The Class Responsibility Assignment Case

Martin Fleck, Javier Troya, and Manuel Wimmer

TTC, Vienna
8th July, 2016
Motivation

ARTIST Project: http://www.artist-project.eu

- Models are **human-oriented** artefacts
  - Maintainability
  - Testability
  - Readability
  - Understandability
  - ...

- How to **improve** the **quality**?
- What is a **good solution**?


Motivation

Class Responsibility Assignment (CRA) Problem

- CRA deals with the creation of high-quality object-oriented models

- For solving a particular CRA problem, one needs to decide where responsibilities, i.e., class operations and attributes, belong

- When do we have to deal with CRA problems?
  - Generating class diagrams: When migrating an application from a procedural language to an object-oriented language
  - Optimizing class diagrams: During the refactoring of an existing object-oriented model

- CRA is a computationally challenging problem
  - Huge search space!
  - Considered as an optimization problem

Motivation
Solving Complex Optimization Problems

- Class Responsibility Assignment Problem
  - Modularization of features into classes -> partitioning problem

\[
B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k \\
B_0 = 1
\]
Modeling the CRA Problem

Input/Output Structures: RDG -> CD

- **Responsibility Dependency Graph (RDG) Language** (without green-colored elements)
  - One `ClassModel` contains all `Features`
  - `Features` have dependencies: functional and data

- **Class Diagram (CD) Language** (black-colored and green-colored elements)
  - `ClassModel` contains `Classes`
  - `Classes` encapsulate `Features`
  - No empty `Classes`/no unassigned `Features`
Modeling the CRA Problem

Fitness Function

\[ \text{CRA-Index} = \text{CohesionRatio} - \text{CouplingRatio} \]

\[ \text{CohesionRatio} = \sum_{c_i \in \text{Classes}} \frac{\text{MAI}(c_i, c_i)}{|M(c_i)| \times |A(c_i)|} + \frac{\text{MMI}(c_i, c_i)}{|M(c_i)| \times |M(c_i) - 1|} \]

\[ \text{CouplingRatio} = \sum_{c_i, c_j \in \text{Classes}, c_i \neq c_j} \frac{\text{MAI}(c_i, c_j)}{|M(c_i)| \times |A(c_j)|} + \frac{\text{MMI}(c_i, c_j)}{|M(c_i)| \times |M(c_j) - 1|} \]

\[ \text{MMI}(c_i, c_j) = \sum_{m_i \in M(c_i)} \sum_{m_j \in M(c_j)} \text{DMM}(m_i, m_j) \]

\[ \text{MAI}(c_i, c_j) = \sum_{m_i \in M(c_i)} \sum_{a_j \in A(c_j)} \text{DMA}(m_i, a_j) \]

\[ \text{DMA}(m_i, a_j) = \begin{cases} 1 & \text{if there is a dependency between method } m_i \text{ and attribute } a_j \\ 0 & \text{otherwise} \end{cases} \]

\[ \text{DMM}(m_i, m_j) = \begin{cases} 1 & \text{if there is a dependency between method } m_i \text{ and } m_j \\ 0 & \text{otherwise} \end{cases} \]
Modeling the CRA Problem

Example

**RDG**

- **rdg**: ClassModel
  - name = "inputRDG"

- **addItem**: Method
  - name = "addItem"

- **cartTotal**: Method
  - name = "cartTotal"

- **checkout**: Method
  - name = "checkout"

- **print**: Method
  - name = "print"

- **itemTotal**: Method
  - name = "itemTotal"

- **items**: Attribute
  - name = "items"

- **name**: Attribute
  - name = "name"

- **price**: Attribute
  - name = "price"

- **qty**: Attribute
  - name = "quantity"

- **MyShop**
  - **items**
    - addItem()
    - cartTotal()
    - checkout()

- **Cart**
  - **items**
    - addItem()
    - cartTotal()
    - checkout()

- **Item**
  - **name**
  - **price**
  - **quantity**
  - **itemTotal()**
  - **print()**

**CD**

- **MyShop**
  - **Cart**
    - **items**
  - **Item**

<table>
<thead>
<tr>
<th>Cart</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAI(cᵢ, cᵢ)</td>
<td>3</td>
</tr>
<tr>
<td>MMI(cᵢ, cᵢ)</td>
<td>1</td>
</tr>
<tr>
<td>CohesionRatio</td>
<td>1.1667</td>
</tr>
<tr>
<td>MAI(cᵢ, cⱼ)</td>
<td>1</td>
</tr>
<tr>
<td>MMI(cᵢ, cⱼ)</td>
<td>1</td>
</tr>
<tr>
<td>CouplingRatio</td>
<td>0.4444</td>
</tr>
<tr>
<td>Σ CohesionRatio</td>
<td>2</td>
</tr>
<tr>
<td>Σ CouplingRatio</td>
<td>0.4444</td>
</tr>
<tr>
<td>CRA-Index</td>
<td>1.5556</td>
</tr>
</tbody>
</table>
Modeling the CRA Problem

Producing Solutions with Transformations: RDG 2 CD

- **Example rules**
  - Shown in Henshin syntax
  - Many other possibilities to solve CRA

```diagram
Rule createClass(className: EString)

«forbid» : Class
  name = className

«forbid» : ClassModel

«create» : Class
  name = className
```

```diagram
Rule assignFeature(featureName: EString, className: EString) @ClassModel

«forbid» assignedClass: Class
  isEncapsulatedBy

«forbid» : Feature
  name = featureName

«create» : Class
  name = className
```
Solving the CRA Problem
Going beyond Random Search

- **Main challenge:** Finding the best output model → finding the best rule application sequence

- **Optimization methods** to orchestrate transformation rules?

---

Evaluation

Input Models and Evaluation Schema

- **Example Models**

<table>
<thead>
<tr>
<th></th>
<th>Input A</th>
<th>Input B</th>
<th>Input C</th>
<th>Input D</th>
<th>Input E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>40</td>
<td>80</td>
</tr>
<tr>
<td>Methods</td>
<td>4</td>
<td>8</td>
<td>15</td>
<td>40</td>
<td>80</td>
</tr>
<tr>
<td>Data Dep.</td>
<td>8</td>
<td>15</td>
<td>50</td>
<td>150</td>
<td>300</td>
</tr>
<tr>
<td>Functional Dep.</td>
<td>6</td>
<td>15</td>
<td>50</td>
<td>150</td>
<td>300</td>
</tr>
</tbody>
</table>

- **Evaluation Schema**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Weight</th>
<th>MaxPoints</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completeness &amp; Correctness</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Optimality</td>
<td>3</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>Complexity</td>
<td>2</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Flexibility</td>
<td>2</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Performance</td>
<td>2</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>
Solutions


- NMF by Georg Hinkel
- VIATRA by András Szabolcs Nagy et al.
- UML-RSDS by Kevin Lano et al.
- ATL/Java by Leif Arne Johnsen et al.
- Excel by Maximiliano Vela et al.
- SDMLib by Christoph Eickhoff et al.
- MDEOptimiser by Alexandru Burdusel et al.
- Henshin by Kristopher Born et al.
- SIGMA by Filip Krikava