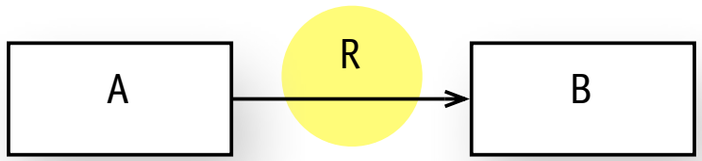
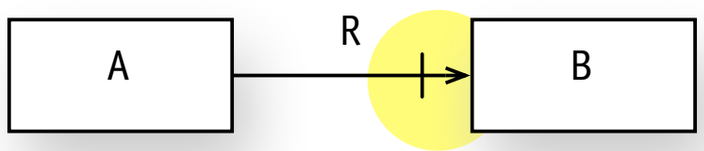
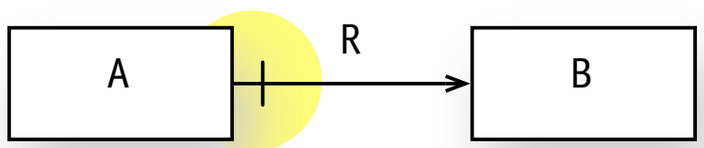
syntax summary relations



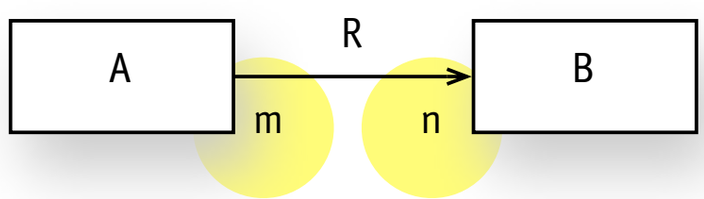
$R \subseteq A \times B$



over time, each A is mapped by R to the same Bs



over time, R maps the same As to each B

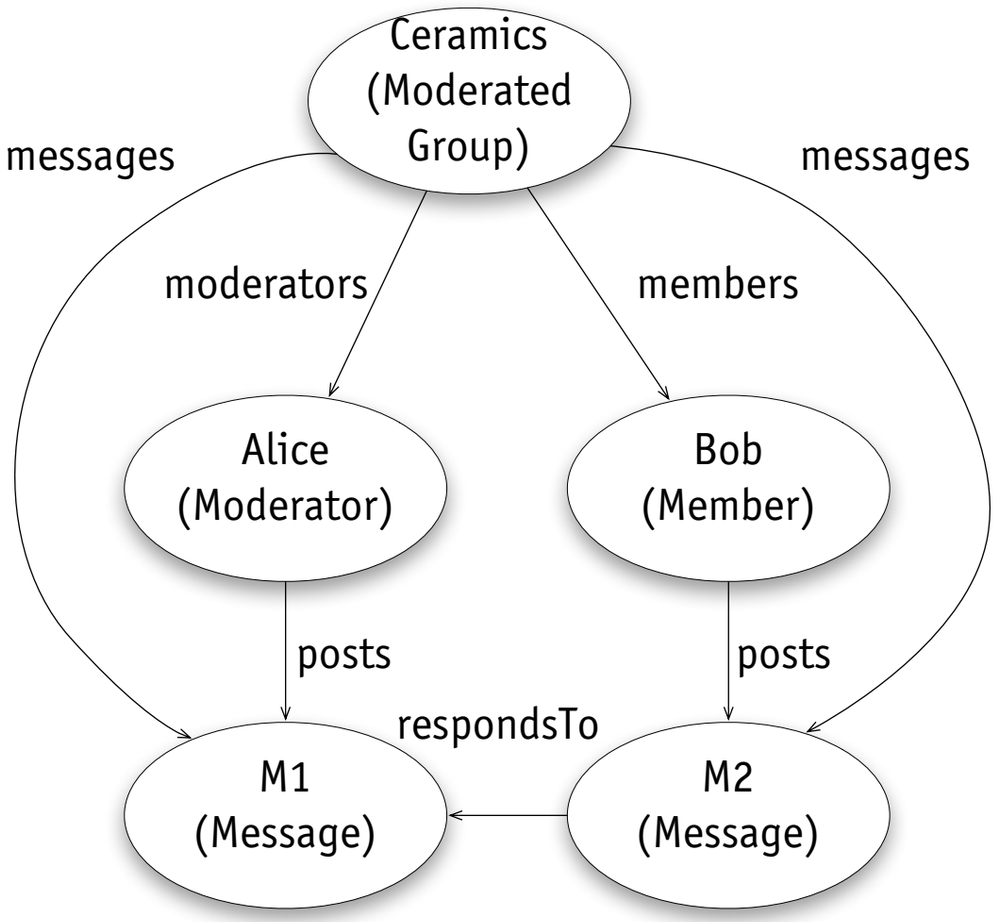


R maps each A to n Bs

R maps m As to each B

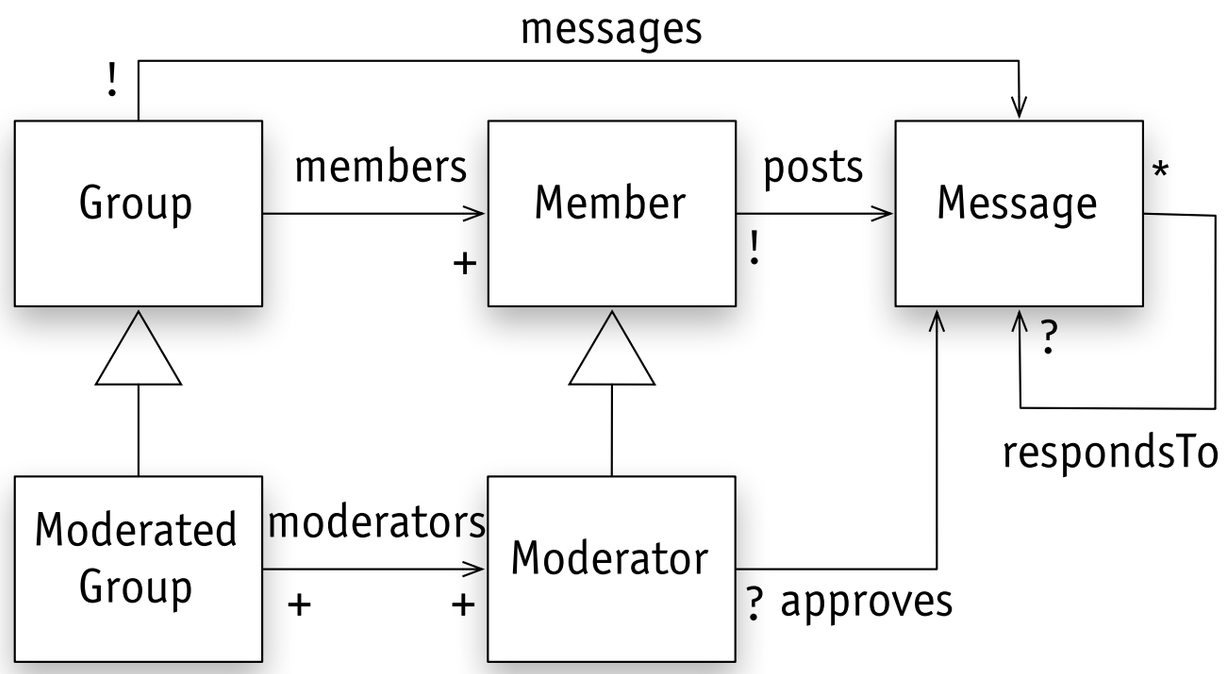
+ one or more
* zero or more
! exactly one
? at most one
omitted = *

semantics examples

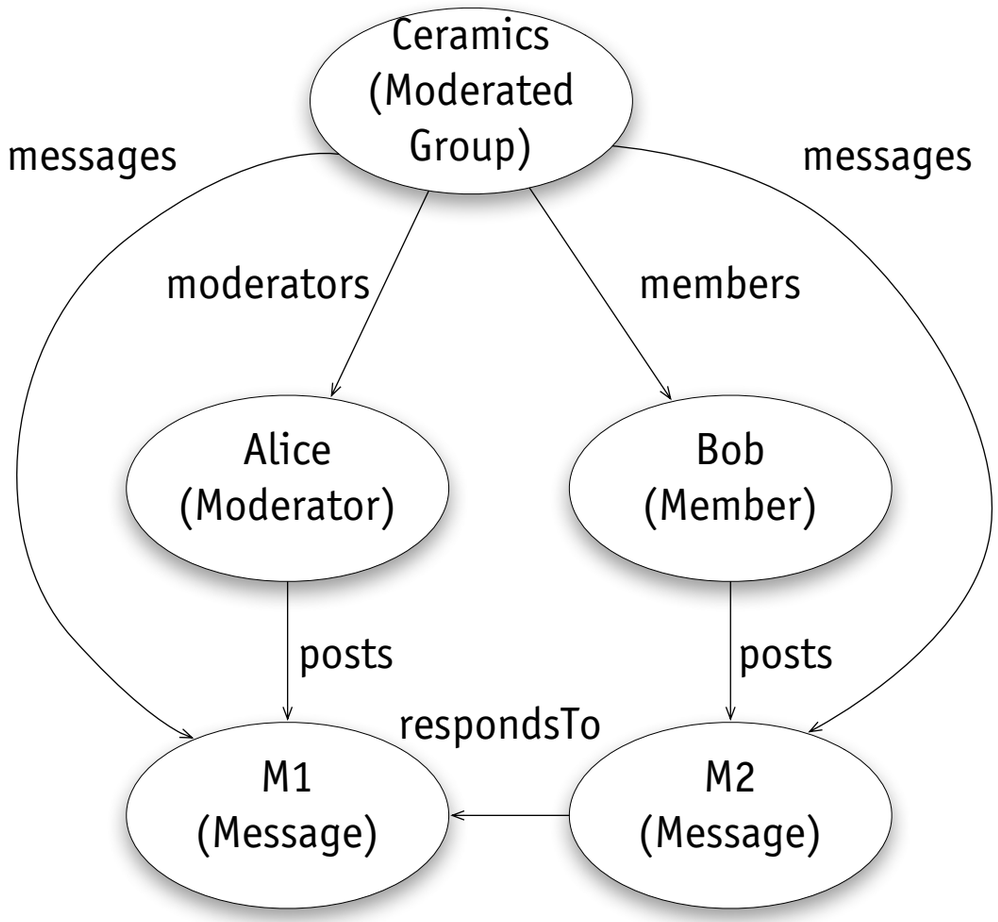


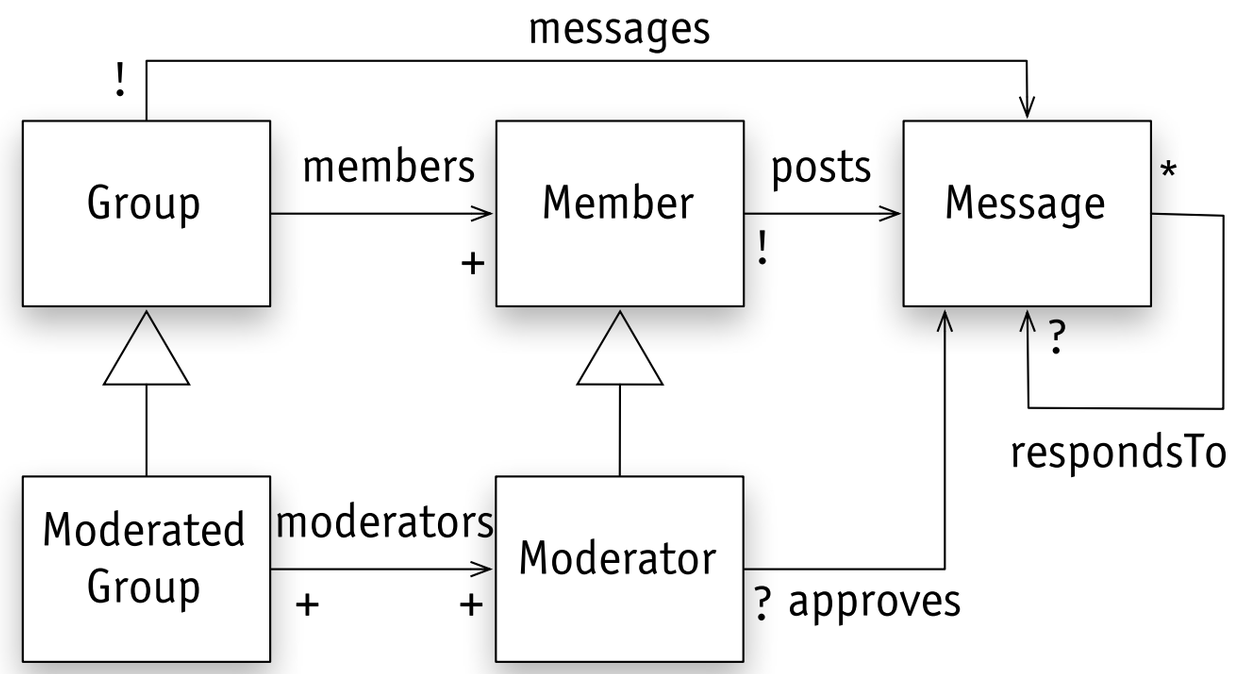
?

∈

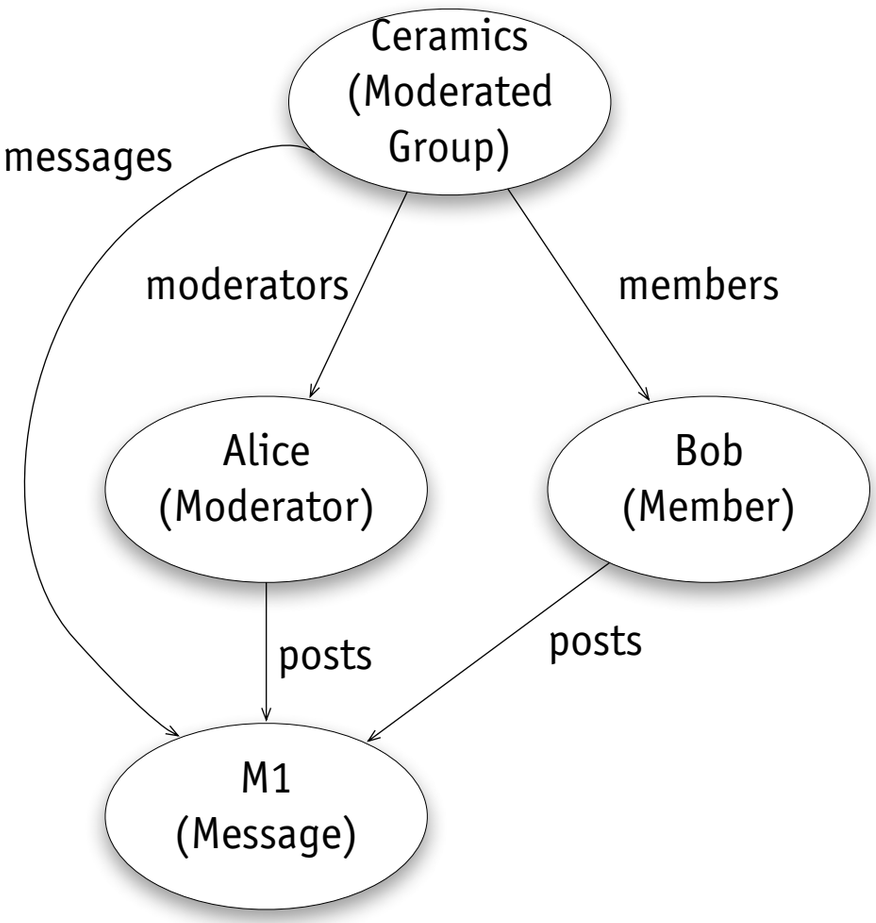


semantics examples



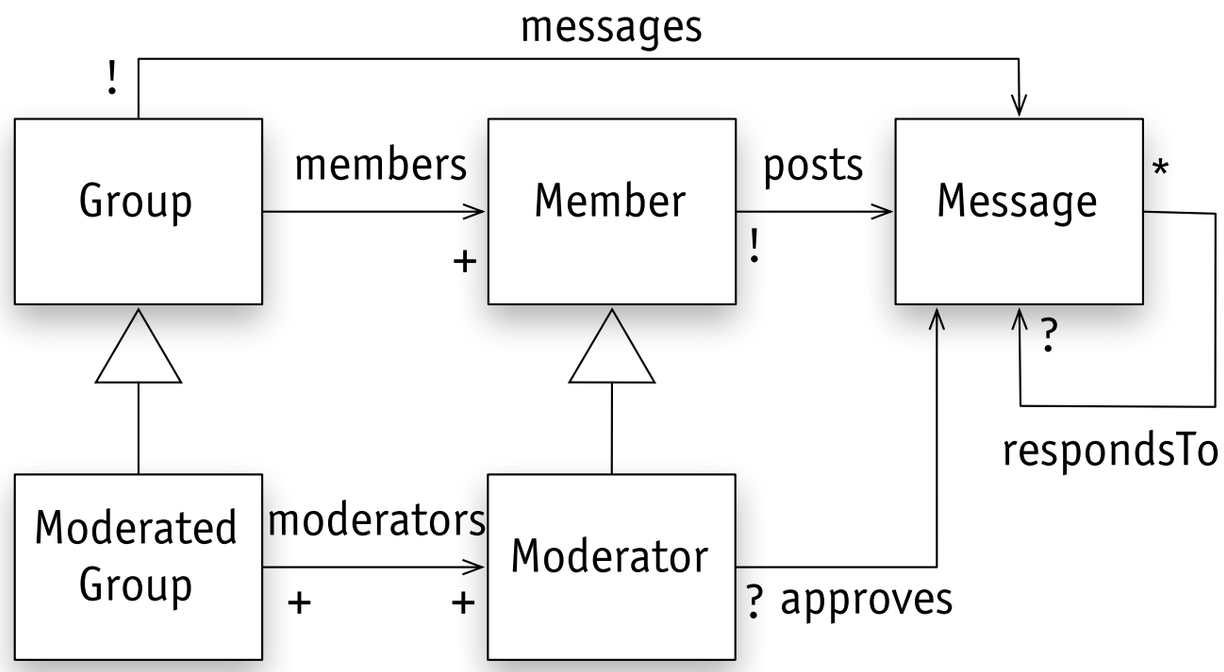


semantics examples

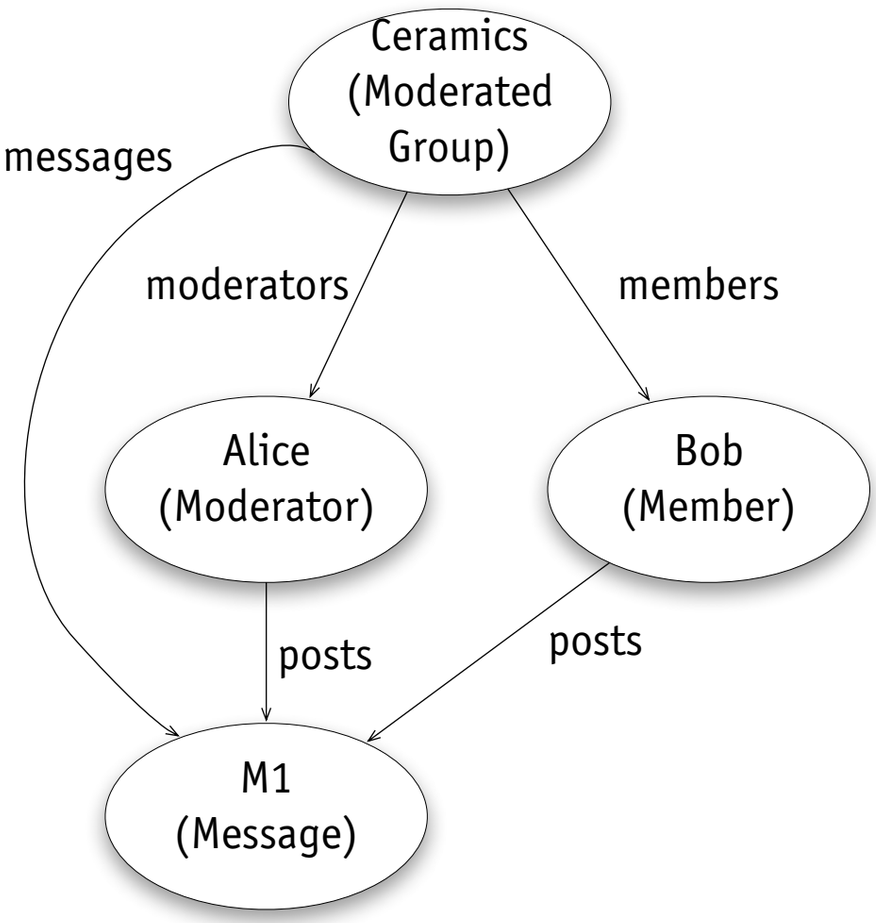


?

∈

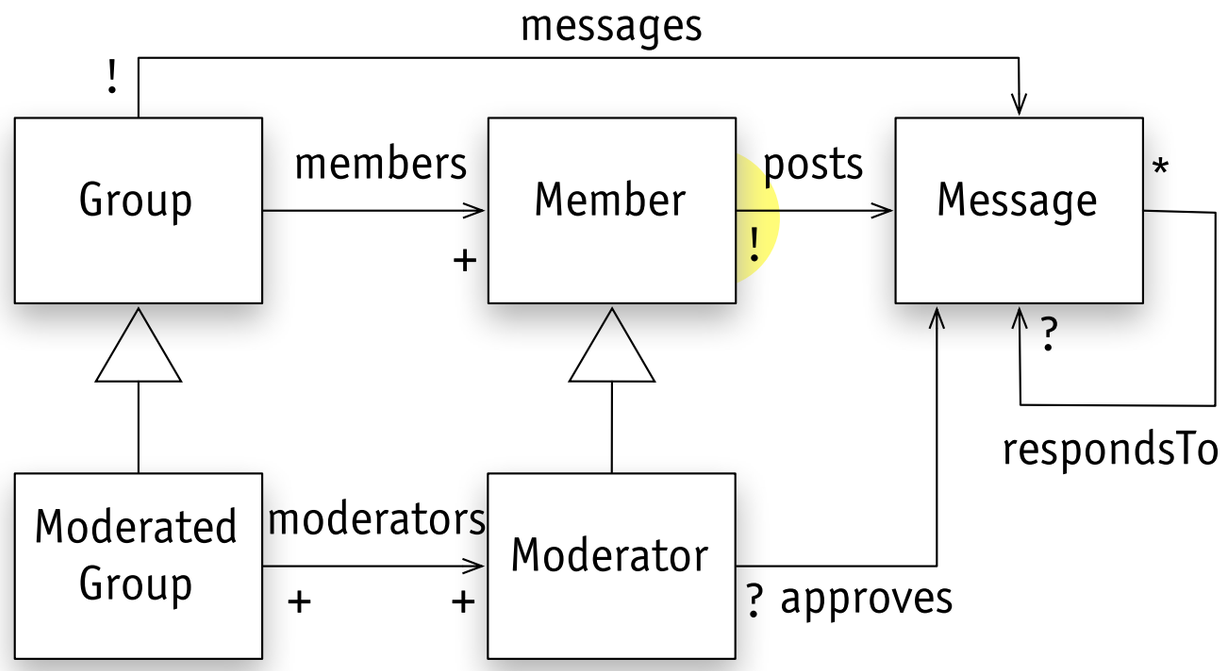


semantics examples

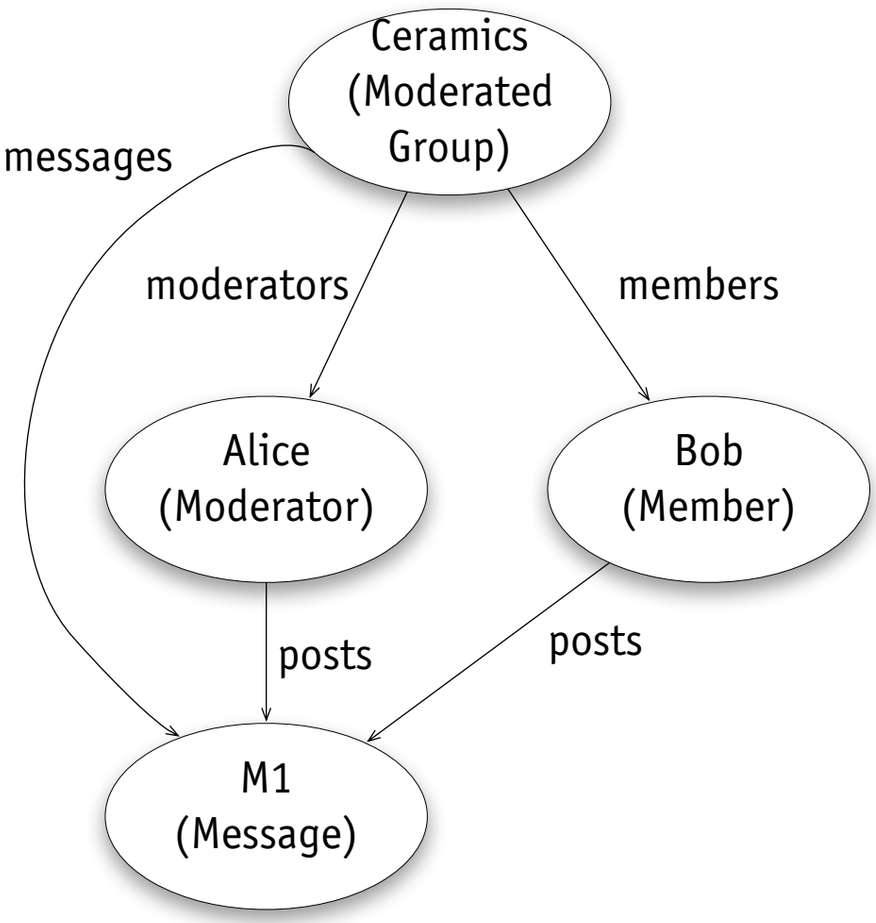


?

∈

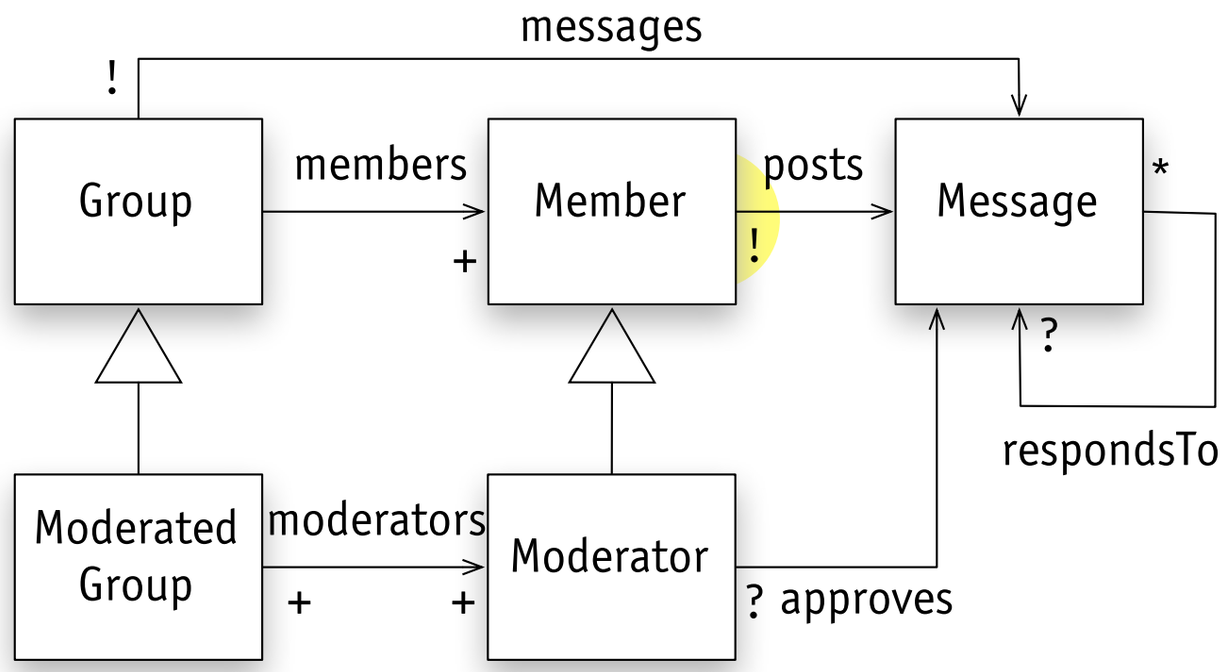


semantics examples

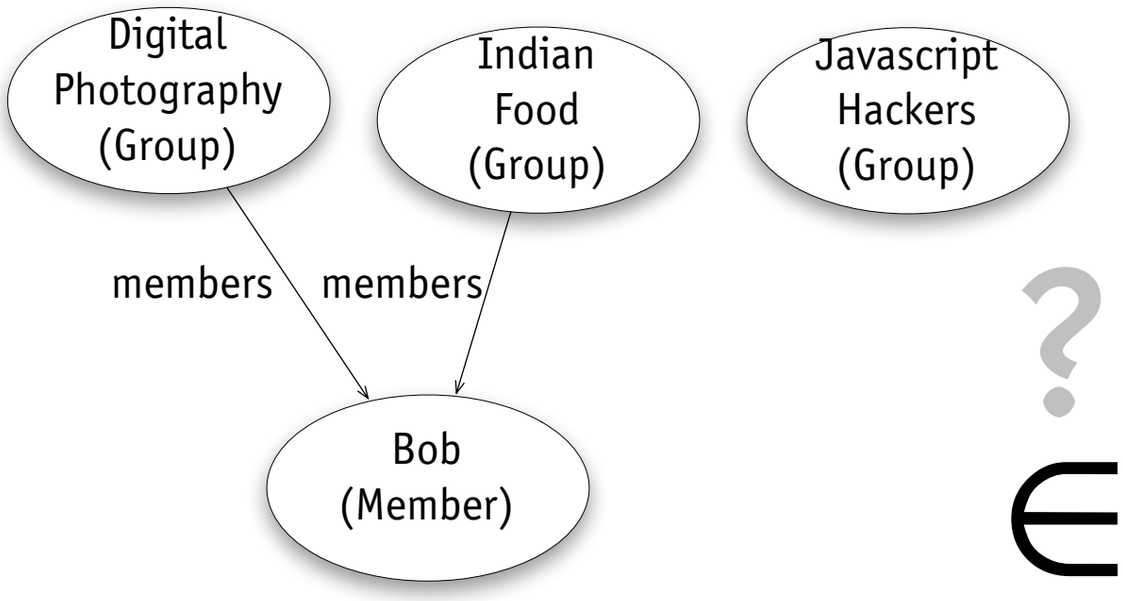


∈

✗

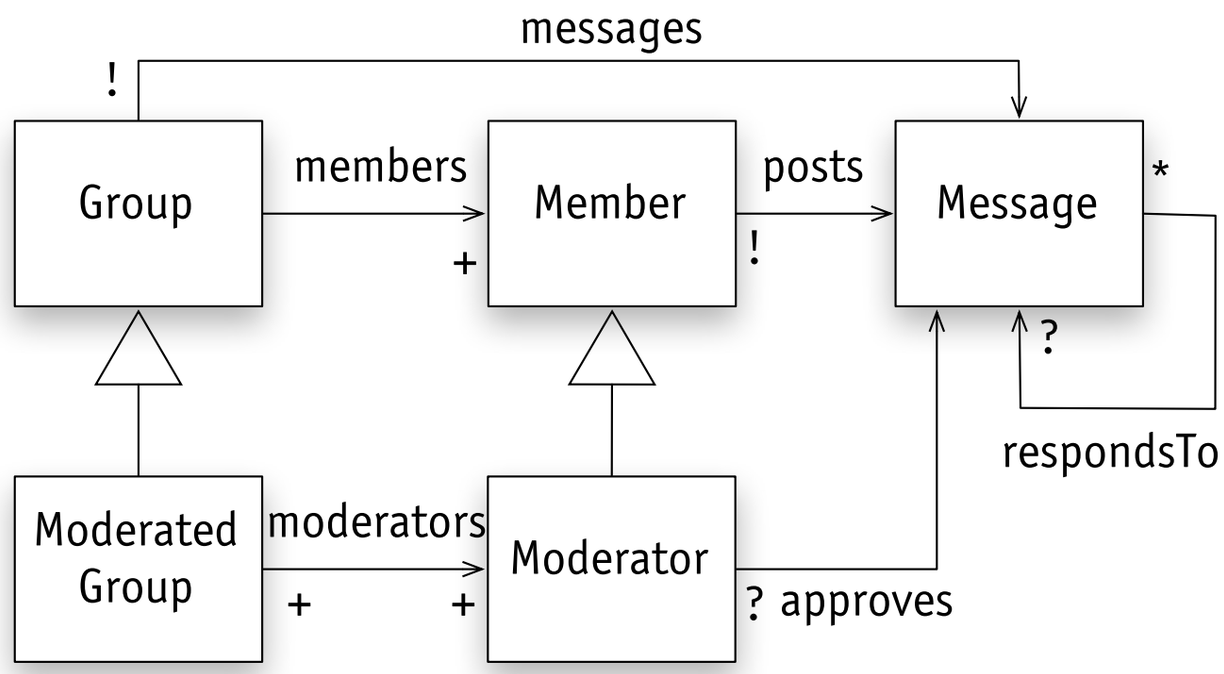


semantics examples

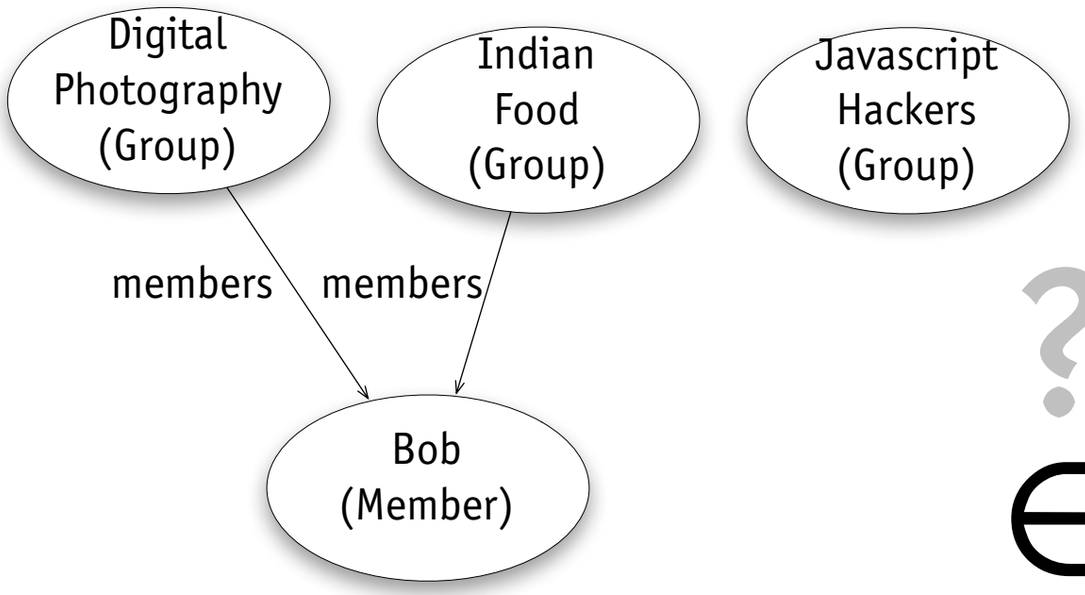


?

∈

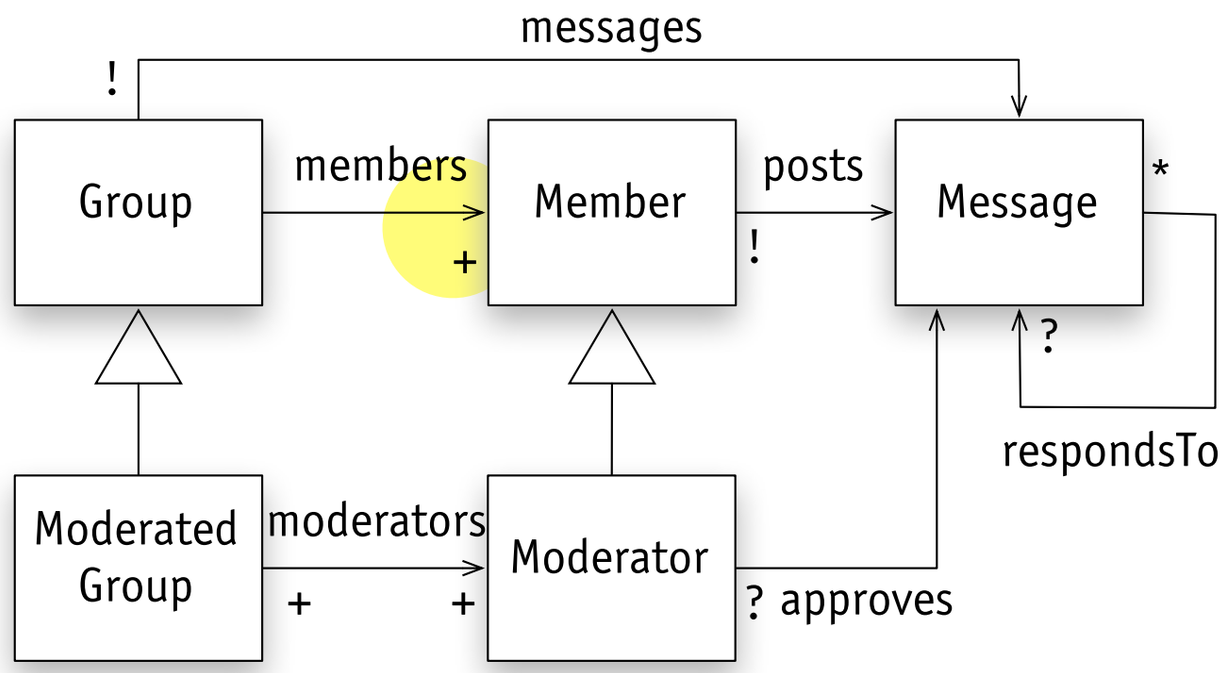


semantics examples

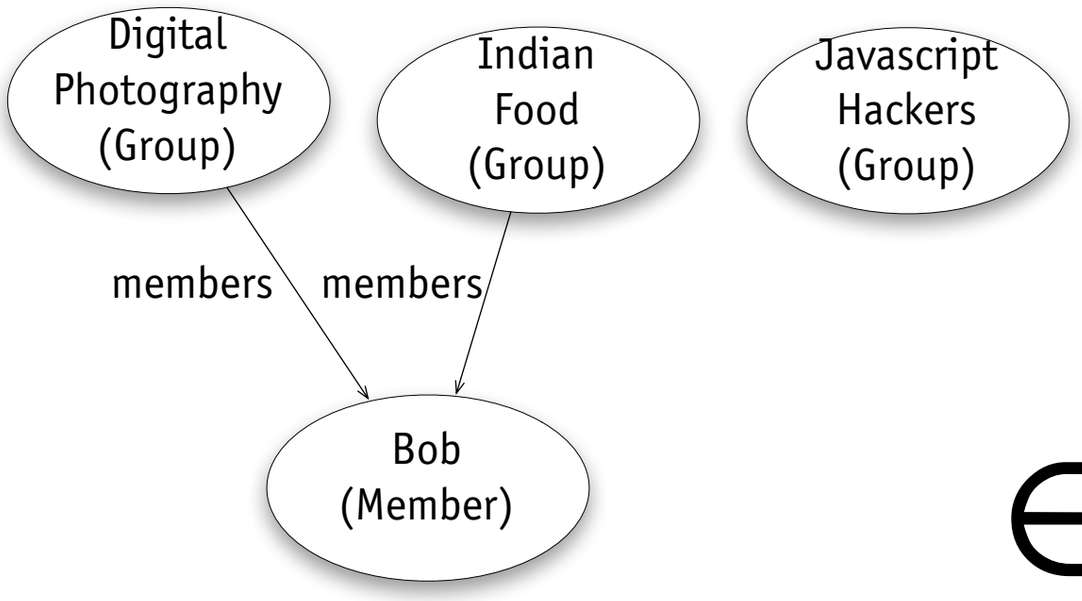


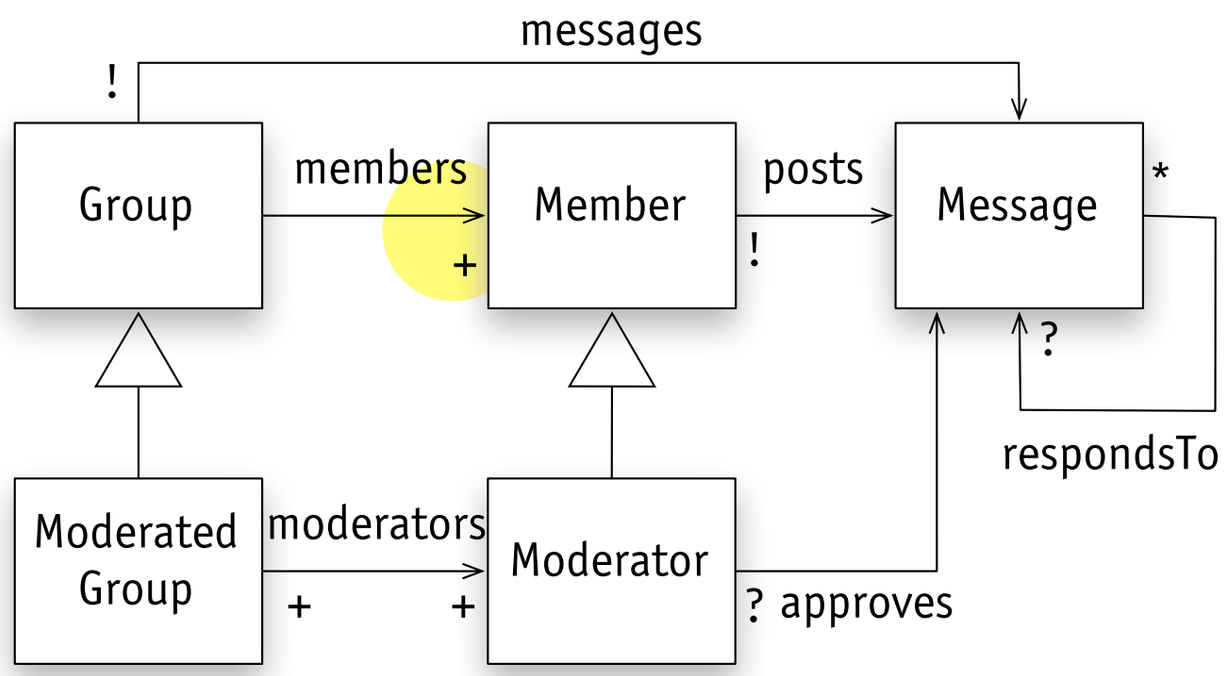
?

∈

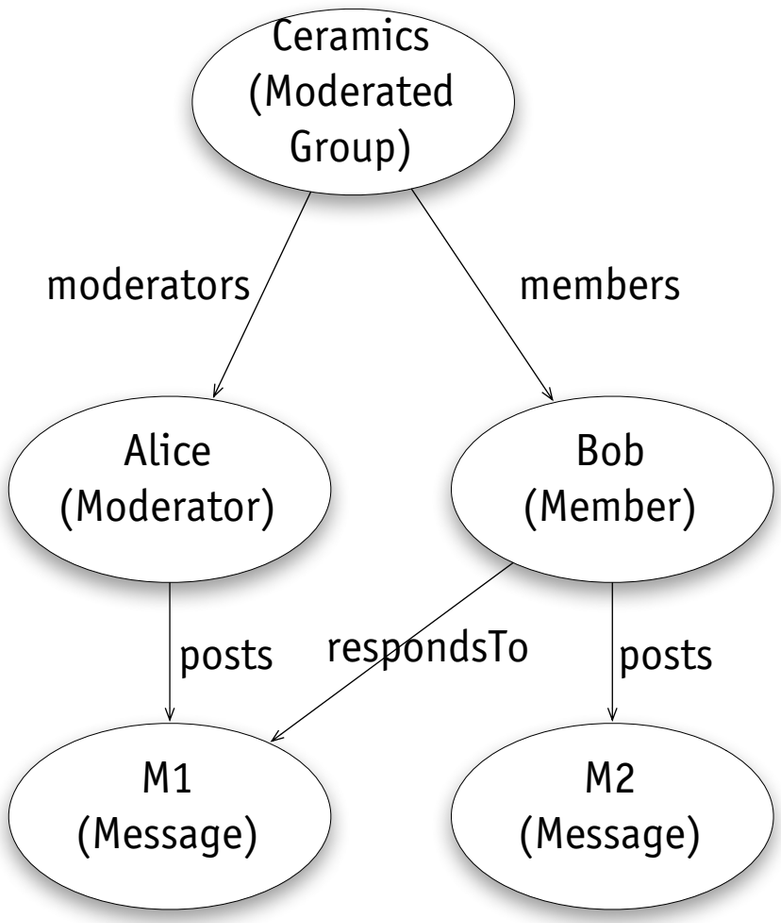


semantics examples



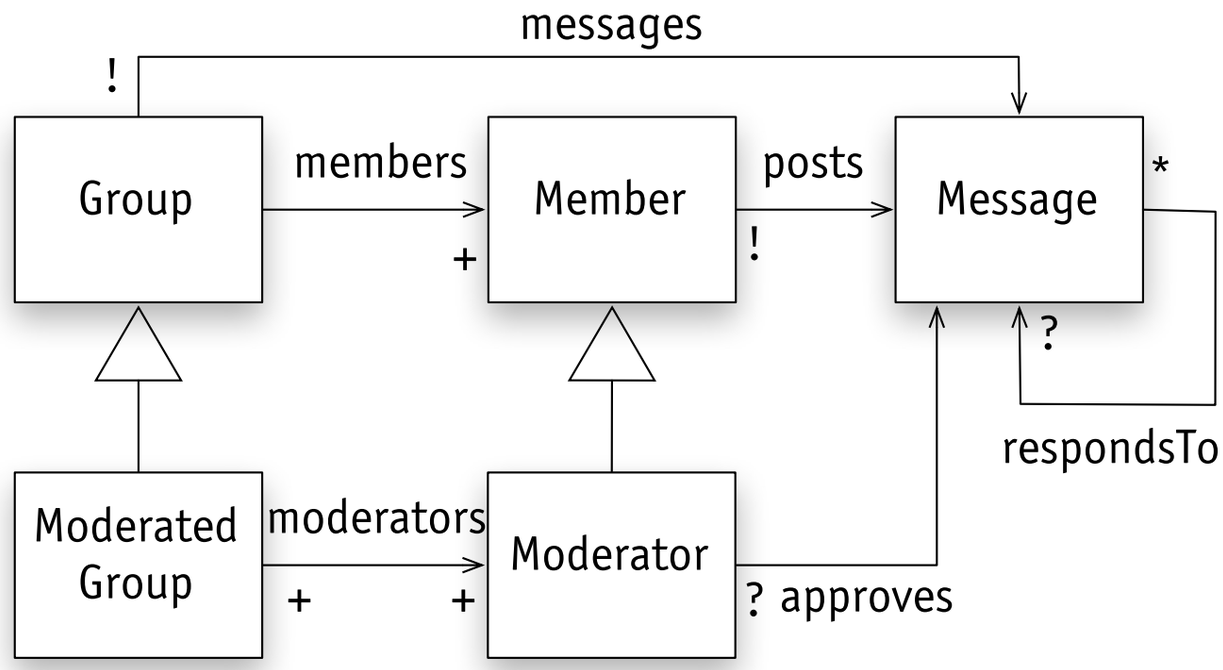


semantics examples

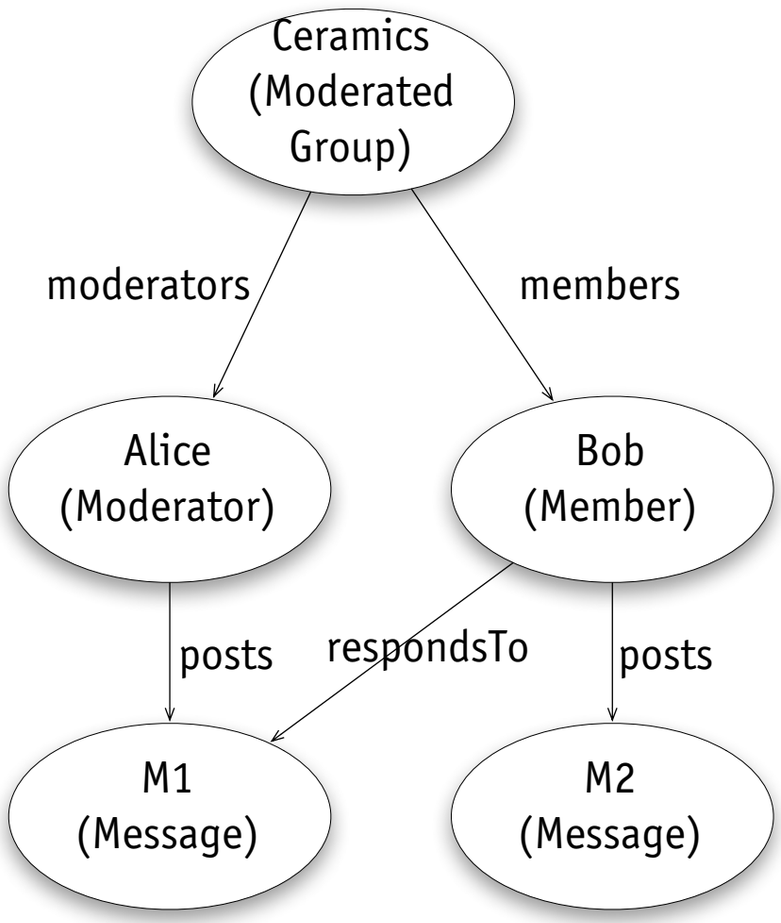


?

∈

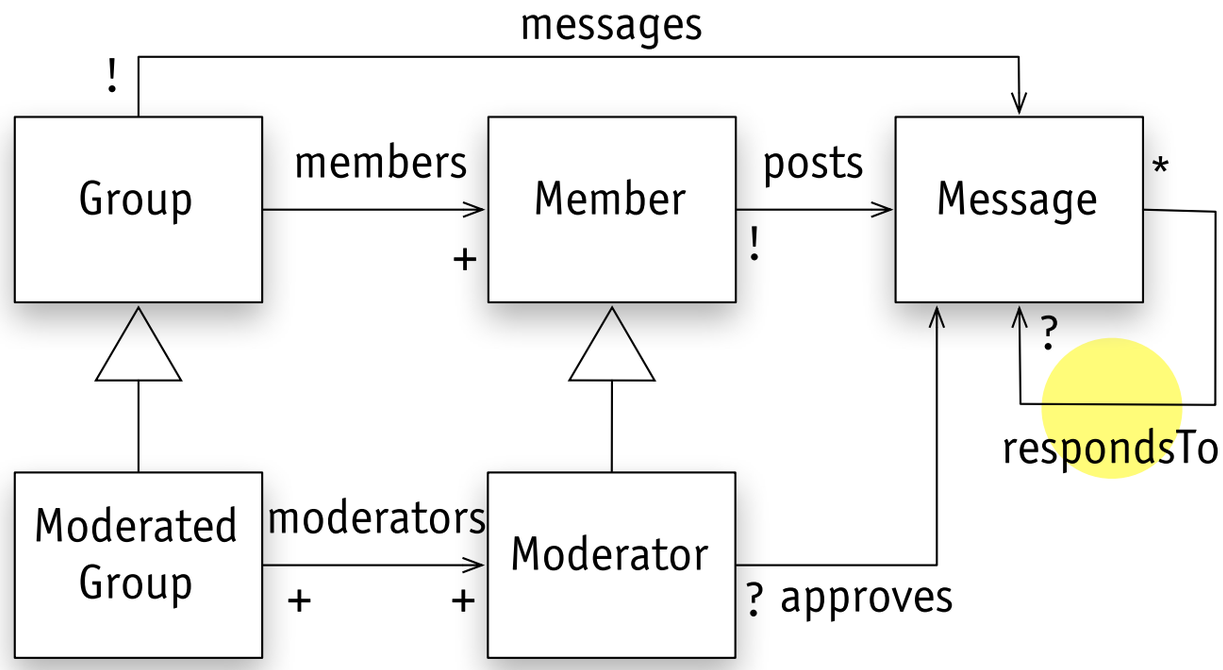


semantics examples

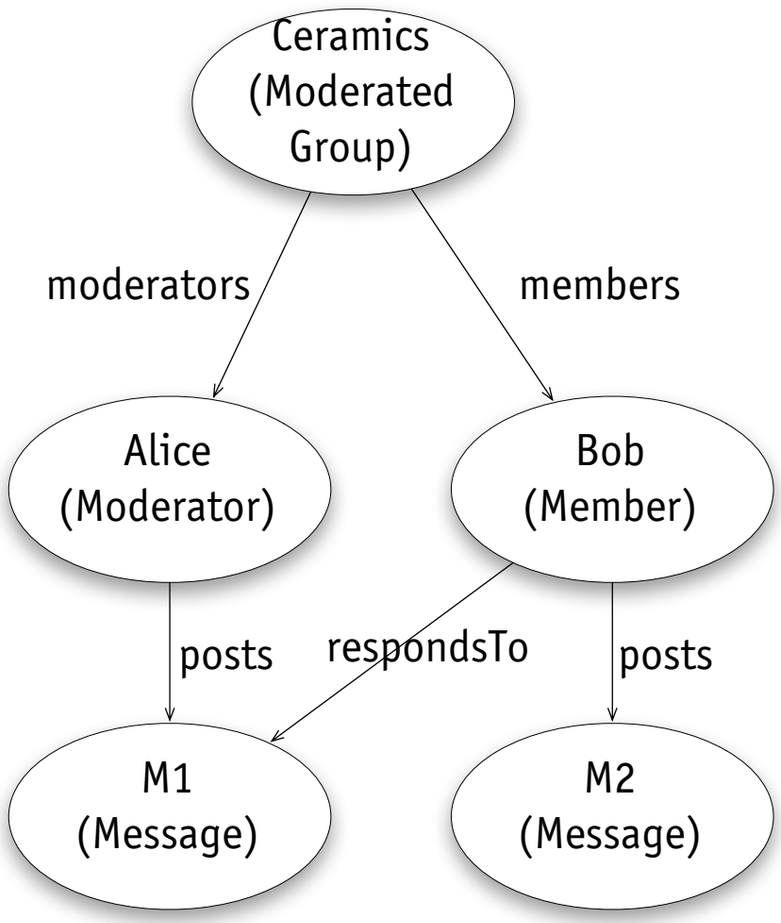


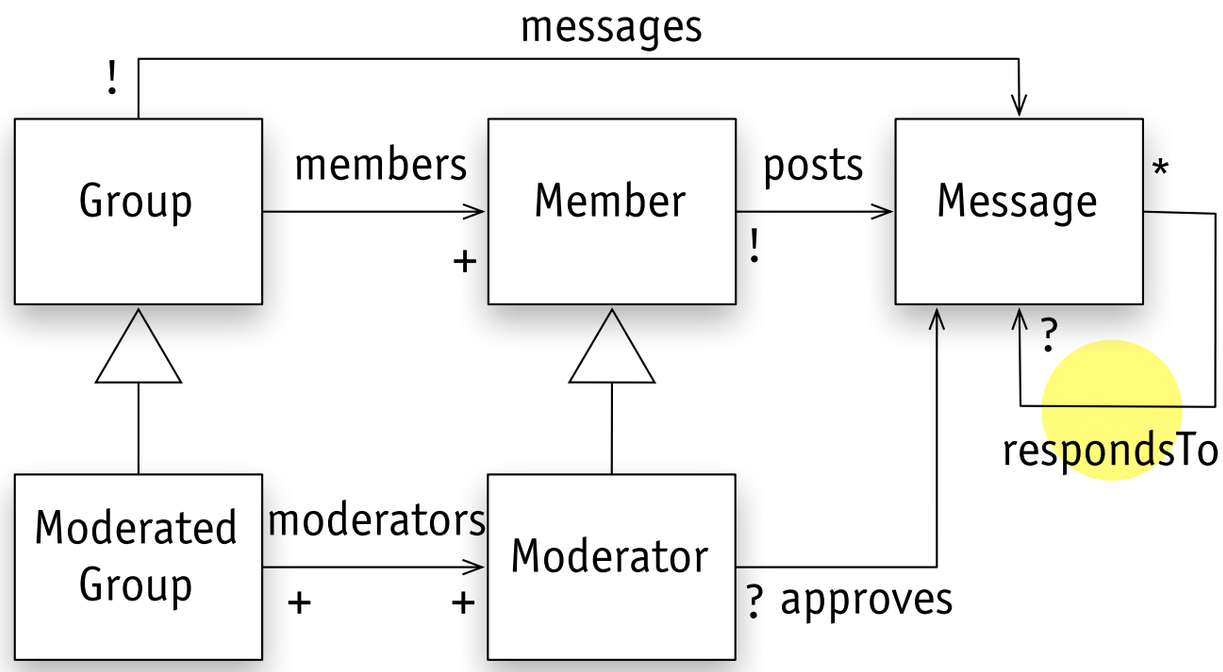
?

∈

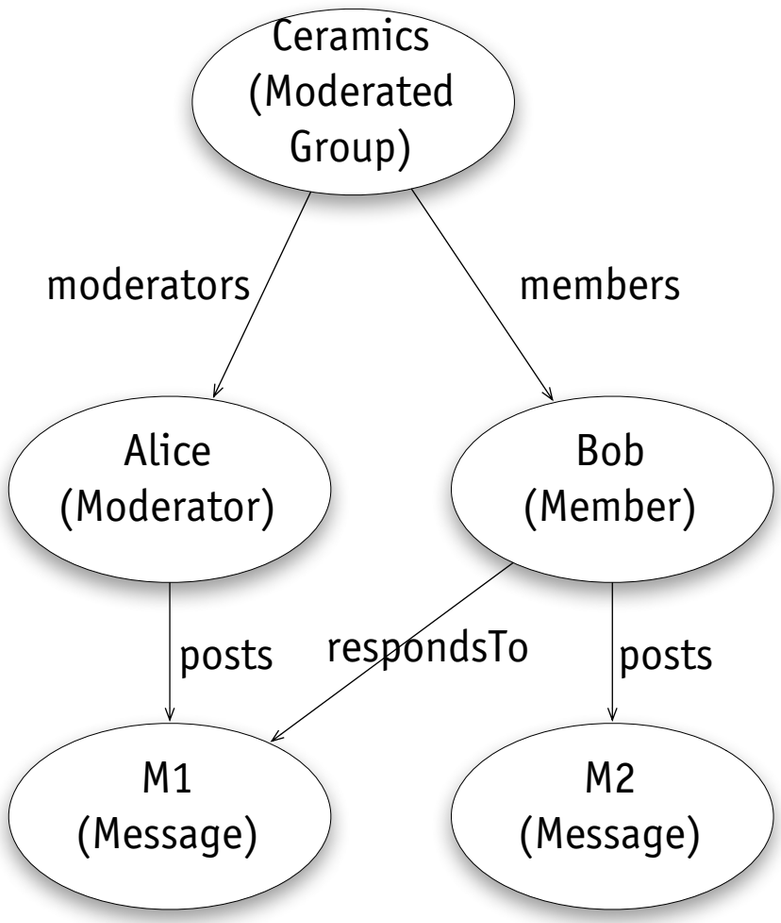


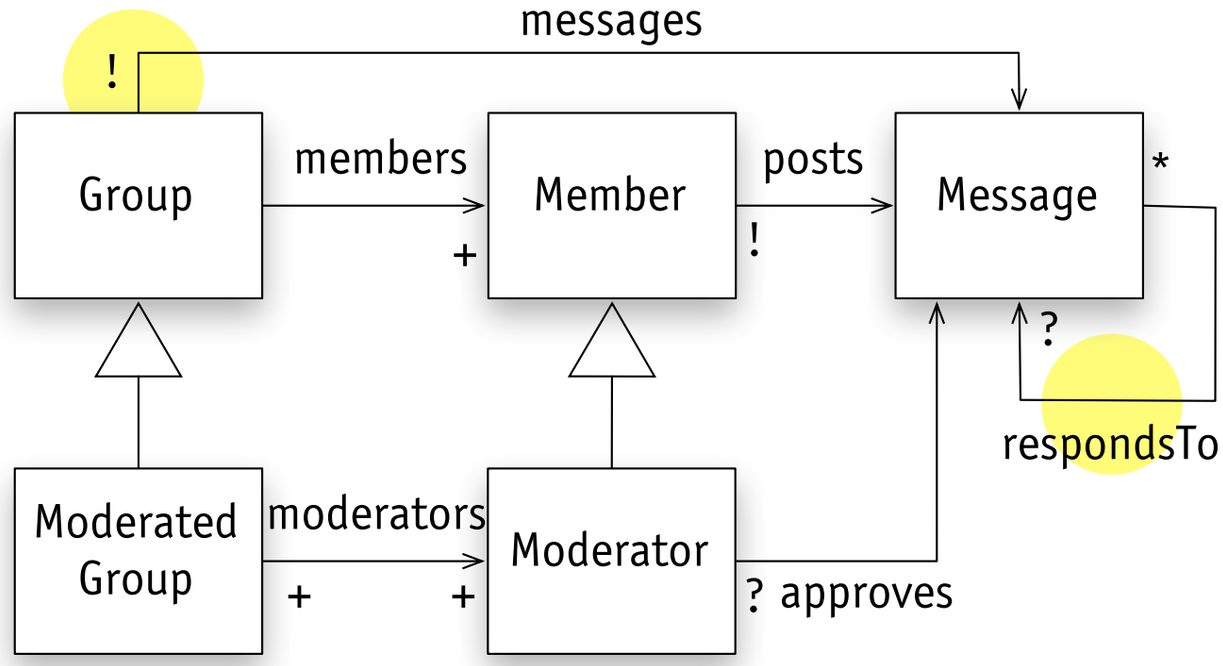
semantics examples





semantics examples





common mistakes

#1. designation confusion

arrivesAt: Elevator -> Floor

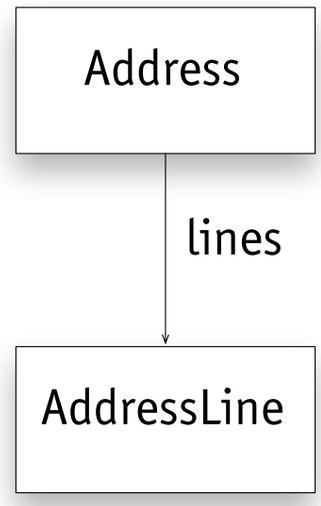
elevator serves floor?

is currently at floor?

will arrive at, or has arrived at floor?

#2. should be split into multiple relations

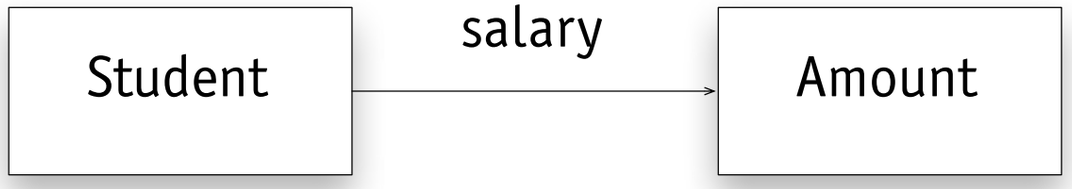
lines: Address -> AddressLine



#3. relates >2 atoms

salary: Student -> Amount

for which job?

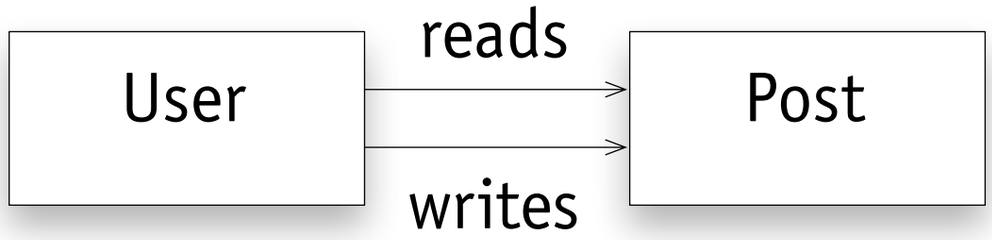


mistake: including operations

a bad smell

› using verbs, especially in present tense

bad:



better:

mistake: inadequacy

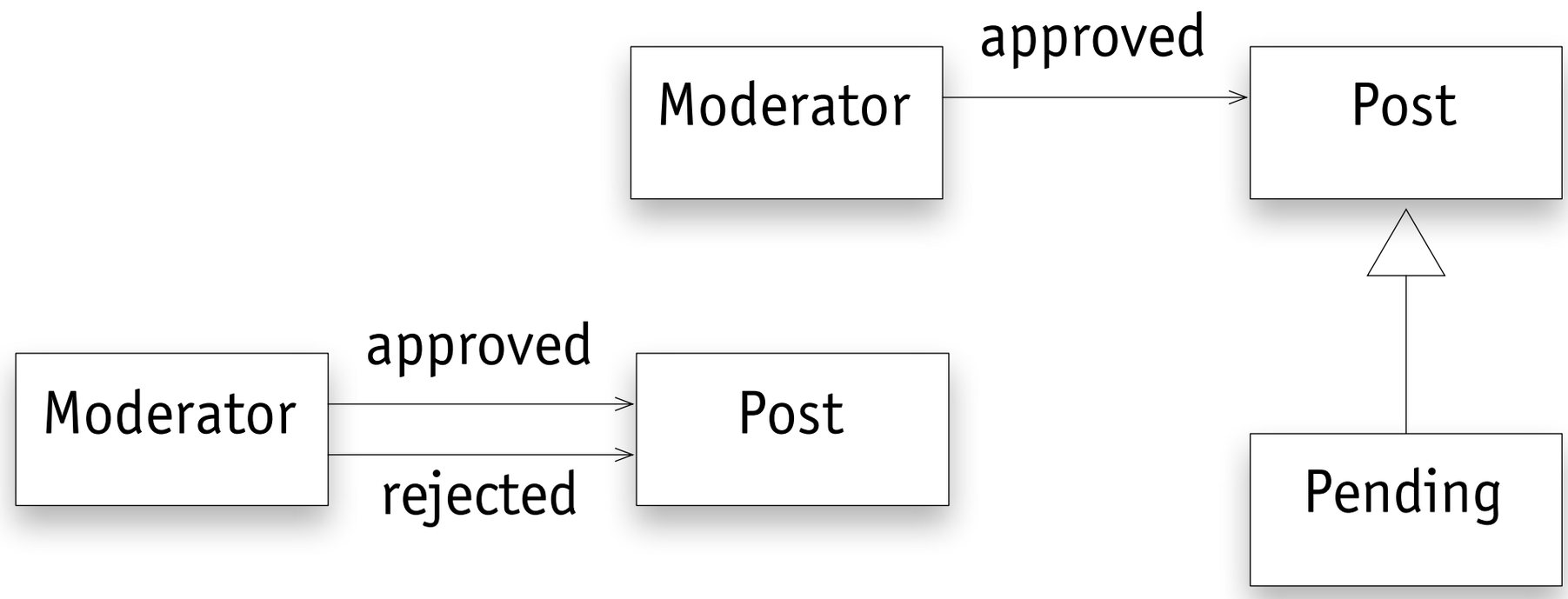
think about each feature

› is there enough state to answer each query?

› to decide which actions are allowed?

bad:

better:

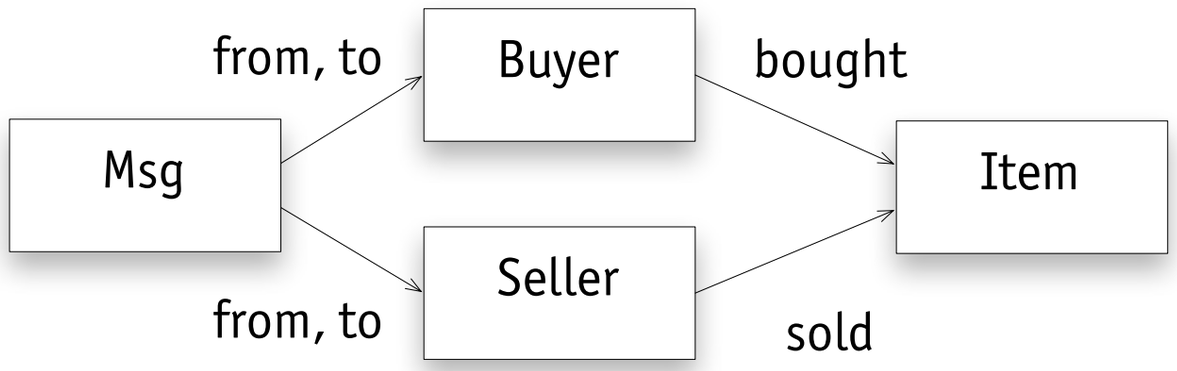


mistake: lack of generalization

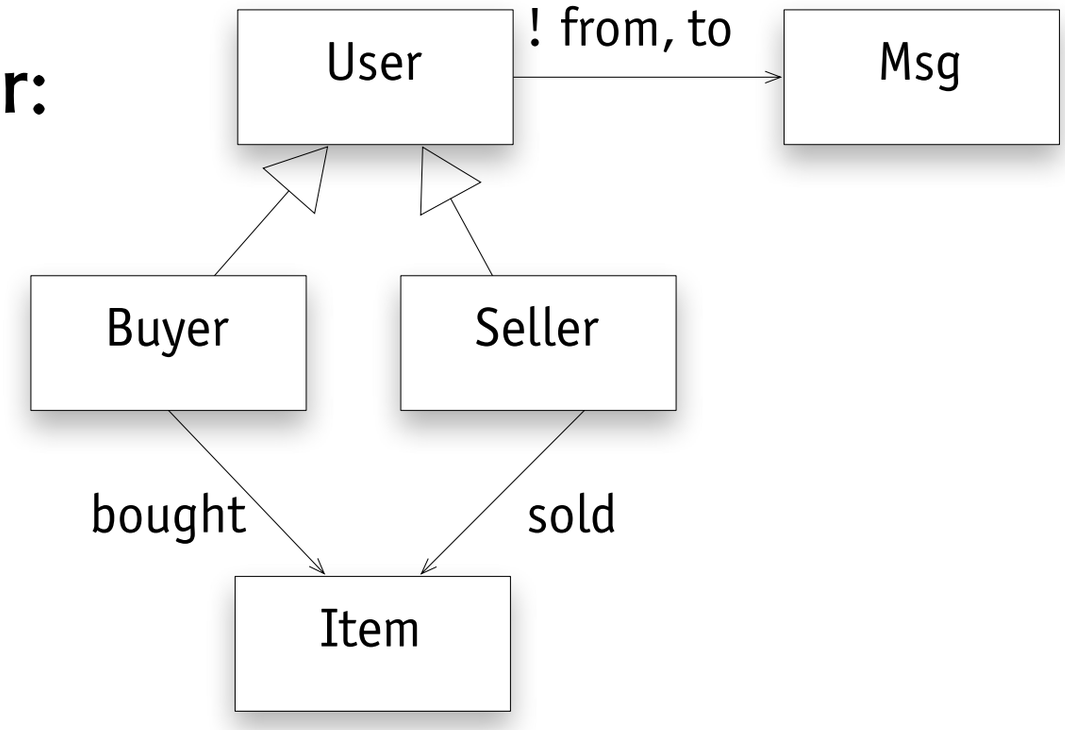
are two sets really subsets of a common set?

› look for duplication of relations

bad:



better:

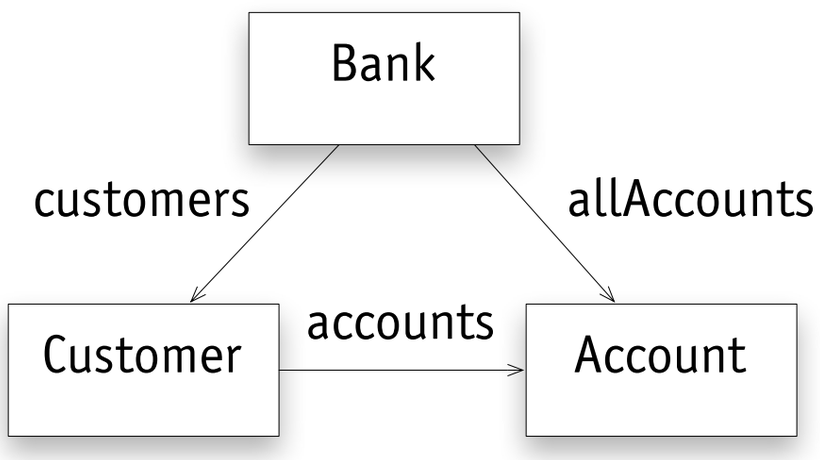


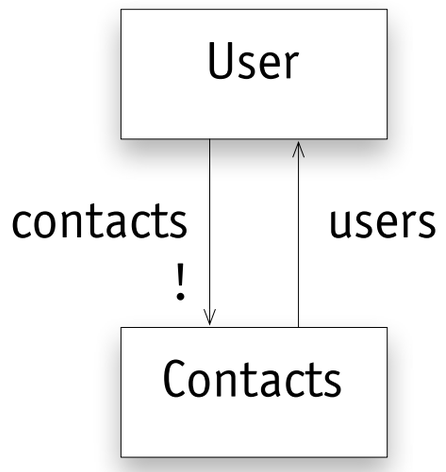
mistake: implementation details

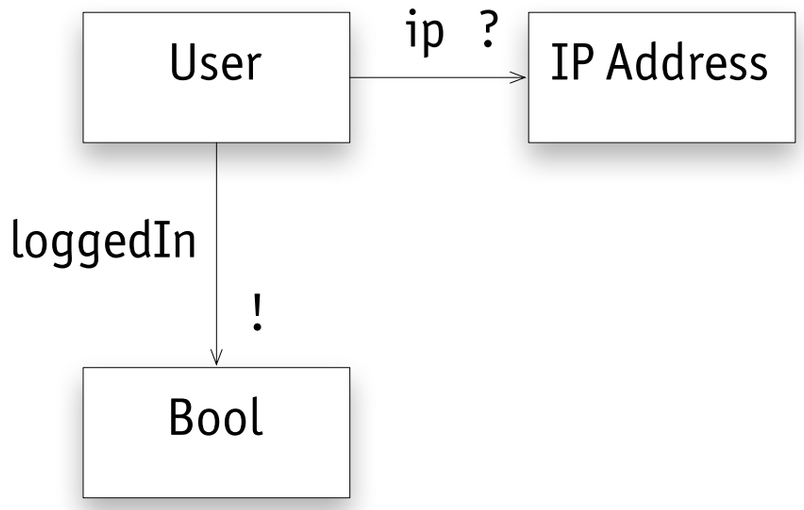
beware of

- collection objects: replace by relation?
- singleton objects: replace by set?
- status: replace by subsets?

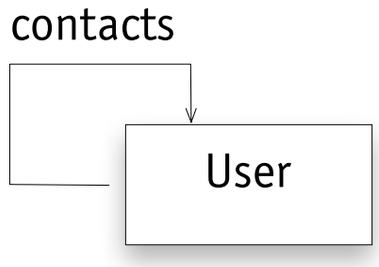
bad:

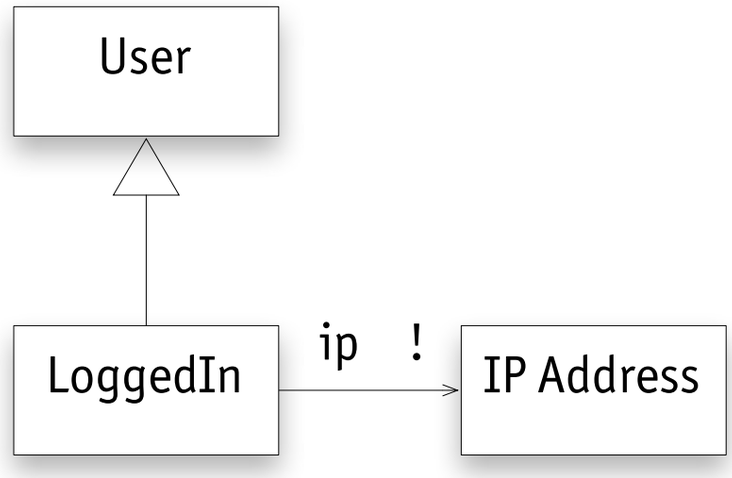




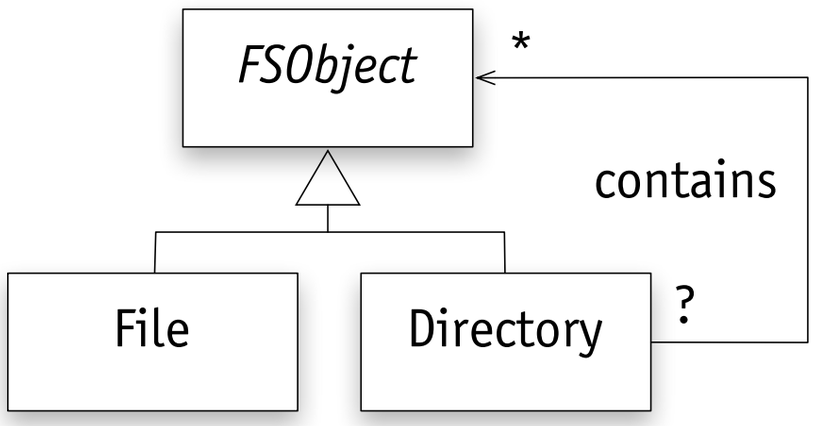


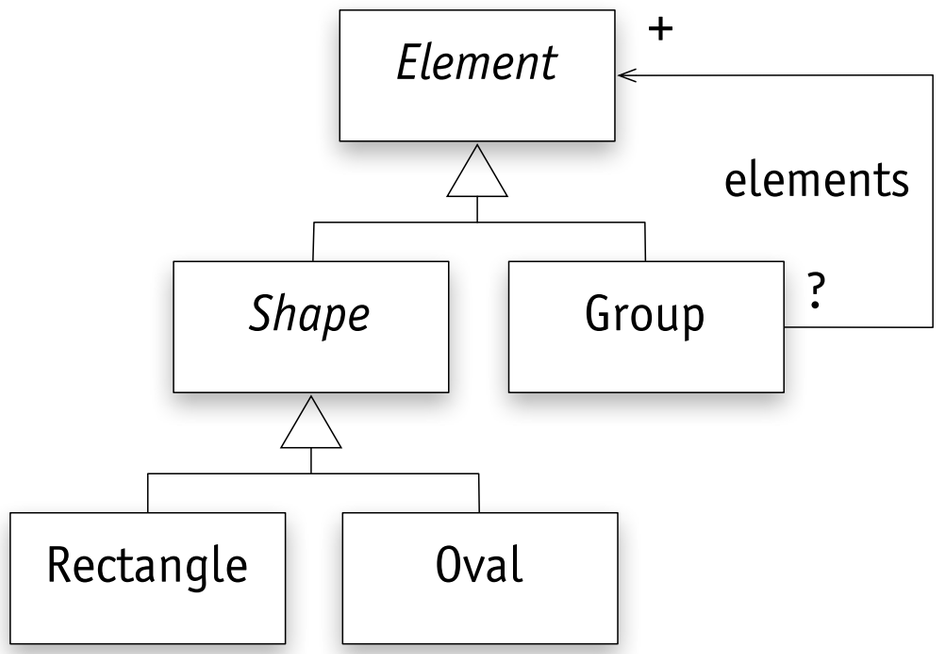
better:

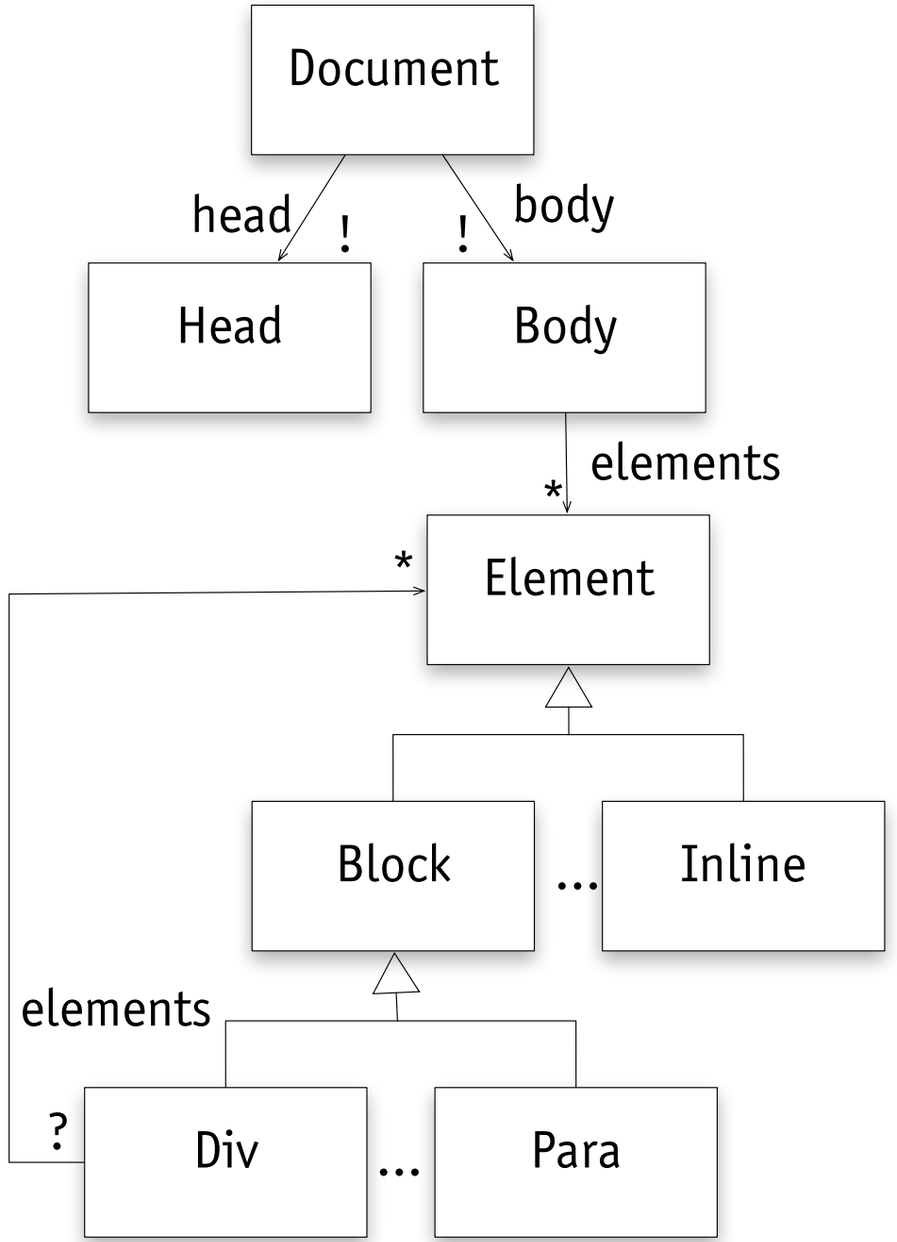




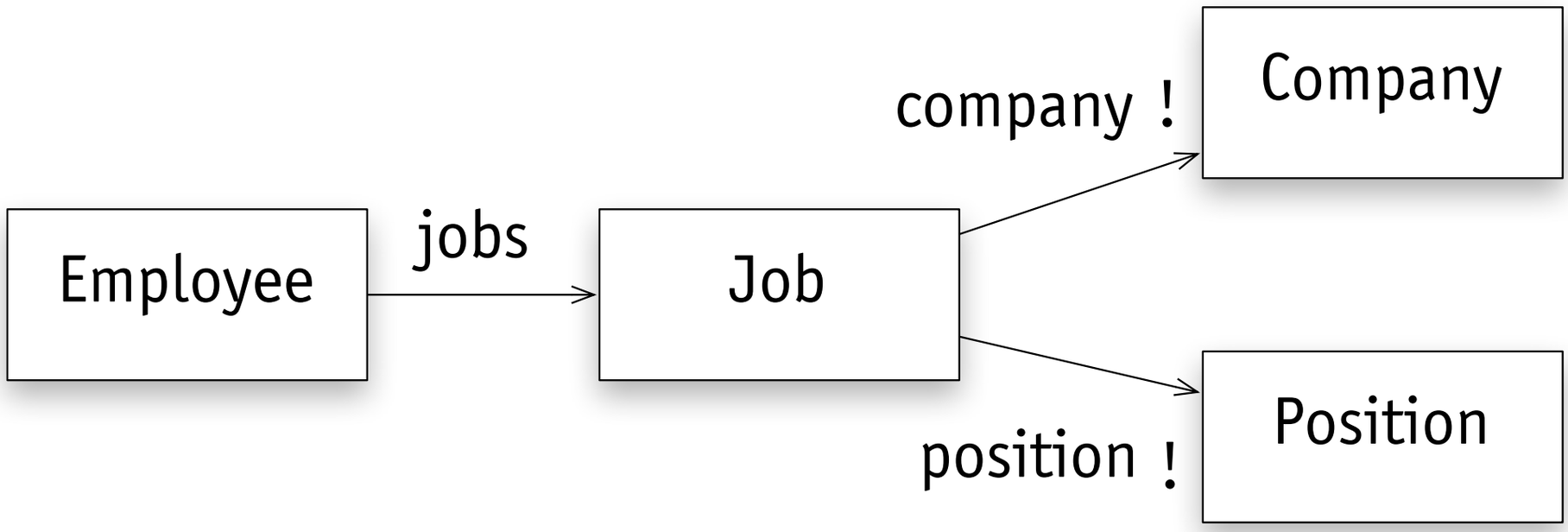
composite


FSObject
*
contains
File
Directory
?


Element
+
elements
Shape
Group
?
Rectangle
Oval


Document
head
!
!
body
Head
Body
elements
*
*
Element
Block
...
Inline
elements
?
Div
...
Para

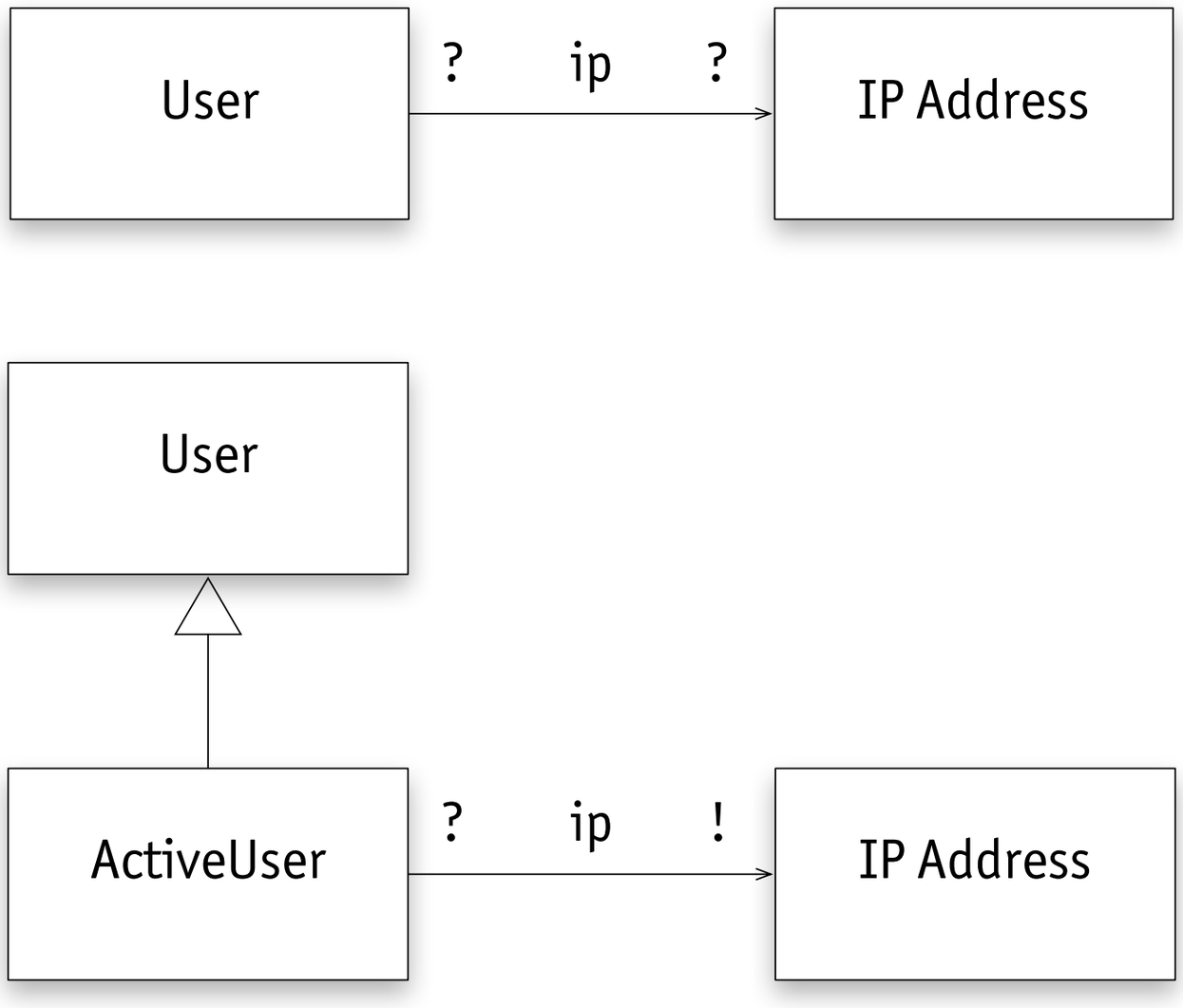
tuple



employee-company-position

› a 3-way relationship, expressed with Job tuple

lowering



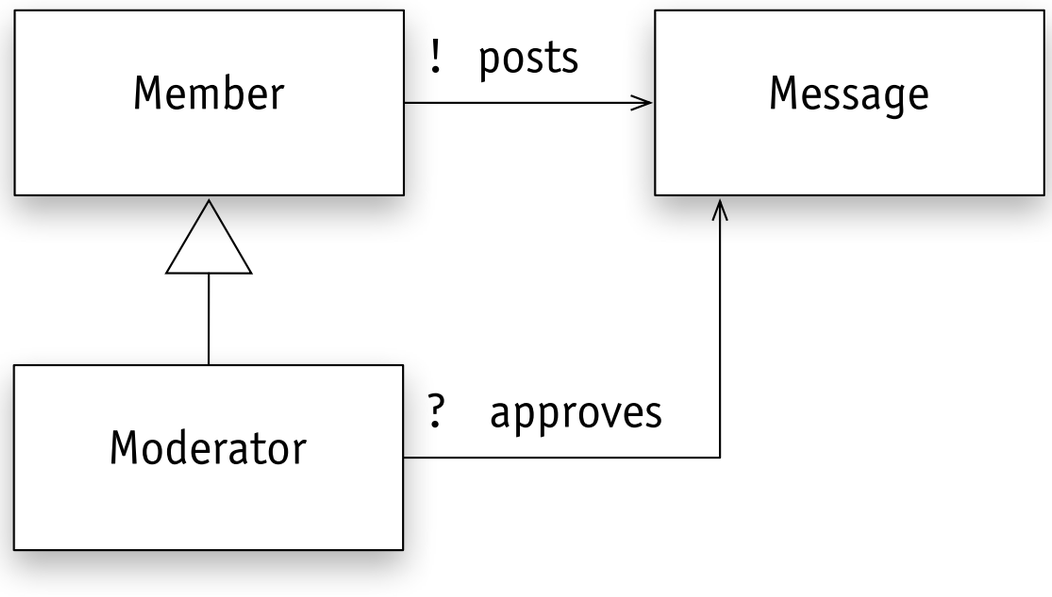
ok

better

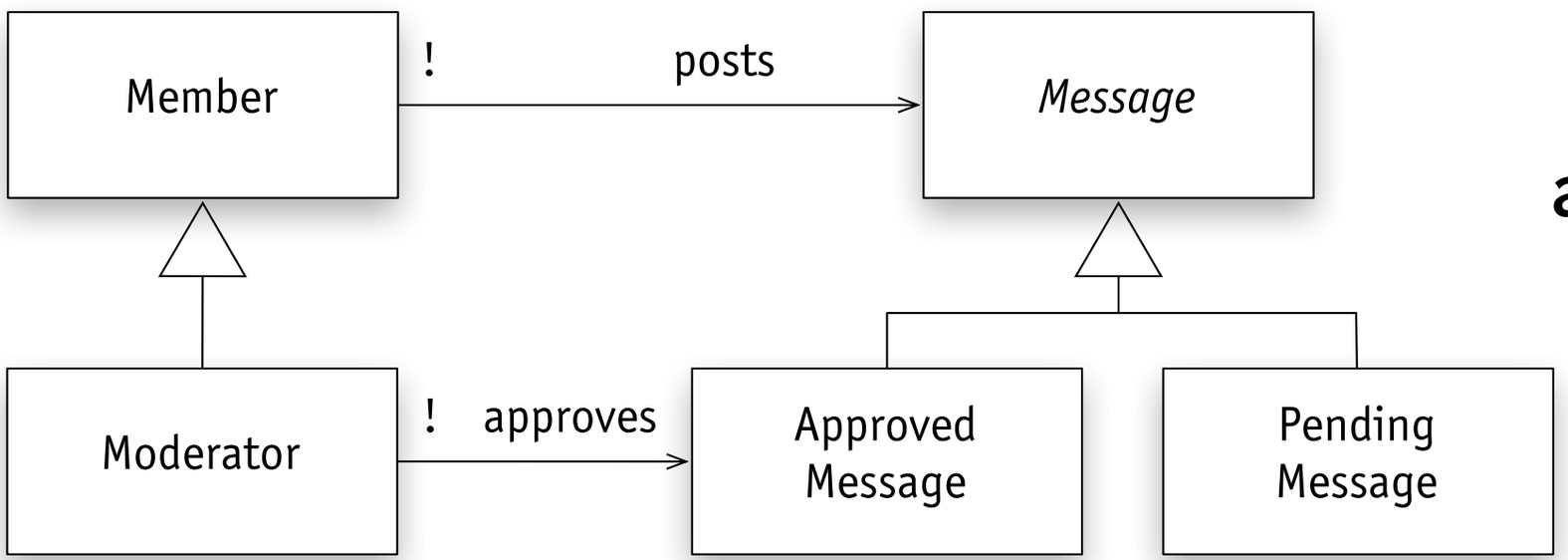
users with IP addresses

- strengthen multiplicity
- define subset of active users that have IPs

another lowering

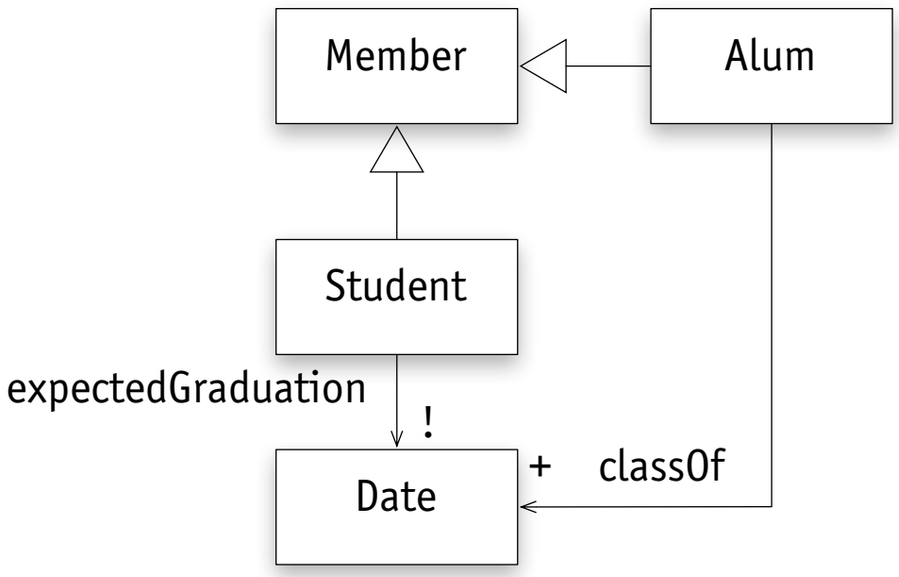


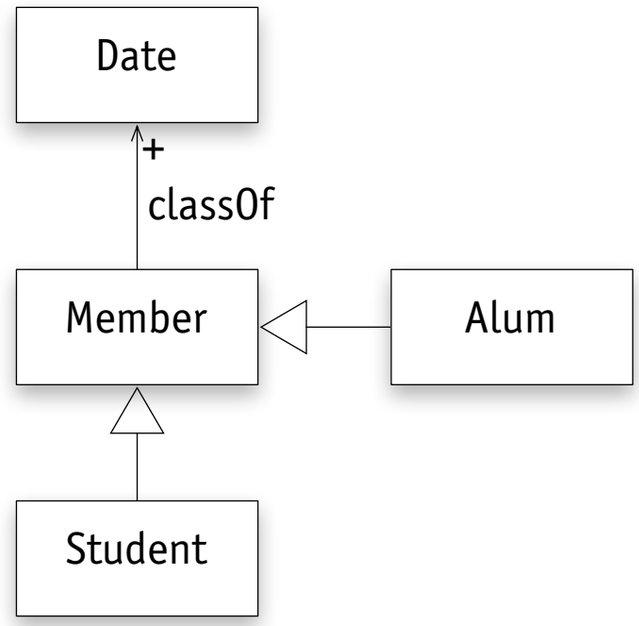
before

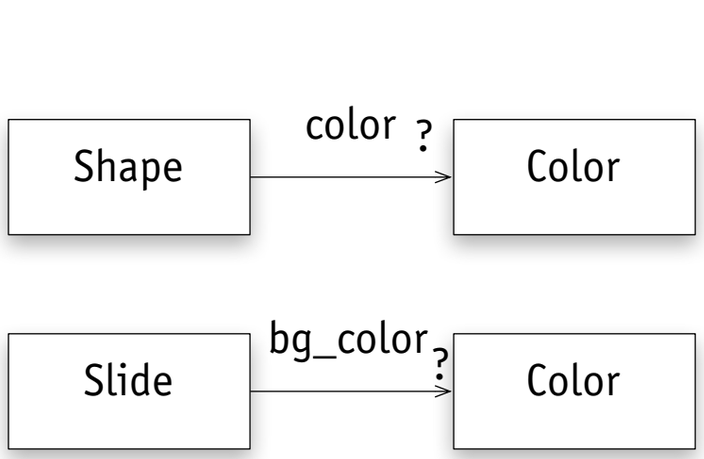


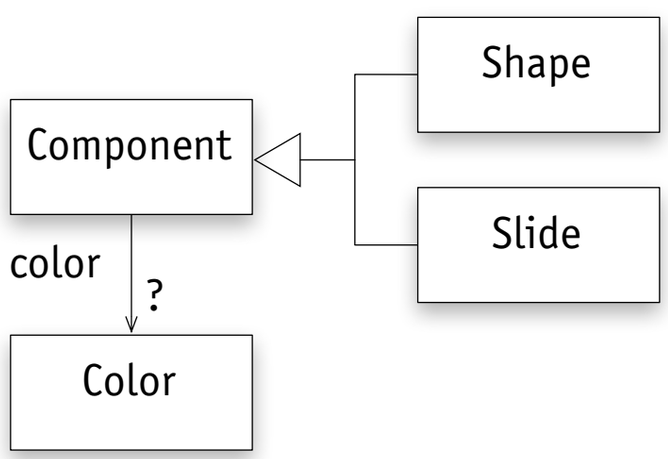
after

lifting





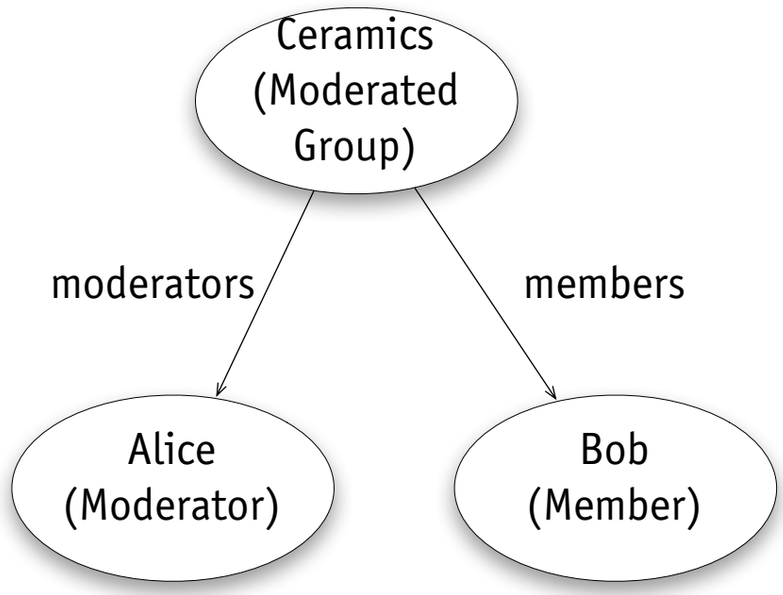


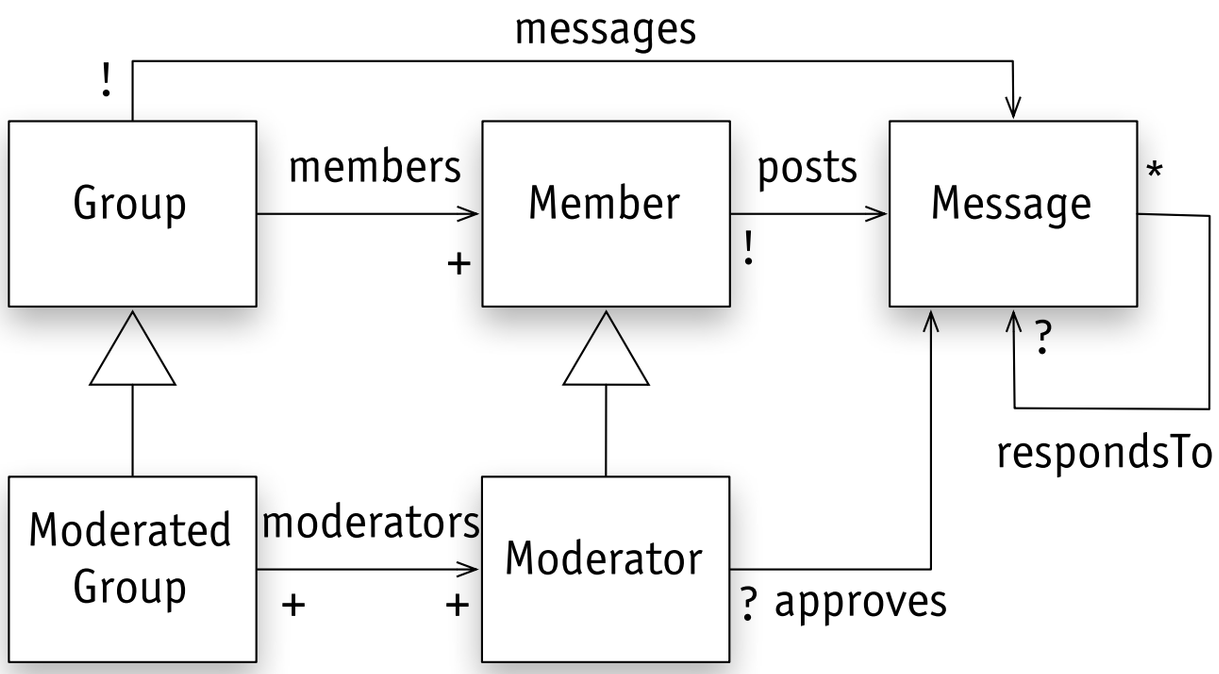


generalize and move relation to larger set

- from Alum/Student to Member
- from Shape/Slide to Component

graphical limitations

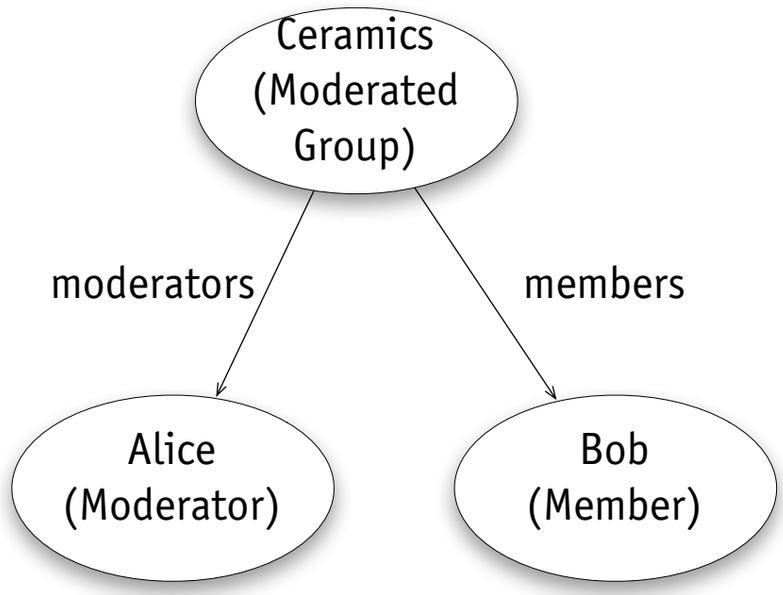


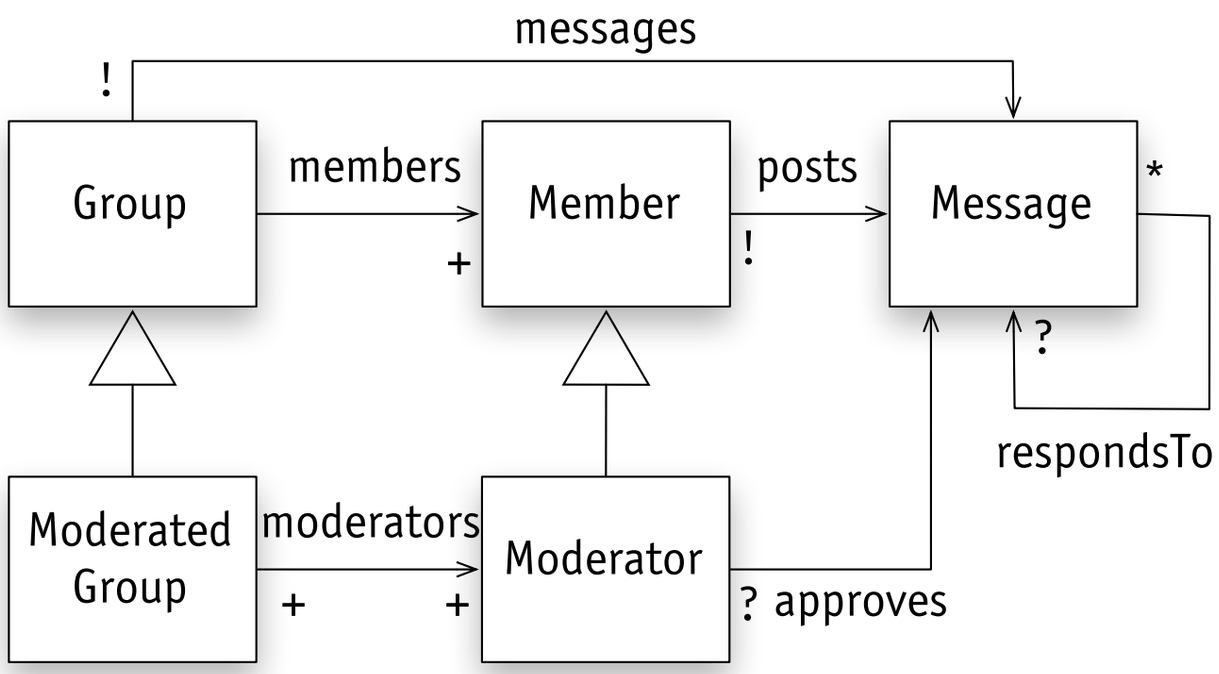


graphical limitations

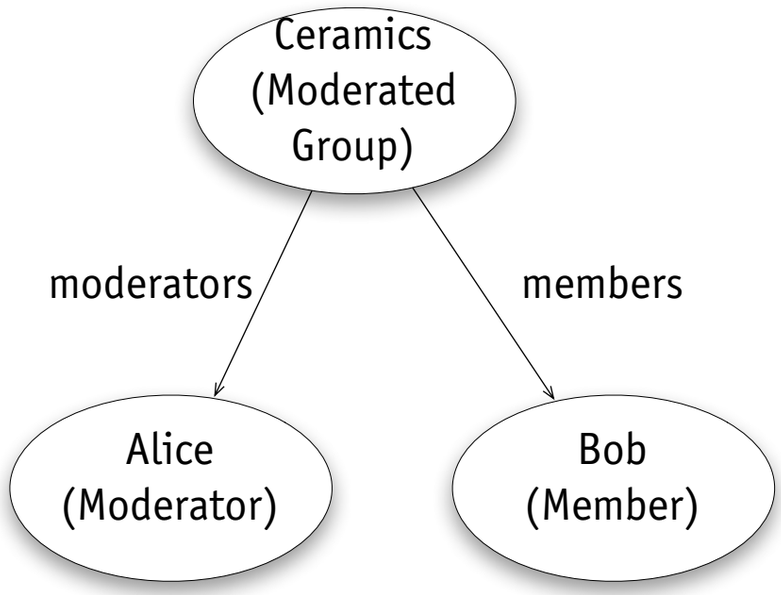
consider this instance

› Alice is not a member of Ceramics





graphical limitations

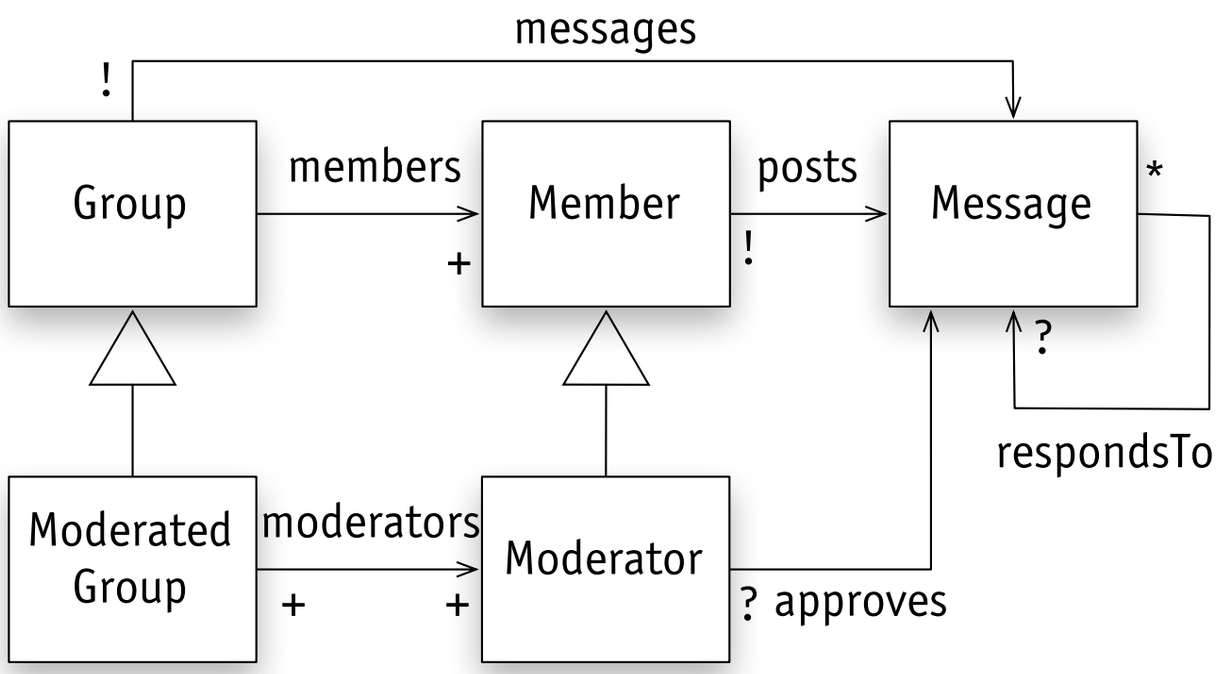


consider this instance

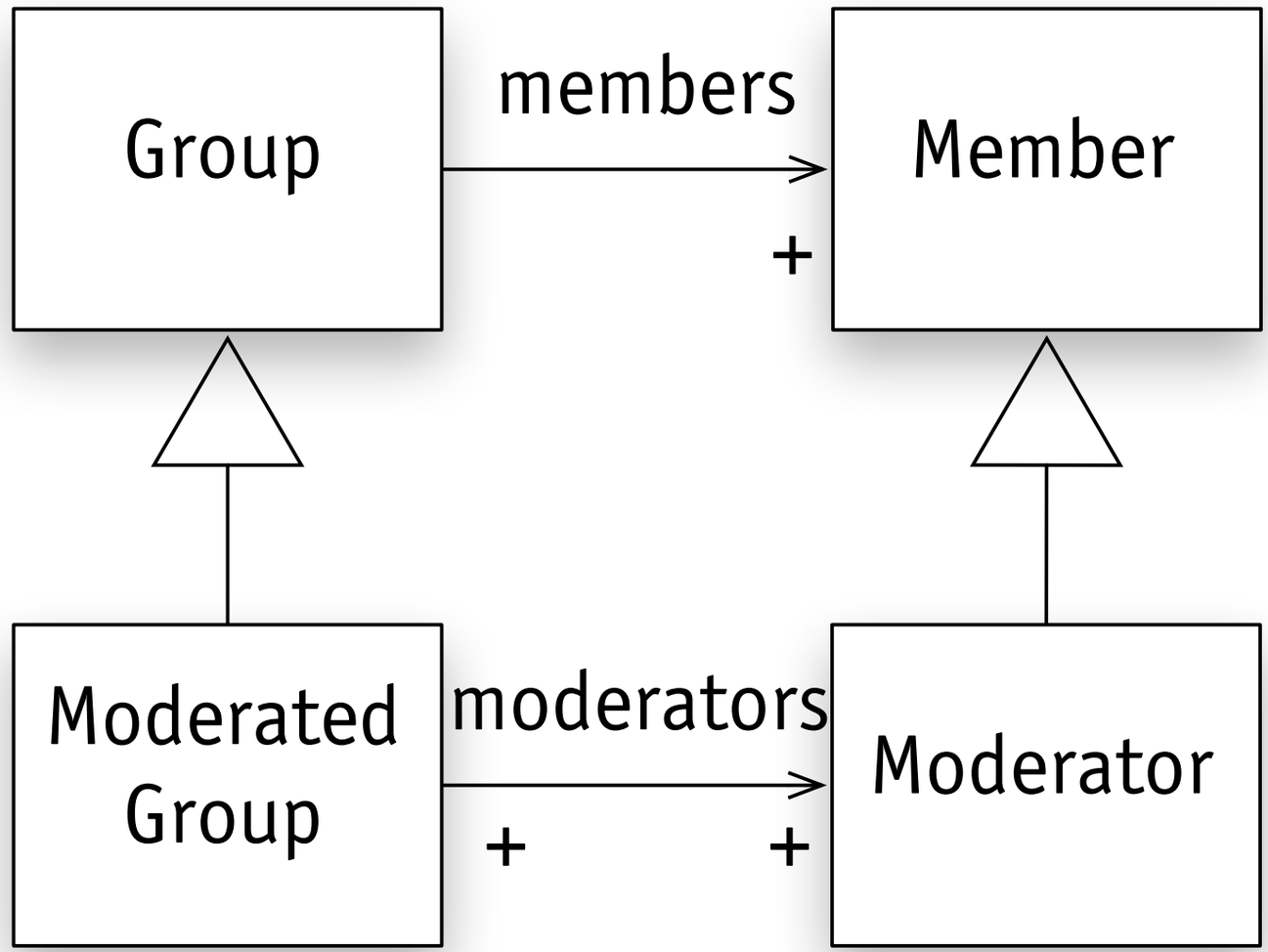
› Alice is not a member of Ceramics

can we say moderators should be members?

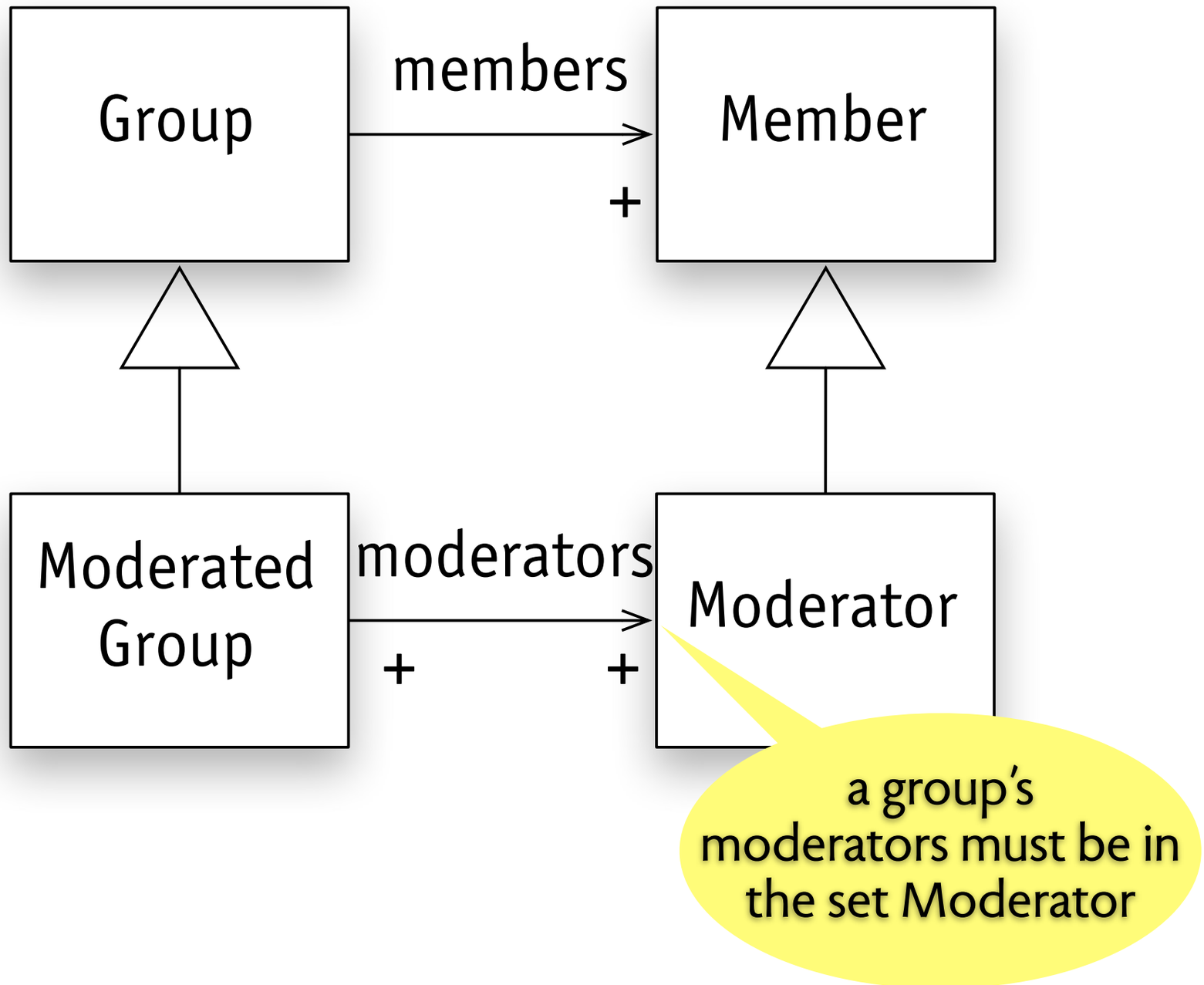
› not in this graphical notation



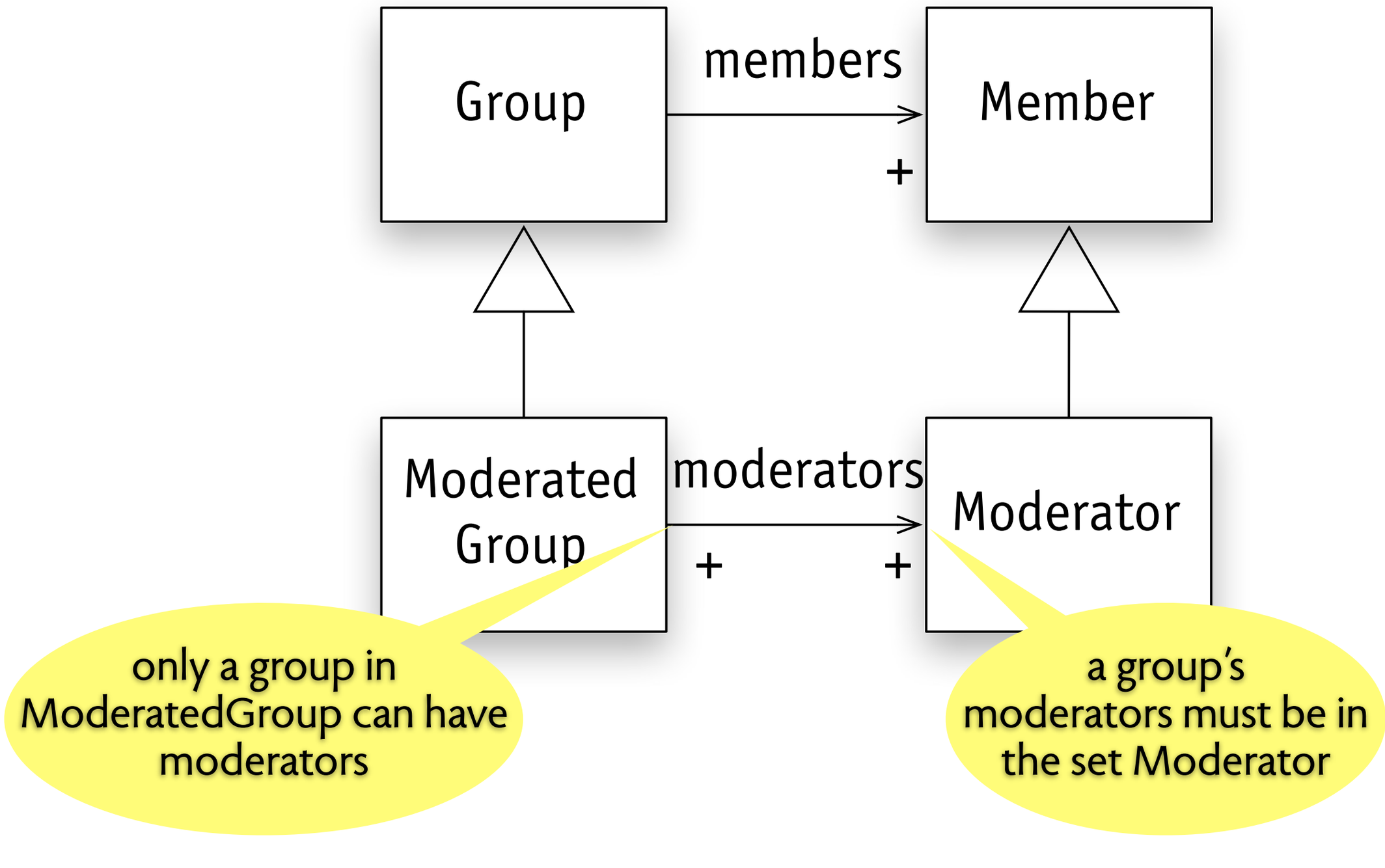
delving into semantics



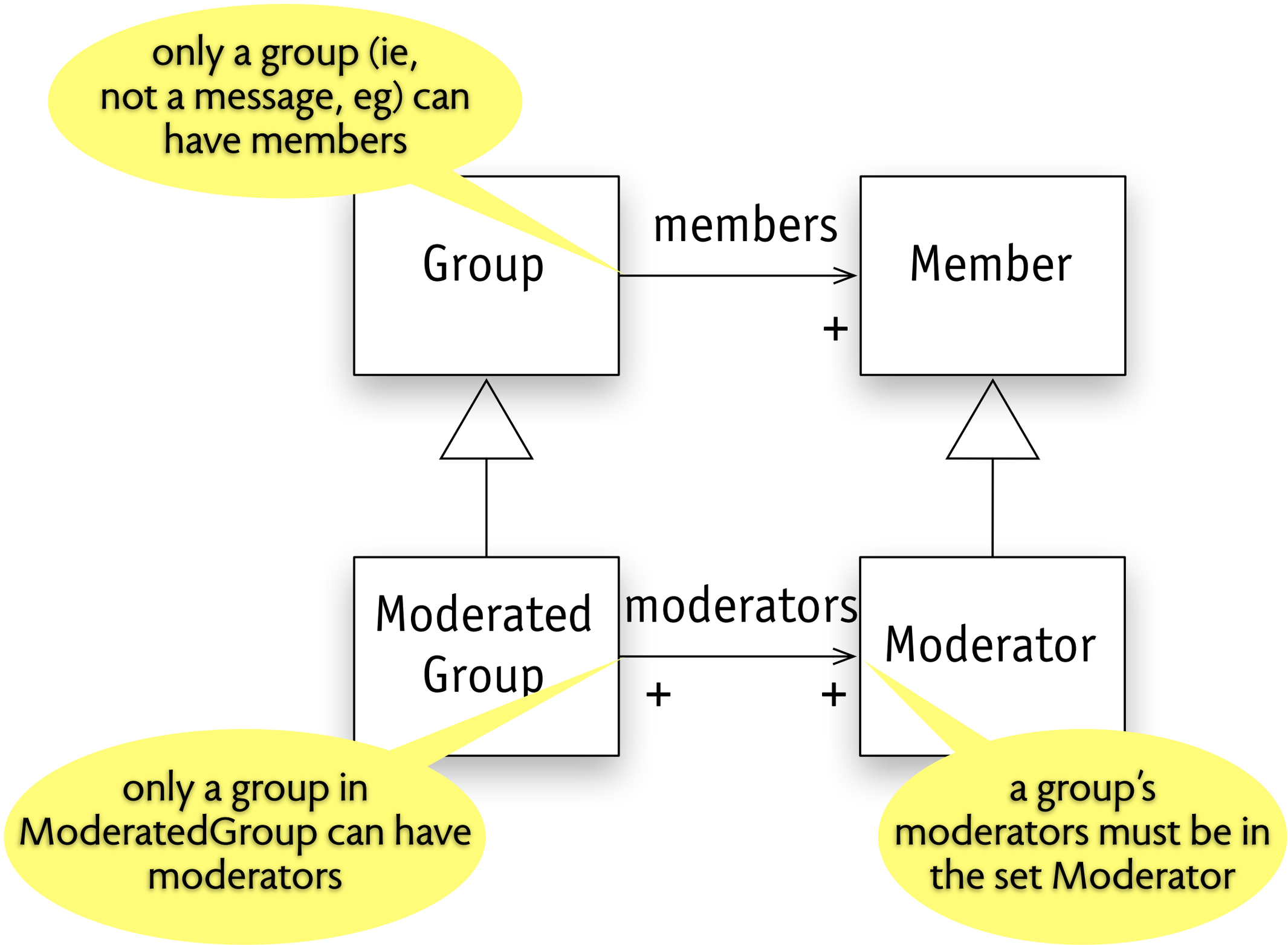
delving into semantics



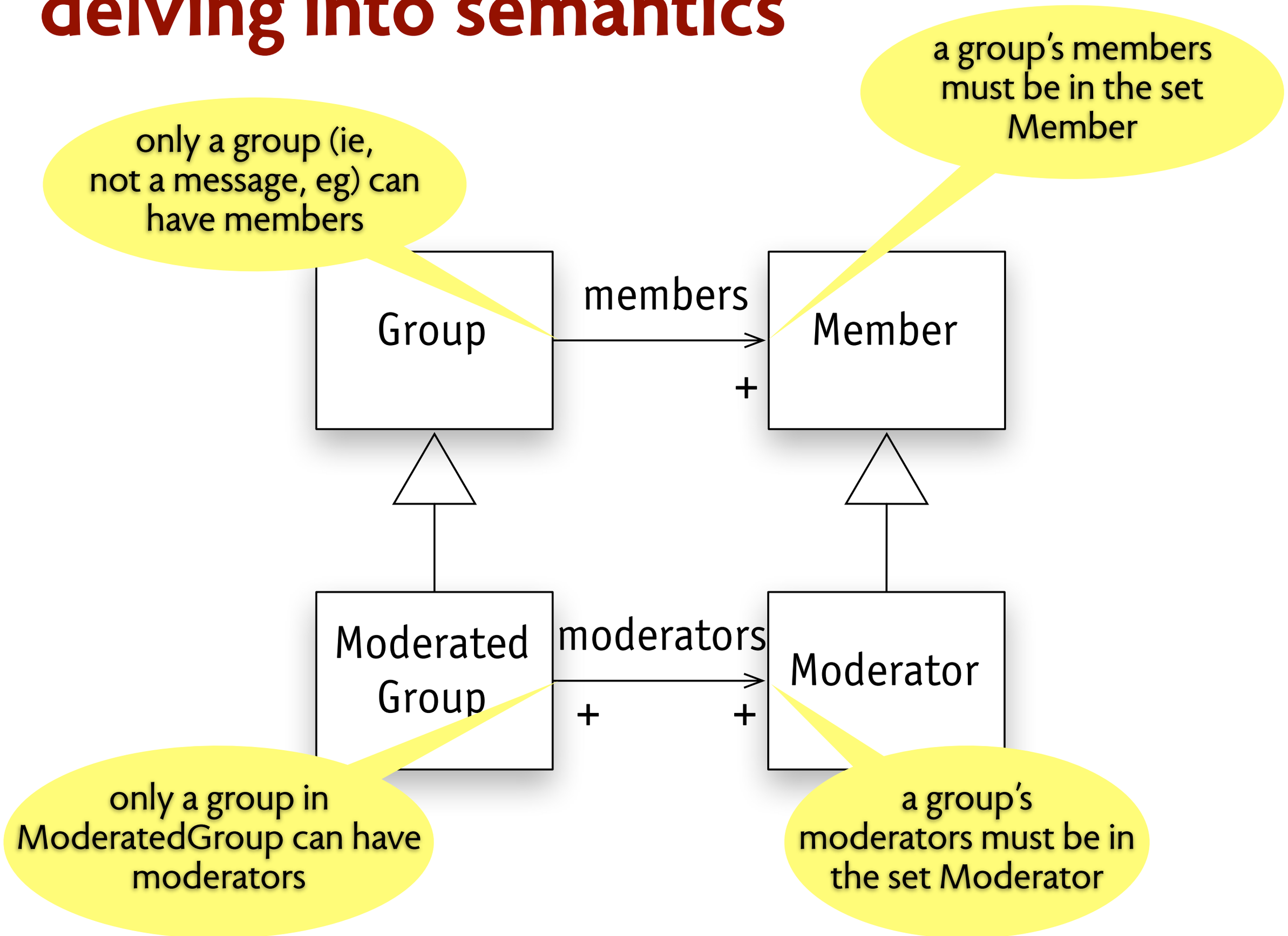
delving into semantics



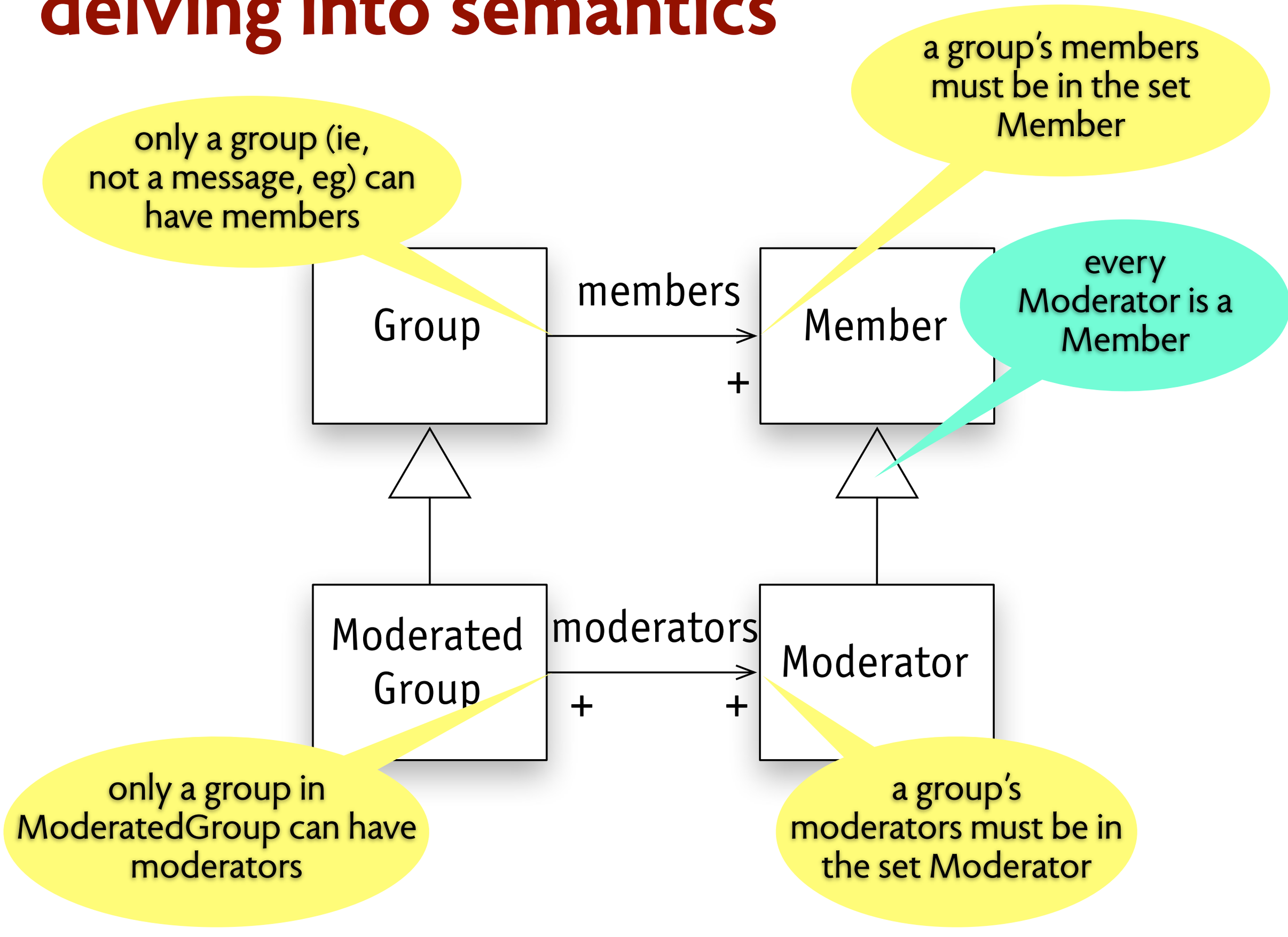
delving into semantics



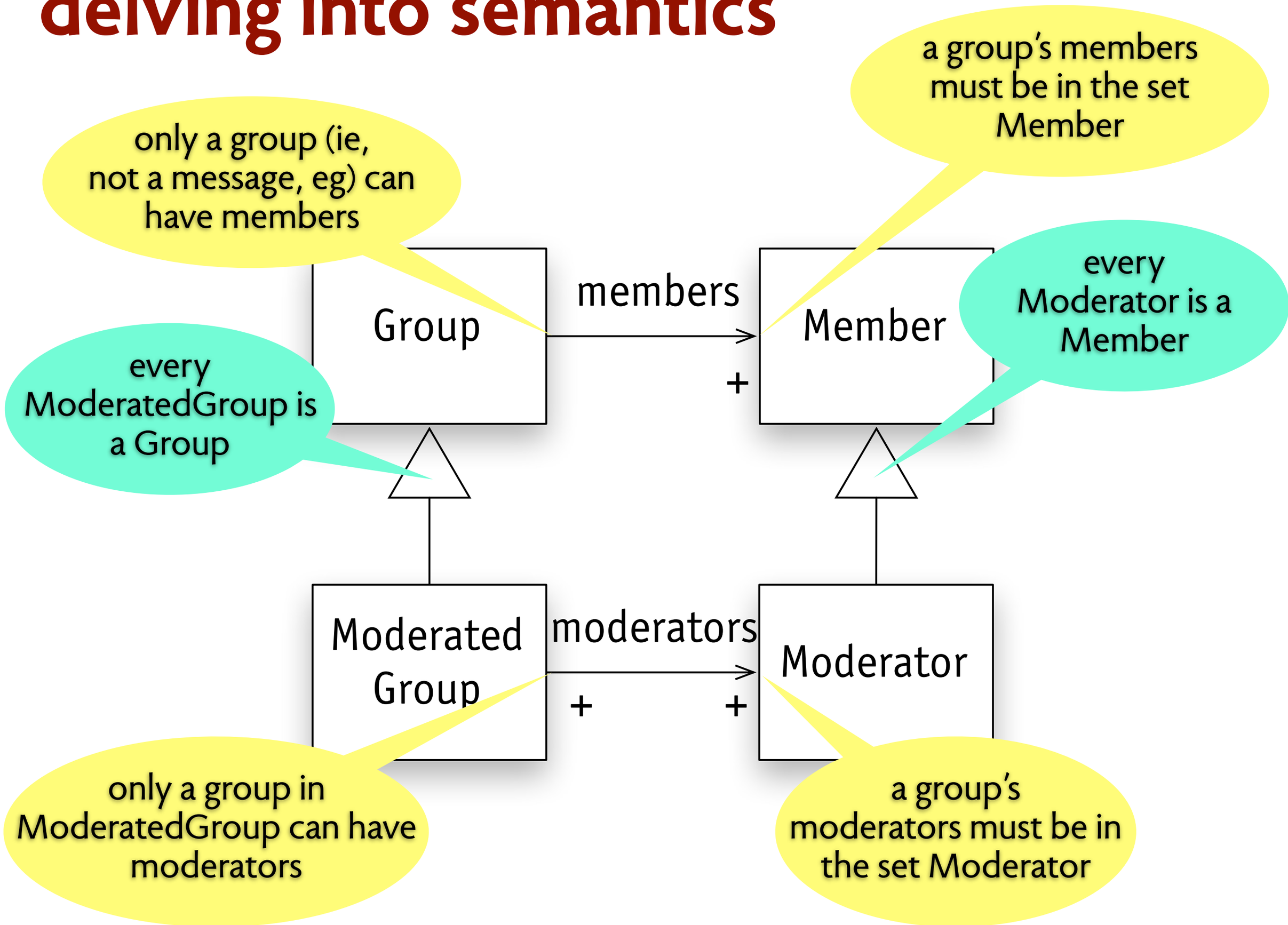
delving into semantics



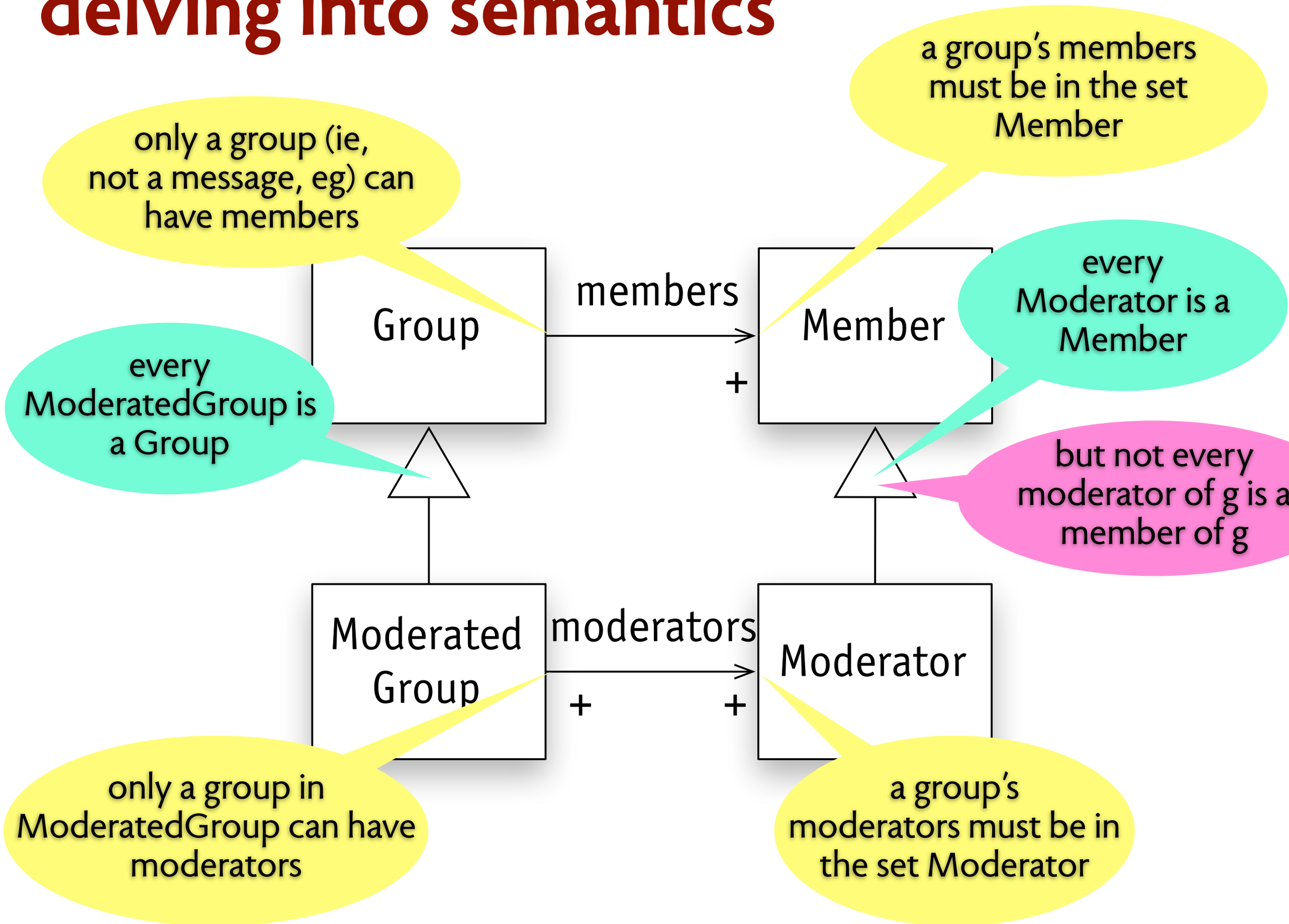
delving into semantics



delving into semantics

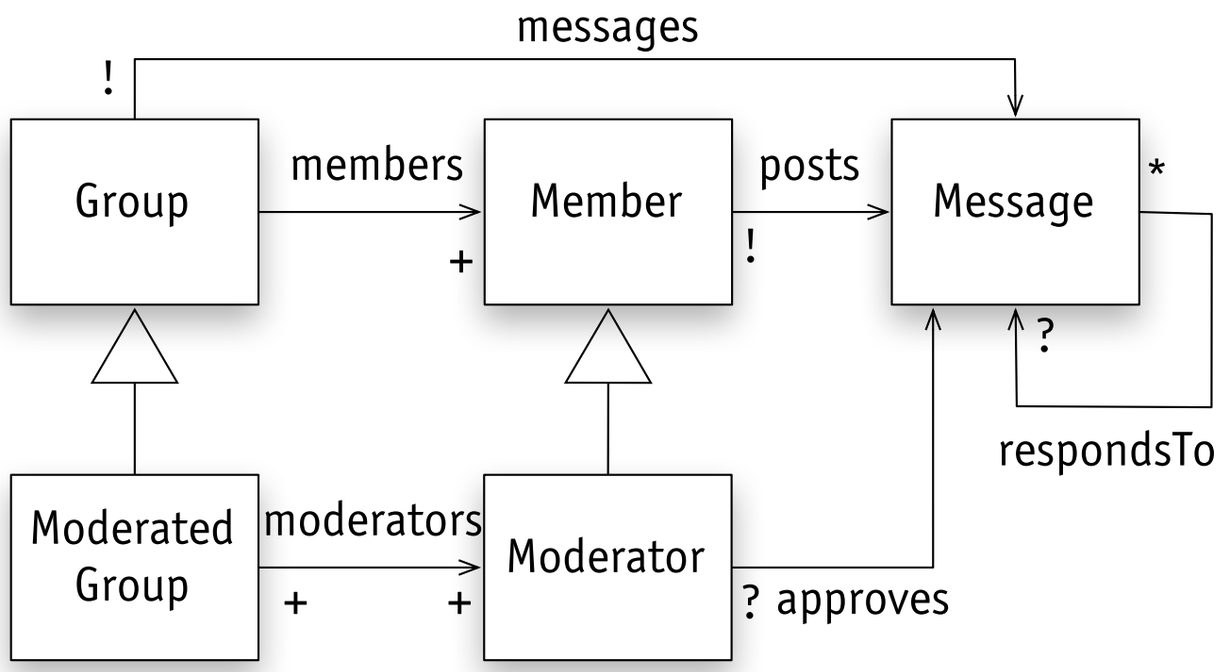


delving into semantics



textual constraints

- moderators must be members of the group
- member only posts message in group she belongs to
- moderators approve messages in groups they moderate
- message only responds to message in same group



in this course, just informal text; more advanced: express in Alloy

in mongo: still choices!

embedded

```
{
    title: "Fury",
    time: "7:00pm",
    theater: {
            name: "West Newton Cinema",
            location: "Newton"
        }
}
```

one document in the collection *Movies*

relational

```
{
    title: "Fury",
    time: "7:00pm",
    theater: 1
}
```

one document in the collection *Movies*

```
{
    _id: 1,
    name: "West Newton Cinema",
    location: ”Newton”
}
```

one document in the collection *Theaters*

repertoire of design moves

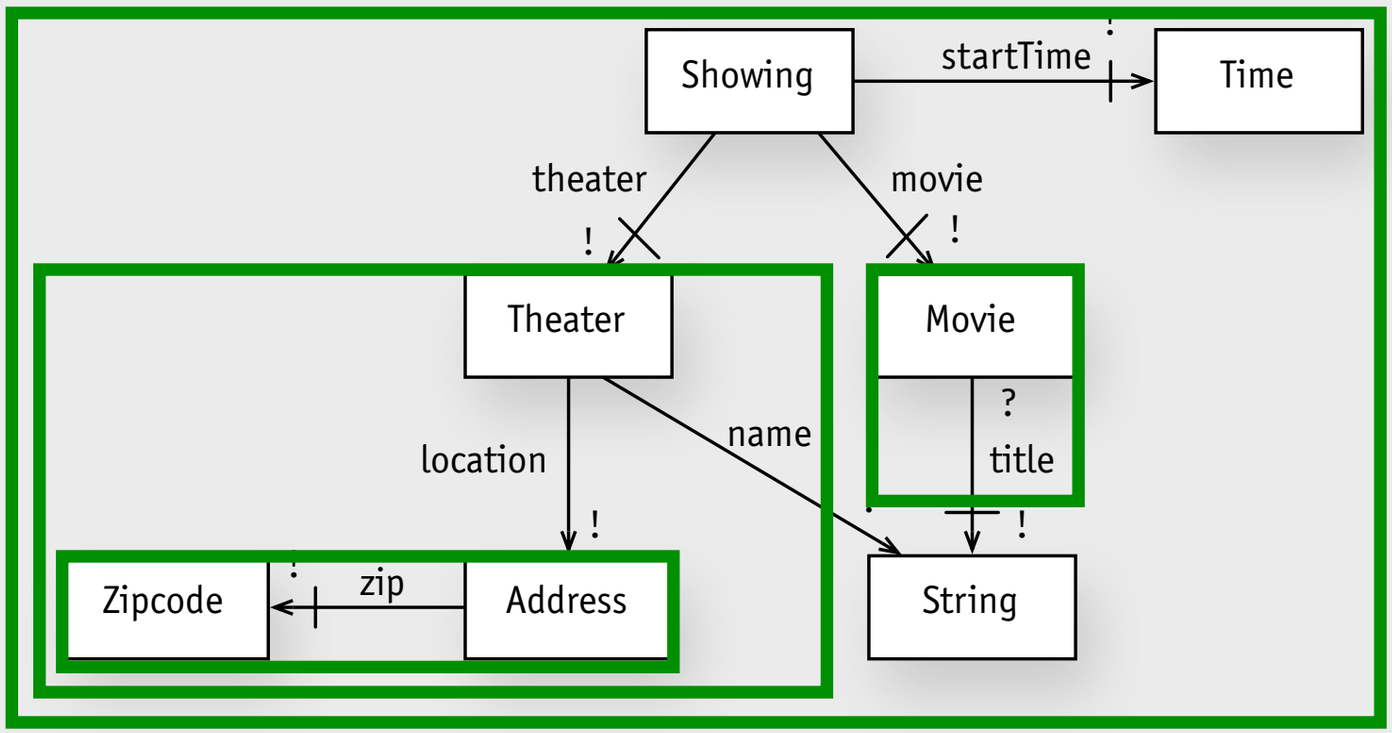
basic moves

- reverse relation
- add/remove object
- nest objects
- choose key

other moves

- add redundancy
- add index

making a nested structure



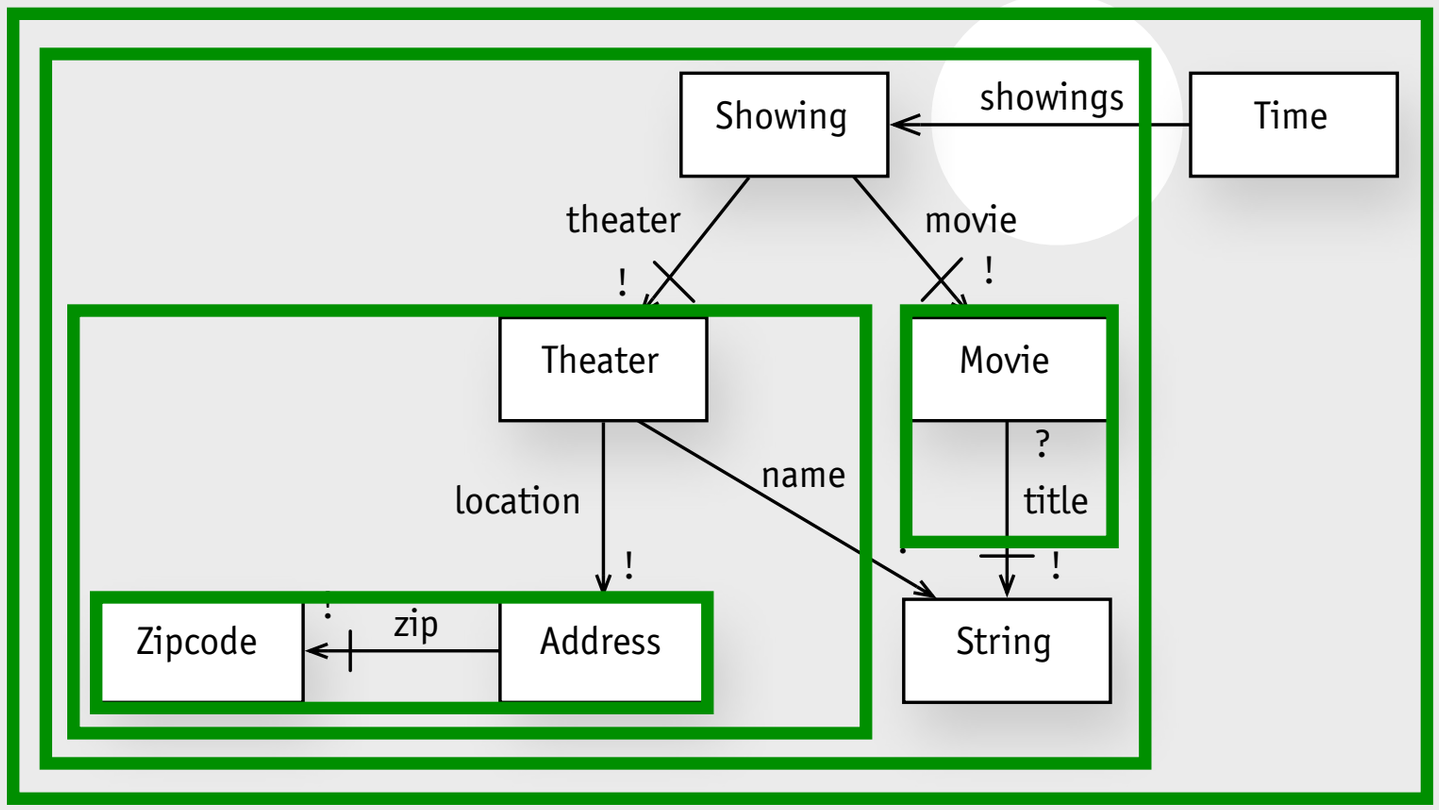

```
{title: "Fury"}
```

```
{zip: "02139"}
```

```
{location: {zip: "02139"},
 name: "Kendall"}
```

```
{
theater: {location: {zip: "02139"},
  name: "Kendall"},
movie: {title: "Fury"}
startTime: "7pm",
}
```

reversing a relation




```
{title: "Fury"}
```

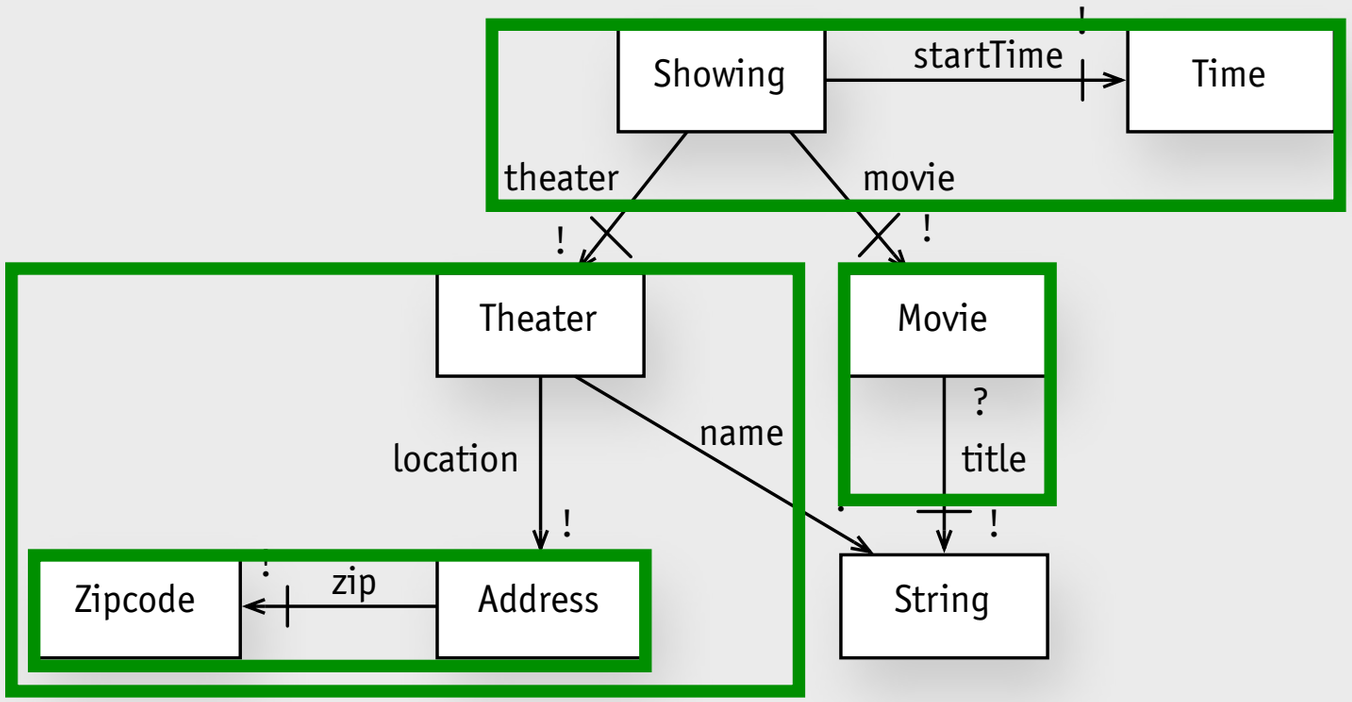
```
{zip: "02139"}
```

```
{location: {zip: "02139"},
 name: "Kendall"}
```

```
{
theater: {location: {zip: "02139"},
          name: "Kendall"},
movie: {title: "Fury"}
}
```

```
{
time: "7pm",
showings: [
 theater: {location: {zip: "02139"},
           name: "Kendall"},
 movie: {title: "Fury"}]
}
```

relating structures




```
{title: "Fury"}
```

```
{zip: "02139"}
```

```
{location: {zip: "02139"},
 name: "Kendall"}
```

```
{
startTime: "7pm",
theater: 2,
movie: 1
}
```

```
{_id: 1, title: "Fury"}
```

```
{_id: 2, location: {zip: "02139"},
 name: "Kendall"}
```