

Specification of the P2P Example

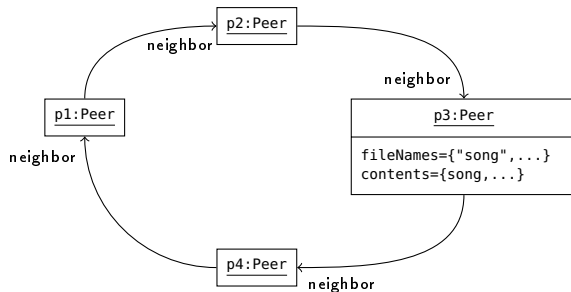
Annabelle Klarl

Ludwig-Maximilians-Universität München, Germany

Sequence of actions without roles

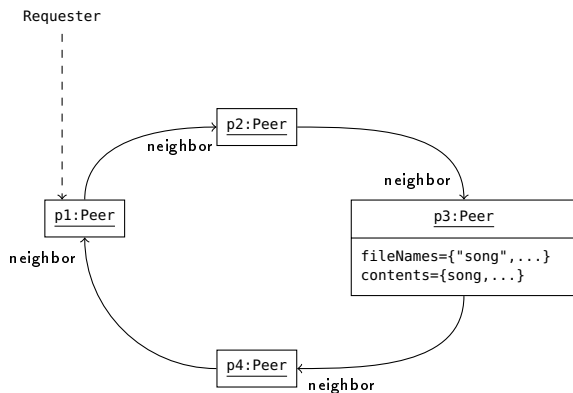
Task: A peer wants to download a file from a network.

«component type» Peer
List<String> fileNames List<char[]> contents



Sequence of actions without roles

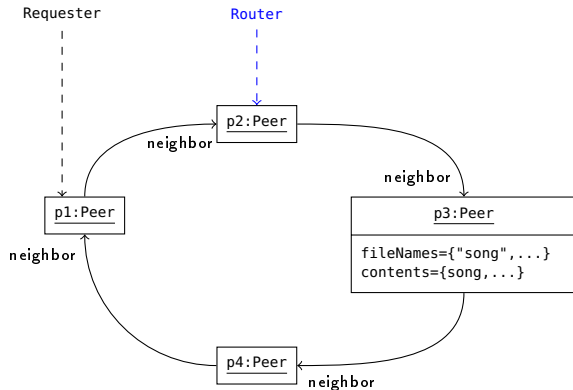
Task: A peer wants to download a file from a network.



«component type» Peer
List<String> fileName List<char[]> contents

Sequence of actions without roles

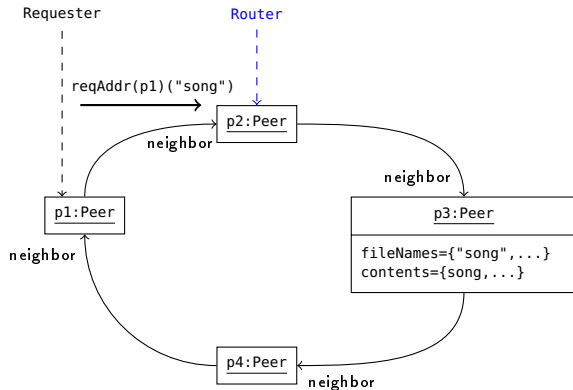
Task: A peer wants to download a file from a network.



«component type» Peer
List<String> fileNames List<char[]> contents

Sequence of actions without roles

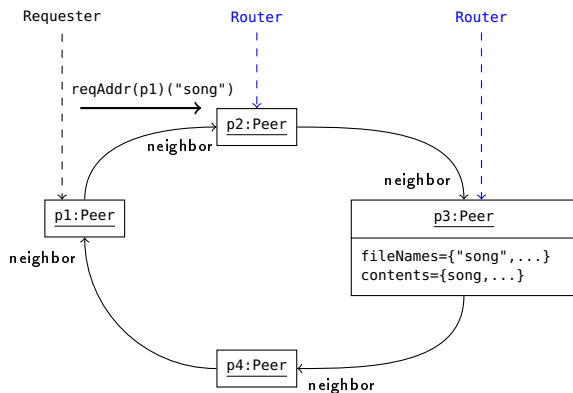
Task: A peer wants to download a file from a network.



«component type» Peer
List<String> fileNames List<char[]> contents

Sequence of actions without roles

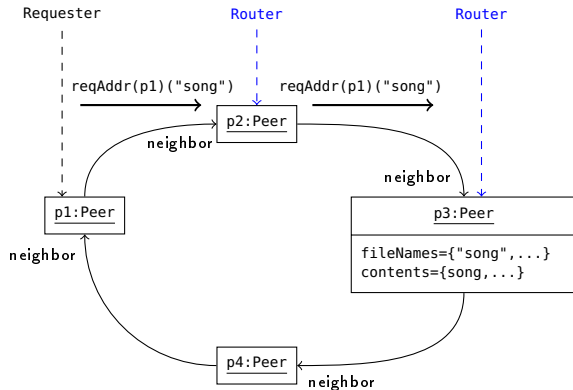
Task: A peer wants to download a file from a network.



«component type» Peer
List<String> fileNames List<char[]> contents

Sequence of actions without roles

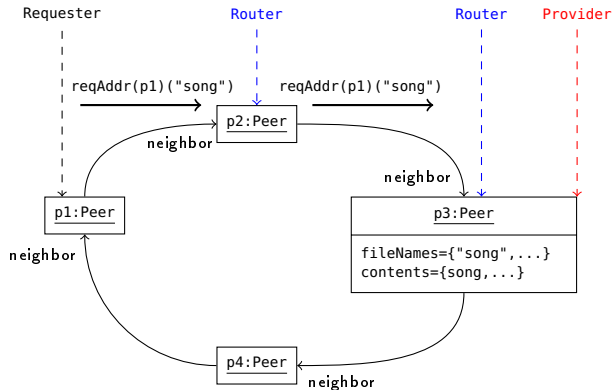
Task: A peer wants to download a file from a network.



«component type» Peer
List<String> fileNames List<char[]> contents

Sequence of actions without roles

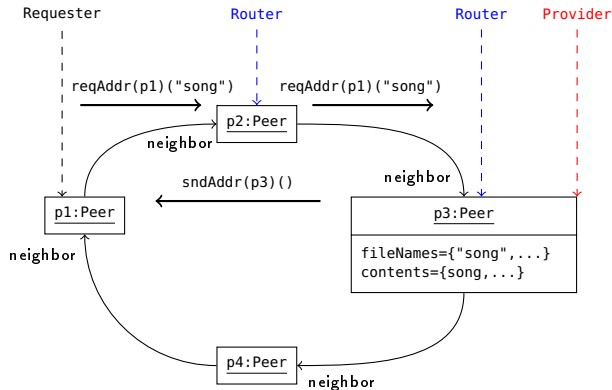
Task: A peer wants to download a file from a network.



«component type» Peer
List<String> fileNames List<char[]> contents

Sequence of actions without roles

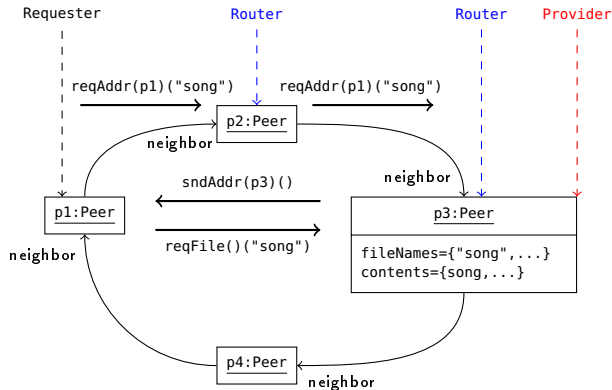
Task: A peer wants to download a file from a network.



«component type» Peer
List<String> fileNames List<char[]> contents

Sequence of actions without roles

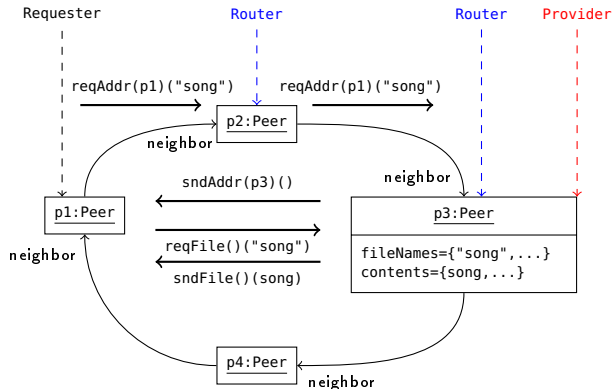
Task: A peer wants to download a file from a network.



«component type» Peer
List<String> fileNames List<char[]> contents

Sequence of actions without roles

Task: A peer wants to download a file from a network.



«component type» Peer
List<String> fileNames List<char[]> contents

Observation

A component instance can play **different roles in the same ensemble**.

Components and Roles

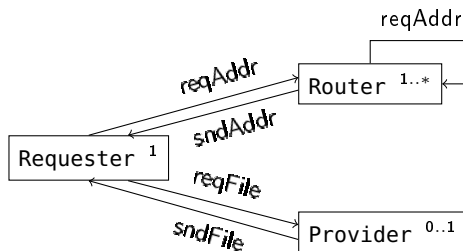
«component type» Peer
List<String> fileNames List<char[]> contents

«role type» Requester:{ Peer }
String fileName
out reqAddr(Requester req)(String fn) in sndAddr(Provider prov)() out reqFile(Requester req)(String fn) in sndFile()(char[] cont)

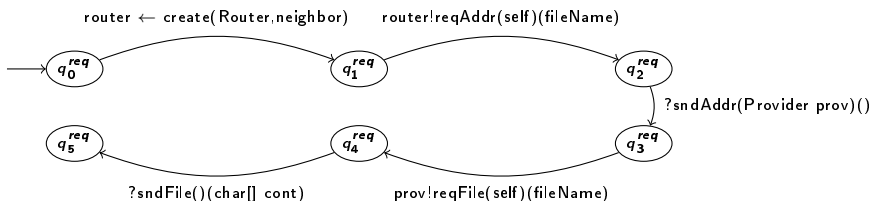
«role type» Router:{ Peer }
out reqAddr(..)(..) in reqAddr(..)(..) out sndAddr(..)(..)

«role type» Provider:{ Peer }
in reqFile(..)(..) out sndFile(..)(..)

Ensemble Structure without Role Connectors



Role Behaviors without Role Connectors: Requester



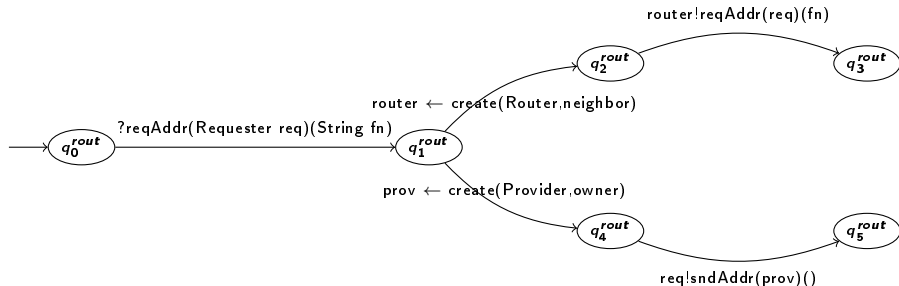
$RoleBeh_{Requester} =$

```

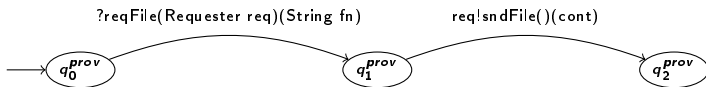
router ← create(Router, neighbor) .
  router!reqAddr(self)(fileName) .
    ?sndAddr(Provider prov)() .
      prov!reqFile(self)(fileName) .
        ?sndFile()(char[] cont) . nil
    
```

Role Behaviors without Role Connectors: Router and Provider

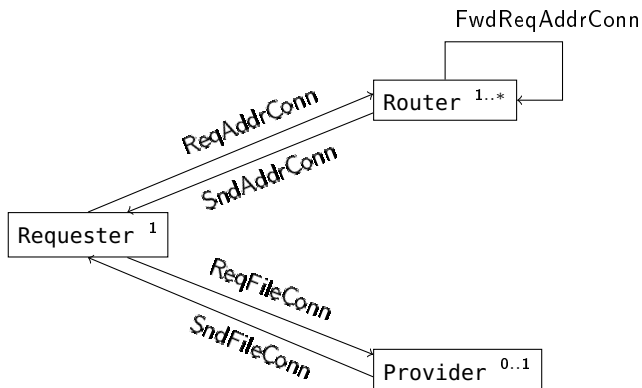
$RoleBeh_{Router} =$



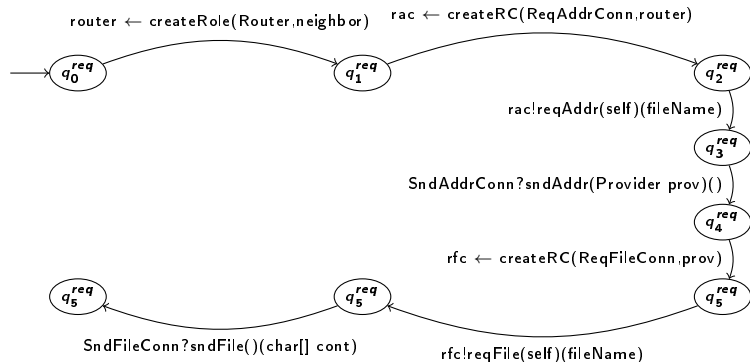
$RoleBeh_{Provider} =$



Ensemble Structure with Role Connectors



Role Behaviors with Role Connectors: Requester



*RoleBeh*_{Requester} =

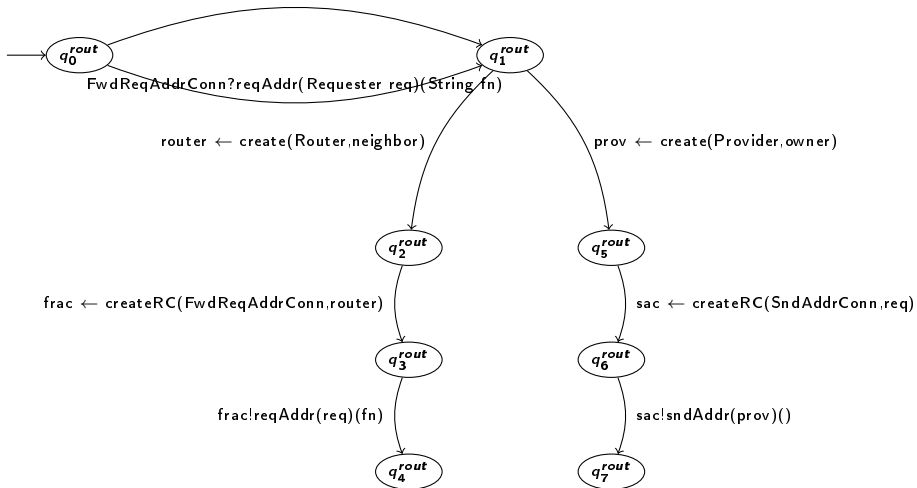
```

router  $\leftarrow$  createRole(Router, neighbor) .
rac  $\leftarrow$  createRC(ReqAddrConn, router) .
rac!reqAddr(self)(fileName) .
SndAddrConn?sndAddr(Provider prov)() .
rfc  $\leftarrow$  createRC(ReqFileConn, prov) .
rfc!reqFile(self)(fileName) .
    
```

Role Behaviors with Role Connectors: Router

$RoleBeh_{Router} =$

$ReqAddrConn?reqAddr(Req\ ester\ req)(String\ fn)$



Role Behaviors with Role Connectors: Provider

$RoleBeh_{Provider} =$

