Gyártórendszerek dinamikája

II. rész

Erőforrás-korlátozás nélküli statikus ütemezés
Kritikus út módszer

CPM: Critical Path Method

Adott

- az ütemezendő elemi műveletek halmaza (activity)
- minden elemi művelet előfeltételei (szintén elemi műveletek) *parciális rendezés* (egymásutániság)
- minden elemi művelet *végrehajtási ideje*

Táblázatba rendezhető

Grafikus leírás: CPM gráf

- *élei megfelelnek az elemi műveleteknek* (egy-egy értelmű megfeleltetés)
- csúcsok megfelelnek eseményeknek (rendszer-állapotok bekövetkezésének)

A kritikus út kiszámítása

Algoritmikusan, két menetben

I. Előrefelé haladó számítás: legkorábbi bekövetkezési idők
   1. A kezdő esemény kezdeti idejét nullára állítjuk
   2. Minden műveletet akkor kezdünk, ha az előfeltételek teljesültek
   3. Minden esemény legkorábbi bekövetkezési ideje (E_i) a beléje vezető műveletek befejeződési idejeinek ***maximuma***

II. Visszafelé haladó számítás: legkésőbbi bekövetkezési idők
   1. A befejező esemény legkésőbbi bekövetkezési idejét a legkorábbira állítjuk
   2. Minden művelet legkésőbbi bekövetkezési idejét a vég-esemény legkésőbbi bekövetkezési idejéből számítjuk, levonva belőle a művelet végrehajtási idejét
   3. Minden esemény legkésőbbi bekövetkezési ideje (L_i) a belőle induló műveletek kezdő idejeinek ***minimuma***

A kritikus út kiszámítása

Csúszási idők

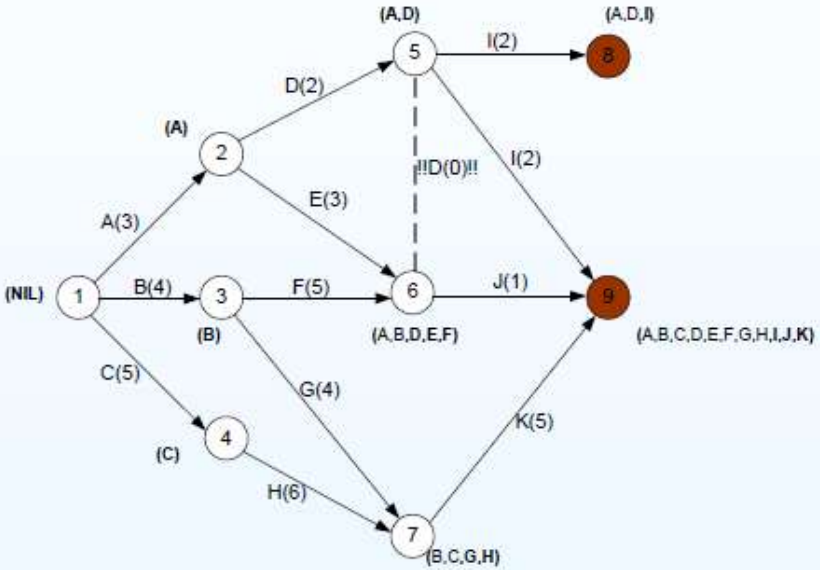
1. Eseményekre: a legkésőbbi és a legkorábbi bekövetkezési idő különbsége, $S_i = L_i - E_i$
2. Az i-edik eseményből a j-edikbe vezető műveletre:

$$S_{ij} = L_j - E_i - t_{ij}$$

ahol t_{ij} a művelet végrehajtási ideje

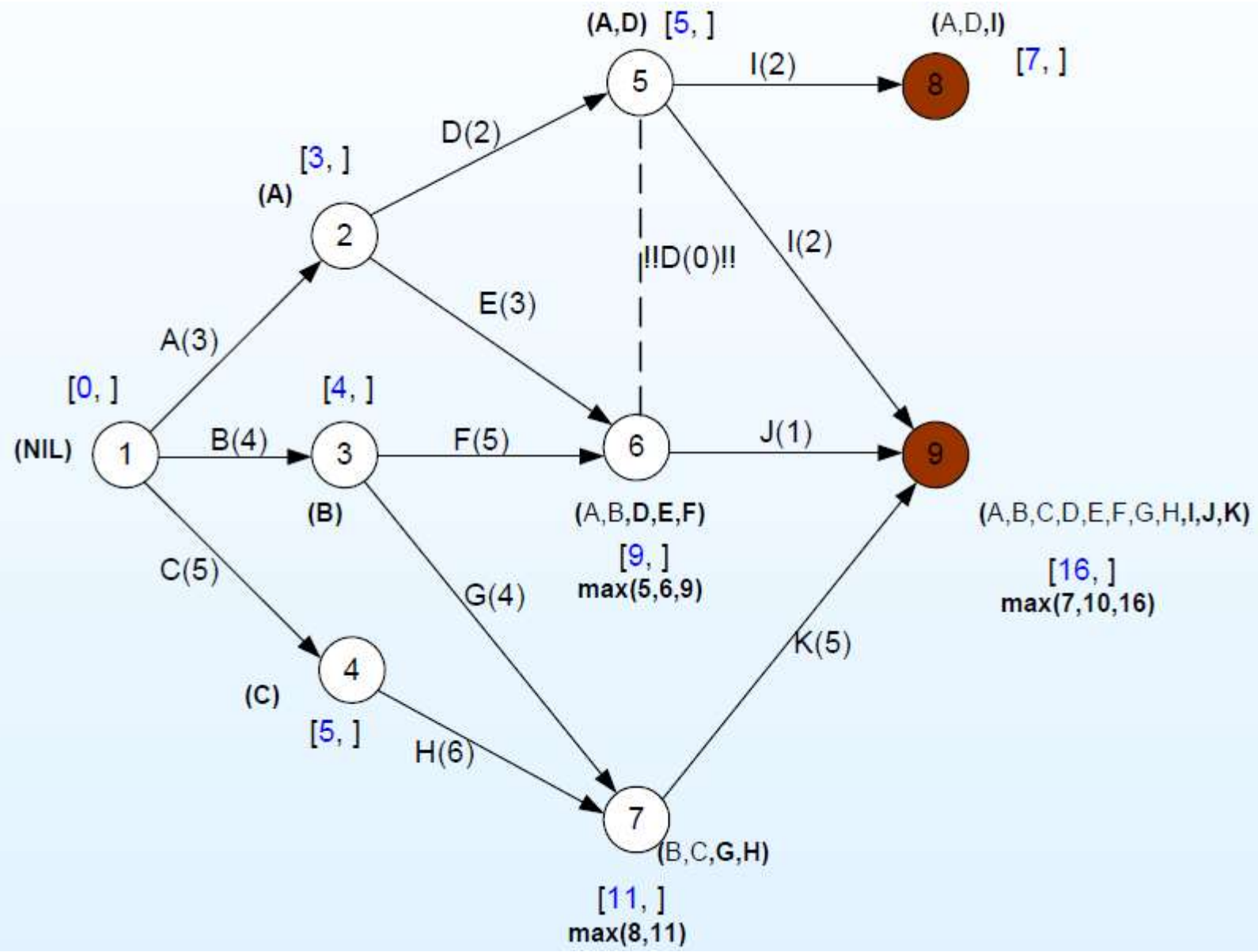
Kritikus út: azon műveletekből (élekből) áll, amelyek csúszási ideje nulla

Példa

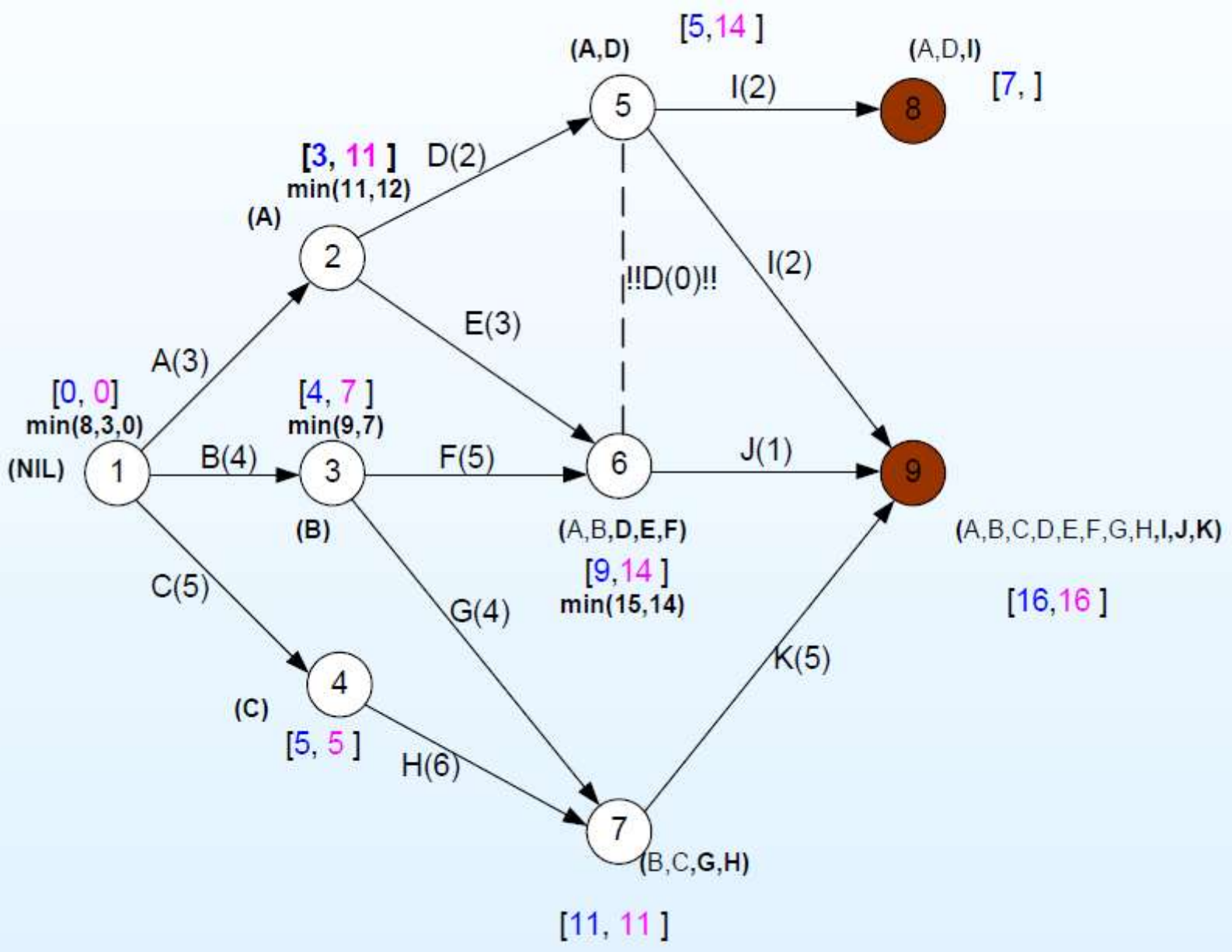


Művelet	Előfeltételek	Idő (min)	Művelet	Előfeltételek	Idő (min)
A	-	3	G	B	4
B	-	4	H	C	6
C	-	5	I	D	2
D	A	2	J	D, E, F	1
E	A	3	K	G, H	5
F	B	5	GOAL	*	

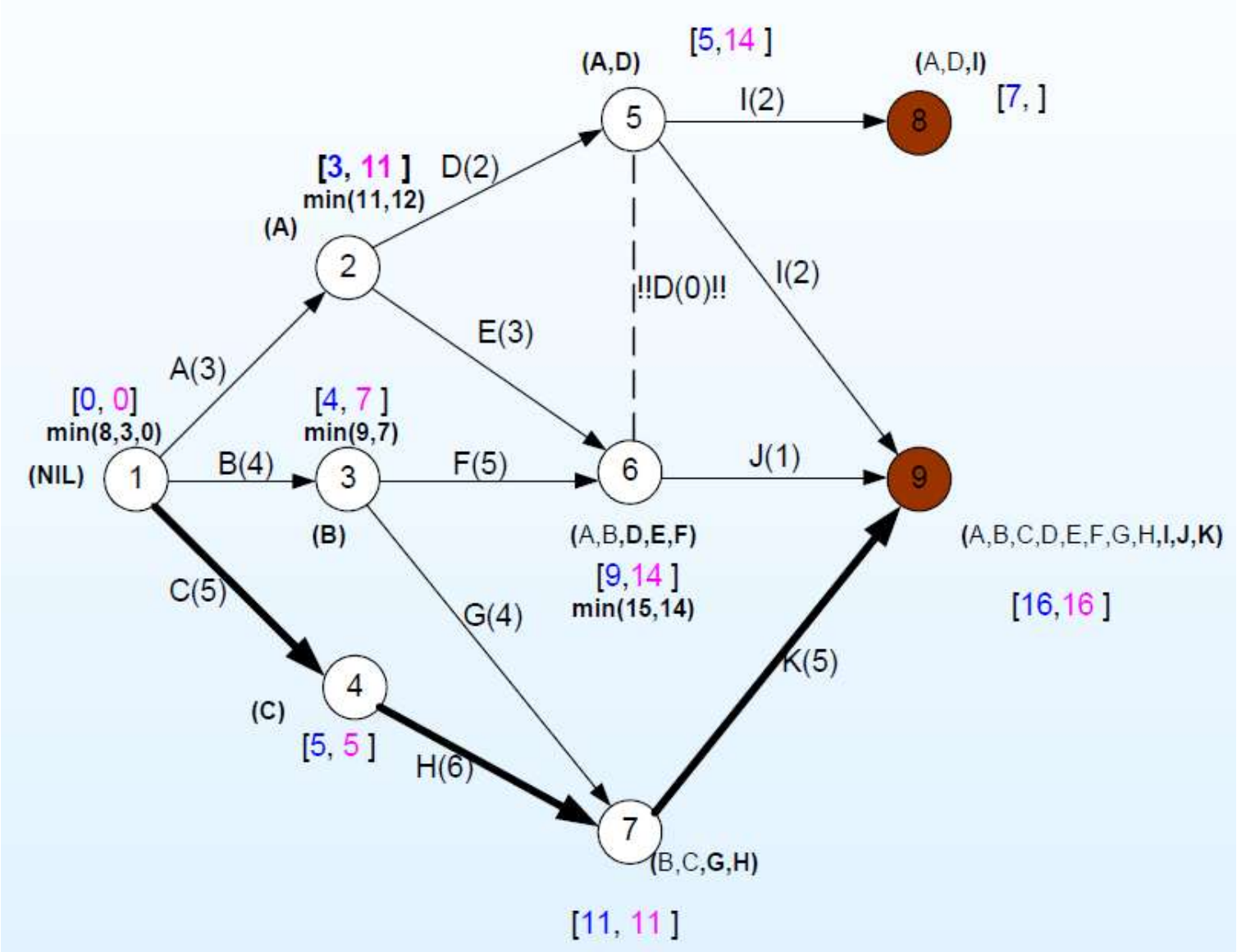
Példa – számolás balról jobbra



Példa – számolás jobbról balra



Kritikus út, kritikus műveletek



Dinamikus programozás - gyártásütemezés

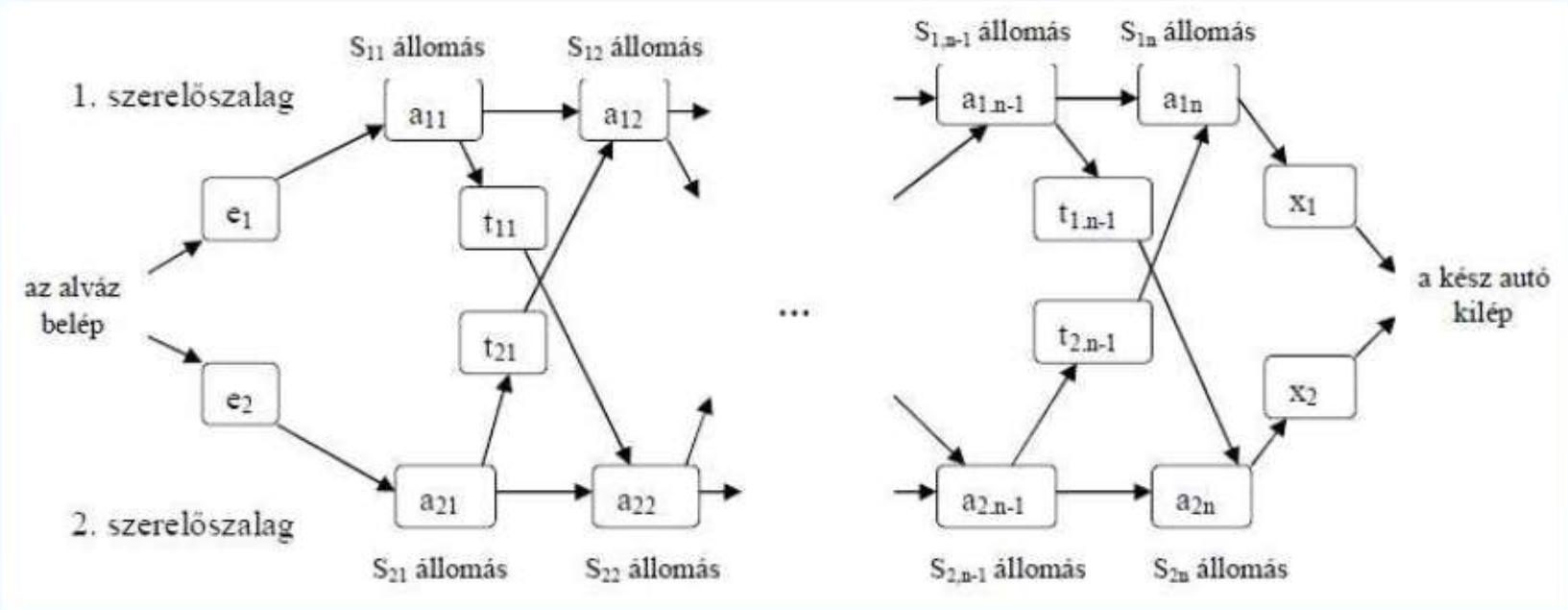
Egy dinamikus programozási algoritmus kifejlesztése 4 lépésre bontható fel:

1. *Jellemezzük az optimális megoldás szerkezetét.*
2. *Rekurzív módon definiáljuk az optimális megoldás értékét.*
3. *Kiszámítjuk az optimális megoldás értékét alulról felfelé történő módon.*
4. *A kiszámított információk alapján megszerkesztünk egy optimális megoldást.*

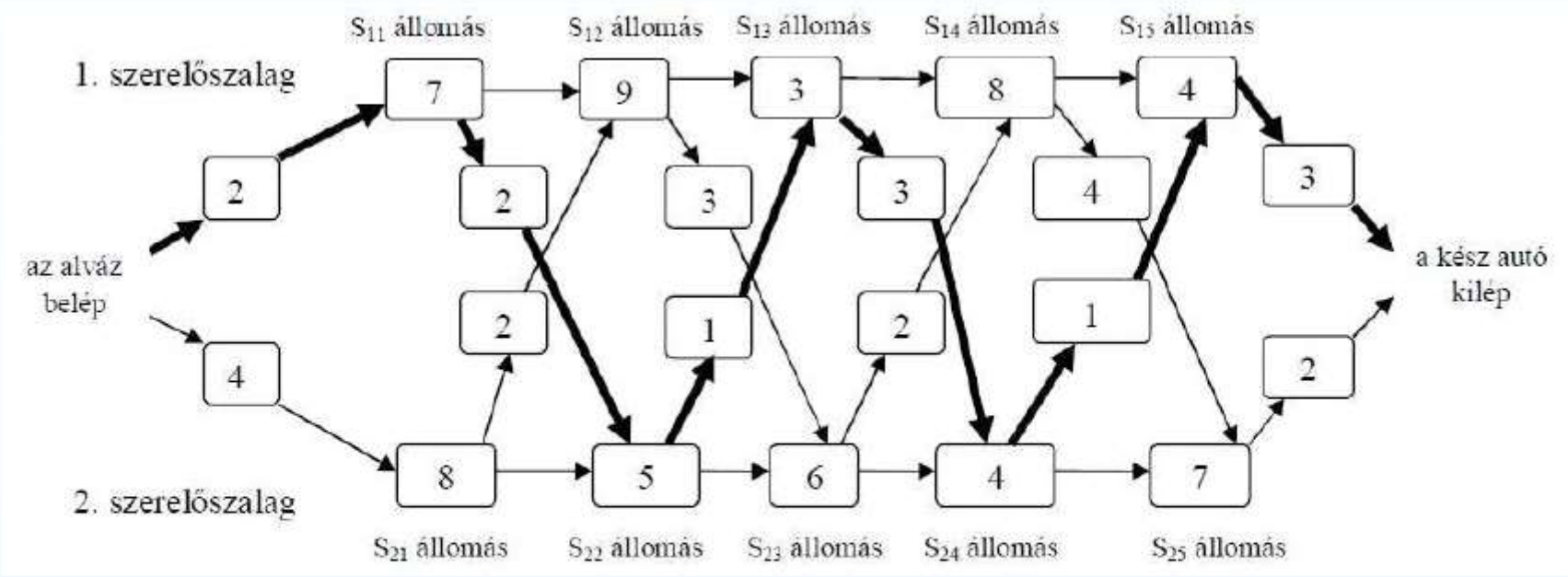
A dinamikus programozás minden egyes részfeladatot és annak **minden részfeladatát pontosan egyszer oldja meg**, **az eredményt egy táblázatban tárolja**, és ezáltal elkerüli az ismételt számítást, ha a részfeladat megint felmerül.

Szerelőszalag ütemezése

Mindegyik szalagnak n állomása van, melyek indexei $j = 1, \ldots, n$. S_{ij} jelöli az i-edik ($i = 1$ vagy 2) szalag j-edik állomását. Az első szalag j-edik állomása (S_{1j}) ugyanazt a funkciót látja el, mint a második szalag j-edik állomása (S_{2j}). Az S_{ij} állomáson a műveleti időt a_{ij} jelöli.



Példa



A teljes leszámlálás számítási ideje $\Omega(2^n)$. Ez nagy n mellett elfogadhatatlan.

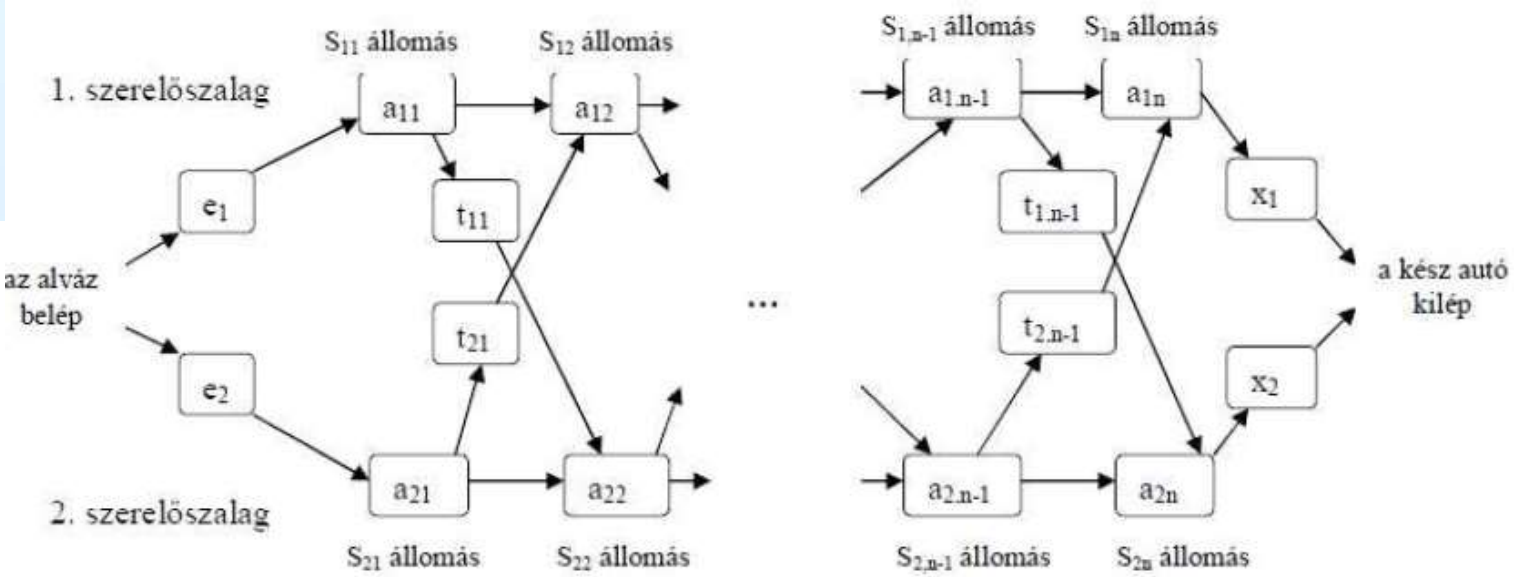
A rekurzív megoldás

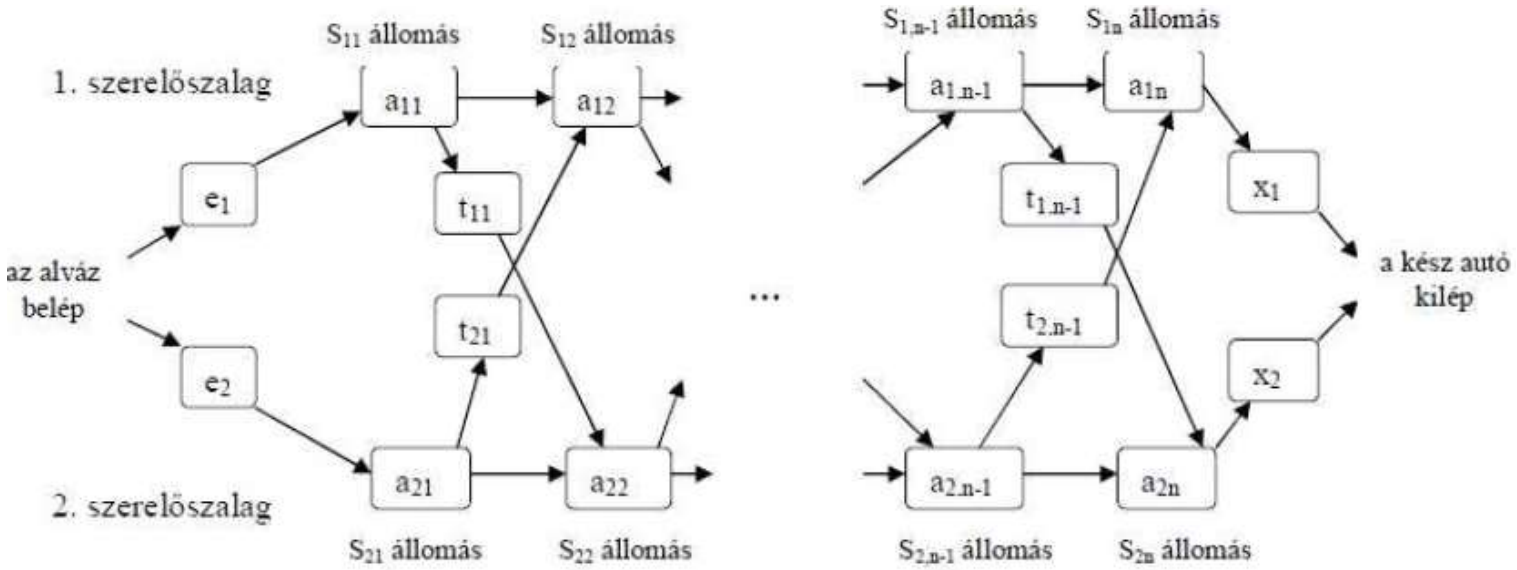
z optimális megoldás értékét a részfeladatok optimális
tékeiből rekurzívan fejezzük ki.
részfeladatok legyenek mindkét szalag j-edik állomásán való
ggyorsabb áthaladás megkeresésének a feladatai $j = 1, \ldots, n$
ellett.
elölje $f_i[j]$ azt a legrövidebb időt, ami alatt az alváz túl tud jutni
: S_{ij} állomáson.
**élunk annak a legrövidebb időnek a meghatározása, ami
att az alváz az egész üzemen keresztül tud haladni.**
zt az időt f^* jelöli.

$$= min\{f_1[n] + x_1, f_2[n] + x_2\}$$

$$[1] = e_1 + a_{11}$$

$$[1] = e_2 + a_{21}$$





Hogyan kell kiszámolni az $f_i[j]$-t $j = 2, \ldots, n$ és $i = 1, 2$ esetén?

$f_1[j] = min\{f_1[j-1] + a_{1j}, f_2[j-1] + t_{2,j-1} + a_{1j}\}$, ha $j = 2, \ldots, n$.

$f_2[j] = min\{f_2[j-1] + a_{2j}, f_1[j-1] + t_{1,j-1} + a_{2j}\}$, ha $j = 2, \ldots, n$.

A következő rekurzív egyenletet kapjuk:

$$f_1[j] = \begin{cases} e_1 + a_{11}, \text{ ha } j = 1, \\ min\{f_1[j-1] + a_{1j}, f_2[j-1] + t_{2,j-1} + a_{1j}\}, \text{ ha } j \geq 2 \end{cases}$$

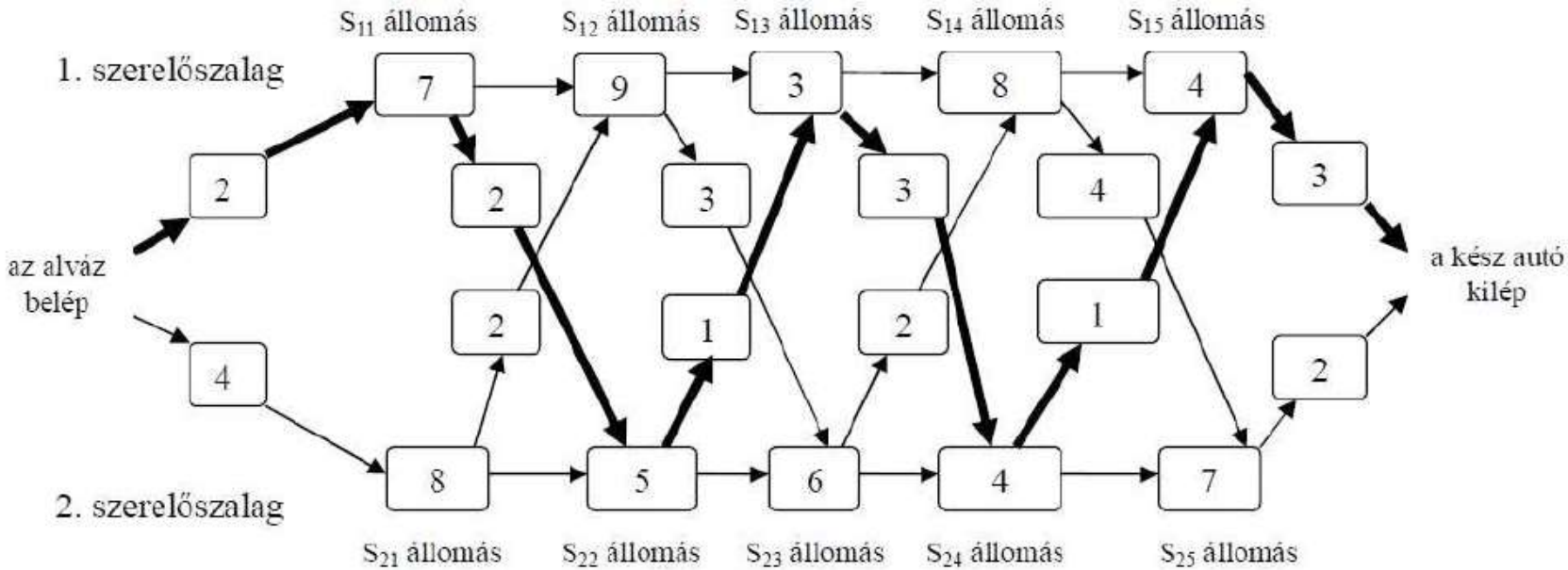
$$f_2[j] = \begin{cases} e_2 + a_{21}, \text{ ha } j = 1, \\ min\{f_2[j-1] + a_{2j}, f_1[j-1] + t_{1,j-1} + a_{2j}\}, \text{ ha } j \geq 2 \end{cases}$$

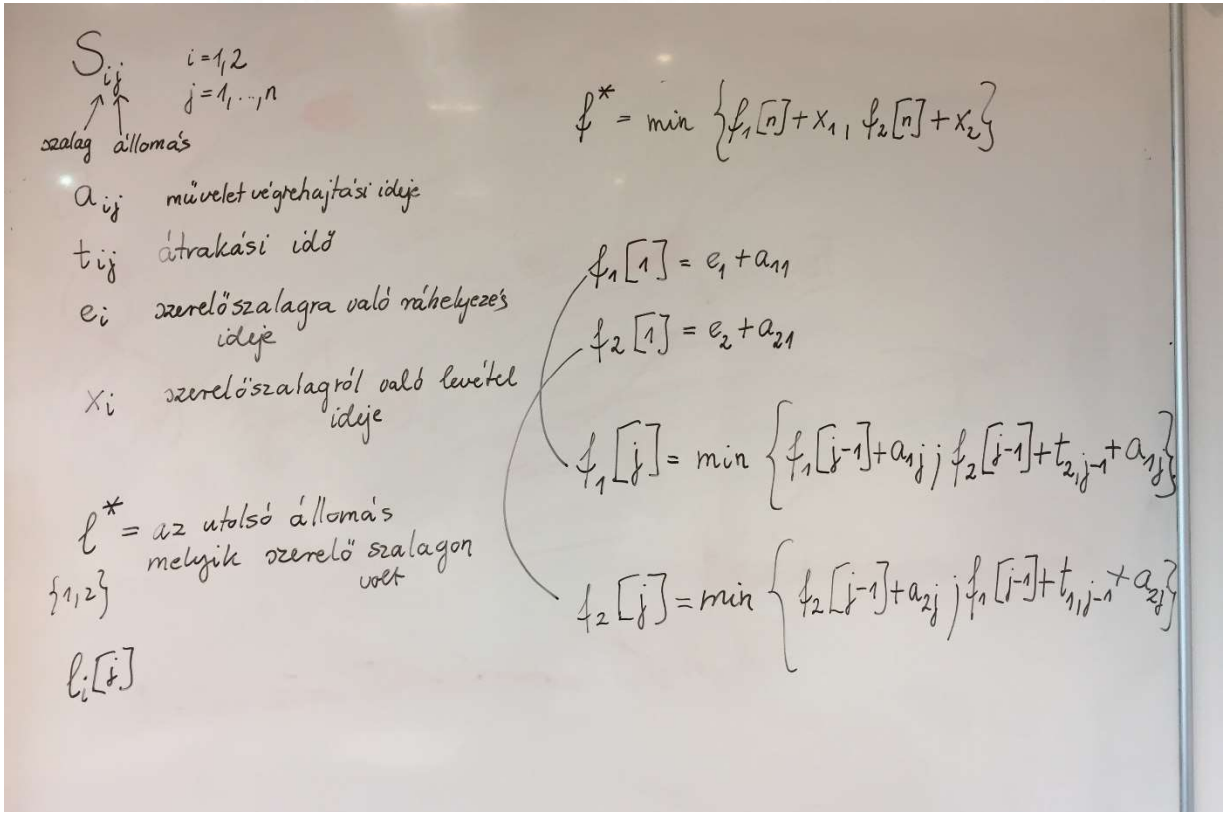
Az $f_i[j]$ mennyiségek a részfeladatok optimális érékei.

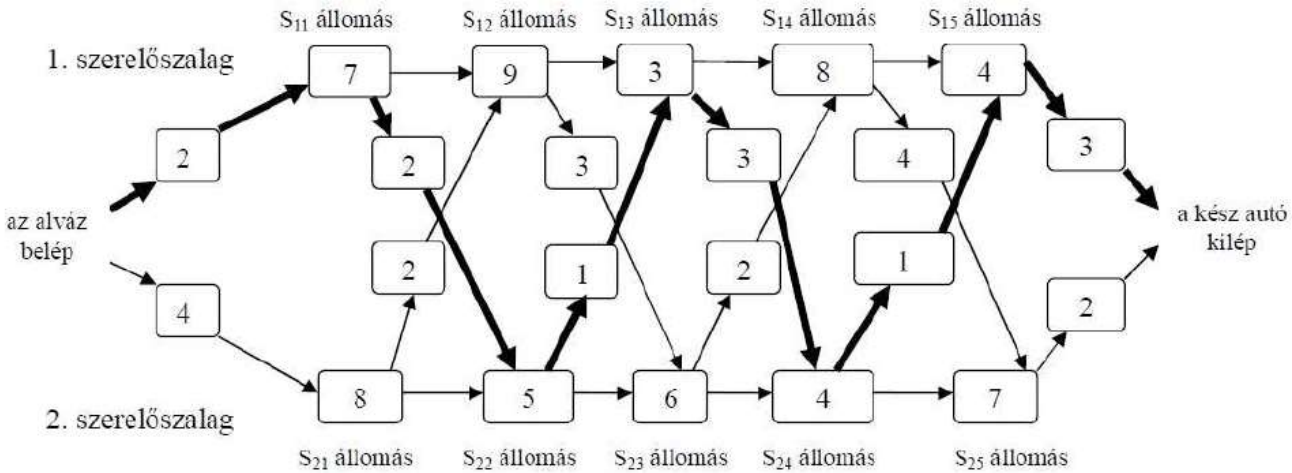
Annak érdekében, hogy vissza tudjuk keresni a legrövidebb utat, definiáljuk $l_i[j]$-t mint azt a szerelőszalagot, amelyiknek a $j-1$-edik állomását használtuk az S_{ij}-n való leggyorsabb keresztülhaladáskor.

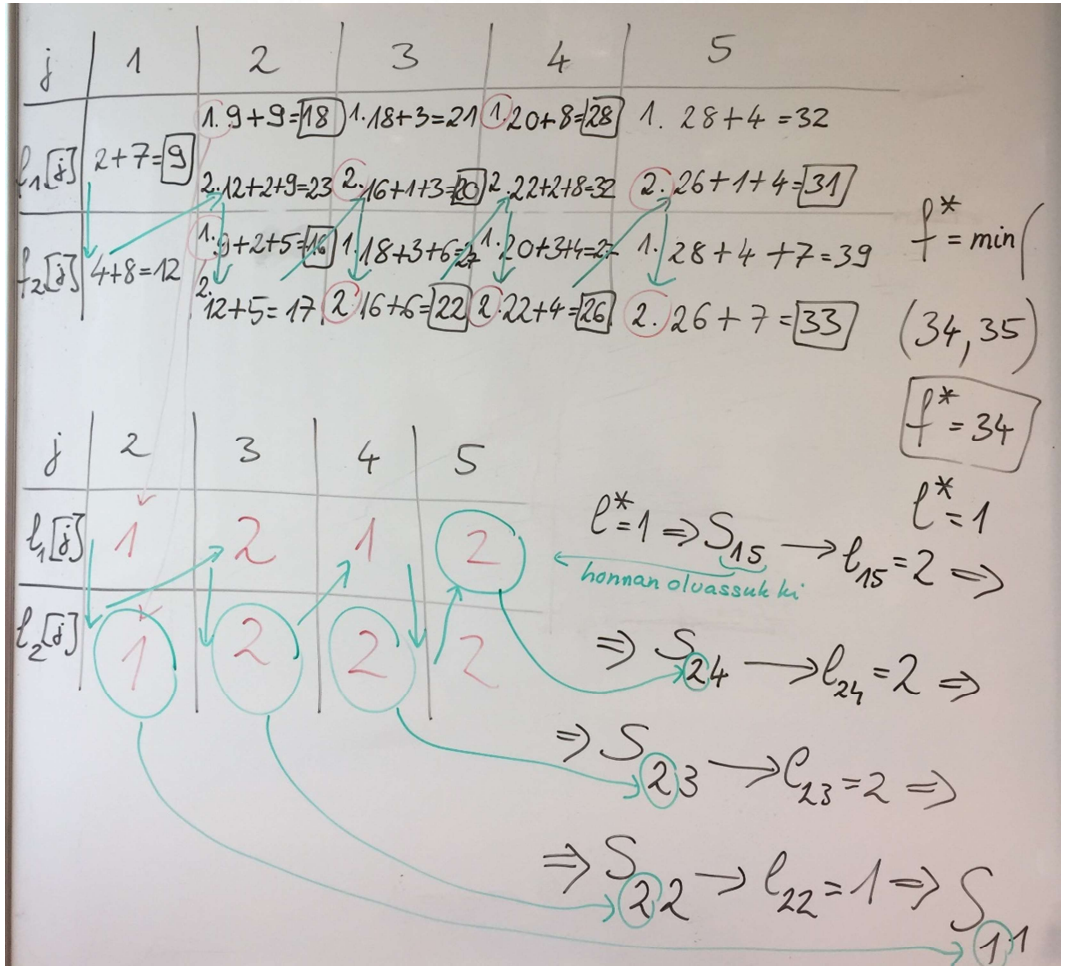
Itt $i = 1, 2$ és $j = 2, \ldots, n$.
(Nem definiáljuk $l_i[1]$-et, mert S_{i1}-t egyik szalagon sem előzi meg másik állomás.)

Az l^* szalag definíció szerint az, amelyiknek az utolsó állomását használjuk az egész üzemen való áthaladáskor.
Az $l_i[j]$ értékek segítségével lehet nyomon követni a legrövidebb utat.


S11 állomás
S12 állomás
S13 állomás
S14 állomás
S15 állomás
1. szerelőszalag
2. szerelőszalag
az alváz belép
a kész autó kilép
7
9
3
8
4
2
2
3
3
4
3
2
2
1
2
1
2
4
8
5
6
4
7
S21 állomás
S22 állomás
S23 állomás
S24 állomás
S25 állomás


S_{ij} $i=1,2$ $j=1,\dots,n$
szalag állomás
a_{ij} művelet végrehajtási ideje
t_{ij} átrakási idő
e_i szerelőszalagra való ráhelyezés ideje
x_i szerelőszalagról való levétel ideje
ℓ^* = az utolsó állomás melyik szerelő szalagon volt
$\{1,2\}$
$\ell_i[j]$
$f^* = \min\{f_1[n]+x_1, f_2[n]+x_2\}$
$f_1[1] = e_1 + a_{11}$
$f_2[1] = e_2 + a_{21}$
$f_1[j] = \min\{f_1[j-1]+a_{1j}; f_2[j-1]+t_{2,j-1}+a_{1j}\}$
$f_2[j] = \min\{f_2[j-1]+a_{2j}; f_1[j-1]+t_{1,j-1}+a_{2j}\}$


S11 állomás
S12 állomás
S13 állomás
S14 állomás
S15 állomás
1. szerelőszalag
2. szerelőszalag
az alváz belép
a kész autó kilép
7
9
3
8
4
2
2
3
3
4
3
2
2
1
2
1
2
4
8
5
6
4
7
S21 állomás
S22 állomás
S23 állomás
S24 állomás
S25 állomás

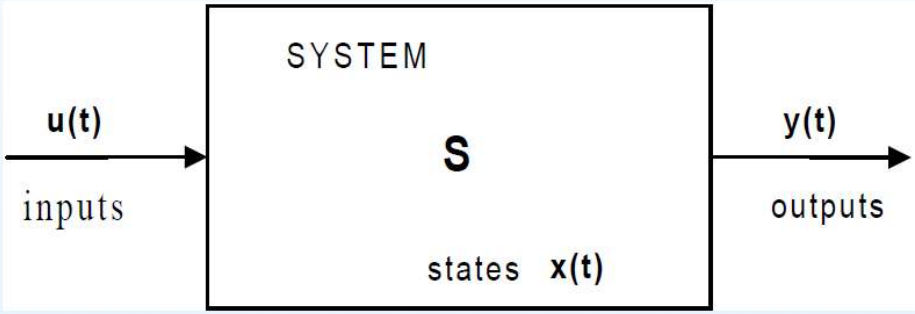

j | 1 | 2 | 3 | 4 | 5
l1[j] | 2+7=9 | 1. 9+9=18 | 1. 18+3=21 | 1. 20+8=28 | 1. 28+4=32
2. 12+2+9=23 | 2. 16+1+3=20 | 2. 22+2+8=32 | 2. 26+1+4=31
l2[j] | 4+8=12 | 1. 9+2+5=16 | 1. 18+3+6=27 | 1. 20+3+4=27 | 1. 28+4+7=39
2. 12+5=17 | 2. 16+6=22 | 2. 22+4=26 | 2. 26+7=33
f* = min(34, 35)
f* = 34
j | 2 | 3 | 4 | 5
l1[j] | 1 | 2 | 1 | 2
l2[j] | 1 | 2 | 2 | 2
l* = 1
l* = 1 ⇒ S15 → l15 = 2 ⇒
honnan olvassuk ki
⇒ S24 → l24 = 2 ⇒
⇒ S23 → l23 = 2 ⇒
⇒ S22 → l22 = 1 ⇒ S11

Irányítástechnikai alapfogalmak

Rendszer (**S**): *jelek*en végez műveletet

$$y = \mathbf{S}[u]$$

- bemenetek (u) és kimenetek (y)
- állapot-változók (x)



A rendszer jel-folyam ábrája

Adott

- a rendszermodell (paraméterekkel együtt)
- ***irányítási cél***

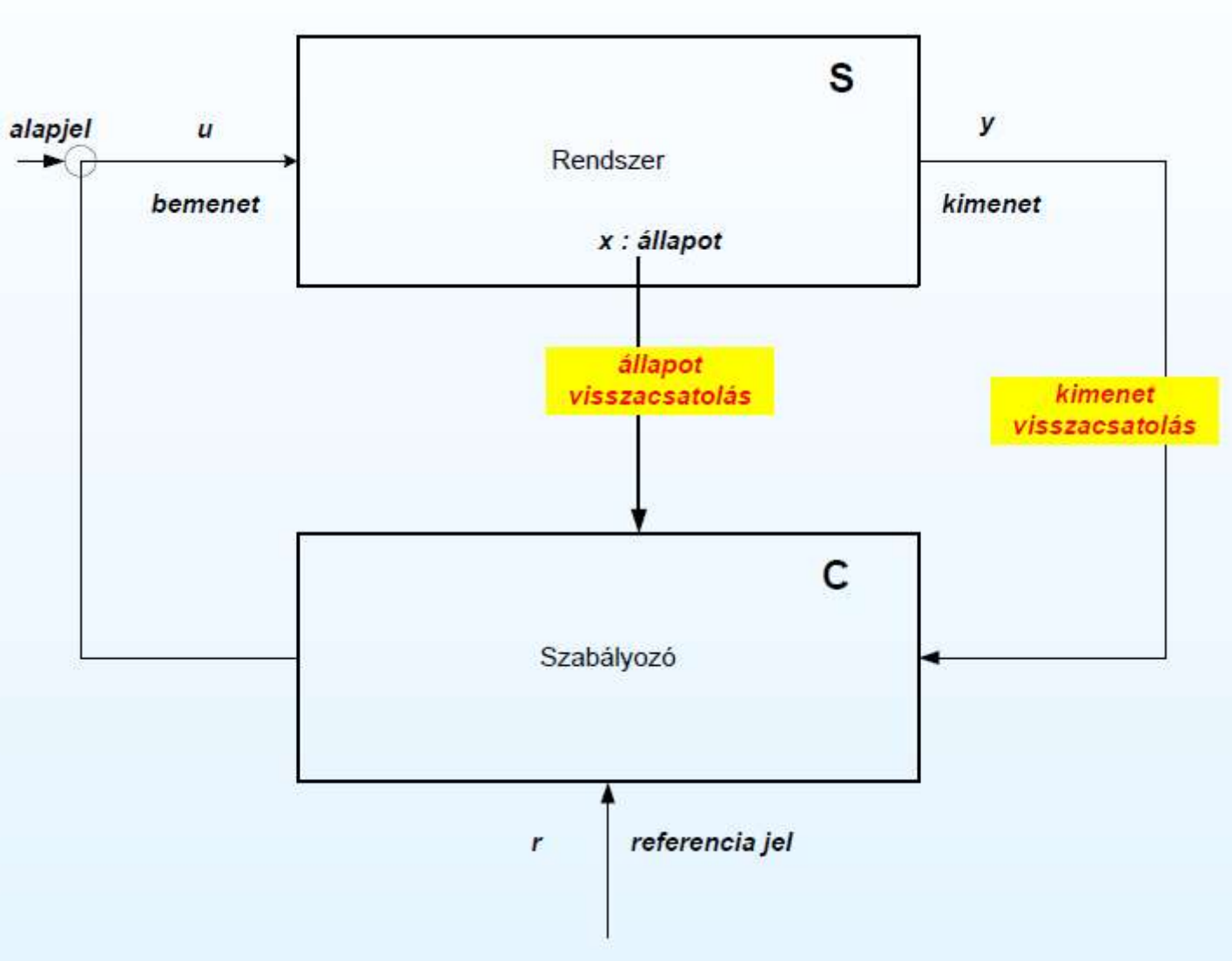
Kiszámítandó

az a ***bemeneti időfüggvény*** amellyel elérhető az irányítási cél

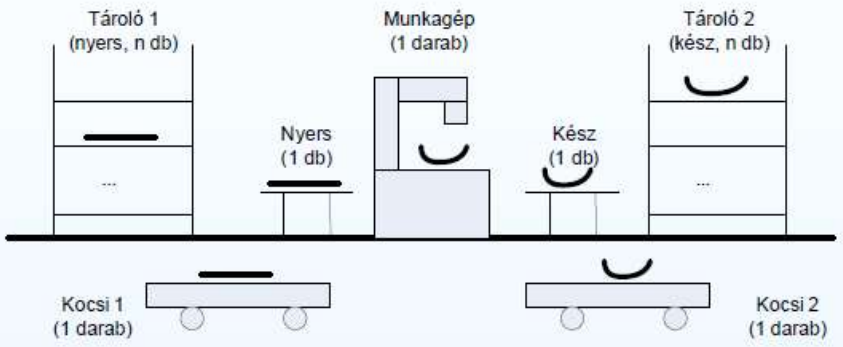
Tipikus irányítási célok:

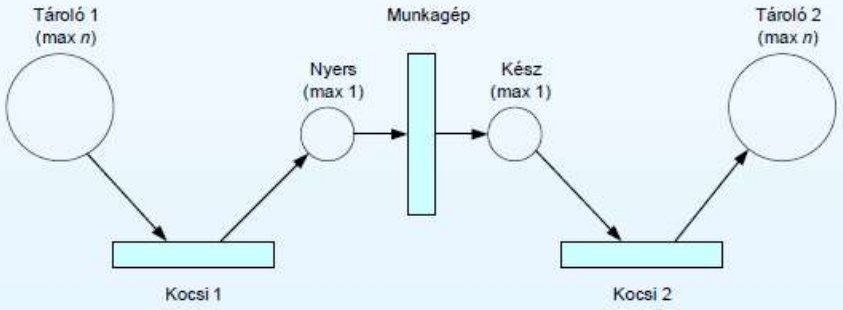
- stabilizálás (szabályozás)
- zavarelnyomás
- optimális irányítás (legrövidebb idő alatt, legkisebb energia-befektetéssel, stb.)

Visszacsatoló szabályozás jel-folyam ábrája



Egyszerűbb gyártórendszer



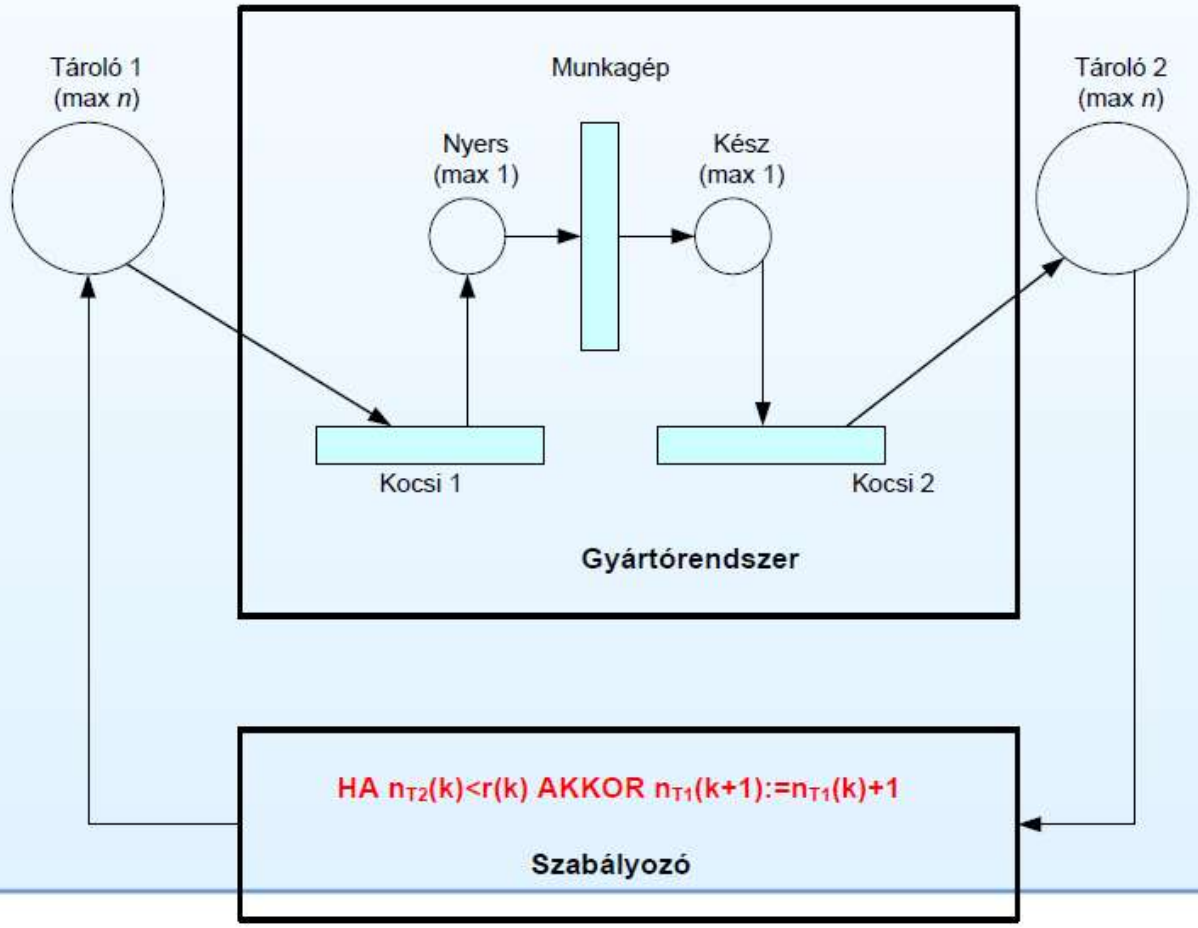


Bemenet: nyersanyag munkadarabok száma "Tároló 1"-en, n_{T1}
Kimenet: késztermék munkadarabok száma "Tároló 2"-en, n_{T2}
Állapotok: munkadarabok száma a "Nyers" és "Kész" tárolókon n_N, n_K

Egyszerűbb gyártórendszer szabályozása

Szabályozási cél: 10 percenként egy tál készüljön el (mintavételi idő 10 perc)

Kimenet visszacsatolásos szabályozó



Összetettebb szabályozás – Egy bemenetű egy kimenetű (SISO) szabályozók

Egy rendszer bemenet kiszámítása egy rendszer kimenet visszacsatolásával

Jelek:

- szabályozott jellemző (rendszer kimenet): $y(t)$
- referencia jel : $r(t)$
- beavatkozó változó (rendszer bemenet): $u(t)$

Leggyakrabban: **PID szabályozó**

$$u(t) = C_P(y(t) - r(t)) + C_I \int_0^t (y(\tau) - r(\tau))d\tau + C_D \frac{d(y(t) - r(t))}{dt}$$

Dinamikus kimenet visszacsatolás

Szervo-szabályozás: az $r(t)$ referencia jel *időfüggő*

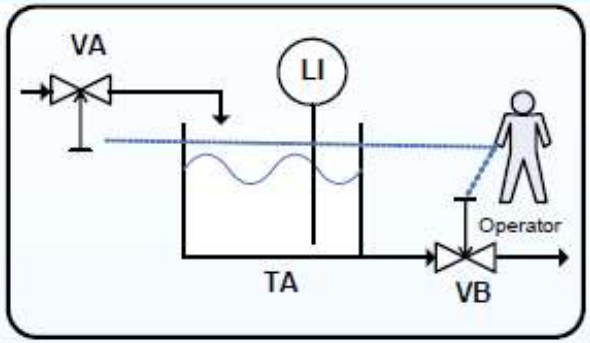
Operátori eljárások

Diszkrét, soros és párhuzamos lépéseket is tartalmazó **vezérlési műveletek**

Fajtái:

- indítási, leállítási, üzemmód-váltási
- biztonsági
- szabályozási

Elemei: diszkrét lépések



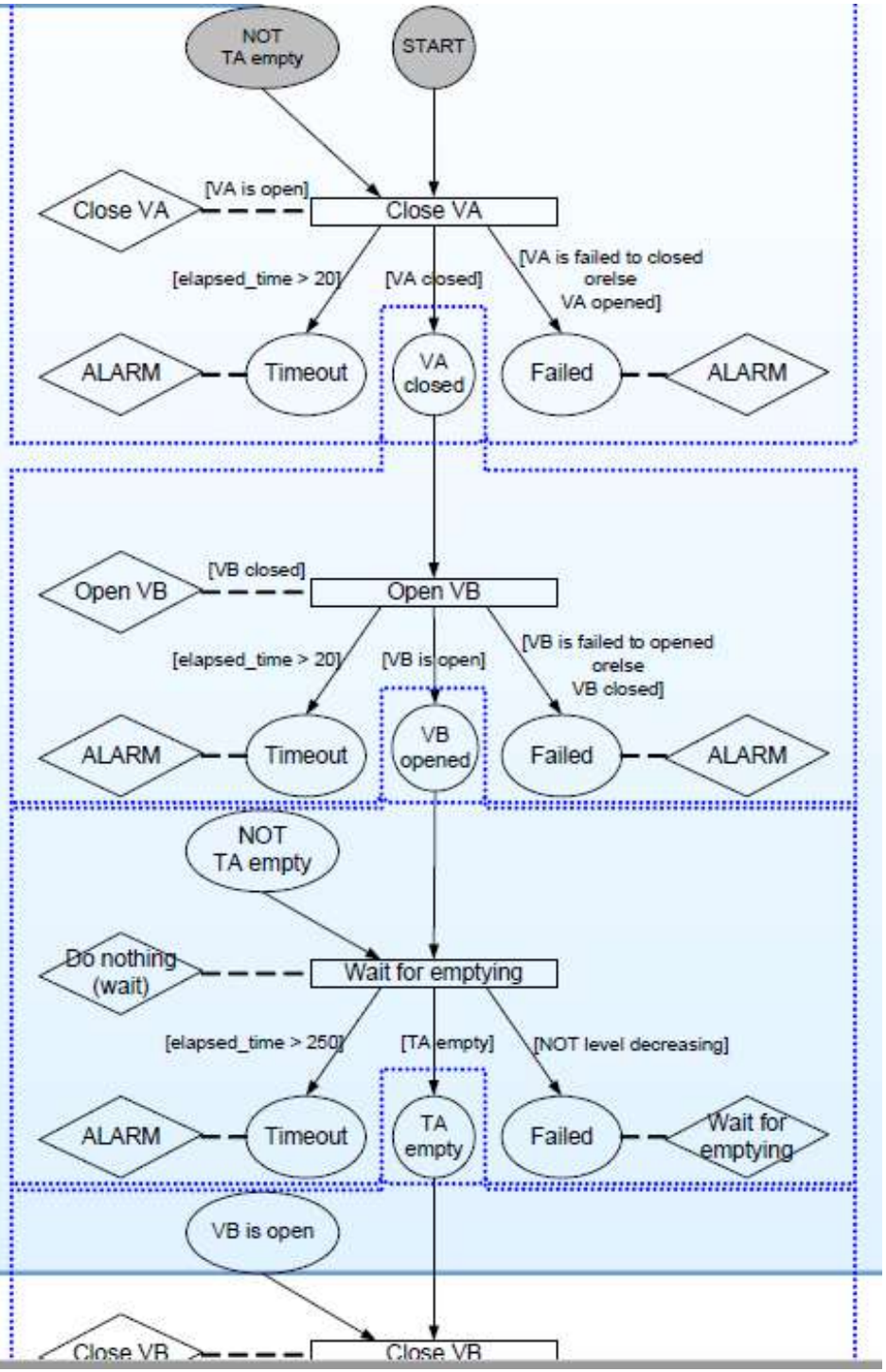
Procedure

Empty the tank TA for maintenance

Steps:

1. Close valve VA
2. Open valve VB
3. Check level decreasing in every 30 timestep (i.e. wait for emptying)
4. If tank TA is empty, close valve VB

* If anything goes wrong ring an ALARM



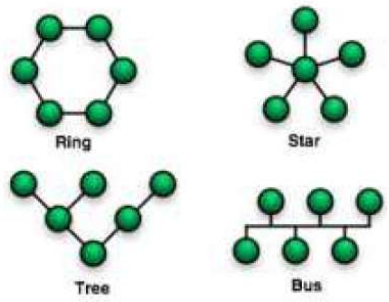
Számítógép hálózatok komponensei

Hálózati **hardver**

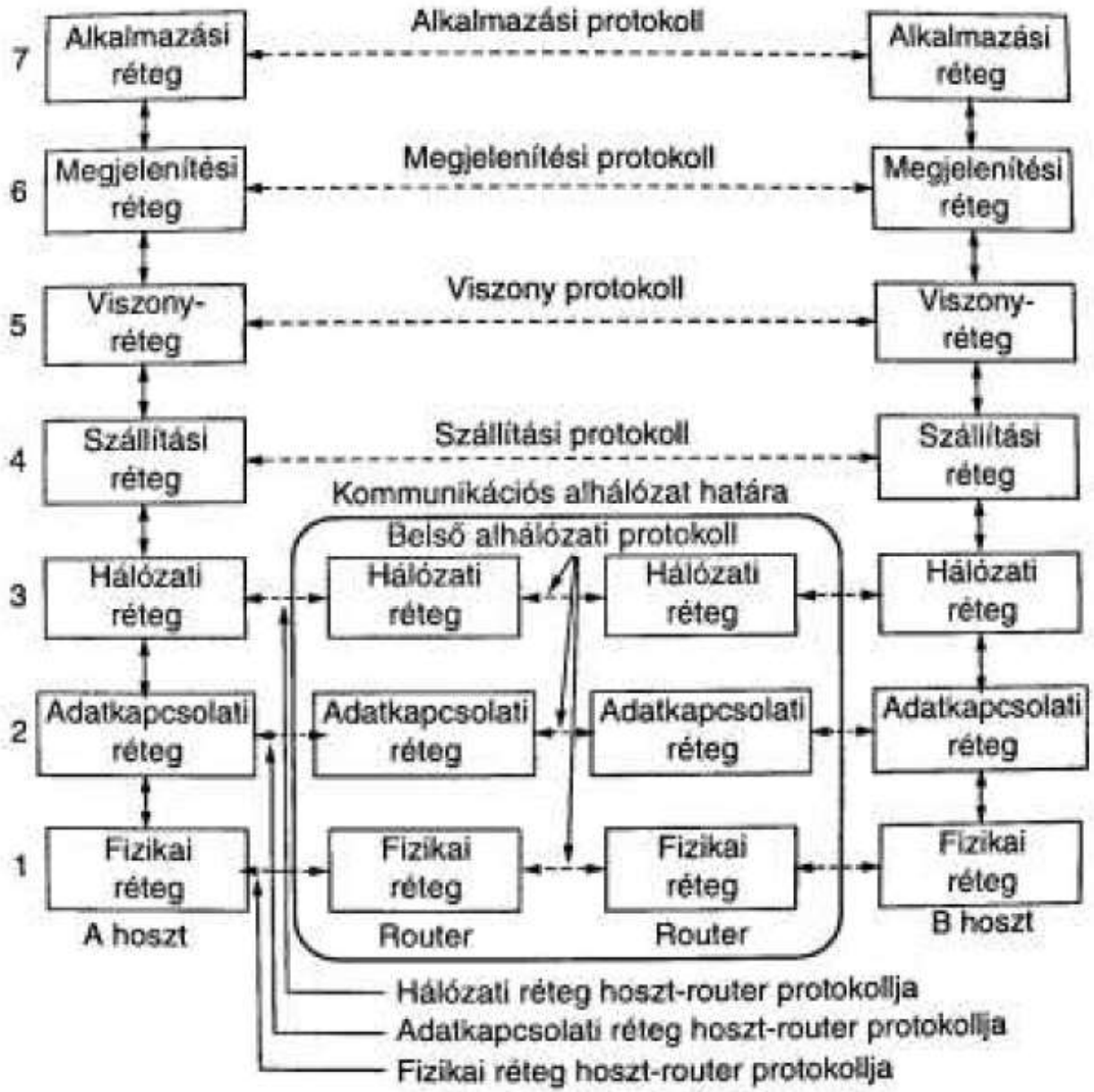
- számítógépek (aktív) és összeköttetések (passzív)
- méret, pl. LAN (local area network)
- átviteli technológia: adatszóró (vagy többszörös hozzáférésű) hálózat, kétpontos hálózat
- topológia: összeköttetés típusa, gépek logikai szerepe
- csatornakiosztási módszerek (adatszóró hálózatokban)

Hálózati **szoftver**

- réteges szerkezet
- az OSI (open systems interface) modell

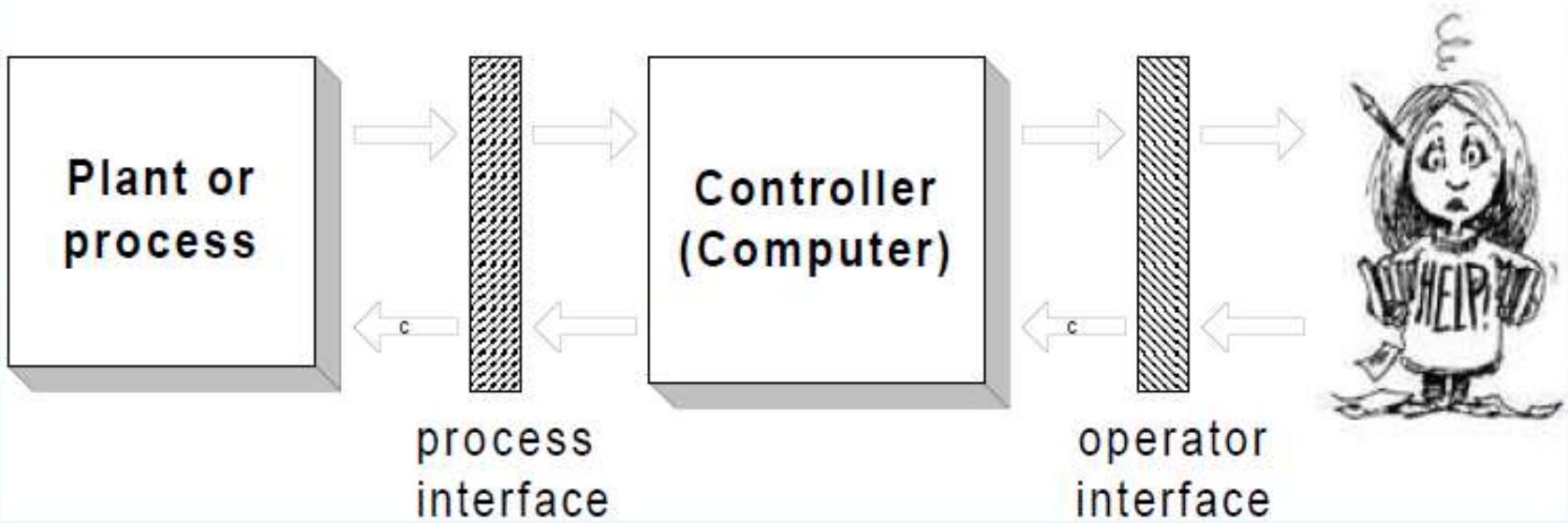


OSI modell


7
Alkalmazási réteg
Alkalmazási protokoll
Alkalmazási réteg
6
Megjelenítési réteg
Megjelenítési protokoll
Megjelenítési réteg
5
Viszony-réteg
Viszony protokoll
Viszony-réteg
4
Szállítási réteg
Szállítási protokoll
Szállítási réteg
Kommunikációs alhálózat határa
Belső alhálózati protokoll
3
Hálózati réteg
Hálózati réteg
Hálózati réteg
Hálózati réteg
2
Adatkapcsolati réteg
Adatkapcsolati réteg
Adatkapcsolati réteg
Adatkapcsolati réteg
1
Fizikai réteg
Fizikai réteg
Fizikai réteg
Fizikai réteg
A hoszt
Router
Router
B hoszt
Hálózati réteg hoszt-router protokollja
Adatkapcsolati réteg hoszt-router protokollja
Fizikai réteg hoszt-router protokollja

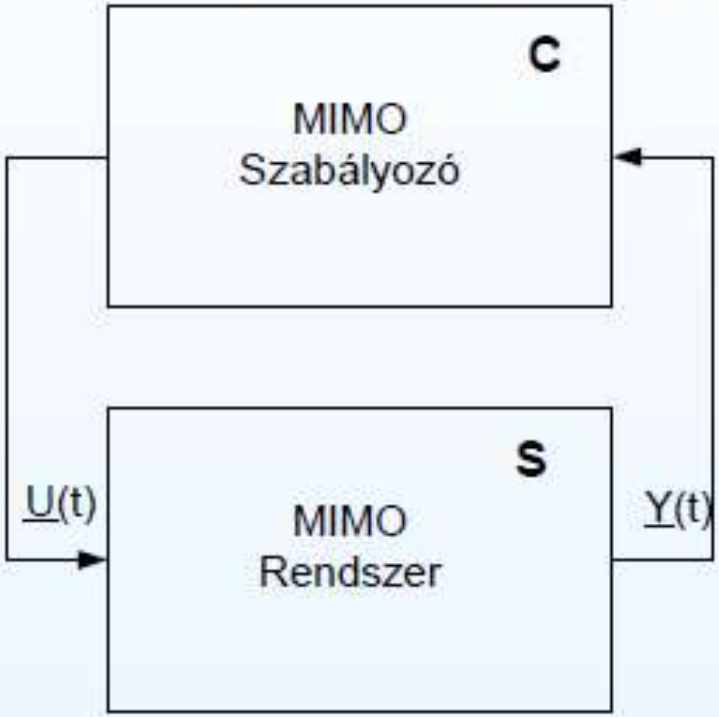
A szoftver rendszerek környezete

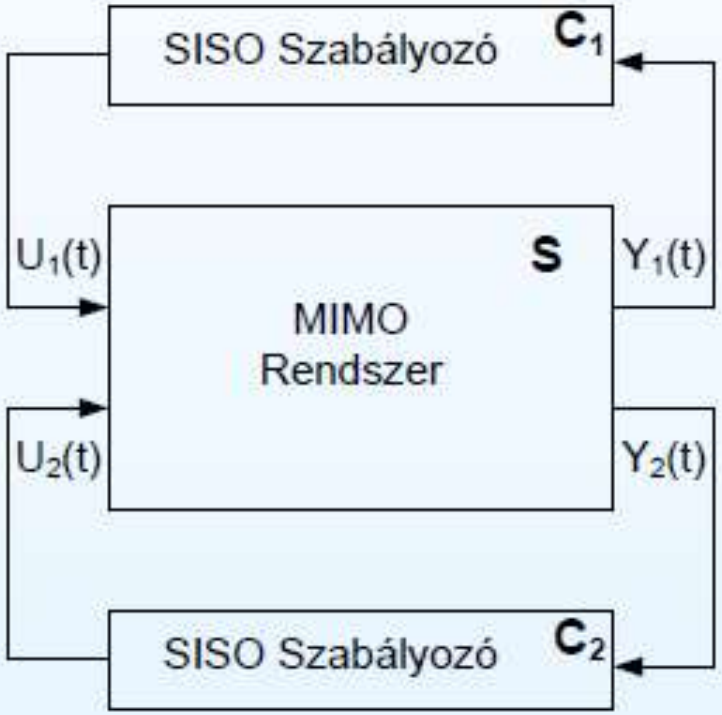
A mérés-adatgyűjtő és irányító szoftver rendszerek környezete



Több bemenetű több kimenetű rendszerek – 2

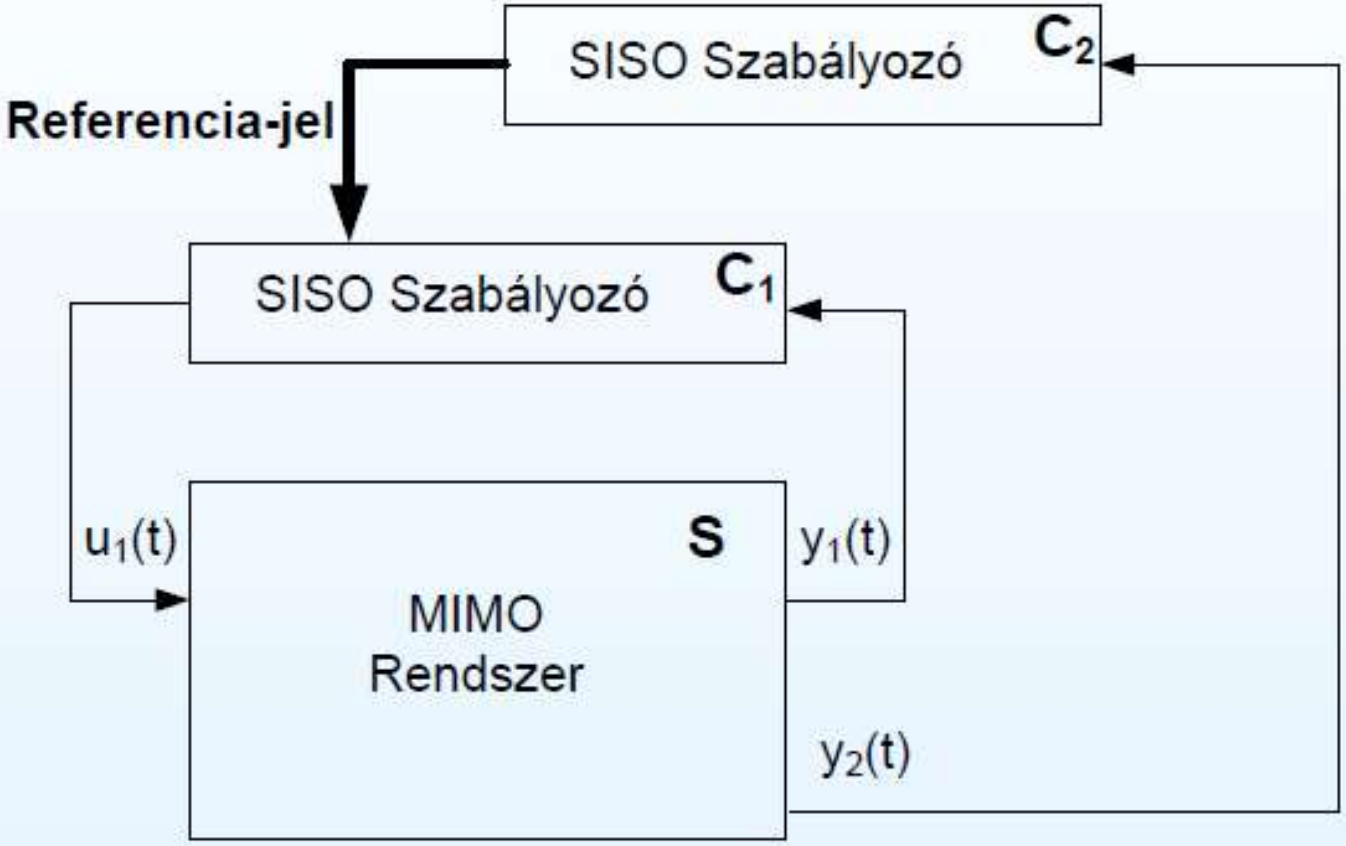
Szabályozás egy vagy több szabályozóval



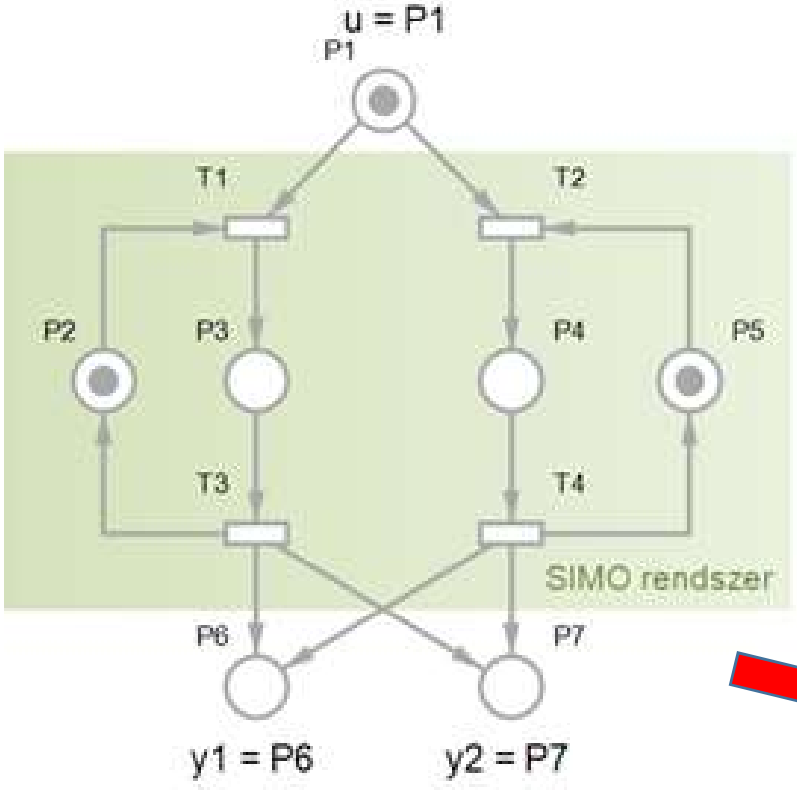


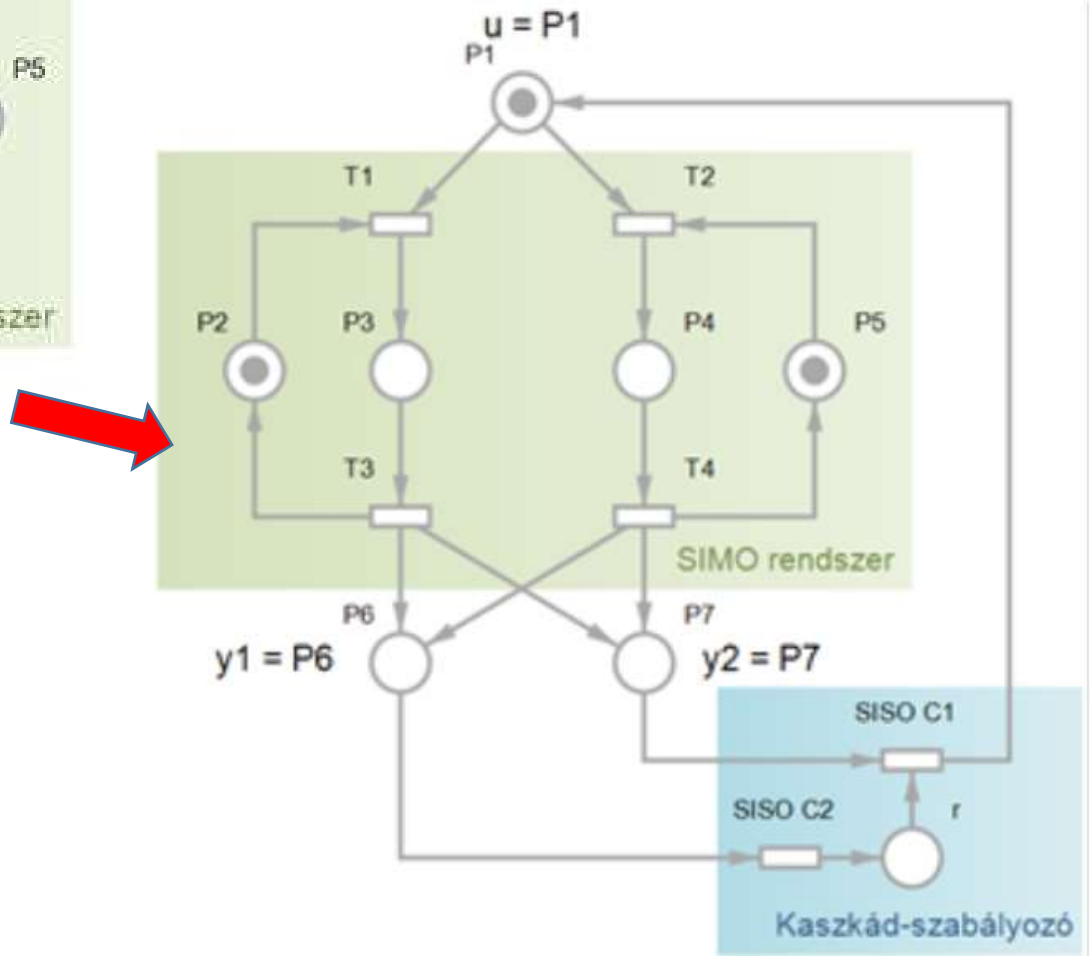
Összetett szabályozórendszerek – 1

Kaszkád szabályozás

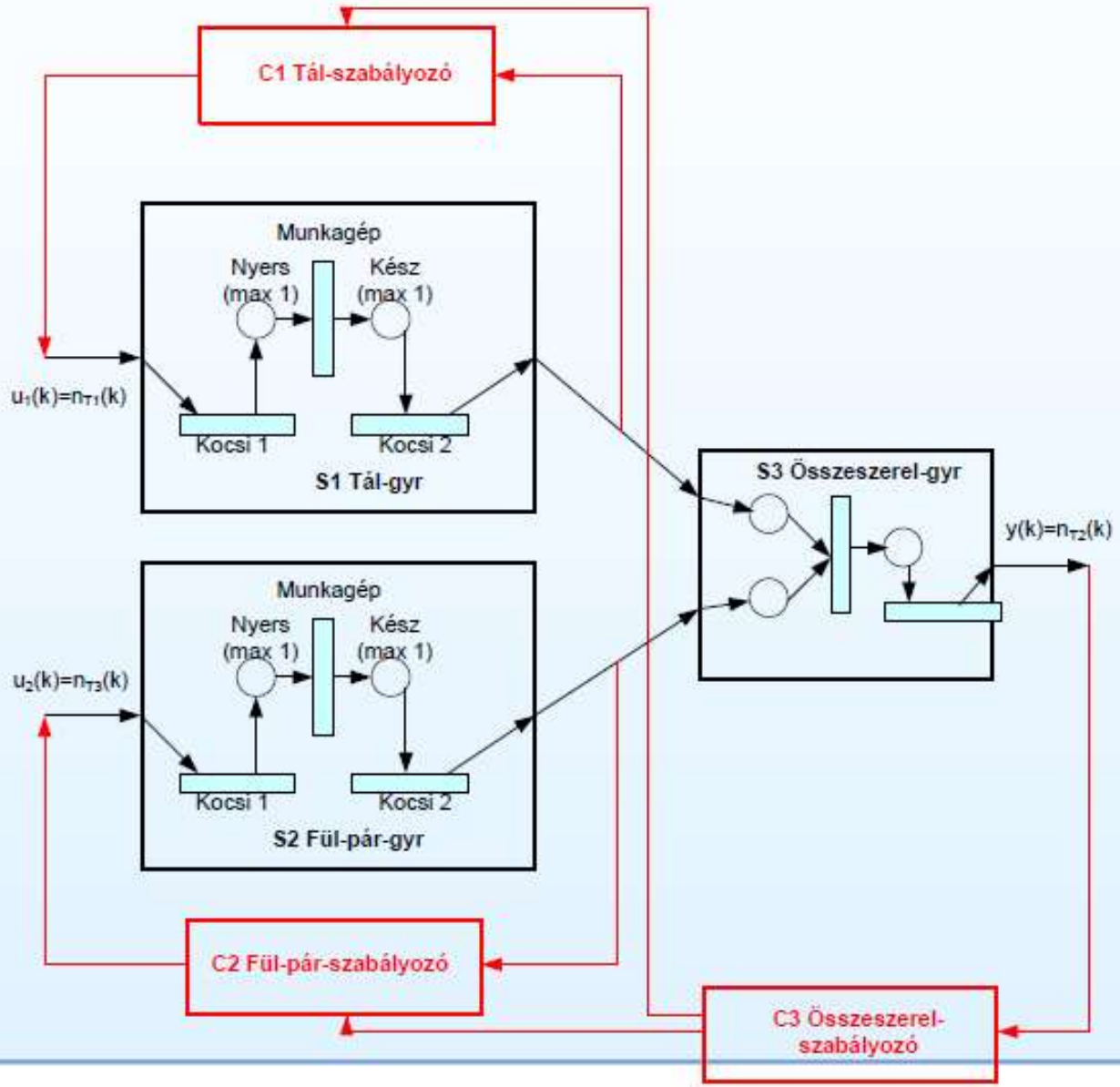


Feladat kaszkád szabályozó használatára





Példa: Összetett gyártórendszer szabályozása



Összetett szabályozórendszerek – 2

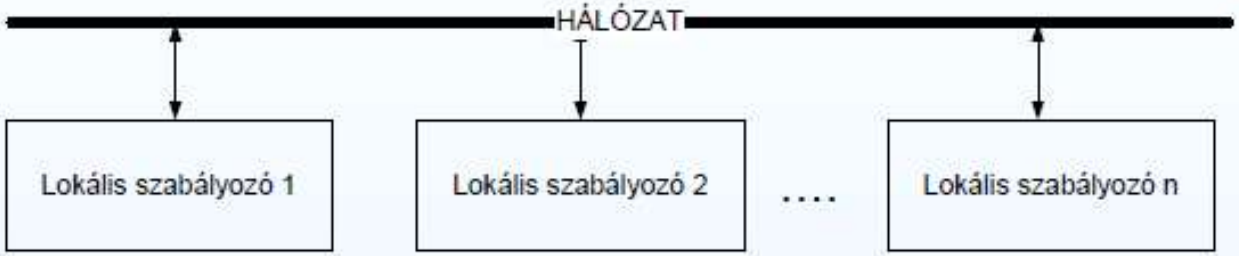
Centrális szerkezetű elosztott szabályozórendszer

Jellemzők

- központi (szerver) gép sokkal nagyobb kapacitású: felügyelő szabályozás, központi mérésadatgyűjtés és archiválás
- lokális szabályozók: pl. PID vagy PLC egységek
- csillag topológiájú hálózathoz illeszkedik

Összetett szabályozórendszerek – 3

"Demokratikus" szerkezetű elosztott szabályozórendszer

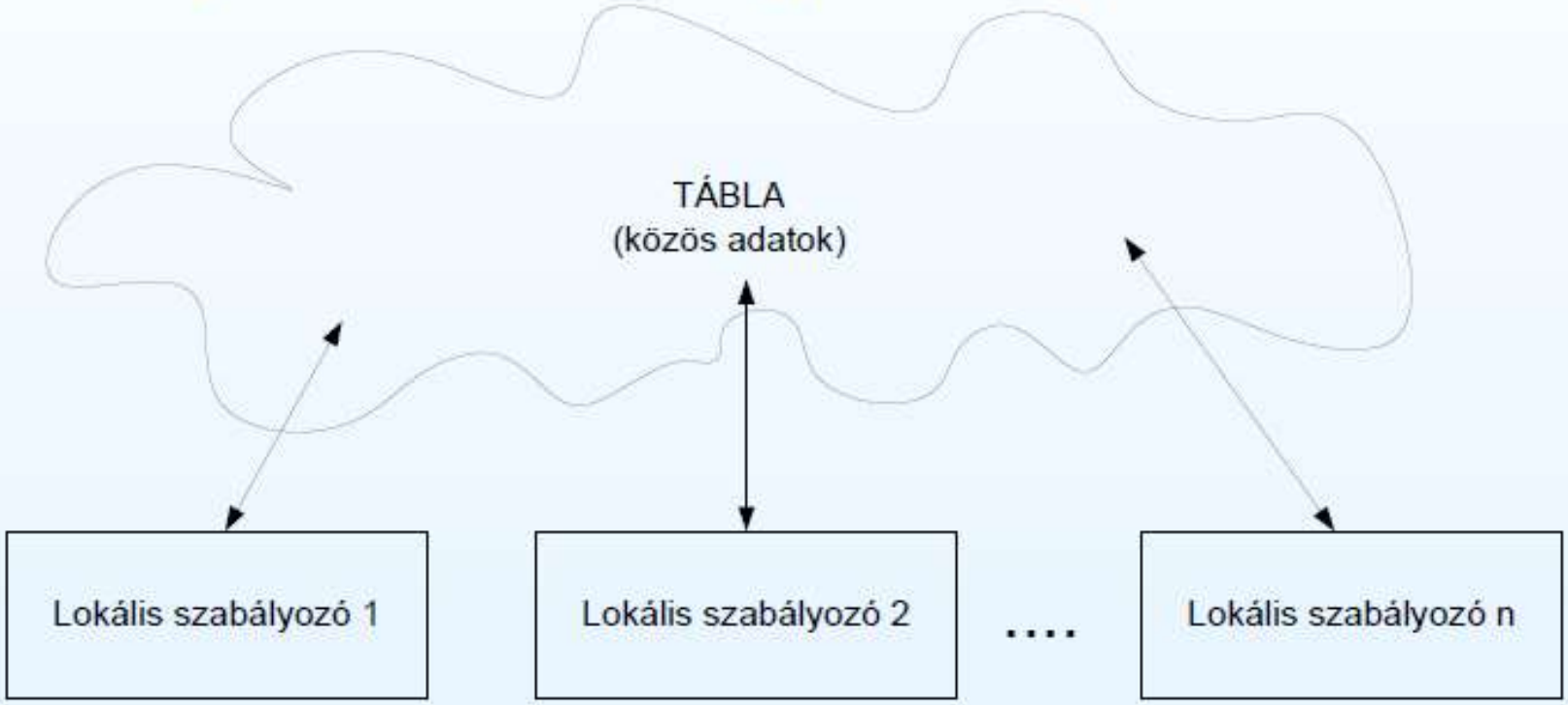


Jellemzők

- a kontrollerek bármilyen kapacitásúak lehetnek, nincs kitűntetett egység
- sín topológiájú adatszóró hálózathoz illeszkedik
- általában statikus csatornakiosztással és kis adatforgalommal

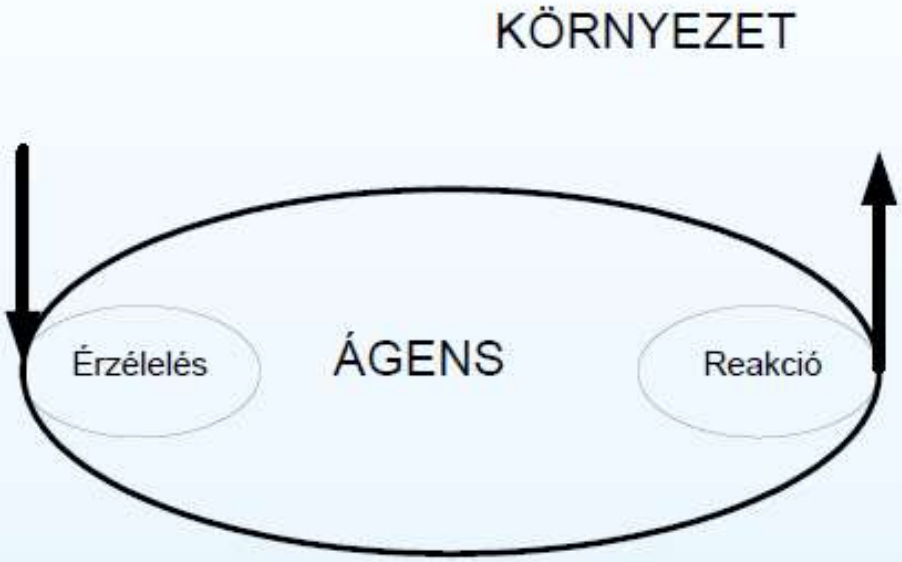
Tábla (blackboard) szerkezet

"Demokratikus" szerkezetű elosztott szabályozórendszerhez nagy adatforgalommal



Ágens rendszerek

Ágensek

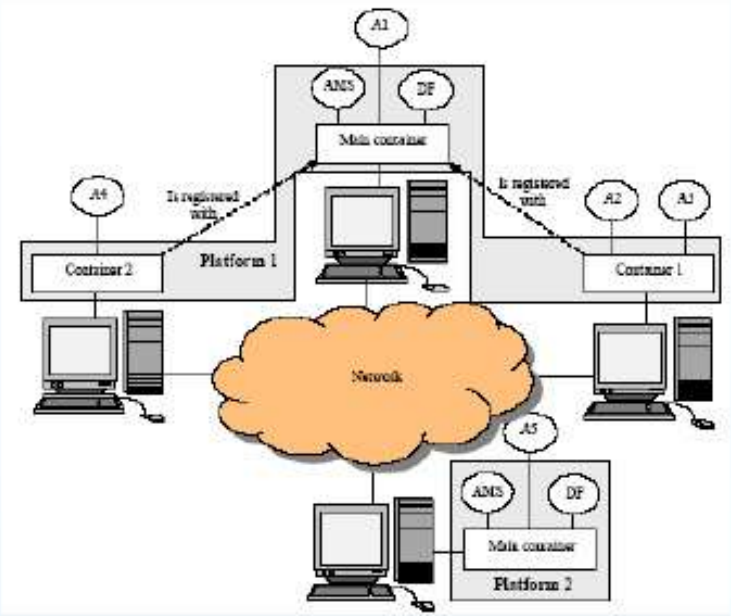


Speciális szoftver komponensek

- **autonóm**ok
- **környezett**el kommunikálnak
- **reaktívak**: érzékelik a környezetet és reagálnak

Kommunikáció és kooperáció ágens rendszerekben

Ágens rendszerek: ágens platformok központi szolgáltatásokkal



- **platformok**: kapcsolatban lévo ágenseket tartalmaznak
- **fo platform**: mindig aktív, beépített ágensekkel (pl.kommunikációra, felhasználói felület kiszolgálására)
- **standard kommunikáció**: ACL üzenetekkel

Folyamatbányászat

Sok-sok adat "Az adat az új olaj!"

Az elmúlt 10 percben több adatot generáltunk mint 2003 előtt összesen.

Adatokat generálunk:

- jegyet vásárolunk egy repülőjáratra
- veszünk egy kávét
- feltöltjük üzemanyaggal az autónkat
- kezdeményezünk egy telefonhívást
- küldünk egy emailt
- megnézünk egy klippet a youtube-on
- stb.

Vizsgálható kérdések

Vegyünk példaként egy gyártási folyamatot:

- Melyik komponens törött el, hibásodott meg?
- Mikor tört el/hibásodott meg és miért?
- Melyik komponenst kell kicserélni?
- Meg tudjuk jósolni, hogy melyik komponenssel fog történni valami?
- Tudunk-e tanulni a létező problémákból, hogy mely részeket kell javítani?
- Mi okozza a késedelmet a folyamatban?
- Meg tudjuk előre határozni, hogy a következő napon az adott osztályon mekkora személyzetre lesz szükség?
- Hogyan lehet csökkenteni a költségeket a minőségi ellátás megtartása mellett?

Kérdések megválaszolása

Számos tudományterületre van szükség:

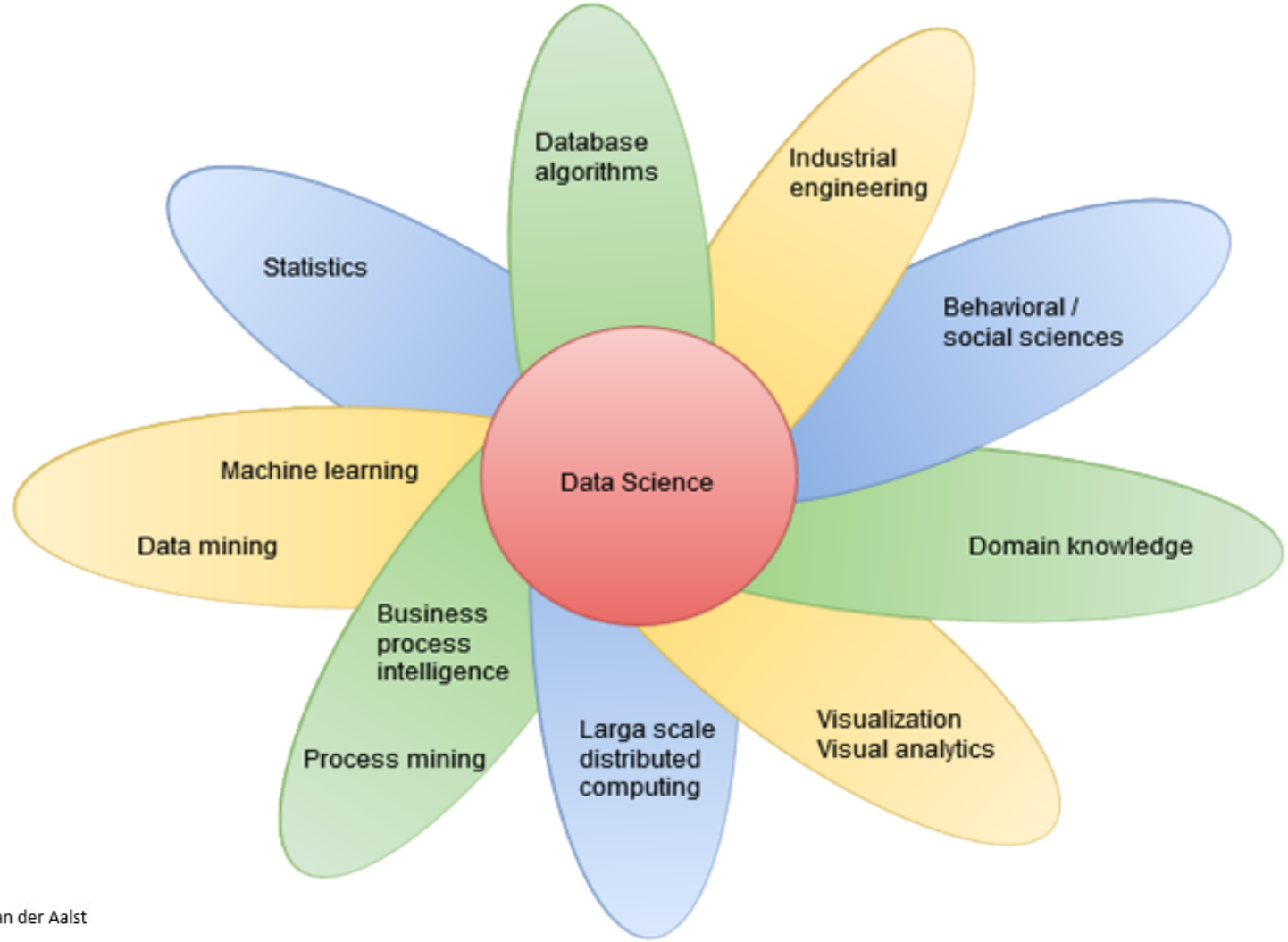
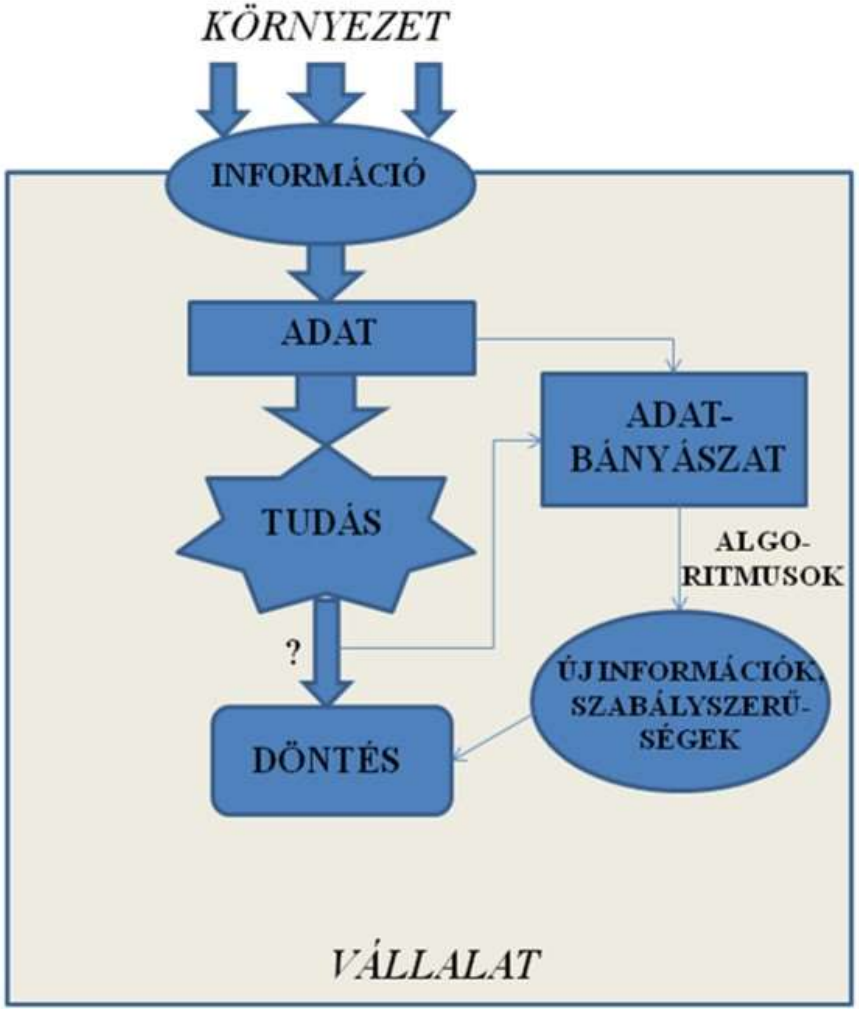


Diagram by W. van der Aalst

Adatok felhasználása

A folyamatbányászat az adatbányászat egy speciális területének is tekinthető



Folyamatbányászat vs. adatbányászat

- mindkettő adatokkal dolgozik
- az adatbányászati technikák nem folyamat centrikusak
- a tradicionális adatbányászati technikákkal a folyamat felfedezés, összehasonlítás, torlódás elemzés nem elvégezhető

A két módszer együttes használatával komplex kérdésekre adhatunk választ.

Mi is a folyamatbányászat?

- A folyamatbányászat célja: a folyamati adatokból valódi hasznosítható tudásanyag kinyerése.
- Felderíthetők és megjeleníthetők azok az összefüggések (pl. az ügyfélcsoportok, régiók, a termékek stb. között), amelyek
  - a hosszú átfutási időért,
  - a magas költségekért,
  - a rossz minőségért felelősek.
- Felderíthetőek a gyártási folyamatban megjelenő hibák, amelyek
  - a leállásokért,
  - a hosszabb gyártási időért felelősek.

Kiemelt területek

- Folyamat felfedezése , feltérképezése (Process discovery): mi történik valójában?
- Megfelelőség elemzés (Conformance checking): az történik amit elvárunk, amit előre meghatároztunk?
- Teljesítményelemzés (Performance analysis): hol lehet javítani, változtatni?
- Folyamat előrejelzés (Process prediction): vajon egy eset be fog következni?
- Folyamat fejlesztés, tökéletesítés (Process improvement): hogyan alakítsuk át a folyamatot?
- Stb.

Példa2: járművek fejlesztése

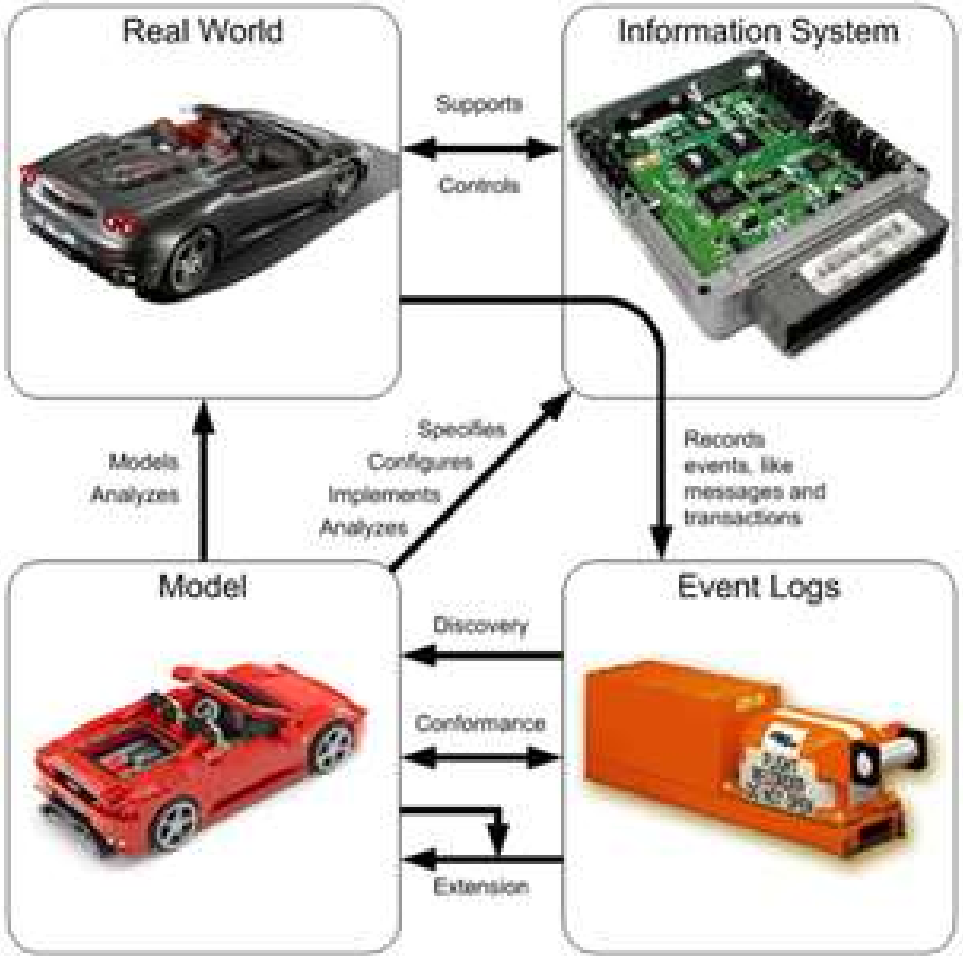


Diagram by W. van der Aalst

Honnan érkeznek az esemény adatok?

- adatbázis rendszerekből (pl. páciensek adatai egy kórházban)
- kórházi információs rendszerekből (pl. Chipsoft)
- tranzakciós logokból (pl. kereskedelmi rendszer)
- üzleti rendszerekből, vállalati erőforrás tervező rendszerekből (pl. SAP, Oracle)
- üzenet logokból (pl. IBM middleware)
- folyamatirányító rendszerekből (pl. WebSphere software, Staffware, Flower, Eastman software, People Soft, Adept)
- web szerverek (pl. Apache)
- Szoftver Konfiguráció Menedzsment rendszerekből (pl. Subversion)
- ügyfél kapcsolatokat menedzselő rendszerekből (pl.: Microsoft Dinamics CRM)
- vesszővel elválasztott értékek fájljaiból (CSV) vagy táblázatokból
- stb.

Kiindulási pont: esemény adatok

Minden sor egy eseményt azonosít (eset azonosító, esemény neve, időbélyeg, egyéb).

Hallgató neve	Kurzus	Vizsga időpontja	Jegy
Kovács Péter	Programozás I.	2016.01.15	4
Molnár Hajnalka	Programozás I.	2016.01.15	5
Fehér Klára	Programozás I.	2016.01.15	2
Fekete Péter	Programozás I.	2016.01.15	3
Kovács Péter	Programozás II.	2016.01.18	4
Molnár Hajnalka	Programozás II.	2016.01.18	5
Fehér Klára	Mesterséges intelligencia	2016.01.22	3
Kovács Péter	Mesterséges intelligencia	2016.01.22	5
Molnár Hajnalka	Mesterséges intelligencia	2016.01.22	4

Folyamatbányászati feladatkörök

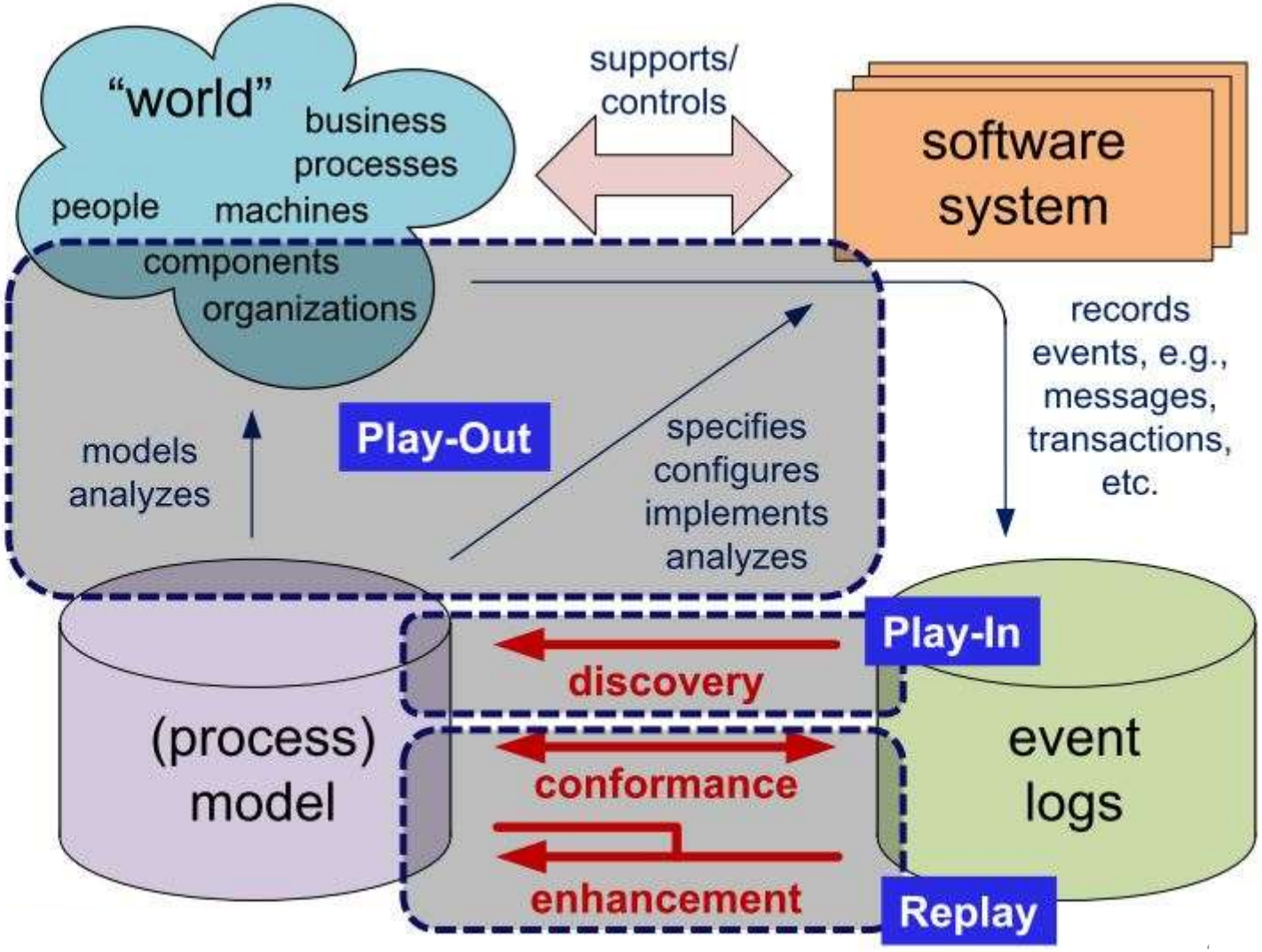


Diagram by W. van der Aalst

Play-In

Folyamat felfedezés: a megfigyelt viselkedés alapján a folyamati modell előállítása, felépítése

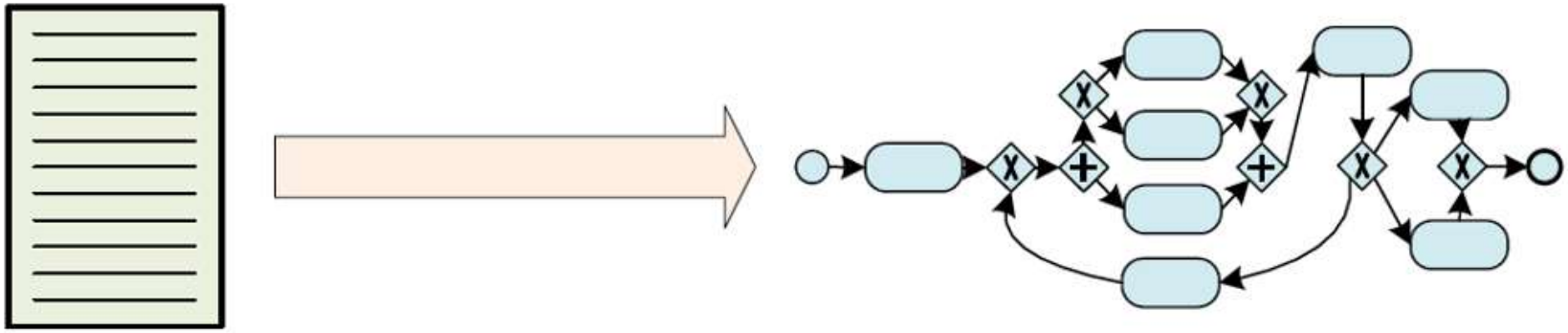


Diagram by W. van der Aalst

Replay

A folyamatbányászat legfontosabb formája, amihez tekintjük a modellt és a viselkedést (valós lefutásokat) szembeállítjuk a modellt a valósággal

Használható:

- megfelelőség ellenőrzéshez (conformance checking),
- előrejelzéshez (prediction),
- torlódás elemzéshez (bottleneck analysis)
- diagnosztikához (diagnostics)

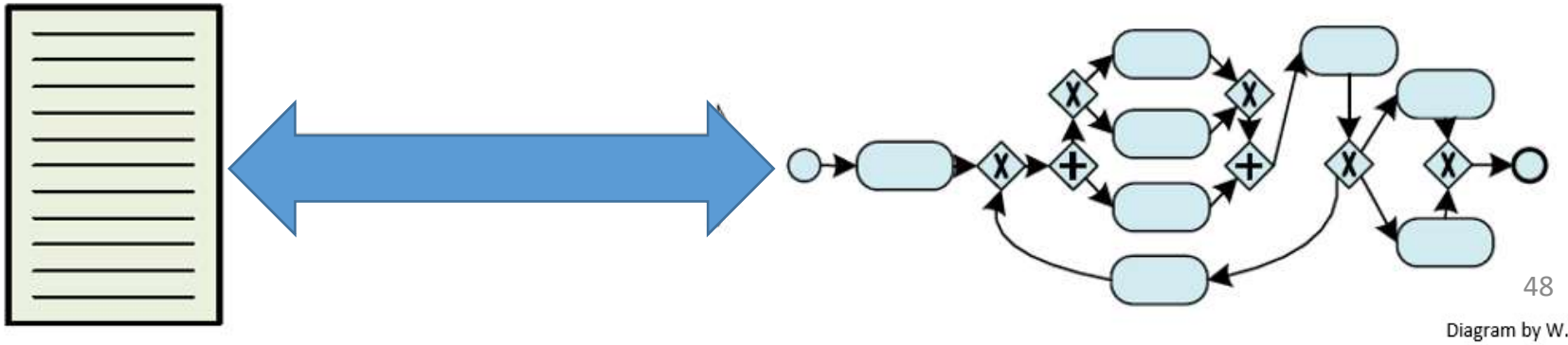


Diagram by W. van der Aalst

Play-Out

- Szimuláció
- Folyamat automatizálás
- Menedzsment játékok
- Modell ellenőrzés

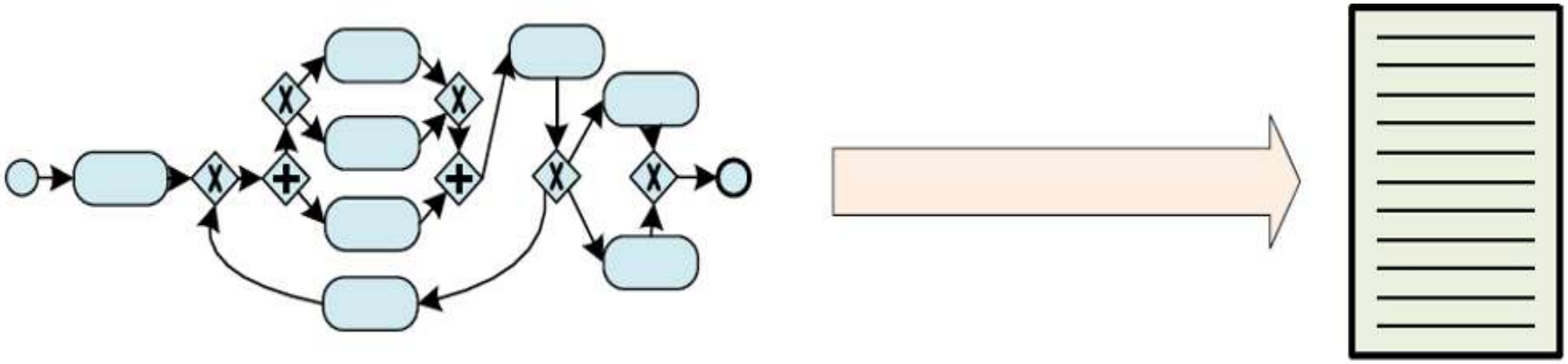
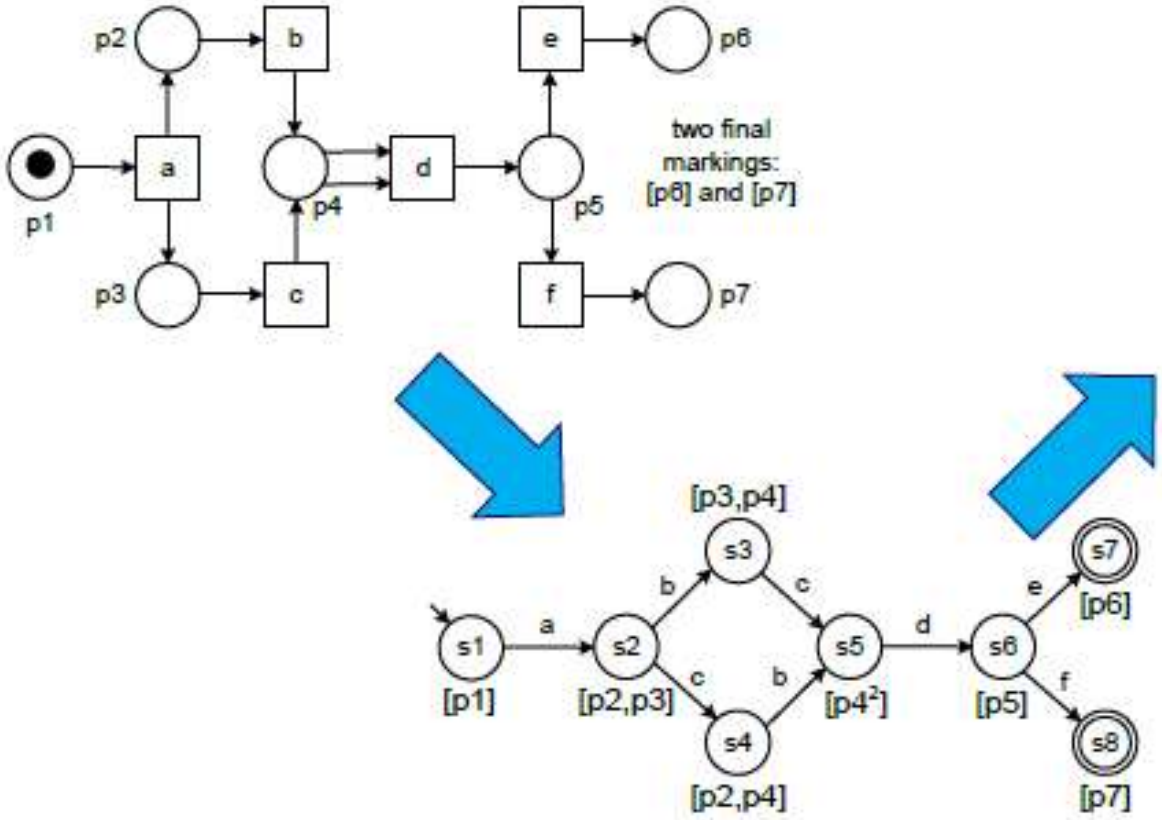


Diagram by W. van der Aalst

Play-Out



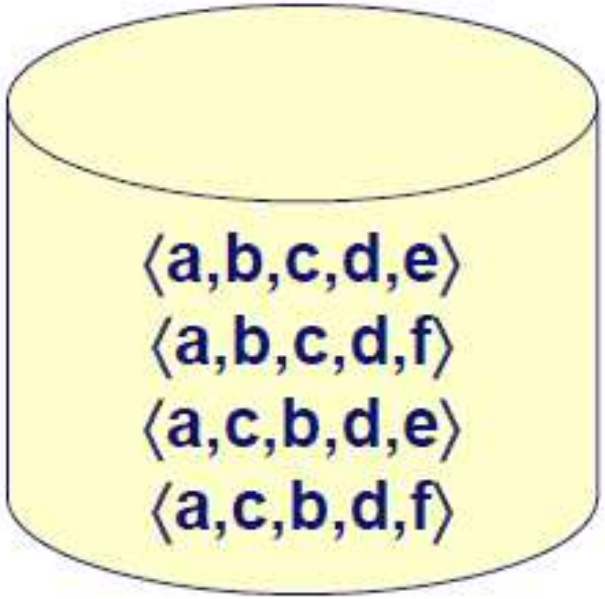


Diagram by W. van der Aalst

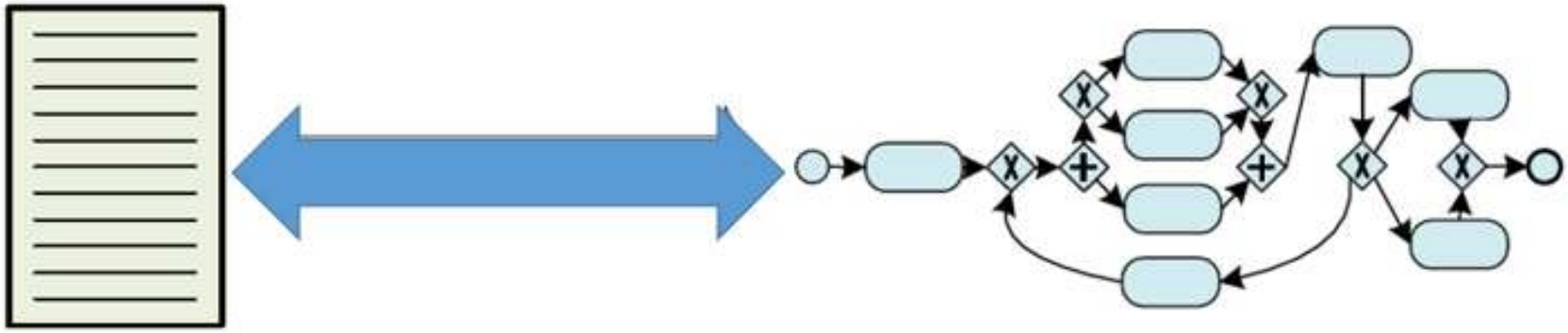
Konformancia elemzés – megfelelőség elemzés

A vizsgálat szükségessége:

- Ipari rendszerekben a munkafolyamatok lefutása sok esetben eltérhet a tervezett (ütemezett) lefutástól.
- Ezen eltérések okainak felkutatásának első lépése, hogy megtaláljuk, hogy a valós lefutás hol, milyen mértékben tér el a tervezettől.
- Egy ipari folyamat modellje tartalmazza az adott műveletsor lépéseit, átmeneteit, erőforrás (emberi, anyagi) felhasználását, időigényeit, viszont nem tartalmazhat minden apró részletet a folyamattal és pl. az emberekkel kapcsolatban.
- Ezért van szükség a konformancia vizsgálatra, mely segítségével megkereshetjük a modell és a tényleges lefutások közötti különbségeket.
- Mivel a modell az optimális lefutást hivatott leírni, ezért bármilyen eltérés a modelltől negatív hatással van a lefutásra és eltávolítja azt az optimálistól.

Konformancia analízis

- Akkor használatos, ha van előzetes modell.
- A létező modellt hasonlítjuk össze a folyamat eseménynaplójával.
- Analizálja az eltéréseket az események és a modell között.
- Segítségével meghatározzuk az egyezés mértékét a modell által leírt működés és a tényleges esemény sorok között.



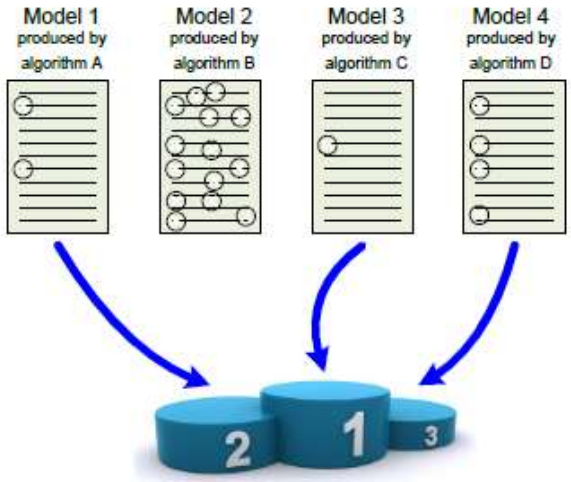
Megfelelően elkészített modell

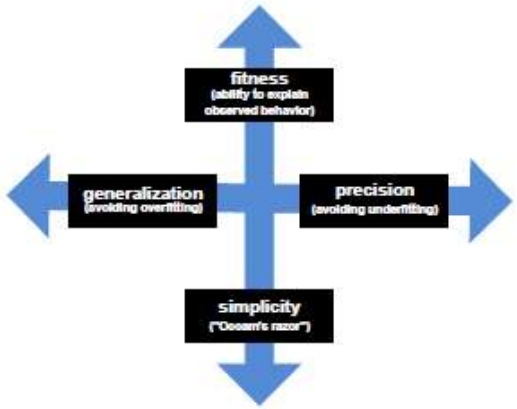
A modell négy fő tulajdonsággal rendelkezik:

- **megegyezőség** (fitness): az eseménynaplóban felsorolt események menyire jól illeszkednek, azaz az általuk leírt útvonallal milyen mértékig lehet végig menni a modellen,
- **egyszerűség** (simplicity): a lehetséges legegyszerűbb modell mely leírja az események viselkedését a legjobb,

(Megj.: A megegyezőség és az egyszerűség önmagukban nem elegendőek a modell minőségének megállapításához.)

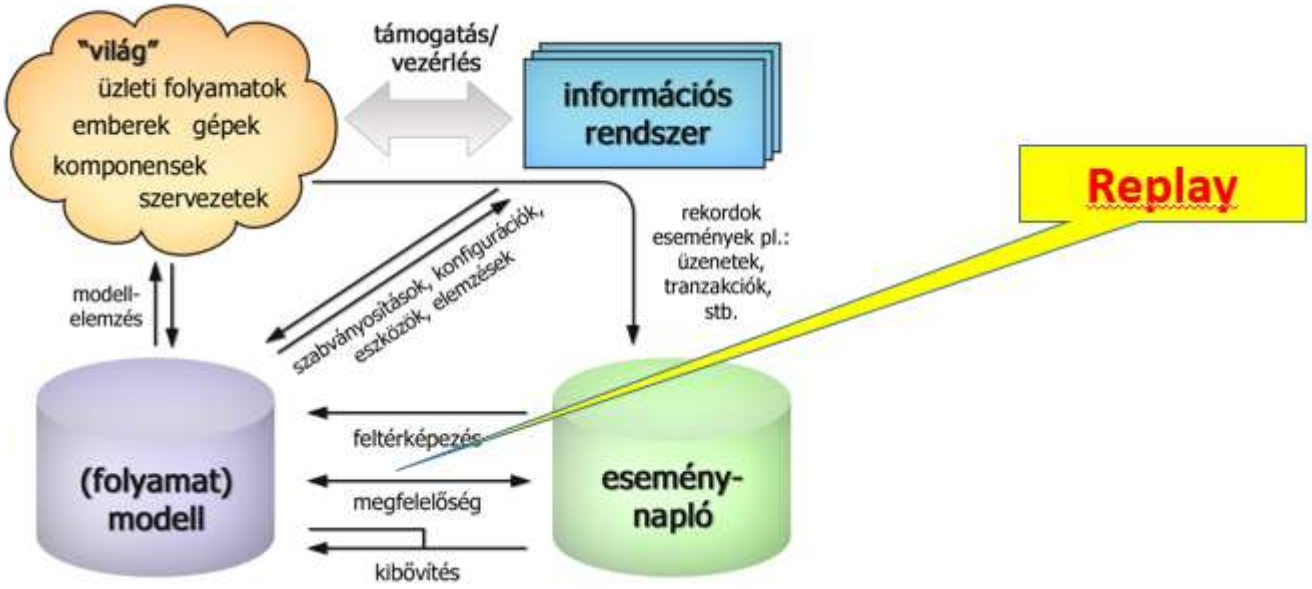
- **általánosítás** (generalization): törekedni kell minél általánosabb modell készítésére, azaz a modell minél több viselkedést ír le annál általánosabb,
- **pontosság** (precision): arra törekszünk, hogy a lehető legkevesebb viselkedést írjuk le (Megj.: Ha túlságosan specifikusan állítjuk össze a modellt, akkor az eseménynaplóban leírt események csak egy kisebb csoportját írja le jól, a többit hiányosan vagy egyáltalán nem.)



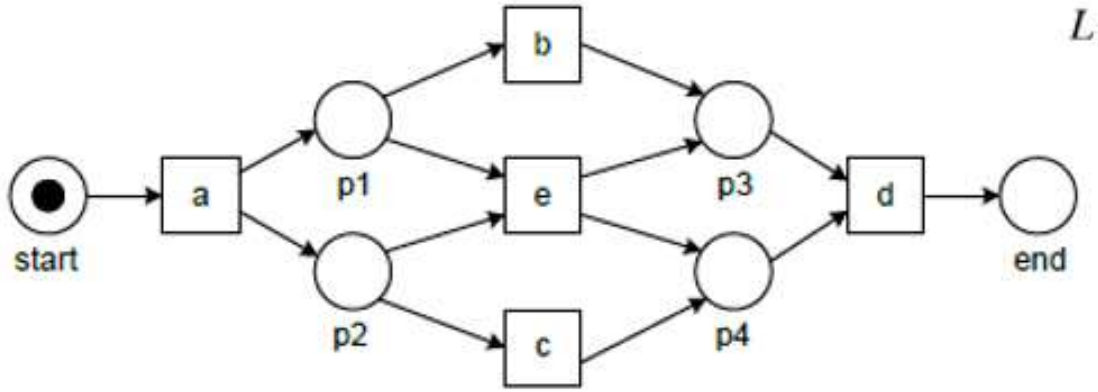


A konformancia ellenőrzés módszerei

1. Konformancia ellenőrzés ok-okozati lábnyomokkal,
2. Konformancia ellenőrzés token alapú visszajátszással,
3. Illesztés alapú megfelelőség ellenőrzés.



Konformancia ellenőrzés ok-okozati lábnyomokkal

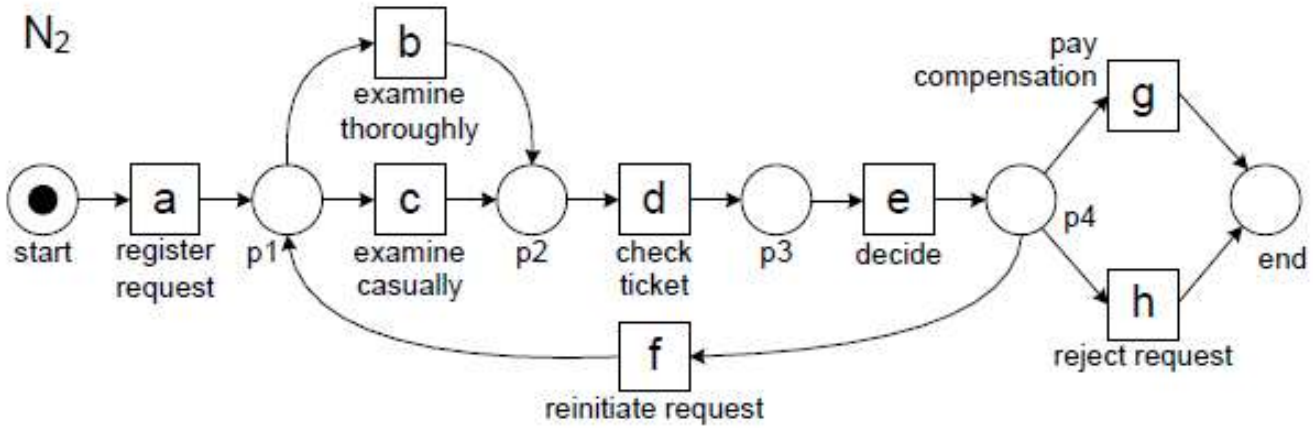


$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

	a	b	c	d	e
a	$\#_{L_1}$	$\rightarrow_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$
b	$\leftarrow_{L_1}$	$\#_{L_1}$	$\parallel_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
c	$\leftarrow_{L_1}$	$\parallel_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
d	$\#_{L_1}$	$\leftarrow_{L_1}$	$\leftarrow_{L_1}$	$\#_{L_1}$	$\leftarrow_{L_1}$
e	$\leftarrow_{L_1}$	$\#_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$

lábnyom alapú megfelelés = 1 (1 = tökéletes egyezés, 0 = nincs egyezés)

A lábnyomok megadják a log és a modell egybevágóságát.



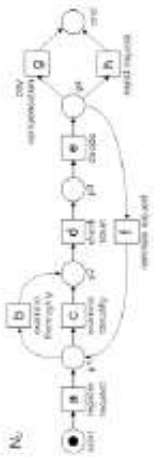
	a	b	c	d	e	f	g	h
a	#	$\rightarrow$	$\rightarrow$	#	#	#	#	#
b	$\leftarrow$	#	#	$\rightarrow$	#	$\leftarrow$	#	#
c	$\leftarrow$	#	#	$\rightarrow$	#	$\leftarrow$	#	#
d	#	$\leftarrow$	$\leftarrow$	#	$\rightarrow$	#	#	#
e	#	#	#	$\leftarrow$	#	$\rightarrow$	$\rightarrow$	$\rightarrow$
f	#	$\rightarrow$	$\rightarrow$	#	$\leftarrow$	#	#	#
g	#	#	#	#	$\leftarrow$	#	#	#
h	#	#	#	#	$\leftarrow$	#	#	#

L_{full}

#	trace
455	accieh
191	abdeg
177	adceh
144	abdeh
111	accieg
82	adceg
56	adbeh
47	accieftbeh
38	adbeg
33	accieftbeh
14	accieftbeg
11	accieftbeg
9	adceftbeh
8	adceftbeh
5	adceftbeg
3	accieftbeftbeg
2	adceftbeg
2	adceftbeftbeg
1	adceftbeftbeh
1	adceftbeftbeg
1	adceftbeftbeftbeg
1381	

	a	b	c	d	e	f	g	h
a	#	$\rightarrow$	$\rightarrow$	$\rightarrow$	#	#	#	#
b	$\leftarrow$	#	#	$\parallel$	$\rightarrow$	$\leftarrow$	#	#
c	$\leftarrow$	#	#	$\parallel$	$\rightarrow$	$\leftarrow$	#	#
d	$\leftarrow$	$\parallel$	$\parallel$	#	$\rightarrow$	$\leftarrow$	#	#
e	#	$\leftarrow$	$\leftarrow$	$\leftarrow$	#	$\rightarrow$	$\rightarrow$	$\rightarrow$
f	#	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\leftarrow$	#	#	#
g	#	#	#	#	$\leftarrow$	#	#	#
h	#	#	#	#	$\leftarrow$	#	#	#

N_2



A különbségek számszerűsítése

1 - x/y, ahol x a log-beli eltérések száma a modellbeli bejegyzésekhez képest, y az összes lábnyom száma a táblázatban

	a	b	c	d	e	f	g	h
a				$\rightarrow$: #				
b				$\|$:$\rightarrow$	$\rightarrow$: #			
c				$\|$:$\rightarrow$	$\rightarrow$: #			
d	$\leftarrow$: #	$\|$:$\leftarrow$	$\|$:$\leftarrow$			$\leftarrow$: #		
e		$\leftarrow$: #	$\leftarrow$: #					
f				$\rightarrow$: #				
g								
h								

$$1 - \frac{12}{64} = 0.8125$$

A kapott eredmény felhasználása diagnosztikához

	a	b	c	d	e	f	g	h
a				$\rightarrow:\#$				
b				$\parallel:\rightarrow$	$\rightarrow:\#$			
c				$\parallel:\rightarrow$	$\rightarrow:\#$			
d	$\leftarrow:\#$	$\parallel:\leftarrow$	$\parallel:\leftarrow$			$\leftarrow:\#$		
e		$\leftarrow:\#$	$\leftarrow:\#$					
f				$\rightarrow:\#$				
g								
h								

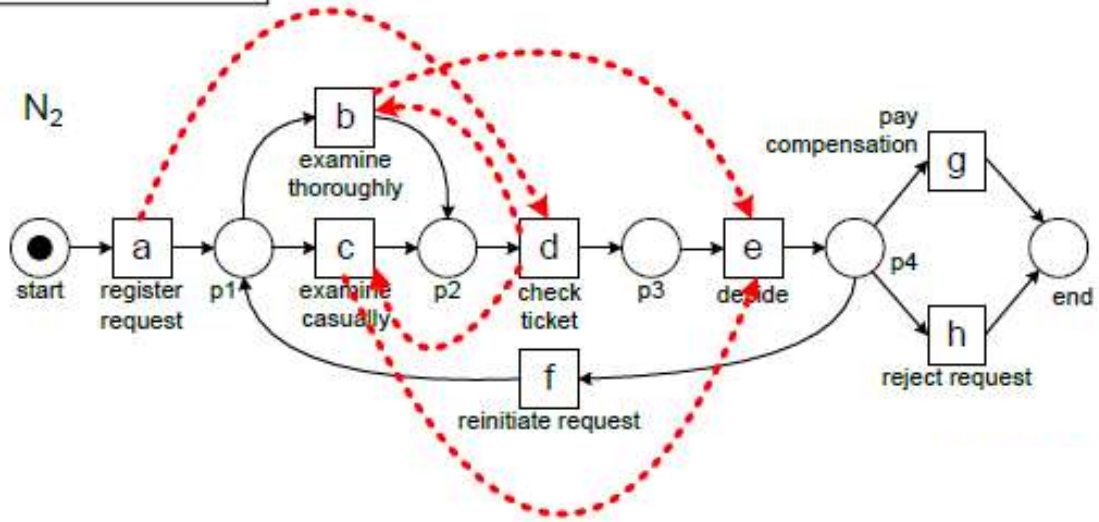


Diagram by W. van der Aalst

Konformancia token alapú egyezőség-vizsgálata

- Ehhez szükség van a folyamat modelljére, pl. Petri háló formájában
- A napló fájlok rengeteg információt tartalmaznak, melyek sok esetben nem feltétlen szükségesek a vizsgálat elvégzéséhez. Ezért az adatoknak előzetes átalakításon kell átesniük, mely folyamán csak azon esemény tulajdonságokat tartjuk meg, melyek számunkra szükségesek.
- Bevezetünk négy különböző mérőszámot:
  1. ***p*** létrehozott token (produced token),
  2. ***c*** elhasznált token (consumed token),
  3. ***m*** hiányzó token (missing token),
  4. ***r*** megmaradt token (remaining token).

A különbségek számszerűsítése

1 - x/y, ahol x a log-beli eltérések száma a modellbeli bejegyzésekhez képest, y az összes lábnyom száma a táblázatban

	a	b	c	d	e	f	g	h
a				$\rightarrow$: #				
b				$\|$:$\rightarrow$	$\rightarrow$: #			
c				$\|$:$\rightarrow$	$\rightarrow$: #			
d	$\leftarrow$: #	$\|$:$\leftarrow$	$\|$:$\leftarrow$			$\leftarrow$: #		
e		$\leftarrow$: #	$\leftarrow$: #					
f				$\rightarrow$: #				
g								
h								

$$1 - \frac{12}{64} = 0.8125$$

A kapott eredmény felhasználása diagnosztikához

	a	b	c	d	e	f	g	h
a				$\rightarrow:\#$				
b				$\parallel:\rightarrow$	$\rightarrow:\#$			
c				$\parallel:\rightarrow$	$\rightarrow:\#$			
d	$\leftarrow:\#$	$\parallel:\leftarrow$	$\parallel:\leftarrow$			$\leftarrow:\#$		
e		$\leftarrow:\#$	$\leftarrow:\#$					
f				$\rightarrow:\#$				
g								
h								

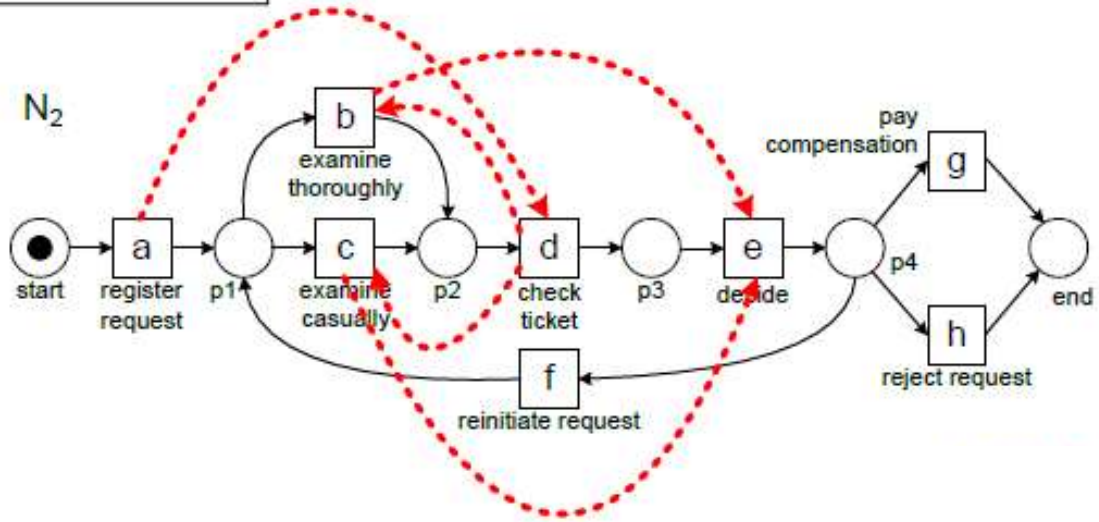
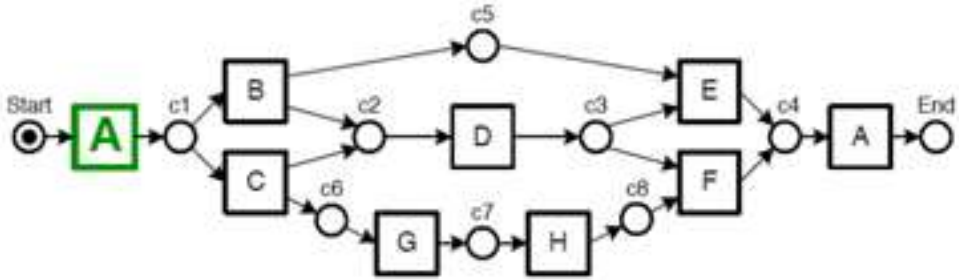


Diagram by W. van der Aalst

Konformancia token alapú egyezőség-vizsgálata

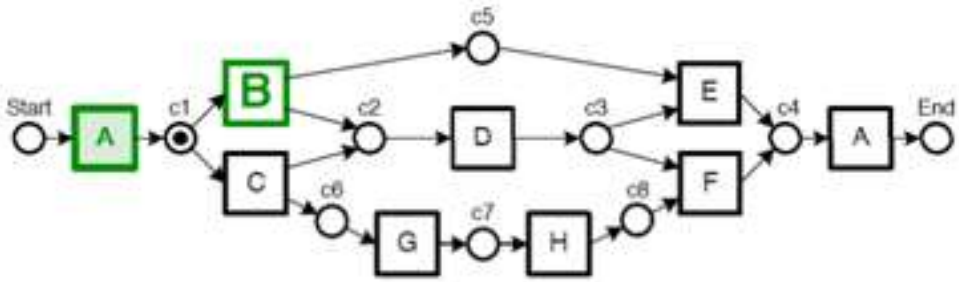
- Ehhez szükség van a folyamat modelljére, pl. Petri háló formájában
- A napló fájlok rengeteg információt tartalmaznak, melyek sok esetben nem feltétlen szükségesek a vizsgálat elvégzéséhez. Ezért az adatoknak előzetes átalakításon kell átesniük, mely folyamán csak azon esemény tulajdonságokat tartjuk meg, melyek számunkra szükségesek.
- Bevezetünk négy különböző mérőszámot:
  1. ***p*** létrehozott token (produced token),
  2. ***c*** elhasznált token (consumed token),
  3. ***m*** hiányzó token (missing token),
  4. ***r*** megmaradt token (remaining token).

Fitnesz érték figyelése az események végig követésével egy trace-ben



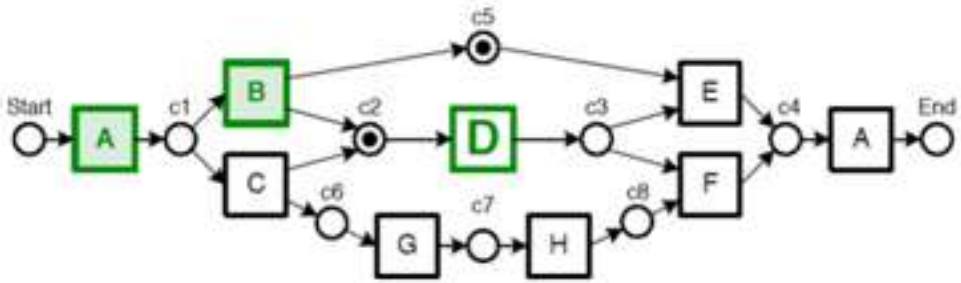
m = 0
r = 0
c = 0
p = 1

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA



m = 0
r = 0
c = 1
p = 2

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

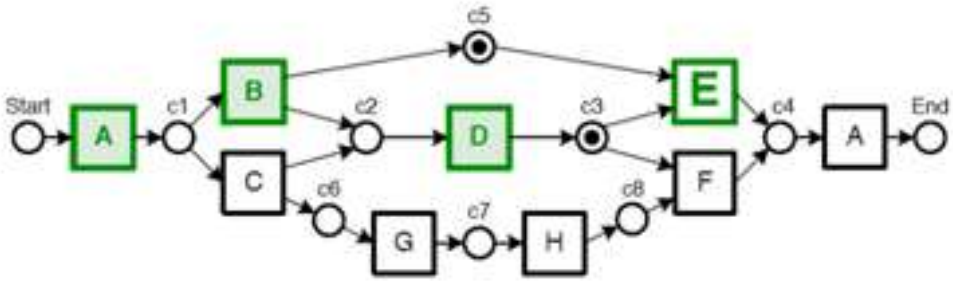


m = 0
r = 0
c = 2
p = 4

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

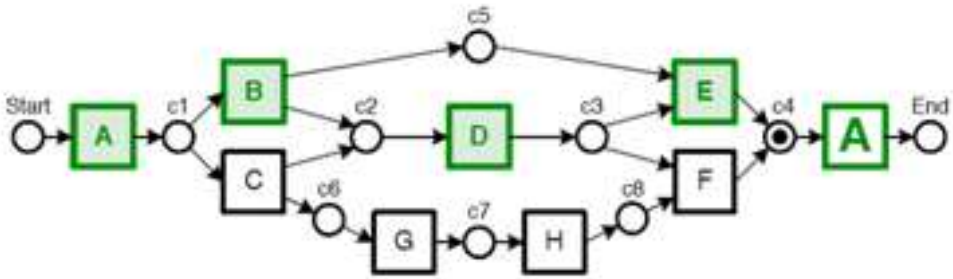
Diagram by W. van der Aalst

Ha nincs probléma (m=0, r=0):



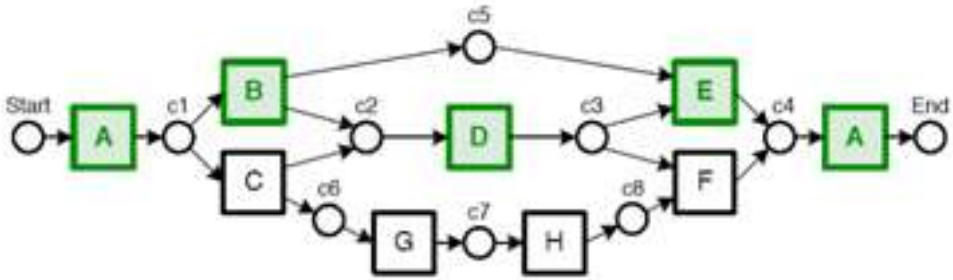
m = 0
r = 0
c = 3
p = 5

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA



m = 0
r = 0
c = 5
p = 6

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

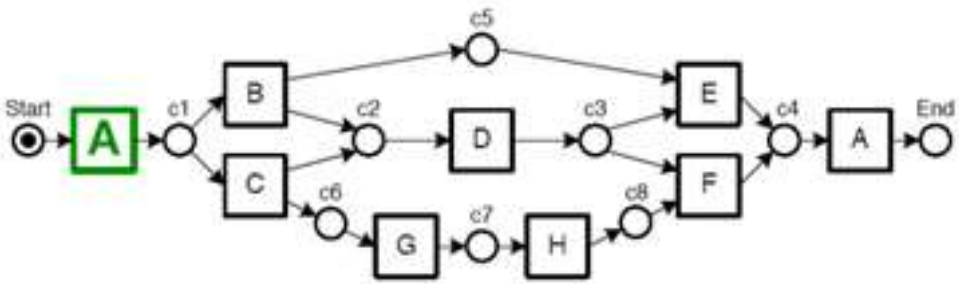


m = 0
r = 0
c = 7
p = 7

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

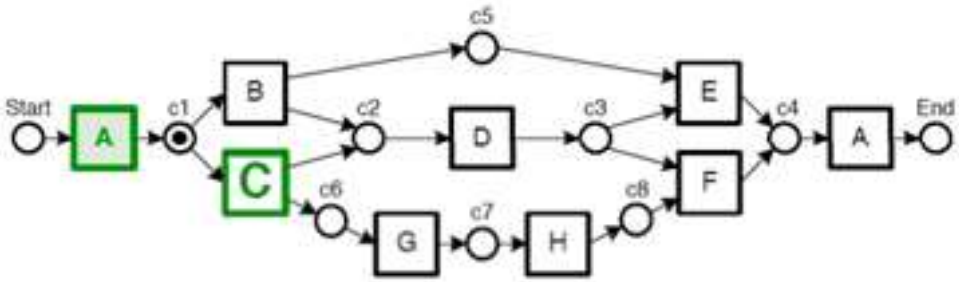
Diagram by W. van der Aalst

Egy másik (megvalósíthatatlan) trace



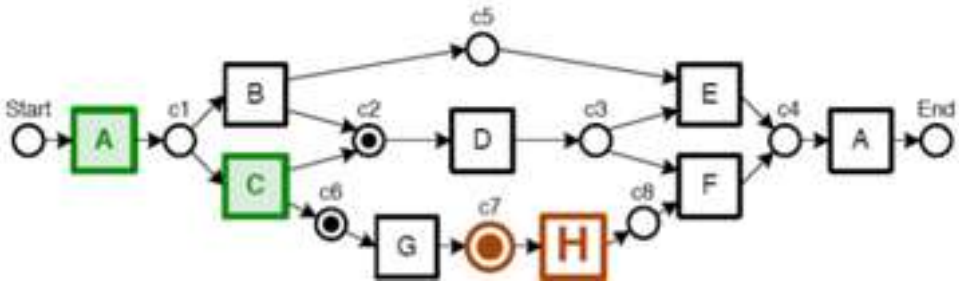
m = 0
r = 0
c = 0
p = 1

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA



m = 0
r = 0
c = 1
p = 2

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

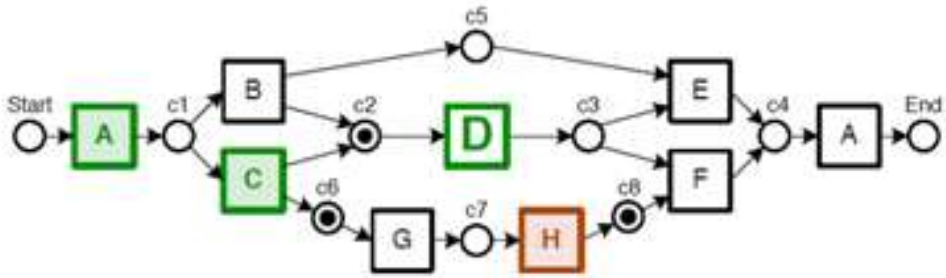


m = 1
r = 0
c = 2
p = 4

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

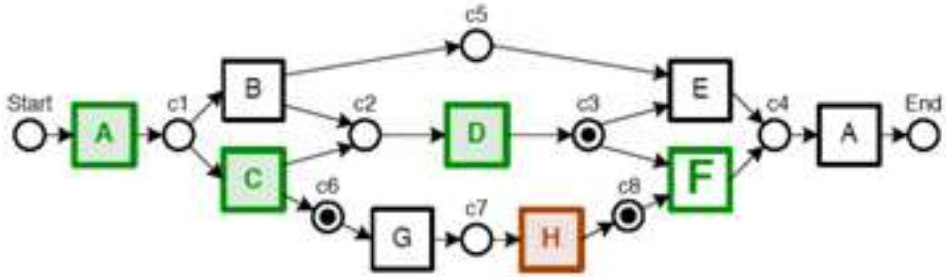
Diagram by W. van der Aalst

Folytatás



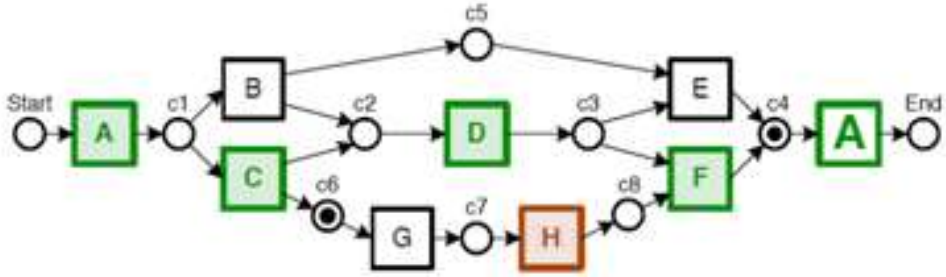
m = 1
r = 0
c = 3
p = 5

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA



m = 1
r = 0
c = 4
p = 6

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

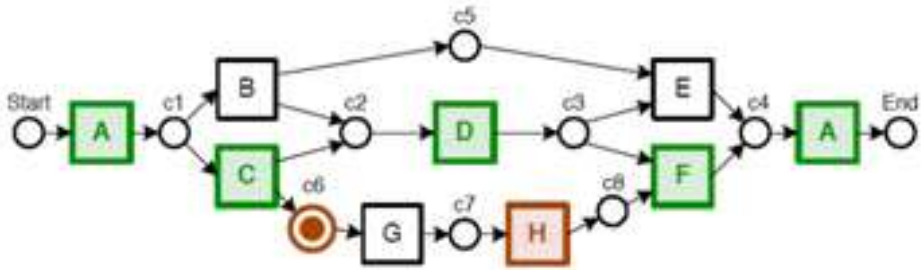


m = 1
r = 0
c = 6
p = 7

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

Diagram by W. van der Aalst

Fitnesz érték számolása



m = 1
r = 1
c = 8
p = 8

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

$$f = \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}) + \frac{1}{2}(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i})$$

$$f(M1, L2) = \frac{1}{2}(1 - \frac{51}{10666}) + \frac{1}{2}(1 - \frac{51}{10666}) \approx 0.995$$

Hogyan használhatjuk a módszert diagnosztikához

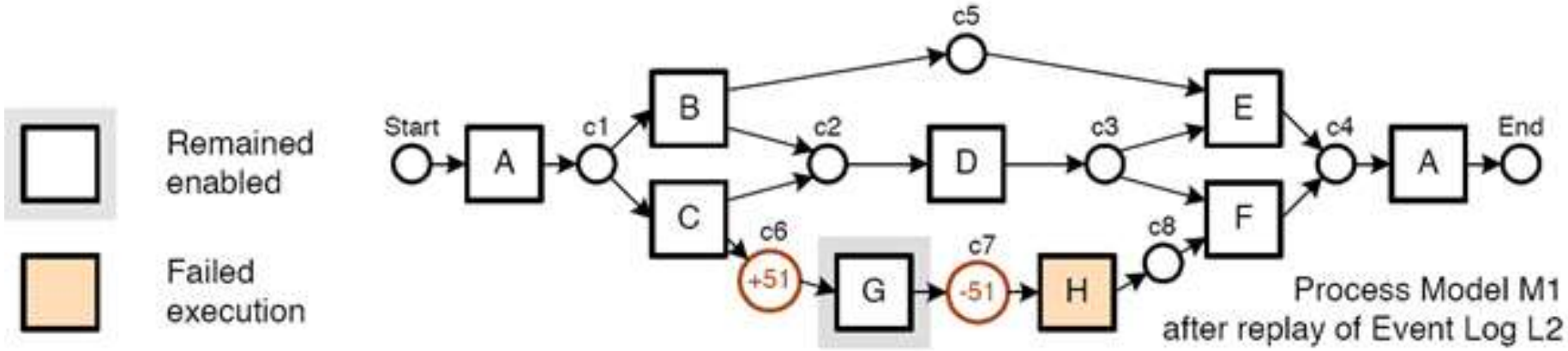
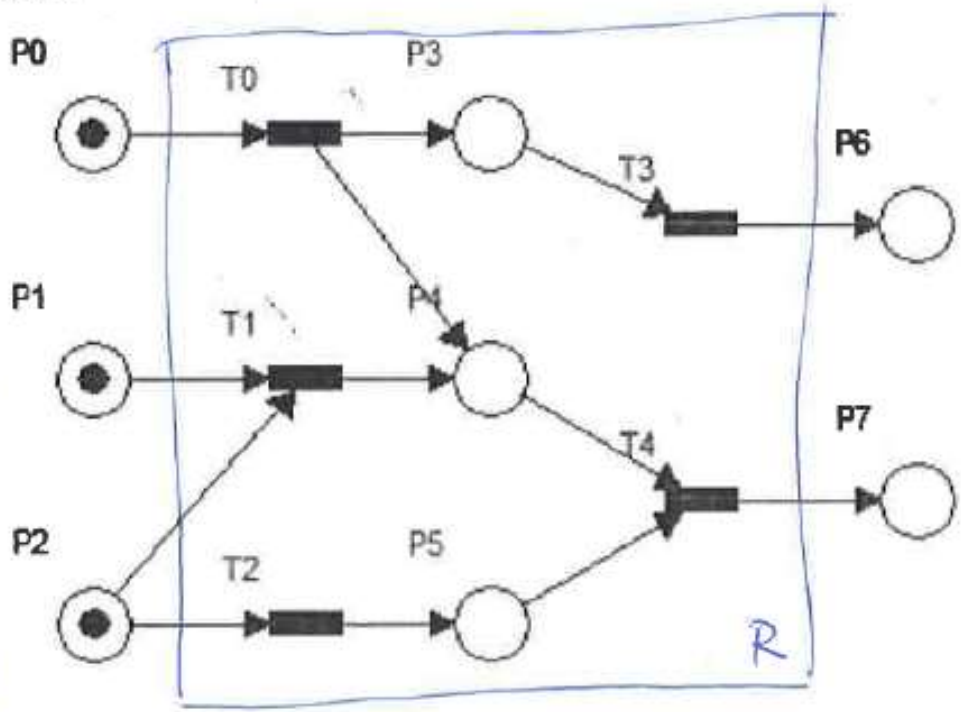


Diagram by W. van der Aalst

1.

Rajzolja rá a rendszer határát az ábrára! Melyek a Petri hálóval adott rendszer bemenetei, kimenetei és állapotai? Adja meg a gyártórendszert szemléltető Petri háló formális leírását!



Bemenetek: p_0, p_1, p_2

Kimenetek: p_6, p_7

Állapotok: p_3, p_4, p_5

$P = \{p_0, p_1, p_2, \dots, p_7\}$

$T = \{T_0, T_1, T_2, T_3, T_4\}$

$I(T_0) = \{p_0\}$ $O(T_0) = \{p_3, p_4\}$

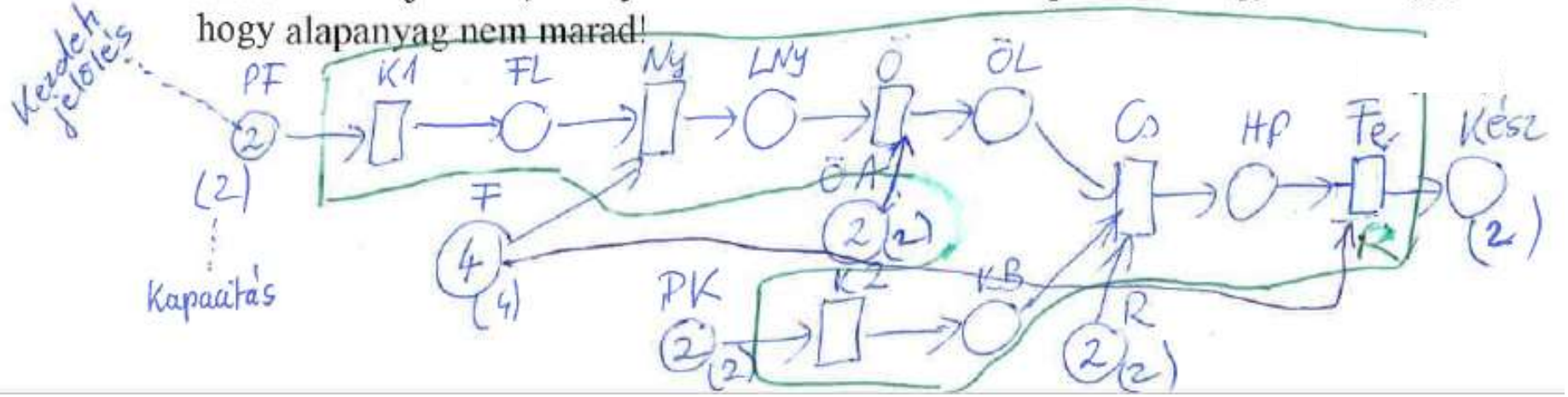
$I(T_1) = \{p_1, p_2\}$ $O(T_1) = \{p_4\}$

$I(T_2) = \{p_2\}$ $O(T_2) = \{p_5\}$

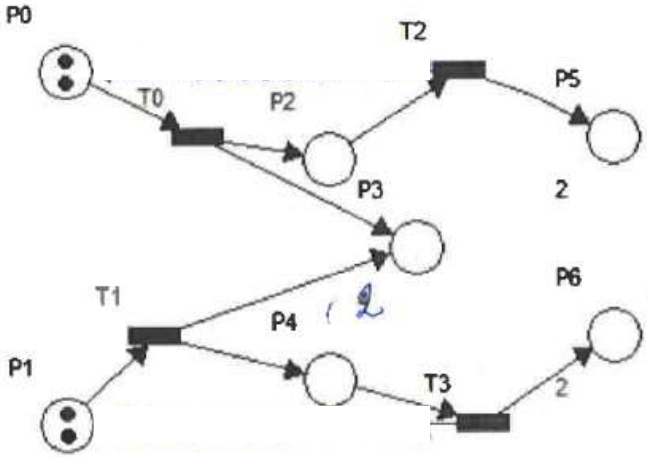
$I(T_3) = \{p_3\}$ $O(T_3) = \{p_6\}$

$I(T_4) = \{p_4, p_5\}$ $O(T_4) = \{p_7\}$

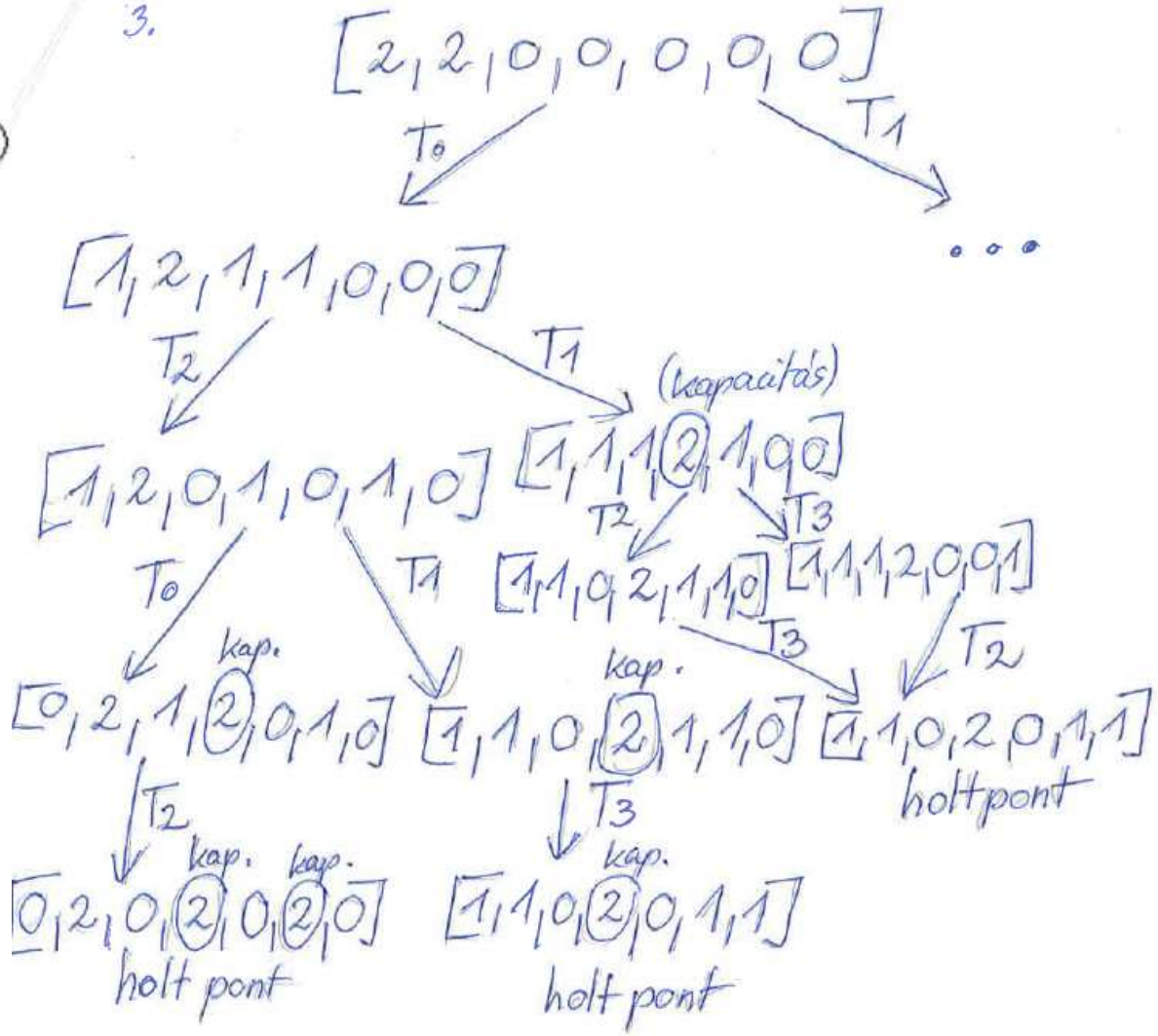
2. Határidő naplót szeretnénk gyártani az alábbi elemi műveletekkel:
(K1) Kivágjuk a papírtekercsből a határidő napló fehér lapjait (FL) (lényegtelen hány lapja van, az összes kivágását 1 műveletnek tekintjük).
(Ny) A határidő napló belső lapjaira rányomtatjuk a szükséges feliratokat (LNy).
(Ö) Összefűzzük a napló belső lapjait (ÖL).
(K2) Kivágjuk a kék vastagabb papírtekercsből a külső borítót (KB).
(Cs) Összeragasztjuk a külső borítót az összefűzött belső lapokkal (HP).
(Fe) Feliratozzuk a külső lapot (Kész).
Alapanyagok: fehér színű papírtekercs (PF), kék színű papírtekercs (PK), összefűzéshez szükséges anyagok (itt most egynek tekintjük) (ÖA), ragasztó (R), festék (F) (Mindegyik alapanyag 1-1 termék előállításához elegendő!)
Késztermékek: kék borítású határidő naplók.
Rajzolja fel a folyamat Petri háló modelljét! Jelölje a rendszer határát! Adja meg azt a kezdeti jelölést, amelynek hatására 2 határidőnapló kerül legyártásra úgy, hogy alapanyag nem marad!



3. Rajzolja fel a kép alapján a Petri hálóhoz tartozó elérhetőségi gráfot! Mire használhatjuk az elérhetőségi gráfokat, mit vizsgálhatunk a segítségükkel? Milyen jelölést nevezünk holtpontnak? Adjon meg egy ilyen jelölést a gráfban!



3.

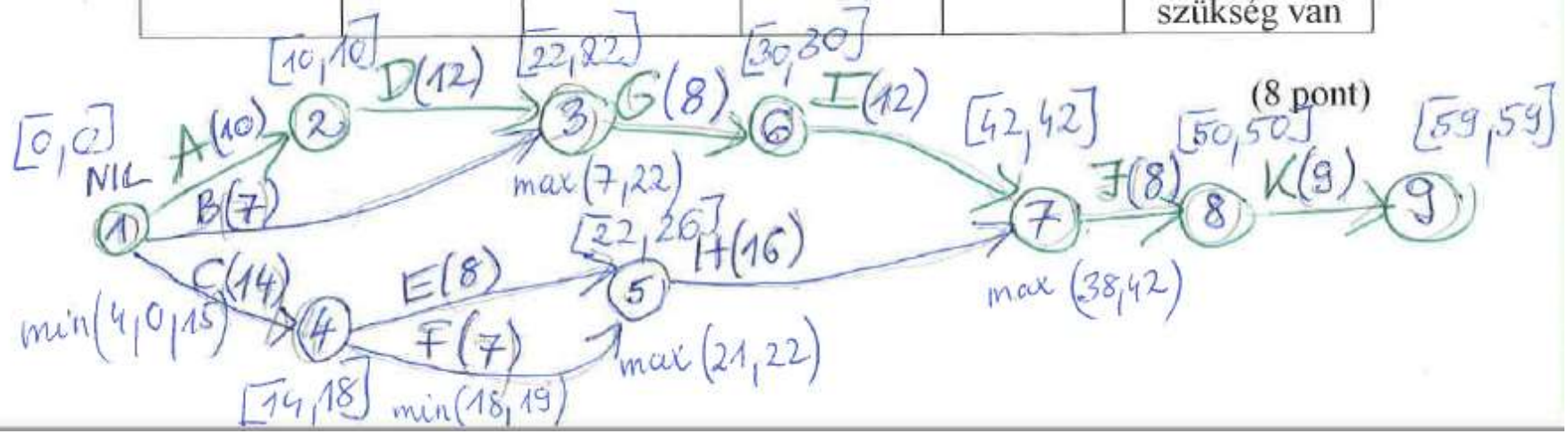


4. Az alábbi táblázat adatai alapján oldja meg az ütemezési feladatot a kritikus út módszerével:
   a) Rajzolja fel a gráfot!
   b) Egészítse ki az ábrát a csomópontokhoz tartozó legkorábbi és a legkésőbbi bekövetkezési időkkel!
   c) Határozza meg a kritikus utat, jelölje az ábrán, és adja meg, mely műveletekre kell nagyobb figyelmet fordítani a folyamat végrehajtása során!

Átlátható, elegendően nagy ábrát készítsen! Használjon színeket, kivéve pirosat!

műveletek	idő	előfeltételek	műveletek	idő	előfeltételek
A	10	-	G	8	D, B
B	7	-	H	16	E, F
C	14	-	I	12	G
D	12	A	J	8	H,I
E	8	C	K	9	J
F	7	C	végtermék elkészítése		összes művelet végrehajtására szükség van

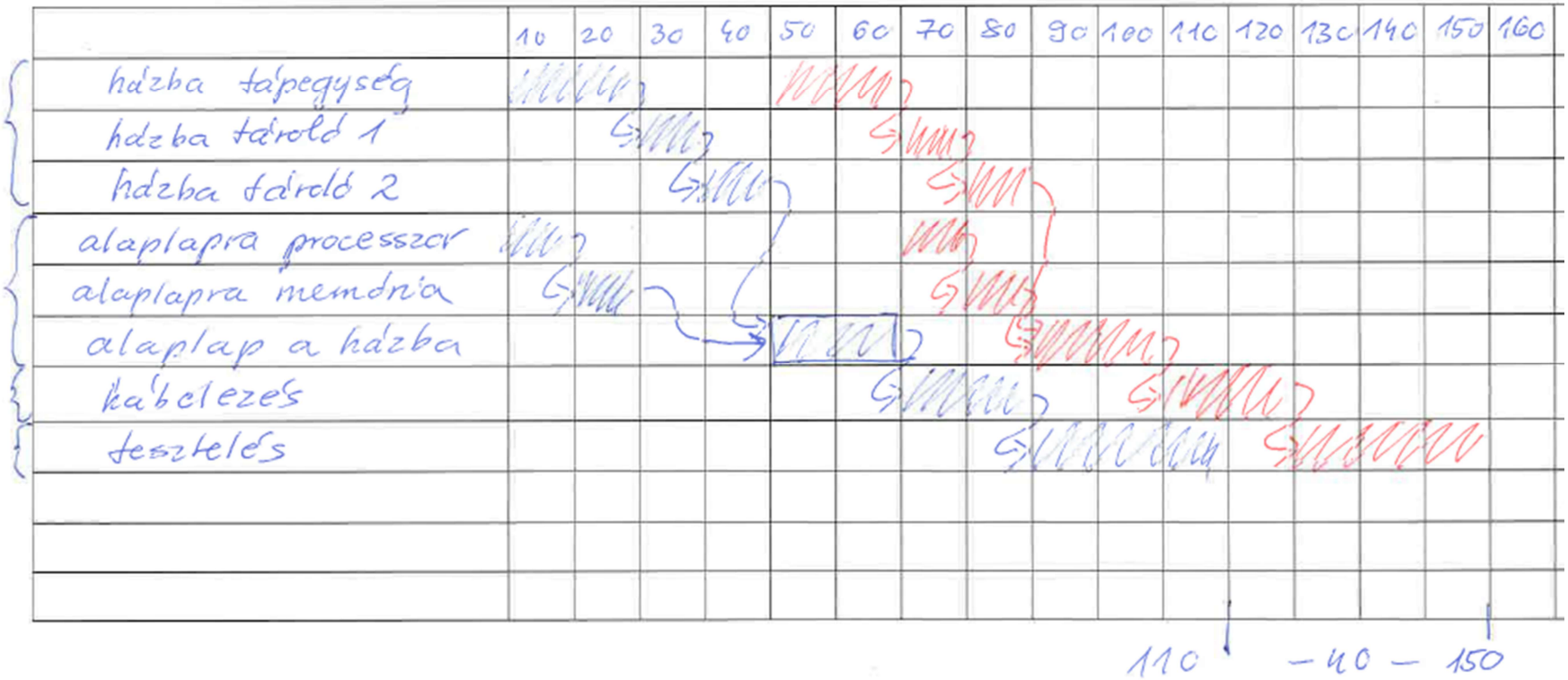
(8 pont)



5. Számítógépeket szeretnénk összeszerelni, az összeszerelésnek 8 fázisa van. Az alkatrészek futószalagon érkeznek az összeszerelési pontokhoz, így nincs szükség adott helyre való szállításnak/átrakásnak.
   - Rajzoljon fel a folyamathoz egy Gantt táblázatot, használja a másik lapon megtalálható hálót! Az egyes műveletek logikai kapcsolatát nyilakkal jelezze a táblázatban.
   - A házba a tápegységet és a tároló egységeket más szereli, mint aki az alaplapra az egységeket. Viszont az alaplapot a házba az szereli, aki az alaplapra az egységeket is rátette. A kábelezést és a tesztelést is más-más személy végzi. (Összesen 4-en végzik a műveleteket.)
   - Az egyes műveletek időigénye (a Gantt táblázatban 1 négyzet 10 időegységet jelezzen, lásd rácsozott háló a másik oldalon):

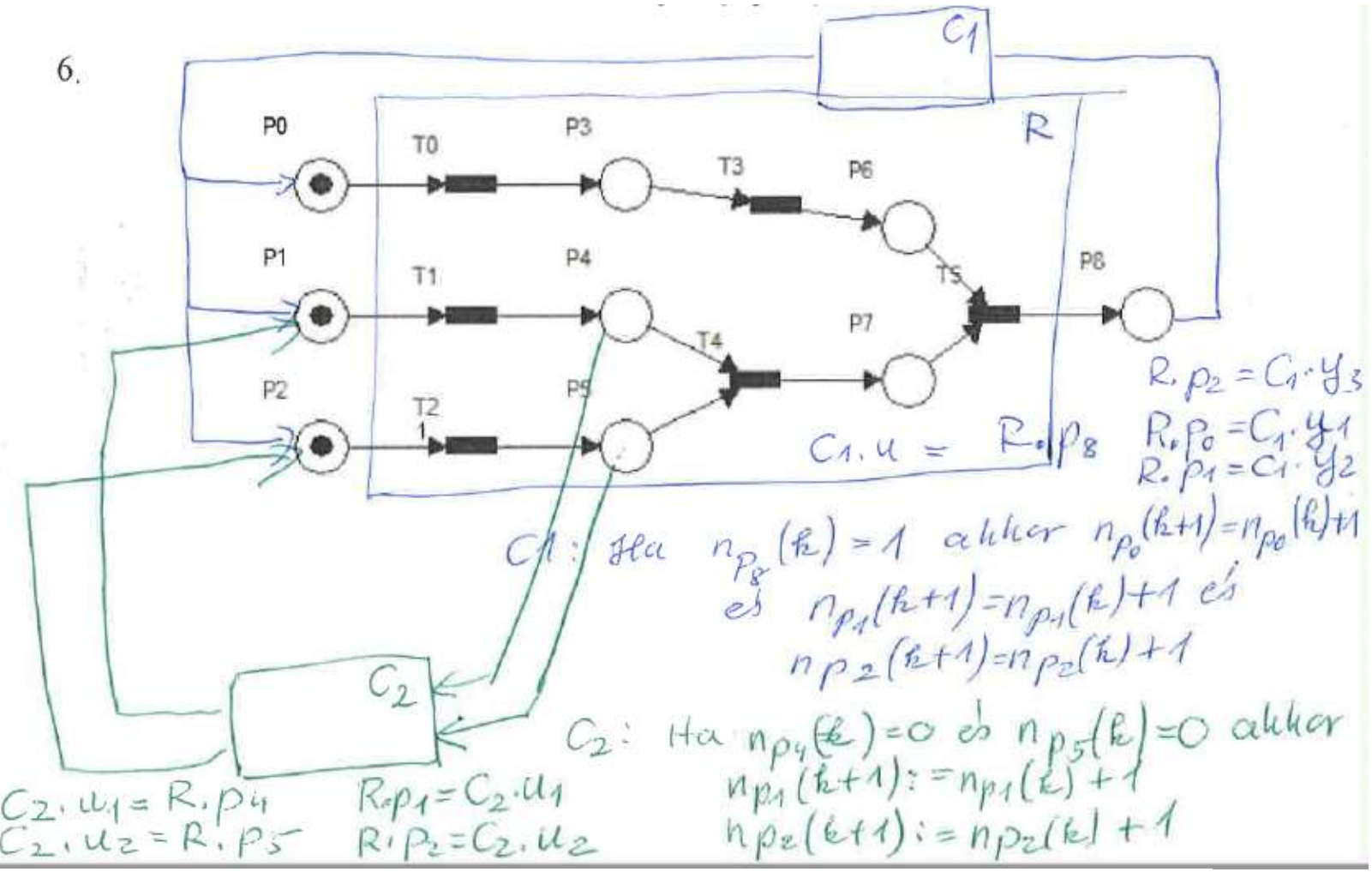
művelet	**időigény**
házba tápegység	20
alaplapra processzor	10
alaplapra memória egységek	10
házba tároló egység	10
házba második tároló egység	10
alaplap házba	20
kábelezés	20
teszt	30

   - 100 számítógépet szerelünk össze. A lapon nem tudja mind a 100 termék idő szükségletét jelezni, de határozza meg a felrajzolható részfolyamatok alapján, hogy mennyi lesz a folyamat idő szükséglete, ha a folyamat során nem következik be váratlan esemény!



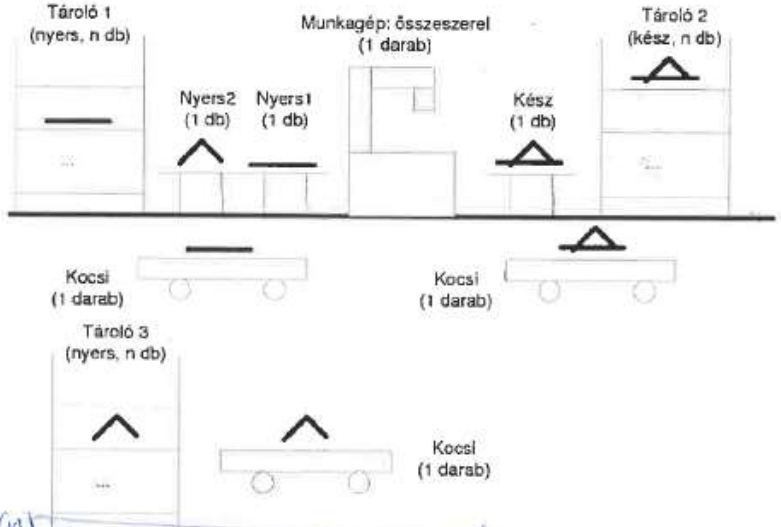
$$110 + 99 \cdot 40 = 4070$$

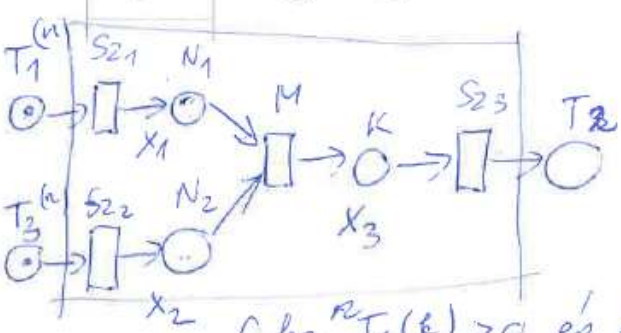
6.



- Rajzoljon fel egy kimenet visszacsatolásos szabályozót a fenti rendszerhez, amely új nyers munkadarabot tesz a p0, p1, p2 helyekre, ha egy kész munkadarab megjelenik a p8 helyen. Írja le formálisan is a szabályozó működését!
- Rajzoljon fel egy állapot visszacsatolásos szabályozót a fenti rendszerhez, amely új nyers munkadarabot tesz a p1 és p2 helyekre, ha azt a gyártósor fogadni tudja (a helyek kapacitása 1). Írja le formálisan is a szabályozó működését!

7. Írja fel az ábrán látható gyártórendszer állapottér modelljét! Mindkét tárolóban n db alkatrész áll rendelkezésre. (Szükséges a folyamat Petri háló modelljének felrajzolása, majd a következő időpillanatbeli állapotok felírása nem lineáris függvénnyel, illetve a kimenet meghatározása!)





$T_1(0) > 0$

$T_3(0) > 0$

$$X_1(k+1) = \begin{cases} \text{ha } {}^nT_1(k) > 0 \text{ és } X_1(k) = 0 \\ \text{akkor } 1 \quad \text{különben } 0 \end{cases}$$

$$X_2(k+1) = \begin{cases} \text{ha } {}^nT_2(k) > 0 \text{ és } X_2(k) = 0 \\ \text{akkor } 1 \quad \text{különben } 0 \end{cases}$$

$$X_3(k+1) = \begin{cases} \text{ha } X_3(k) = 0 \text{ és } X_1(k) = 1 \\ \text{és } X_2(k) = 1 \quad \text{akkor } 1 \quad \text{különben } 0 \end{cases}$$

$${}^nT_2(k) = X_3(k) + {}^nT_2(k-1)$$