**Instructor’s Solution**

**CSE687 – Object Oriented Design**

**Project #1**

Jim Fawcett, February, 28 2010

Purpose:

Evaluate the design of XmlProcessor project with respect to partitioning, complexity, reusability, and use of Design Patterns.

1. **Partitioning:**XmlProcessing has 8 packages, including an interface package and a package used only for testing.
	1. Packages are reasonably balanced in size and complexity, as shown in the diagrams, below.
	2. The worst metrics for individual functions are 68 lines of code and cyclomatic complexity of 11 – almost all are within recommended limits of 50 lines and 10 cc.
	3. Application specific processing is nicely balanced between XmlNode, XmlDocument, and XmlElementParts.
2. **Reusability:**Three of the 8 packages are reusable, e.g., Tokenizer, MTree, and MNode. No application specific code appears in any of these packages.
3. **Use of Patterns:**Two well-known patterns are used here:
	1. Composite Pattern:
	Represents a tree structure with nodes drawn from an inheritance hierarchy. Derived classes refer to other elements in the tree using the base type, as shown in the class diagram below.
	2. Hook Pattern:
	A reusable package provides a base class for application specific code to derive from. Classes in the reusable part invoke base class functions through pointers that are bound, by a registration or factory process, to derived objects.

	Here, the MTree uses an instance of a class derived from Operation by invoking its operator() on every node it visits during a walk. The application registers its derived instance with the tree’s setOperation() function. The XmlDocument uses classes derived from Operation() for finding elements and for display through node’s ToString() methods.

Hook

Composite

