

Flyweight Design Pattern



**CSE 776 DESIGN PATTERNS
SUMMER '10**

**BHARATH SETTY
Dr. JAMES FAWCETT**

Intent

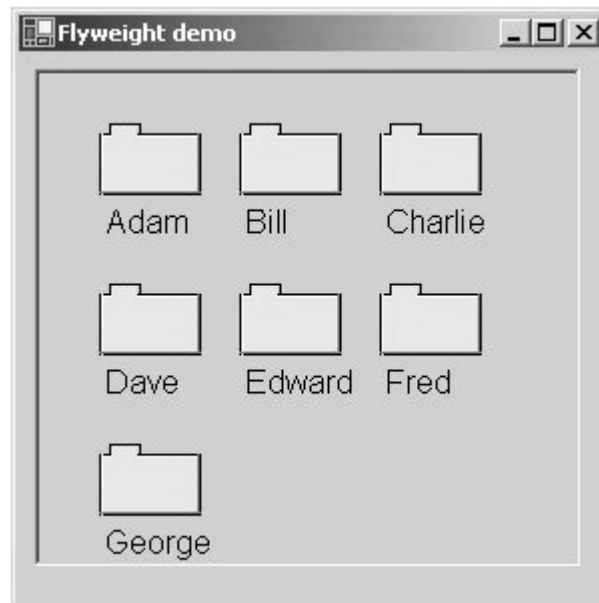


- **“USE SHARING TO SUPPORT LARGE NUMBERS OF FINE-GRAINED OBJECTS EFFICIENTLY.”**
 - One object instance holding shared (intrinsic) state
 - Unique (extrinsic) state is stored outside of the shared object

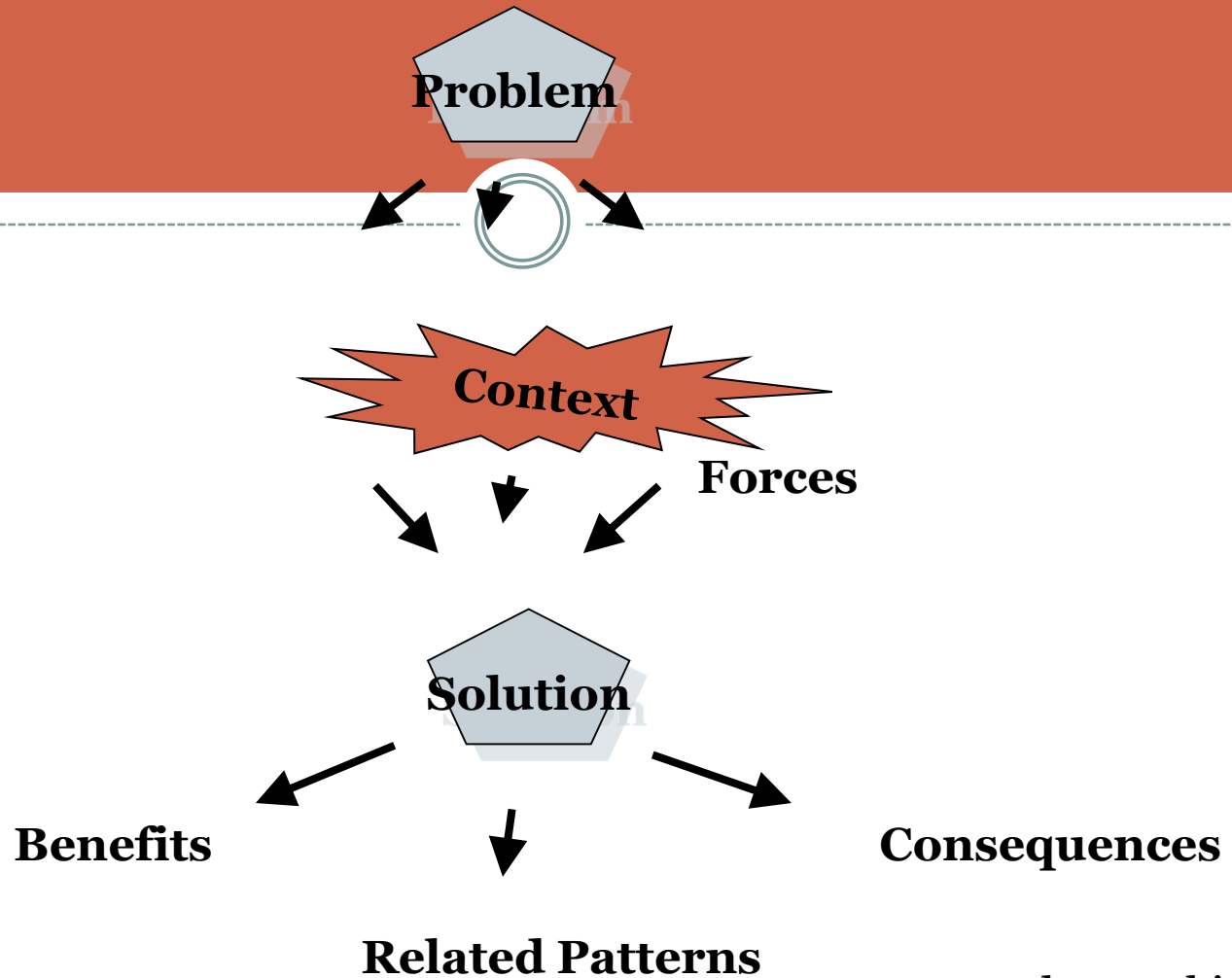
Motivation



Folder Representation



Forces



John Reekie (UTS)

Forces

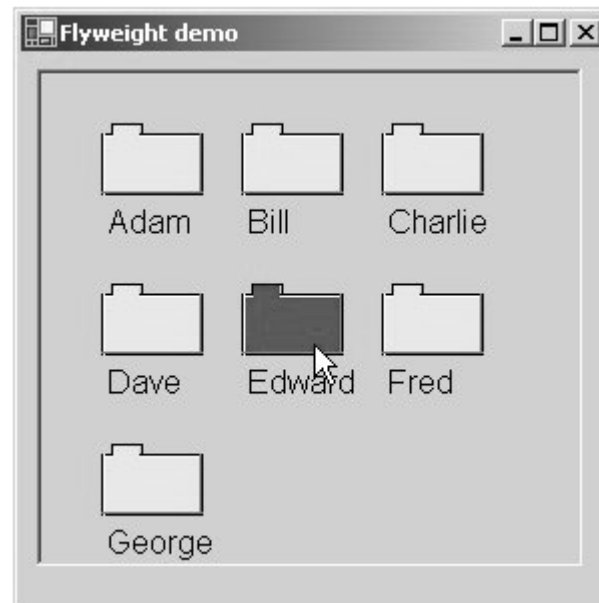


- To obtain a fine grained object structure
- We will possibly have many objects
- Cost of storing a copy of each object is high

Motivation Example



Folder Representation



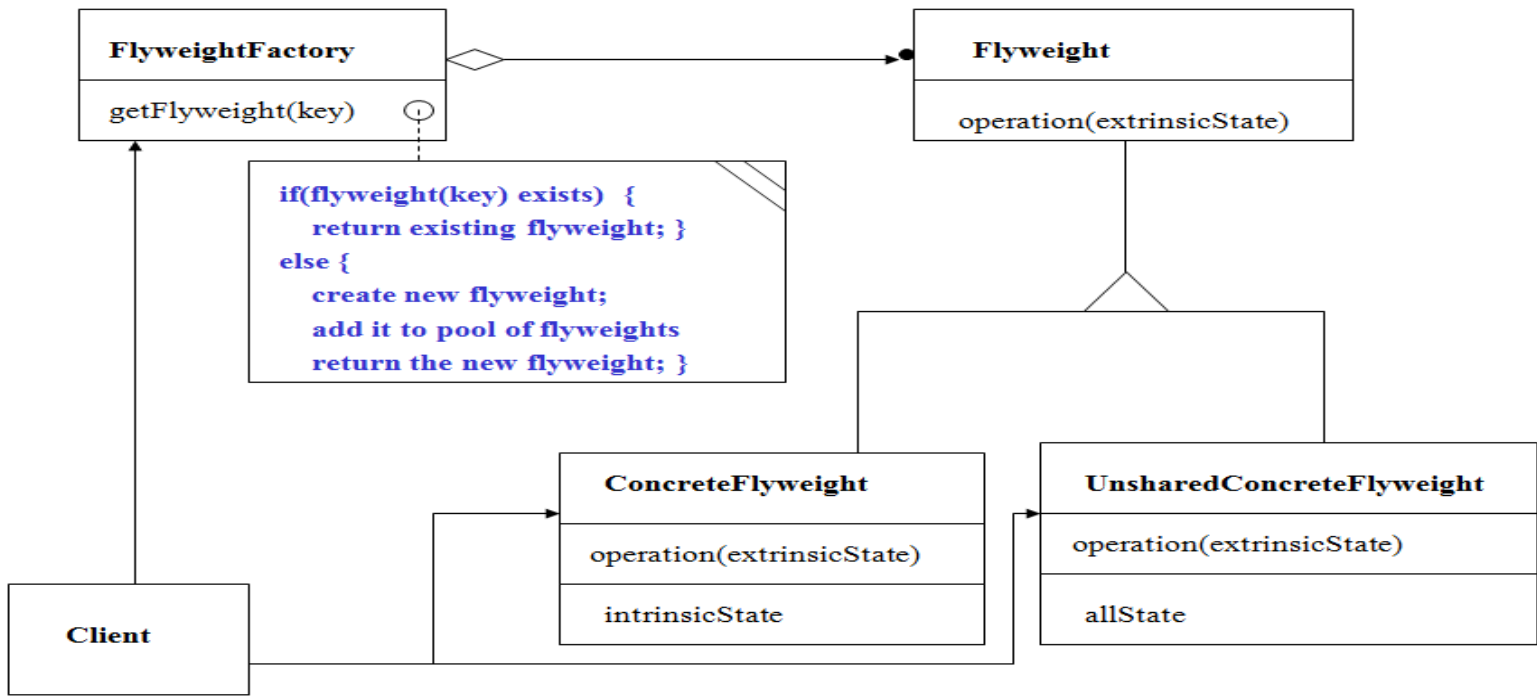
Applicability



FLYWEIGHT PATTERN IS USED WHEN ALL OF THE FOLLOWING ARE TRUE

- An application has a large number of objects
- Store costs are high
- Most object state can be made extrinsic???
 - Authors claim. I strongly disagree
 - You want most of object state intrinsic, e.g., shared
- Many groups of objects will be replaced by few shared objects(intrinsic)
- The application doesn't depend on object identity

Structure

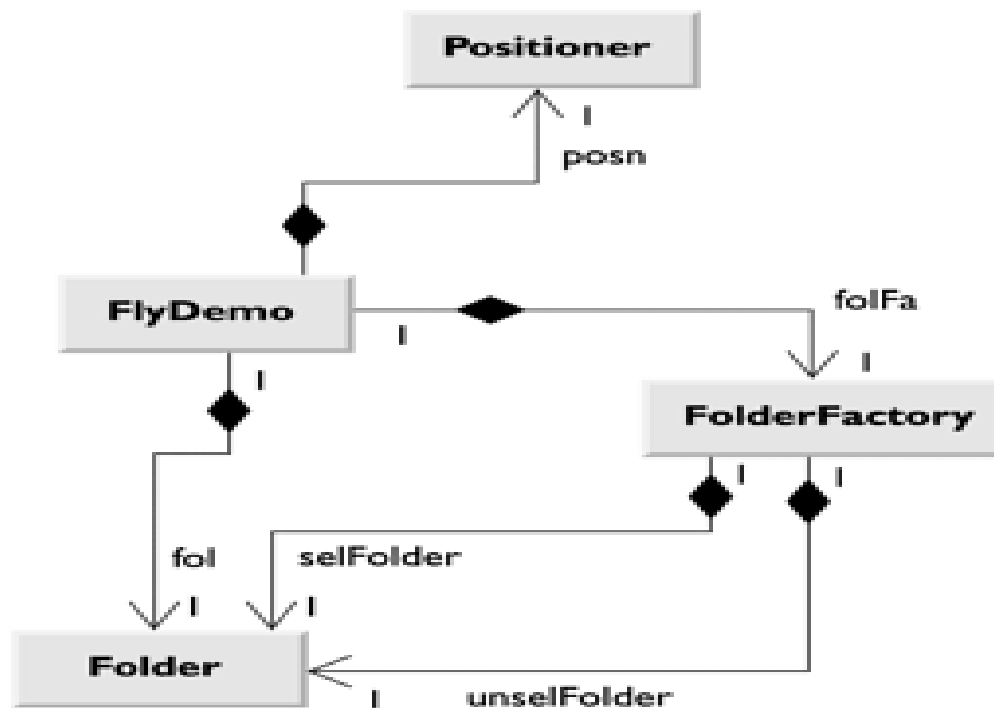


Participants



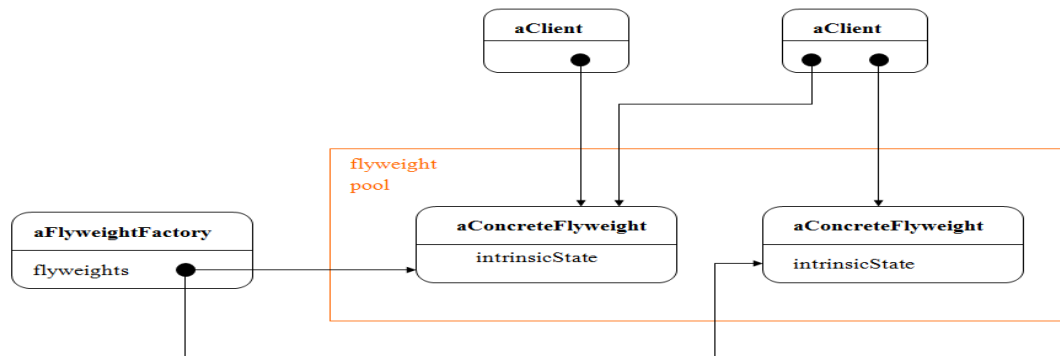
- **Flyweight (Window)**
 - Declares interface that flyweights can use to receive and act on intrinsic state
- **ConcreteFlyweight (Icon)**
 - Implements flyweight interface and adds storage for intrinsic state. Must be shareable
- **UnsharedConcreteFlyweight (Name, Location)**
 - Commonly has ConcreteFlyweights as children
- **Flyweightfactory**
 - Creates and manages flyweight objects
- **Client**
 - Maintains references to flyweights
 - Computes or stores extrinsic state of flyweights

Application



Collaborations

- **State of Flyweight is characterized by intrinsic and extrinsic state**
 - Intrinsic state stored in ConcreteFlyweight
 - Extrinsic state stored or computed by Client Objects
- **Clients should not instantiate ConcreteFlyweights directly**
 - Proper sharing will not occur



Consequences



- **Pros:**
 - Cost saved by space savings (Function of reduction of number of instances and amount of intrinsic state per object)
- **Cons:**
 - Cost increased in run-time to transfer, find or compute extrinsic state

Implementation



- **Removing Extrinsic State**

- Must be easily identifiable and be removed from shared objects
- Pattern is only useful if state can be shared

- **Managing Shared Objects**

- Clients should not instantiate ConcreteFlyweights directly
- Flyweight factory allows clients to locate a particular flyweight
- Reference counting and garbage collection can be used

Known Uses



- **2D/3D Vector drawing program**
- **2D/3D Video game**
- **Cad applications**

Related Patterns



- Flyweight is often combined with Composite pattern to implement a logically hierarchical structure in terms of a graph with shared leaf nodes
- State and Strategy can be implemented as flyweights
 - State: An object can alter its behavior when its internal state changes
 - Strategy: Define a family of algorithms and make them interchangeable

References



- **Design Patterns, Elements of Reusable Object-Oriented Software, Erich Gamma, et. al.**
- <http://www.informit.com/articles/article.aspx?p=31563&seqnum=3>
- http://sourcemaking.com/design_patterns/flyweight
- <http://www.xml.com/pub/a/2000/01/19/feature/index.html?page=3>
- <http://www.blackwasp.co.uk/flyweight.aspx>
- <http://www.ecs.syr.edu/faculty/fawcett/handouts/cse776/presentations-students/flyweight/>

Questions????

