

Proposal to Classify Aspects

Dimensions to Classify Aspects

- **Level of Application: Design vs. Implementation?**
- **Granularity: Subsystem, Component, Class/Object, Method, Statements**
- **Weaving Time: Code-Transformation, Compile-Time, Load-Time, Runtime**
- **Dependence on Programming-Technique: OO vs. declarative**

Semantical Facets and Related Work

- Semantical Properties
 - functional vs. non-functional aspects
 - semantic preserving vs.
semantic changing aspects
- **Relation to Refactoring: 2 steps**
 - semantic-preserving code transformation
 - change of semantic
- **Aspects do this 'at once'**

Commonalities and Differences of Approaches

- **Message Interception**
(Message in OO/Java sense)
 - HyperJ: by overriding methods
 - MOP? AOP = MOP - Meta?
- **Base System not aware of Aspect**

Even more open issues . . .

- AOP favours Expressiveness over Security
 - AOP to reinvent lost Expressiveness in JAVA?
 - ,Type Check‘ for Aspects?
- Classification of Aspect =
Classification of Crosscutting?