DOI 10.34185/1562-9945-4-135-2021-06
УДК 004.434

О.К. Очеретяний, І.В. Баклан

АНАЛІЗ СУЧАСНИХ ПРОГРАМНИХ РІШЕНЬ ДЛЯ СТВОРЕННЯ ПРОБЛЕМНО-ОРІЄНТОВАНИХ МОВ ПРОГРАМУВАННЯ

Анотація. Мови для вирішення задач із конкретних доменів потребують структур даних та функціоналу для їх перетворення що характерні саме для даного домену. Таким чином програмування за допомогою таких мов спирається на конструкти, що можуть бути зовсім не зрозумілі для комп'ютера, але фахівець у проблемній області одразу зрозуміє, що саме відбувається. Для пришвидшення і популяризації створення проблемно-орієнтованих мов програмування виникли автоматизовані системи підтримки проєктування мов. Дані засоби забезпечують в рамках кінцевого продукту : засоби для інтерпретації мови, текстові редактори орієнтовані на мову та додаткові інструменти мовної інженерії. Однак розвиток даних засобів є не системним та нерегульованим жодними стандартами, що призводить до зростання кількості часу на навчання для роботи з такими інструментами та кількості часу на вибір необхідного інструменту для конкретної задачі. В статті наводиться порівняльна характеристика деяких засобів для створення проблемно-орієнтованих мов програмування. Визначені основні критерії для порівняння та запропоновані критерії що не реалізовані в описаних засобах. Наприкінці статті наводяться висновки отримані в результаті порівняння та пропозиції до покращення існуючих інструментів .
Ключові слова: проблемно-орієнтовані мови, текстова нотація, декларативне програмування, системи підтримки проектування мов програмування,інтегровані засоби розробки, TEF, TCS, EMFText.

Постановка проблеми. Проблемно-орієнтоване програмування відкриває нові можливості для використання абстракцій вищого рівня, та генерації продуктивного коду для вирішення сучасних проблем. Звичайно ідея не є істотно новою та є дуже схожою до перетворень мов програмування високого рівня в ассемблери, з використанням компіляторів. Тим часом розробники програмного забезпечення потребують засобів для вираження проблем конкретної доменної області. За часту використання мов програмування загальної орієнтації не вистачає для виконання таких задач. Результатом вирішення такої проблеми і стає проєктування проблемно-орієнтованих мов програмування. В процесі

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

автоматизації проєктування і впровадження проблемно-орієнтованих мов програмування в розробку почали з'являтися засоби підтримки створення таких мов. Дані засоби забезпечують систематизований підхід до створення нової мови, ефективну підтримку мові після випуску її релізу та додаткові можливості для виправлень і доповнень. В даній статті ми порівнюємо існуючі засоби проєктування проблемно-орієнтованих мов програмування що базуються на платформі eclipse.

Аналіз останніх досліджень і публікацій. Системи підтримки проєктування мов програмування (СППМП) як засоби для створення проблемно-орієнтованих мов популяризував Мартін Фаулер у 2010 році [1]. Такі засоби забезпечують ефективне проєктування, повторне використання та підтримку мов та їх інтегрованих засобів розробки. СППМП надають можливості широкому колу інженерів до розробки нових мов і, як наслідок, створюють новий рівень мовної інженерії, де набори синтаксично та семантично інтегрованих мов можуть бути розроблені із порівняно невеликими зусиллями. Це може призвести до появи середовищ програмування з багатьма парадигмами та метамови, орієнтованих на створення мови [2, 3], які можуть вирішити важливі завдання програмної інженерії.

Із початком створення мов програмування розробники створювали інструменти, що полегшують розвиток мови та її продуктивніше використання. Найперші системи підтримки проєктування мов програмування з'явилися в наукових дослідженнях американських та японських вчених 1980-1989 роках. Досягнення даних досліджень отримали розвиток в двох напрямках: системи проєктування із графічною та текстовою нотацією . Графічні СППМП які розробляються донині, включають MetaEdit+, DOME, та GME. З іншого боку, СППМП що підтримують текстові нотації включають Centaur , the Synthesizer generator , the ASF+SDF Meta-Environment , GemMex/Montages , LRC . Оригінально ці системи базувались на інструментах для формальної специфікації мов програмування загального призначення. Незважаючи на це , багато з них були використані для проєктування проблемно-орієнтованих мов[4].

СППМП з текстовою нотацією розроблені в 2009-2011 роках TEF TCS EMF JastAdd , Rascal, Spoofax [5], можна розглядати як наступників цих систем, що використовують досягнення в технології мовної інженерії основних середовищ розробки. У той же час проекційні мовні робочі середовища, такі як Intentional, відроджують і вдосконалюють стару ідею редакторів структур мов програму-

вання, відкриваючи можливість поєднання парадигм на концептів програмування.

Протягом усього свого розвитку в промисловості використовувались СППМП та мови, що стосуються конкретних доменів. Приклади включають:

- Eurofighter Typhoon, із використанням засобів IPSYS's HOOD (пізніше ToolBuilder).

- Nokia's feature phones , із використаням MetaEdit+.

- WebDSL та Mobl для розробки веб додатків та мобільних додатків відповідно розроблені із допомогою Spoofax[6]

- R ISLA , проблемна орієнтована мова для розрахунку процентної ставки продуктів[7], із використанням A SF +S DF .

- Polar монітори серцевого ритму та спортивні годинники, розроблені із використанням MetaEdit+.

- Проблемно-орієнтована мова для потреб цифрової криміналістики, розроблена із використанням Rascal.

Наразі системи підтримки проєктування мов програмування мають значне зростання кількості та різноманітності, що зумовлено високим науковим та промисловим інтересом. Існуючі засоби настільки різні за дизайном, підтримуваними функціями та використаною термінологією, що користувачам та розробникам важко зрозуміти основні принципи та альтернативи дизайну. Для подолання проблем, що пов'язані із різноманітностю існуючих засобів, необхідним є їх систематичний огляд та порівняння.

Мета дослідження. Таким чином, виникає необхідність дослідження ефективності існуючих систем підтримки проєктування мов програмування з отриманням критеріїв якості процесу проєктування проблемно-оріентованих мов програмування. Також, необхідно з'ясувати, які системи підтримки проєктування найкраще забезпечують ефективну розробку проблемно-оріентованих мов.

Викладення основного матеріалу дослідження. Для проведення якісного порівняння існуючих систем підтримки проєктування мов програмування необхідно визначити критерії для порівняння. Ми пропонуємо порівнювати дані засоби за вимогами які класифікують продукт як сучасне програмне забезпечення. Основними нашими параметрами є : можливість створення проблемно-орієнтованих мов ,підтримка для інтегрованих засобів розробки, під-

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

тримка комбінування мов, підтримка внесення змін в створену мову, забезпечення базових можливостей проведення тестування мови, інтеграція з іншим засобами програмної інженерії. Окремим пунктом що є показником високого рівня розвитку систем підтримки проєктування є наявність метрик для оцінювання вихідного результату. Дана можливість є відсутньою для класичних інтегрованих засобів розробки так як їх основний фокус іде на розробку програмного забезпечення без специфікації доменної області. Така ситуація унеможливлює будь-які порівняння результуючих продуктів через те що для веб додатку відповідь серверу в 3 секунди може бути зависокою а для веб платформи для обробки надвеликих даних це найменший час відповіді серверу.Описані вище критерії будуть використані для оцінки наступних СППМП : tef tcs та emf .

TEF (Textual Editing Framework) [8] спочатку був розроблений Маркусом Шейдгеном під час докторської дисертації в Берлінському університеті. Для конкретного синтаксису (CS) TEF забезпечує мову визначення синтаксису, що називається TSL (мова текстового синтаксису). TSL описує текстову нотацію для існуючої мета-моделі Ecore у файлі .etslt. За допомогою звичайних засобів Eclipse EMF (загальна модель) генерується необхідна підтримка EMF. Для перетворення проблемно-орієнтованої мови в модель TEF створює синтаксичний аналізатор RunCC, який інтерпретується під час виконання (RunCC уникає генерації коду). TEF генерує три різні редактори через точки розширення eclipse:

Текстовий редактор. Даний компонент аналізує текстові моделі та дозволяє їх комфортно редагувати.

1. Редактор на основі моделі. Редактор на основі моделі діє як вдосконалений загальний редактор Ecore. Насамперед це редактор дерев, але можливі й інші нотації. Тут не проводиться синтаксичний розбір тексту, оскільки редагування “стилю” цього не дозволяє.

2. Вбудований редактор Це текстовий редактор, вбудований в редактор дерев на основі моделі. На кожному елементі моделі користувач може відкрити текстовий редактор за допомогою гарячої клавіші.

Отже, TEF поєднує різні стилі редагування (на основі дерева / тексту). Створений редактор підтримує виділення синтаксису, завершення коду, навігацію та посилання, згортання, анотацію помилок та кілька інших функцій. Перевірка моделі можлива за допомогою проектів Eclipse Modeling.

TCS (Textual Concrete Syntax) [9] був розроблений Фредеріком Жуо з EMN (Ecole des Mintes в Нанті) та командою ATLAN-Mod. Він також є частиною прое-

кту eclipse TMF і використовується в інших проектах eclipse, наприклад ATL2. Абстрактний синтаксис зазначений в текстовій мові для метамоделювання, що називається KM3 . Після збереження TCS на ходу генерує відповідний файл Ecore. Конкретний синтаксис вказаний у файлі .tcs. Знову ж таки, при збереженні відповідний парсер ANTLR в процесі роботи TCS, таким чином, дозволяє уникнути великої кількості непотрібної генерації коду, який часто зустрічається в інших інструментах. TCS уникає якомога більше генерації коду, а отже, дозволяє дуже швидко і коротко проходити цикли. Для впровадження оновлень в мову немає необхідності перезапускати середовище розробки, збереження файлів оновлює контекст виконання. Для редагування та створення проблемно-орієнтованої мови, що відповідає потребам користувача, TCS пропонує текстовий загальний редактор (TGE). TGE підтримує підсвічування синтаксису, гіперпосилання та підказки для кожної мови, текстовий синтаксис якої вказаний у TCS. За потреби TGE можна додатково налаштувати під конкретні потреби / компонування. Перевірка моделі можлива за допомогою проектів Eclipse Modeling або, бажано, безпосередньо за допомогою мови ALT. TCS підтримує роботу із понад 50 мовами на своєму веб-сайті. TCS також повторно використовується Furcas, іншим інструментом TMF.

EMFText [10] спочатку був розроблений як частина структури повторного програмного забезпечення в Університеті Дрездена. Згодом проект перейшов у розробку як незалежний інструмент. Подібно до TCS, EMFText дозволяє вказати конкретний синтаксис для існуючої моделі EMF (абстрактний синтаксис). Щоб дозволити EMFText використовувати моделі під час виконання, необхідно створити модуль EMF-моделі .

Конкретна специфікація синтаксису (файл .cs) складається з 3 блоків:

- Блок конфігурації, який містить ім'я, базову модель та кореневий клас Meta (символ запуску).За бажанням можна імпортувати інші синтаксиси та метамоделі та вказати параметри генерації.
- Розділ TOKEN, в якому можуть бути вказані лексеми для лексичного аналізатора.
- Розділ RULES, який визначає синтаксис кожного конкретного класу Meta.

EMFText має деяку спеціальну підтримку для визначення синтаксису:

1. Автоматичне генерування синтаксисів за замовчуванням
2. Модульна специфікація (підтримка абстрактних синтаксисів та імпорту синтаксису)
3. Механізми розпізнавання посилань за замовчуванням та всебічний

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

синтаксичний аналіз, щоб попередити про потенційні проблеми синтаксису

У процесі роботи EMFText за замовчуванням створюється редактор. Розробники можуть перезаписати або налаштувати особливу поведінку. З коробки редактор підтримує декілька функцій IDE, наприклад, контур, підсвічування синтаксису (також через файл .cs), доповнення коду, обробку дужок та звичайну підтримку гіперпосилання та посилань. EMFText також підтримує роботу з приблизно 50 мовами, з мовами реального світу, такими як Java5.

Ми оцінили кожну із технологій на відповідність характеристикам необхідних для систем підтримки проєктування проблемно-орієнтованих мов програмування. Порівняння наведене у таблиці 1. Можливість створення проблемно-орієнтованих мов. Деякі засоби СППМП мали на меті автоматизацію створення мов програмування загального використання. Ці мови не потребують детального опису домену , абстрактних типів даних домену та специфікованих методів перетворення даних. Відсутність можливостей розвитку такого функціоналу створює суттєві перешкоди для розробки проблемно-орієнтованих мов програмування.

Таблиця 1

Порівняння систем підтримки проєктування мов програмування

	TEF	**TCS**	**EMFText**
Можливість створення проблемно-орієнтованих мов	+	+	+
Підтримка інтегрованих засобів розробки	+	+	+
Підтримка комбінування мов	+	-	-
Підтримка внесення змін в створену мову	+	-	+
Інтеграція з іншим засобами програмної інженерії	+	+	-
Наявність метрик для оцінювання вихідного результату	-	-	-

Підтримка інтегрованих засобів розробки. Кінцевим продуктом СППМП є мова програмування. Необхідним додатком для використання такого продукту є текстові редактори або інтегрованих засобів розробки. Відсутність плагінів для роботи із синтаксисом та іншими задачами що допомагають в розробці ,

суттєво знижує популярність створенної мови та зменшує коло розробників які зможуть швидко розпочати роботу із мовою.

Підтримка комбінування мов. Парадигма мовно-орієнтованого програмування передбачає створення мов для виконання конкретних функцій. Кожна окрема мова має бути скомбінована з іншими в рамках одного програмного проекту що забезпечить ефективну роботу із різнорідними доменами. Така комбінація є одним із основних підходів лямбда числення що є основою функціонального програмування.

Підтримка внесення змін в створену мову. Кожен програмний продукт створюється із перспективою використання цього продукту різними розробниками для роботи в різних умовах. Врахування потреб різних груп користувачів на ранніх етапах розвитку продукту є неможливим. Для подолання цих ситуацій необхідною є постійна підтримка і внесення змін у початковий продукт. Таким чином продукт буде набирати більшу популярність через відповідність потребам своєї аудиторії.

Забезпечення базових можливостей проведення тестування мови. Попередньо описаний компонент додає системі як позитивних рис так і негативних. В процесі внесення змін частою є ситуація відмови роботи попередньо написаних частин мови через додавання нових можливостей. Для подолання цієї ситуації має бути передбачено в системі базове тестування наявного функціоналу. Без цього компоненту жоден реліз не зможе бути покращенням для попереднього.

Інтеграція з іншим засобами програмної інженерії. В процесі випуску релізів мови програмування може стати необхідною робота команди розробників над продуктом. Класичним засобом для розподіленної роботи є система контролю версій. Доступність такого компоненту дозволить маштабувати розробку і забезпечити доступ різних спеціалістів із мовної інженерії.

Наявність метрик для оцінювання вихідного результату. Метрики для оцінювання мов програмування не є поширеними в галузі розробки проблемно-орієнтованих мов. Розробка таких критеріїв додасть унікальну можливість для перевірки ефективності внесених змін та допоможе зрозуміти розробникам мов напрямки в яких вони зможуть покращувати свої продукти в рамках роботи із системою підтримки проєктування мов програмування.

Висновки. У досліджені наведено порівняння систем підтримки проєктування мов програмування, що орієнтовані на роботу із проблемно-

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

орієнтованими мовами. Перевагами даних систем є орієнтація на роботу із конкретними областями для опису нотацій мов програмування Системи дозволяють створювати необхідні засоби для роботи із спроєктованими мовами програмування та забезпечують базові покращення в процесі проєктування. Недоліками даних систем є відсутність схожості в роботі систем та продуктах які отримує розробник на виході. Також важливим недоліком є відсутність жодних метрик оцінювання якості створених мов. Для покращення даних систем пропонується створити метрики оцінки проблемно-орієнтованих мов програмування та засоби підтримки розробки мови програмування за шаблоном розробки сучасних програмних продуктів. Дані покращення суттєво пришвидшать внесення змін у створені за допомогою цих систем мови програмування та зменшить поріг входу за рахунок наближення цих систем до інтегрованих засобів розробки програмного забезпечення.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. M. Fowler, Domain Specific Languages (1st. ed.). Addison-Wesley Professional, 2010,552 p.

2. S. Dmitriev, Language oriented programming: The next programming paradigm. OnBoard, Online Magazine, 2005, 1-14.

3. M. P. Ward. Language-oriented programming. Software – Concepts and Tools,1995,15:147–161.

4. M. Mernik, J. Heering, A. M. Sloane. When and how to develop domain-specific languages. ACM Comput. Surv.,2005, 37(4):316–344.

5. L. C. L. Kats , E. Visser. The Spoofax language workbench: Rules for declarative specification of languages and IDEs. OOPSLA ACM.2010, 444–463.

6. Z. Hemel,E. Visser. Declaratively programming the mobile web with Mobl. OOPSLA, ACM,2011,695–712.

7. J. van den Bos ,T. van der Storm. Bringing domain-specific languages to digital forensics. ICSE SEIP, ACM,2011, 671–680.

8. Scheidgen, Markus . Textual Modelling Embedded into Graphical Modelling. Model Driven Architecture - Foundations and Applications,2008,153-168.

9. Jouault F, Bézivin J, Kurtev I.. TCS: A DSL for the specification of textual concrete syntaxes in model engineering. Proceedings of the 5th International Conference on Generative Programming and Component Engineering, 2006. 249-254.

10. Heidenreich F., Johannes J., Karol S., Seifert M., Wende C. Model-Based Language Engineering with EMFText. Generative and Transformational Techniques in Software Engineering. Springer. 2011.322-345.

REFERENCES

1. M. Fowler, Domain Specific Languages (1st. ed.). Addison-Wesley Professional, 2010,552 p.

2. S. Dmitriev, Language oriented programming: The next programming paradigm. OnBoard, Online Magazine, 2005, 1-14.

3. M. P. Ward. Language-oriented programming. Software – Concepts and Tools,1995, 15:147–161.

4. M. Mernik, J. Heering, A. M. Sloane. When and how to develop domain-specific languages. ACM Comput. Surv.,2005, 37(4):316–344.

5. L. C. L. Kats , E. Visser. The Spoofax language workbench: Rules for declarative specification of languages and IDEs. OOPSLA ACM.2010, 444–463.

6. Z. Hemel,E. Visser. Declaratively programming the mobile web with Mobl. OOPSLA, ACM,2011,695–712.

7. J. van den Bos ,T. van der Storm. Bringing domain-specific languages to digital forensics. ICSE SEIP, ACM,2011, 671–680.

8. Scheidgen, Markus . Textual Modelling Embedded into Graphical Modelling. Model Driven Architecture - Foundations and Applications,2008,153-168.

9. Jouault F, Bézivin J, Kurtev I.. TCS: A DSL for the specification of textual concrete syntaxes in model engineering. Proceedings of the 5th International Conference on Generative Programming and Component Engineering, 2006. 249-254.

10. Heidenreich F., Johannes J., Karol S., Seifert M., Wende C. Model-Based Language Engineering with EMFText. Generative and Transformational Techniques in Software Engineering. Springer. 2011.322-345.

Received 29.03.2021.
Accepted 30.03.2021.

Анализ современных программных решений для создания проблемно-ориентированных языков программирования

В статье приводится сравнительная характеристика некоторых средств для создания проблемно-ориентированных языков программирования. Определены основные критерии для сравнения и предложены критерии не реализованные в описанных средствах. В конце статьи приводятся выводы полученные в результате сравнения и предложения по улучшению существующих инструментов.

Analysis of modern software solutions for creating problem-oriented programming languages

Recent research and publications.Support systems for designing programming languages (SPPMP) as a means to create problem-oriented languages were popularized by Martin Fowler in 2010. Such tools provide efficient design, reuse, and support for languages and their integrated development tools. SPPMPs enable a wide range of engineers to develop new languages and, as a

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

result, create a new level of language engineering where sets of syntactically and semantically integrated languages can be developed with relatively little effort. This can lead to the emergence of programming environments with many paradigms and metalanguages focused on creating a language [2, 3], which can solve important problems of software engineering.

The aim of the study. Thus, there is a need to study the effectiveness of existing systems to support the design of programming languages to obtain quality criteria for the design process of problem-oriented programming languages. You also need to find out which design support systems best support the effective development of problem-oriented languages.

Main material of the study.To make a qualitative comparison of existing systems for supporting the design of programming languages, it is necessary to define criteria for comparison. We propose to compare these tools according to the requirements that classify the product as modern software. Our main parameters are: the ability to create problem-oriented languages, support for integrated development tools, support for language combination, support for making changes to the created language, providing basic language testing capabilities, integration with other software engineering tools. A separate point that is an indicator of the high level of development of design support systems is the availability of metrics for evaluating the initial result. This feature is absent for classical integrated development tools as their main focus is on software development without domain domain specification. This situation makes any comparison of the resulting products impossible because for a web application the server response in 3 seconds may be too high and for a web platform for processing large data it is the shortest server response time. The criteria described above will be used to evaluate the following SPPMP: tef tcs and emf.

Conclusions. The study compares the support systems for designing programming languages that are focused on working with problem-oriented languages. The advantages of these systems are the focus on working with specific areas to describe the notations of programming languages. The systems allow you to create the necessary tools to work with the designed programming languages and provide basic improvements in the design process. The disadvantages of these systems are the lack of similarity in the operation of systems and products that the developer receives at the exit. Another important drawback is the lack of any metrics for assessing the quality of created languages.

Баклан Ігор Всеволодович – к.т.н, доцент, доцент кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського".
Очеретяний Олександр Костянтинович – аспірант, асистент кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського".

Баклан Игорь Всеволодович – к.т.н, доцент, доцент кафедры автоматизированных систем обработки информации и управления Национального технического университета Украины “Киевский политехнический институт им. Игоря Сикорского”.

Очеретяный Александр Константинович – аспирант, ассистент кафедры автоматизированных систем обработки информации и управления Национального технического университета Украины “Киевский политехнический институт им. Игоря Сикорского”.

Baklan Ihor Vsevolodovych – Ph.D., Associate Professor, Associate Professor of Department of Computer-aided management and data processing systems of the National Technical University of Ukraine " Igor Sikorsky Kyiv Polytechnic Institute”.

Ocheretianyi Oleksandr Kostyantynovych – p.h.d. student, lecturer assistant of Department of Computer-aided management and data processing systems of the National Technical University of Ukraine " Igor Sikorsky Kyiv Polytechnic Institute”.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)