

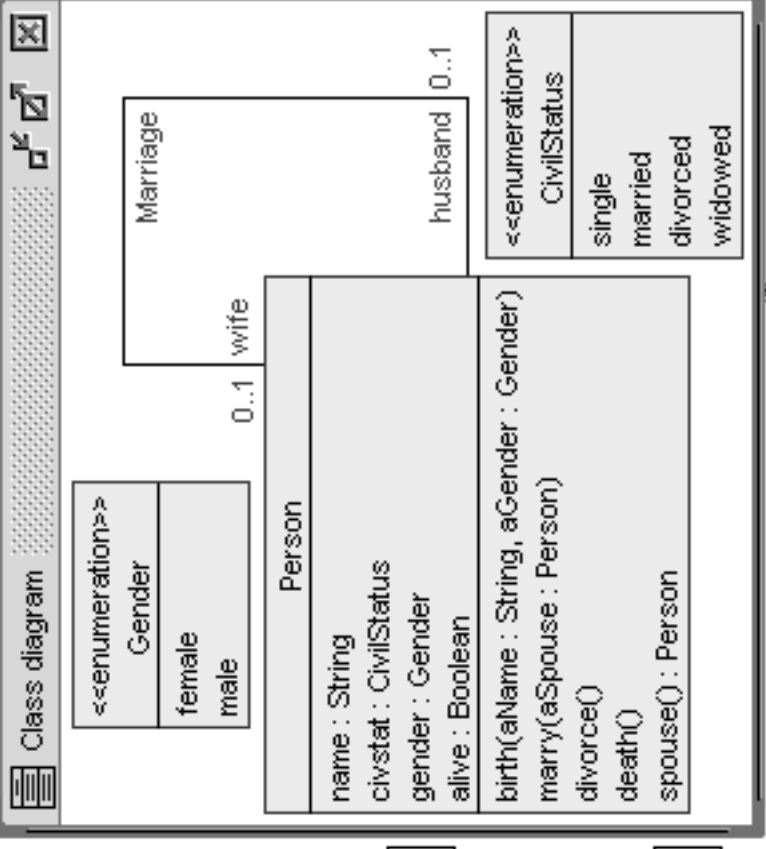
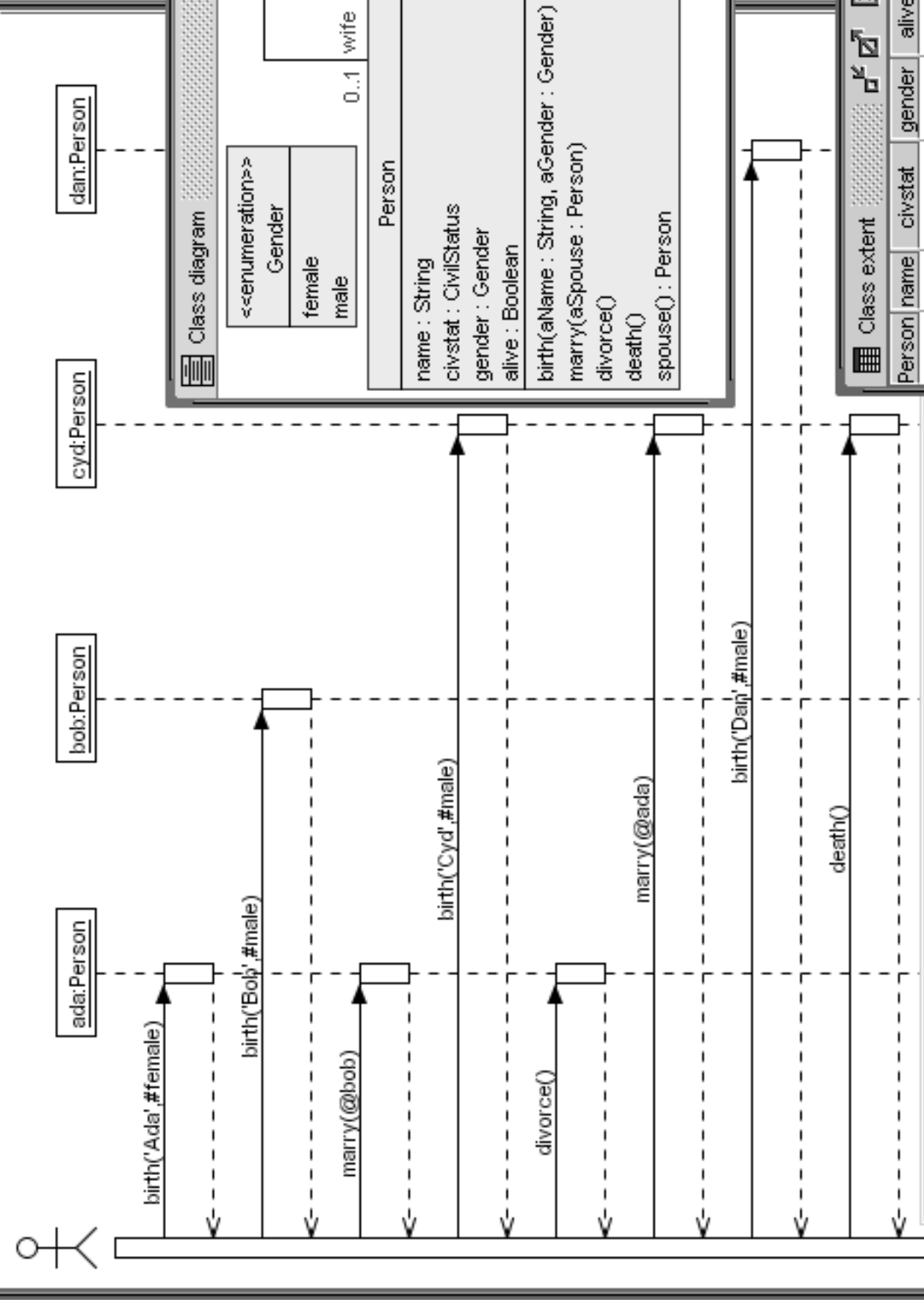


- CivilStatusWorld
- Classes
  - Person
- Associations
  - Marriage
- Invariants
  - Person::attributesDefined
  - Person::nameCapitalThenSmallLetters
  - Person::nameIsUnique
  - Person::femaleHasNoWife
  - Person::maleHasNoHusband
- Pre-/Postconditions
  - pre birth::freshUnlinkedPerson
  - post birth::nameAssigned
  - post birth::civstatAssigned
  - post birth::genderAssigned
  - post birth::isAliveAssigned
  - pre marry::aSpouseDefined
  - pre marry::isAlive
  - pre marry::aSpouseAlive
  - pre marry::isUnmarried
  - pre marry::aSpouseUnmarried
  - pre marry::differentGenders
  - post marry::isMarried
  - post marry::femaleHasMarriedHusband
  - post marry::maleHasMarriedWife
  - pre divorce::isMarried
  - pre divorce::isAlive
  - pre divorce::husbandAlive
  - pre divorce::wifeAlive
  - post divorce::isDivorced
  - post divorce::husbandDivorced
  - post divorce::wifeDivorced
  - pre death::isAlive
  - post death::notAlive
  - post death::husbandWidowed
  - post death::wifeWidowed

```

context Person::marry(aSpouse : Person)
pre differentGenders: (self.gender <>
aSpouse.gender)
  
```

Sequence diagram



Class extent

Person	name	civstat	gender	alive
ada	'Ada'	#widowed	#female	true
bob	'Bob'	#divorced	#male	true
cyd	'Cyd'	#married	#male	false
dan	'Dan'	#single	#male	true

Evaluate OCL expression

Enter OCL expression:  
 Person.allInstances->select(plp.alive)->collect(p|Sequence(p.name,p.civstat))

Result:  
 Bag(Sequence('Ada',#widowed),Sequence('Bob',#divorced),Sequence('Dan',#single)) : Bag(Sequence(OclAny))

Evaluate Clear Result Close