**Motivation**
- Many apps built on frameworks
- Can’t analyze app in isolation
- But frameworks are large, complex, and hard to analyze
- Need model of framework
- Simplify and abstract actual framework
- Problem
  - Mostly written by hand
  - Hard to write and maintain, error-prone

**Synthesis of Framework Models**
- Construct framework model automatically
- Easy to revise/update/improve
- Two key inputs:
  - Samples or tutorials that call key APIs
  - Run these to see framework behavior
- Template of framework structure
  - Specifies design patterns used
- Focus to date:
  - Observer pattern in Java Swing

**Tutorials/Sample Programs**
- Programs that illustrate pieces of the API
- E.g., Swing comes with 79 tutorials
- Executable specification
  - **ButtonDemo.java**
  - **ComboBoxDemo.java**

**Templates**
- Annotated framework APIs
  - Names/types for public classes/methods
  - Hints about design patterns used

```java
@ObserverPattern(ActionEvent)
public class AbstractButton ...
```

“This class participates in observer pattern, triggered by ActionEvent.”

**Sketch – behavioral constraints**

```java
void addActionListener(V.Observable self, V.Observable l) {  
    /* addActionListener has id 19 */  
    int [] params = { 19, self.obj_id, l.obj_id };  
    ObservableCall (19, self, l, null);  
    params = { -19 };  
    ObservableCall (19, self, l, null);  
}
```

```java
assert subcls [observable] [belongsTo[attached]];  
assert subcls [observable] [belongsTo[detached]];  
assert subcls [observable] [belongsTo[handle]];  
```

**Sketch – structural constraints**

```java
int observable = ??; int attach = ??;  …
```

**Sketch Solution**
- ```java
  observable = 11;  ```
- ```java
  attach = 19;  ```

**Executable Model**
- ```java
  public class AbstractButton  
  extends JComboBox implements SwingConstants {  
  LinkedList <ActionListener> _obs;  
  public void addActionListener (ActionListener l) {  
    _obs.add(l);  
  }  
  }  ```

```
// logs translated from the sample
```

```
// logs translated from the sample
```