

First-class aspects and separate compilation

First-class aspects:

- explicit representation at run time

Consequences

- aspects as first-class objects (with state)
(as opposed to transformation)
- traceable form of abstraction (own identity)

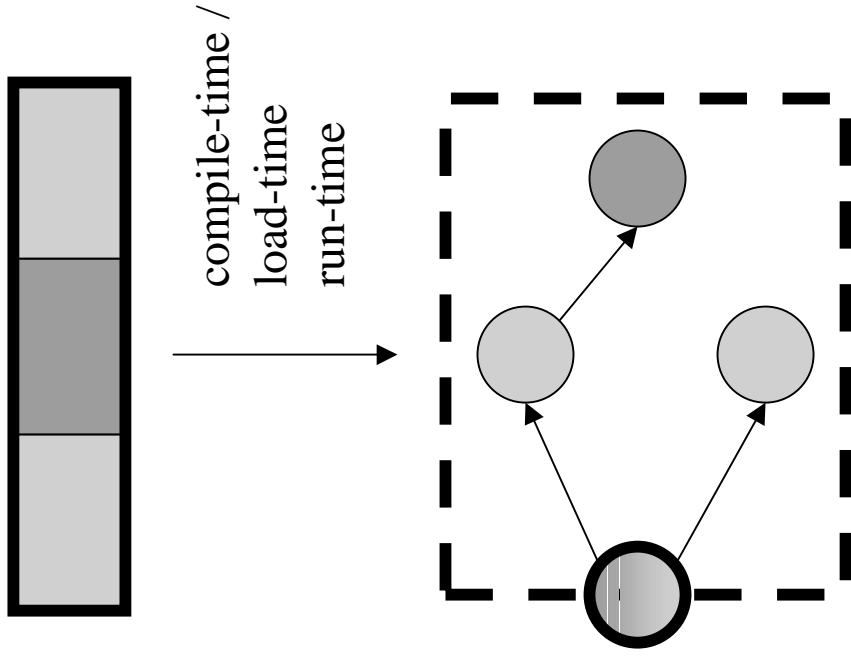
First-class aspects

Contradiction?

- objects comprise a set of aspects
- aspects are objects

No!

- conceptual objects comprise a set of aspects
- aspects are represented as objects at runtime



Separate compilation

- source units are compiled module per module
- only the interface but not the implementation code of other modules is used in compilation of a module
- compilation of aspect-code must not take a global snapshot

Advantages

- no recompilation required after changes of internal implementation
 - of base modules
 - of aspect definitions(?)