Swing Data Validation

Karsten Lentzsch
www.JGoodies.com
Learn how to validate data in a Swing application
Speaker Qualifications

- Karsten builds elegant Swing apps
- works with Objects since 1990
- helps others with UI and architectures
- provides libraries that complement Swing
- provides examples for Swing architectures
- writes about Java desktop issues
Agenda

- Introduction
- A Validation Framework
- Who can validate?
- When and how to validate?
- How to display validation results?
- Findings
I - Introduction
Questions

- Who can and who shall validate?
- When shall I validate and present results?
- How can I specify validation rules?
- Where do I display validation results?
- How can I mark invalid data?
- How can I assist in typing valid data?
II – Validation Basics

A general purpose validation framework
Structure

- Someone validates – Validator
- He returns a result – ValidationResult
- That consists of ValidationMessages
- These have a Severity
- ValidationResultModel reports changes
- ValidationComponentUtils marks components
- ValidationResultViewFactory vends views
Validator creates ValidationResultModel.

ValidationResultModel holds ValidationResult.

ValidationResult consists of ValidationResultMessage.

ValidationResultMessage has Severity.
Design Goals

- Works with standard Swing components
- Works with custom Swing components
- Requires no special components
- Requires no special panels
- Integrates well with binding
- Works with different validation styles
- Is extensible
What offers JGoodies Validation?

- Implements the approach mentioned before
- Is a general purpose solution, highly flexible
- Does not require any other library
- Can be used as-is
- Can be used as foundation for higher-level validation frameworks
III – Who?

Who validates what?
3-Tier Client Architecture

- **Presentation**
  - JList
  - JButton
  - JTextField

- **Model Layer**
  - ListModel
  - Action
  - ValueModel

- **Domain Layer**
  - Domain Object
  - DO
  - DO
Domain Object Validates

Presentation

Model layer

Domain layer

JTextField

JButton

JTable

BookModel

ValueModel

Action

ListModel

Validator

aBook

Validation code

:: JGOODIES :: Java User Interface Design
Validator for Domain Object

- JTextField
- JButton
- JTable

- BookModel
- ValueModel
- Action
- ListModel

- aBook
- BookValidator

- Validator

Presented layer

This is my favorite approach
PresentationModel Validates aBook

- JTextField
- JButton
- JTable

BookModel
- ValueModel
- Action
- ListModel

Validation code
implements

Validator

Domain layer
Validator for PresentationModel

- JTextField
- JButton
- JTable

BookModel

- ValueModel
- Action
- ListModel

aBook

BookModel-Validator

implements

Validator

:: JGOODIES :: Java User Interface Design
Presentation Validates

BookPanel

JTextField JTextField

JButton JButton

JTable JTable

Validation code

BookModel

ValueModel ValueModel

Action Action

ListModel ListModel

aBook aBook

implements

Validator

Model layer

Domain layer

:: JGOODIES :: Java User Interface Design
View Filters Invalid Data

- JFormattedTextField
- JTable
- Formatter
- BookModel
- ValueModel
- Action
- ListModel
- aBook

Presentation

Model layer

Domain layer

:: JGOODIES :: Java User Interface Design
IV – When?

When to validate? What initiates validation?
Validation Times/Events

- Key typed
- Focus lost
- Form committed (OK/Apply)
- Save/print/send data

- Find examples in the Validation Demo
Example Application:

JGoodies Validation Demo
Tasks and Approaches for Data Validation

www.JGoodies.com/freeware/validationdemo
Validation cons overlay Swing components
V – How to display?

*How to present validation results and how to mark invalid input?*
Validation Result Views

- Single label with validation text and -icon
- Text area for multiple validation result texts
- JList/JTable for multiple results
- Overlayes, translucent text

- Find examples in the Validation Demo
Validation Markers

- Text field background
- Component border
- Icon overlays component
- Translucent text overlays component

- Find examples in the Validation Demo
VI - Findings

How does the Validation work in projects?
General

- Use Jakarta Commons
- Favor validating unbuffered values
- Favor validators over self-validating objects
- Favor domain validators over model validators
- Avoid presentation validation, but use view filters (JFormattedTextField)
Where does Validation stand?

- Architecture is stable since December 2003.
- No known bugs or problems.
- Comes with a tutorial and demo.
- Only a few predefined validation rules
End

Summary and References
Summary

- We've seen a general validation architecture
- We've learned who can validate data
- We've learned when to validate data
- We've seen approaches to mark invalid data
JGoodies Swing Suite

Example 1  Example 2  Example n  Applications

Convenience Sources  Helper Code

User Interface Framework (UIF)  Framework

Looks  Forms  Binding  Validation  Basic Libraries

JRE / Swing  Foundation

:: JGOODIES :: Java User Interface Design
References I

- JGoodies Validation
  validation.dev.java.net
- JGoodies Articles
  www.JGoodies.com/articles/
- JGoodies Demos
  www.JGoodies.com/freeware/
- JGoodies Binding
  binding.dev.java.net
References II

- Oracle's JClient and ADF
  otn.oracle.com/, search for 'JClient'
- Spring Rich Client Project
  www.springframework.org/spring-rcp.html
- JDNC (Java Desktop Network Components)
  jdnc.dev.java.net
Questions and Answers
End

Good Luck!

Karsten Lentzsch

:: JGOODIES :: Java User Interface Design