

An Aspect-Oriented Framework for Development of Dynamic Content

Kohei Nagashima

Graduate School of Science and Technology, Keio University
Yokohama, Japan
k-nagashima@yy.ics.keio.ac.jp

Abstract

Modern web contents, including applications, enhance user experiences by introducing Web 2.0 technologies such as Ajax. Ajax applications are implemented with three different languages, including HTML, CSS, and JavaScript. The artifacts of each these languages are basically described in a separate file. In such a context, many Web pages have dynamic contents such as a button that changes background color or layout of text. For this behavior, developers usually describe pieces of code with such three languages; therefore, they have to modify three different files at the same time when the change of behavior is required. In this way, pieces of code that provide a single behavior are scattered in different files.

This paper proposes a framework, which adopts the concept of aspect-orientation to solve the above mentioned problems. This framework enables developers to describe pieces of code to provide a single behavior as an aspect in a file, provides a weaver to weave them before running.

Categories and Subject Descriptors D.3.3 [PROGRAMMING LANGUAGES]: Language Constructs and Features—Modules, packages, Frameworks

General Terms Design, Languages

Keywords aspect-oriented programming, web development, web contents, modularization

1. Problem and motivation

In the early stages of web development, users who provided web content described in HTML with little care for visual effects. In the 1990s, CSS and JavaScript appeared, and the former provides visual effect with web contents and the latter

provides the dynamic behavior with them. Then, users began to utilize these languages to provide visual effects and dynamic behaviors and pieces of code for the effects/behaviors are usually described in a single HTML file directly. With the increase of web applications, the descriptions of each language were being complicated and they began to be managed in different files as module. This shift is reasonable because each language provides different functionalities.

However, with the Web 2.0 paradigm, the usage of these languages is changing because developers need to describe pieces of code with the three different languages at the same time. For example, when developers implement an animation with Ajax, they need to specify method invocation in a HTML element, describe the visual aspect of the element with CSS, and then prepare a method which defines how to move the target and/or visual changes of it with JavaScript. In this way, pieces of code to provide a single behavior are scattered in separate files.

2. Background and related work

One of the way to modularize the scattered codes is aspect-oriented programming[3](AOP). In AOP, the codes which are scattered in many files are gathered into the new module called an aspect, and an aspect is woven into the target modules. When we create a dynamic content, we often use JavaScript. A kind of the aspect-oriented JavaScript programming framework is the AspectScript[4]. The AspectScript adopts the join point model which is like the AspectJ[2] and prepares the feature of JavaScript as higher order function. Though we can make various aspects by utilizing the AspectScript, it is not an optimal solution for modularizing the dynamic content.

3. Approach and uniqueness

We propose a new aspect-oriented programming framework which is based in JavaScript to modularize efficiently the dynamic contents such as an animation. JavaScript is a powerful language because it can change the DOM elements described in the HTML and appearance described in the CSS. However, when the developers create the dynamic contents,

join point	points in the program execution at which...
attr	an attribute of HTML element is matched
call	a function or object is called
exec	a function or object is executed
set	an object is set
get	an object is read

Table 1. Join point model

they cannot use only JavaScript. We believe that the code of dynamic content which was created only JavaScript is difficult to understand and become redundant.

3.1 Join point model

To modularize the dynamic contents as an aspect, this framework supports five join points as shown in Table 1. A join point “attr” is for HTML and the others are for JavaScript. Currently, we don’t provide the join point for CSS. Since the change of the CSS code is rather easy and independently achieved.

3.2 Describing the aspect

We show an example for explaining this framework. Figure 1 shows the example of weaving when a HTML element has been specified as a join point. In the aspect for the dynamic content, we can describe the HTML, CSS and JavaScript codes. We provide four advices such as before, after, around and handler. The former three are provided in the AspectJ. The handler advice is for weaving the aspect into HTML files, another three advices is for weaving the aspect into JavaScript files. In Figure 1, the HTML code which is in the handler advice will be inserted into the code portion in a dashed frame. The CSS codes in the aspect and JavaScript code in the handler advice will be added in the bottom of head element.

4. Results and contribution

In this section, we evaluate this framework and describe the conclusions of our research.

4.1 Evaluation for our framework

We evaluate the efficiency of the development technique using our framework. We use a Web page[1] which is provided by Apple Inc. This Web page has a dynamic content which is the animation which rotates the groups of pictures. Then we suppose there comes a change request which requires a change of motion of a group of pictures from rotation to left-to-right slide. The number of changed files for this change request is shown in Table 2, the second column shows in the case of using our framework and the third column shows the case of not using the framework. As shown in Table 2, the number of changed files is obviously decreased by using the framework.

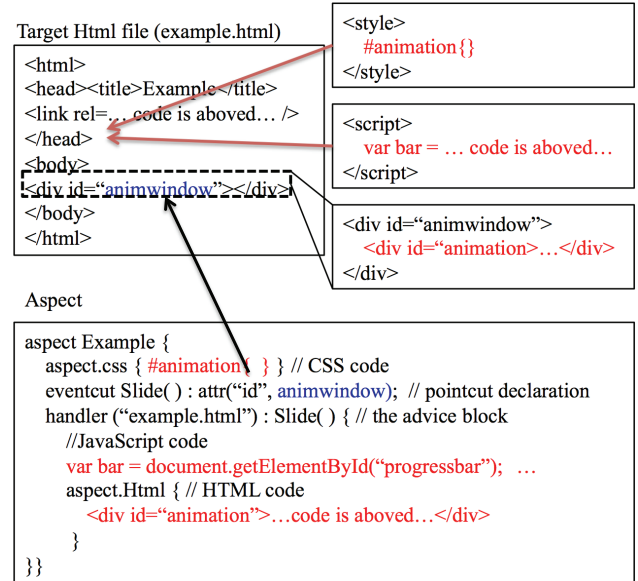


Figure 1. Weaving mechanism for handler advice

	using this framework	not using this framework
All files	1	5
HTML files	0	1
CSS files	0	2
JavaScript files	0	2
Aspect files	1	0

Table 2. The number of files to deal with the change request

4.2 Conclusions

We proposed a framework for modularize the scattered code for dynamic contents of a Web page that spread over several files. By using our framework, developers are able to program dynamic contents as a combination of a join point and an aspect. It can reduce the number of changed files if once a change request is issued on the dynamic contents. As a consequent, we can easily respond to a change request of the dynamic contents by using the framework.

References

- [1] Apple Inc. Apple. <http://www.apple.com/jp/iphone/>.
- [2] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. G. Griswold. An overview of aspectj. In *Proceedings of the 15th European Conference on Object-Oriented Programming, ECOOP '01*, pages 327–353. Springer-Verlag, 2001.
- [3] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J. marc Loingtier, and J. Irwin. Aspect-oriented programming. In *ECOOP*. Springer-Verlag, 1997.
- [4] R. Toledo, P. Leger, and É. Tanter. Aspectscript: expressive aspects for the web. In *Proceedings of the 9th International Conference on Aspect-Oriented Software Development, AOSD '10*, pages 13–24. ACM, 2010.