

Автоматизация миграции программного кода на новый набор библиотек

Артем Алексюк

12 октября 2016 г.

Санкт-Петербургский политехнический университет Петра Великого

Table of contents

1. Elements

2. Conclusion

Миграция в новое библиотечное окружение?

- Новая программная платформа
- Новая аппаратная платформа
- Унаследованный код, несовместимый с современными системами, средствами разработки, библиотеками

Сложности

- Можно перенести вручную
- Много рутинной работы
- Попробовать автоматизировать?

Общий план решения задачи

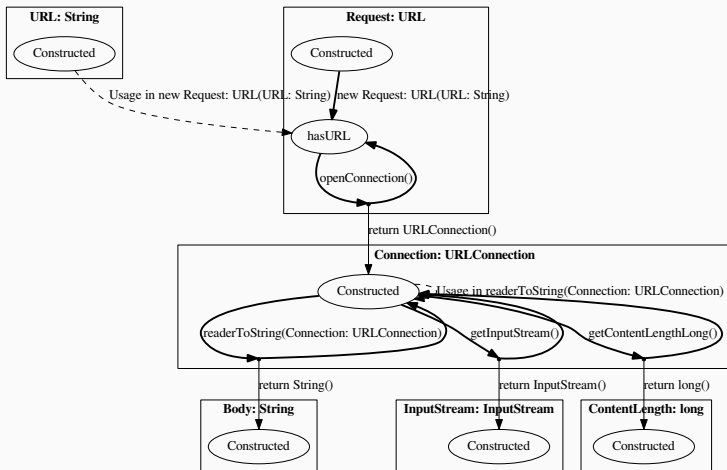
- Создать модели используемых библиотек
- Модель - частная спецификация, описывающая внешнее поведение библиотеки
- Найти соответствие между двумя моделями
- Преобразовать изменения обратно в код

Почему модели, а не сам код?

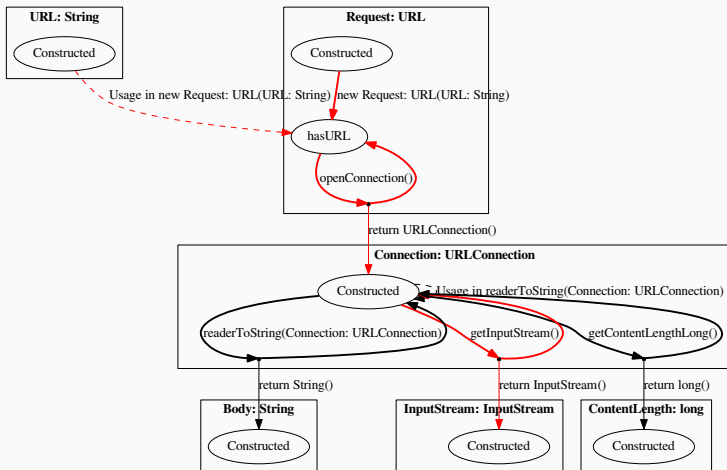
Не тьюринг-полные модели проще анализировать

- Для описания моделей используются расширенные конечные автоматы

Пример автомата



Пример автомата



Отображение кода на модель

- Динамическое отображение: программа инструментируется, на основе трассы выполнения составляется путь в модели
- Для инструментирования используется AOP (аспекты) в реализации AspectJ
- Бонус: проверка корректности использования исходной библиотеки

Поиск соответствия между моделями

- При переносе необходимо сохранить:
 - Набор действий, совершенных над библиотекой
 - Зависимости по данным
- Первые версии инструмента: модель преобразуется в граф, с помощью алгоритма Дейкстры ищем кратчайший путь между вершинами
- Сейчас: используется волновой алгоритм

- DSL на базе Kotlin для описания моделей
- Инструмент для миграции программ на языке Java
- Система проверки корректности преобразования

Пример миграции

До:

```
URL url = new URL("http://api.ipify.org/");
URLConnection conn = url.openConnection();
if (conn.getContentLengthLong() > 0) {
    String response = new BufferedReader(new
        InputStreamReader(conn.getInputStream())).
        lines().collect(Collectors.joining("\n"));
    System.out.println(response);
} else {
    System.out.println("Error!");
}
```

Пример миграции

После:

```
HttpGet url = new HttpGet("http://api.ipify.org/");
CloseableHttpClient newMachine_Client_0 =
    HttpClients.createDefault();
CloseableHttpResponse conn = newMachine_Client_0.
    execute(url);
long linkedEdge_ContentLength_1 = conn.getEntity().
    getContentLength();
if (linkedEdge_ContentLength_1 > 0) {
    InputStream linkedEdge_InputStream_2 = conn.
        getEntity().getContent();
    String response = new BufferedReader(new
        InputStreamReader(linkedEdge_InputStream_2)
        ).lines().collect(Collectors.joining("\n"))
        ;
    System.out.println(response);
}
```

This frame uses the `smallcaps` titleformat.

Potential Problems

Be aware, that not every font supports small caps. If for example you typeset your presentation with pdfTeX and the Computer Modern Sans Serif font, every text in smallcaps will be typeset with the Computer Modern Serif font instead.

ALL SMALL CAPS

This frame uses the `allsmallcaps` titleformat.

Potential problems

As this titleformat also uses smallcaps you face the same problems as with the `smallcaps` titleformat.

Additionally this format can cause some other problems. Please refer to the documentation if you consider using it.

As a rule of thumb: Just use it for plaintext-only titles.

This frame uses the `allcaps` titleformat.

Potential Problems

This titleformat is not as problematic as the `allsmallcaps` format, but basically suffers from the same deficiencies. So please have a look at the documentation if you want to use it.

Elements

Typography

The theme provides sensible defaults to `\emph{emphasize}` text, `\alert{accent}` parts or show `\textbf{bold}` results.

becomes

The theme provides sensible defaults to *emphasize* text, **accent** parts or show **bold** results.

Font feature test

- Regular
- *Italic*
- SMALLCAPS
- **Bold**
- ***Bold Italic***
- **BOLD SMALLCAPS**
- Monospace
- Monospace *Italic*
- **Monospace Bold**
- **Monospace Bold Italic**

Lists

Items

- Milk
- Eggs
- Potatos

Enumerations

1. First,
2. Second and
3. Last.

Descriptions

PowerPoint Meeh.

Beamer

Yeeeha.

- This is important

Animation

- This is important
- Now this

Animation

- This is important
- Now this
- And now this

Animation

- This is really important
- Now this
- And now this

Таблица 1: Largest cities in the world (source: Wikipedia)

City	Population
Mexico City	20,116,842
Shanghai	19,210,000
Peking	15,796,450
Istanbul	14,160,467

Blocks

Three different block environments are pre-defined and may be styled with an optional background color.

Default

Block content.

Alert

Block content.

Example

Block content.

Default

Block content.

Alert

Block content.

Example

Block content.

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$$

Line plots

Bar charts

Veni, Vidi, Vici

References

Some references to showcase [allowframebreaks]
[4, 2, 5, 1, 3]

Conclusion

Summary

Get the source of this theme and the demo presentation from

`github.com/matze/mtheme`

The theme *itself* is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Questions?

Backup slides

Sometimes, it is useful to add slides at the end of your presentation to refer to during audience questions.

The best way to do this is to include the `appendixnumberbeamer` package in your preamble and call `\appendix` before your backup slides.

METROPOLIS will automatically turn off slide numbering and progress bars for slides in the appendix.

References I



P. Erdős.

A selection of problems and results in combinatorics.

In *Recent trends in combinatorics* (Matrahaza, 1995), pages 1–6. Cambridge Univ. Press, Cambridge, 1995.



R. Graham, D. Knuth, and O. Patashnik.

Concrete mathematics.

Addison-Wesley, Reading, MA, 1989.

References II



G. D. Greenwade.

The Comprehensive Tex Archive Network (CTAN).

TUGBoat, 14(3):342–351, 1993.



D. Knuth.

Two notes on notation.

Amer. Math. Monthly, 99:403–422, 1992.



H. Simpson.

Proof of the Riemann Hypothesis.

preprint (2003), available at

<http://www.math.drofnats.edu/riemann.ps>,
2003.