Two-Faced Data

One data fragment has several alternative structural representations tailored toward specific data manipulation approaches.

Vadim Zaytsev

Foreword by Eugene Syriani
Richard Paige
Steffen Zschaler
Huseyin Ergin

2015
Motivation


V. Zaytsev, A.-H. Bagge, Parsing in a Broad Sense, MoDELS, 2014
Two-Faced Data

<table>
<thead>
<tr>
<th>Applicability</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ software language processing</td>
</tr>
<tr>
<td>✓ language definition</td>
</tr>
<tr>
<td>✓ multiple syntaxes</td>
</tr>
<tr>
<td>✓ visual and textual</td>
</tr>
<tr>
<td>✓ (un)serialisation</td>
</tr>
<tr>
<td>✓ lossless</td>
</tr>
<tr>
<td>✓ data-specific algorithms</td>
</tr>
<tr>
<td>✓ multiple views</td>
</tr>
<tr>
<td>✓ multiple tools</td>
</tr>
<tr>
<td>✓ interoperability &amp; comfort</td>
</tr>
<tr>
<td>✓ FORBIDDEN EXAMPLE</td>
</tr>
</tbody>
</table>
Two-Faced Data

Structure

Metamodel_1 \text{ conformsTo } Model_1 \text{ conformsTo } F_1

Language \text{ definitionOf } Instance \text{ correspondsTo } F_1

Metamodel_2 \text{ conformsTo } Model_2 \text{ conformsTo } F_2
Two-Faced Data

Participants

✓ Language
✓ Instance
✓ Left side
  ✓ Model
  ✓ Metamodel
  ✓ Mapping
✓ Right side
  ✓ Model
  ✓ Metamodel
  ✓ Mapping
Collaborations

✓ I is an instanceOf L
  ✓ elementOf
✓ Mx models I
  ✓ correspondsTo
✓ MMx models L
  ✓ representationOf
✓ Fx maps Mx
  ✓ inputOf
✓ if Mx is updatable, need BX
✓ can have more than two faces
Two-Faced Data

Sample Code

data A = foo(bool) | bar(list[A] xs);

syntax A = "foo"? | "bar" "(" "A+ "")";

T = implode(parse(#A, input)

visit(T)
{
    foo(True) : cx += 1
    bar([]) => foo(False)
}


Two-Faced Data

Implementation
Two-Faced Data

Consequences

✓ Questions?