

Komponens-alapú UML modellek fordításának vizsgálata

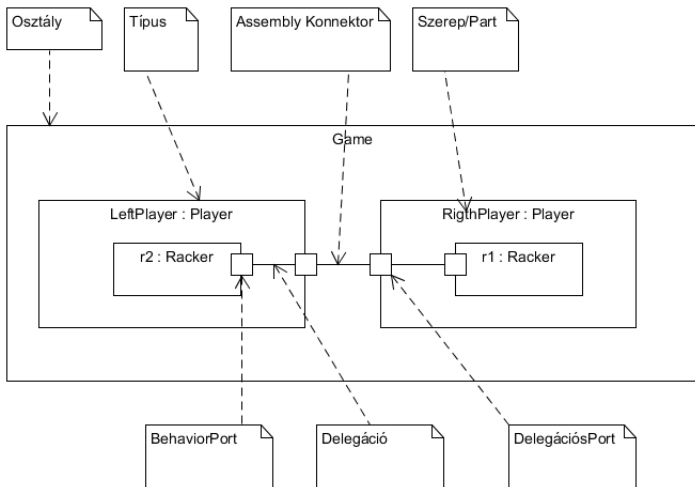
Nagy András

2019. január

- Program szereplőinek izolációja.
- A szereplők függetlenek a környezettől.
- Egymással interfész-portokkal kommunikálnak az egyes szereplők
- Előnyei: független telepíthetőség, explicit interfész függőségek..

- *txtUML* keretrendszerben írjuk le a komponens-alapú modellt, Java-szerű nyelven, mely végrehajtható.
- Lefordítható egy szabványos UML2 modellre.
- A cél a kompozit struktúrák és akciók megfelelő UML2-es szabványának megtalálása, melyből hatékony C++ kód generálása.

Példa egy konkrét kompozit struktúrára



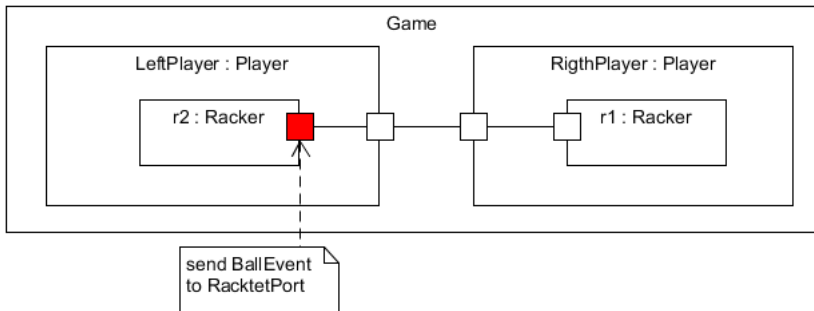
Már

az UML2-es reprezentáció sem triviális.

- Interfész port reprezentálása: mi a típusa, hogyan fejezzük ki az elvárt interfészt.
- Két port összekapcsolása futási időben.
- Porta való üzenetküldés.
- Mi a modell szemantikája??

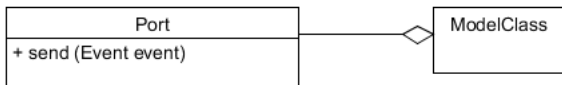
- A *Connector* értelemszerű (de mi az a *type* referencia?)
- A problémás a *connect* művelet
 - Nincs *connect* akció UML-ben
 - *DefaultConstructionStrategy* (de az a szerkezet nem mindig egyértelmű..)
 - A *CreateLinkAction* segítségével összeköthetünk két portot a konnektor típusa mentén (Itt jön be a *type* referencia, mely egy asszociációnak felel, ezt még ki kell generálni.).

- Sok alternatíva (típusbiztonság, adatrepresentációs különbségek, stb)
- Különböző elvárások a generált kóddal szemben (hatékonyság, olvashatóság, külső kóddal történő biztonságos illesztés, stb.)
- Ez főleg a megfelelő kettrendszerek kialakításából és az UML szemantika értelmezéséből áll.
- Ezeket a szempontokat figyelembe véve elemezni az egyes elemek generálását (interfész: 3+1, különböző port típusok: 2, konnektor struktúrák: 2, kapcsolódási végpontok tárolása: 2 + 1, portok összekapcsolásának kifejezése 2 + 1, üzenetküldés/fogadás: 2, üzenetfeldolgozás: 2).



Kérdések:

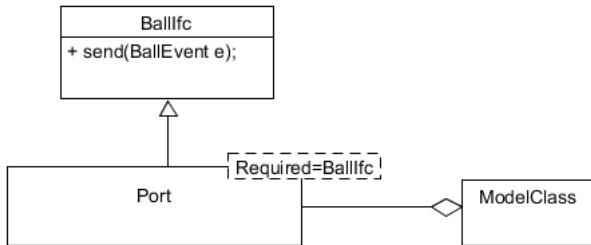
- Hogyan reprezentáljuk a szabványban a portra való üzenetküldést, hogyan kell ezt értelmezni?
- Interfész portok reprezentálása, generálása?



Problémák:

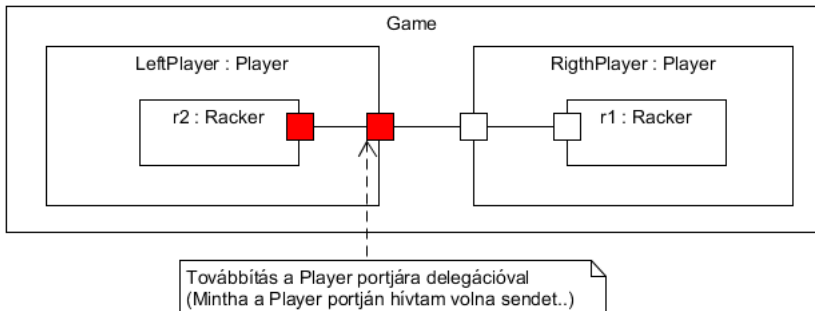
- A port bármilyen üzenetet fogadhat, hol vannak az interfészek?
- Mi a send szemantikája, milyen UML akciónak felel meg?

Interfész portok üzenetküldéssel



- A típushelyességet biztosítottuk.
- A továbbiakban a *send* szemantika érdekes.

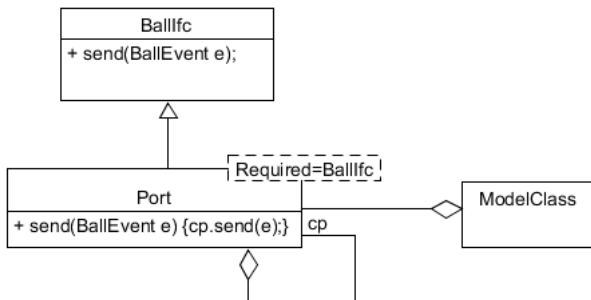
Üzenet továbbítása a szülő komponens felé - Delegáció



Problémák:

- Hogyan adom tovább az üzenet a szülő felé?
- Hogyan kapcsolom össze a gyerek és a szülő portját?
- Honnan tudom, hogy delegációs kapcsolat áll a két port között?

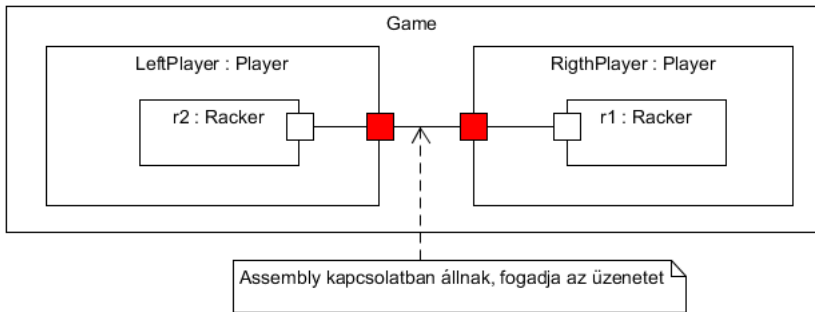
Naiv kapcsolat referencia



Problémák:

- Delegációs kapcsolat esetén működik csak.
- Delegációs összekapcsolás esetén kell beállítani a referenciát.

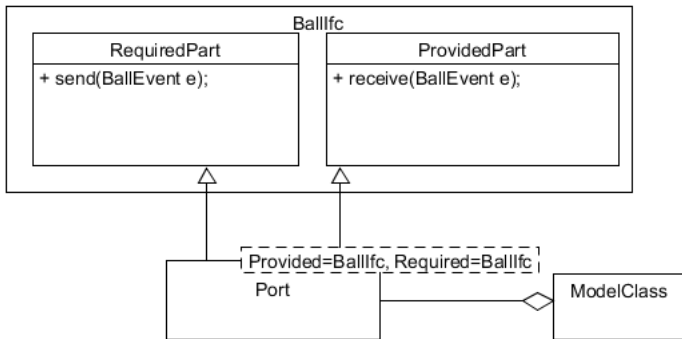
Üzenet átadása testvér komponensek - Assembly



Újabb problémák:

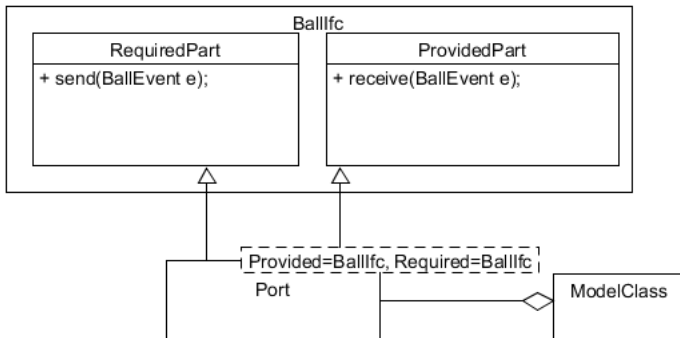
- A *send* másképp viselkedik, mint delegációs kapcsolat esetén.
- A jobb oldali célportnak nem delegálok, hanem fogadja az üzenetet, ami befele áramol tovább. (Az elvárt interfész érdekes számomra)

Interfészek szétbontása



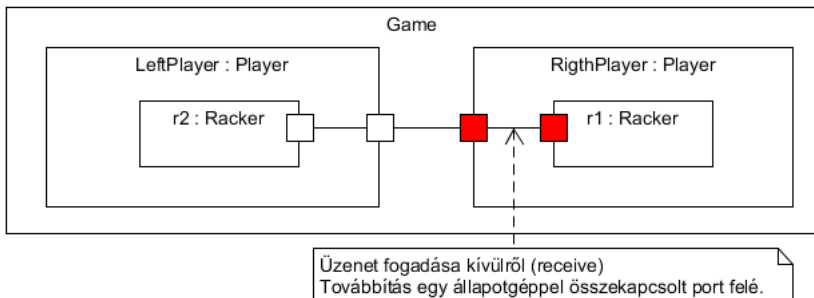
- Az PSCS szemantika szerint az üzenetküldés kontextusától függ az üzenetáram iránya.
- *send* - belülről jött üzenetküldés, *receive* - kívülről jött üzenet fogadása

Kapcsolat referencia szétbontása



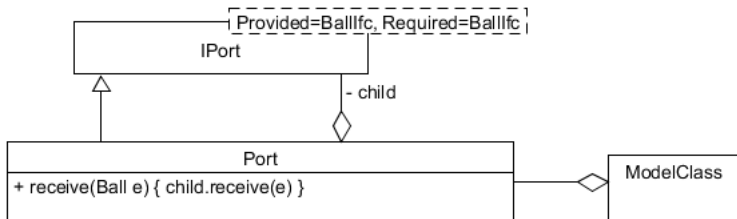
- A kapcsolat referenciát öröklődéssel tehetjük a legtisztábban poliformmá.
- Az assembly kapcsolódás miatt *send* helyett *receive*-et kell hívni a kapcsolt porton.

Kapcsolat referencia szétbontása



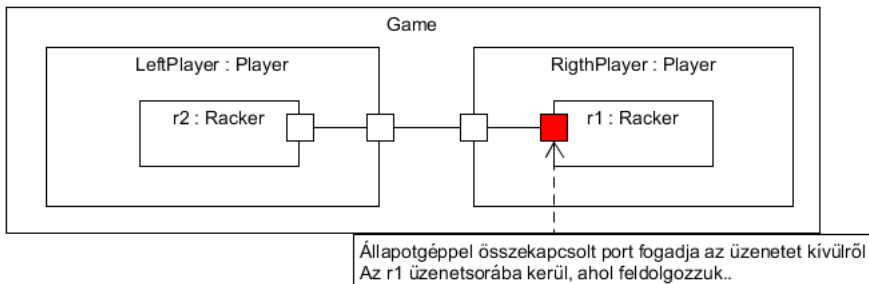
- A delegáció miatt továbbítani kell az üzenetet a belső komponens felé.
- Ez a kapcsolat azonban különbözik az általános kapcsolat referenciától.
- Mintha a gyerek *receive*-jét hívtuk volna..

Gyerek felé továbbítás delegáció esetén



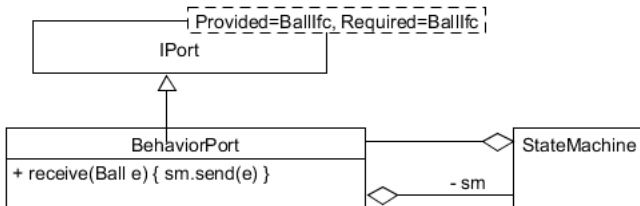
- A gyerek referenciát egy delegációs összekapcsolás állítja be.
- Más lesz ennek a viselkedése állapotgéppel összekapcsolt port esetén, ezért kell az interfész..

Gyerek felé továbbítás delegáció esetén



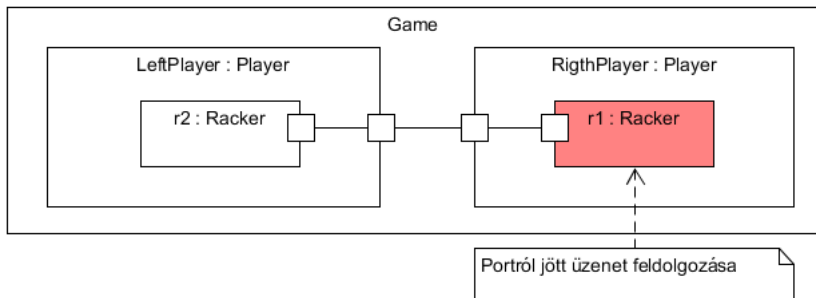
- Gyerek komponens referencia helyett az állapotgépre kell referenciát birtokolnunk.
- Más viselkedés, más adattagok.

Gyerek felé továbbítás delegáció esetén



- A *send* hatására bekerül az állapotgép üzenetsorába az *Ball* üzenet.
- Tudnunk kell, melyik portról érkezett az üzenet, amikor átmenetben dolgozzuk fel.

Gyerek felé továbbítás delegáció esetén



- Átmeneteknél (esemény, állapot) mellett megadhatjuk, mely portokról jött üzenetek relevánsak az átmenet szempontjából.
- Állapotgép reprezentálása mint eddig, figyelembe véve, honnan érkezett az üzenet (átmenet tábla bővítése, maszkolás).

- Az UML kompozit szabvány alapos értelmezése, a megfelelő reprezentációk és szemantika megtalálása.
- C++ kódgenerálási stratégiák készítése.
- Szabványos modell generálása txtUML modell alapján, egy stratégia implementációja.
- A diplomamunka összefoglalja, hogy miben segítenek a portok a modellezésben (könnyebb párhuzamosíthatóság, jobb FMU wrapper, példamodellek készítése oktatási célokra komponens-alapú oktatáson, stb.).

Köszönöm a figyelmet!