TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Elektrotechnik und Informationstechnik Institut für Automatisierungstechnik

Integration der domänenspezifische Sprache Movisa in den nutzerzentrierten Entwicklungsprozess der Useware

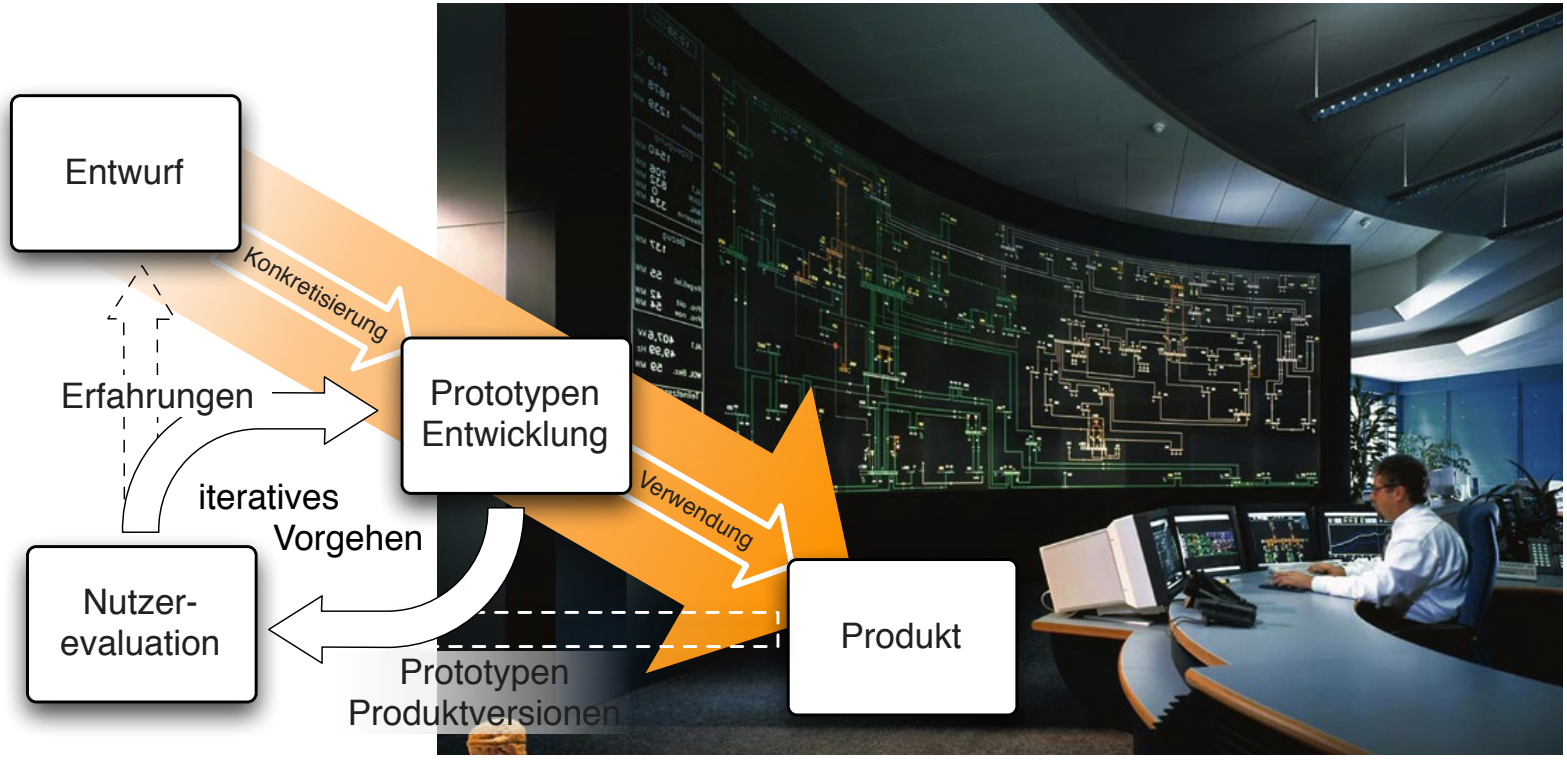
Workshop: Modellbasierte Entwicklung von Benutzungsschnittstellen

H. Hager, St. Hennig, M. Seißler, A. Braune

Dresden, 06.10.2011

Gebrauchstaugliche UIs

Nutzerzentrierte Entwicklung für die Automatisierungstechnik



Agenda

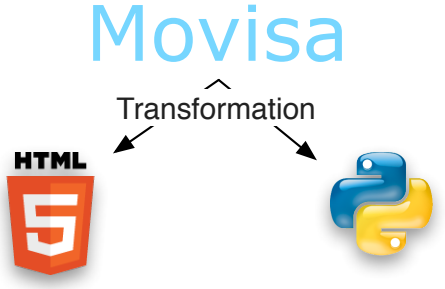
① **Einführung**: Movisa + Useware-Entwicklungsprozess
② **Problemstellung:** Herausforderungen bei der Transformation
③ **Lösungsvorschlag**: Interaktive Transformation mit Petmap
④ **Auswertung**

01 Einführung

Integration von Movisa in den Useware-Entwicklungsprozess

Movisa

- Domänenspezifische Sprache für die AT
- Plattformunabhängige Beschreibung

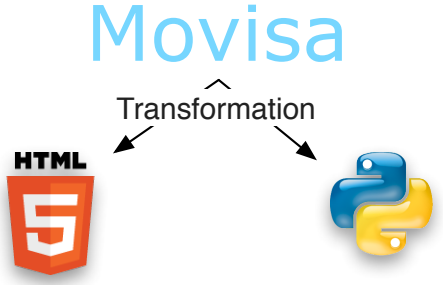


01 Einführung

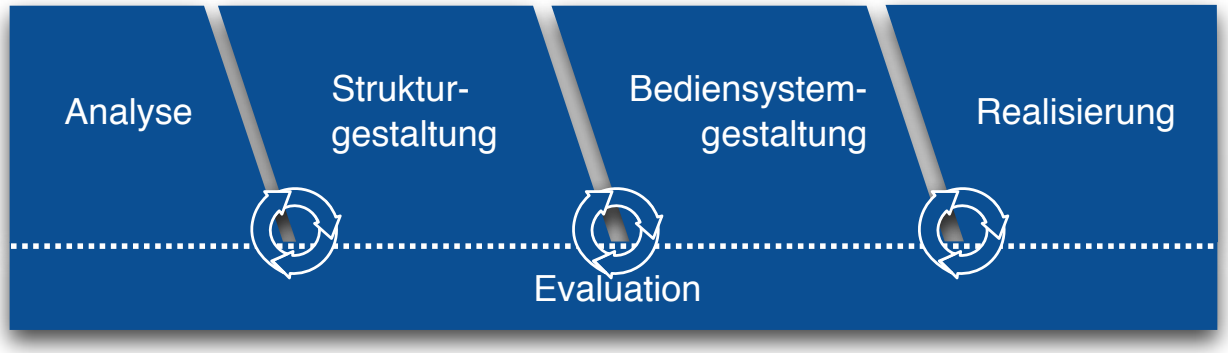
Integration von Movisa in den Useware-Entwicklungsprozess

Movisa

- Domänenspezifische Sprache für die AT
- Plattformunabhängige Beschreibung



Useware-Entwicklungsprozess

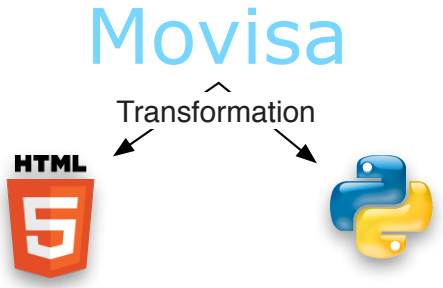


01 Einführung

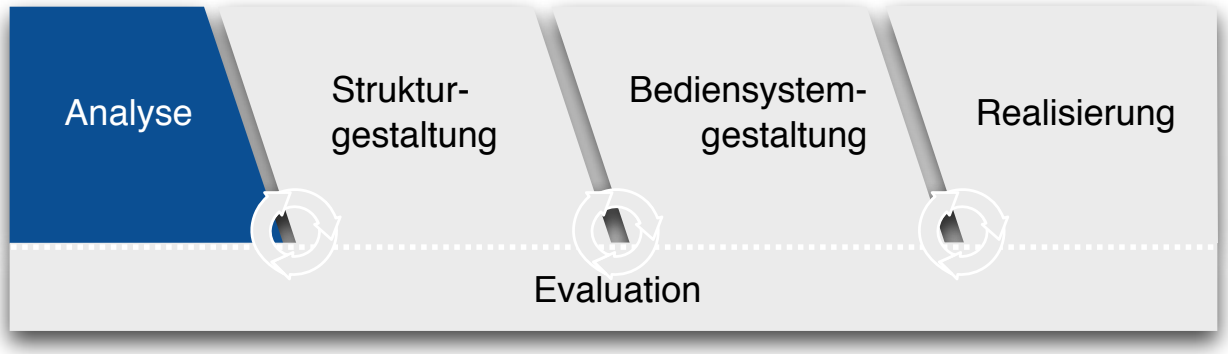
Integration von Movisa in den Useware-Entwicklungsprozess

Movisa

- Domänenspezifische Sprache für die AT
- Plattformunabhängige Beschreibung



Useware-Entwicklungsprozess



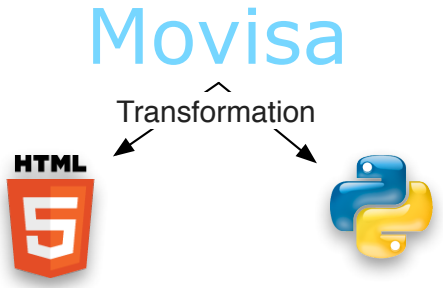
- Datenerhebung
- Datenaufbereitung
- (UseML)

01 Einführung

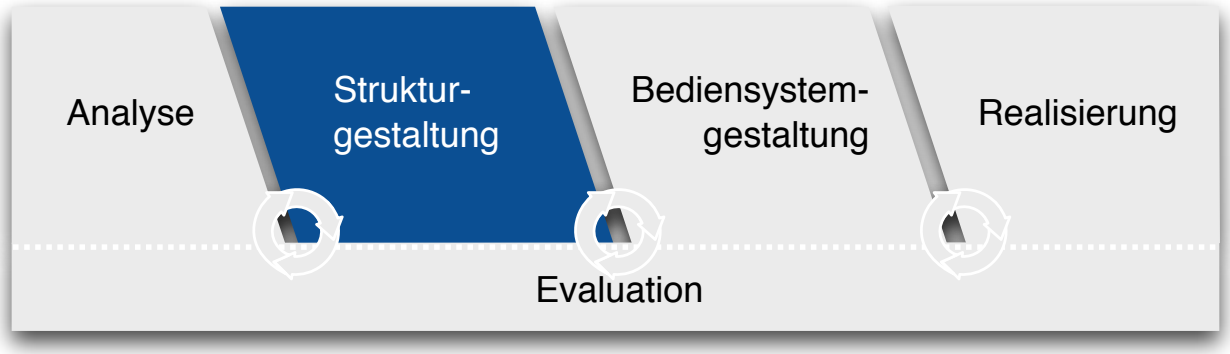
Integration von Movisa in den Useware-Entwicklungsprozess

Movisa

- Domänenspezifische Sprache für die AT
- Plattformunabhängige Beschreibung



Useware-Entwicklungsprozess



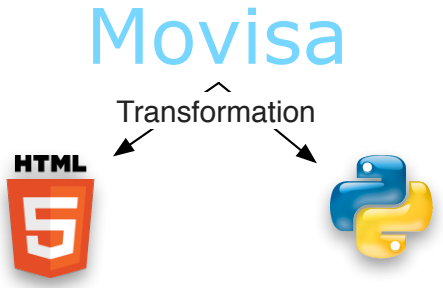
- Datenerhebung
- Datenaufbereitung
- (UseML)
- Aufgabenstruktur
- Benutzerinteraktionen
- (DISL)

01 Einführung

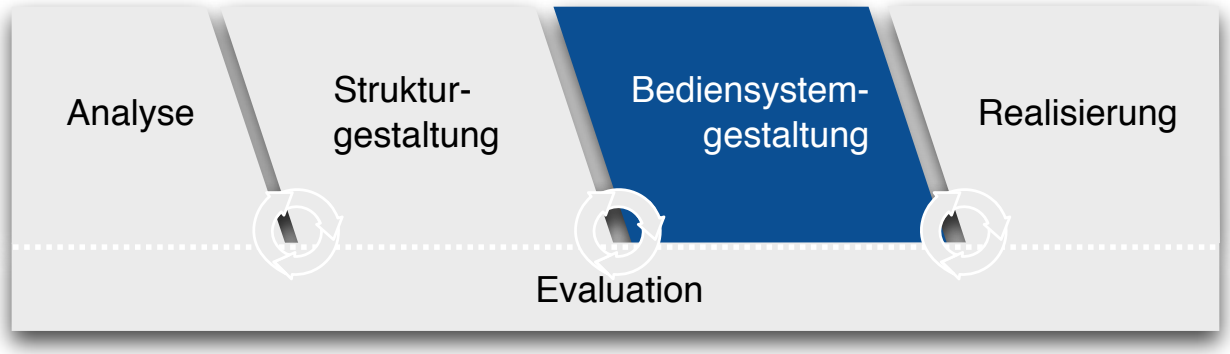
Integration von Movisa in den Useware-Entwicklungsprozess

Movisa

- Domänenspezifische Sprache für die AT
- Plattformunabhängige Beschreibung



Useware-Entwicklungsprozess



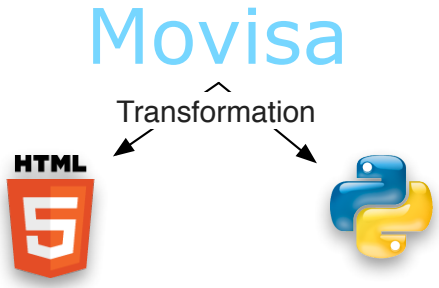
• Datenerhebung	• Aufgabenstruktur	• Layout
• Datenaufbereitung	• Benutzerinteraktionen	• Modalitätenwahl
• (UseML)	• (DISL)	• (UIML,VoiceXML)

01 Einführung

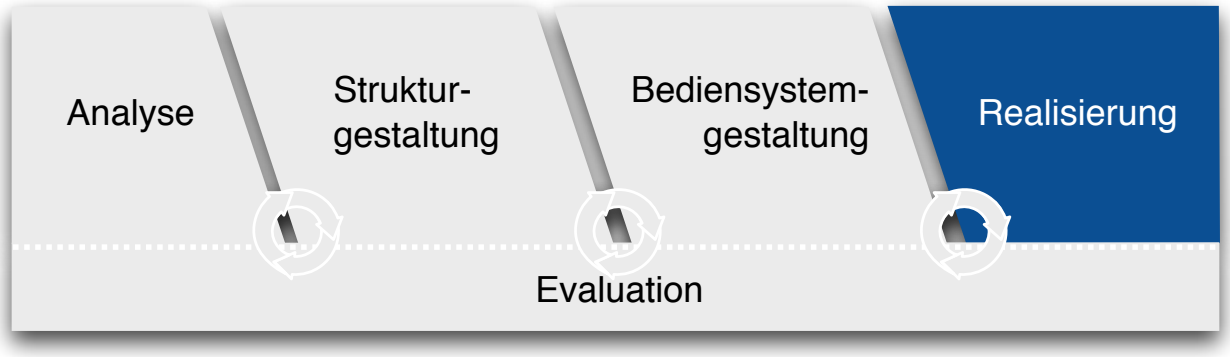
Integration von Movisa in den Useware-Entwicklungsprozess

Movisa

- Domänenspezifische Sprache für die AT
- Plattformunabhängige Beschreibung



Useware-Entwicklungsprozess



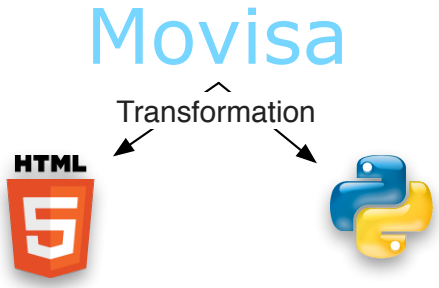
• Datenerhebung	• Aufgabenstruktur	• Layout	• Feingestaltung
• Datenaufbereitung	• Benutzerinteraktionen	• Modalitätenwahl	• (XAML)
• (UseML)	• (DISL)	• (UIML, VoiceXML)	

01 Einführung

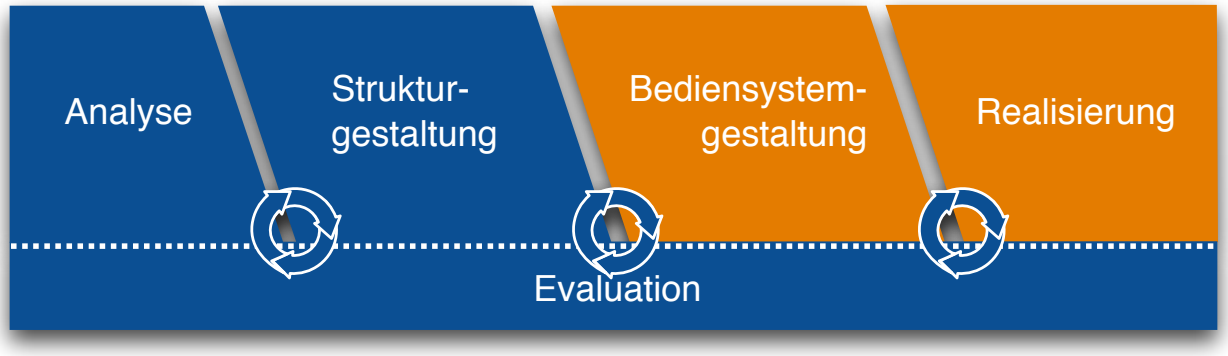
Integration von Movisa in den Useware-Entwicklungsprozess

Movisa

- Domänenspezifische Sprache für die AT
- Plattformunabhängige Beschreibung



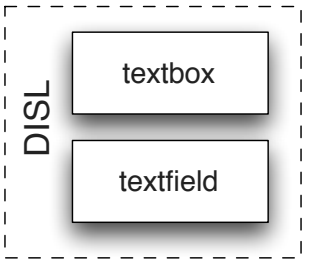
Useware-Entwicklungsprozess

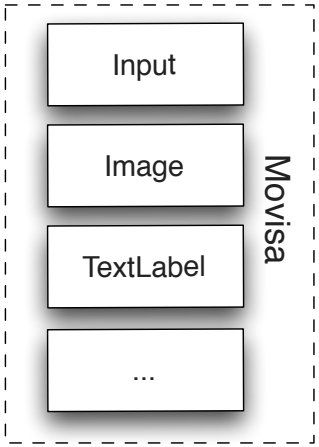


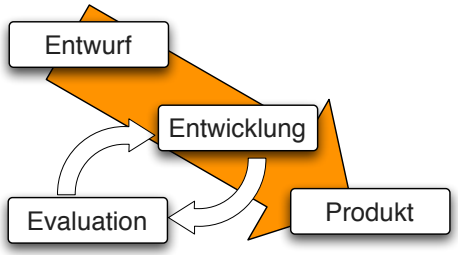
• Datenerhebung	• Aufgabenstruktur	• Layout	• Feingestaltung
• Datenaufbereitung	• Benutzerinteraktionen	• AT Spezifika	• (HTML5,Python)
• (UseML)	• (DISL)	• (Movisa)	

02 Problemstellung

Herausforderungen bei der Transformation





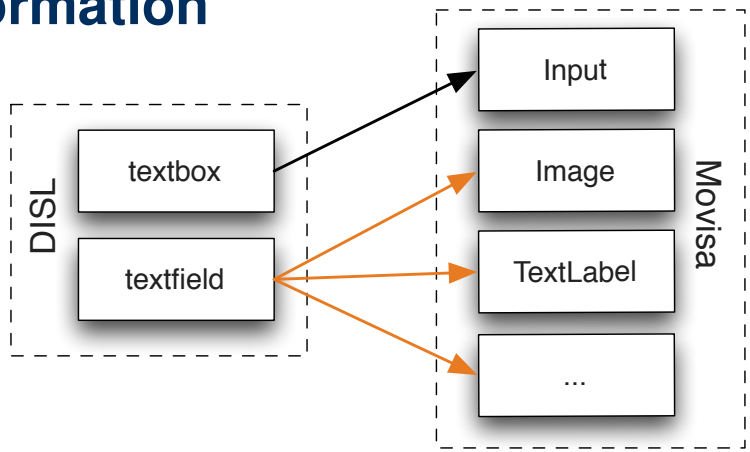


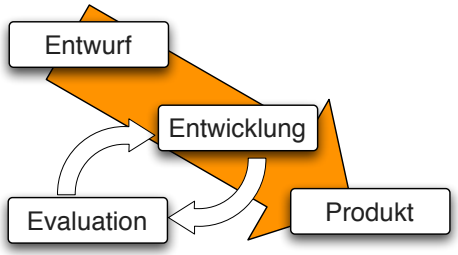
02 Problemstellung

Herausforderungen bei der Transformation

❶ **Mehrdeutige Elementzuordnungen**

- 6 DISL Interaktionsobjekte
- 12 Movisa Interaktionsobjekte





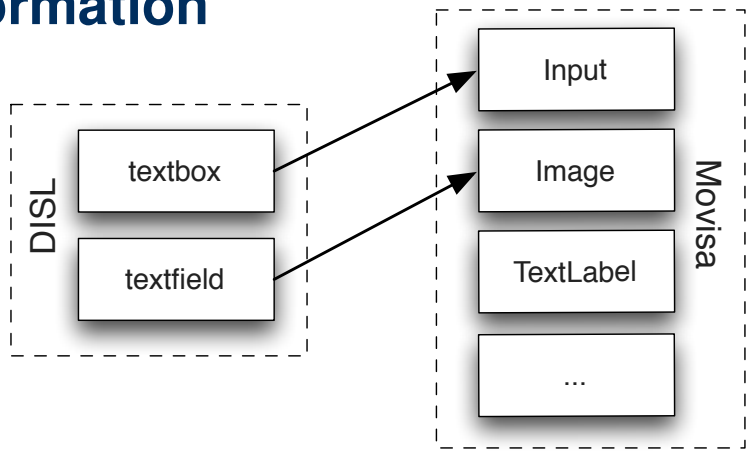
02 Problemstellung

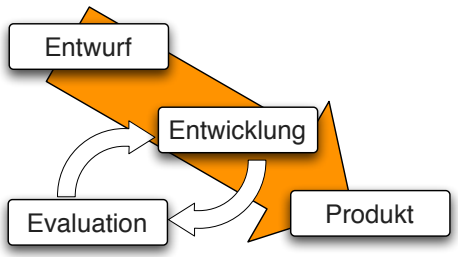
Herausforderungen bei der Transformation

❶ **Mehrdeutige Elementzuordnungen**

- 6 DISL Interaktionsobjekte
- 12 Movisa Interaktionsobjekte

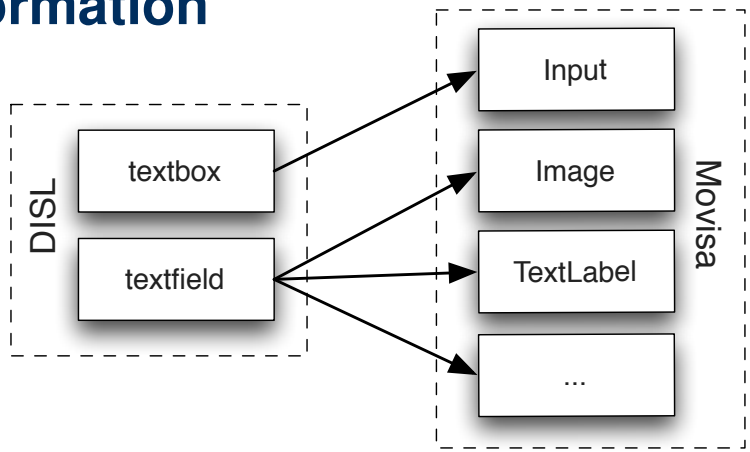
✘ Standardzuordnungen





02 Problemstellung

Herausforderungen bei der Transformation

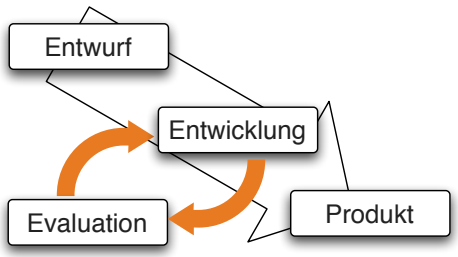


❶ **Mehrdeutige Elementzuordnungen**

- 6 DISL Interaktionsobjekte
- 12 Movisa Interaktionsobjekte
- ✘ Standardzuordnungen

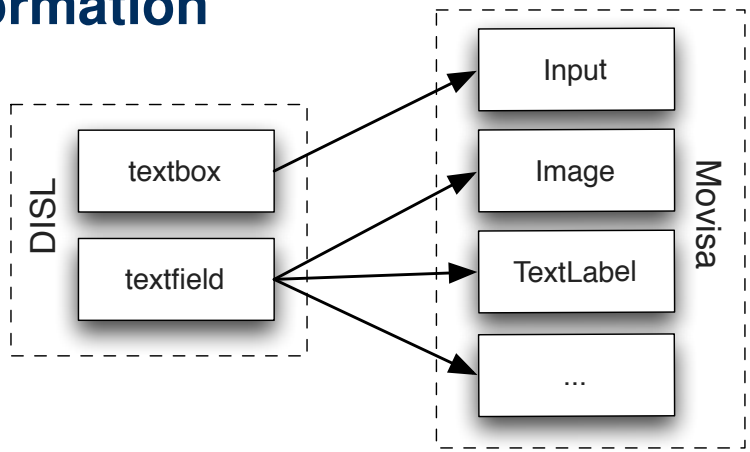
❷ **Iterative Entwicklung** der Useware

- Veränderung/Anpassung aller Modelle zu jeder Zeit
- Konsistenz der Modelle bzw. Transformation beachten



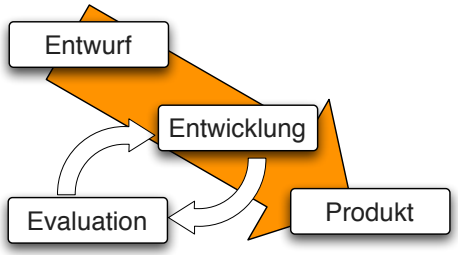
02 Problemstellung

Herausforderungen bei der Transformation



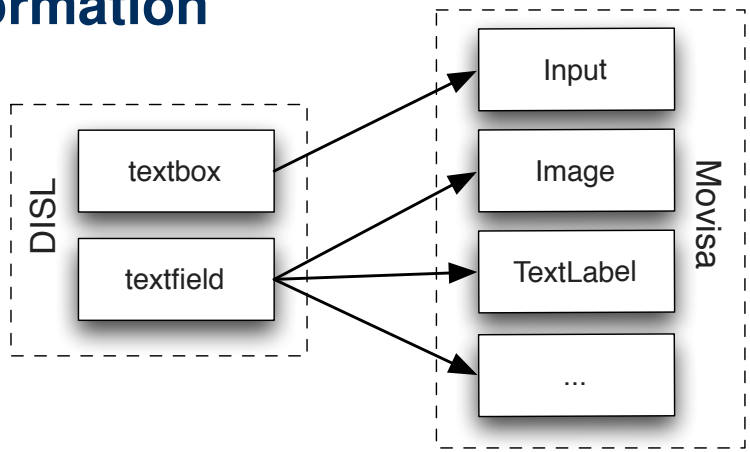
❶ **Mehrdeutige Elementzuordnungen**
- 6 DISL Interaktionsobjekte
- 12 Movisa Interaktionsobjekte
- ✘ Standardzuordnungen
- → Interaktive Transformation

❷ **Iterative Entwicklung** der Useware
- Veränderung/Anpassung aller Modelle zu jeder Zeit
- Konsistenz der Modelle bzw. Transformation beachten



02 Problemstellung

Herausforderungen bei der Transformation



❶ **Mehrdeutige Elementzuordnungen**

- 6 DISL Interaktionsobjekte
- 12 Movisa Interaktionsobjekte

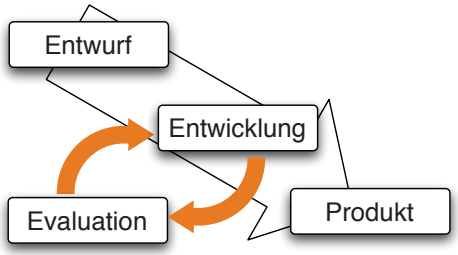
✘ Standardzuordnungen

→ Interaktive Transformation

❷ **Iterative Entwicklung** der Useware

- Veränderung/Anpassung aller Modelle zu jeder Zeit
- Konsistenz der Modelle bzw. Transformation beachten

→ Persistant Transformation Mapping



03 Lösungsvorschlag

Interaktive Transformation mit Petmap

Interaktive Transformation

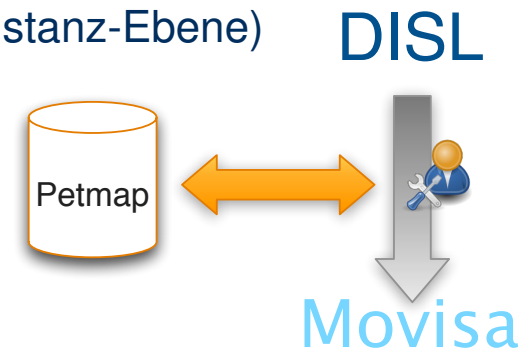
- Der Entwickler wählt das konkrete Element aus (Instanz-Ebene)
- Vorauswahl der Optionen

03 Lösungsvorschlag

Interaktive Transformation mit Petmap

Interaktive Transformation

- Der Entwickler wählt das konkrete Element aus (Instanz-Ebene)
- Vorauswahl der Optionen



Persistant Transformation Mapping (Petmap)

- Speicherung der ausgewählten Zuordnungen (Trace-Modell)
- Wiederverwendung bei erneuter Transformation (Mapping-Modell)

03 Lösungsvorschlag

Anwendung der Transformation

DISL Modell

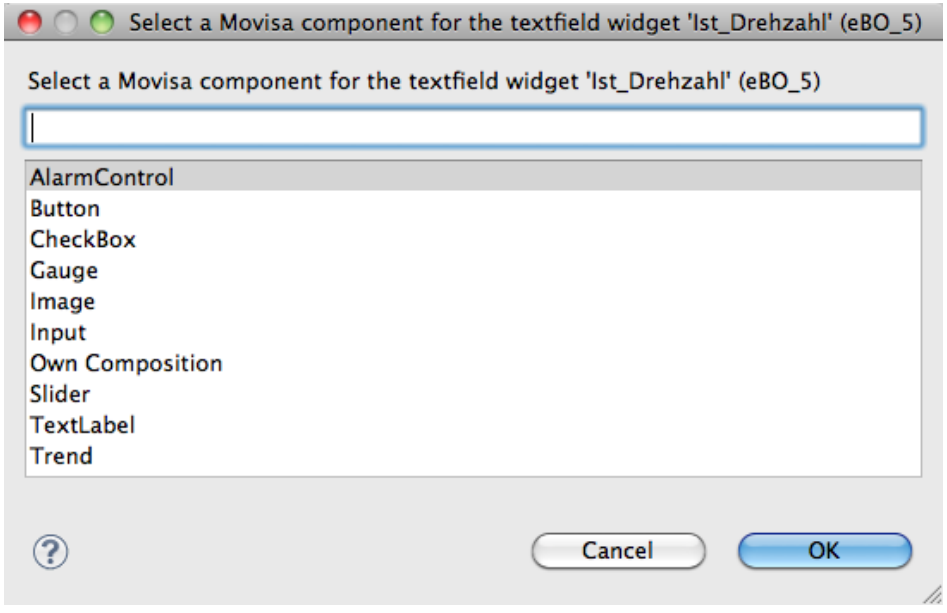
```
<!--  -->
<interface id="View" state="start">
    <structure>
      <widget generic-widget="textfield" id="fuellevel"/>
      <widget generic-widget="command" id="pump_start"/>
      <widget generic-widget="textfield" id="pump_speed"/>
    </structure>
  <style>    <!-- ... --> </style>
  <behavior> <!-- ... --> </behavior>
</interface>
<!--  -->
```

03 Lösungsvorschlag

Anwendung der Transformation

DISL Modell

```
<!--  -->
<interface id="View" state="start">
    <structure>
      <widget generic-widget="textfield" id="fuellevel"/>
      <widget generic-widget="command" id="pump_start"/>
      <widget generic-widget="textfield" id="pump_speed"/>
    </structure>
  <style>    <!-- ... --> </style>
  <behavior> <!-- ... --> </behavior>
</interface>
<!--  -->
```



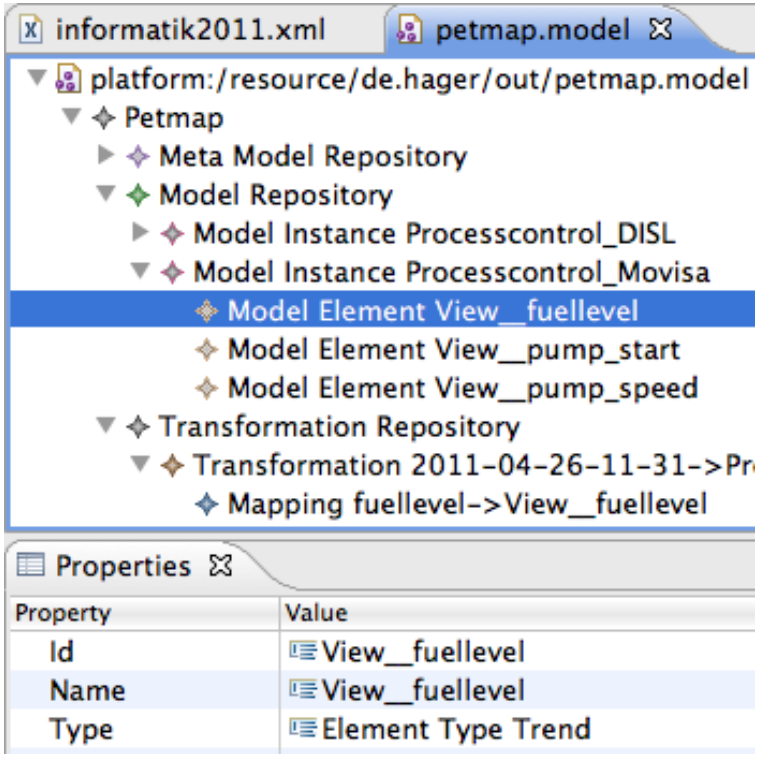
- Drei Interaktionen notwendig

03 Lösungsvorschlag

Anwendung der Transformation

DISL Modell

```
<!--  -->
<interface id="View" state="start">
    <structure>
      <widget generic-widget="textfield" id="fuellevel"/>
      <widget generic-widget="command" id="pump_start"/>
      <widget generic-widget="textfield" id="pump_speed"/>
    </structure>
  <style>    <!-- ... --> </style>
  <behavior> <!-- ... --> </behavior>
</interface>
<!--  -->
```



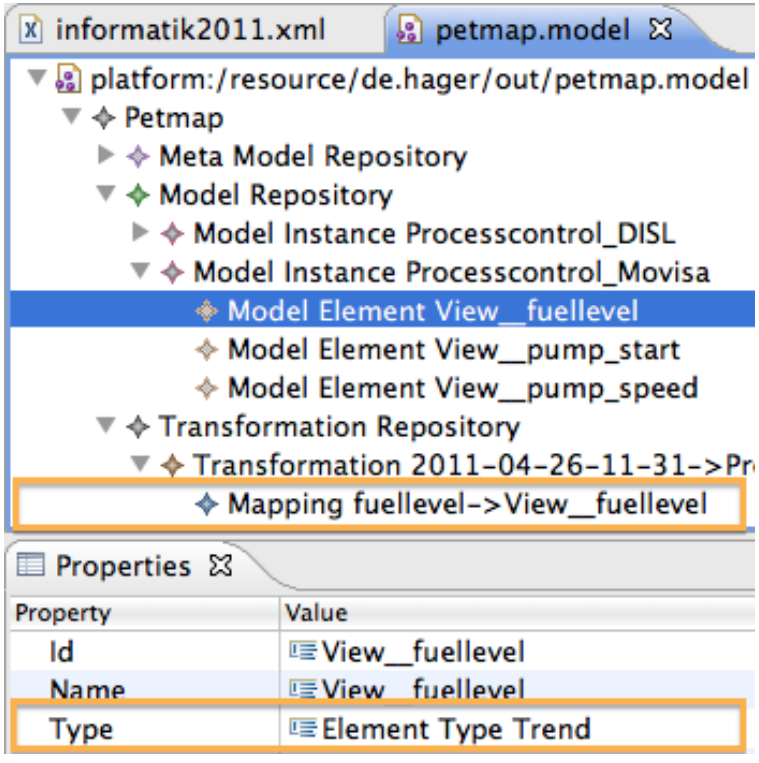
- Drei Interaktionen notwendig

03 Lösungsvorschlag

Anwendung der Transformation

DISL Modell

```
<!--  -->
<interface id="View" state="start">
    <structure>
      <widget generic-widget="textfield" id="fuellevel"/>
      <widget generic-widget="command" id="pump_start"/>
      <widget generic-widget="textfield" id="pump_speed"/>
    </structure>
  <style>    <!-- ... --> </style>
  <behavior> <!-- ... --> </behavior>
</interface>
<!--  -->
```



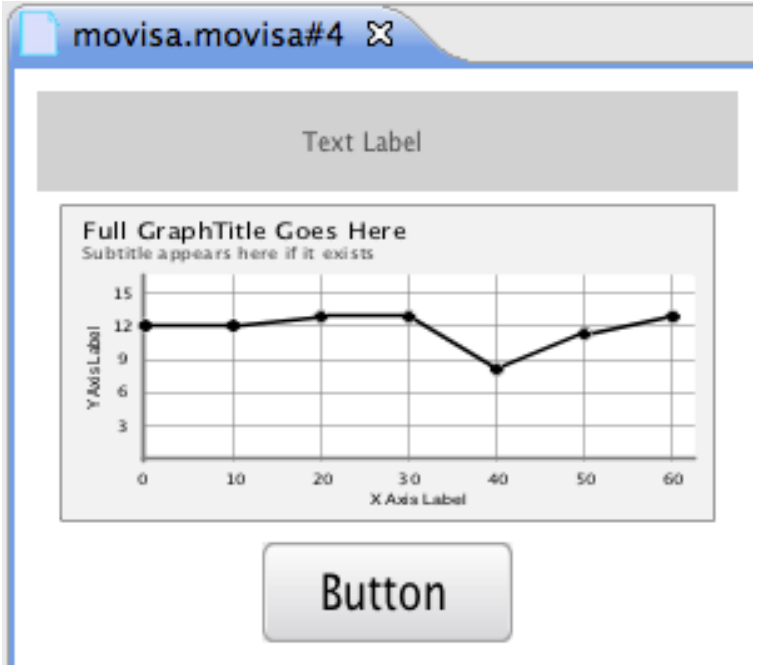
- Drei Interaktionen notwendig

03 Lösungsvorschlag

Anwendung der Transformation

DISL Modell

```
<!--  -->
<interface id="View" state="start">
    <structure>
      <widget generic-widget="textfield" id="fuellevel"/>
      <widget generic-widget="command" id="pump_start"/>
      <widget generic-widget="textfield" id="pump_speed"/>
    </structure>
  <style>    <!-- ... --> </style>
  <behavior> <!-- ... --> </behavior>
</interface>
<!--  -->
```



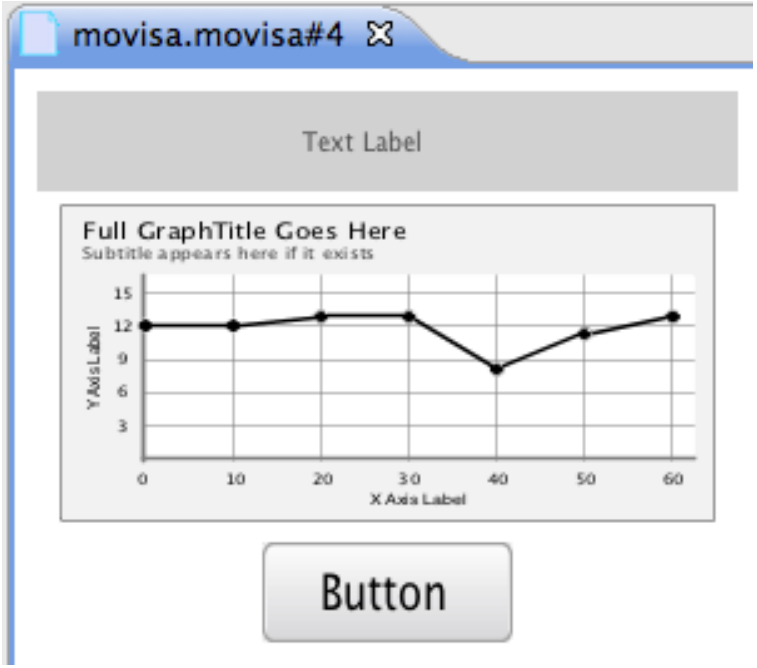
- Drei Interaktionen notwendig
- Beschriftungen, Interaktionseigenschaften generiert

03 Lösungsvorschlag

Anwendung der Transformation

DISL Modell

```
<!--  -->
<interface id="View" state="start">
    <structure>
      <widget generic-widget="textfield" id="fuellevel"/>
      <widget generic-widget="command" id="pump_start"/>
      <widget generic-widget="command" id="pump_stop"/>
      <widget generic-widget="textfield" id="pump_speed"/>
    </structure>
  <style>    <!-- ... --> </style>
  <behavior> <!-- ... --> </behavior>
</interface>
<!--  -->
```



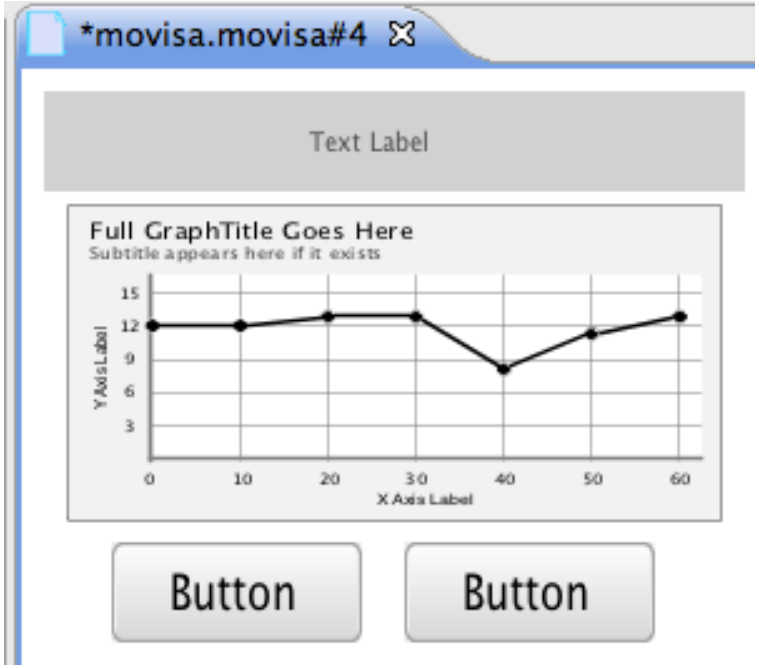
- Drei Interaktionen notwendig
- Beschriftungen, Interaktionseigenschaften generiert
- Anordnung der Elemente erfolgt manuell

03 Lösungsvorschlag

Anwendung der Transformation

DISL Modell

```
<!--  -->
<interface id="View" state="start">
    <structure>
      <widget generic-widget="textfield" id="fuellevel"/>
      <widget generic-widget="command" id="pump_start"/>
      <widget generic-widget="command" id="pump_stop"/>
      <widget generic-widget="textfield" id="pump_speed"/>
    </structure>
  <style>    <!-- ... --> </style>
  <behavior> <!-- ... --> </behavior>
</interface>
<!--  -->
```



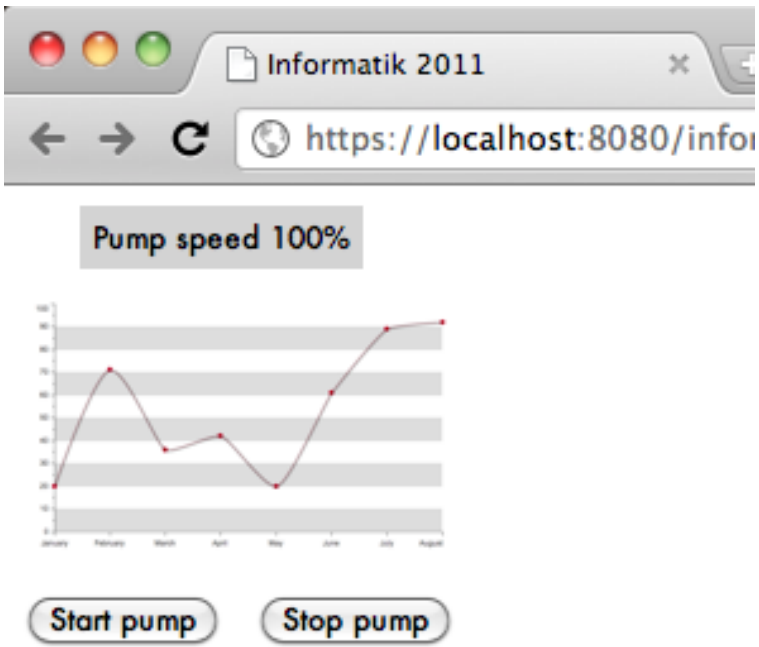
- Drei Interaktionen notwendig
- Beschriftungen, Interaktionseigenschaften generiert
- Anordnung der Elemente erfolgt manuell
- Erneute Transformation: Benutzerinteraktion nur für neue Elemente

03 Lösungsvorschlag

Anwendung der Transformation

DISL Modell

```
<!--  -->
<interface id="View" state="start">
    <structure>
      <widget generic-widget="textfield" id="fuellevel"/>
      <widget generic-widget="command" id="pump_start"/>
      <widget generic-widget="command" id="pump_stop"/>
      <widget generic-widget="textfield" id="pump_speed"/>
    </structure>
  <style>    <!-- ... --> </style>
  <behavior> <!-- ... --> </behavior>
</interface>
<!--  -->
```



- Drei Interaktionen notwendig
- Beschriftungen, Interaktionseigenschaften generiert
- Anordnung der Elemente erfolgt manuell
- Erneute Transformation: Benutzerinteraktion nur für neue Elemente

04 Auswertung

Integration der DSL Movisa in den Useware-Entwicklungsprozess

Ergebnisse

- Transformation mit Entwicklerinteraktion
- Kombination von Trace- und Mapping-Modell (Petmap)
- Wiederverwendung der Mappings

04 Auswertung

Integration der DSL Movisa in den Useware-Entwicklungsprozess

Ergebnisse

- Transformation mit Entwicklerinteraktion
- Kombination von Trace- und Mapping-Modell (Petmap)
- Wiederverwendung der Mappings

Forschungspotential

- Roundtrip-Engineering: Veränderungen im Movisa-Modell gehen verloren
- Synchronisierungsmechanismen