Types of Models

Prasun Dewan
Department of Computer Science
University of North Carolina at Chapel Hill
dewan@cs.unc.edu
Code available at: https://github.com/pdewan/ColabTeaching
PRE-REQUISITES

- Model-Interactor Separation
Types of models?
**General Pattern**

```
Interactor

Model

Read methods

Write methods
```
**Example**

Interactor

ASimpleList

- size()
- get()

Read methods

Write methods

add()
List Models

Models define methods to read and write indexed elements in variable length lists.

They differ in the set of write methods.
public class ASimpleList<ElementType>
    implements SimpleList<ElementType> {
    List<ElementType> simpleList = new ArrayList<>;
    List<ListObserver<ElementType>> observers = new ArrayList<>;

    public void add(ElementType anElement) {
        simpleList.add(simpleList.size(), anElement);
    }

    public void observableAdd(int anIndex, ElementType anElement) {
        add(anIndex, anElement);
        notifyAdd(anIndex, anElement);
    }

    public void notifyAdd(List<ListObserver<ElementType>> observers, int index, ElementType newValue) {
        for (ListObserver<ElementType> observer : observers)
            observer.elementAdded(index, newValue);
    }

    ...}
**Other Models?**

List Model

- Interactor
- add()
- delete()
- size()
- get()
- delete()

Models define methods to read and write indexed elements in variable length lists.

They differ in the set of write methods.

Other important kinds of models?
Bean Models

Models define getter and setter methods to read fixed number of typed properties.
**Read-only and Editable Properties**

Typed, Named Unit of Exported Object State

```java
public class C {
    public T getP() {
        ...
    }

    public void setP(T newValue) {
        ...
    }
}
```

- **Bean**
- **Name P**
- **Type T**
- **Read-only**
- **Editable**
- **Getter method**
- **Setter method**

**Violates Bean convention:**

For humans and tools
INDEXED BEAN

Bean also defines fixed length indexed collections which we will ignore
MODEL COMPOSITION

Bean Model

List Model

Bean Model

Bean Model
COMPOSING HISTORY MODEL

We already have a model for History
EXAMPLE MODEL COMPOSITION

IM Bean Model

Simple List<String>
- History

Simple List<Character>
- Topic
CONNECTING MODEL/INTERACTOR HIERARCHIES

A model subtree can be connected to a single interactor

A model can be connected to an interactor subtree
SUMMARY OF MODELS

- Lists
  - Variable length indexed lists
  - Differ based on subsets of list operations exposed

- Beans
  - Property collections
  - Differ in properties

- Table model is another important kind not needed in this course

- Model composition
  - Useful when user interfaces are composed

- Model hierarchies can be connected to interactor hierarchies in arbitrary ways