TYPO3.Flow - Feature #10427

Support inter-type properties for entites

2010-10-25 12:44 - Martin Eisengardt

Status:	Resolved	**Start date:**	
Priority:	Should have	**Due date:**	
Assignee:		**% Done:**	0%
Category:	AOP	**Estimated time:**	0.00 hour
Target version:			
PHP Version:		**Complexity:**	
Has patch:	No		

Description

AspectJ defines inter-type properties. Flow3 is not using the syntax right now.

Reason why we need this feature:
Let us say you want to develop a module that lets you vote on several entites. You do not want to let the module know which types of entities you are voting on. Instead use aspects to extend the entites. First of all each entity will implement a "Votable" interface. That is already possible. The second task is to introduce variables in form of a simple counter "good/bad". Now you got a problem because currently the persistence framework asks the class service for the class scheme. They will never see those new variables and will never make them persistent.

There will be other use cases where you need to introduce persistent variables into entites. However non-persistent variables are already supported through AOP_Proxy::getProperty/setProperty.

AspectJ
Taken from http://www.eclipse.org/aspectj/doc/released/progguide/starting-aspectj.html#inter-type-declarations

```
aspect PointObserving {
    private Vector Point.observers = new Vector();
    ...
}
```

That example would be translated in PHP/FLOW3 to:

```
/**
 * @aspect
 */
class MyAspect {

    private Point::$observers;

}
```

Thats not clever because this syntax causes compile errors in php.

Possible syntax

```
/**
 * @aspect
 */
class MyAspect {

    /**
     * @intertype class(Point), private
     * @var array
     */
    protected $MyAspect_Observers;

}
```

The "intertype" tag followed by the full qualified target class name (or a pattern) and optional an access level (private/protected/public) will cause the aop framework to add this variable declaration tro the target class. If an interface is specified the variable will be added to each class implementing the interface. It should be possible to use introduces as well. The following aspect whould add the variable $observers to class FooClass:

```
/**
 * @aspect
 */
class MyAspect {

    /**
     * @introduces FooInterface, class(FooClass)
     */
    protected $fooInterface;

    /**
     * @intertype class(FooInterface), private
     * @var array
     */
    protected $$MyAspect_Observers;

}
```

Possible problems:

The above syntax is compatible to php but it may be misleading. Implementors or aspects may expect that "$this->MyAspect_Observers" is working. That will not be true. As before you need to access the intertype variable through:

```
$joinPoint->getProxy()->FLOW3_AOP_Proxy_getProperty
$joinPoint->getProxy()->FLOW3_AOP_Proxy_setProperty
```

History

#1 - 2010-11-12 16:10 - Karsten Dambekalns

- Subject changed from AOP: inter-type properties for entites to Support inter-type properties for entites

- Start date deleted (2010-10-25)

#2 - 2012-03-12 18:34 - Christian Müller

- Status changed from New to Resolved

- Has patch set to No

Take a look at Packages/Framework/TYPO3.FLOW3/Classes/Persistence/Aspect/PersistenceMagicAspect.php there we introduce the FLOW3_persistence_identifier, you could do the same for some property you need, if not directly at least by introducing a new annotation for this "inter-type" property.