```
Loading vc-cvs...
package caltool.view_ui;

import caltool.schedule.*;
import mvp.*;
import java.util.*;

/****
 *
 * Class ItemEditor is a dispatching class for the currently selected item.
 * The update and show methods ask the workspace for the currently selected
 * item, then dispatch to the update and show methods of the appropriate
 * item-editor subclasss, namely one of AppointmentEditor, MeetingEditor,
 * TaskEditor, or EventEditor.
 *                                                                          <p>
 * There's no particularly elegant way to avoid the explicit dispatch via
 * dyanic binding, since the dispatch is to view classes, not model classes.
 * In order to use dynamic-binding dispatch from an instance of ScheduledItem,
 * we would have to implement update and show methods in the model classes,
 * which goes against the style of view-ignorant model design we're using.
 *
 */

public class ItemEditor extends View {

    public ItemEditor(ViewUI viewUI) {
        this.viewUI = viewUI;
        editor = null;
    }

    public void update(Observable o, Object arg) {

        if (arg.getClass() == Appointment.class)
            editor = viewUI.getAppointmentEditor();
        else if (arg.getClass() == Meeting.class)
            editor = viewUI.getMeetingEditor();
        else if (arg.getClass() == Task.class)
            editor = viewUI.getTaskEditor();
        else if (arg.getClass() == Event.class)
            editor = viewUI.getTaskEditor();

        editor.update(o, arg);

    }

    public void show() {
        if (editor != null) {
            editor.show();
        }
    }

    View editor;
    ViewUI viewUI;

}
```