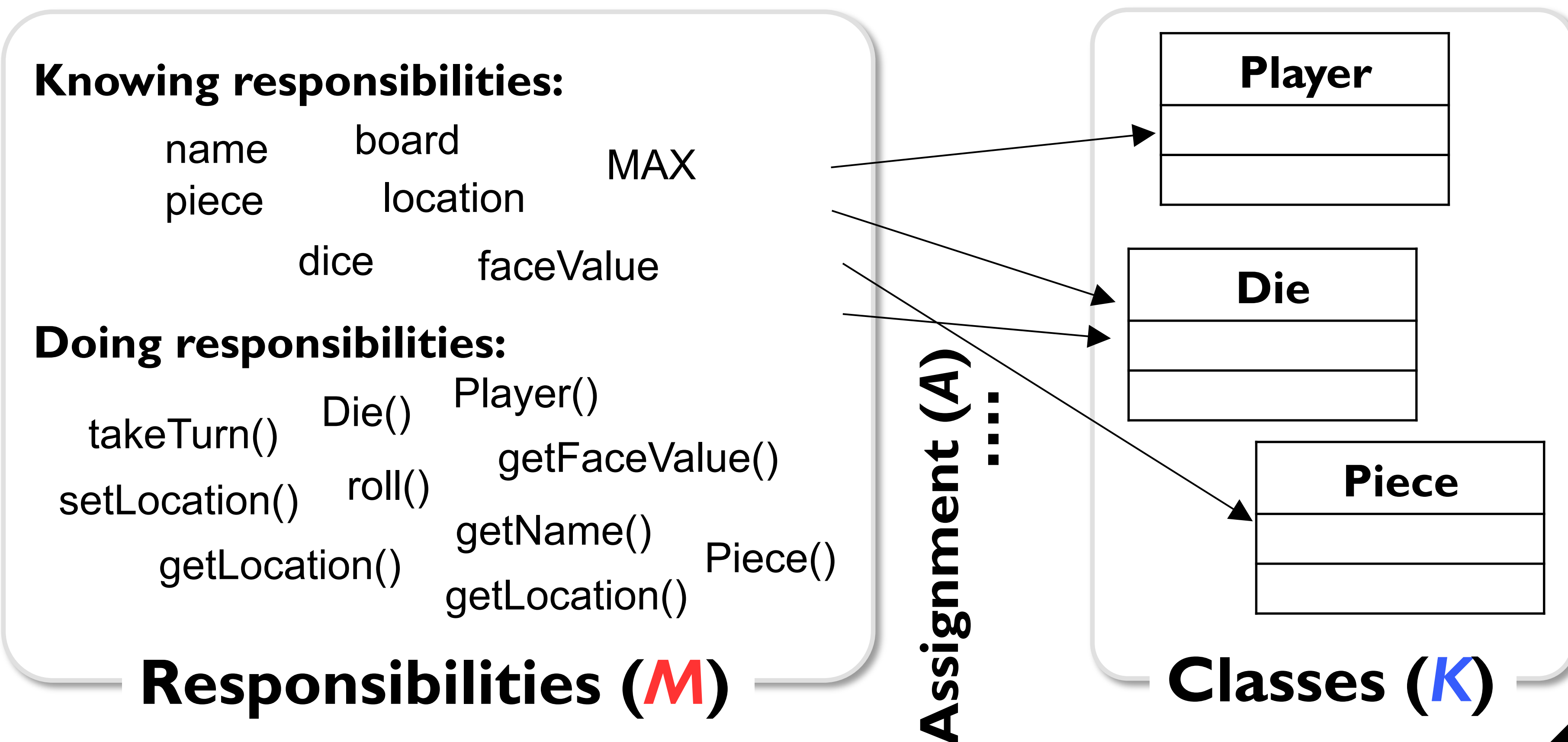
Class Responsibility Assignment as Fuzzy Constraint Satisfaction

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

[†]Tokyo Institute of Technology [‡]Shizuoka University

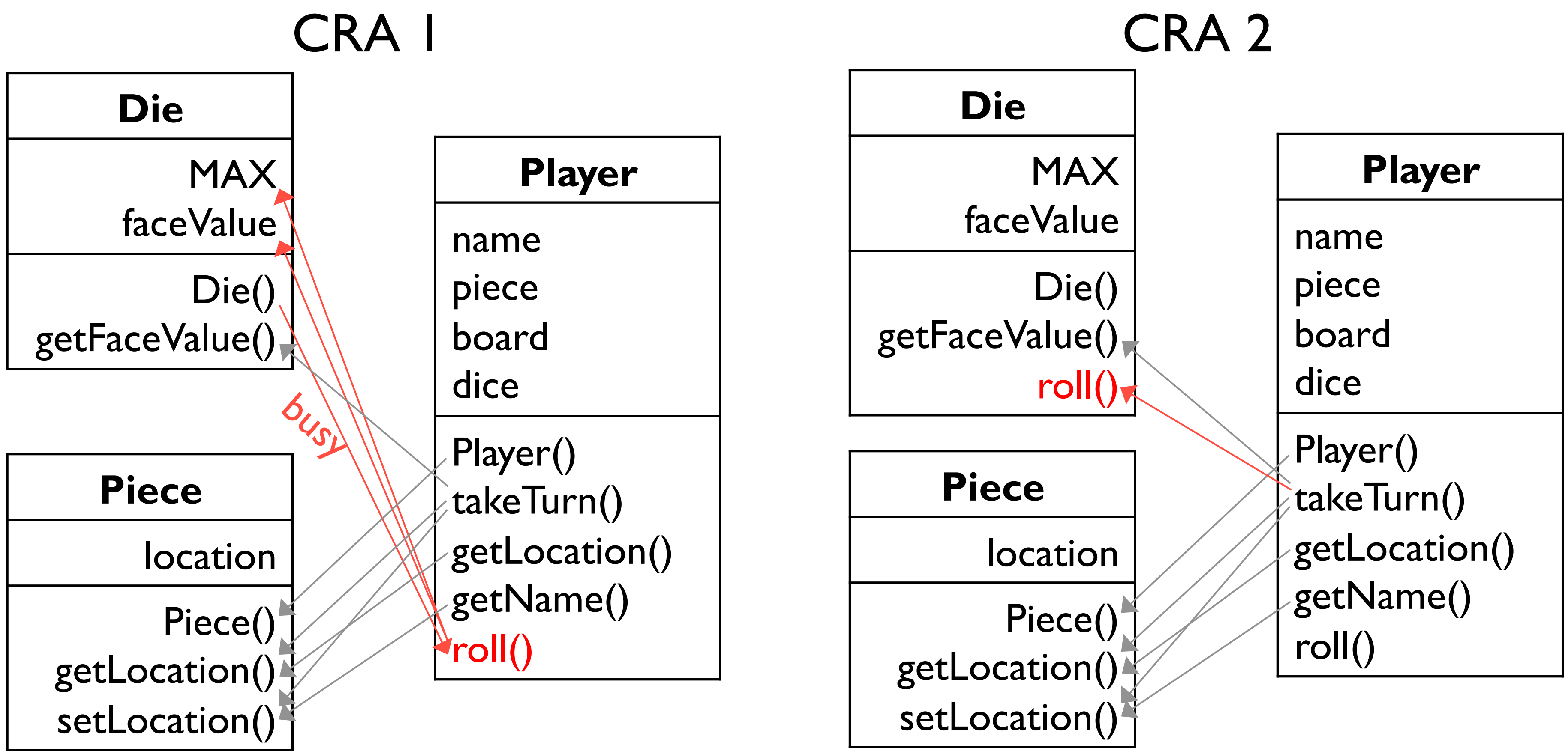
Class Responsibility Assignment (CRA)

- Deciding a mapping ***A*** **:** ***M*** → ***K***



Towards Quality CRA

- Example criterion: *Low Coupling*



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.

Trade-off

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*"

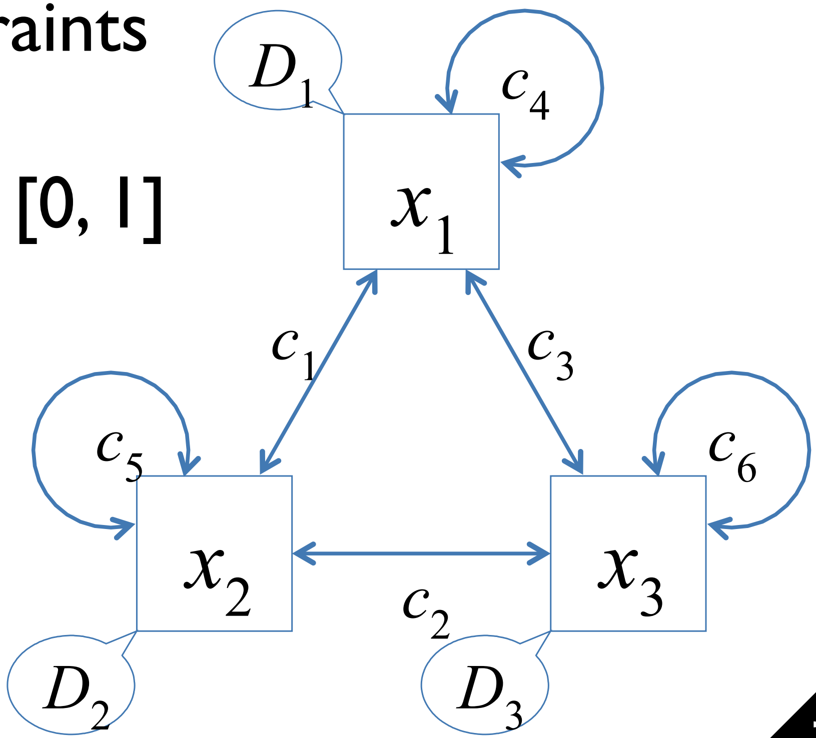
→ flexibly configurable technique needed

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

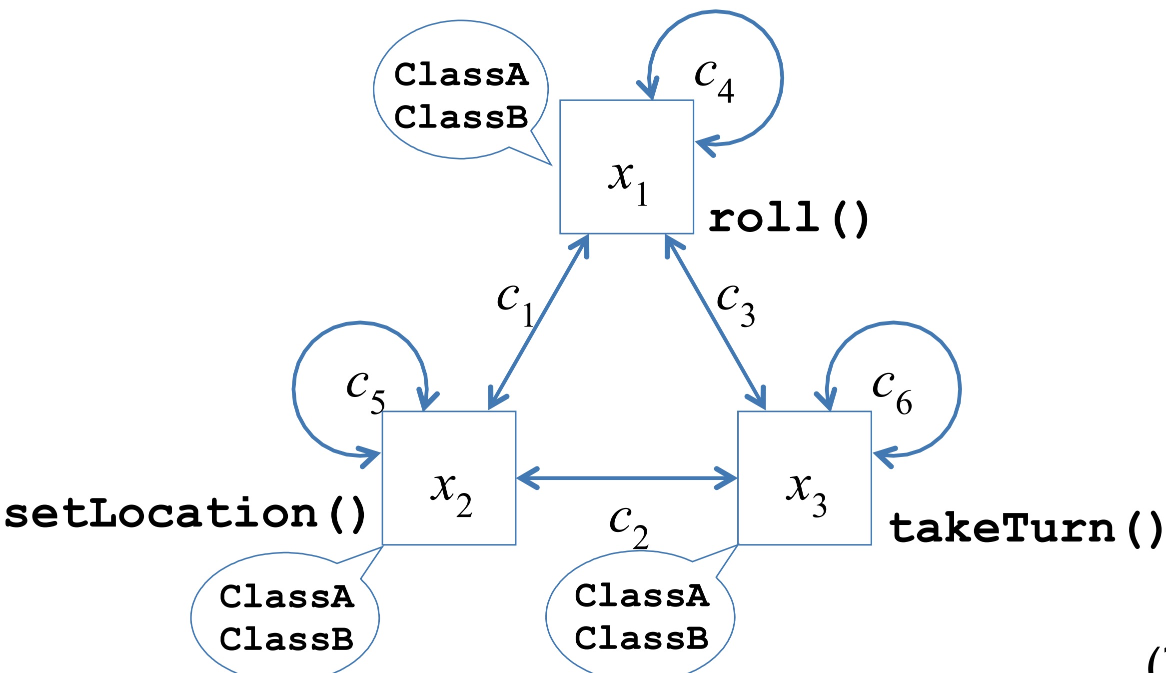
FCSP

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



Formulation

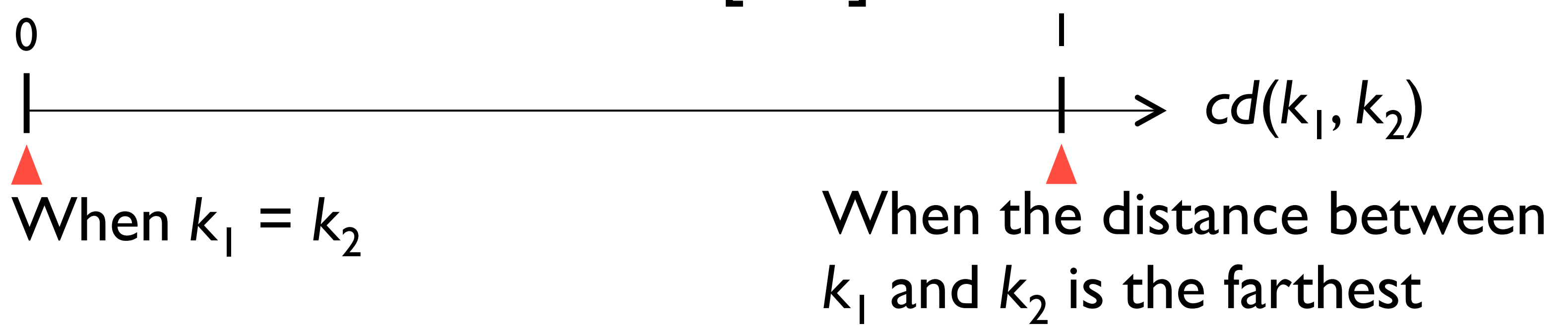
- Variable x ← Responsibility $m \in M$
- Domain D ← Set of classes K
- Constraint c ← Assignment strategy



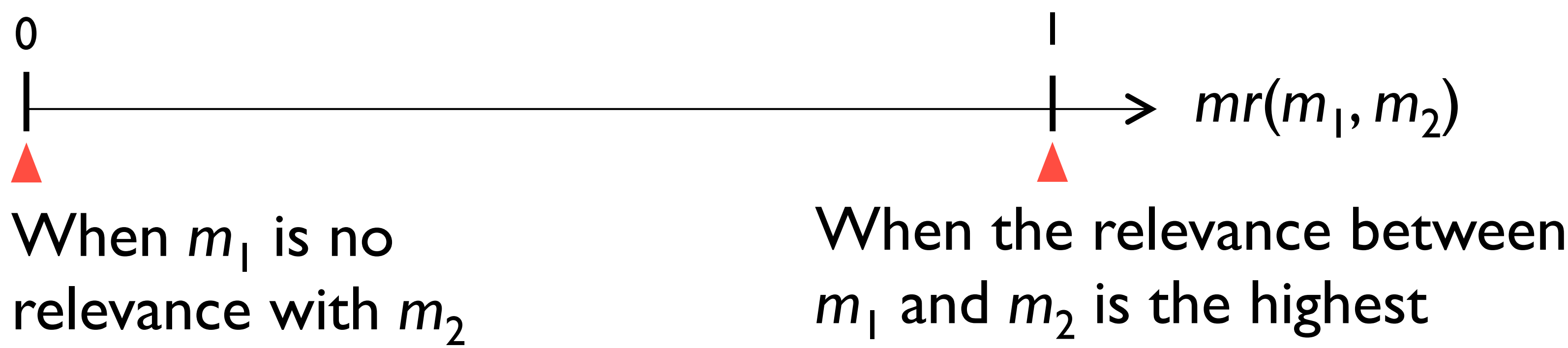
(3 responsibilities)

Given Information

- Normalized two measures are used
  - Class Distance $cd : K^2 \rightarrow [0, 1]$



  - Responsibility Relevance $mr : M^2 \rightarrow [0, 1]$



Constraints

- c^{lc}: Low Coupling

relevant responsibilities are in distant classes

- c^{hc}: High Cohesion

irrelevant responsibilities are in closer classes

- c^{s}: Stability

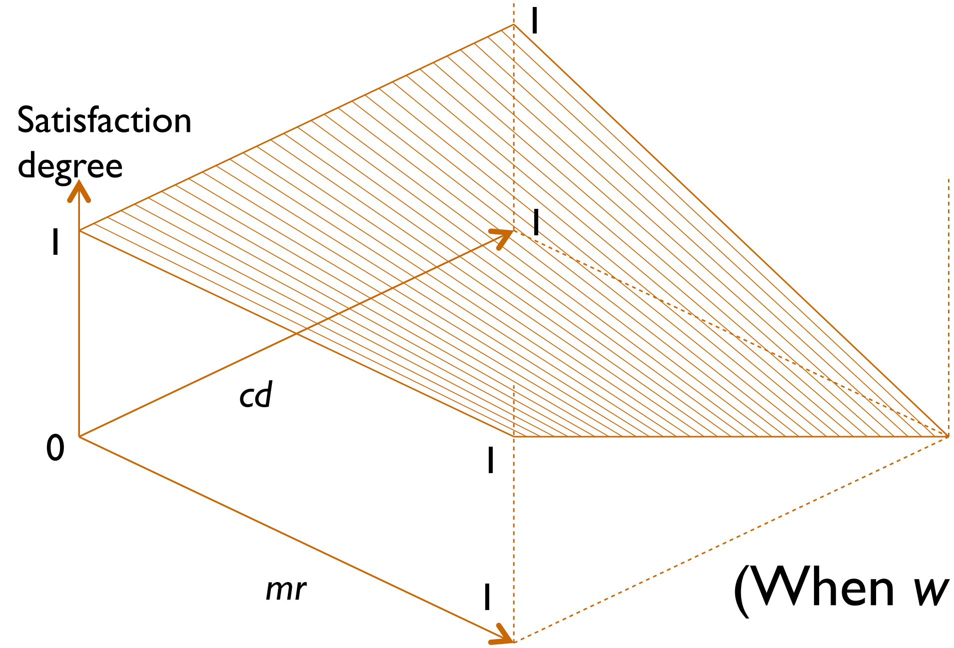
responsibilities moved from the initial assignment

- 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



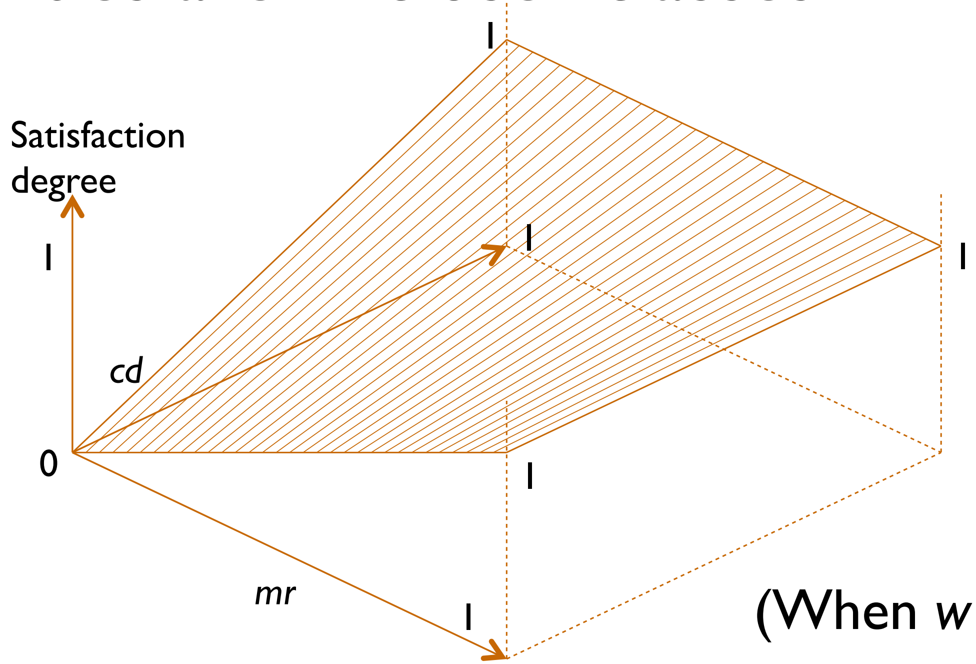
(When w = 1)

For m_1 and m_2,

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

c^{hc}: High Cohesion

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



(When w = 1)

For m_1 and m_2,

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

c^s: 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*, $\boxed{\mu R_c(k) = \{\, 1 - cd(k_{orig}, k)\, \}^w}$

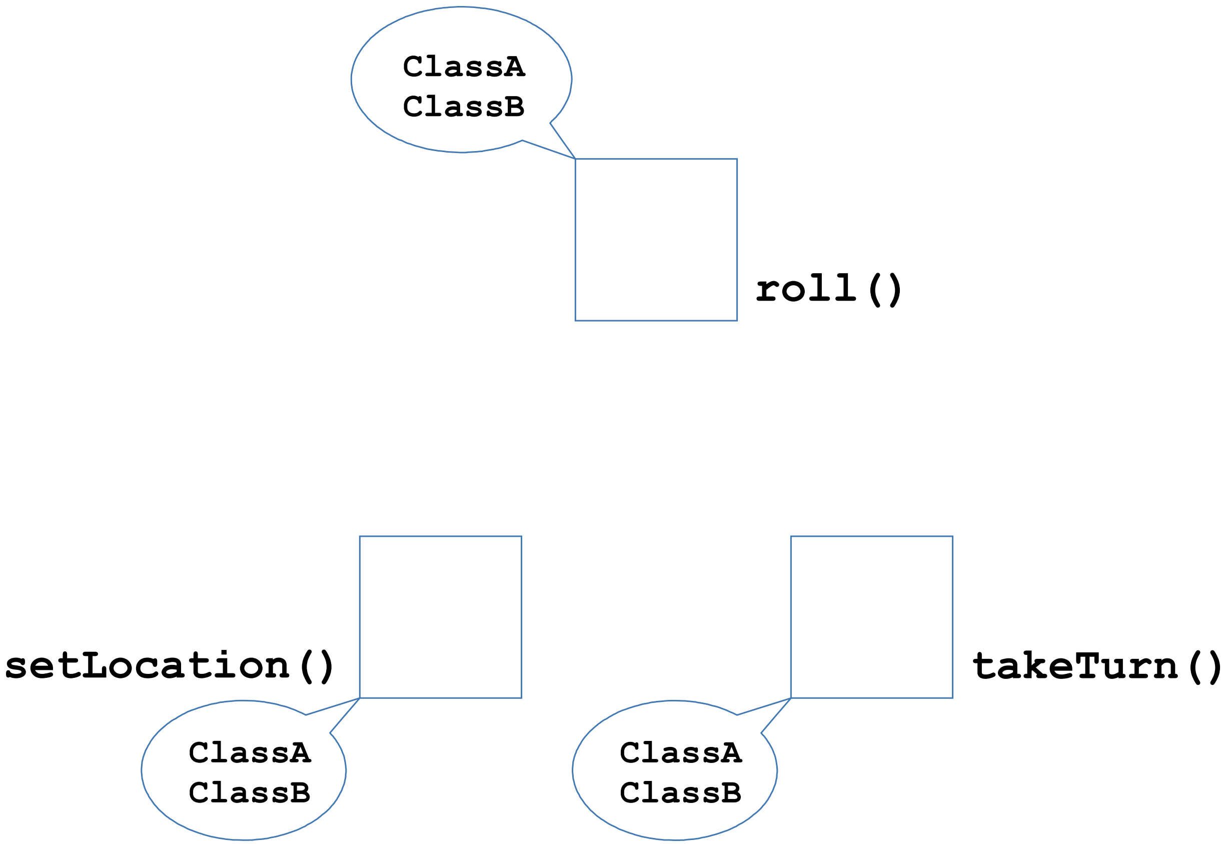
c^{same} / c^{diff} : Intention

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

For m_1 and m_2,

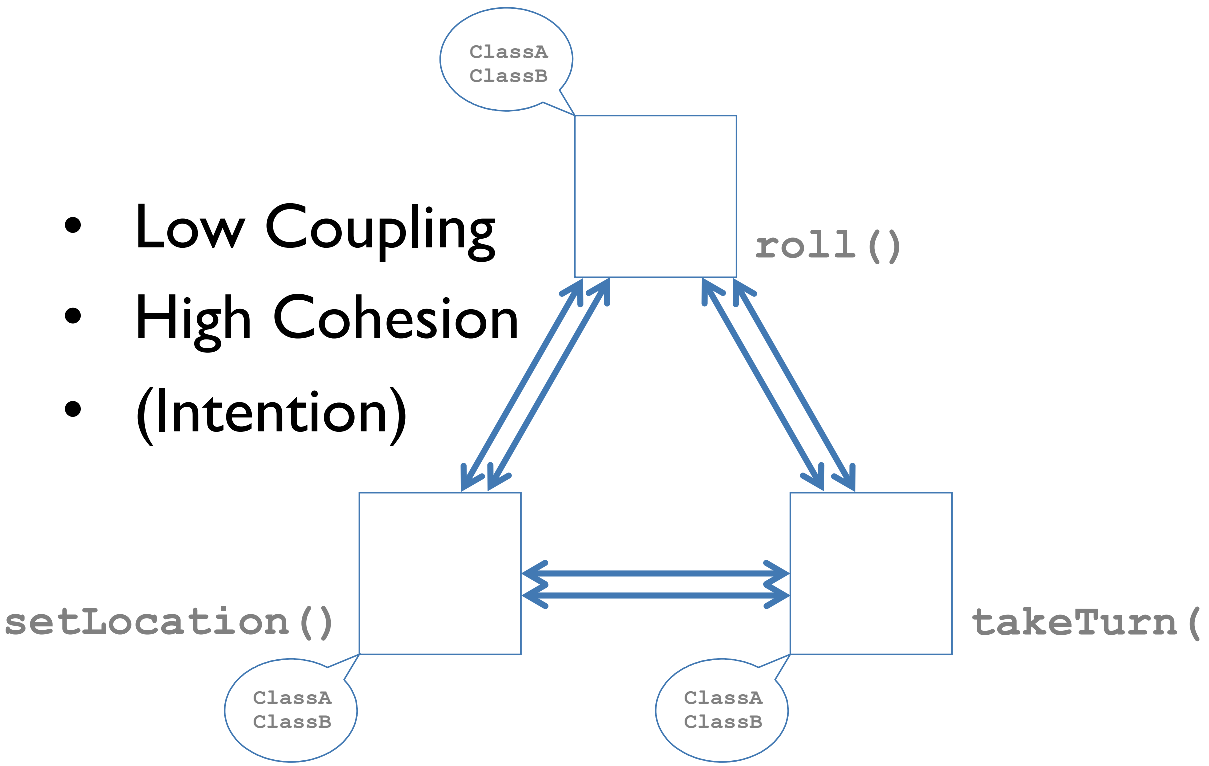
$$\mu R_{c^{\mathrm{same}}}(k_1, k_2) = \{\ 1 - cd(k_1, k_2)\ \}^w$$
$$\mu R_{c^{\mathrm{diff}}}(k_1, k_2) = cd(k_1, k_2)^w$$

Example: Constraints



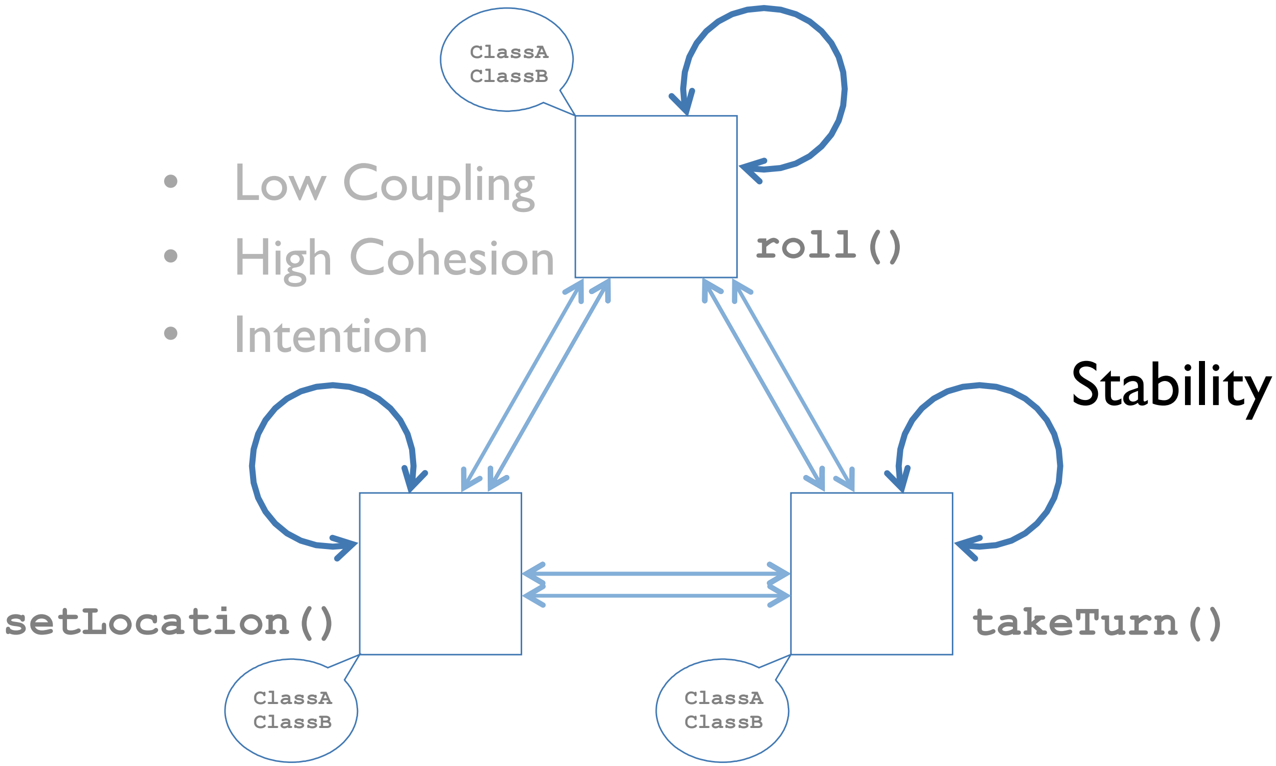
Example: Constraints

- 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?

  So-so.

- EQ 2:

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

  Good.

- EQ 3:

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

  Yes.

- EQ 4:

  Is the calculation of the assignment performed fast enough?

  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: <1ms

Experimental Setup

- Example models from a CRA textbook

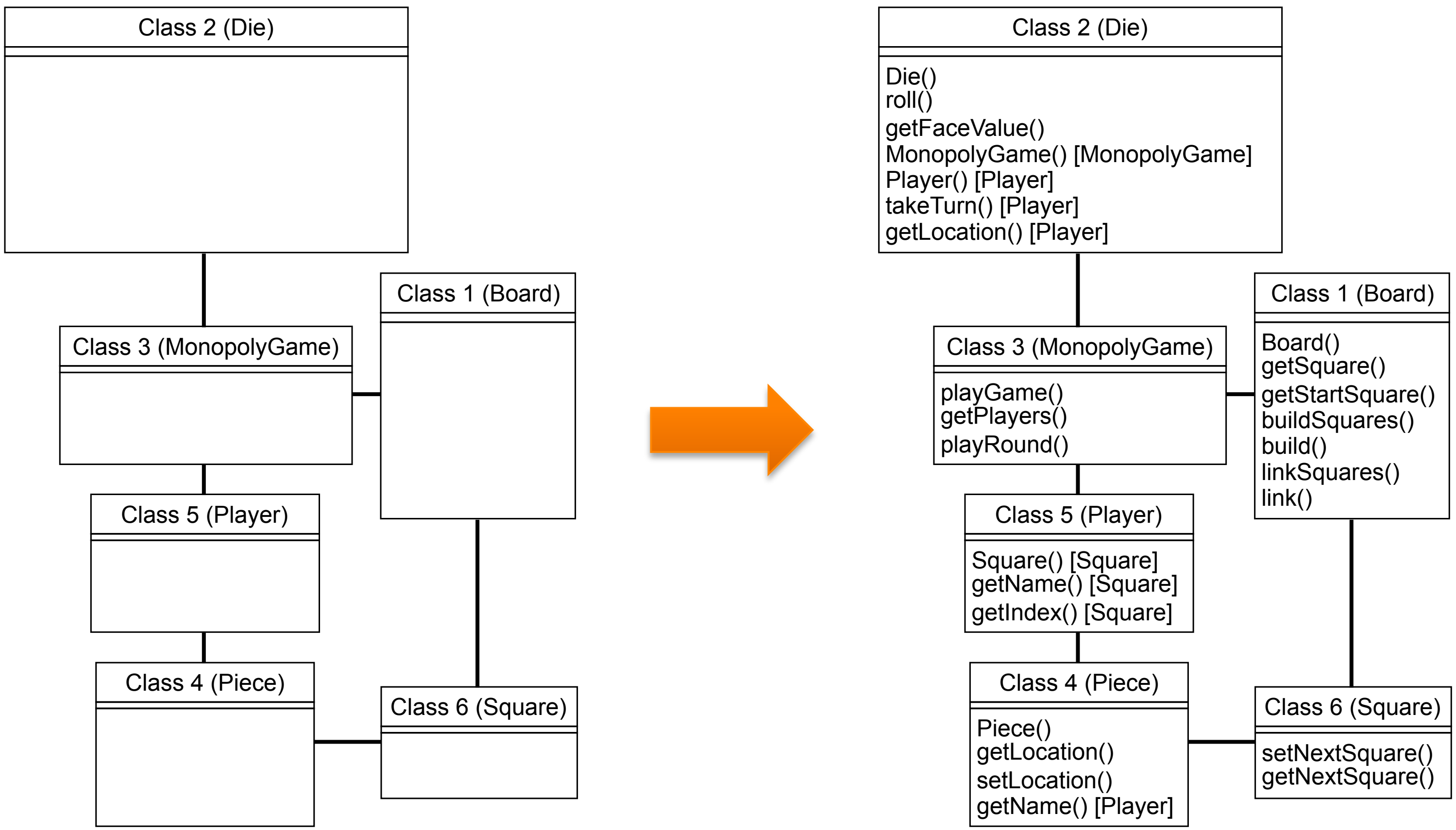
System	# classes	# responsibilities
Monopoly	6	26
NextGenPos	9	30

- 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?

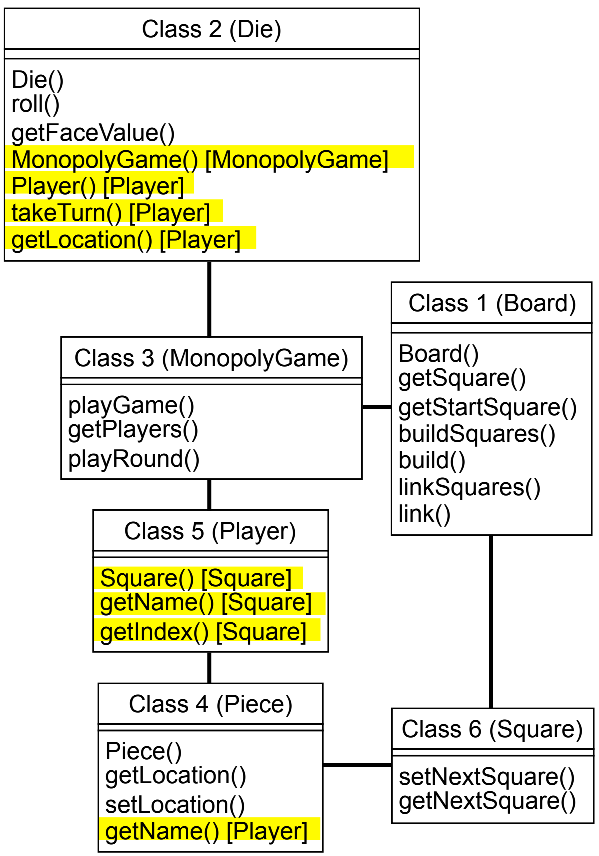
- Prepared an empty model and assigned all the responsibilities



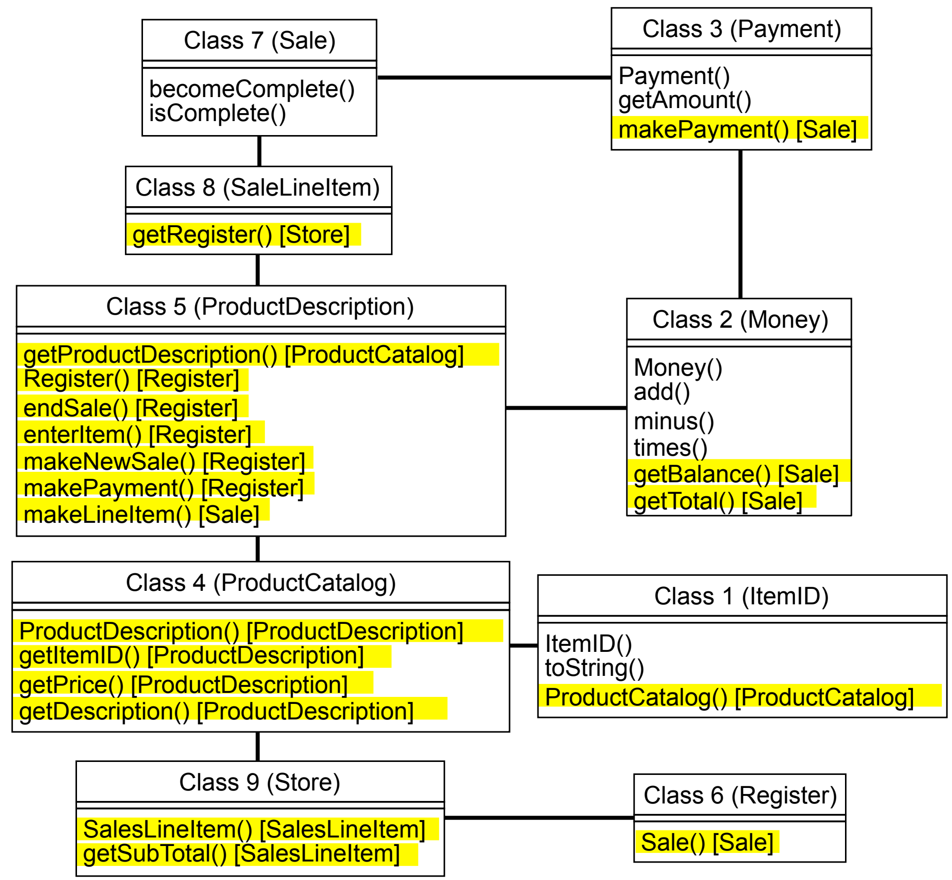
EQ 1 (from scratch)

How accurately does our technique assign responsibilities from scratch?

Monopoly: 69%



NextGenPos: 33%

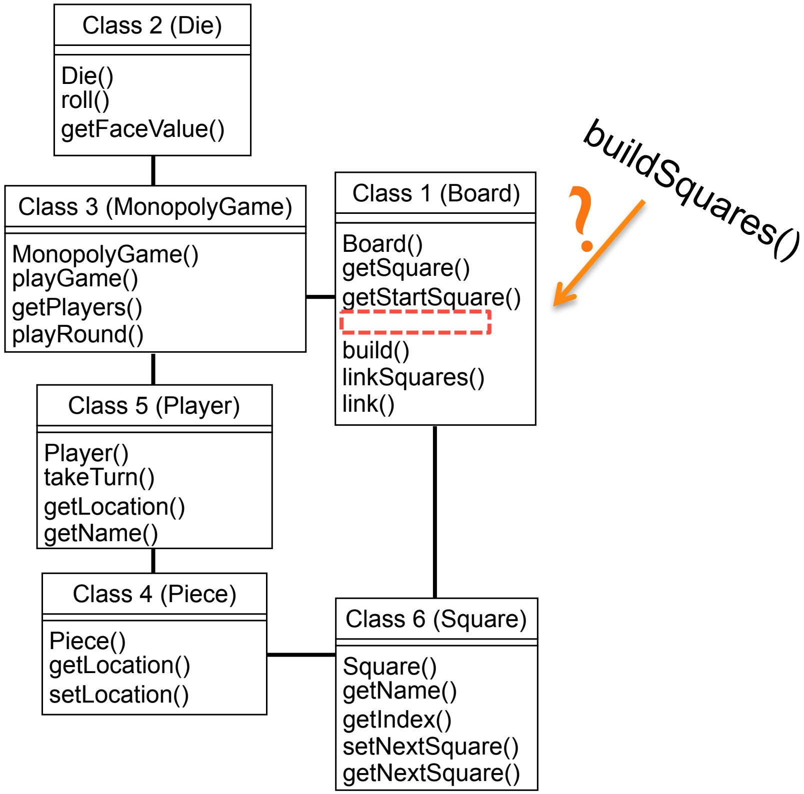


Incorrect assignment [Oracle]

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



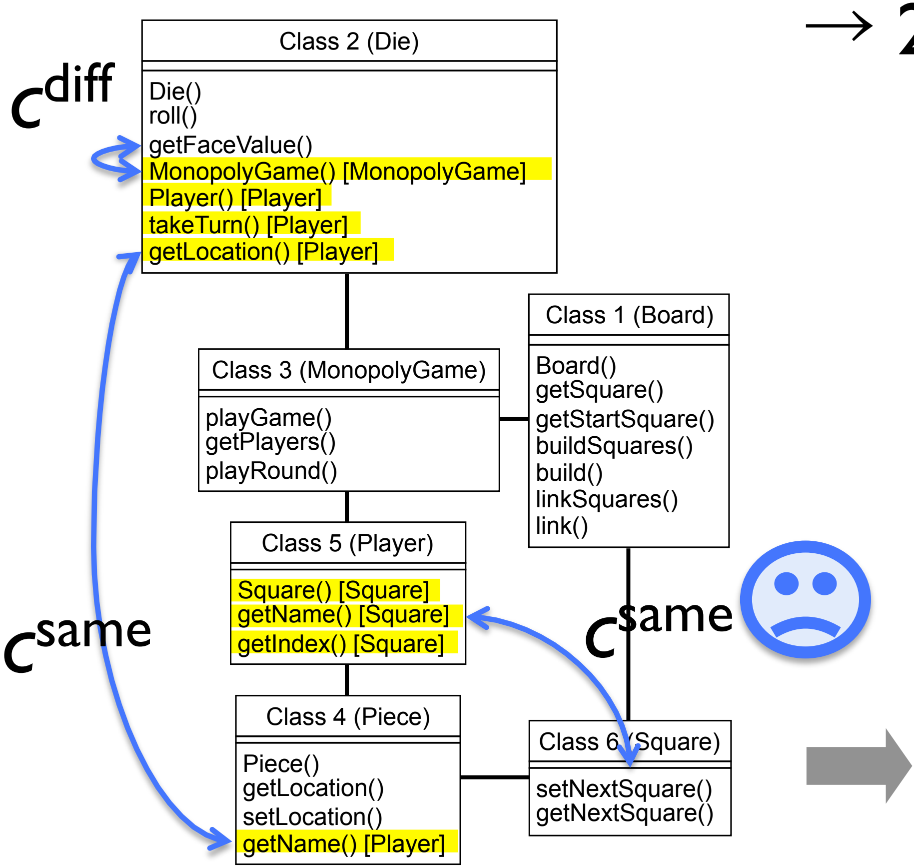
- **Result**
  - Monopoly: 58% (15 resp.)
  - NextGenPos: 73% (22 resp.)

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** ($\neq$ 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)

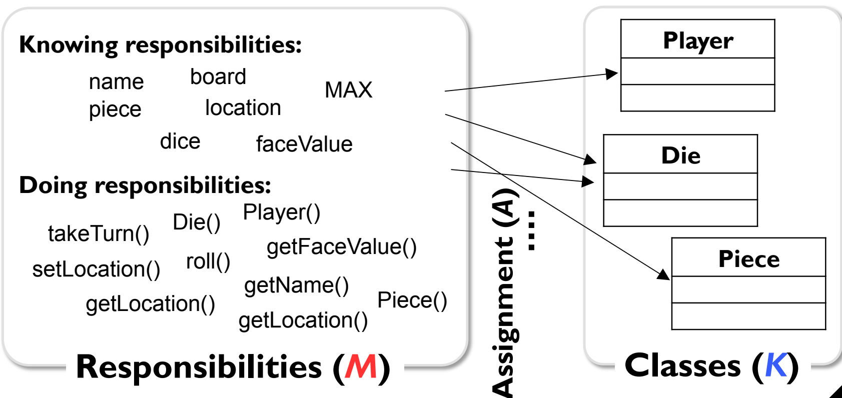
Future Work

- 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

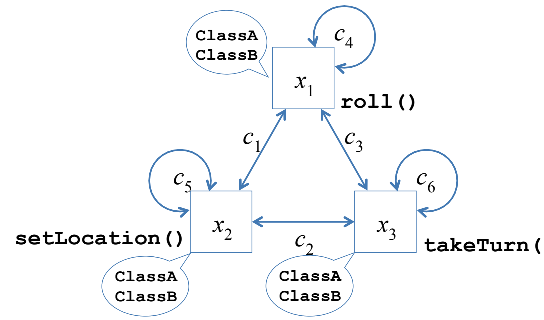
Class Responsibility Assignment (CRA)

- Deciding a mapping ***A*** **:** ***M*** → ***K***



Formulation

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



(3 responsibilities)

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 move from the initial assignment
- c^{same}, c^{diff}: Users Intention
  - 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/