Modellezés és szimuláció Petri hálókkal

dr. Bartha Tamás
Dr. Pataricza András

BME Méréstechnika és Információs Rendszerek Tanszék

Diszkrét rendszermodellezés célja

- Informatikai rendszerek: jól tagoltak
  - rendszerépítés a komponensek integrációjával
  - elemi komponensek kapcsolata
    - explicit logikai kapcsolat: sorrendiség, ok-okozati függőség
    - implicit függőség: pl. osztott erőforrás használata
- Célkitűzés: minőségi vagy/és mennyiségi analízis
  - kvalitatív: logikai helyességbizonyítás
  - kvantitatív: teljesítményelemzés, megbízhatóság és rendelkezésre állás, biztonságosság

Az informatikai rendszerekre a jól tagoltság jellemző, azaz a rendszert elemi komponensek összeintegrálásával szokás felépíteni:

•az elemi komponenseket részben explicit logikai kapcsolat fűzi össze (pl. egy folyamat egymás utáni lépései, vagy két lazán csatolt folyamat közötti üzenetváltás),

•részben pedig implicit függőség lehet köztük (pl. két független taszk egy-egy lépése egy közös erőforrást oszt meg).

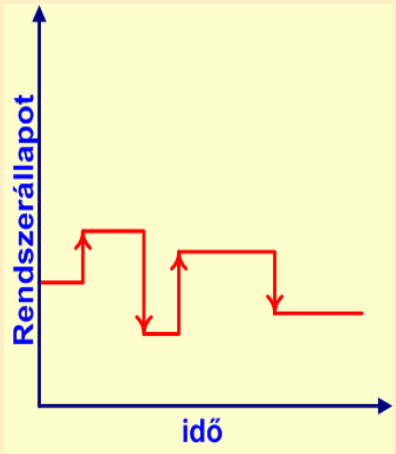
A diszkrét rendszermodellezés célkitűzése:

•egy rendszer működésének logikai helyességbizonyítása, illetve

•mennyiségi jellemzőkkel bővített modell alapján a rendszer teljesítményének, megbízhatóságának vagy rendelkezésre állásának vizsgálata.

Rendszermodellek felbontása

- Modellezés célja
  - Korreláció: modell ↔ „valóság”
  - Érthetőség, áttekinthetőség
    - Nem cél a minimális modell
  - A teljes rendszer dinamikája
    - Komponensek modellje egyszerű
    - Bonyolult kölcsönhatások
- Csoportosítás
  - Folytonos } • értékben
  - Diszkrét } • időben



Ennek megfelelően a modellalkotás során az elsődleges cél a modell áttekinthetősége és érthetősége, nem pedig minimális volta.

A korszerű modellezési paradigmák az áttekinthetőséget úgy támogatják, hogy a modellezőnek csak az explicit függőségeket kell a modellben felépítenie (pl. mik és milyen sorrendűek az elemi lépések), valamint megadnia azt, hogy mik a potenciálisan implicit kapcsolatot létrehozó kapcsolatok (pl. egy-egy elemi lépés milyen erőforrást használ). Az implicit kapcsolatok részleteinek kezelése (pl. az erőforrásokra vonatkozó kölcsönös kizárás) a futtatórendszer feladata.

Az elemi komponensek modellezése általában viszonylag egyszerű, a rendszeranalízis bonyolultságának fő okai a folyamatközi keresztkapcsolatok, így a rendszerszintű szimulációs kísérletek és formális vizsgálatok elsődleges célja az, hogy a rendszernek a dinamikus viselkedéséről alkossunk képet.

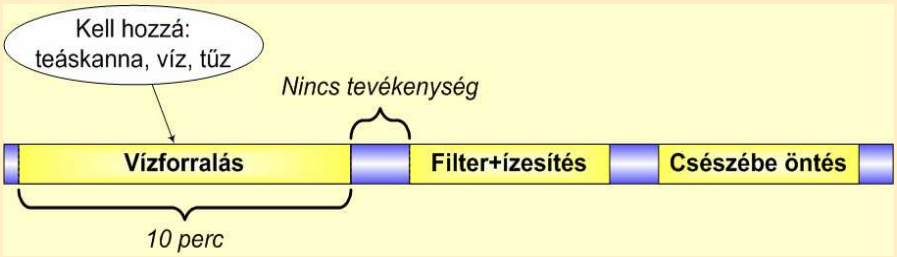
A rendszerek modellezése során megkülönböztethetünk *folytonos*, illetve *diszkrét* érték- illetve időmodellezést aszerint, hogy a rendszer állapotváltozóit, illetve az állapotváltozások időpillanatait folytonos vagy diszkrét skála szerint ábrázoljuk.

Az informatikai rendszerek zöménél mind az állapotváltozók értékkészletét, mind pedig az állapot-átmenetek időpillanatait diszkrét módon szokás modellezni. Természetesen nagyon nagy, sok ezer vagy tízezer komponensből álló rendszerek esetén, ahol már egy-egy komponens hatása önmagában kicsi, gyakorta hatásos a diszkrét tulajdonság „elvesztése” egy folytonos modellezésre történő átmenettel.

Folyamatorientált modellezés

- Elemi tevékenységekből áll
- Jellemzői:
  - Erőforrást igényel
  - Mennyi ideig tart
- Folyamat: tevékenységek logikailag rendezett sora

+ járulékos információk:
- erőforrások
- kölcsönhatások, pl.
- erőforrás használat



A rendszerszintű modellezésben a *folyamat* az elemi tevékenységek logikailag rendezett sorozata. A *folyamatorientáltság* azt jelenti, hogy a modellezés során a tervező a rendszer által ellátott, a végrehajtás során egymás után elvégzett részfunkciók sorozataként specifikálja a rendszert. Ezt az elemi működési funkciók láncolataképpen felépített modellt egészíti ki mindazzal a járulékos információval, amely a rendszeren belüli különböző funkciók közötti kölcsönhatást okozó mechanizmusok (pl. megosztott erőforrás-használat) jellemzéséhez szükséges.

Egy kis, egyszerű példán a teafőzés rendszerszintű modelljét mutatjuk be. A teafőzés folyamata három tevékenységből áll: a vízforralásból, a filter betételéből és az édesítésből, valamint a csészébe történő kiöntésből. A tevékenységek közül például a vízforraláshoz hozzárendelhetjük a „teáskanna”, „víz”, illetve „tűz” erőforrásokat, valamint a vízforralás idejét.

Tevékenységek

- Modellezés hierarchikus és funkcionális
  - (rész)folyamatok
  - tovább nem bontott, elemi lépések → tevékenységek
- Tevékenység
  - elemi lépés
  - felhasznált erőforrások
  - végrehajtási idő
    - determinisztikus
    - sztochasztikus

Egy-egy részfunkció leírása hierarchikus modellezésnél maga is lehet (rész)folyamat, vagy tovább nem bontott elemi tevékenység.

A tevékenység tehát a modellezés során a modellben tovább nem finomított olyan elemi egység, amelynek hozzárendelt jellemzői a felhasznált erőforrások, illetve az esetleges mennyiségi jellemzők (például a végrehajtási idő, amelyet akár egy determinisztikus, akár egy sztochasztikus paraméter írhat le).

A modell diszkrét idejűsége pedig azt jelenti, hogy az időtengely mentén csak azokat az időpillanatokat kezeljük, amelyekben a rendszerben állapotváltozás történik.

Tevékenységek modellezése Petri hálókban

Petri háló alapú modellek esetén a tevékenység:

- elemi lépés → tranzíció tüzelése
- felhasznált erőforrások → bemeneti / kimeneti helyek
- végrehajtási idő
  - determinisztikus
  - sztochasztikus


determinisztikus időzítésű tranzíció


exponenciális időzítésű tranzíció

Az engedélyezettséggel kapcsolatban felmerülő kérdések:

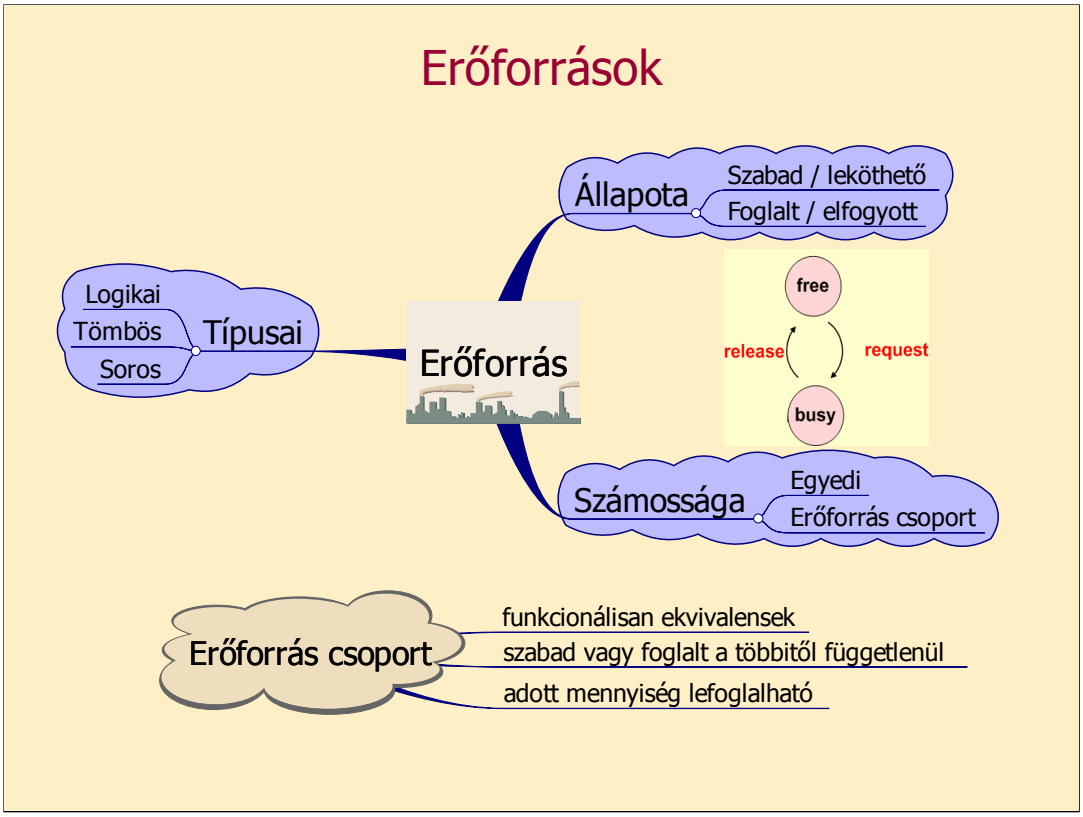
- Időzítetlen tranzíciók „prioritása” nagyobb: előbb tüzelnek
- Engedélyezettség megszűnése: mi lesz a gyújtási idővel?
  - újrainduló (újra sorsol)
  - megőrződő (korábban sorsoltból fennmaradó idő folytatódik)

Abban az esetben, ha a Petri háló tranzícióihoz időzítést rendelünk, prioritási probléma lép fel. El kell dönteni ugyanis, hogy a tüzelésre engedélyezett tranzíciók közül az időzített, vagy a nem időzített tüzeljen először. (Szemantikailag ez annak a dilemmának felel meg, hogy a működés logikai engedélyezettsége vagy a teljesítményviszonyokat származtató időzítés az elsődleges fontosságú.

Az időzített Petri hálók esetében a fenti dilemmát úgy oldják fel, hogy minden esetben a nem időzített tüzelést hajtják végre először, azaz az időzített tüzeléseknek kisebb a prioritása.

Felvetődik az a kérdés, hogy mi történik akkor, ha egy megkezdett időzített tüzelés végrehajtási ideje alatt egy vele konfliktusban levő tüzelés miatt elmúlik a tüzelhetősége, azaz a bemenetén levő helyekről eltűnik az engedélyezéséhez szükséges tokenmennyiség.

Az irodalom erre az esetre nem egységes. Az alternatív definíciók vagy az időzítés újraindulását, vagy a tüzelésből már letelt idő megőrzését (reward saving) írják elő. (A sztochasztikus Petri hálókban leggyakrabban használt exponenciális időzítésnél az örökifjú tulajdonság miatt a két definíció hatása a hálózatjellemzők várható értékeire azonosak.)



A korábbiakban informálisan említett definíció szerint a rendszerben *erőforrás*nak tekintünk minden olyan modellezési elemet, amelyet egyszerre legfeljebb egy tevékenység birtokolhat.

Az erőforrás állapota lehet *szabad* vagy *foglalt*. Elemi erőforrásnak tehát egy egyetlen kétállapotú rendszerkomponenst tekintünk.

Számos feladat esetében azonban a rendszer több, egymással a felhasználás szempontjából ekvivalens erőforrást tartalmazhat, amelyek közül egy új igénylő valamely előre meghatározott darabszámban próbálhatja lefoglalni a számára igényelt mennyiséget. Egyedi erőforrások mellett így a modellezőrendszerek az *erőforrás csoport* használatát is megengedik. Erőforrás csoportnak nevezzük az erőforrások egy olyan halmazát, amelyek egymás között a felhasználás szempontjából ekvivalens viselkedésűek, mindegyikük lehet szabad vagy foglalt a többitől függetlenül, és adott mennyiség lefoglalható belőlük, ha szabadon rendelkezésre áll ennyi.

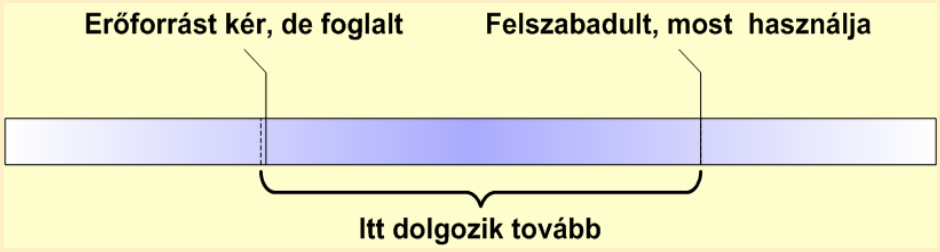
Összefoglalásképpen tehát az erőforrások kétállapotú rendszerek. Állapotuk lehet foglalt vagy szabad, mennyiségük lehet egyedi, illetve többszörös

erőforrás. Az erőforrások és a tevékenységek kapcsolatát két akció jellemzi: egy tevékenység megkísérelhet egy vagy adott mennyiségű erőforrást lefoglalni. Ha akciója sikeres, akkor az adott erőforrás vagy az erőforrás csoportból adott mennyiségű erőforrás foglalttá válik és a tevékenység végrehajtása megkezdődhet.

Blokkoló / nem blokkoló erőforrás

- Blokkoló

- Nem blokkoló



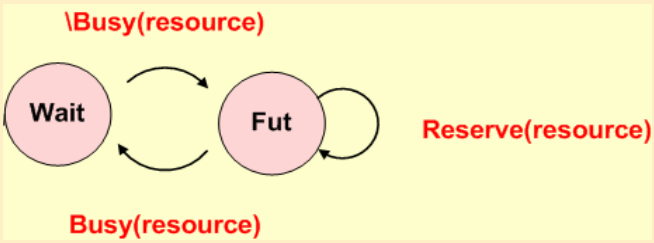
A modellezés során, hasonlóan a jellegzetes szoftvertechnikai megoldásokhoz, megkülönböztethetünk blokkoló és nem blokkoló erőforrás foglalást.

Blokkoló típusú modellezésnél (amely domináns a rendszerszintű modellezésben) abban az esetben, ha az erőforrás lefoglalása nem sikerül, mert az egyedi erőforrás foglalt állapotban van, illetve az erőforráscsoportban nincs a kívánt mennyiségnek megfelelő darabszámú erőforrás, a tevékenység végrehajtása nem kezdődik meg, az felfüggesztődik mindaddig, amíg erőforrás igényét a rendszer ki nem tudja elégíteni.

Nem blokkoló erőforrás foglalás esetén egy speciális üzenet jön létre sikertelenség esetén.

Események

- Rendszerszintű modellezésben eseménynek tekinthető:
  - Valamely tevékenység kezdete, illetve vége
  - Az erőforrások állapotváltozása
- Tevékenység mit tud csinálni?
  - Erőforrást foglal: Reserve (resource_list)
  - Erőforrásra vár: Wait (usage_time)
  - Erőforrást elenged: Release (other_resources_list)



A rendszerszintű modellezésnek egyik fő célja az, hogy a modell dinamikáját vizsgálni lehessen. A diszkrét idejű modellezésnél a keretrendszer futtatórendszerének azokat az időpontokat kell kezelnie, amelyekben a rendszerben állapotváltozás lép fel. *Eseménynek* a konkrét időpillanathoz rendelhető, a rendszer bármely részében fellépő állapotváltozásokat nevezzük.

A rendszerszintű modellezés során az alábbi eseményeket szokás megkülönböztetni:

•Valamely tevékenység kezdete, illetve vége. Ebben az esetben az állapotváltozás azt jelenti, hogy valamely folyamat megkezdődik vagy véget ér, illetve megkezdődhet a folyamatban a befejezett tevékenységre logikailag következő tevékenység végrehajtása. Szokásosan a rendszer specifikációjánál azt tételezik fel, hogy hacsak másképp nem modellezik explicit módon, egy tevékenység végét követően közvetlenül megindulhat és meg is indul a rákövetkező esemény.

•Az erőforrások állapotváltozása. Definíció szerint az erőforrások azok a rendszerben modellezett eszközök, amelyeket egyszerre csak egy tevékenység birtokolhat (azaz az erőforrásokra kölcsönös kizárás áll fenn).

Miután a tevékenységet elemi lépésnek tekintjük, az erőforrások foglalását egy tevékenység megkezdéséhez, elengedését pedig ugyanezen vagy valamely logikailag rákövetkező tevékenység végéhez szokás kötni. Hasonlóan a csatolt folyamatok közötti üzenetváltást is egy tevékenység valamely határpontjához szokás kötni.

Szimuláció során minden egyes erőforráshoz, illetve erőforrás csoporthoz felvehetünk egy olyan listát, amely azt tárolja, hogy az adott erőforrástípusra mely tevékenységek, illetve folyamatok várakoznak. Abban az esetben, ha egy erőforrás típusból az egyedi példány vagy valamennyi mennyiségű csoportos példány felszabadul, a futtatórendszer feladata az, hogy a várakozó tevékenységek közül egyet kiválasszon és elindítson. (A futtatórendszer így tulajdonképpen erősen hasonlít egy valósidejű operációs rendszer futtatórendszerére azzal a különbséggel, hogy a futtatórendszert mozgató események itt szimulált események.)

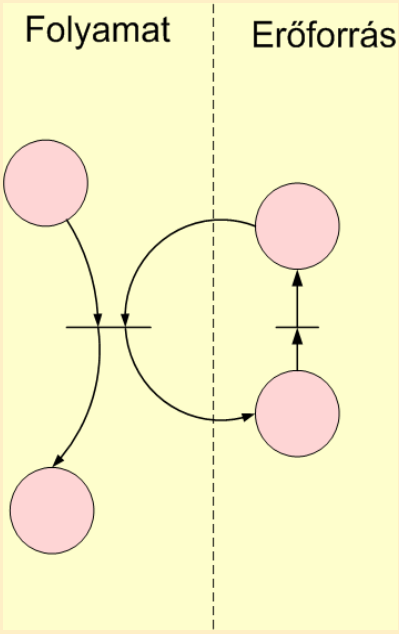
Erőforrás allokáció Petri hálókban

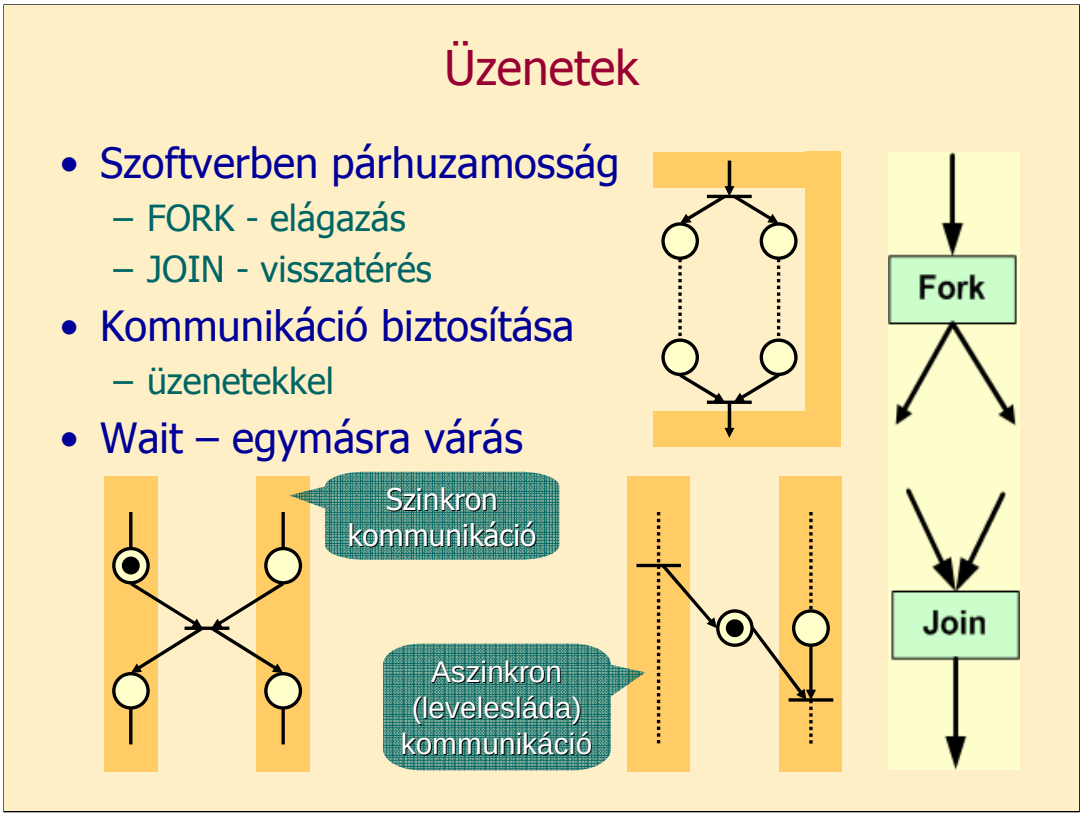
- Kölcsönös kizárás
- Több darab lefoglalása

Cél:

nem a minimális, hanem az

ÉRTHETŐ PETRI HÁLÓ!





A párhuzamosan futó folyamatok szinkronizálására ismert módon fork és join struktúrákat szokás használni. A szinkron, illetve aszinkron kommunikáció esetén sormintaszerűen modellezhető az üzenetváltás. Szinkron esetben a küldő folyamat mindaddig megáll és várakozik, amíg a vevőfolyamat ezt az üzenetet el nem vette, aszinkron esetben pedig az üzenet elküldése után a folyamat szabadon továbbhaladhat.

Rendszermodellezés

Modellépítés folyamata:

- a három fő modellelem-fajta:
  - folyamatok, illetve tevékenységek
  - erőforrások
  - a logikailag csatolt folyamatok közti esetleges üzenetek
- felépítése először egyedileg
- majd ezekből a modell összeintegrálása

A rendszermodellezés során a következő metodika használatát javasoljuk, amely a rendszerekben a három fő modellelem-fajtát (folyamatok, illetve tevékenységek, erőforrások, a logikailag csatolt folyamatok közötti esetleges üzenetek) először egyedileg építi fel, majd a modellt ezekből integrálja össze:

1. A folyamat modellje (egyelőre az erőforrás használat, illetve üzenetváltás részletes feltűntetése nélkül). Ebben a modellben az erőforrások foglalása, elengedése illetve az üzenetek küldése és fogadása csak mint egy-egy erőforrás illetve üzenet pufferhez nem kapcsolt állapotátmenet szerepel.

2. Minden egyes erőforráshoz fel kell építeni az ő foglalt/szabad állapotát jellemző kétállapotú véges automata modellrészletet, valamint az üzenetek pufferjének modelljét. Ezekben ugyancsak állapotátmenetek jelzik a szabaddá/foglalttá illetve üressé/telivé válást.

3. A folyamat és erőforrás (illetve az üzenetek pufferjének) modelljében a megfelelő állapotátmenetek fork-szerű összevonása. Például egy tevékenység egy általa használandó erőforrás blokkoló lefoglalása esetén akkor kezdődhet meg, ha az erőforrás egyidejűleg foglalt → szabad átmenetet hajt végre. Abban az esetben, ha egy erőforrást vagy üzenetpuffert több folyamat is használhat, természetesen mindegyikükhöz megfelelő szabaddá/foglalttá illetve üressé/telivé tevő állapotátmenet hozandó létre.