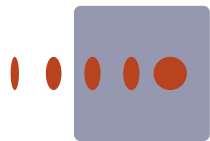
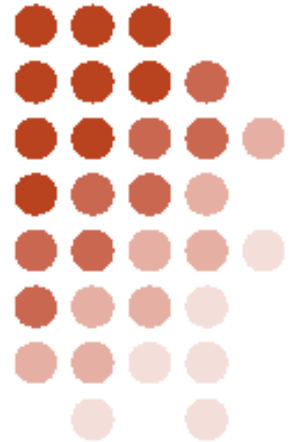




Web-specific crosscutting concerns

William Candillon

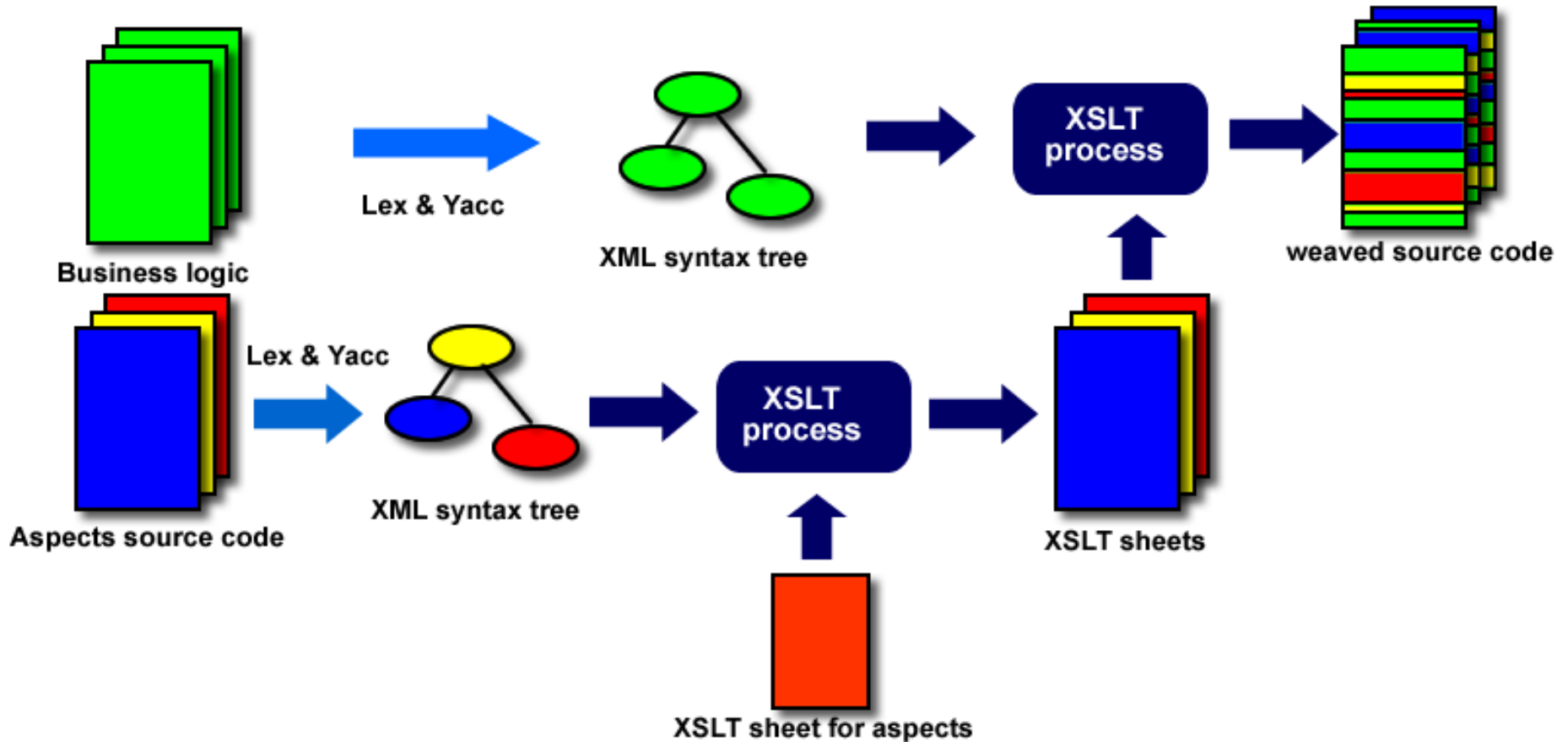
{wcandillon@elv.telecom-lille1.eu}



Aspect-Oriented Programming

- An active field of research and development
<http://scholar.google.com/scholar?q=aop>
- A new programming paradigm...
 - Working with OOP
 - To separate crosscutting concerns from the business logic
- ...defining mechanisms for
 - Writing aspects as a new software entity
 - Weaving technical concerns on business logic

Weaving chain of phpAspect



Toward web-specific AOP

- phpAspect got aspectJ style (eclipse.org/aspectj)
- Next step: integration of web-specific joinpoints
 - XML enclosing context identification
 - Web page joinpoint
 - Session instantiation of aspects
 - Interception of PHP global variables (`$_GET`, `$_POST`) to prevent XSS failures:

```
pointcut XssProtect: get($_POST[*]) || set($_POST[*]);
```

A virtual cart

```
<?php
class Order{

    private $items = array();
    private $amount = 0;

    public function addItem($reference, $quantity){
        $this->items[] = array($reference, $quantity);
        $this->amount += $quantity*Catalog::getPrice($reference);
    }

    public function getAmount(){ return $this->amount; }
}

class Catalog{
    private static $priceList = array('Largo Winch' => 9.31,
        'Astérix' => 8.46, 'XIII' => 8.70);

    public static function getPrice($reference){
        return self::$priceList[$reference];
    }
}

$myOrder = new Order;
$myOrder->addItem('Largo Winch', 2);
$myOrder->addItem('Astérix', 2);
$myOrder->addItem('Largo Winch', -6);
?>
```

- A client add products in the cart.
- Business logic without any technicals concerns.

A logging aspect

```
<?php
aspect TraceOrder{
    pointcut logAddItem:exec(public
Order::addItem(2));
    pointcut logTotalAmount:call(Order->addItem(2));

    after logAddItem{
        printf("%d %s added to the cart\n", $quantity,
$reference);
    }

    after logTotalAmount{
        printf("Total amount of the cart : %.2f €\n",
$thisJoinPoint->getObject()->getAmount());
    }
}
?>
```

- Log every added articles and give status of the order

- After weaving

```
$myOrder->catalog = new Catalog;
$myOrder->addItem('Largo Winch', 2);
printf("Montant total de la commande : %.2f €\n",
    $myOrder->getAmount());
$myOrder->addItem('Astérix', 1);
printf("Montant total de la commande : %.2f €\n",
    $myOrder->getAmount());
```

- Result

```
2 Largo Winch added to the cart
Total amount of the cart : 18.62 €
1 Astérix added to the cart
Total amount of the cart : 27.08 €
-6 Largo Winch added to the cart
Total amount of the cart : -28.78 €
```

A security aspect

```
<?php
aspect Security{
    pointcut logAddItem:exec(public Order::addItem(2));

    before logAddItem{
        if(!Catalog::getPrice($reference) ||
            (float)$quantity < 0){
            echo "Wrong parameters";
            return false;
        }
    }
}
?>
```

- Make a filter on the customer input
- Protection against cross scripting injection
- Result

```
2 Largo Winch added to the cart
Total amount of the cart : 18.62 €
1 Astérix added to the cart
Total amount of the cart : 27.08 €
Wrong parameters
Total amount of the cart : 27.08 €
```

Acknowledgment

- All the Google SoC crew (code.google.com)
- The PHP community (php.net)
- Gilles Vanwormhoudt
(vanwormhoudt@telecom-lille1.eu)
 - Doctor in computer science at Telecom Lille 1
 - Collaborator on the phpAspect project

Thanks for your attention