AOWP:
Web-specific AOP framework for PHP

Keiji Hokamura, Ryoto Naruse, Masaru Shiozuka, Naoyasu Ubayashi
(Kyushu Institute of Technology, Japan)

Shin Nakajima (National Institute of Informatics, Japan)

Akihito Iwai (DENSO Corporation, Japan)
✓ Aspect-Oriented Programming (AOP) is a technique for modularizing cross-cutting concerns (CCCs)

✓ AOWP is AOP framework for PHP, including Web-specific AOP mechanisms

✓ Using AOWP, programmers can easily and intuitively modularize CCCs in Web applications

✓ We present some example aspects using real-world Web application (PukiWiki) in this tool demonstration
Motivation

Problem of general-purpose AO languages (GPALs)

Motivation example: access control

It is not easy to use GPALs in Web application development because GPALs do not support Web-specific concepts including:
- page accesses (HTTP request transaction)
- user session
In addition to AOP mechanism of GPALs, AOWP provides
✓ request join point and Web-specific pointcut designators
✓ Web-specific aspect instantiation mechanism
AOWP’s weaver is **command line tool** implemented in PHP. *(PHP_Parser in the PEAR library is used for the parser.)*
Example: AuthAspect
Example: AuthAspect

Access control concern

Add user-authentication-based access control to the whole of a PukiWiki which is a famous open-source Wiki system.
Example: AuthAspect

Aspect description

```php
class AuthAspect extends PerSessionAspect {
    private $_authed = false;

    public function __construct() {
        $accChkPC = new RequestPointcut('.*');
        $accChkAdv = new BeforeAdvice();
        $accChkAdv->setPointcut($accChkPC);
        $accChkAdv->setAdviceBody('accChk');
        $this->addAdvice($accChkAdv);
    }

    public function accChk(JoinPoint $jp) {
        if (!$this->_authed) {
            jumpToLoginPage();
        }
    }

    // Omit the advice for user authentication and logout transaction because of space limitation
}
```

**Pointcut**: select all page requests for checking user authority

**Advice**: Display the login page if the user have not been authenticated

Keep the user authentication result
Example: AuthAspect

Weaving

```php
<?php
class AuthAspect {
    //
    //
    //
    //
    ?>
```
Example: AuthAspect

Demo
More examples

• **ConditionsOfUseAspect**
  - Display the page of *conditions of use* when a user updates contents without access to the page
  - Implement a concern based on *page flow of each user*

• **QuestionnaireAspect**
  - Show the questionnaire after 1 minute after the first user access
  - Implement a concern based on *duration time of user access*

• **MonitorAspect**
  - Stop the service for managing the contents when accessing user is too many
  - Implement a concern based on *state of a Web application*
Thank you for your attentions

AOWP is developed as open-source project, and the framework is available on

http://sourceforge.jp/projects/aowp/
Implementation of AOWP framework
Implementation of request JP

When weaving, request join points are related to suitable program execution points

<table>
<thead>
<tr>
<th>Type of advice</th>
<th>Related program execution point in PHP</th>
</tr>
</thead>
<tbody>
<tr>
<td>before</td>
<td>• before the first execution of a script file in transaction of a page request</td>
</tr>
</tbody>
</table>
| after          | • after the first execution of a script file in transaction of a page request  
                 • before a call of exit function |
| around         | • the whole of a transaction of a page request |
Implementation of aspect instantiation

Instances are managed by AspectInstanceManager (using cookie and object serialization)
More examples

• **QuestionnaireAspect**
  - Show the questionnaire after 1 minute after the first user access
  - Implement a concern based on *duration time of user access*

• **MonitorAspect**
  - Stop the service for managing the contents when accessing user is too many
  - Implement a concern based on *state of a Web application*