Using Concrete Visual Notations as First Class Citizens for Model Transformation Specification

Iman Avazpour
John Grundy
Outline

- Why CONVERt
- Approach
- Basic examples
- Case study – Minard’s Map
- Future work
Motivation

- Complex data mapping is hard
  - Programming languages & APIs
  - Scripting languages e.g. QVT, ATL, XSLT
  - Abstraction visual mappers
  - Concrete, domain-specific visual mappers
- Wanted to provide end-users with concrete, example-based data mapping tool
  - Specify own visualisations of complex data
  - Visualise source / target model data
  - Drag and drop between elements to specify mappings
  - Generate model mapping script / code (XSLT)
Example

M:1 record

1:1 field

M:1 field

M:M records

M:N record selection
c.f. Orion health data mapper (ASE 2001)
c.f. Form-based mapper (HCC 2002)
Our New Approach - CONVERdT
CONVERt – Specifying Concrete Model Visualisations

Step 1
- Notation View
- Map
- Notation Model
- Reusable Notation Repository
- Save
- Use

Step 2
- Notation Model
- Map
- Input Model
- Notation Model
- Customised Notations
- Save
- Use

Step 3
- Compose
- Create
- Visualisation

Flow diagram showing the process of specifying concrete model visualisations.
1. Specify Notational Elements

```xml
<FormFieldData>
<FieldLabel>Label</FieldLabel>
<FieldText>FieldText</FieldText>
</FormFieldData>

<StackPanel Orientation="Horizontal">
    <Label>
        <Label.Content LinkTo="FieldLabel">FieldLabel</Label.Content>
    </Label>
    <Label Content="":"/
        <TextBox Background="White" MinWidth="50">
            <TextBox.Text LinkTo="FieldText">Text1</TextBox.Text>
        </TextBox>
    </Label>
</StackPanel>
```

```
<table>
<thead>
<tr>
<th>Personal Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: Iman</td>
</tr>
<tr>
<td>Last Name: Avazpour</td>
</tr>
<tr>
<td>Age: 30</td>
</tr>
<tr>
<td>Sex: Male</td>
</tr>
<tr>
<td>Address: Here and there</td>
</tr>
</tbody>
</table>
```

```xml
<StackPanel Orientation="Horizontal">
    <Label>
        <Label.Content LinkTo="FieldLabel">FieldLabel</Label.Content>
    </Label>
    <Label Content="":/
        <TextBox Background="White" MinWidth="50">
            <TextBox.Text LinkTo="FieldText">Text1</TextBox.Text>
        </TextBox>
    </Label>
</StackPanel>
```
2. Map data fields to elements
3. Compose basic notational elements
Case study – Minard’s Map (see the paper!)

Figurative Map of successive losses in men of the French army in Russian Campaign 1812 – 1813
Evaluation and Future Work

- Range of example models visualised and mapped
  - Business, Buildings and eHealth, software (MDE)

- Range of end users surveyed
  - 11 people - business charts; 12 people – MDE (UML + Java code); Its pretty good!!! 😊

- Key issues to improve:
  - Scaling (set of “Suggesters” provided)
  - More reusable functions, notational elements
  - By-example function specification & reuse
  - Other implementations e.g. ALT, JavaScript/HTML etc
  - Live, incremental visualisation; web-based GUI
Summary

- Support end users to interactively specify rich, human-centric visualisations of complex data using a visual, drag-and-drop, by-example approach.

- Support end users to generate reusable visualisation implementations from these high-level specifications.

- Allow end users to reuse their generated, reusable model visualisations to visualise two (or more) complex data sets (i.e. example models).

- Support end users to specify model element mappings between these data sets via drag-and-drop between their concrete visualisation elements.

- Generates complex, reusable model transformation implementations from these visually specified mappings.
Questions?

CONVERt Videos & Web site:

http://www.youtube.com/watch?v=RExa0MT-zqU
https://sites.google.com/site/iavazpour/tools-manuals
CAD building design data to Hierarchical org data
References


