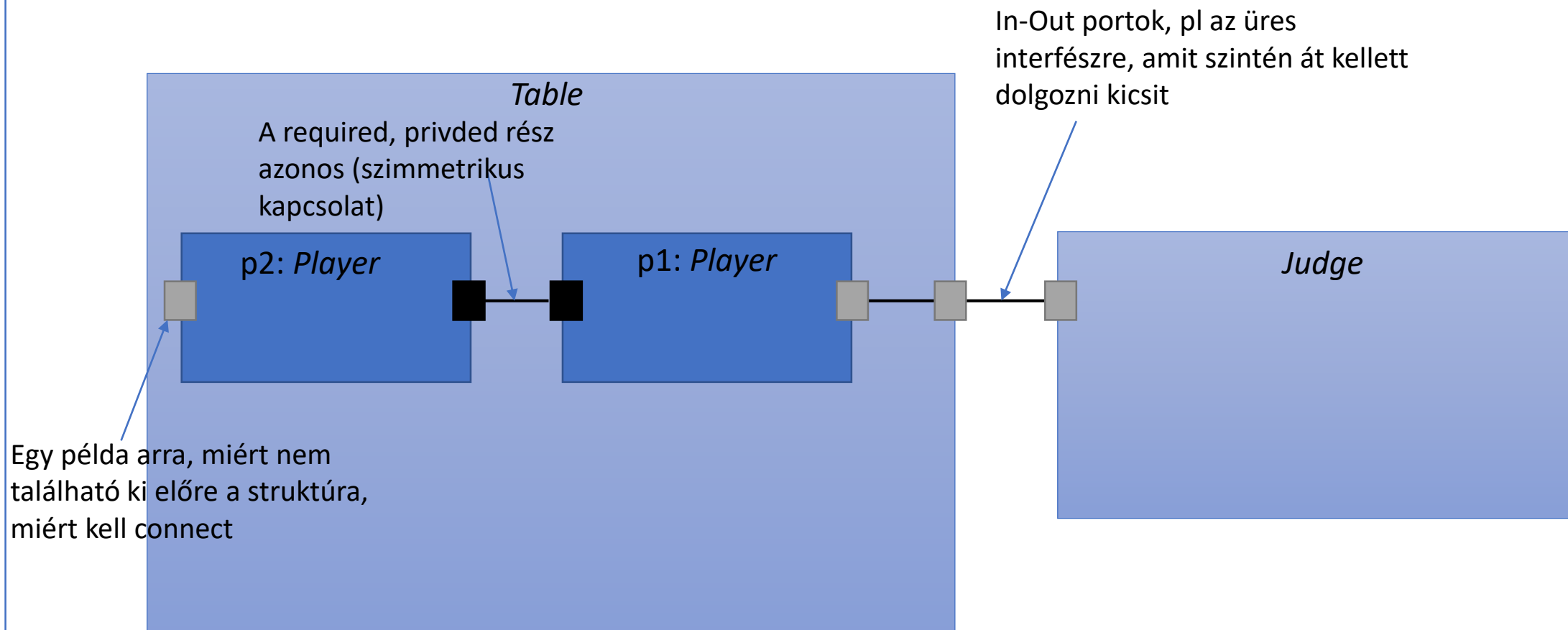


Ping-Pong demo
komponentsekkel

PingPongMatch

■ Behavior port



Connect reprezenációja UML-ben

- A connectorokhoz asszociációkat generálunk, mellyel fel tudjuk tipusozni az adott connectort.
- Az asszociáció végpontjainak a típusa megegyezik az interfész-port típusával.
- Lényegében két interfészt kötünk össze.
- Így validan használható a create link akció a connect művelethez.
- txtUML-ben két különböző connect van, assembly és delegation.
- Assembly esetén meg kell adni mindkét végpontot, így mindkét interfész könnyen kinyerhető. Delegation esetén csak a gyerek interfész érhető el könnyen, így trükközni kell. (Ez kicsit furcsa, hack gyanús)

Connect a generált kódban

- Egyelőre csak az interfész kapcsolatot generálom
- Ha látok egy create linket, jelenleg még nem tudom egységesen kezelni abban az esetben, ha port az argumentuma. (ezen később jó lenne változtatni)
- Valamint azt sem látom explicit, hogy delegate vagy assembly connectet hívott-e a felhasználó, ezt a két port viszonya dönti el. (szülő-gyerek, vagy testvérek)
- Delegate connect esetén azonos típusú interfészeket kötök össze, így itt további trükközésre van szükség, hogy kinyerjem a szolgáltatott interfészt.
- A végpontokat csak a required-provided típushelyesség ellenőrzésére használom fel, ami kódgenerálás esetén elegendő (mivel ott feltesszük, hogy a modell helyes), de felhasználó által írt kódnál nem feltétlenül, könnyű megheceklni. (Ezért is kéne a kapcsolatokat egységesen kezelni, de ez egy nagyobb munka lenne..)

Send megvalósítása UML-ben

- Jelenleg a legegyszerűbb megoldás van implementálva, amely kihasználja, hogy a port egy property, így lehet a send object akció targetje.
- Mivel olyan szignált küldhetünk egy portnak, ami a required interfészen van, így nem hívhatjuk meg a required interfész receptiönjét a porton az adott szignállal, mivel UML-ben nem típusa ez az interfész.
- Valószínűleg a sendOnPort tulajdonságot kéne kihasználni a send object action esetén.

Send megvalósítása generált kódban

- Generált kódban már fel van tipusozva az adott port a required interfésszel, így a send gond nélkül megvalósítható.
- A send implementáció függő, máshogy viselkedik behavior port és sima port esetén.
- A típushelyességet az biztosítja, hogy minden interfész required partjához annyi send műveletet generálunk, ahány receptiönje van, túlterheljük a különböző típusú szignálokkal.
- Annak idején ez tűnt a legelegánsabb megoldásoknak, de lehetne az UML reprezentációhoz közelebbi megoldást is kitalálni (in progress, ahogy lesz rá idő..)