

Specialization Patterns

ASE'2000

Ulrik P. Schultz, Julia L. Lawall, Charles Consel
Compose Group, IRISA / INRIA / LaBRI

Friday September 15th

Program **Specialization**

+

Design **Patterns**

Specialization **Patterns**

Design Patterns

- A design pattern systematically names, explains, and evaluates an important and recurring design in object-oriented systems
- A design pattern encapsulates a useful design idea, and allows it to be communicated in the object-oriented community

Programming with Design Patterns

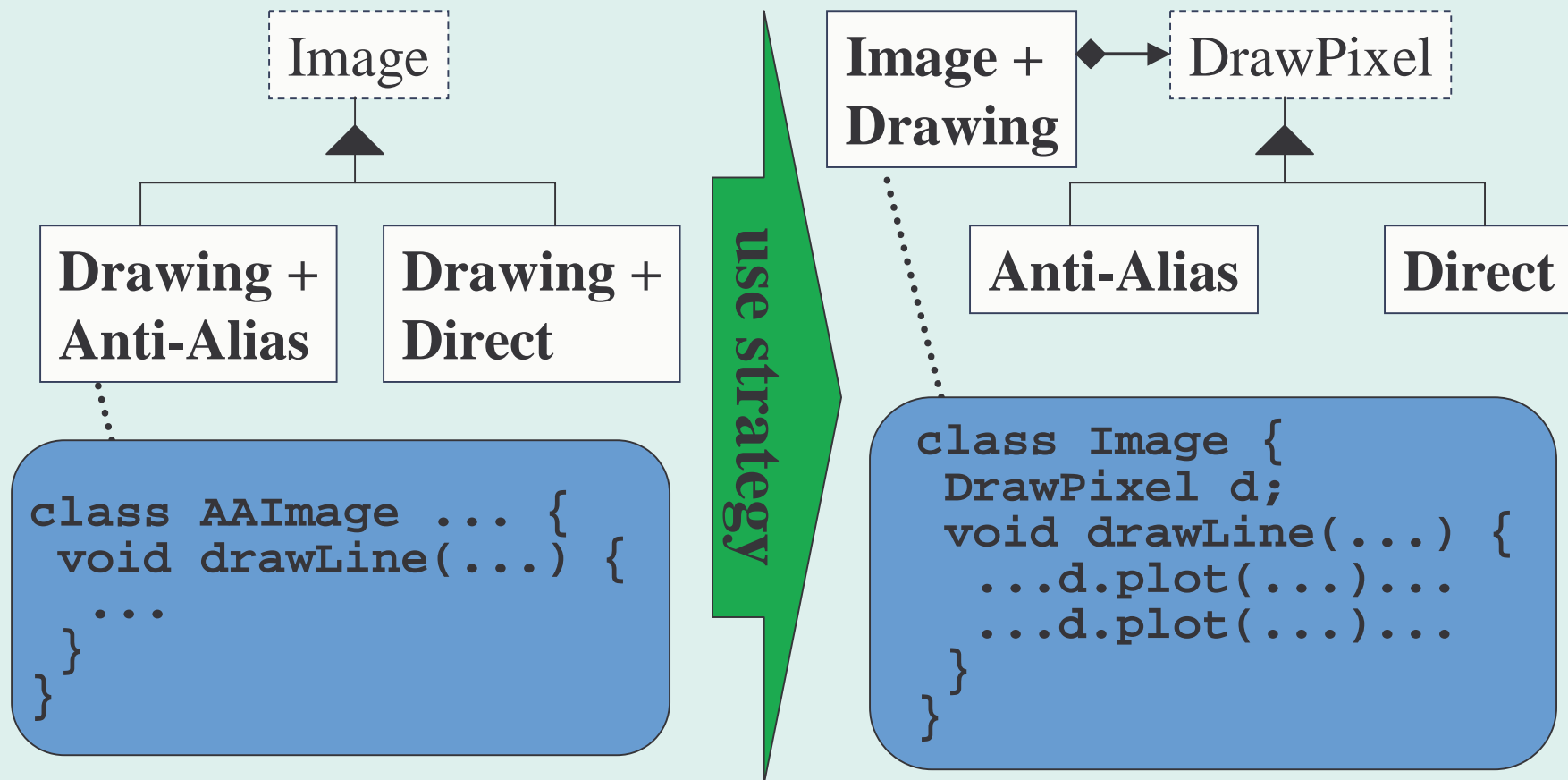
Many design patterns are characterized by:

- Focus on modularity and code reuse
- Extensive parameterization by objects known only by a generic interface
- Highly adaptable designs

Problem: Efficiency

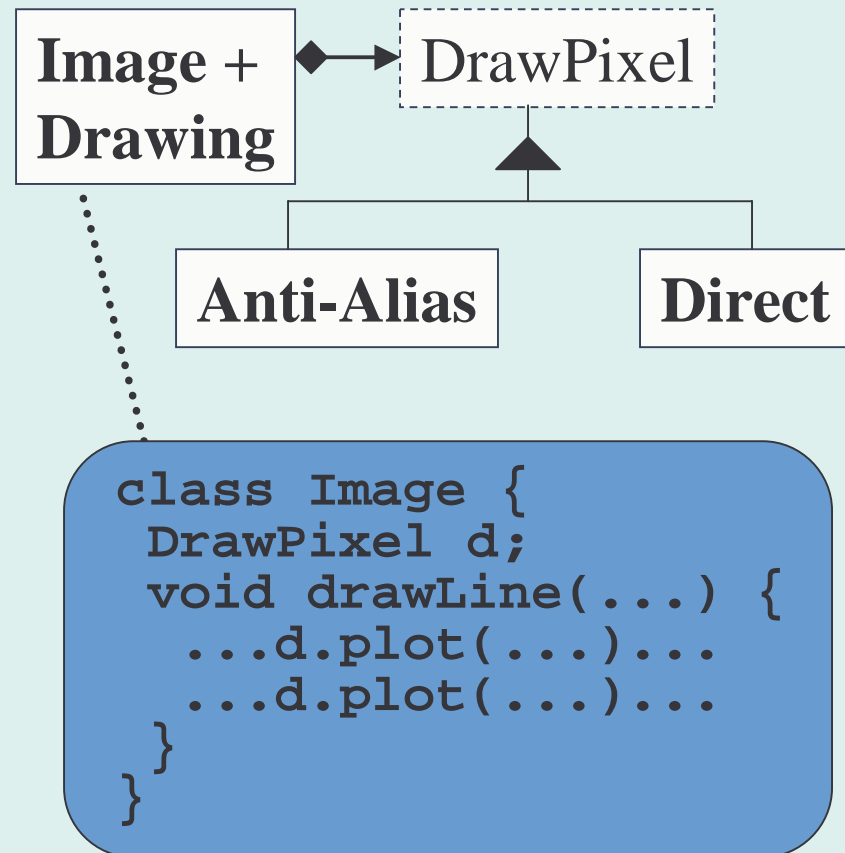
- Adaptability at the design level is also present at the implementation level!
- Parameterization of program parts by objects enables run-time adaptation
- But state-of-the-art compiler technology cannot eliminate unneeded adaptability

Example: Strategy Design Pattern



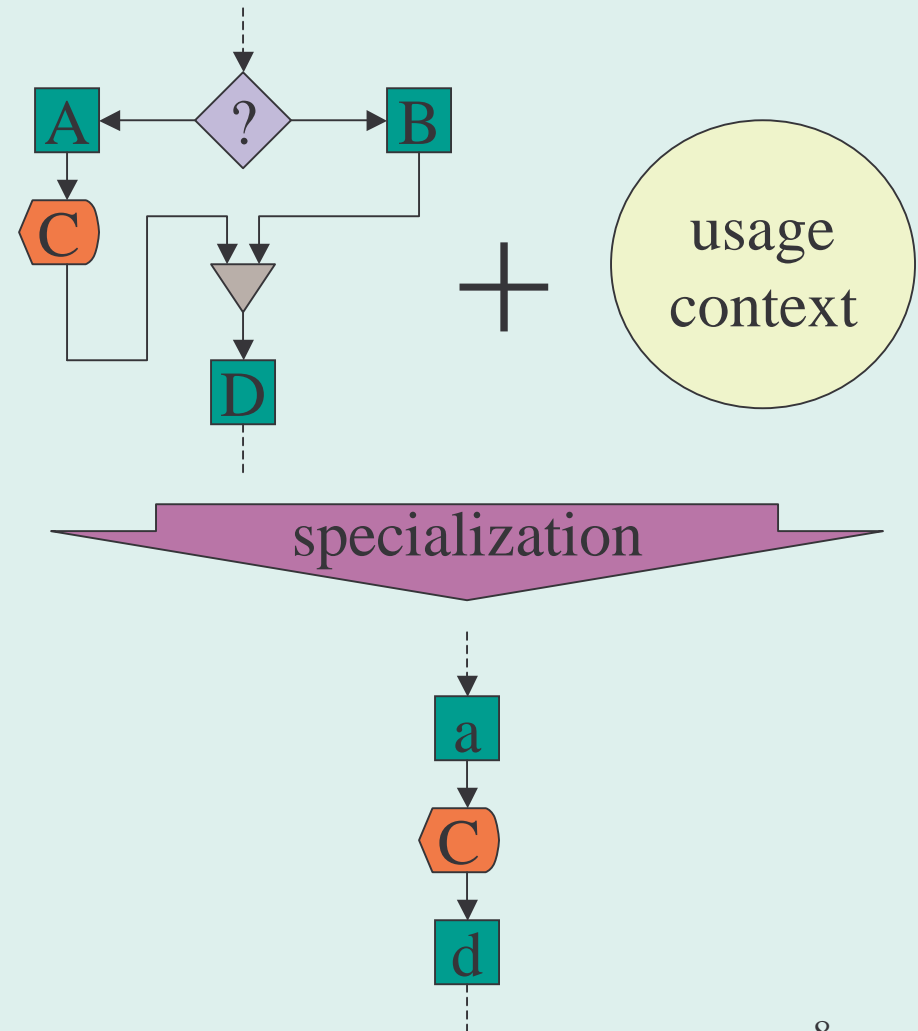
Overhead in Strategy Pattern

- Virtual dispatch is used to choose pixel drawing strategy
- Eliminating virtual dispatches can give up to two-times speedup
- But from a local point of view, the strategy can change at any moment



Solution: Program Specialization

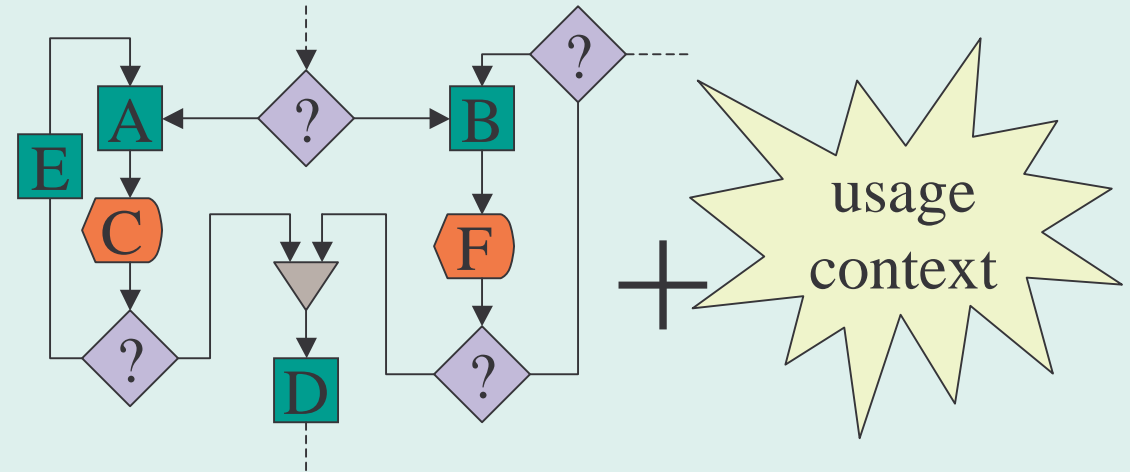
- An adaptable software system with a very general behavior...



- ...is specialized to only include the behavior needed in a specific usage context

Problem: Complexity

- Complex software system



- The usage context must reflect the program structure
- The potential for adaptation introduced by design patterns complicates the program structure

What to do?

What to do?

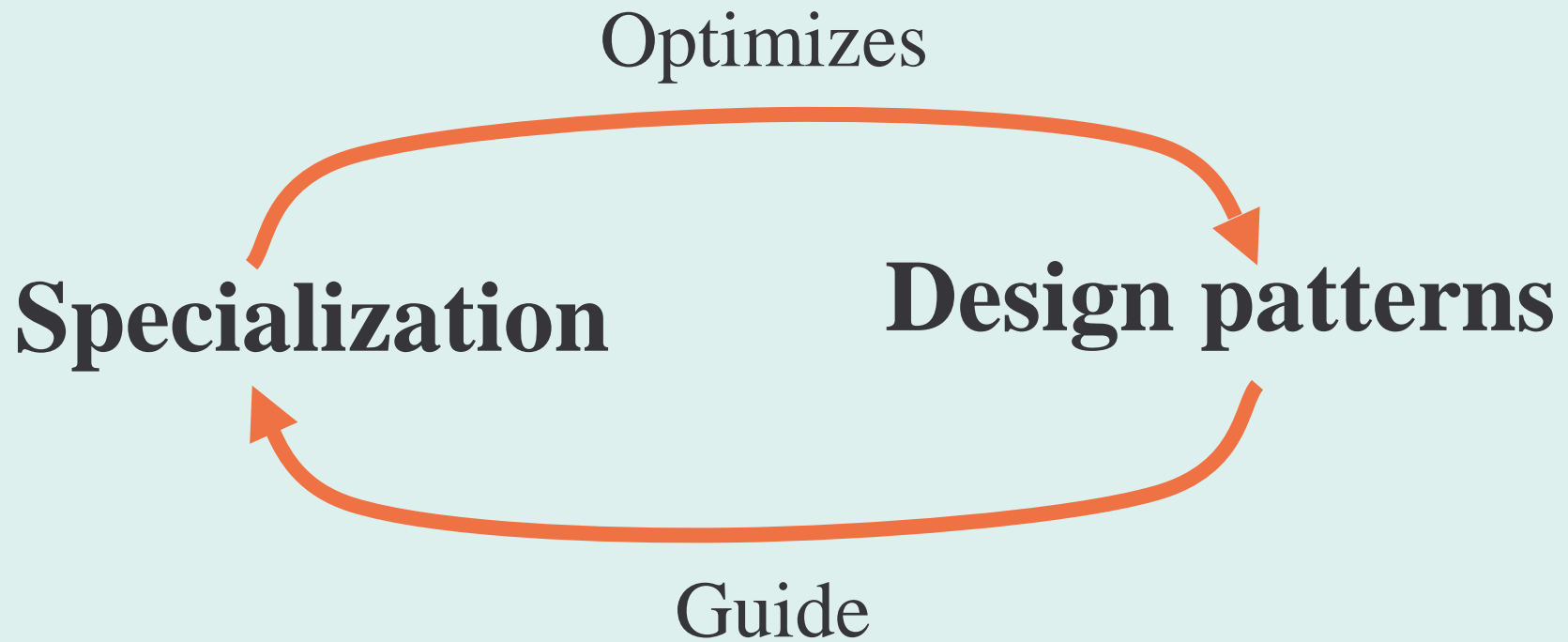
Look to existing methodology for dealing
with complexity in programs:

What to do?

Look to existing methodology for dealing
with complexity in programs:

Design Patterns!

Our Approach



Specialization Patterns

- A design pattern encapsulates all aspects of a general solution to a design problem
- A **specialization pattern** encapsulates all aspects of specializing uses of a design pattern

Write your program according to design patterns
Make it efficient using specialization patterns

Specialization Pattern Template

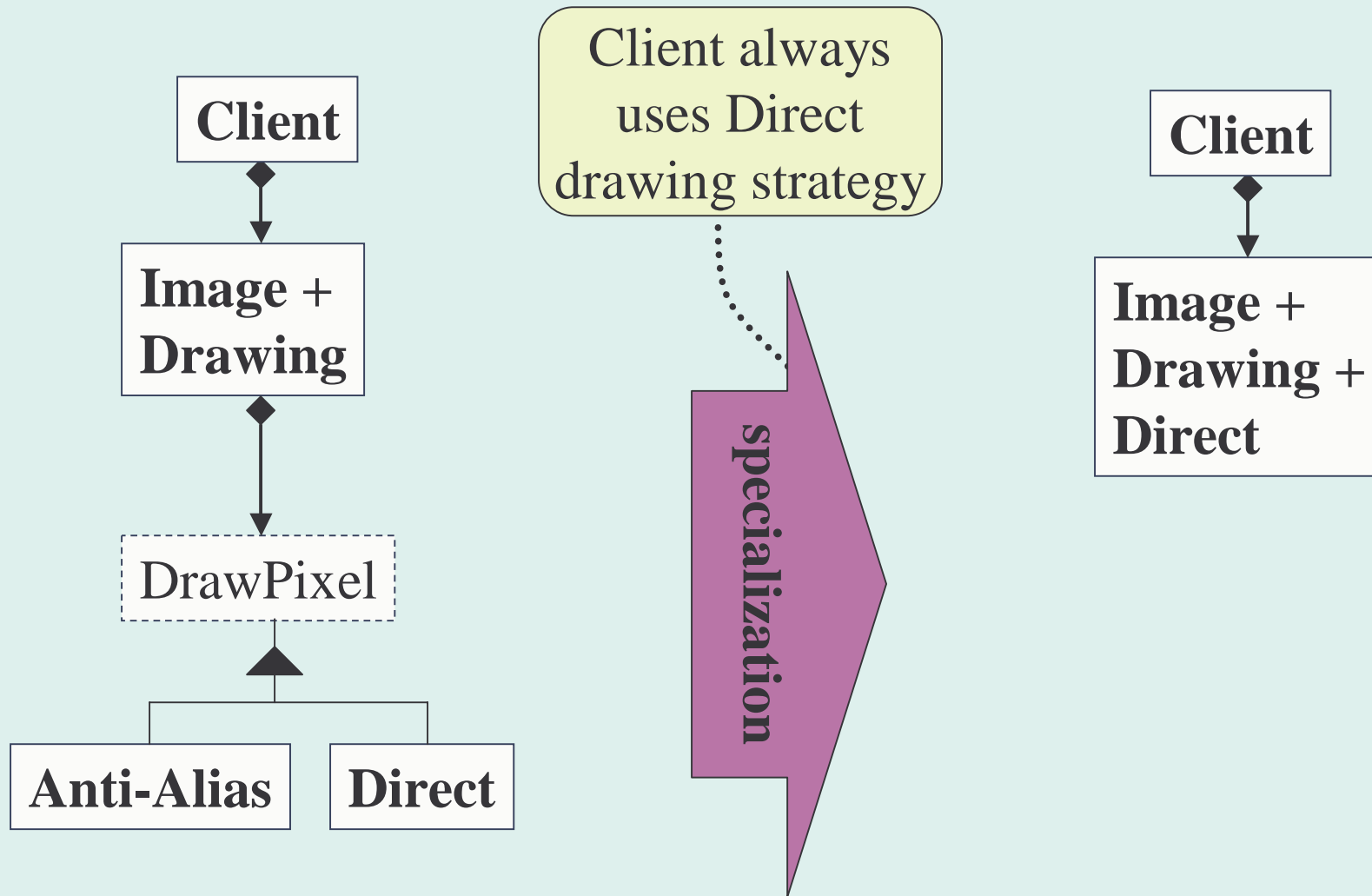
A specialization pattern

- describes overheads from using a design pattern
- describes a specialization approach for eliminating these overheads
- defines conditions for when specialization is useful
- gives examples illustrating specialization
- rates the specialization pattern applicability

Example: Strategy Specialization Pattern

- Overhead: algorithms are accessed through an abstract interface
- Approach: specialize to particular algorithm
- Condition: algorithm must not change, or a local invariant must be artificially introduced outside critical code
- Applicability: high when algorithm used repeatedly

Applying The Specialization Pattern



Specialization Pattern Catalog

In the paper:

- Builder (creational pattern); medium applicability
- Strategy (behavioral pattern); high applicability

In addition:

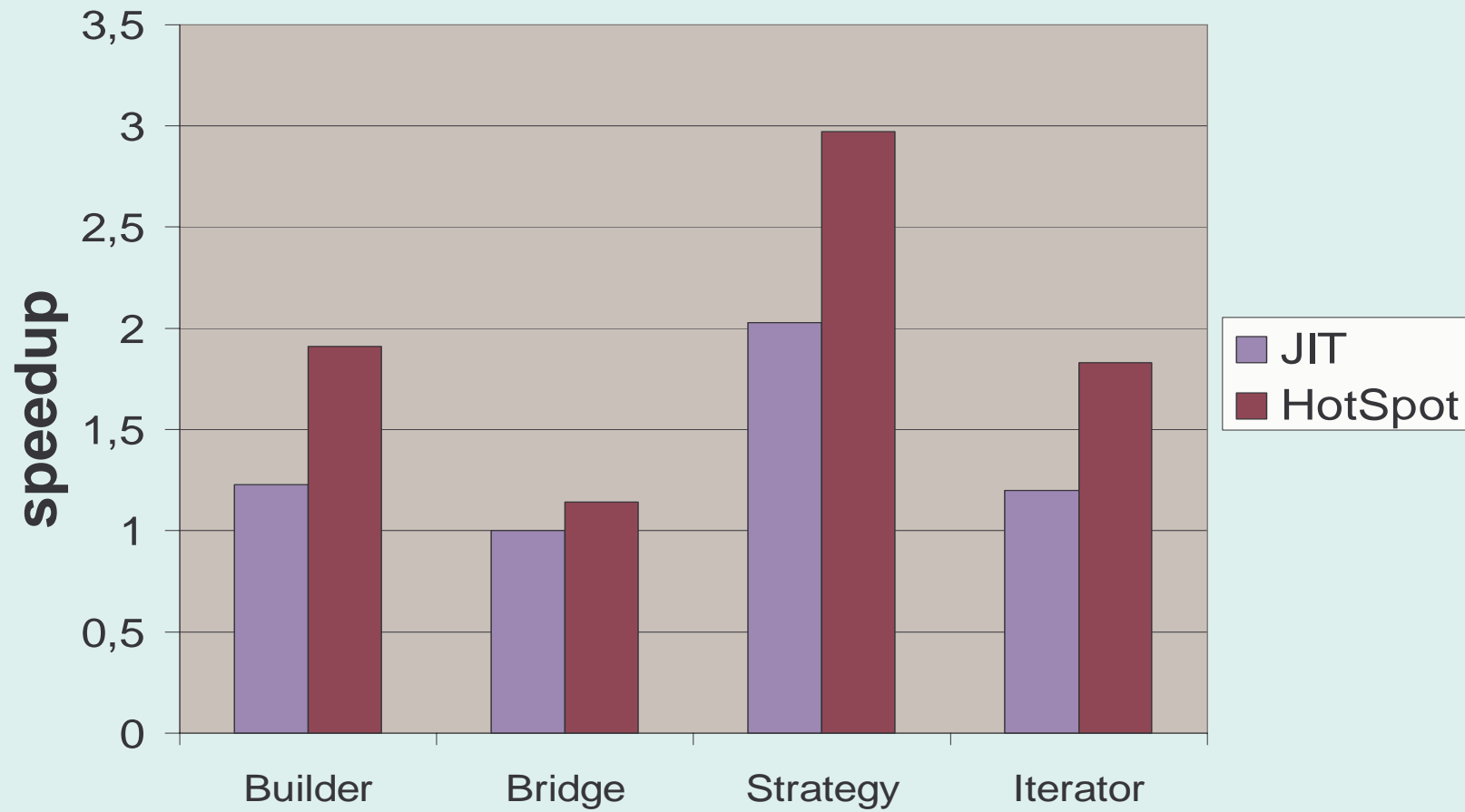
- Bridge (structural pattern); low applicability
- Iterator (behavioral pattern); medium applicability

JSpec

- Automatic program specializer for Java
- Incorporates a wide range of aggressive analyses and transformations
- Specializes Java source or bytecode, and produces Java source code
- Controlled using declarative language

[Initial prototype presented at ECOOP'99, public release soon]

Benchmarks



Conclusion & Future Work

Specialization patterns:

- encapsulate knowledge about specialization
- guide the specialization process
- make design patterns efficient
- tested with JSpec

Next: CASE tool or language support

More information: <http://www.irisa.fr/compose>

Specialization

