды создания DSL

В данном пункте наиболее подробно остановимся на проблеме определения семантической модели, как наиболее сложной и важной составляющей DSL.

М. Фаулер [1], говоря об определении семантической модели предметно-ориентированных языков, разделяет определение схемы семантической модели (структуры данных семантической модели, статической семантики) и поведения семантической модели, то есть того, каким образом будет выполняться семантическая модель, наполненная данными из конкретного сценария на DSL. Выполнение семантической модели может быть реализовано как непосредственно, само-

стоятельно (по аналогии с интерпретацией), так и путем генерации выполняемого кода (например, на некотором скриптовом языке – по аналогии с компиляцией).

Важно отметить, что так как структура семантической модели определяется предметной областью предметно-ориентированного языка, то в основе ее может лежать альтернативная модель вычисления [1] или парадигма программирования (альтернативная императивной парадигме языка программирования общего назначения, на котором разрабатывается программная система). В качестве альтернативных вычислительных моделей и парадигм программирования могут выступать, например, модель конечного автомата, таблицы решений, системы правил вывода ЕСЛИ...ТО, сеть зависимостей, декларативное описание конфигурации и т. д.

Возможности среды редактирования программ на DSL

Важной особенностью сред редактирования сценариев на создаваемых DSL является использование *проекционной системы редактирования*, а не редактирования непосредственно исходных текстов [1]. При редактировании на основе исходных текстов редактируемое текстовое представление не привязано к инструментарию и в качестве редактора сценариев на DSL, по сути, можно использовать любой текстовый редактор, а редактируемое представление совпадает с хранимым (текстовый файл). При проекционном редактировании хранимое представление семантической модели для конкретного сценария редактируется с помощью специального инструментария, причем для разных элементов сценария могут быть использованы различные проекции (представления). Они могут быть как в привычном текстовом виде, так и в виде графов, форм, таблиц, схем и т.д. Каждый вид представления отражает один из конкретных синтаксисов созданного DSL.

При проекционном редактировании сам предметно-ориентированный язык и среда редактирования сценариев на DSL представляют собой единое неразрывное целое. В качестве примера систем проекционного редактирования, рассчитанных на конечных пользователей, можно рассматривать инструментарий для работы с базами данных Microsoft Access или редактор электронных таблиц Microsoft Excel. В качестве примера полноценного языкового инструментария можно рассматривать программный продукт JetBrains MPS [2]. Он предназначен для создания DSL, а для разработки программ на DSL в нем используется проекционный редактор, который представляет сценарий на DSL как синтаксическое дерево. Такой подход к редактированию позволяет разрешить проблемы неоднозначности редактирования синтаксических конструкций, которые могут возникнуть при расширении текстовой грамматики языка или при совмещении нескольких расширений [3]. Стоит отметить, что инструментарий JetBrains MPS ориентирован на программистов, а не на конечных пользователей.

Проекционный редактор позволяет привнести возможности интеллектуальных редакторов (автодополнение, генерация кода, средства рефакторинга, отладки, управления версиями и т. д.), делая процесс создания программ на DSL максимально легким, быстрым и удобным для программистов или экспертов в предметной области. Кроме того, так как каждое представление связано с одним из синтаксисов DSL, то среда редактирования сценариев на DSL должна реагировать на изменение DSL, разрабатываемого в языковом инструментарии, и перестраиваться в соответствии с изменениями конкретных синтаксисов языка.

Заключение

Перечислим проблемы использования современных языковых инструментальных средств и разработки в языково-ориентированном стиле без их применения:

1. Высокие затраты на разработку «вручную» (без языкового инструментария) DSL (в первую очередь внешних DSL), замкнутость и нерациональность их повторного использования в смежных предметных областях.
2. Сложность использования (освоения) существующих языковых инструментариев.
3. Сложность организации собственного языкового инструментария – интегрированной среды для создания и поддержки DSL не только на уровне синтаксиса, но и на уровне семантики.
4. Отсутствие в современных языковых инструментариях средств представления семантической модели языка и отображения семантики языка на синтаксис (за исключением Intentional Workbench (Knowledge Workbench) – разработки компании Intentional Software [4]).

Говоря о проблеме представления семантики предметно-ориентированных языков в языковом инструментарии, можно отдельно отметить, что альтернативные вычислительные модели, примеры которых приводились выше, зачастую сходны со способами представления знаний. Так, например, семантика сети зависимостей может быть представлена в виде семантической сети, сис-

тема правил вывода ЕСЛИ...ТО – в виде системы продукций. Поэтому можно предположить, что эффективные методы представления семантики предметно-ориентированных языков и их реализация в языковом инструментарии могут быть созданы на стыке прикладного программирования и программной инженерии, с одной стороны (в качестве практической реализации и приложения), и с другой стороны, инженерии знаний (как теоретической основы).

Литература

1. Fowler, M. *Domain Specific Languages* / M. Fowler. Addison-Wesley Professional, 2010. 640 p.
2. JetBrains MPS, *MetaProgramming System*. – <http://www.jetbrains.com/mps/>
3. Соломатов, К. Как система JetBrains MPS позволяет достичь более широкого использования DSL-ей (языков специфичных для предметной области) / К. Соломатов. – <http://habrahabr.ru/post/66094/>
4. Knowledge Workbench, *Intentional Software*. – <http://www.intentsoft.com>

Ботов Дмитрий Сергеевич, аспирант кафедры ЭВМ, Южно-Уральский государственный университет (г. Челябинск), dm@comp.susu.ac.ru

Bulletin of the South Ural State University
Series “Computer Technologies, Automatic Control, Radio Electronics”
2013, vol. 13, no. 2, pp. 128–130

FEATURES AND PROBLEMS OF USING MODERN LANGUAGE WORKBENCH IN LANGUAGE-ORIENTED PROGRAMMING

D.S. Botov, South Ural State University, Chelyabinsk, Russian Federation, dm@comp.susu.ac.ru

Generalized features of various modern software tools that let you create and maintain object-oriented programming languages are considered briefly in this article. The feature of domain-specific languages development environment and editing environment are considered absolutely and irrespectively among different features. Consideration of editing environment features focuses on the description of the projectional editing environment, which are used in modern language workbenches and features that allow you to bring intelligent editors modern IDEs.

In conclusion in this brief report describes problems of using modern language workbenches and software development in the language-oriented programming without applying them. The focus of the opinion is given to the problem of representing the semantics of object-oriented languages and display semantics to syntax. Making assumptions about the use of various knowledge representation methods to represent the semantic models of object-oriented languages.

Keywords: DSL, domain-specific language, language workbench, language-oriented programming, semantic model, projection editor.

References

1. Fowler M. *Domain Specific Languages*. Addison-Wesley Professional, 2010. 640 p.
2. JetBrains MPS, *MetaProgramming System*. Available at: <http://www.jetbrains.com/mps/> (accessed at 7 February 2013).
3. Solomatov K. *Kak sistema JetBrains MPS pozvolyaet dostich' bolee shirokogo ispol'zovanie DSL-ey (yazykov spetsifichnykh dlya predmetnoy oblasti)*. [How does the system JetBrains MPS achieves greater use DSL (domain-specific languages)]. Available at: <http://habrahabr.ru/post/66094/> (accessed at 7 February 2013).
4. Knowledge Workbench, *Intentional Software*. Available at: <http://www.intentsoft.com> (accessed at 7 February 2013)

Поступила в редакцию 10 марта 2013 г.