JGoodies Karsten Lentzsch

JSR 296 – SWING APP FRAMEWORK

JGoodies

- Elegant Swing applications
- Swing libraries
- Example application sources
- Design assistance
- General Swing consulting

- Expert group member for the JSR 296/295
- Offer alternative 296 implementations

Goal

Learn why & how it started, what it is, how to use it, whether you can use it.

It's easy to program Swing ...

It's easy to program Swing badly.

What's the problem?

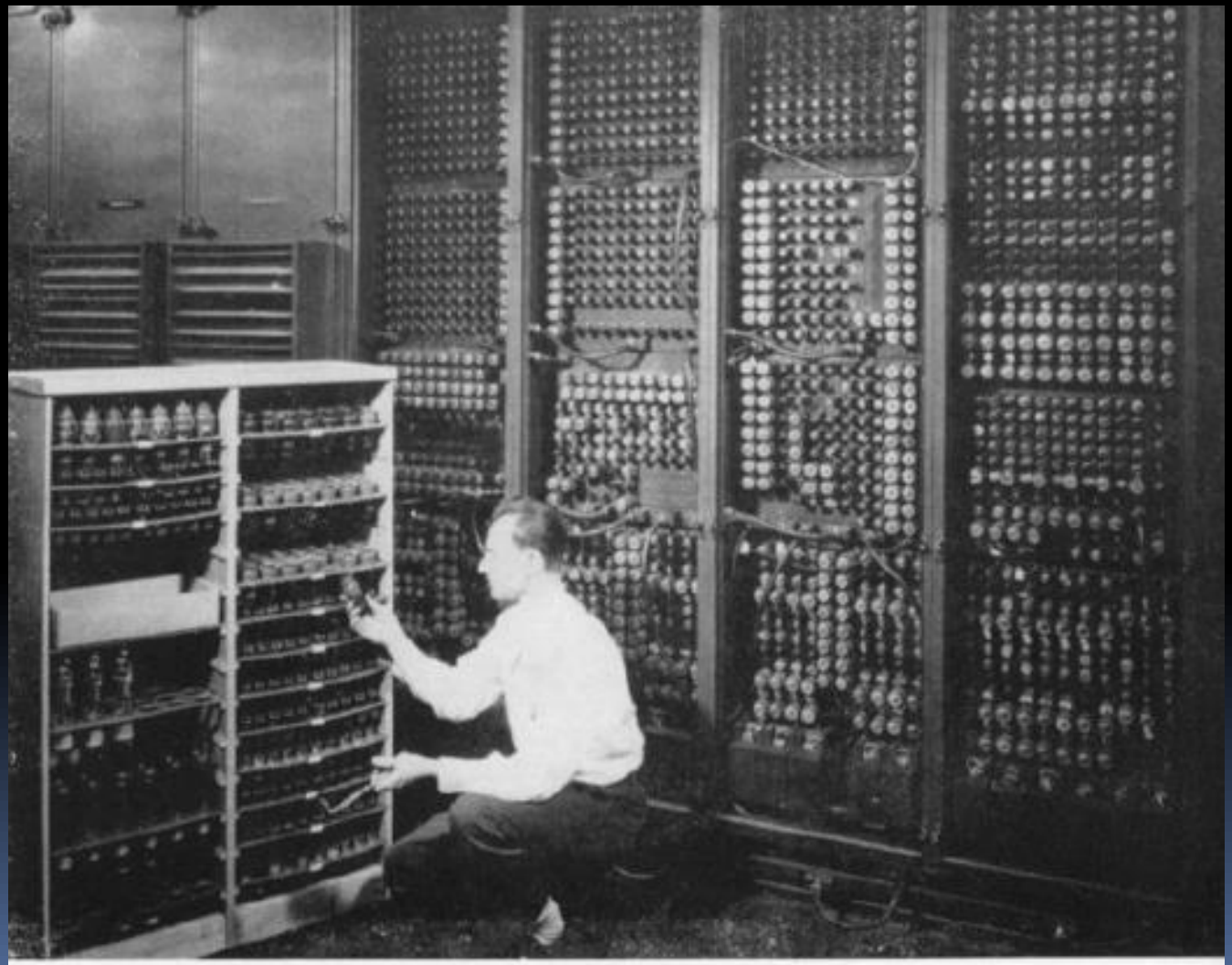
- Swing API is big / High learning curve
- No guidance beyond the toolkit level
- No standard for desktop apps
- Hard to find desktop patterns
- Difficult for beginners and even experts

Laboratory results ->

The Solution

- Reusable, extensible framework for issues common to typical Swing apps
- Public prototype at java.net
- Developed through a JSR
- [Was] intended for Java 7

A Scary Monster?



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

Monsters

- Eclipse RCP
- Netbeans RCP
- Spring RCP / [Spring Desktop]

296: Not Scary

- As small as possible
- Much smaller than Eclipse or Netbeans RCP
- About 20 classes
- Can be explained in less than an hour
- Targets small to medium apps

- No modules, docking, scripting, GUI markup, generic data model, event bus

Draft

- The JSR has not reached the early-draft state.
- Classes, types, methods are work in progress
- Slides focus on features, not implementation

Agenda

Lifecycle

Resources

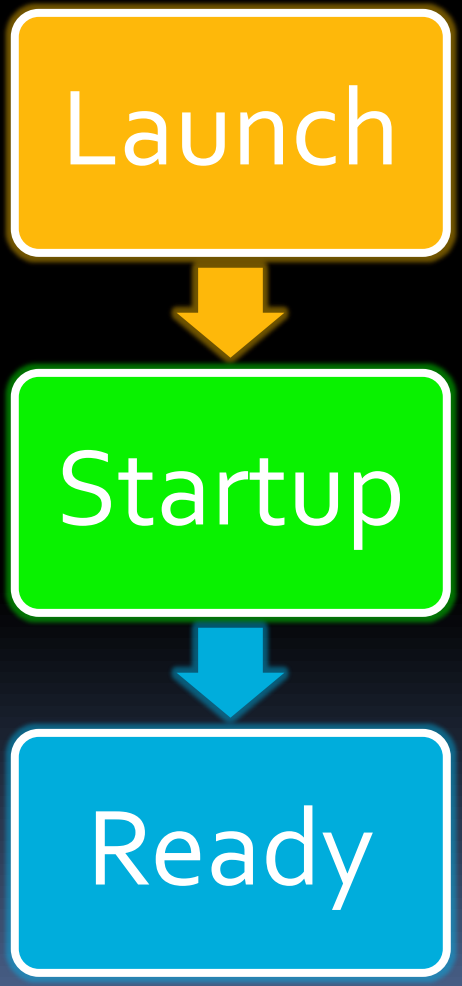
Actions

Tasks

Misc

State of the JSR

Application Lifecycle I



Calls startup() and ready() in the EDT.
Usually invoked from main().

Creates, configures, and shows the GUI.
Mandatory.

Work that must wait until the GUI is visible and ready for input.

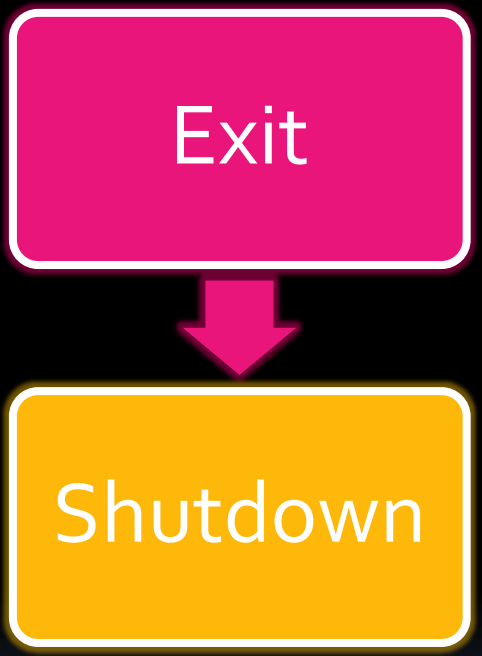
Application Launch

```
public final class Starter {

   public static void main(String[] args) {
      Application.launch(MyApp.class, args);
   }

}
```

Application Start

```
public class MyApp extends Application {

    protected void startup(String[] args) {
        // Create, configure, and show the GUI
    }

    protected void ready() {
        // Load images, fetch data, etc.
    }

}
```

Application Lifecycle II



Calls shutdown(), if the ExitListeners don't veto. Notifies ExitListeners about the exit.

Takes the GUI down. Final cleanup.

Application Exit

```
public void windowClosing(WindowEvent e) {
  Application.getInstance().exit(e);
}

public interface ExitListener {
  // Is the application allowed to exit?
  Promise<Boolean>
      applicationExiting(EventObject e);

  // Do sth. before the app is shut down
  void applicationExited();
}
```

Agenda

Lifecycle

Resources

Actions

Tasks

Misc

State of the JSR

TuneQ: Anlagen > Antrieb HW61, AVV_ZVV

TuneQ | Hilfe | Anlagen | Instandhaltung | Analyse | Wartung

Navigation

- **Strecken**
  - Strecke 1
    - EW 7 / 1445
      - EW 7
        - Antrieb HW61, AVV_ZVV
        - Heizung HWH
      - Steuerung HN-P 7 / 1445
      - Außenanlagen
    - EW 12 / 1449
  - Strecke 2
  - Strecke 3
  - Strecke 4
  - Strecke 5
- **Datensammler**
  - GUW Albertplatz
  - Btf. Gorbitz
  - Btf. Reich (wird angepasst)
  - Btf. Trachenberge
  - GUW Coswig
  - GUW Jahnstraße
  - GUW Meschwitzstrasse
  - GUW Postplatz
- **Anlagen**

Übersicht | Dokumente | Inspektionen | Meldungen | Historie

Antrieb EW 7 - Strecke 1
Friedrichstr. - Maxstr./Könneritzstr.

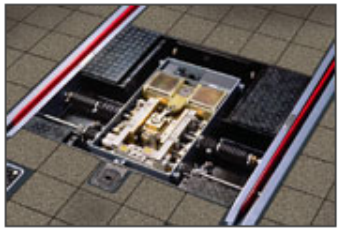
Gerätenr.:	320 513
EDV-Nr.:	109 244 318
Einbau:	12.03.2005
Inspektion:	21.02.2007

Status:	In Betrieb
Montage:	Gleismitte
Ausführung:	Flachbettweiche

Eigenschaften bearbeiten

Produkt

Baureihe:	HW 61	Stellkraft:	5000 N
Antriebsform:	Elektromagnetisch	Verschluß:	Ja
Variante:	AVV-ZVV	Feder:	Ja
Art:	Umstellweiche	Dämpfung:	Ja
Zungenaufschlag:	30 - 70 mm	Richtungsschalter:	nein
Höhe mit Kasten:	300 mm	Zungenprüfer:	Ja
Höhe ohne Kasten:	205 mm	Verschlossen:	Ja
Betriebsspannung:	600/750 V DC	Auffahrbar:	Ja



Produktdetails zeigen

Baugruppen:

Bezeichnung	EDV-Nummer	Gerätenr.	Einbau	Bemerkung
Erdkasten		320 514	12.03.2005	
Doppelzugmagnet	30032002		12.03.2005	
Antriebswelle	31051008		30.11.2006	

Neu... | Bearbeiten... | Löschen | Hoch | Runter

! 3 Ungelesene Nachrichten

ResourceMap

- Defined with ResourceBundles
- Organized in **`resources`** subpackages
- Used to set properties specific to:
  - locale, platform, look&feel, customer

- "Rich" ResourceBundle
  - Converts strings to types
  - Expands variables
  - Adds hierarchy (chain of parents)

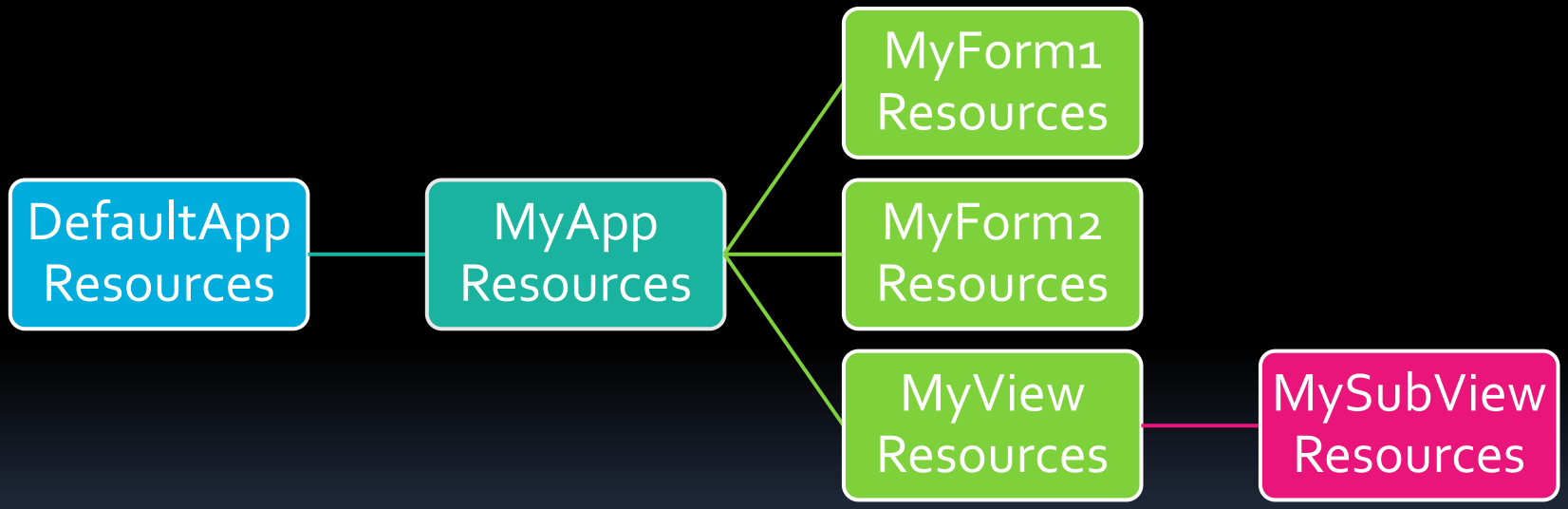
Properties Example

```
search.enabled=true
background.color=#A0A0A0

open.icon=open.png
open.icon=/myapp/resources/open.png

properties.title=%s Properties

editCustomer.title=${edit.title}
```

Using ResourceMap

```
public class MyForm1 {

    static final ResourceMap RESOURCES =
        Application.getInstance().
            getResourceMap(MyForm1.class);
    ...
    RESOURCES.getColor("background.color");
    RESOURCES.getIcon("open.icon");
    RESOURCES.getString("properties.title",
        objectName);
```

ResourceMap Chain



Agenda

Lifecycle

Resources

Actions

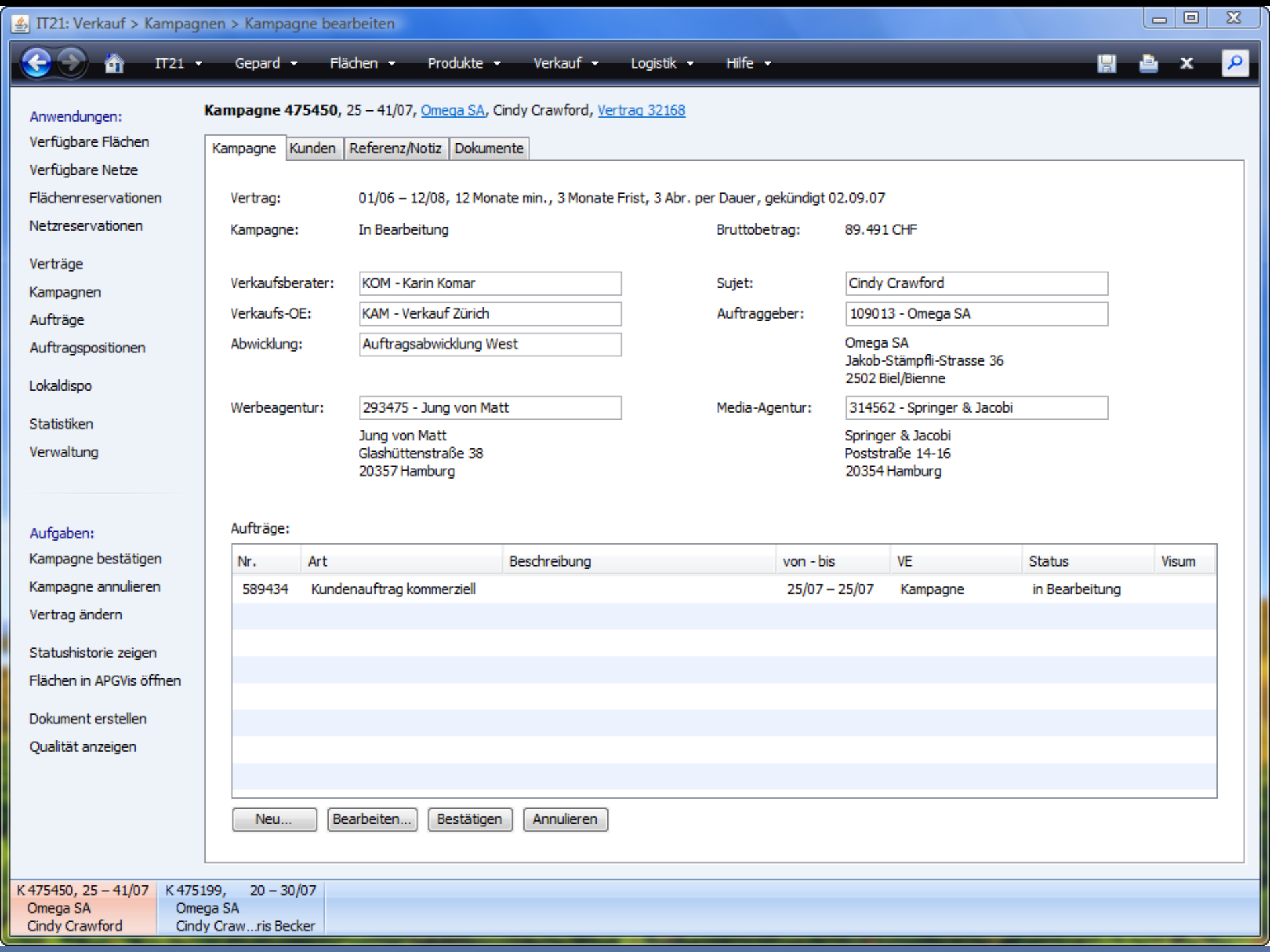
Tasks

Misc

State of the JSR

Swing Actions

- ActionListener plus visual properties:
  - text, shortcut, mnemonic, tooltip, help text
  - enabled state


IT21: Verkauf > Kampagnen > Kampagne bearbeiten
IT21
Gepard
Flächen
Produkte
Verkauf
Logistik
Hilfe
Anwendungen:
Verfügbare Flächen
Verfügbare Netze
Flächenreservationen
Netzreservationen
Verträge
Kampagnen
Aufträge
Auftragspositionen
Lokaldispo
Statistiken
Verwaltung
Aufgaben:
Kampagne bestätigen
Kampagne annulieren
Vertrag ändern
Statushistorie zeigen
Flächen in APGVis öffnen
Dokument erstellen
Qualität anzeigen
Kampagne 475450, 25 – 41/07, Omega SA, Cindy Crawford, Vertrag 32168
Kampagne
Kunden
Referenz/Notiz
Dokumente
Vertrag: 01/06 – 12/08, 12 Monate min., 3 Monate Frist, 3 Abr. per Dauer, gekündigt 02.09.07
Kampagne: In Bearbeitung
Bruttobetrag: 89.491 CHF
Verkaufsberater: KOM - Karin Komar
Verkaufs-OE: KAM - Verkauf Zürich
Abwicklung: Auftragsabwicklung West
Sujet: Cindy Crawford
Auftraggeber: 109013 - Omega SA
Omega SA
Jakob-Stämpfli-Strasse 36
2502 Biel/Bienne
Werbeagentur: 293475 - Jung von Matt
Jung von Matt
Glashüttenstraße 38
20357 Hamburg
Media-Agentur: 314562 - Springer & Jacobi
Springer & Jacobi
Poststraße 14-16
20354 Hamburg
Aufträge:
Nr.
Art
Beschreibung
von - bis
VE
Status
Visum
589434
Kundenauftrag kommerziell
25/07 – 25/07
Kampagne
in Bearbeitung
Neu...
Bearbeiten...
Bestätigen
Annulieren
K 475450, 25 – 41/07
Omega SA
Cindy Crawford
K 475199, 20 – 30/07
Omega SA
Cindy Craw...ris Becker

Old Style Action Definition

```
Action action = new AbstractAction("New…"){
  public void actionPerformed(ActionEvt e){
    // perform the new operation here
  }
};

aTextField.setAction(action);
aButton.setAction(action);
```

Old Style Action Definition

```
public class MyModel {

  private Action newAction;

  public Action getNewAction() {
    if (newAction == null) {
       newAction = new AbstractAction("New…") {
         public void actionPerformed(ActionEvent e){
           // perform the new operation here
         }
       };
       newAction.putValue(Action.MNEMONIC, …);
       newAction.putValue(Action.SHORTCUT, …);
    }
    return newAction;
  }
}
```

Old Style Action Definition

```
public class MyModel {

  private Action newAction;

  public Action getNewAction() {
    if (newAction == null) {
       newAction = new AbstractAction("New…") {
         public void actionPerformed(ActionEvent e){
           // perform the new operation here
         }
       };
       newAction.putValue(Action.MNEMONIC, …);
       newAction.putValue(Action.SHORTCUT, …);
    }
    return newAction;
  }
}
```

Swing Actions

- Creating Action objects is inefficient
- Text, mnemonic, shortcut, etc. should be internationalized and may vary with the platform
- Asynchronous Actions are difficult

- Many inner Action classes
- Dispatching Action classes (one class for many Actions) help a bit

New Action Definition

```
public class MyModel {

  @Action
  public void onNewPerformed(ActionEvent evt) {
      // perform the new operation here
  }


  @Action(enabled=false)
  public void onEditPerformed(ActionEvent evt) {
      // perform the edit operation here
  }
```

Action Properties

```
newItem.Action.text=&New…
newItem.Action.accelerator=Ctrl N
newItem.Action.shortDescription=New item
newItem.Action.icon=images/new.png
```

@Action with direct resources

```
public class MyModel {

 @Action(text="_New…", accelerator="CTRL N")
 public void onNewPerformed(ActionEvent evt) {
     // perform the new operation here
 }

…

}
```

Using Actions (1/2)

```
public class MyView {

   private MyModel model;
   ...

   ActionMap map =
         Application.createActionMap(model);
   Action action = map.get("newItem");
   JButton button = new JButton(action);
```

Using Actions (2/2)

- JGoodies convenience types:
  - IActionObject mit #getAction(String actionName)
- Implemented by:
  - ActionObject
  - ActionBean
  - ActionPresentationModel

Agenda

Lifecycle

Resources

Actions

Tasks

Misc

State of the JSR

Don't Block the EDT!

- Use background threads for:
  - operations that might block, e.g. file or network IO
  - computationally intensive operations
- Approaches
  - SwingWorker
  - Spin
  - Foxtrot
- We also want:
  - progress and messages
  - convenient definition
  - dependencies between background tasks

Task and BlockingScope

- Task inherits the SwingWorker features
- Adds progress convenience ops
- Messages
- Configured from ResourceMap
- Safe exit behavior
- Blocks: nothing, Action, component, window, application

Task Definition

```
public class SaveTask extends Task {

    public SaveTask() {
        super(BlockingScope.APPLICATION,
              SaveTask.class);
    }

    protected Object doInBackground() {
        setMessage("A message");
        setProgress(30);
    }
    protected void succeeded() { … }
```

Using Tasks 1/2

```
public class MyModel {

   @Action // External resources
   public Task save(ActionEvent e) {
      if (!valid()) {
        // Show notifier immediately
        return null;
      }
      return new SaveTask();
   }
```

Using Tasks 2/2

```
public class MyModel {

   @Action(text="_Save", enabled=false)
   public Task save(ActionEvent e) {
      if (!valid()) {
        // Show notifier immediately
        return null;
      }
      return new SaveTask();
   }
```

TaskService, TaskMonitor

- TaskService defines how a Task is executed
  - serially
  - by a thread pool
  - etc.
- TaskMonitor
  - provides a summary for multiple Tasks
  - bound properties for a *foreground* Task
  - simplifies status bar implementations

Agenda

Lifecycle

Resources

Actions

Tasks

Misc

State of the JSR

Resource Injection

Set properties from like-named resources

```
resrcMap.injectComponents(aComponent)
myPanel.setBackground(Color c)
myLabel.setIcon(Icon i)
```

Set marked fields from like-named resources

```
resrcMap.injectFields(anObject)
@Resource Color foreground;
@Resource Icon icon;
```

Resource Injection II

- Pros:
  - localizable by default
  - easy to change visual properties
  - visual properties can be edited by non-developers
  - visual properties can change at runtime
- Cons:
  - No compile-time safety
  - Multiple sources
  - Almost no IDE support

Persistent Application State

- An app should store some app state:
  - window positions
  - table column widths
  - split bar positions
  - etc.

- The JSR 296 aims to do this automatically
- See also the UIState library

SessionStorage, LocalStorage

- SessionStorage
  - save(rootComponent, filename)
  - restore(rootComponent, filename)
- LocalStorage
  - abstracts per-user files
  - works for unsigned apps too
- Preferences?
  - already in the Java core
  - limited in data size

Resource Variants Proposal

```
prefs.Action.text=${prefs.Action.text.[$os]}
prefs.Action.text.default=Preferences
prefs.Action.text.win=Optio&ns
prefs.Action.accelerator=
                    ${prefs.Action.accelerator.[$os]}
prefs.Action.accelerator.default=${null}
prefs.Action.accelerator.mac=meta COMMA
```

Agenda

Lifecycle

Resources

Actions

Tasks

Misc

State of the JSR

State of the JSR

- Inactive
- Spec lead and EG failed to provide a milestone draft in more than 18 month

- Otherwise: Zombie

Brief History: Before 2006

- Desktop Blueprints discussions
- Lack of desktop patterns
- Almost no Sun folks for the app-level
- Background tasks:
  - Old unsupported SwingWorker
  - Spin
  - Foxtrot (synchronous)

2006

- Project started
- Project and spec lead: Hans Muller
- JSR submitted by Sun
- EG formed
- EG discussions about the feature set

- November: Major breakthrough for Swing

2007

- Initial public pre-draft prototype
- Removed nonsense
- Feb – Aug: Versions 0.1 – 0.4
- September: Version 1.0
- November: stuck

2008

- Jan – May: stuck
- May: Hans Muller left sun
  - See "Hans's swan song"
- July: New spec lead Alexander Potochkin
- Aug: Beta versions
- Sep: stuck

2009

- March: Spec lead back again
- Some updates without EG discussion

Discussions

- EG almost dead
- Many messages in appframework mailing list

- API is work in progress, almost not discussed

appframework Implementation

- Showstoppers require API changes
- Several problems not even identified
- API may change dramatically

- Not ready for production

Alternative Implementations

- Commercial public JGoodies code
  - Lifecycle, Resources, Actions, Safe SwingWorker
  - Preferences
  - Simple local storage
- Commercial non-public JGoodies code
  - Adds Tasks, Blocking
  - No Resource Injection
- Your framework moved towards the JSR 296

Summary

- JSR scope meets what people need
- Some features are pretty stable:
  - Lifecycle, Resources, Actions
- Implementation[s] still buggy

- However: A key success factor!
- You can benefit from this JSR

References

- Google "JSR 296"
- appframework.dev.java.net
- appframework user mailing list

- www.jgoodies.com/articles

A Swing Survivor's Guide

Desktop Patterns

Data Binding

JSR 296

First Aid for Swing

Layout Management

Meta Design

How to structure an app?

- Scott Delap: *Desktop Java Live* (slightly outdated)
- JGoodies: *Desktop Patterns & Data Binding*

QUESTIONS AND ANSWERS

JGoodies Karsten Lentzsch

JSR 296 – SWING APP FRAMEWORK