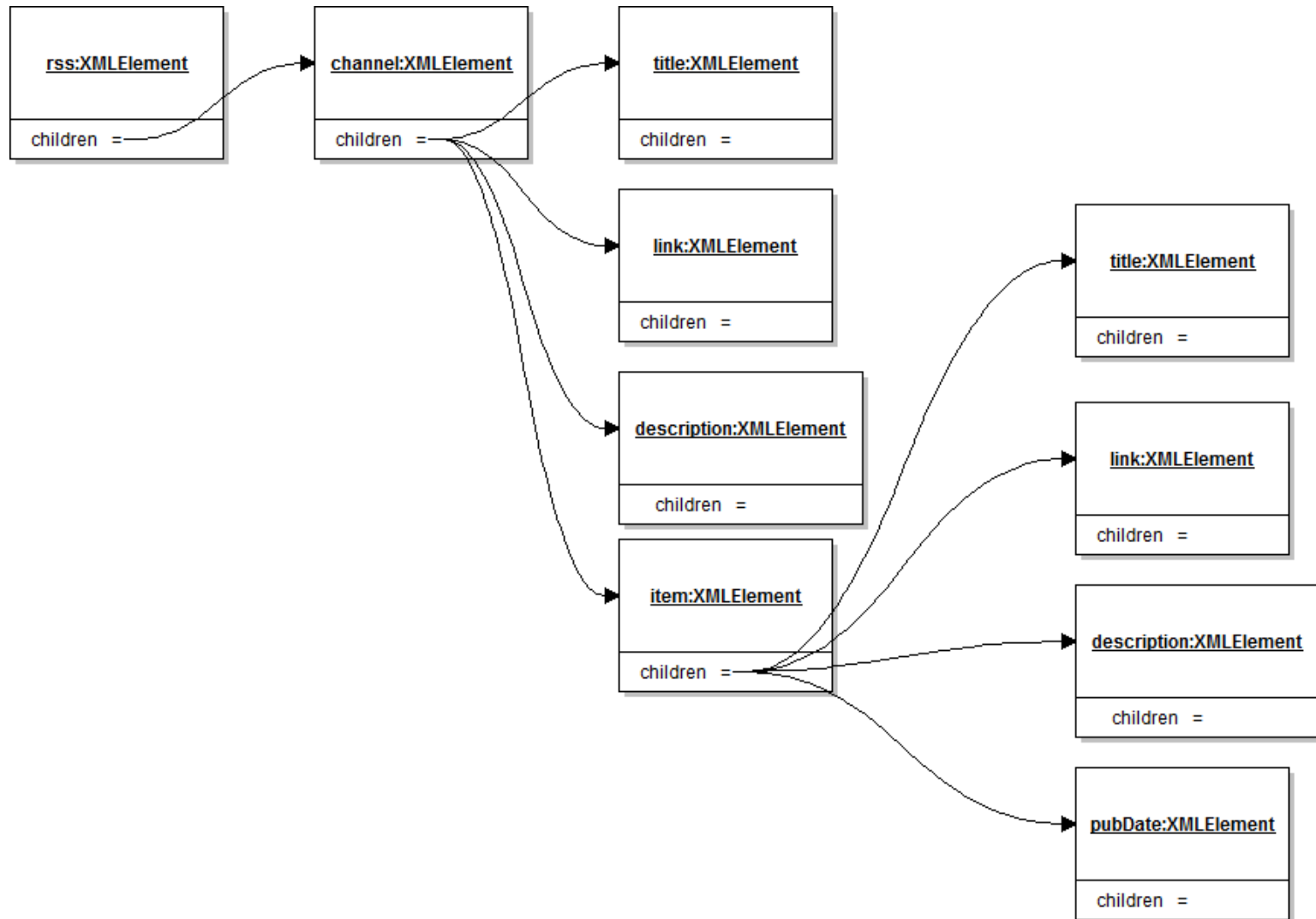


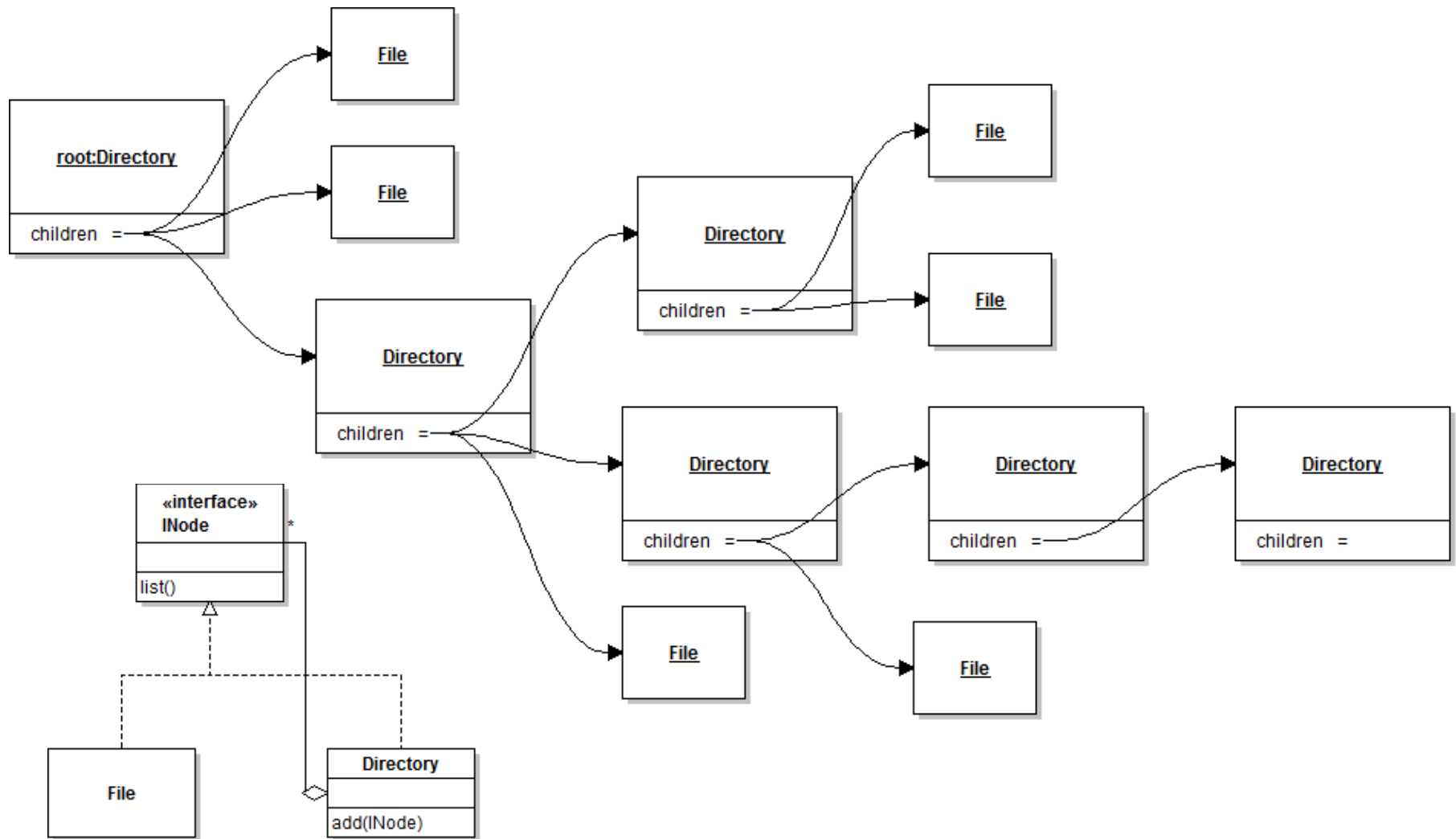
Today: Visitor Design Pattern

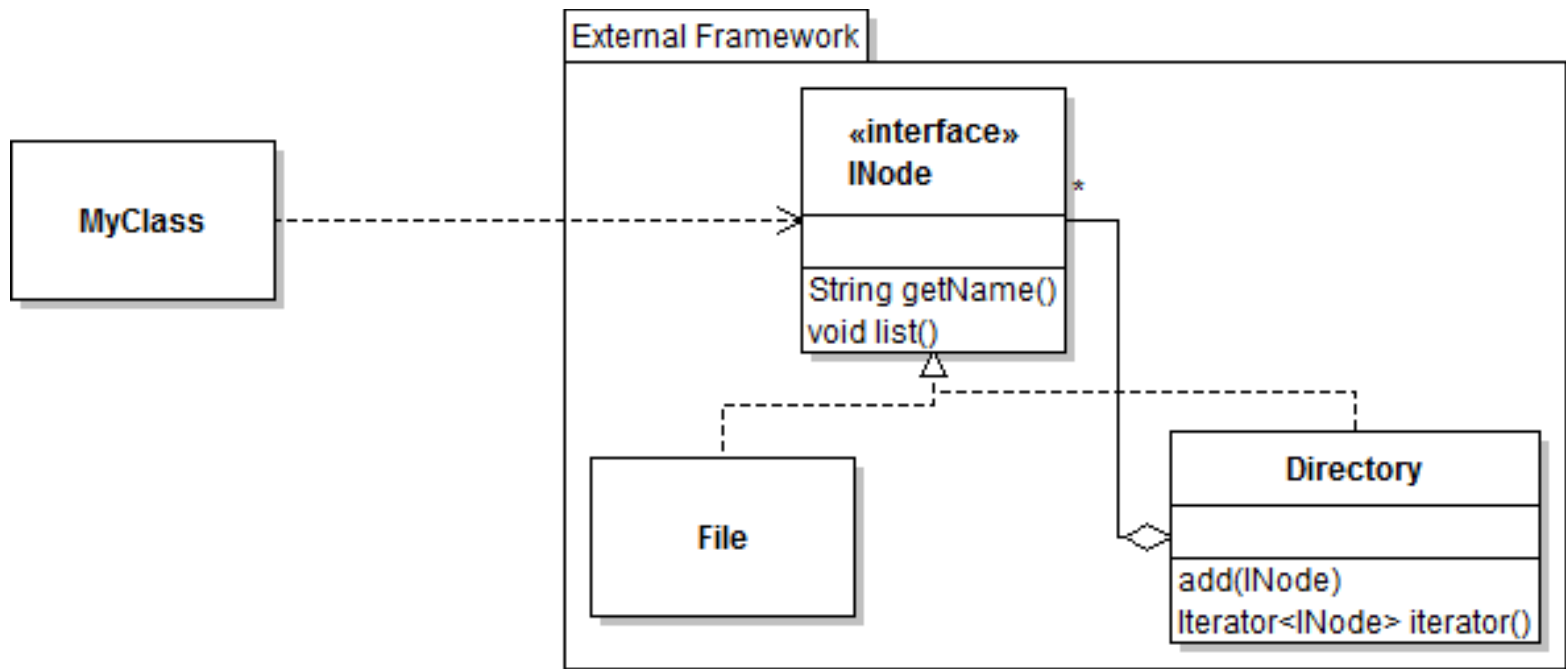
1. Why is this needed?
2. How it works
3. Detailed example illustrating variants

A RSS Object Graph



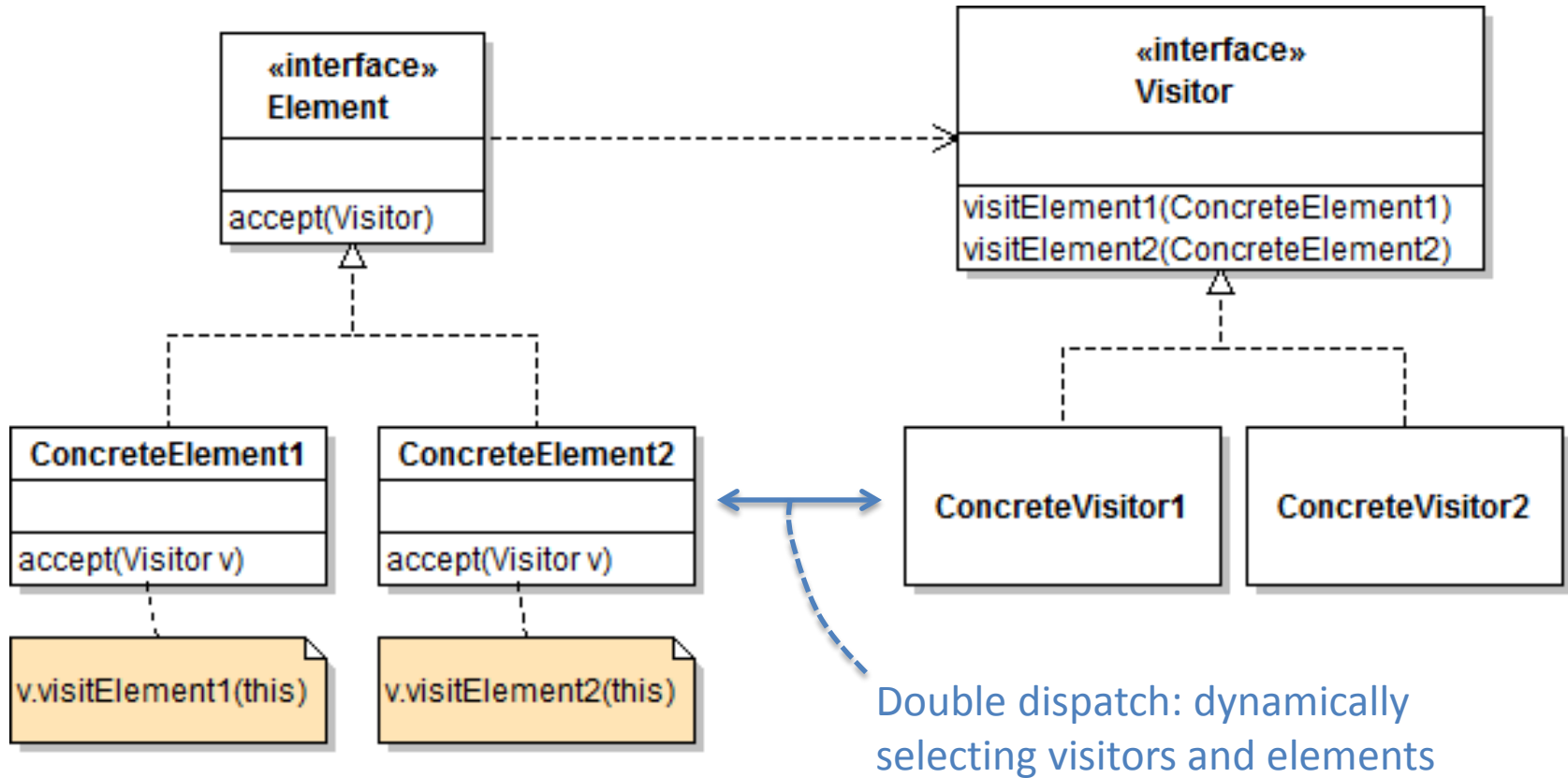
Recursive Object Graphs



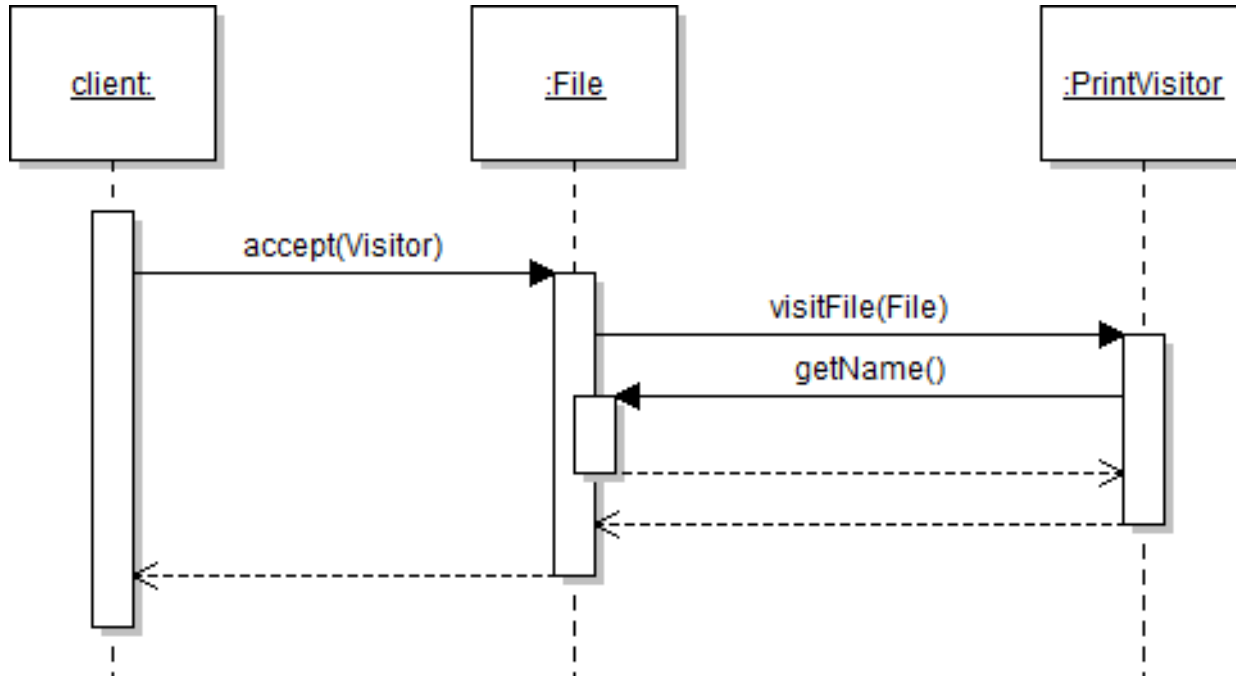


Requirement: Support an open-ended number of operations on a (relatively fixed) recursive structure.

Visitor Design Pattern Class Structure



Double Dispatch



`node.accept(new PrintVisitor())`

Dynamic dispatch based on node type

```
public void accept(Visitor v)
{
    v.visitFile (this);
}
```

Dynamic dispatch based on visitor type

Visitor: Structure Traversal Code

Advantages

Disadvantages

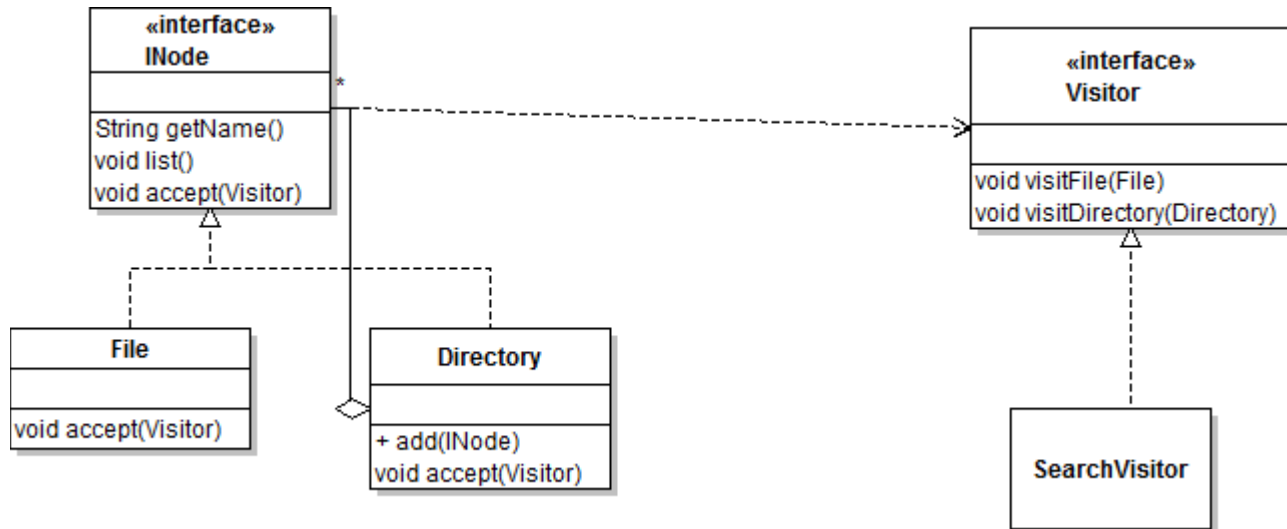
In Element
Hierarchy

- Better encapsulation of the recursive structure and information hiding
- Visitor does not need to worry about the recursion

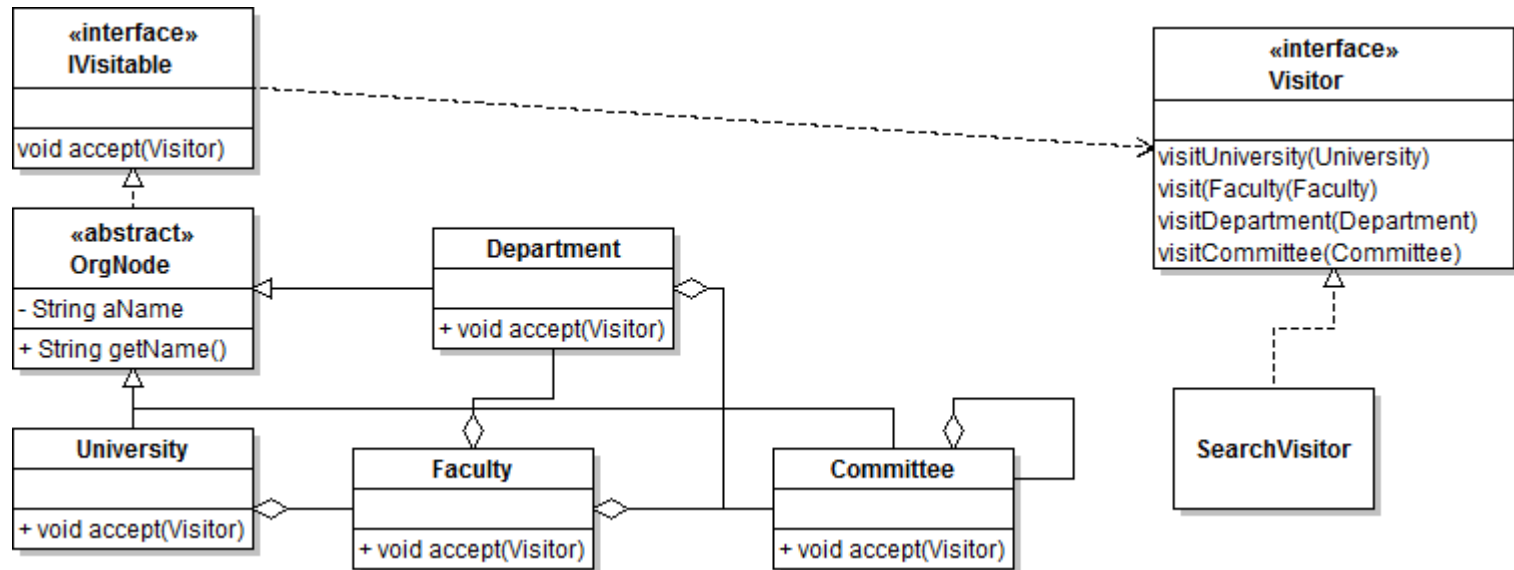
- No flexibility for traversal order
- Scattering of recursion code

In Visitor

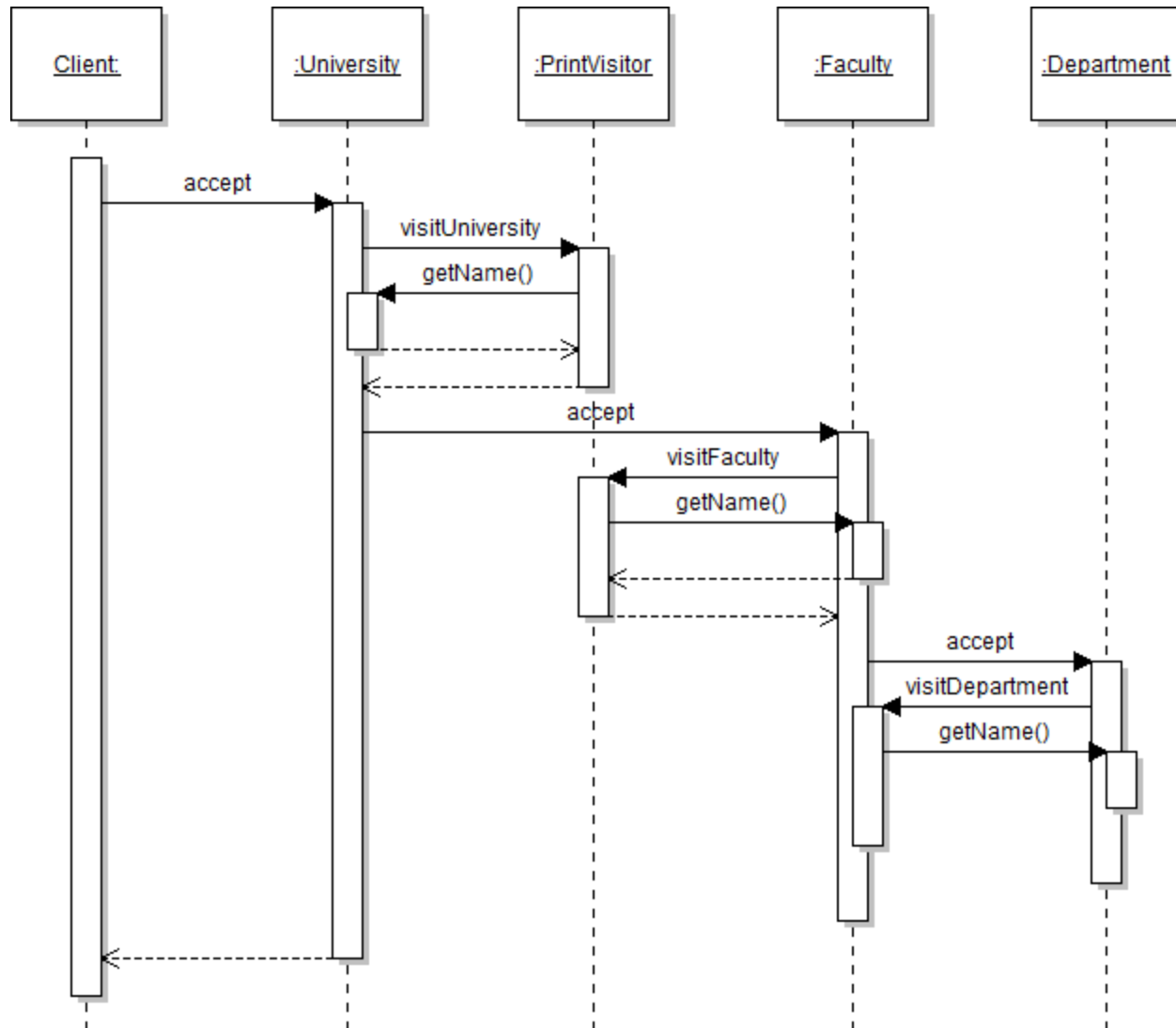
Visitor Support of the Node Hierarchy



Visitor Support of the University Hierarchy



Traversal Code in the accept Methods



Traversal Code in the visit Methods

