

Architektúra tervek ellenőrzése

Majzik István

<http://www.inf.mit.bme.hu/>

Tartalomjegyzék

- Motiváció
 - Mit határoz meg az architektúra?
 - Milyen vizsgálati módszerek vannak?
- Követhetőség
- Szisztematikus vizsgálati módszerek
 - Interfész analízis
 - Hibahatás analízis
- Modell alapú vizsgálatok
 - Megbízhatósági modellezés
 - Teljesítmény modellezés

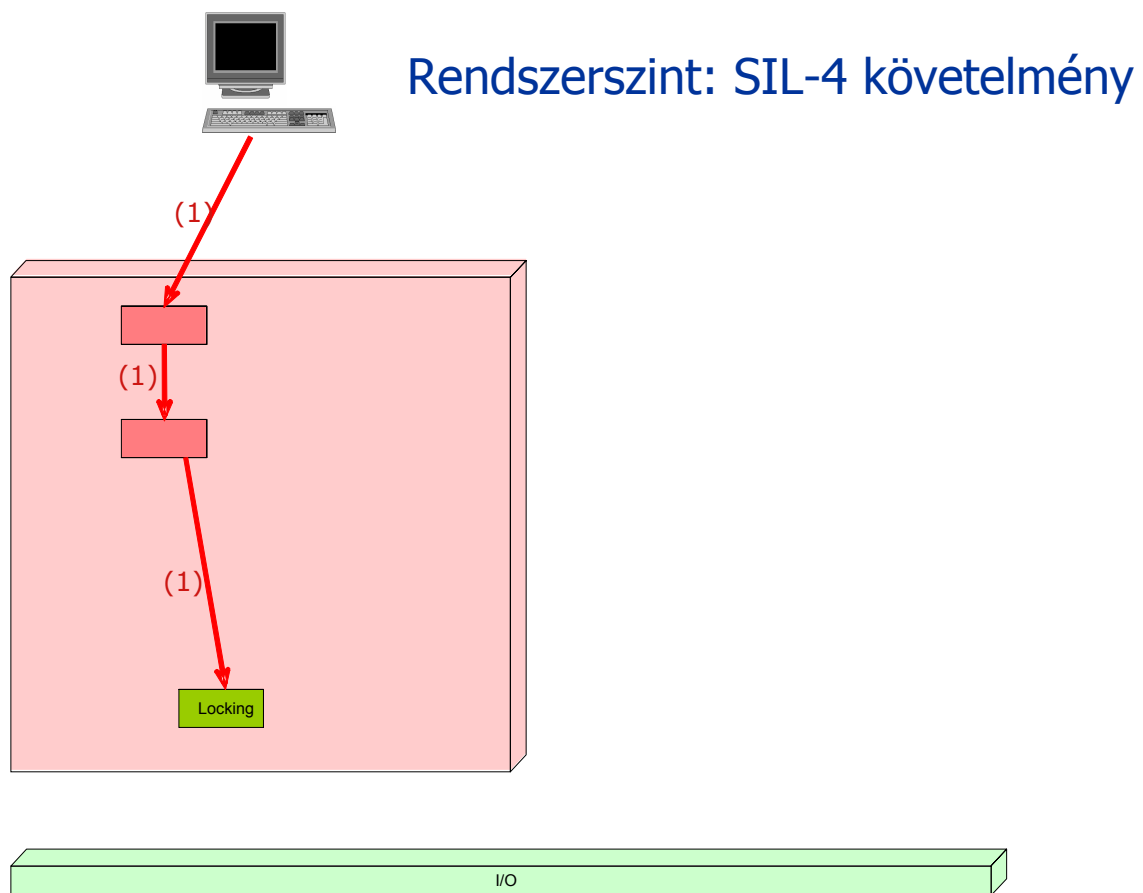
Szoftverarchitektúra

- **Architektúra**
 - Komponensek (tulajdonságokkal)
 - Köztük lévő kapcsolatok (telepítés, információáramlás, ...)
- **Tervezői döntések**
 - Komponensek és kapcsolatok azonosítása
 - Hardver-szoftver együttműködés is
 - Hibahatások is
 - Architektúra minták használata
 - Pl. MVC, N-tier, ...
 - Méretezés
 - Redundancia, teljesítmény, biztonságosság, ...
 - Újrafelhasználás (korábban fejlesztett komponensek)
 - (C)OTS komponensek használata

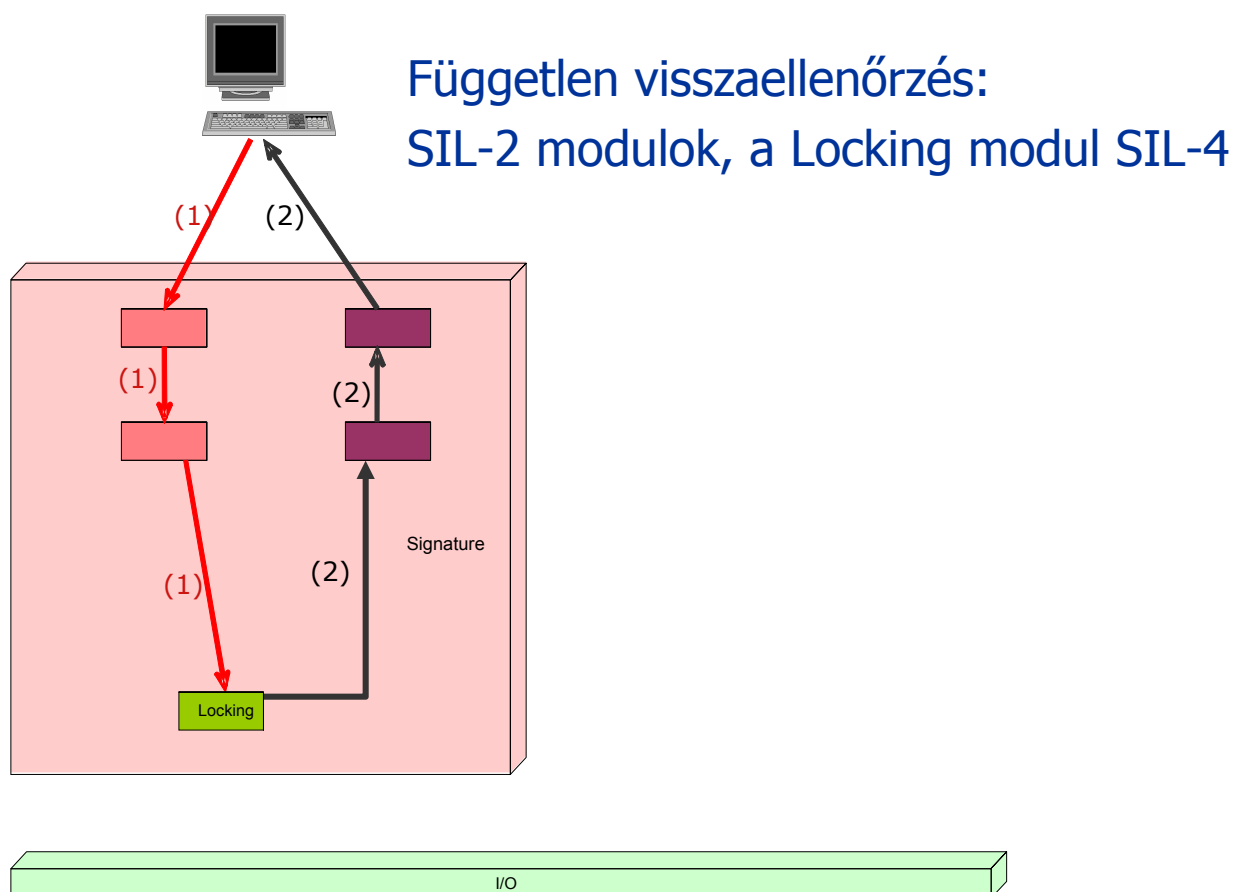
Biztonsági megfontolások

- **Komponensek SIL szintje**
 - Alapértelmezés: Egyező a rendszerével
 - Csökkentés: Megakadályozható a szoftver komponens hibájából adódó rendszerszintű hiba
 - SPOF elkerülése
 - Függetlenséget igazolni kell
- **(C)OTS komponensek használata**
 - SIL 0: Általánosan elfogadott
 - SIL 1 és 2: Validációnak tartalmaznia kell
 - SIL 3 és 4:
 - Validációnak tartalmaznia kell +
 - Lehetséges meghibásodások elemzése kell +
 - Védelmi stratégia és ennek tesztelése kell +
 - Hibanaplók felvétele és kiértékelése szükséges

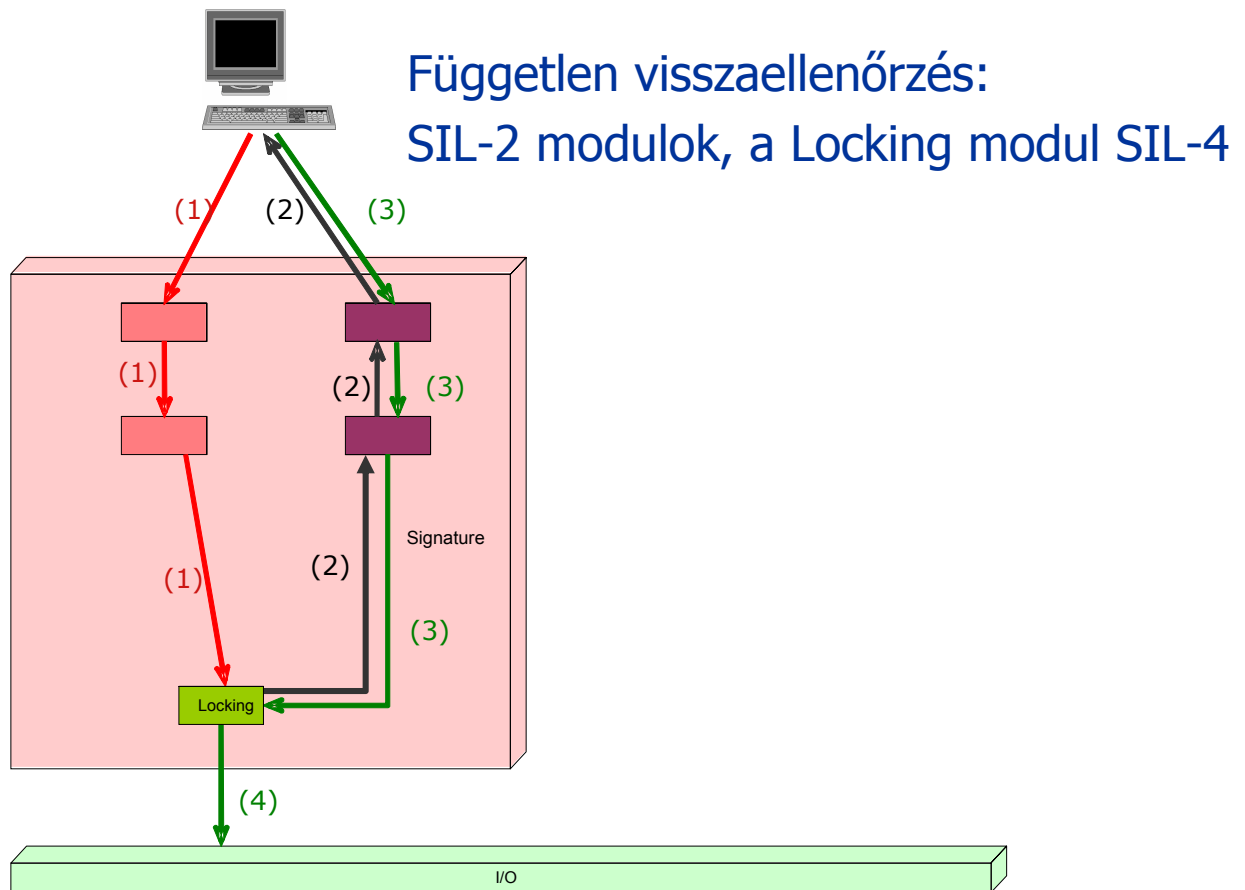
Architekturális megoldás (példa) SIL csökkentésre



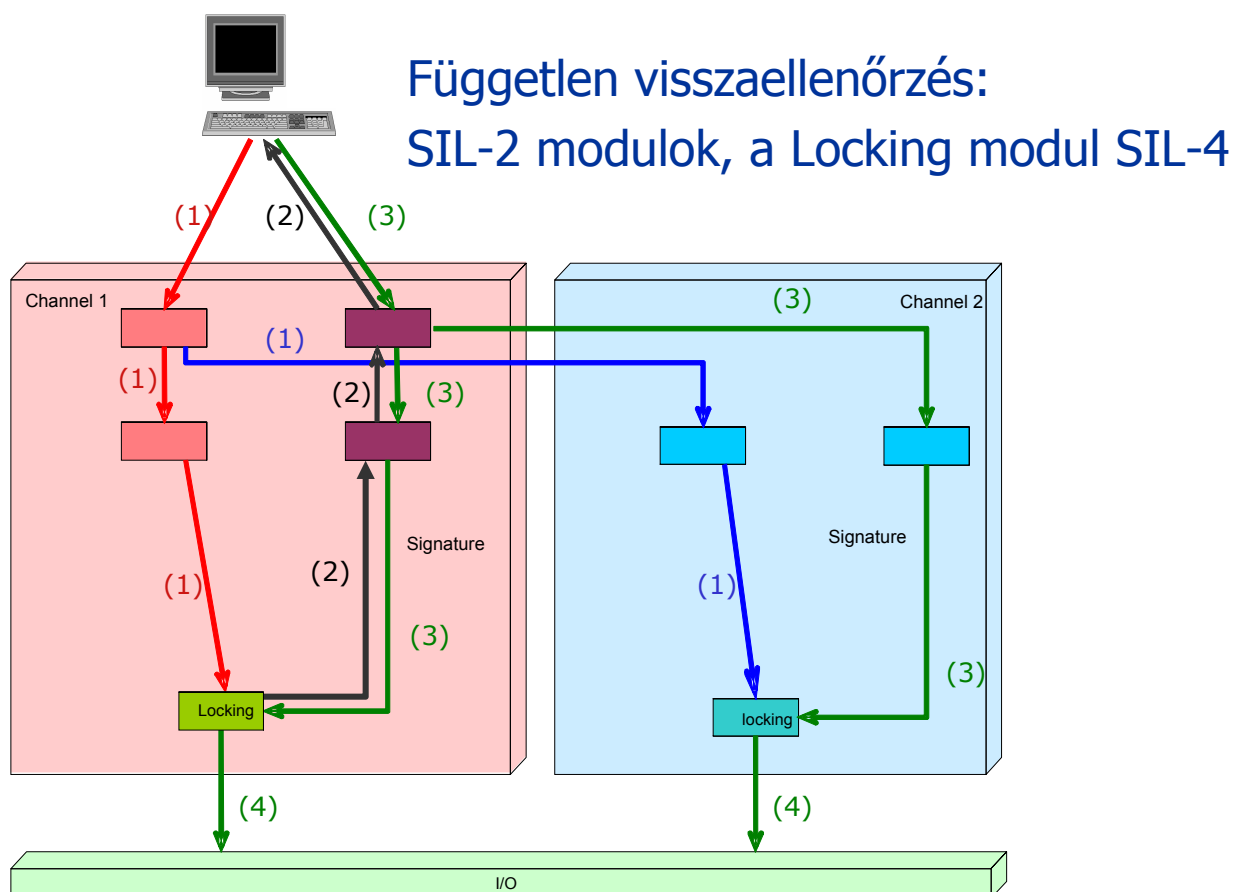
Architekturális megoldás (példa) SIL csökkentésre



Architekturális megoldás (példa) SIL csökkentésre

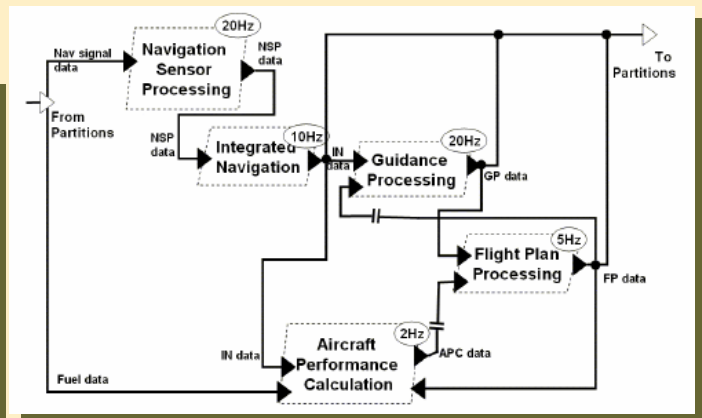
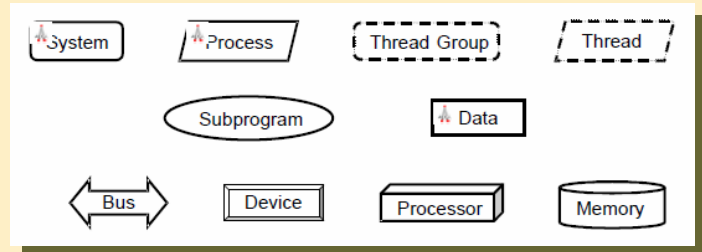


Architekturális megoldás (példa) SIL csökkentésre



Jellegzetes nyelvek architektúra tervezéshez

- UML
- SysML (Id. Block diagram)
- AADL: Architecture Analysis and Design Language
 - Komponensek
 - Kapcsolatok: Portokon
 - Adatok, események áramlása
 - Leképzés hardverre
 - Tulajdonságok analízishez



```
thread implementation CoinPublisher.impl
  calls(u: subprogram updateTotal);
  properties

  Compute_Execution_Time => 30ms .. 40ms;
  Dispatch_Protocol => ( Sporadic );
  annex behavior {**
    compute(5ms);
    compute(10ms);
    compute(15ms);
    raise(availableContent);
  **};
end CoinPublisher.impl;
```

Mit határoz meg az architektúra?

Elérendő tulajdonság	Tervezési tér
Szolgáltatásbiztonság	Hibadetektálás, hibabehatárolás, hibakezelés (redundancia)
Teljesítmény	Erőforrás hozzárendelés, erőforrás menedzsment
Biztonságosság	Veszély csökkentés, veszély kontroll (monitorozás)
Adatbiztonság	Támadás felderítés, helyreállítás
Tesztelhetőség	Vezérelhetőség, megfigyelhetőség
Karbantarthatóság	Elkülönítés (pl. MVC minta)

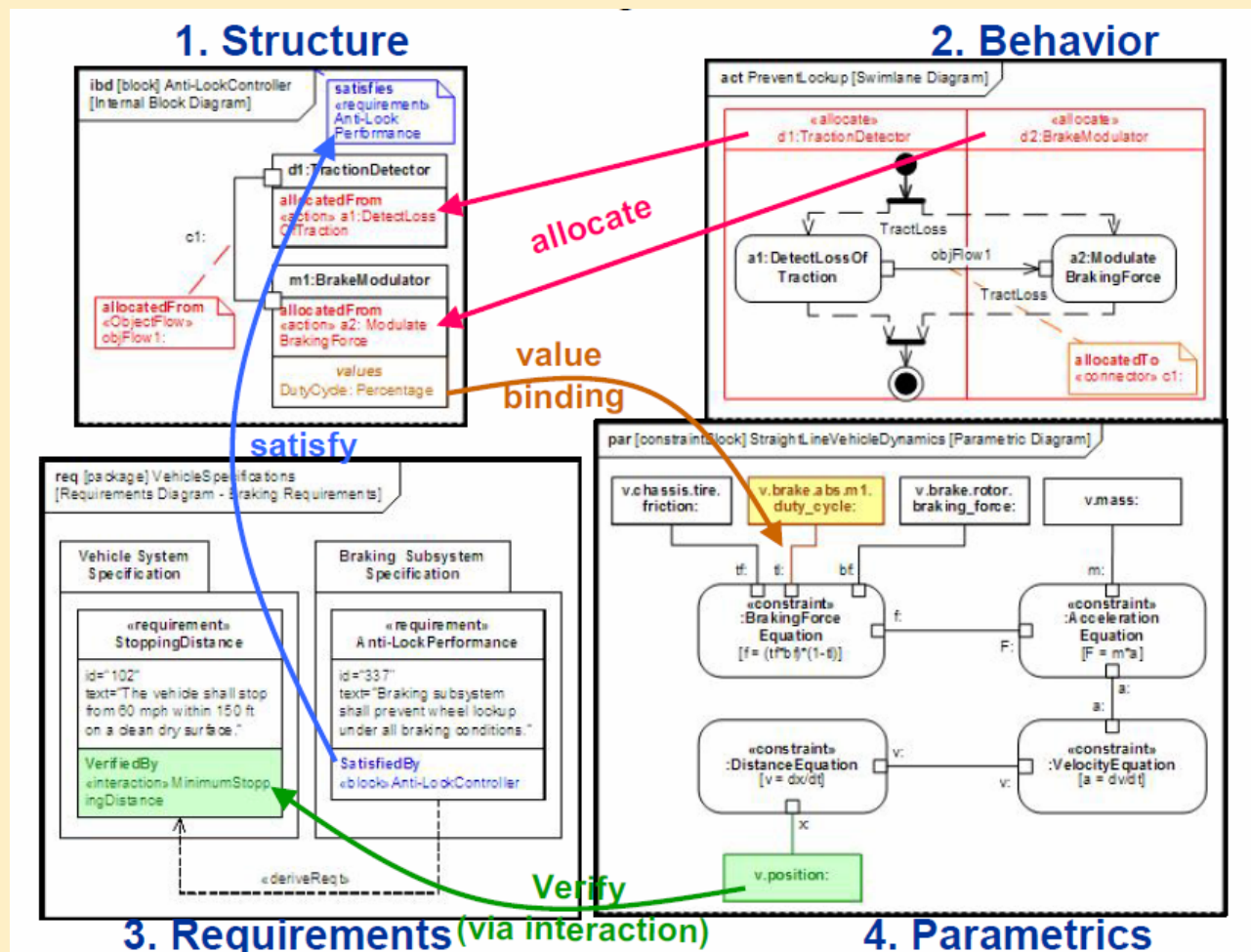
Milyen vizsgálati technikák vannak?

- (Funkcionális) követelményekkel való összevetés
 - Architektúra megfelelősége: Ld. követhetőség
- Szisztematikus elemzés (a teljesség igényével):
Az architektúra „bejárása” minden komponens vizsgálatával
 - Interfész analízis
 - Elvart és nyújtott interfészek megfelelősége
 - Funkcionális, viselkedésbeli, időzítésbeli analízis
 - Hibahatás analízis kombinatorikus módszerekkel
 - Komponens szintű hibák ↔ Rendszerszintű hatások
- Modell alapú vizsgálatok
 - Extra-funkcionális jellemzők meghatározása
 - Tipikusan sztochasztikus módszerek
 - Lokális (komponens illetve kapcsolati szintű) paraméterek alapján rendszerszintű jellemzők számítása
 - Az analízis modell konstruálása: Az architektúra alapján

Tartalomjegyzék

- Motiváció
 - Mit határoz meg az architektúra?
 - Milyen vizsgálati módszerek vannak?
- Követhetőség
- Szisztematikus vizsgálati módszerek
 - Interfész analízis
 - Hibahatás analízis
- Modell alapú vizsgálatok
 - Megbízhatósági modellezés
 - Teljesítmény modellezés

Példa: Követhetőség SysML relációk alapján



Tartalomjegyzék

- Motiváció
 - Mit határoz meg az architektúra?
 - Milyen vizsgálati módszerek vannak?
- Követhetőség
- Szisztematikus vizsgálati módszerek
 - Interfész analízis
 - Hibahatás analízis
- Modell alapú vizsgálatok
 - Megbízhatósági modellezés
 - Teljesítmény modellezés

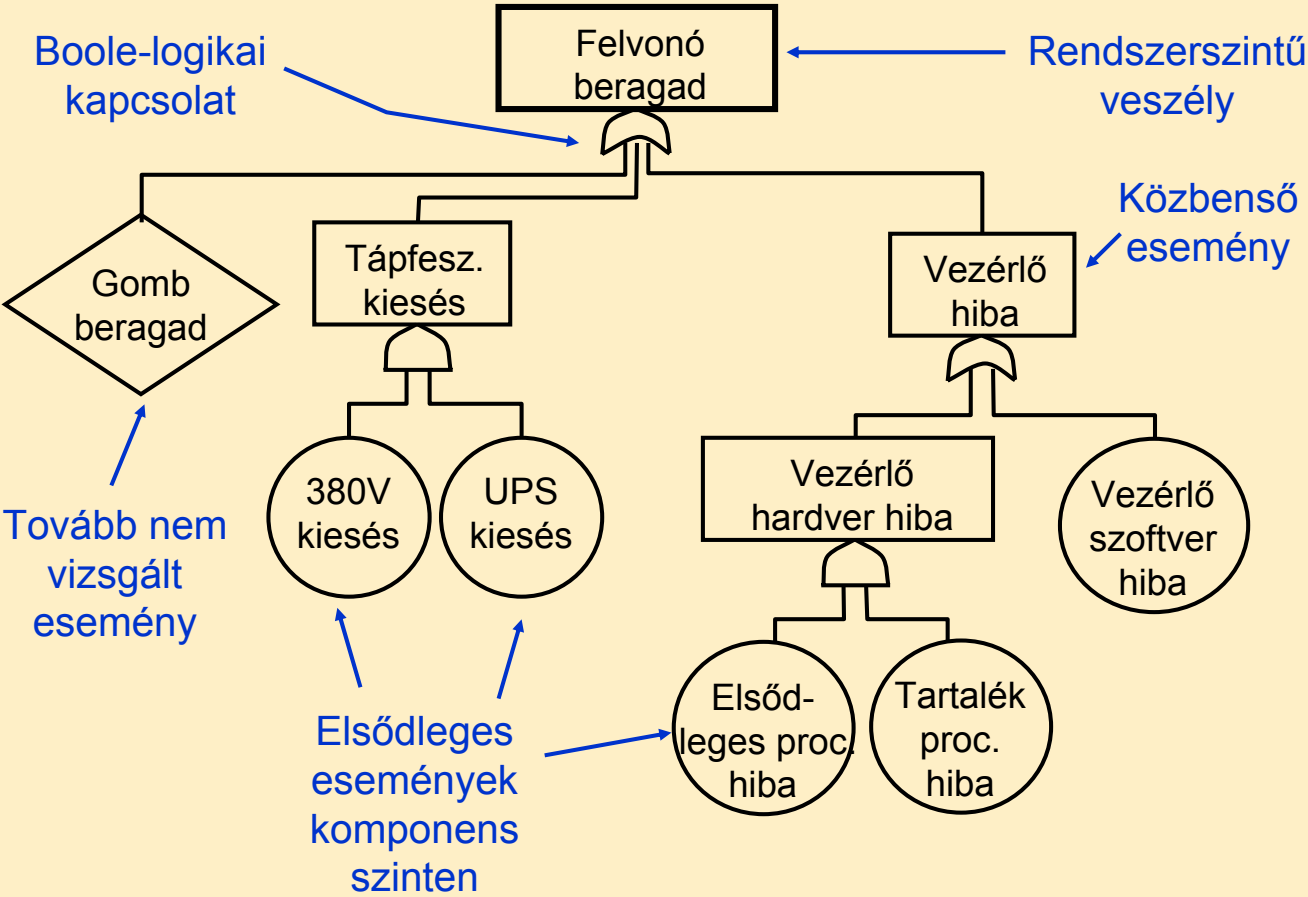
Szisztematikus vizsgálat példa: Interfész analízis

- **Célkitűzés**
 - Komponens interfészek megfelelősége
 - Teljesség: A kapcsolatok szisztematikus „bejárása” biztosítja
- **Szintaktikus analízis**
 - Hívási paraméterek száma, típusa (signature)
- **Szemantikus analízis**
 - Komponensekhez rendelt funkcionalitás leírása alapján
 - Szerződés (contract) alapú analízis
- **Viselkedési analízis**
 - Komponens viselkedés specifikáció alapján
 - Összetett protokollok esetén szokásos (konformancia)
 - Viselkedési ekvivalencia relációk definiálhatók
 - Trace ekvivalencia
 - Biszimuláció
 - Időzítések is vizsgálhatók

Szisztematikus vizsgálat példa: Hibahatás analízis

- **Célkitűzés**
 - Komponens jellemzők és rendszerszintű jellemzők közötti kapcsolatok megállapítása
 - Teljesség: Minden komponens hibamódjai szerepelnek
- **Analízis megközelítése**
 - Az architektúra szempontjából
 - Alulról felfelé: A komponensek (alrendszerek) felől
 - Felülről lefelé: Rendszerszintről lebontva
 - Ok-okozati szempontból:
 - Előrelépő (induktív): Esemény hatásainak vizsgálata
 - Visszalépő (deduktív): Veszély okainak felderítése
- **Tipikus módszerek a vizsgálatra**
 - Hibafa
 - Eseményfa
 - Ok-következmény analízis
 - Hibamód és -hatás analízis (FMEA)

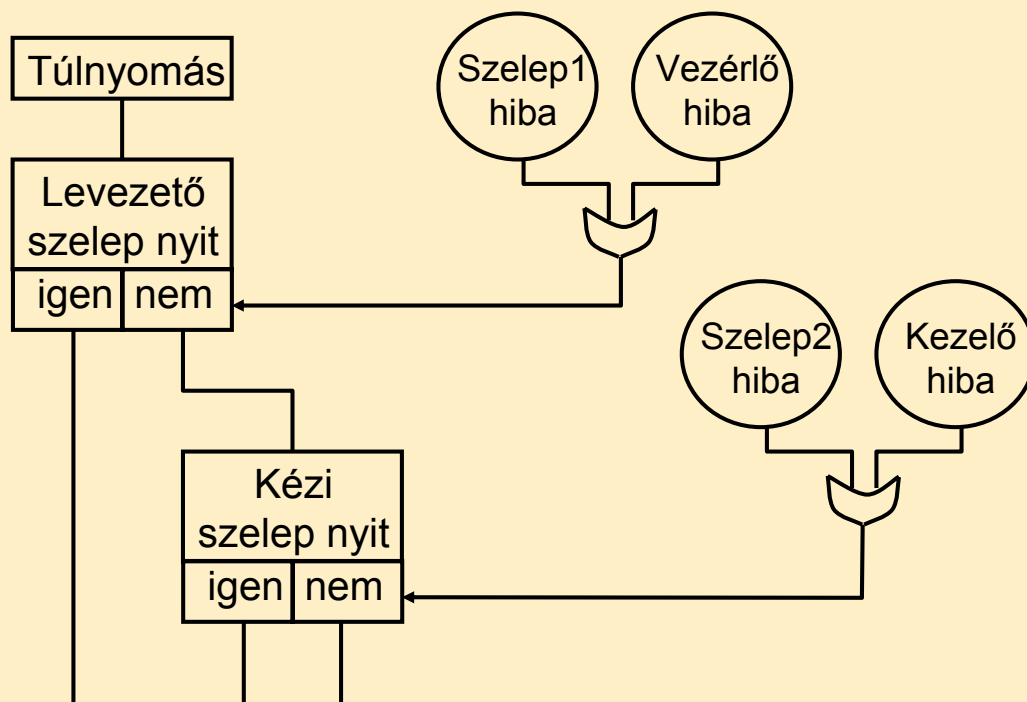
Hibafa példa: Felvonó



Eseményfa példa: Reaktorhűtés

Hűtőrsz. csőszakadás	Elektromos hálózat	Vész-hűtés	Reagens eltávolítás	Folyamat leállítás	
kiváltó esemény	van	nem indul	meghiúsul	sikeres	✓
		nem indul	meghiúsul	meghiúsul	✗
		sikeres	meghiúsul		✓
	kiesett	indul	sikeres	sikeres	✓
		indul	meghiúsul	meghiúsul	✗
		indul	meghiúsul	meghiúsul	✗

Ok-következmény analízis példa: Túlnyomás védelem



Hibamód és -hatás analízis (FMEA) példa

- Hibák és hatásaik felmérése
- A táblázat teljessége
 - Minden komponens
 - Minden hibamód
- Hatás analízis nem egyszerű

Komponens	Hibamód	Valószínűség	Hatás
L határérték-túllépés vizsgálat	> L átmegy	65%	- túlnyomás
	≤ L nem megy át	35%	- technológiai hiba
...	...	...	...

Tartalomjegyzék

- Motiváció
 - Mit határoz meg az architektúra?
 - Milyen vizsgálati módszerek vannak?
- Követhetőség
- Szisztematikus vizsgálati módszerek
 - Interfész analízis
 - Hibahatás analízis
- Modell alapú vizsgálatok
 - Megbízhatósági modellezés
 - Teljesítmény modellezés

Modell alapú architektúra elemzés

Cél: Architektúra változatok kiértékelése

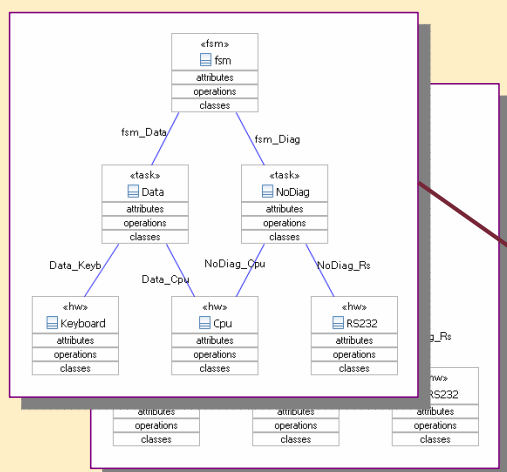
- Analízis modellek készítése és paraméterezése az architektúra modellje alapján
 - Teljesítmény modell
 - Megbízhatósági modell
 - Biztonsági modell
 - Adatbiztonsági modell
- Moduláris modellalkotás – az automatizálás alapja
 - Architektúra: Komponensek és kapcsolatok
 - Analízis modell: Ezekhez analízis modellkönyvtár elemei
- Rendszerszintű analízis modellek megoldása
 - Lokális (komponens illetve kapcsolati szintű) paraméterek alapján rendszerszintű jellemzők számítása

Rendszerszintű analízis modellek

	Megbízhatósági modell	Teljesítmény modell	Biztonsági modell
Komponens paraméterek	Meghibásodási tényező, lappangási idő, javítási tényező, hibafedés, ...	Funkció lokális végrehajtási idő, taszk prioritás, processzor ütemezés	Veszély gyakoriság
Kapcsolat paraméterek	Hibaterjedési valószínűség, hibaterjedés feltételei, javítási stratégia	Hívás továbbítási gyakoriság, hívás szinkronitás	Veszély forgatókönyv, veszély kombinációk
Modell	Markov-lánc, Petri-háló	Sorbanállási háló	Markov-lánc, Petri-háló
Rendszer jellemzők (számított)	Megbízhatóság, rendelkezésre állás, készenlét, MTTF, MTTR, MTBF	Kiszolgálási idő, taszk áteresztő-képesség, processzor kihasználtság	Rendszerszintű veszély gyakoriság

UML alapú megbízhatósági modellezés

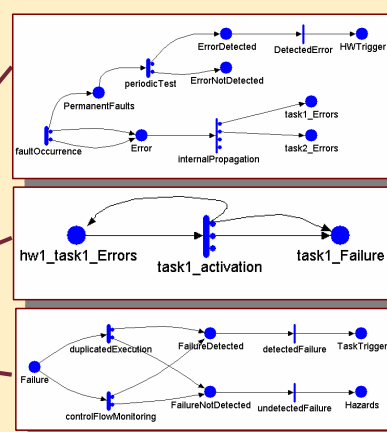
UML architektúra modell



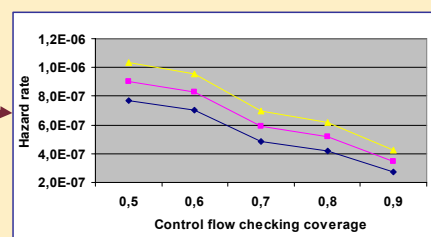
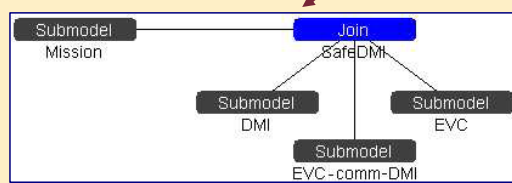
Megbízhatósági modell konstrukció



Analízis modellkönyvtár



Rendszerszintű megbízhatósági modell



Analízis eredmények

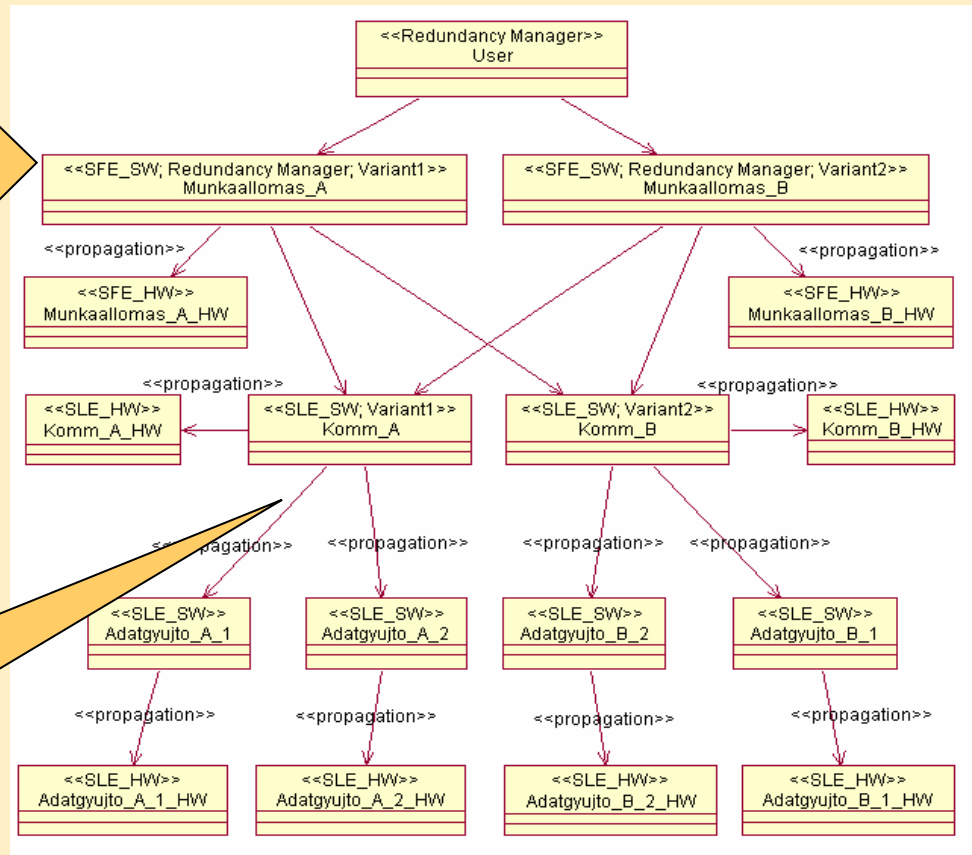
Megbízhatósági modellezés – Architektúra modell

Komponens:

- Típus (HW, SW)
- Szerep (variáns, menedzser)
- Meghibásodási jellemzők:
 - * meghibásodási gyakoriság,
 - * lappangási idő,
 - * javítási idő

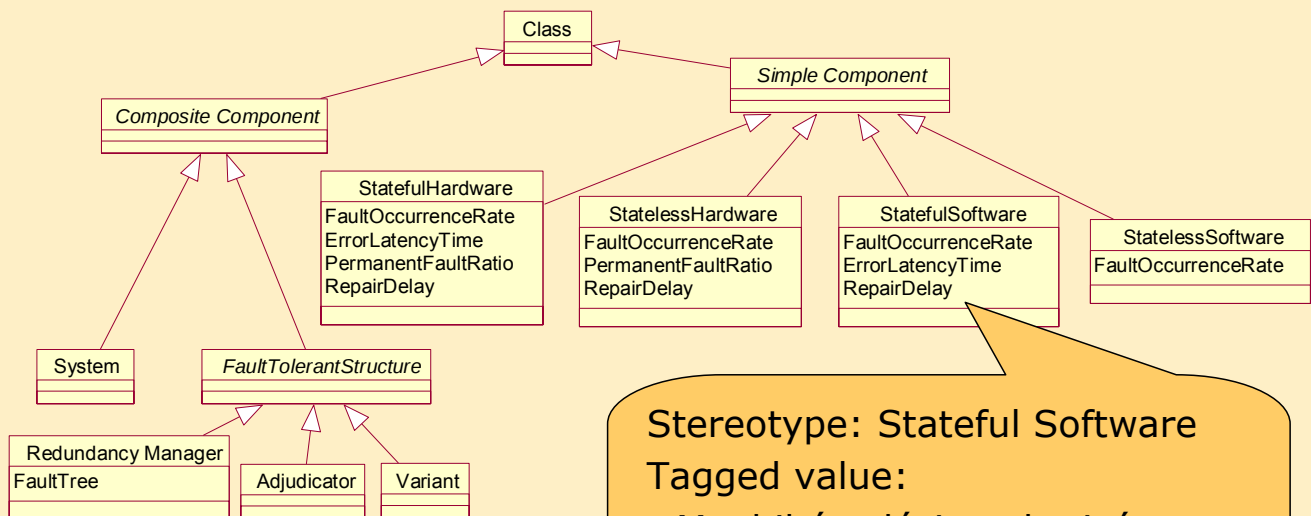
Kapcsolat:

- Hibaterjesztési jellemzők:
 - * hibaterjesztési valószínűség



Komponensek lokális tulajdonságai

- UML profil az extra-funkcionális tulajdonságokhoz

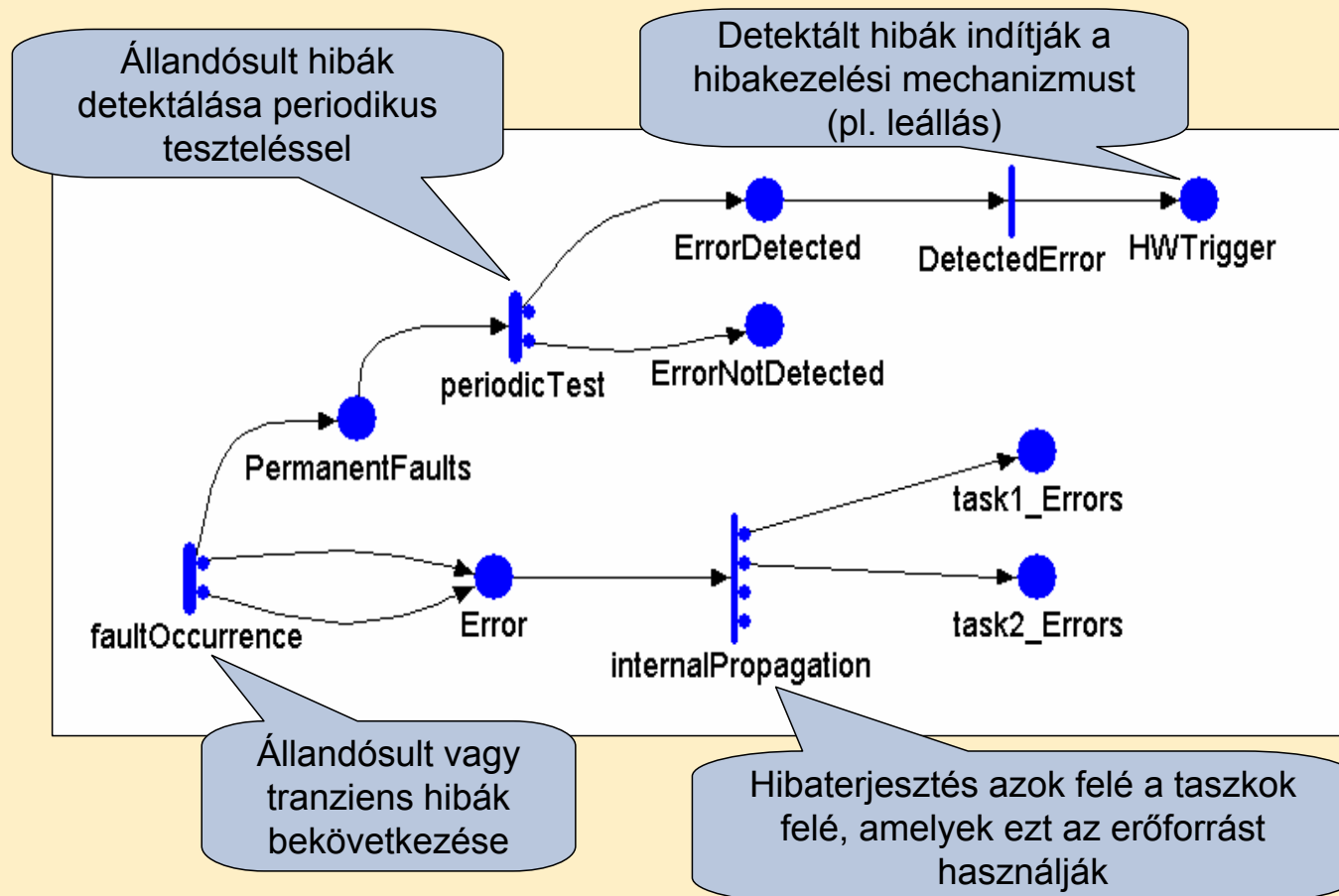


Stereotype: Stateful Software

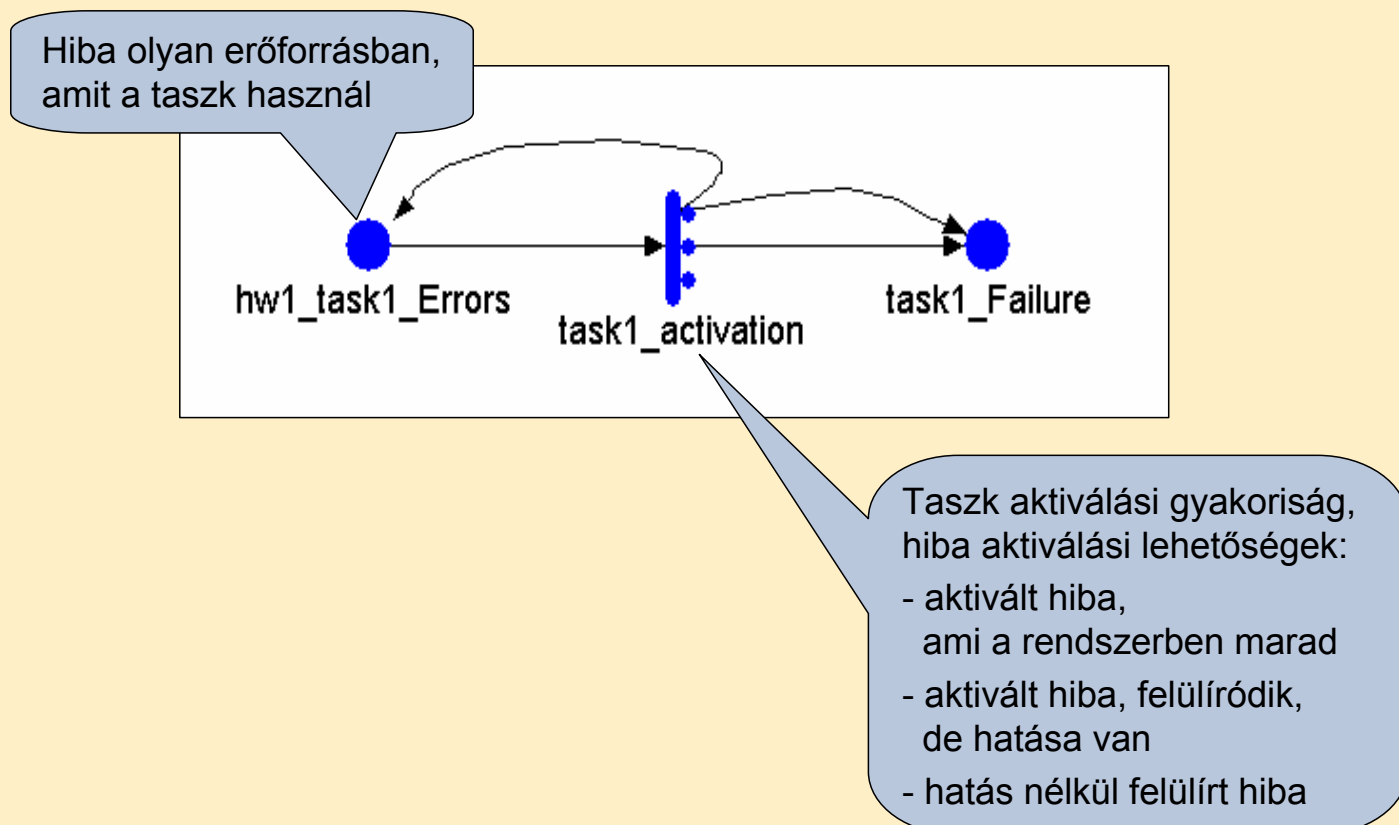
Tagged value:

- Meghibásodási gyakoriság
- Hiba lappangási idő
- Javítási tényező

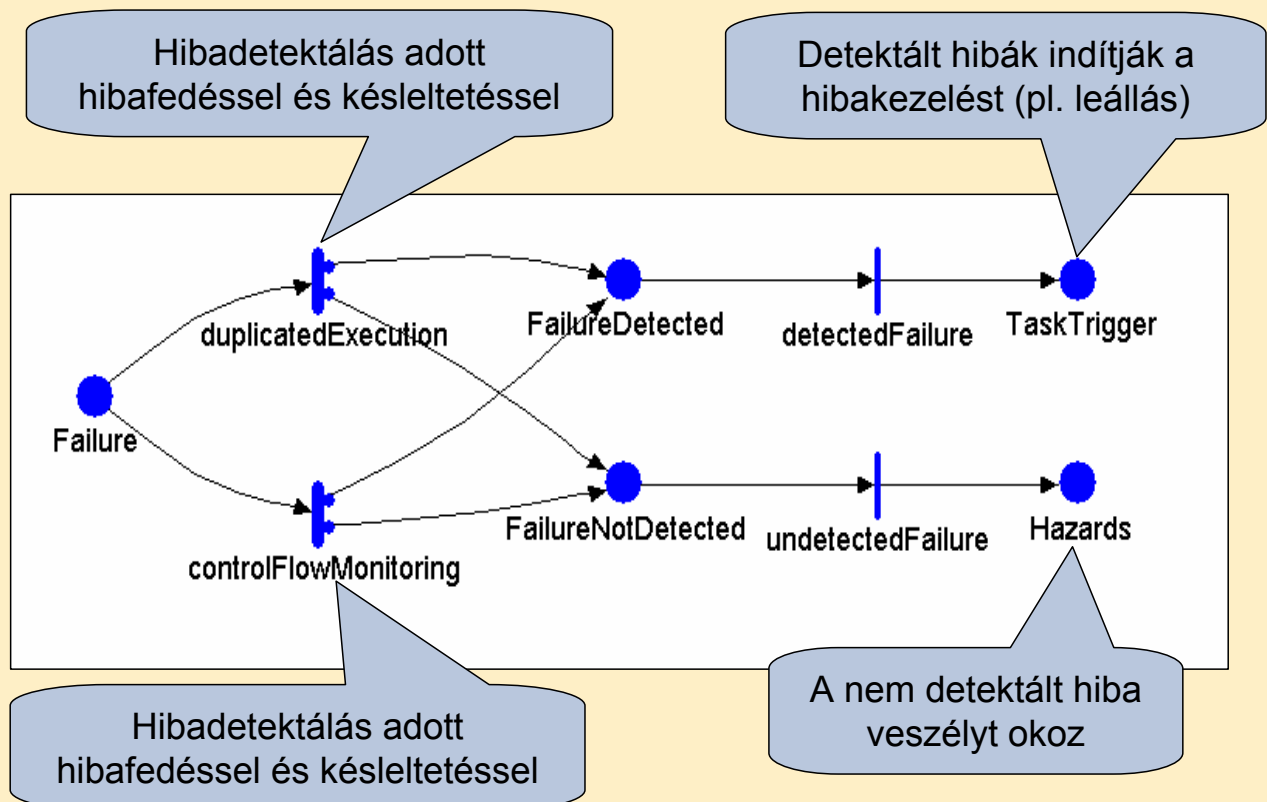
Komponens analízis alháló: Egy HW erőforrás (pl. memória)



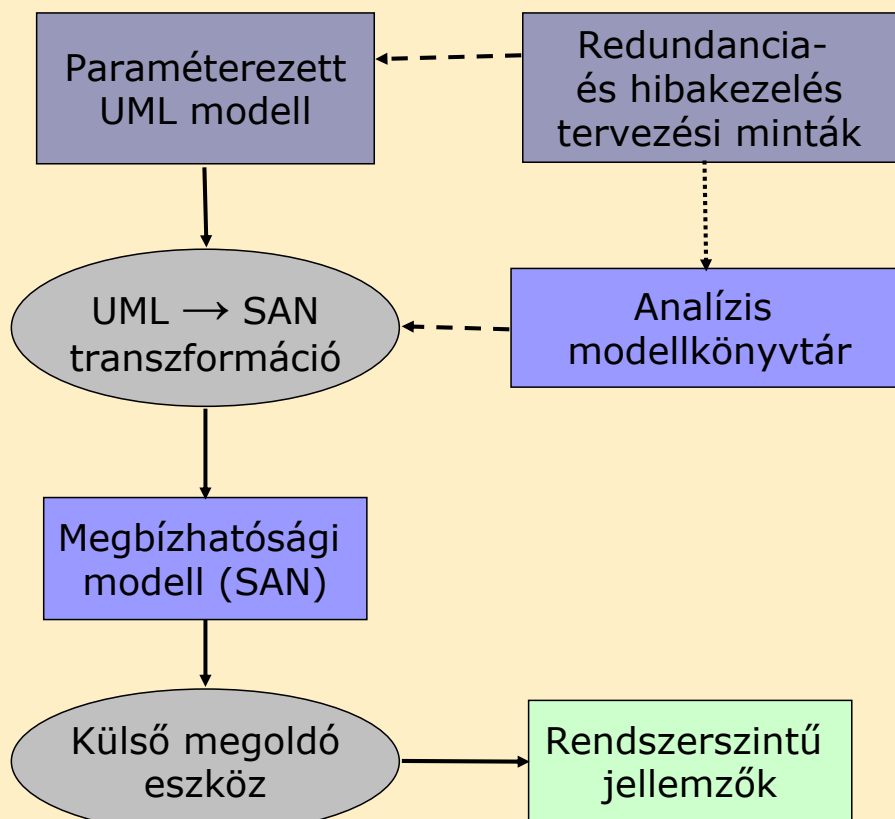
Kapcsolat analízis alháló: Hibaterjesztés



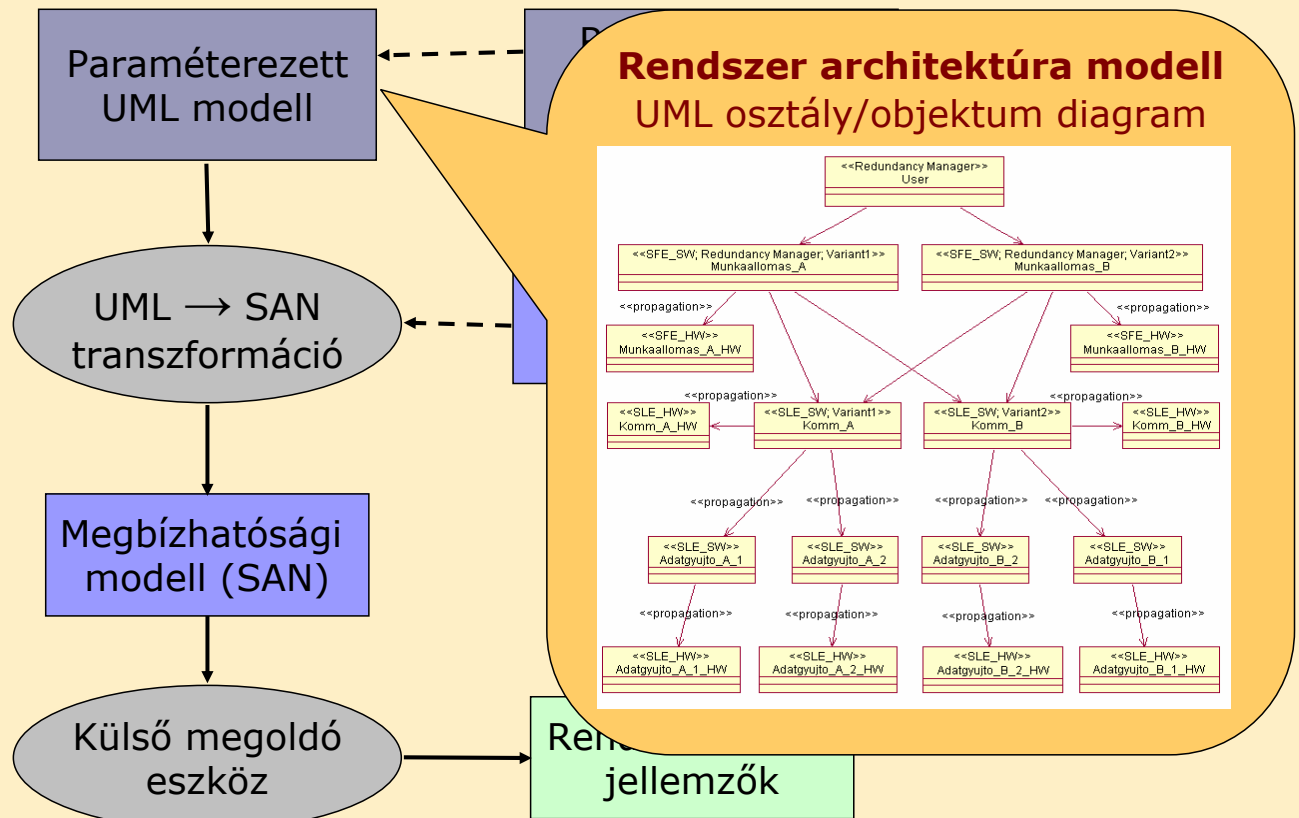
Komponens analízis alháló: Egy szoftver taszk



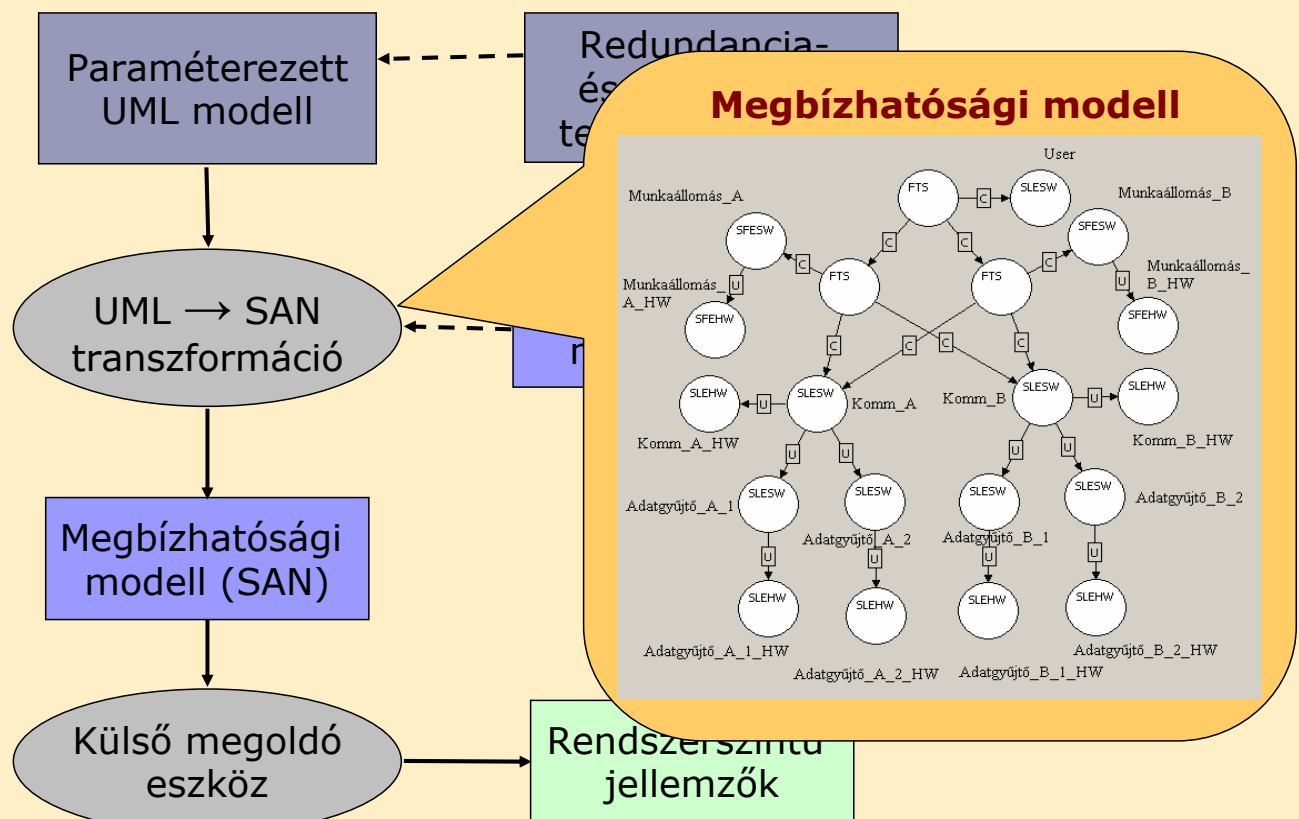
Automatikus analízis eszköz



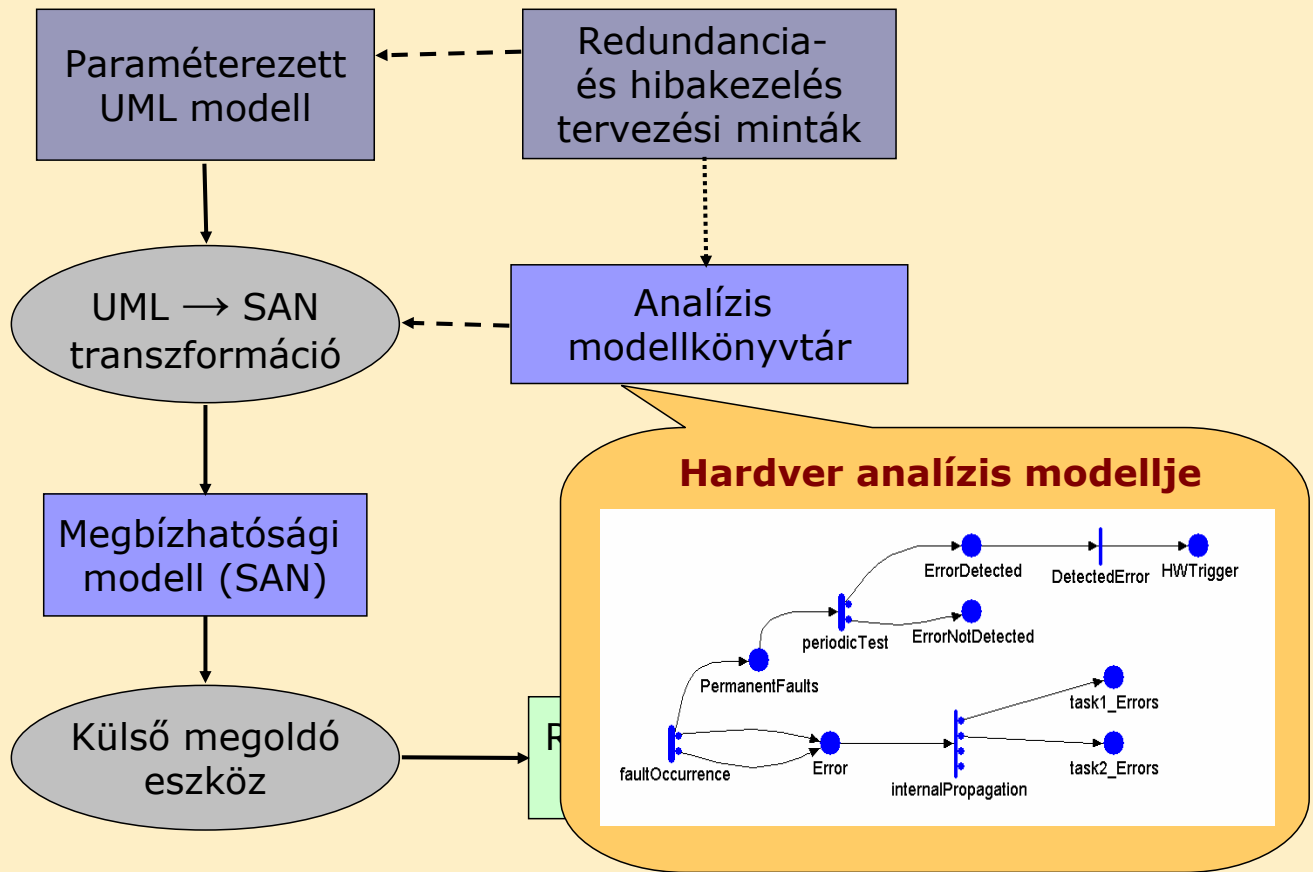
Automatikus analízis eszköz



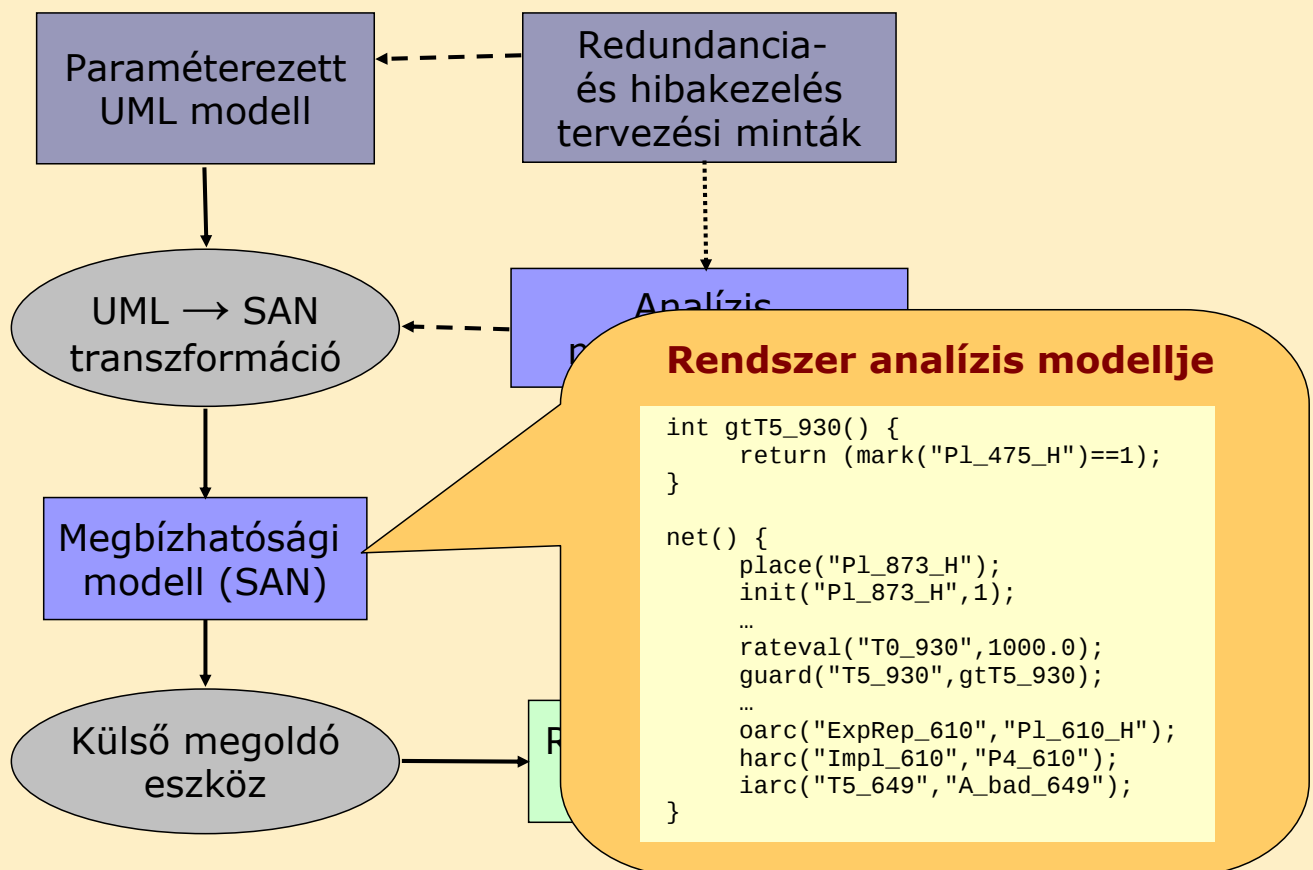
Automatikus analízis eszköz



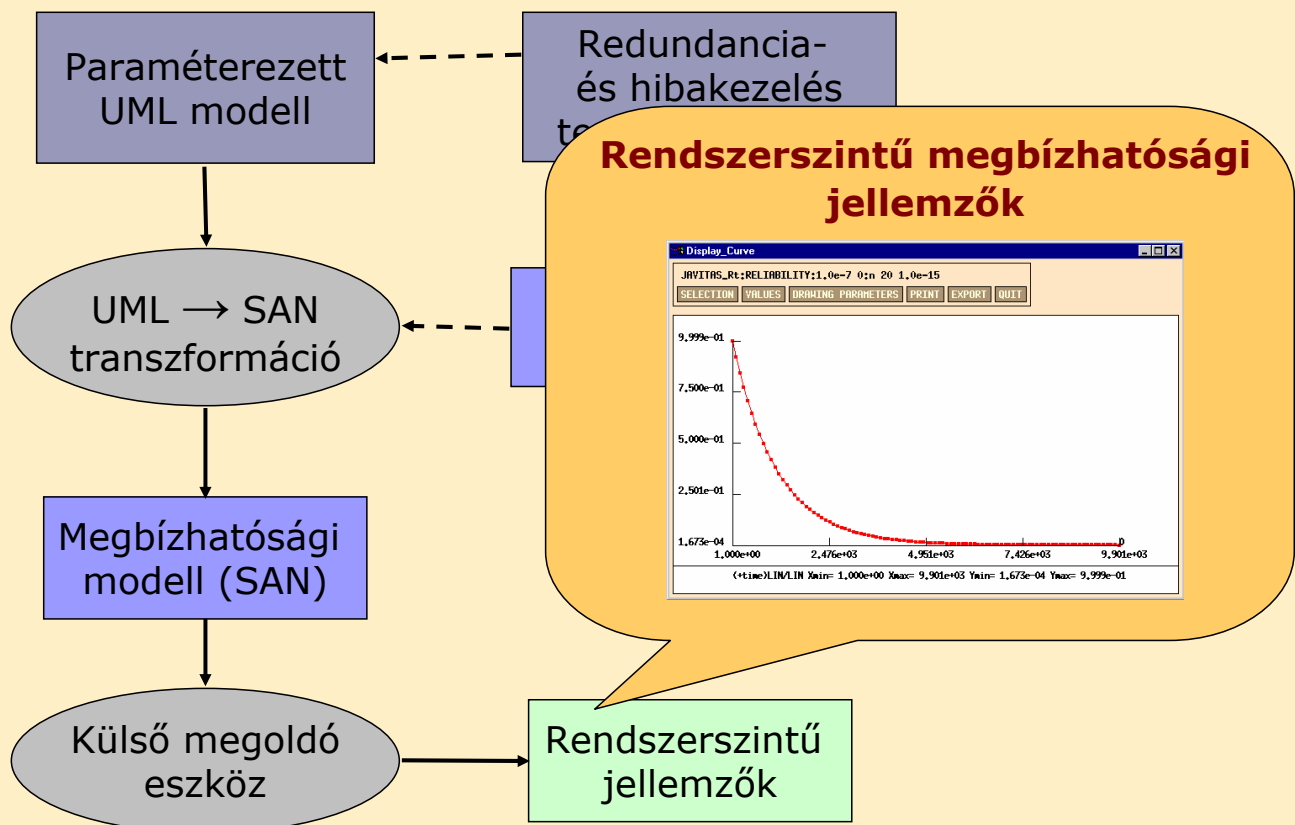
Automatikus analízis eszköz



Automatikus analízis eszköz



Automatikus analízis eszköz



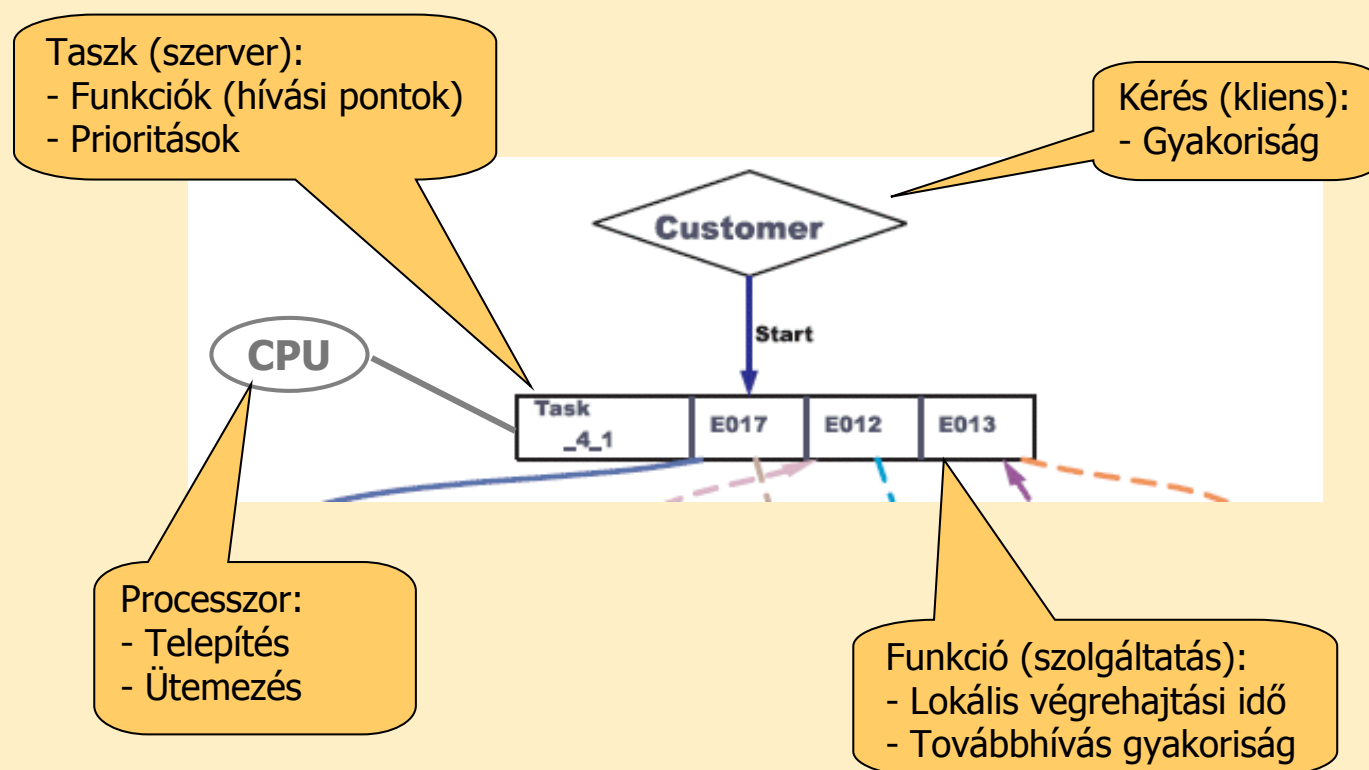
Tartalomjegyzék

- **Motiváció**
 - Mit határoz meg az architektúra?
 - Milyen vizsgálati módszerek vannak?
- **Követhetőség**
- **Szisztematikus vizsgálati módszerek**
 - Interfész analízis
 - Hibahatás analízis
- **Modell alapú vizsgálatok**
 - Megbízhatósági modellezés
 - Teljesítmény modellezés

Modell alapú architektúra vizsgálatok

	Megbízhatósági modell	Teljesítmény modell	Biztonsági modell
Komponens paraméterek	Meghibásodási tényező, lappangási idő, javítási tényező	Funkció lokális végrehajtási idő, taszk prioritás, processzor ütemezés	Veszély gyakoriság
Kapcsolat paraméterek	Hibaterjedési valószínűség, hibaterjedés feltételei, javítási stratégia	Hívás továbbítási gyakoriság, hívás szinkronitás	Veszély forgatókönyv, veszély kombinációk
Modell	Markov-lánc, Petri-háló	Sorbanállási háló	Markov-lánc, Petri-háló
Rendszer jellemzők (számított)	Megbízhatóság, rendelkezésre állás, készenlét, MTTF, MTTR, MTBF	Kiszolgálási idő, taszk áteresztő-képesség, processzor kihasználtság	Rendszerszintű veszély gyakoriság

Teljesítmény modellezés (LQN): Szerverek és szolgáltatások



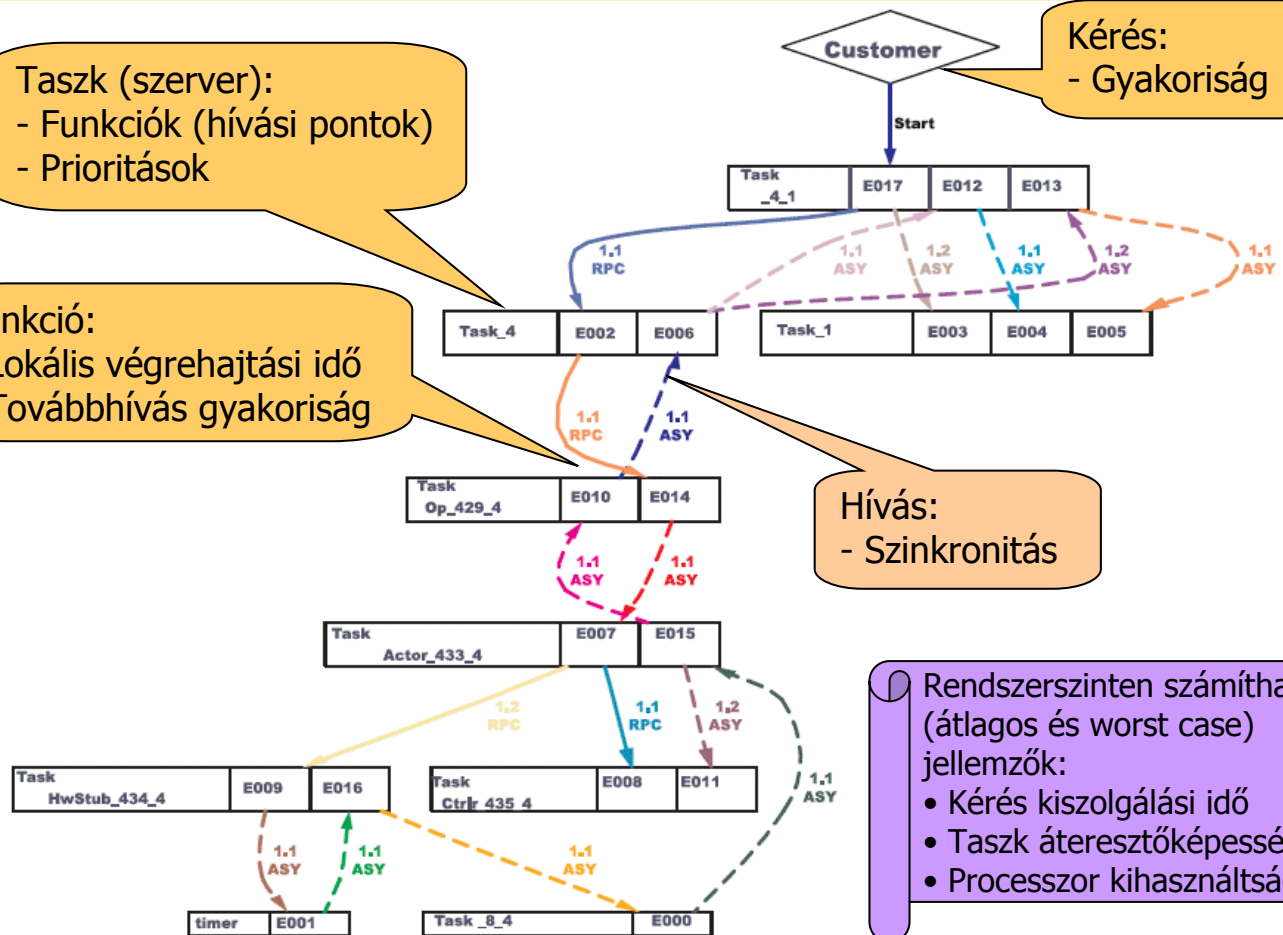
Teljesítmény modellezés (LQN): Rétegek

Taszk (szerver):
- Funkciók (hívási pontok)
- Prioritások

Funkció:
- Lokális végrehajtási idő
- Továbbhívás gyakoriság

Kérés:
- Gyakoriság

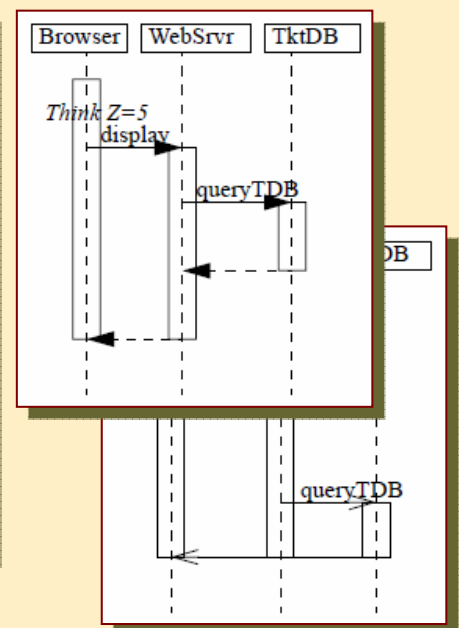
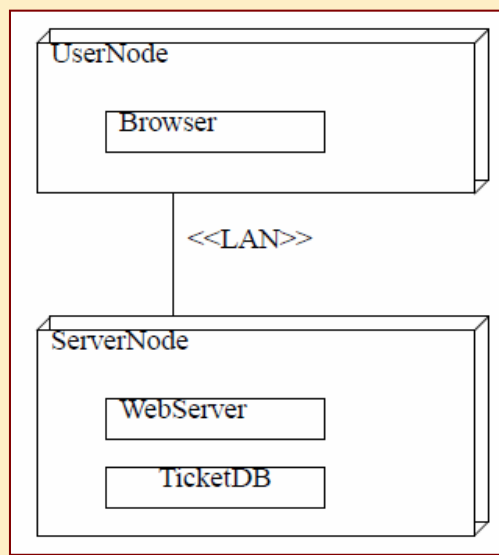
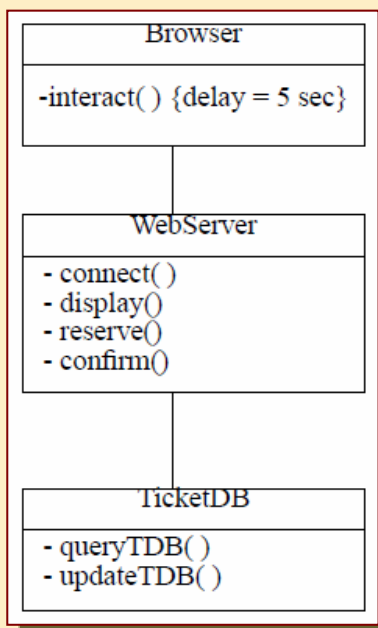
Hívás:
- Szinkronitás



Rendszerszinten számítható
(átlagos és worst case)
jellemzők:

- Kérés kiszolgálási idő
- Taszk áteresztőképesség
- Processzor kihasználtság

Az architektúra leképezése a teljesítmény modellre

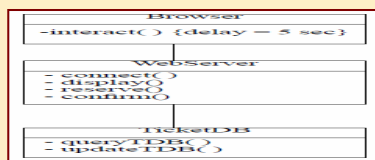


Osztályok és
lokális jellemzők

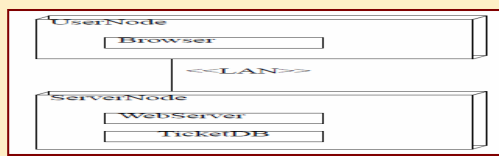
Processzorok
és telepítés

Interakciók
(hívások)

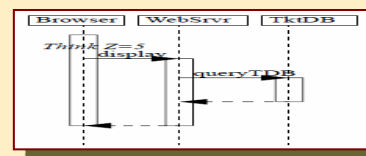
Az architektúra leképezése a teljesítmény modellre



Osztályok

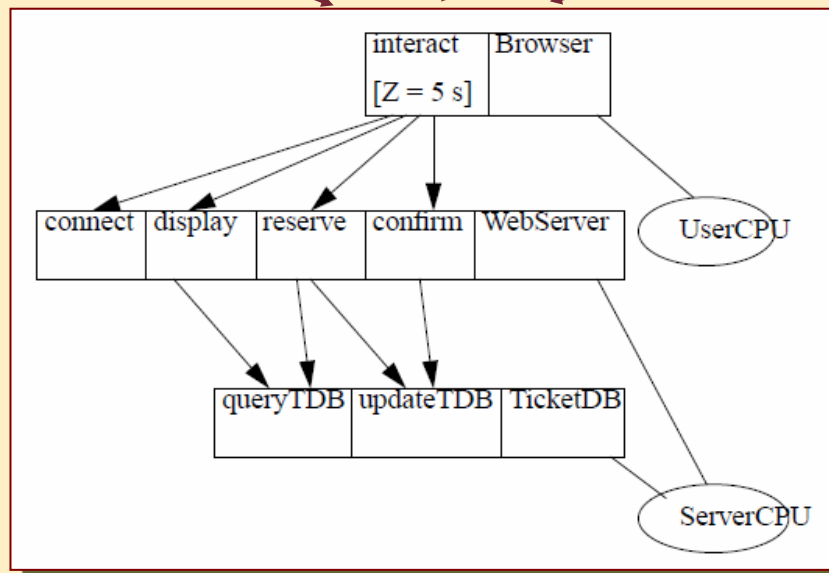


Telepítés



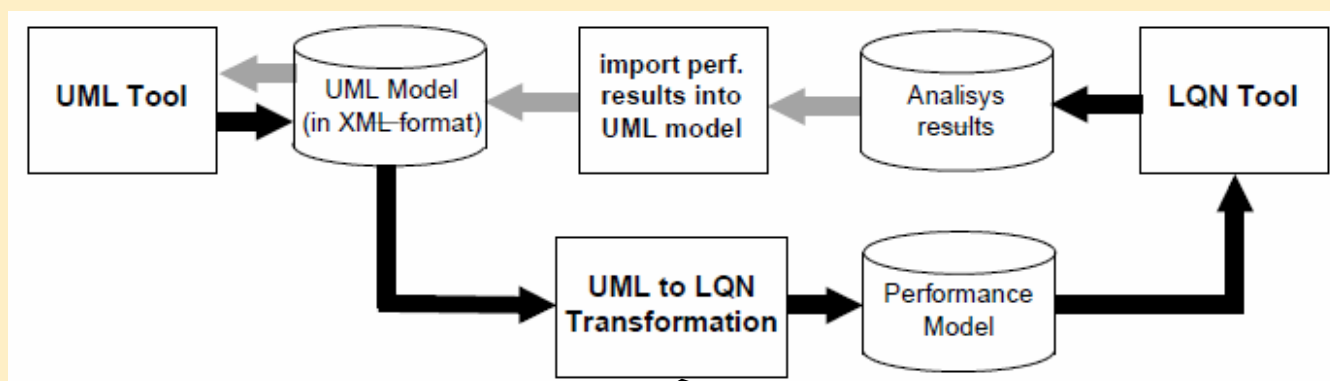
Interakciók

Modell-
transzformáció



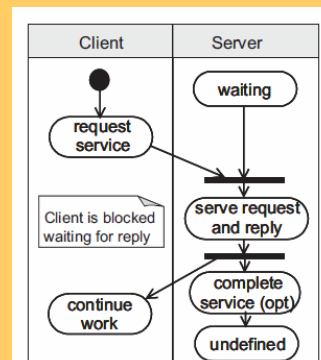
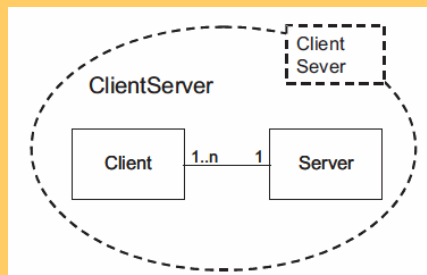
LQN
teljesítmény-
modell

Az architektúra leképezése a teljesítmény modellre



- Architektúra
minták
használata

Szinkron üzenetküldés:



Miről volt szó?

- Motiváció
 - Mit határoz meg az architektúra?
 - Milyen vizsgálati módszerek vannak?
- Követhetőség
- Szisztematikus vizsgálati módszerek
 - Interfész analízis
 - Hibahatás analízis
- Modell alapú vizsgálatok
 - Megbízhatósági modellezés
 - Teljesítmény modellezés