Façades, Strategies, and Templates

BRIAN LAMARCHE

COMPUTER SCIENCE 323 – SOFTWARE DESIGN

Component

Module

Software Package (software development kit)

Web Service

Components decouple elements by providing or requiring (depending on) interfaces

Component

Components can require several interfaces

A component can encapsulate a large sub-system of objects with various interfaces

These objects can have highly functional cohesive intention

Component

While the objects themselves are loosely coupled, and the component is well modularized from the context (application) use of the component’s objects can be daunting

How to create a simple interface to a sub-system of objects?

Consider a compiler

Parser

Tokenizer

Scanner

Statement evaluation

Expression Evaluation

Semantic Analyzer

Code generator

Stream Writer (e.g. assembly, binary)

Stream

….

Consider a compiler

Parser

Tokenizer

Scanner

Statement evaluation

Expression Evaluation

Semantic Analyzer

Code generator

Stream Writer (e.g. assembly, binary)

Stream

….

Various elements make it difficult for the context to use!

Consider a compiler

Parser

Tokenizer

Scanner

Statement evaluation

Expression Evaluation

Semantic Analyzer

Code generator

Stream Writer (e.g. assembly, binary)

Stream

….

Consider:

Compile(“print hello world”);

Façade

The façade is a structural pattern

Intentions

- Create a unified interface to a set of interfaces in a subsystem
- This higher level interface makes the sub-system easier to use.

Façade Players

Façade

- Knows which subsystem classes are responsible for a request
- Delegates client requests to those subsystem objects

Subsystem classes

- Implement the subsystem functionality
- Handle work assigned by the façade object
- Have no knowledge of the façade (keep no references or associations to it)

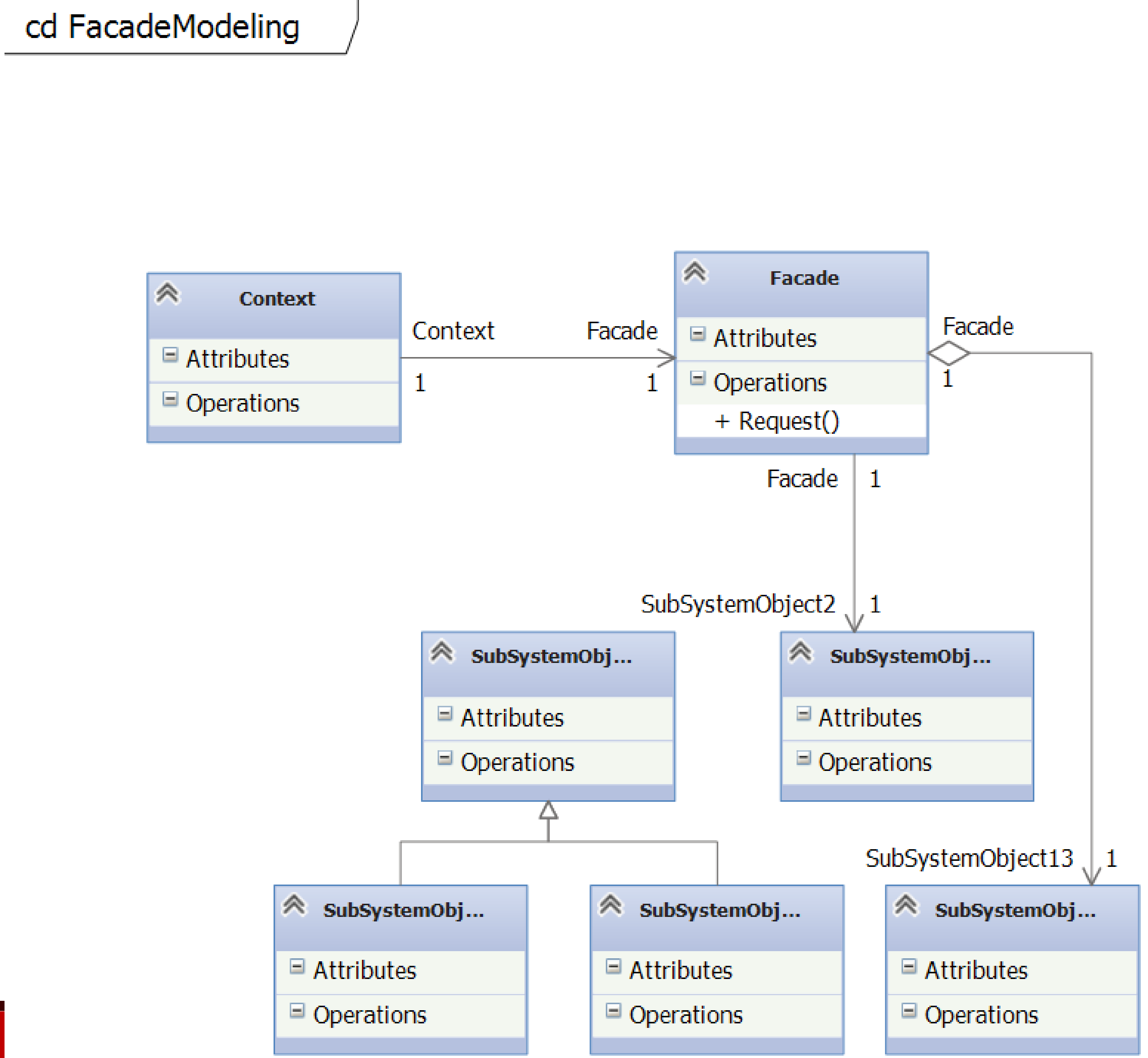
Façade Players

Façade

- Knows which subsystem classes are responsible for a request
- Delegates client requests to those subsystem objects

Subsystem classes

- Implement the subsystem functionality
- Handle work assigned by the façade object
- Have no knowledge of the façade (keep no references or associations to it)
    - CONSIDER EVENTS
    - CONSIDER DELEGATES
    - CONSIDER SYNCHRONOUS REQUESTS (message calls)


cd FacadeModeling
Context
Attributes
Operations
Context
1
Facade
1
Facade
Attributes
Operations
+ Request()
Facade
1
Facade
1
SubSystemObject2
1
SubSystemObj...
Attributes
Operations
SubSystemObj...
Attributes
Operations
SubSystemObject13
1
SubSystemObj...
Attributes
Operations
SubSystemObj...
Attributes
Operations
SubSystemObj...
Attributes
Operations

Façade

Does the façade promote coupling?

Why? Or Why not?

Façade

The façade actually helps **reduce** coupling between your **application** and the **subsystem**.

It allows you to swap out the subsystem easily through the concrete façade

Related Patterns

Mediator

- Abstracts communication from objects, where the façade abstracts functionality from objects
- With mediator, existing objects can know about the mediator.
- With façade, existing objects know nothing

Abstract Factory

- Create objects from a sub system

Façade and the project

Components and facades, almost go hand in hand

Components can be object rich, and provide many interfaces

Coordination of these objects is pinnacle

Façade and the project

Image Processing

- EMGU/OpenCV/AFORGE
  - Frame Co-adding
  - Edge Detection
  - Skeletonization
  - Shape Recognition
  - Shape Confidence Calculation

Image Processing brings up an interesting question

What algorithms are the best to try?

How to test?

Image Processing

You’ll want to test several algorithms.

If you embed them in a façade or mediator, you’ll couple image processing to your context.

Keep beating on the Coupling Drum….

Image Processing

Use interfaces or abstract classes to help separate your algorithm implementation from your context

This implementation is called the Strategy pattern.

Strategy

This is actually called the strategy pattern

Behavioral

Define a family of algorithms, encapsulate each one, then make them interchangeable.

Strategy Benefits

Allows you to vary an implementation without have to vary the existing context

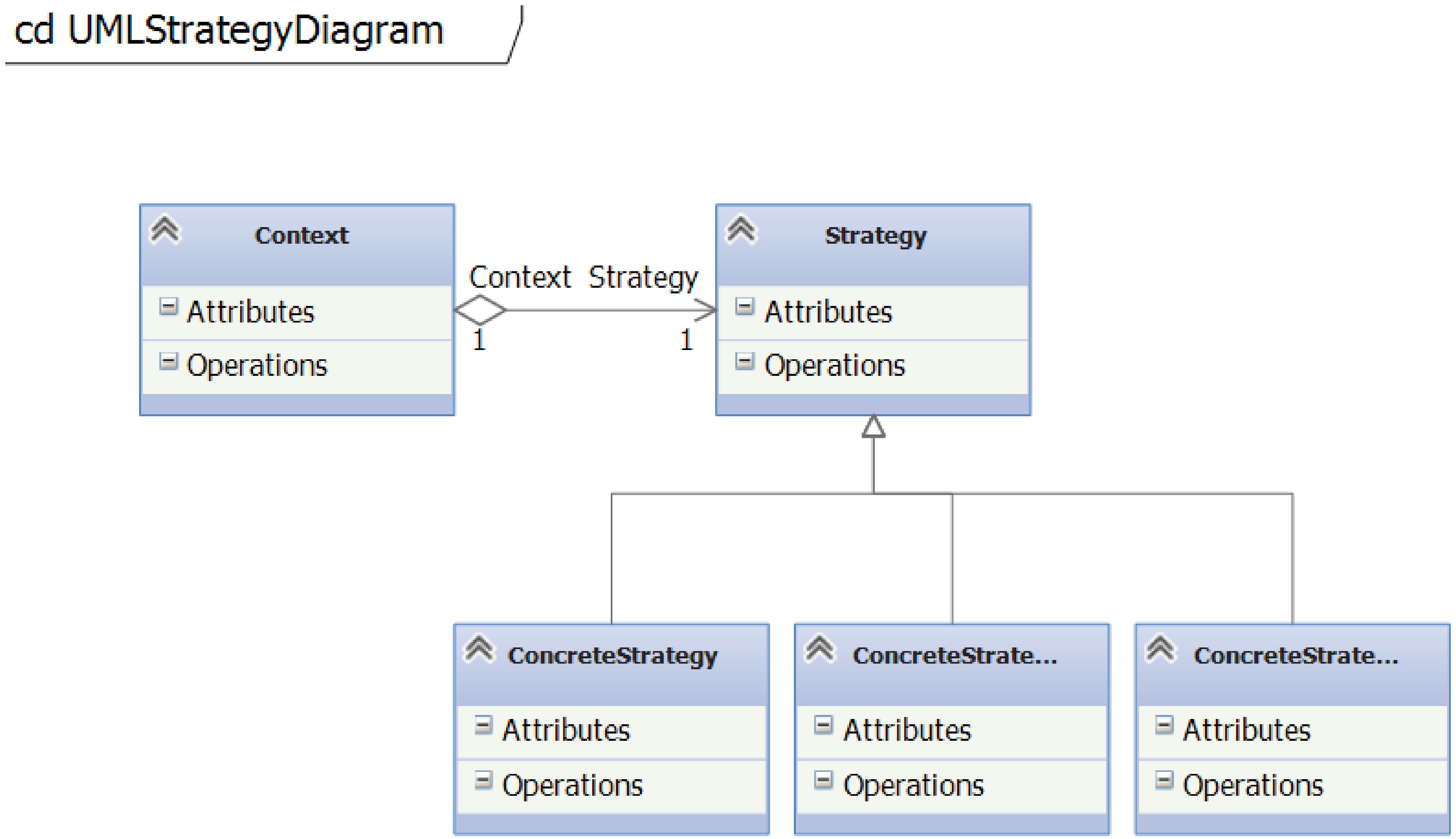
Specific algorithms are required at specific times

Strategy Players

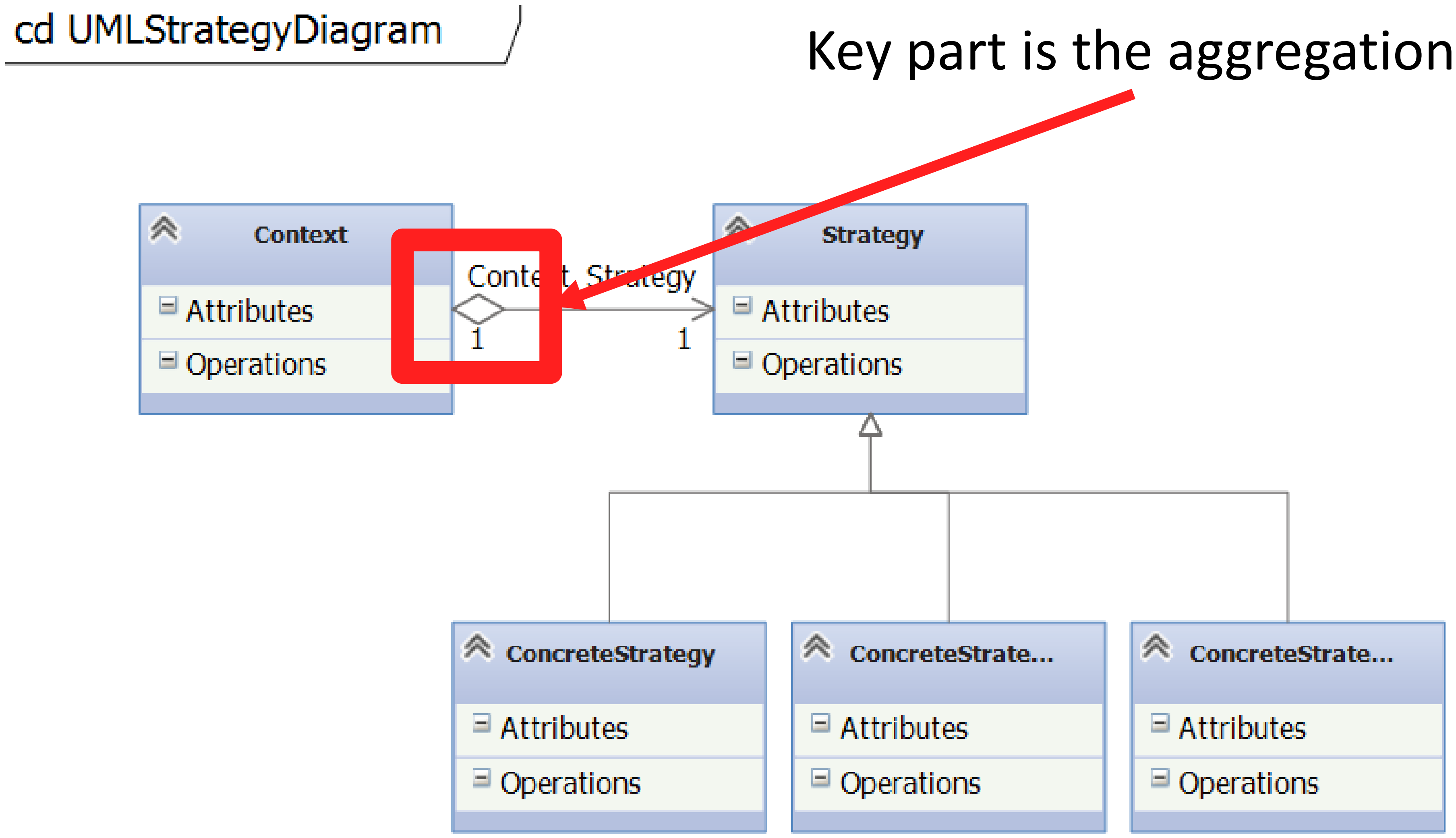
Strategy
- Declares the interface (abstract class or your interface)

Concrete Strategy
- Implements the algorithm use the strategy interface

Context
- Maintains a reference to the strategy
- Configured with the Concrete Strategy
- May define an interface for Strategy to access its data


cd UMLStrategyDiagram
Context
Attributes
Operations
Context
Strategy
1
1
Strategy
Attributes
Operations
ConcreteStrategy
Attributes
Operations
ConcreteStrate...
Attributes
Operations
ConcreteStrate...
Attributes
Operations

Key part is the aggregation



Strategy

Strategies can have similar behaviors

- Sorting
  - Merge vs. Quick

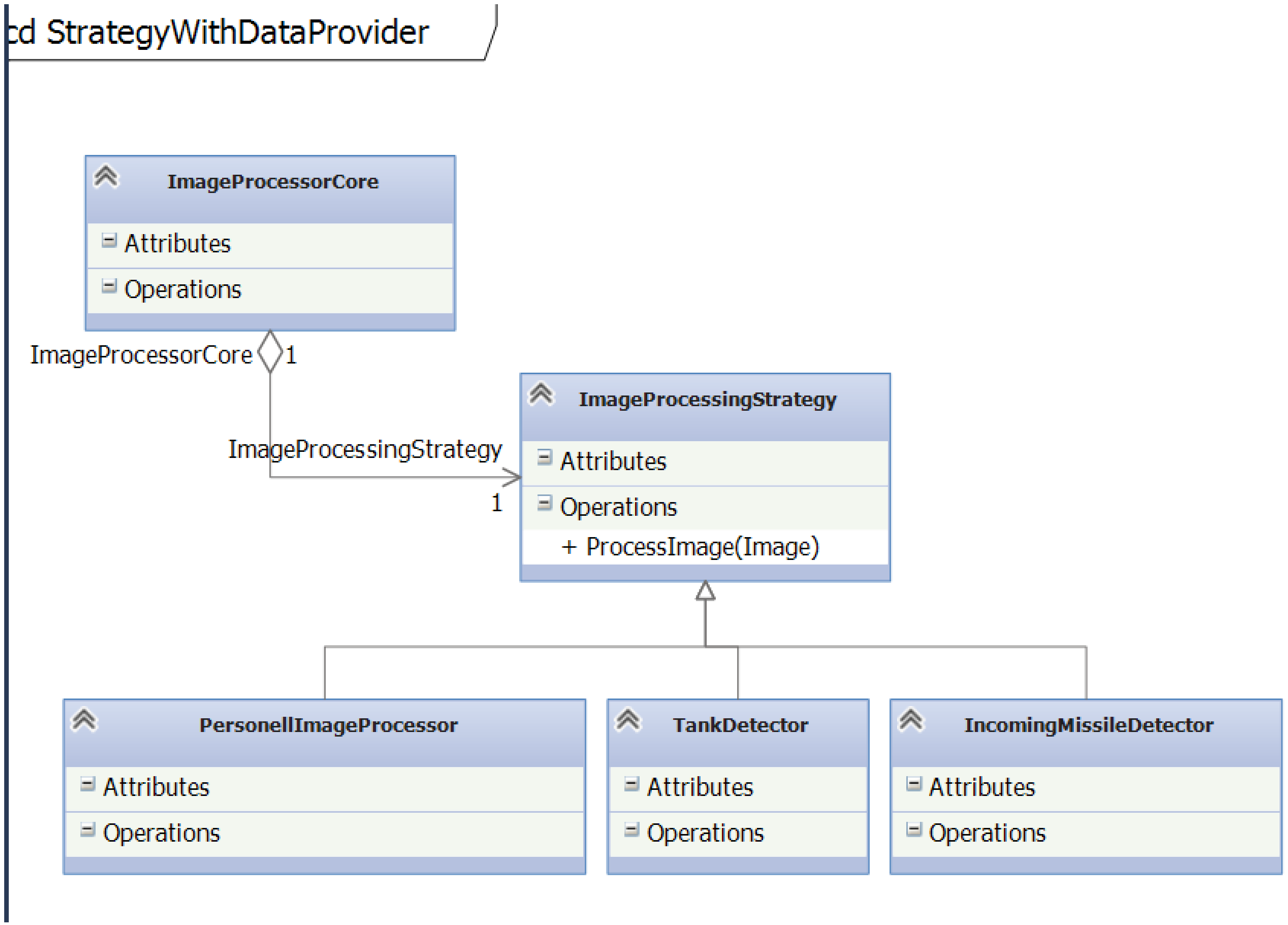
Client can choose from behaviors

- But the client should be aware of the potential implications
- Merge vs. Quick sort has its tradeoffs

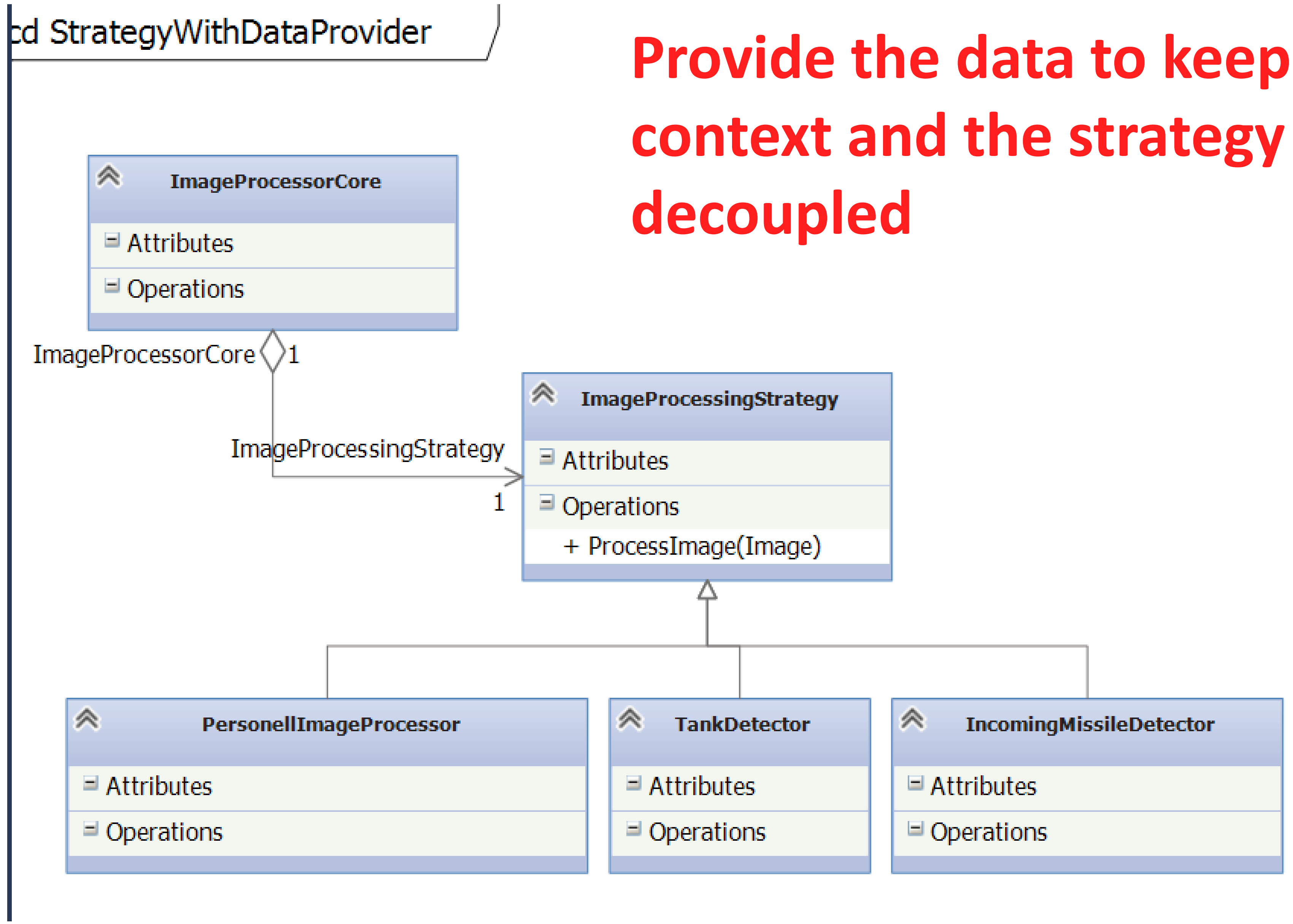
Providing Data to a Strategy

Let’s say you have an image processing application

Initially, files are small, and your application has to process a single image at a time.

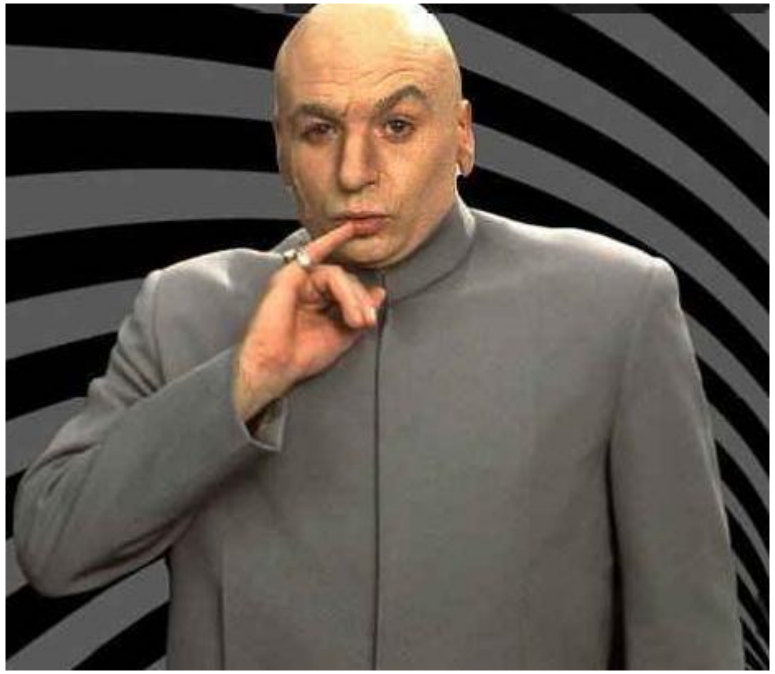

cd StrategyWithDataProvider
ImageProcessorCore
Attributes
Operations
ImageProcessorCore
1
ImageProcessingStrategy
1
ImageProcessingStrategy
Attributes
Operations
+ ProcessImage(Image)
PersonellImageProcessor
Attributes
Operations
TankDetector
Attributes
Operations
IncomingMissileDetector
Attributes
Operations

Provide the data to keep the context and the strategy decoupled



Providing Data to a Strategy

But let's say you have to process several images, no no no…several THOUSAND images….no wait…Millions of high resolution images.



Providing Data to a Strategy

You can’t give the data to the strategy all at once, you’ll run out of memory

How do you fix this problem?

Providing Data to a Strategy

First Consider

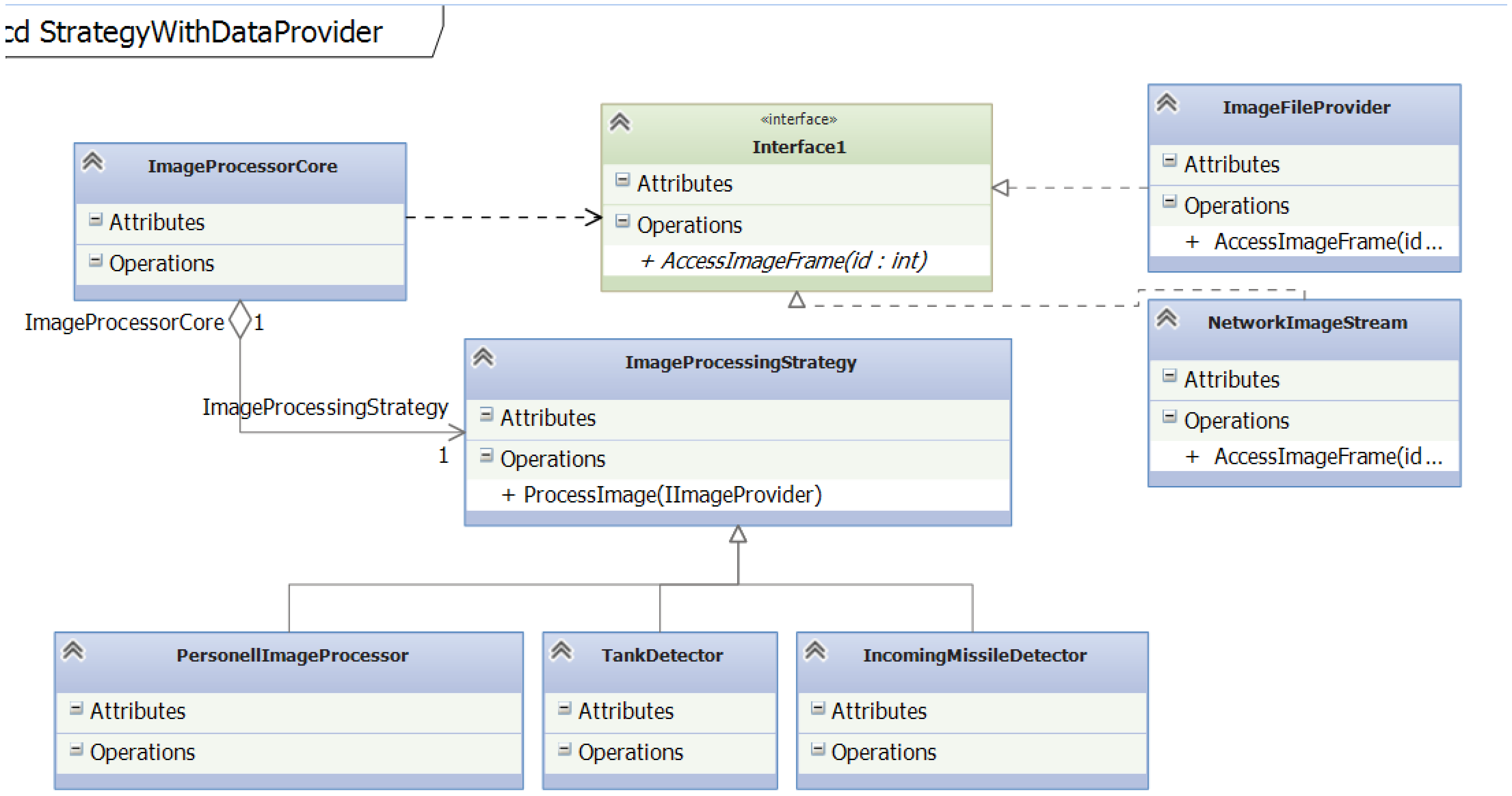
- What format are the images in now?
  - Files?
  - Network video stream?

You cannot, and should not, assume that you can just give it a million paths to image files.

Provider

Give it an interface to access data from the strategy

```
Image GetImage(int imageId)
```


cd StrategyWithDataProvider
ImageProcessorCore
Attributes
Operations
«interface»
Interface1
Attributes
Operations
+ AccessImageFrame(id : int)
ImageFileProvider
Attributes
Operations
+ AccessImageFrame(id ...
NetworkImageStream
Attributes
Operations
+ AccessImageFrame(id ...
ImageProcessorCore
1
ImageProcessingStrategy
1
ImageProcessingStrategy
Attributes
Operations
+ ProcessImage(IImageProvider)
PersonellImageProcessor
Attributes
Operations
TankDetector
Attributes
Operations
IncomingMissileDetector
Attributes
Operations