JavaBeans: Bound Properties

JavaBeans support bound properties
- A property has get/set methods
- PropertyChangeEvent are sent to registered listeners

All (public) state of the application should be exposed as properties.
Collaboration

Centralized:
- A single copy of the application is shared by broadcasting the display.
- Events are returned to the central copy.
- X Windows uses this model.

Replicated:
- Each user runs a copy of the application.
- Events are broadcast from/to each user.
- A central controller MIGHT be used to coordinate events.
- More natural for Java.
Collaboration with JavaBeans

- Replicated Architecture.
- All relevant state MUST be (bound) properties.
- Any change to the interface generates an event passed to the copies.
- Some coordination needed to ensure all copies process all events in the same order.
Another Design Problem

Problem: Define the functions for a simple component-based statistical analysis system.

Types of components:
- Input components (file readers)
- Filters
- Statistical tools/visualizers

File readers output a defined 2D table format.

Filters do things like:
- select fields (columns)
- select ranges (rows)

Tools/visualizers
- Sampler: Raw Table $\rightarrow$ Raw Table
- Histogram: Frequency
- Boxplot: Descriptive view
- Tableview: Any Table
- Crosstabs: Categorized
- T-test: Descriptive $\times$ Descriptive
- Regression Plot: Raw Table

Design question: What filters should we provide, and how (if at all) do the tools interact?