Class Responsibility Assignment as Fuzzy Constraint Satisfaction

Shinpei Hayashi†, Takuto Yanagida‡, Motoshi Saeki†, and Hidetoshi Mimura‡

†Tokyo Institute of Technology ‡Shizuoka University
Class Responsibility Assignment (CRA)

- Deciding a mapping $A : M \rightarrow K$

Knowing responsibilities:
- name
- piece
- board
- location
- dice
- faceValue

Doing responsibilities:
- takeTurn()
- Die()
- roll()
- setLocation()
- getPlayer()
- getFaceValue()
- getName()
- setLocation()

Responsibilities ($M$)

Assignment ($A$)

Classes ($K$)
Towards Quality CRA

- Example criterion: *Low Coupling*

### CRA 1

<table>
<thead>
<tr>
<th>Die</th>
<th>Player</th>
<th>Piece</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX faceValue</td>
<td>name</td>
<td>location</td>
</tr>
<tr>
<td>Die()</td>
<td>piece</td>
<td>get_location()</td>
</tr>
<tr>
<td>getFaceValue()</td>
<td>board</td>
<td>getName()</td>
</tr>
<tr>
<td>Player()</td>
<td>dice</td>
<td>roll()</td>
</tr>
<tr>
<td>takeTurn()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>getLocation()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>getName()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>roll()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CRA 2

<table>
<thead>
<tr>
<th>Die</th>
<th>Player</th>
<th>Piece</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX faceValue</td>
<td>name</td>
<td>location</td>
</tr>
<tr>
<td>Die()</td>
<td>piece</td>
<td>get_location()</td>
</tr>
<tr>
<td>getFaceValue()</td>
<td>board</td>
<td>getName()</td>
</tr>
<tr>
<td>roll()</td>
<td>dice</td>
<td>roll()</td>
</tr>
<tr>
<td>Player()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Challenges for Automating CRA

- CRA is over-constrained
  - **Low Coupling**: The distance between two classes having related responsibilities should be short.
  - **High Cohesion**: The relation between two responsibilities in close classes should be close.

A realistic solution needed, which satisfies constraints *to some extent*
Toward Interactive Tool

- Support of trial-and-error in design process
  
  - **Stability:**
    - "I want to improve my manually-assigned model. *Do not DRASTICALLY modify it!*"
  
  - **Users Intention:**
    - "I found that these two responsibilities should be assigned to *the same class / different classes*"

  flexibly configurable technique needed
Our Approach

- Formulating CRA using *Fuzzy Constraint Satisfaction Problem* (FCSP)
  - Combinational search problem in AI field
  - Benefits
    - No need to define a monolithic evaluation function
      - Each criterion is naturally represented as *fuzzy constraints*
    - Usage of well-maintained solvers
FCSP

- **Variable:** $X = \{x_1, x_2, ..., x_n\}$
- **Domain:** $D = \{D_1, D_2, ..., D_n\}$
- **Constraint:** $C = \{c_1, c_2, ..., c_r\}$
  - inc. Unary and binary constraints
  - Each constraint has its satisfaction degree ($\mu_R$) [0, 1]
- **Objective:**
  - Maximizing $\min_{c \in C} \mu_R$
Formulation

- **Variable** $x$ \(\leftarrow\) Responsibility $m \in M$
- **Domain** $D$ \(\leftarrow\) Set of classes $K$
- **Constraint** $c$ \(\leftarrow\) Assignment strategy

(3 responsibilities)
Given Information

- Normalized two measures are used
  - **Class Distance** $cd : K^2 \rightarrow [0, 1]$
    
    0
    \[ \begin{array}{c}
    \text{When } k_1 = k_2 \\
    \end{array} \]
    
    1
    \[ \begin{array}{c}
    \text{When the distance between } k_1 \text{ and } k_2 \text{ is the farthest} \\
    \end{array} \]
  
  - **Responsibility Relevance** $mr : M^2 \rightarrow [0, 1]$
    
    0
    \[ \begin{array}{c}
    \text{When } m_1 \text{ is no relevance with } m_2 \\
    \end{array} \]
    
    1
    \[ \begin{array}{c}
    \text{When the relevance between } m_1 \text{ and } m_2 \text{ is the highest} \\
    \end{array} \]
Constraints

- $c^{lc}$: Low Coupling
  - relevant responsibilities are in distant classes

- $c^{hc}$: High Cohesion
  - irrelevant responsibilities are in closer classes

- $c^{s}$: Stability
  - responsibilities moved from the initial assignment

- $c^{same}$, $c^{diff}$: Users Intention
  - distance between the specified responsibilities does not follow
**c^{lc}: Low Coupling**

- Binary constraint for a pair of variables
- Satisfaction degree decreases when relevant responsibilities are in distant classes

For $m_1$ and $m_2$,

$$\mu R_c(k_1, k_2) = \{ -mr(m_1, m_2)cd(k_1, k_2) + 1 \}^w$$

(When $w = 1$)
**c^{hc}: High Cohesion**

- Binary constraint for a pair of variables
- Satisfaction degree decreases when irrelevant responsibilities are in closer classes

For $m_1$ and $m_2$,

$$\mu R_c(k_1, k_2) = \{ (1 - mr(m_1, m_2))cd(k_1, k_2) + mr(m_1, m_2) \}^w$$

(When $w = 1$)
\( c^s: \text{Stability} \)

- Unary constraint for each variable
- Satisfaction degree decreases when the class to which a responsibility belongs in the current assignment is far from that in the given assignment

For \( m \),

\[
\mu R_c(k) = \left\{ 1 - cd(k_{\text{orig}}, k) \right\}^w
\]
\( \text{c}_{\text{same}} / \text{c}_{\text{diff}} : \text{Intention} \)

- Binary constraint for each pair of variables
- Satisfaction degree decreases based on the distance between the target classes

For \( m_1 \) and \( m_2 \),

\[
\begin{align*}
\mu R_{\text{c}_{\text{same}}}(k_1, k_2) &= \{ 1 - cd(k_1, k_2) \}^w \\
\mu R_{\text{c}_{\text{diff}}}(k_1, k_2) &= cd(k_1, k_2)^w
\end{align*}
\]
Example: Constraints

- ClassA
- ClassB

- roll()
- setLocation()
- takeTurn()
Example: Constraints

- Low Coupling
- High Cohesion
- (Intention)
Example: Constraints

- Low Coupling
- High Cohesion
- Intention
Evaluation Questions

• EQ 1:  
  How accurately does our technique assign responsibilities from scratch?

• EQ 2:  
  How accurately does our technique fix the assignment of responsibilities if an initial assignment is given?

• EQ 3:  
  Does our technique fix the assignment when users’ intentions are given?

• EQ 4:  
  Is the calculation of the assignment performed fast enough?
Summary of Evaluation

- **EQ 1:** How accurately does our technique assign responsibilities from scratch? 
  - **Response:** So-so.

- **EQ 2:** How accurately does our technique fix the assignment of responsibilities if an initial assignment is given? 
  - **Response:** Good.

- **EQ 3:** Does our technique fix the assignment when users’ intentions are given? 
  - **Response:** Yes.

- **EQ 4:** Is the calculation of the assignment performed fast enough? 
  - **Response:** Yes.
Summary of Evaluation

- **EQ 1:** How accurately does our technique assign responsibilities from scratch?
  - A certain level of precision.
    - Monopoly: 69%
    - NextGenPos: 33%

- **EQ 2:** How accurately does our technique fix the assignment of responsibilities if an initial assignment is given?
  - Good level of precision.
    - Monopoly: 58%
    - NextGenPos: 73%

- **EQ 3:** Does our technique fix the assignment when users' intentions are given?
  - Yes.
    - 2 of 3 constraints hold.

- **EQ 4:** Is the calculation of the assignment performed fast enough?
  - Yes.
    - e.g., Fix: <1 ms
Experimental Setup

- Example models from a CRA textbook

<table>
<thead>
<tr>
<th>System</th>
<th># classes</th>
<th># responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monopoly</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>NextGenPos</td>
<td>9</td>
<td>30</td>
</tr>
</tbody>
</table>

- Reverse engineering from source code
  - Examples and oracles were extracted from textbook
  - Class distance $cd$ and Responsibility relevance $mr$ were measured based on the oracle
EQ 1 (from scratch)

How accurately does our technique assign responsibilities from scratch?

- Prepared an empty model and assigned all the responsibilities
EQ 1 (from scratch)

How accurately does our technique assign responsibilities from scratch?

Monopoly: 69%

NextGenPos: 33%

Incorrect assignment [Oracle]
EQ 2 (w/ initial model)

How accurately does our technique fix the assignment of responsibilities if an initial assignment is given?

- Detached each responsibility and re-assigned it

**Result**

- Monopoly: 58%
  (15 resp.)
- NextGenPos: 73%
  (22 resp.)
EQ 3 (intention)

Does our technique fix the assignment when users’ intentions are given?

- Added 3 intention constraints in Monopoly
  → 2 of 3 were worked well

Users intention-based constraints are feasible.
EQ 4: Execution Time

Is the calculation of the assignment performed fast enough?

• Implementation
  – Our FCSP library w/ fuzzy forward checking
  – on Java 7 (Window 7, Intel Core i7, 2.93GHz)

• Result
  – **Experiment for EQ 1** (≠ actual usage)
    • Monopoly: 20 ms
    • NextGenPos: 8550 ms
  – **Experiment for EQ 2**
    • < 1 ms
  – **Experiment for EQ 3**
    • 20 ms

Yes, fast enough.
Discussion/Conclusion

EQ 2: improvement of existing model

EQ 3: Addition of users intention

EQ 4: Execution time

Might be feasible to develop an interactive CASE tool for supporting CRA

(Flexibility by formulating CRA as fuzzy CSP)
Future Work

- Richer case studies for confirming scalability
  - Applying our technique to real systems
- Use of other software metrics
  - e.g., LCOM*
- Expressing other strategies as fuzzy constraints
  - e.g., GRASP

Implementing CASE tool for designers
Class Responsibility Assignment (CRA)

- Deciding a mapping $A : M \rightarrow K$

Knowing responsibilities:
- name
- board
- piece
- location
- MAX

Doing responsibilities:
- roll()
- setLocation()
- takeTurn()
- getPlayer()
- getFaceValue()
- getLocation()

Formulation

- Variable $x \leftarrow$ Responsibility $m \in M$
- Domain $D \leftarrow$ Set of classes $K$
- Constraint $c \leftarrow$ Assignment strategy

Syntax:
$$c_1 \leq c_2 \leq c_3 \leq c_4 \leq c_5 \leq c_6$$

Constraints

- $c^l$: Low Coupling
  - relevant responsibilities are in distant classes

- $c^h$: High Cohesion
  - irrelevant responsibilities are in closer classes

- $c^s$: Stability
  - responsibilities move from the initial assignment

- $c^{same}$, $c^{diff}$: Users Intention
  - $c^{same}$: Distance between the specified responsibilities do not follow

Discussion

EQ 2: Improvement of existing model
EQ 3: Addition of users intention
EQ 4: Execution time

Might be feasible to develop an interactive CASE tool for supporting CRA

(Flexibility by formulating CRA as fuzzy CSP)
Credits

- Photo by teamaskins
  - CRC Cards | Flickr
    http://www.flickr.com/photos/teamaskins/130003950/