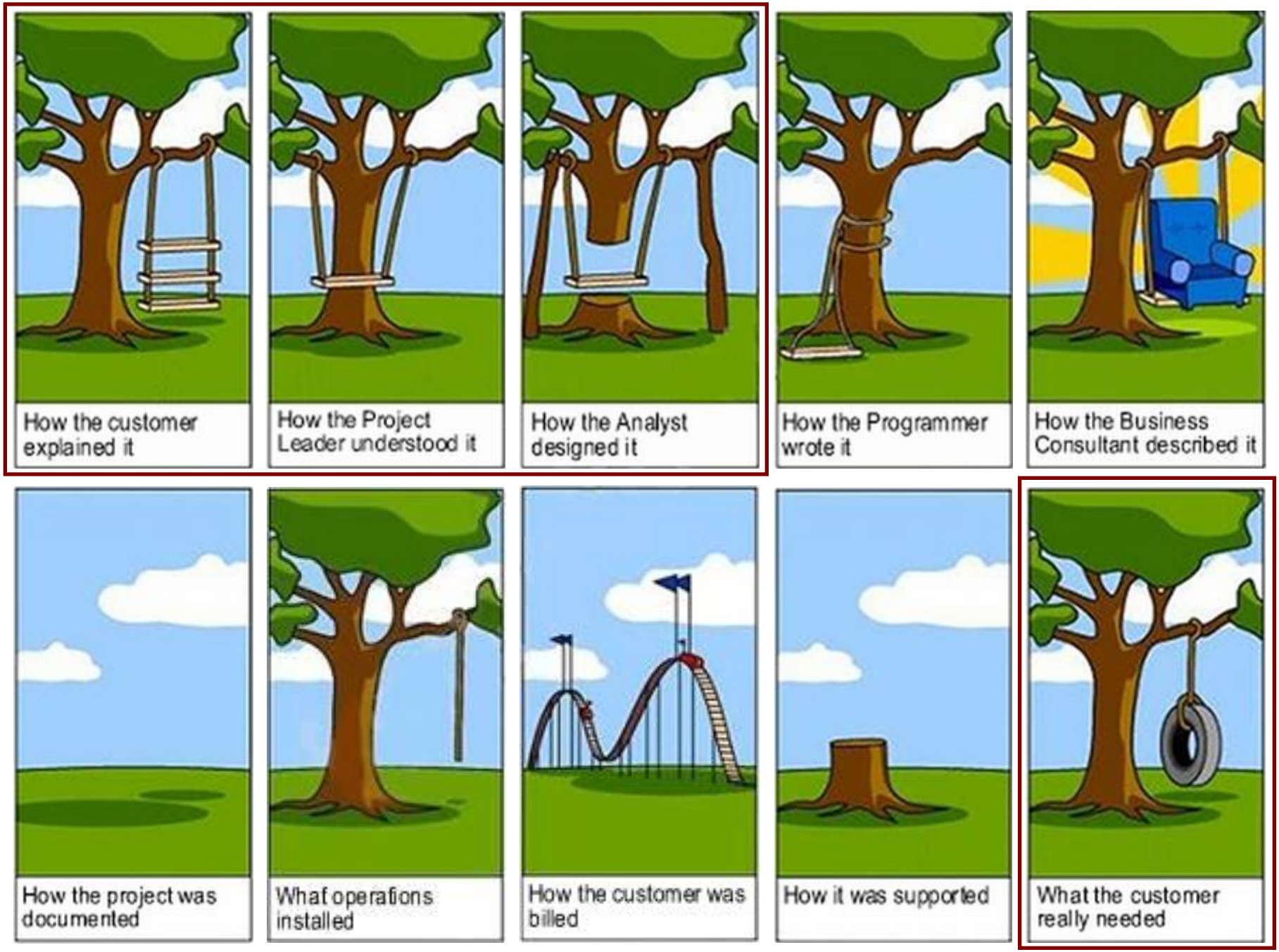
Követelménykezelés
A követelményspecifikáció ellenőrzése

Majzik István
Egyes ábrák: Pap Zsigmond, Polgár Balázs

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

Tartalomjegyzék


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Motiváció

- Tapasztalat: Sok hiba visszavezethető hiányos vagy ellentmondásos specifikációra
  - Példa: Meta Group felmérés, 2003:
    - Az IT projekt kudarcok 60%-70%-a a nem kielégítő követelményelemzésre vezethető vissza
  - Példa: 200 szoftverfejlesztési projekt utólagos felülvizsgálata

    „An analysis of defect densities found during software inspections” (Journal on Systems Software)
    - Gyakoribbak a hibák a specifikáció elkészítésének fázisában, mint a későbbi, implementációs fázisokban
  - Példa: Voyager és Galileo űrszondák szoftver tesztelése során felfedezett hibák okainak elemzése

    78% (149/192) specifikációs hiányosság, ebből
    - 23% veszélyes állapotban ragadás (nincs kilépés)
    - 16% időzítési kényszerek megadásának hiánya
    - 12% nincs specifikált reakció külső eseményre
    - 10% bemeneti érték ellenőrzésének hiánya

Követelmények és specifikáció

- Követelmény (requirement):
  - Bejövő igény, vízió, elvárás
    - Felhasználóktól (user)
    - Érdekeltektől (stakeholder: hatóság, vezetőség, operátor, ...)
  - Validáció alapja
- Specifikáció (requirement specification, system specification, software specification):
  - Tervezők, fejlesztők felé átalakított elvárások
  - Elemzés (absztrakció, strukturálás, szűrés) eredménye
  - Sokféle típus
    - Property specification, behaviour specification, …
    - Rendszerspecifikáció, biztonsági specifikáció, …
  - Verifikáció alapja

Elvárások a specifikációval szemben

- Az elvárások teljes lefedése
  - Fukcionális követelmények
  - Extra-funkcionális követelmények
- Megfogalmazás: Egyértelmű, igazolható, megvalósítható
- Lehetséges megoldások:
  - Szigorú specifikációs nyelv (pl. formális nyelvek)
  - Ellenőrzött (tervezési illetve specifikációs) minták használata
  - Utólagos ellenőrzés
- Példa: EN 50128 szabvány által adott lehetőségek
  - Formális módszerek (VDM, Z, B, TL, PN, ...)
  - Félformális módszerek (diagram alapú technikák, UML)
  - Strukturált metodika (JSD, SADT, SSADM)
  - Emellett természetes nyelvű megadás is szükséges

Tartalomjegyzék

A követelménykezelés feladatai (áttekintés)

- Követelmények hatékony, strukturált tárolása
  - Hierarchikus elrendezés, tulajdonságokkal
- Követelmény életciklus támogatása
  - Felvétel, törlés, változás, kapcsolatok megjelenése
- Követelmények finomítása és követhetősége
  - Követelményspecifikáció -> Rendszerterv -> Modulterv -> Forráskód -> Teszt -> Teszteredmény felé
- Analízis lehetőségek
  - Hatás analízis (impact analysis): változáskezelés
    - Mit befolyásol, ha a követelmény megváltozik?
  - Eredet analízis (derivation analysis): költség-haszon elemzés
    - Milyen követelményre vezethető vissza? Miért van itt, szükséges-e?
  - Fedettség analízis (coverage analysis): projekt követés
    - Mely követelmények nincsenek implementálva?
    - Mely követelmények nincsenek tesztelve?

A követelménykezelés „kézi” módszerei

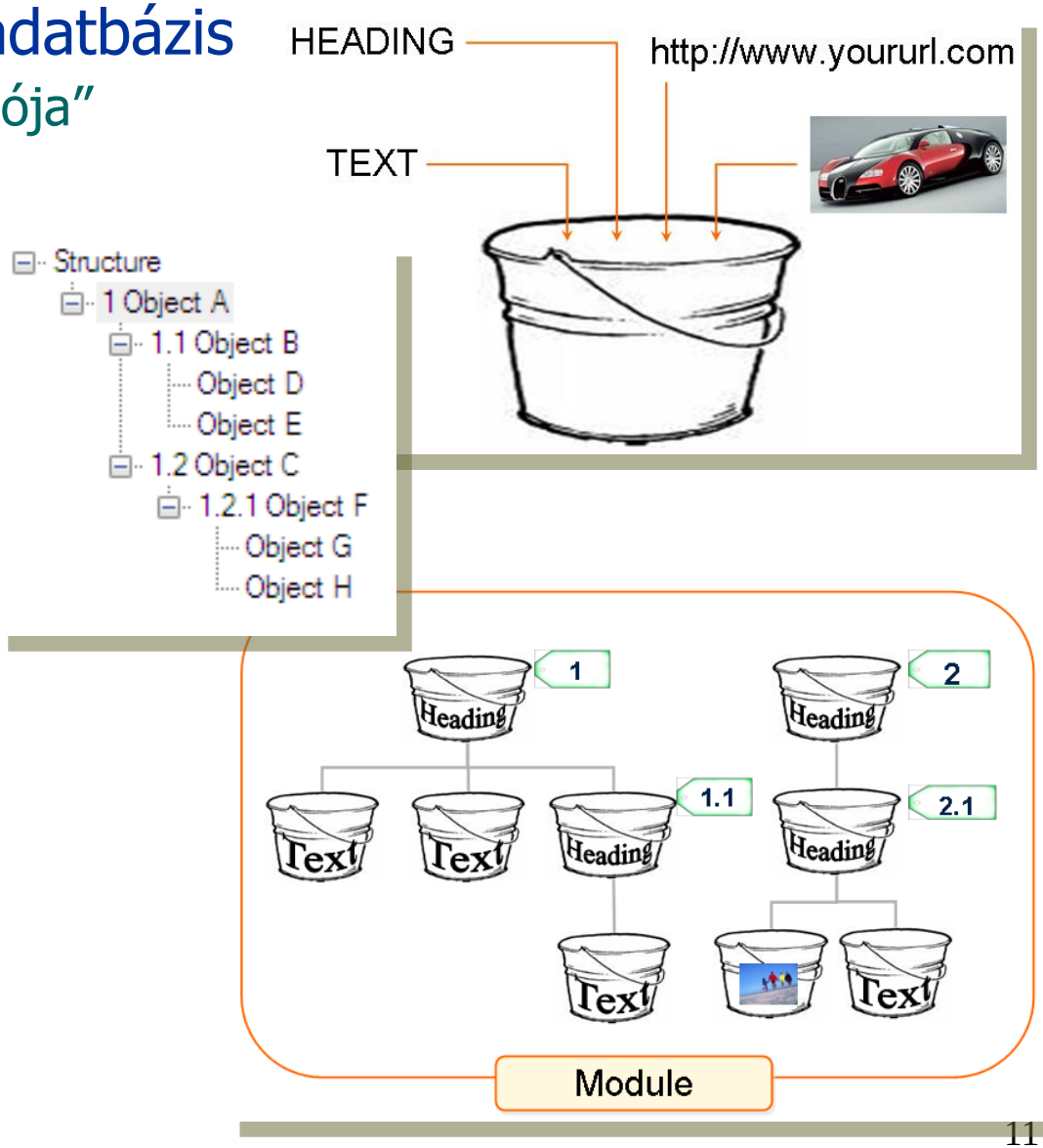
- Természetes nyelvű követelmény dokumentum
  - Strukturálás (fejezetek, alfejezetek)
  - Követelményazonosítók felvétele (pl. R2.3.1)
- Követelményfinomítás: Táblázatos nyilvántartás
  - Kiindulás (sorok): Követelményazonosítók
  - Kitöltés: Architektúra, modul, implementáció, teszt ID-k
    - Különböző dokumentumokból (SRS, SA, MDS, MTS, ...)
  - Követhetőségi mátrix: Analízis makrókkal támogatható
    - Üres, többszörös, ... mezők kikeresése
    - Nem implementált követelmények
    - Teszteletlen követelmények
  - Sikeres/sikertelen teszteredmény is bejelölhető

Automatikus követelménykezelők feladatai

Követelmények nyilvántartása:	Hierarchikus felépítés
Kapcsolatok nyilvántartása:	Sokféle reláció: Kompozíció, származtatás, finomítás, bizonyítás, .. Követelmény – Modell – Kód – Teszt – Teszteredmény között
Követelmény változások kezelése:	Időbeli struktúra, triggerek
Navigáció a kapcsolatokon:	Előre: pl. hatás analízishez Vissza: pl. eredet analízishez
Fedettségi listák készítése:	Lefedetlen követelményekhez Indokolatlan megvalósításhoz
Jogosultságok kezelése:	Hozzáférés szerepek
Értesítési rendszer:	Változásokról
Biztonsági megoldások:	Sértetlenség

Megvalósítás: Strukturált tárolás

- Általános objektum adatbázis
  - Követelmények „tárolója”
  - Egyedi azonosítók
  - Modulok
- Hierarchia
- Tulajdonságok
  - Fejléc / szöveg
  - Hozzáférési jogok
  - Történet
  - Attribútumok
    - Prioritás
    - Státusz
    - Költség
    - ...
  - Linkek



Példa: DOORS

ID	Last Modified By	Car user requirements	Priority	Percentage cost	Comments
TRN-CSR-1	Bill Young	**1 Introduction**	Mandatory	0.172835	
TRN-CSR-2	Bill Young	This module contains the user requirements for a new car to be commercially available by 1 August 2006.	Mandatory		e form te a text field.
TRN-CSR-3	Bill Young	**2 User types**	Desirable	1.370889	
TRN-CSR-4	Bill Young	**2.1 Nationalities**	Mandatory	0.642687	
TRN-CSR-5	Bill Young	The car will be used in the following countries: UK, USA, Northern Europe, Eastern Europe, Japan, Russia, Australia.	Mandatory	0.769025	
TRN-CSR-6	Bill Young	**2.2 User sizes**	Manda		
TRN-CSR-7	Bill Young	People come in all shapes and s es. The car must be suitable for people maxi um and minimum sizes fgfgf to 2 m ighing 25 kilograms t	Manda		

Tulajdonságok

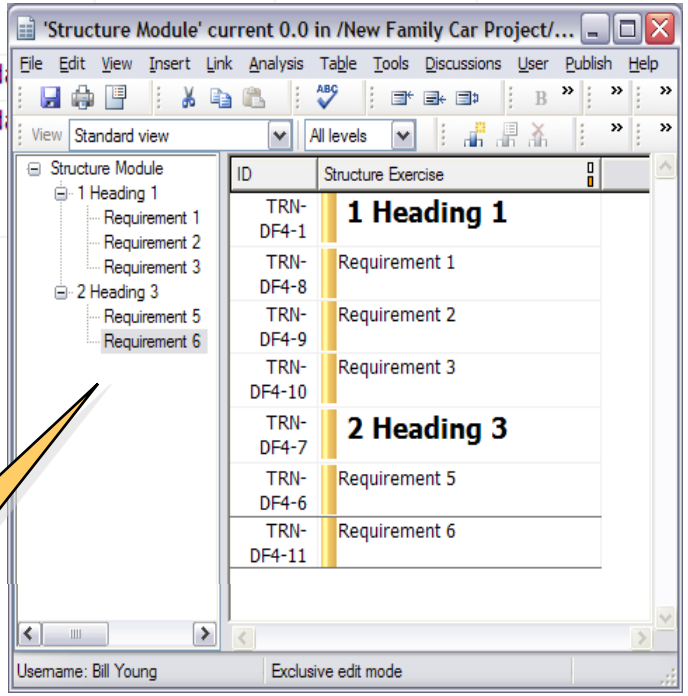
Objektum azonosító

Változás-jelző

Fejléc objektum

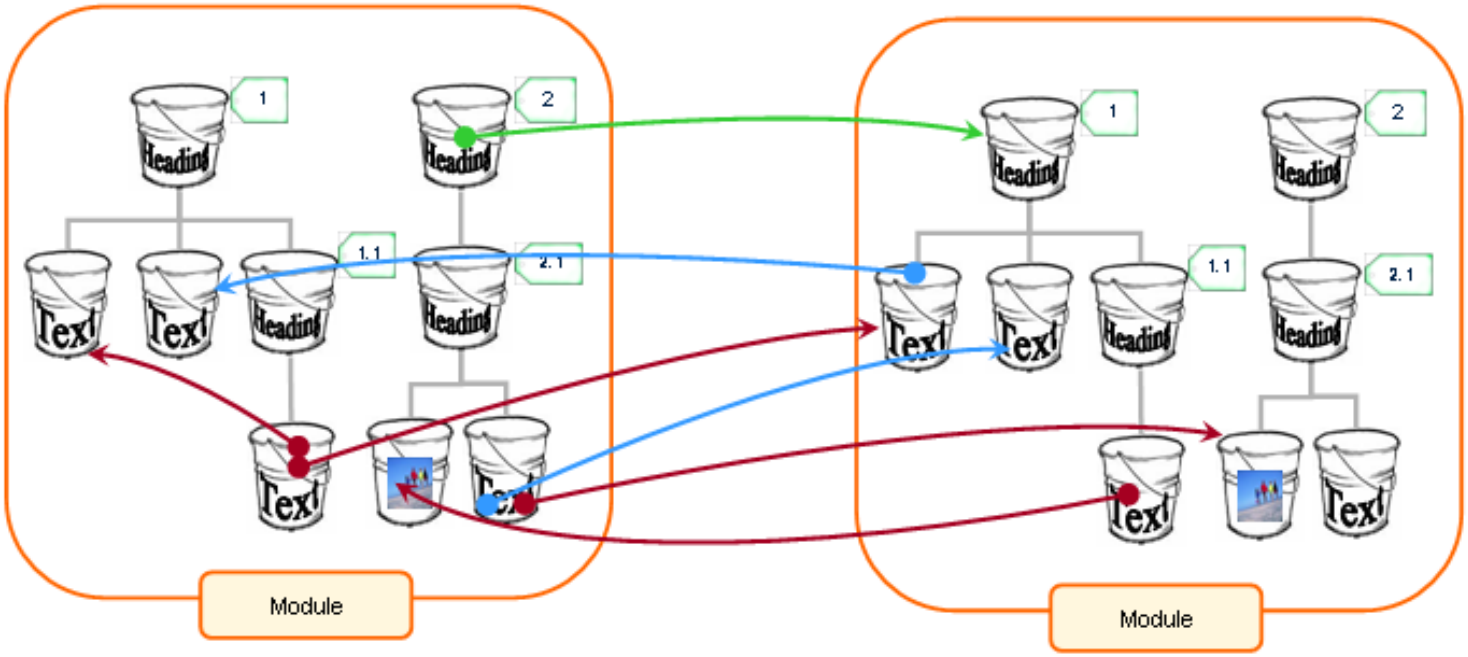
Szöveges objektum

Hierarchia

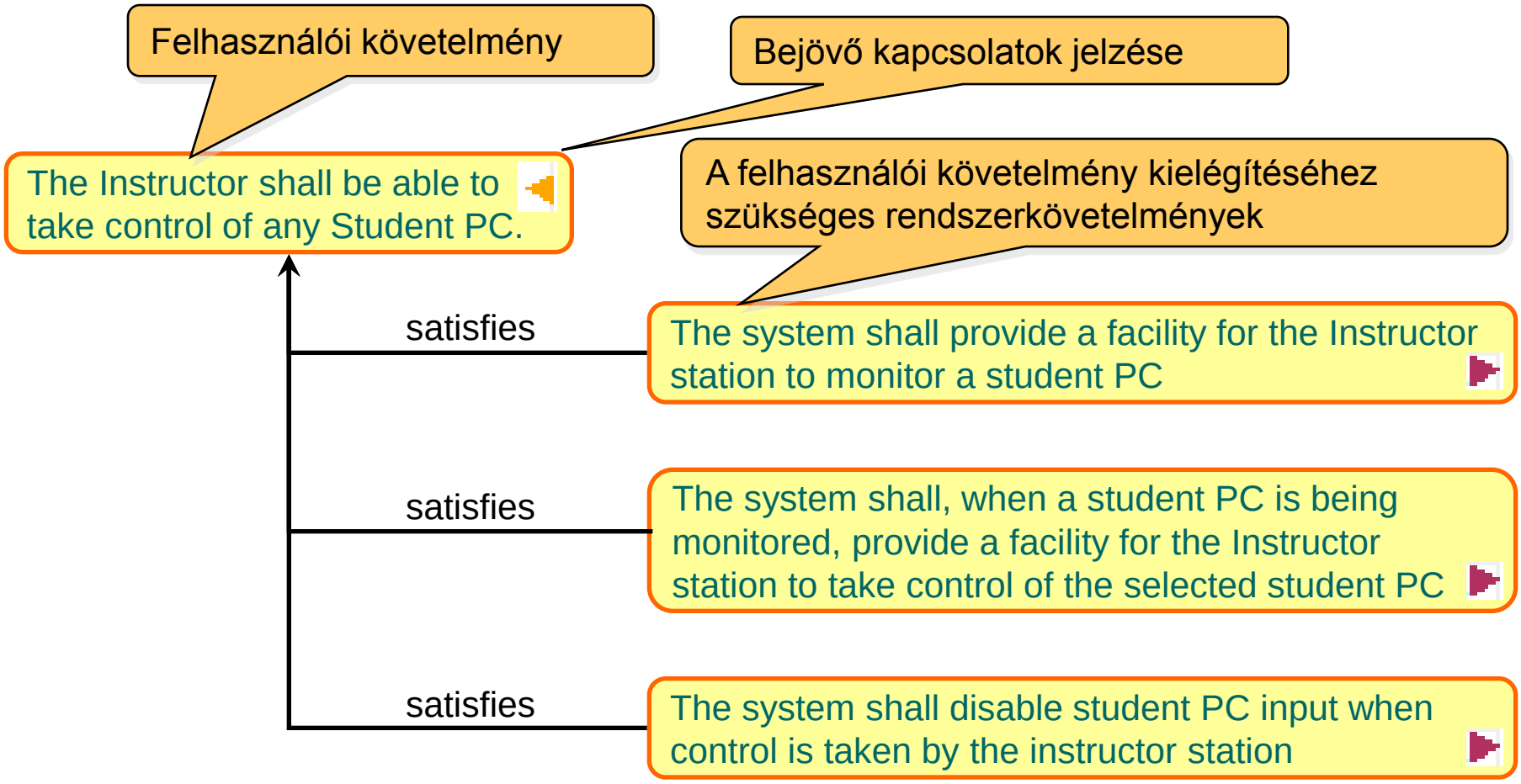


Megvalósítás: Kapcsolatok (linkek)

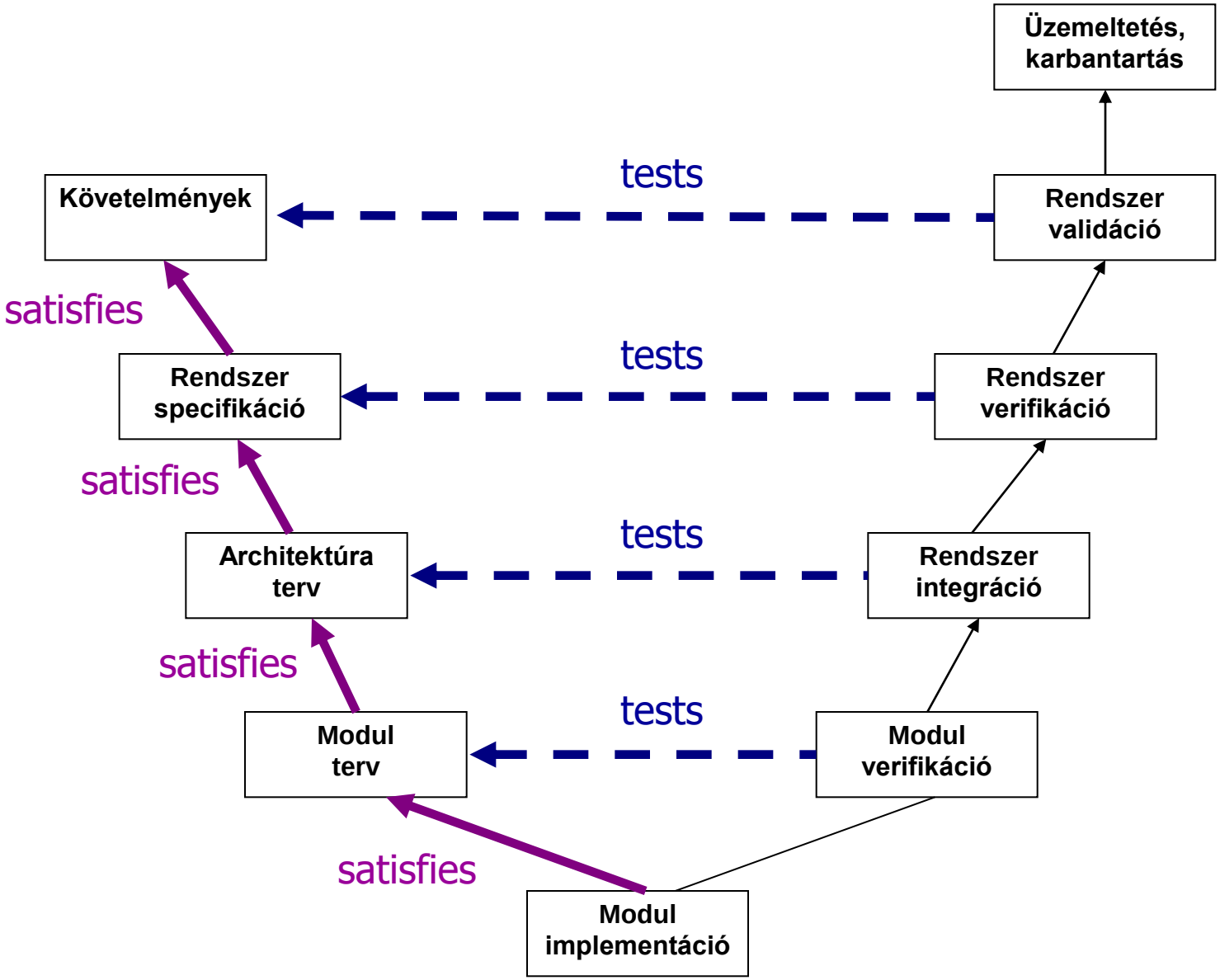
- Relációk: Rendezett párok
  - Objektumok között
  - Külső kapcsolatok
- Típusok
  - Finomítja
  - Kielégíti
  - Teszteli
  - ...



Példa: DOORS


Felhasználói követelmény
Bejövő kapcsolatok jelzése
The Instructor shall be able to take control of any Student PC.
A felhasználói követelmény kielégítéséhez szükséges rendszerkövetelmények
satisfies
The system shall provide a facility for the Instructor station to monitor a student PC
satisfies
The system shall, when a student PC is being monitored, provide a facility for the Instructor station to take control of the selected student PC
satisfies
The system shall disable student PC input when control is taken by the instructor station

Tipikus relációk a V-modellben



Megvalósítás: Követelmény életciklus

- Követelmény szintű változások
  - Közvetlen szerkesztés (létrehozás, módosítás, törlés)
  - Megváltozott objektum kapcsolatok jelzése (suspect link)
  - Változtatási javaslatok menedzselése
    - Elkészítés, csoportosítás, felülvizsgálat, érvényre juttatás
- Változások mint események
  - Scriptek indítására használhatók
- Nagyléptékű változások
  - Baseline definiálás (állapot rögzítése)
    - Inkrementális változások vizsgálhatók
    - Összehasonlítás lehetséges
  - Követelmény-partíciók kiajánlása, távoli szerkesztés, szinkronizálás, visszavétel

Megvalósítás: Analízisek

- Követhetőség alapja:
  Navigálás a kapcsolatokon keresztül
  - Kiterjedés (scope) és mélység kijelölhető
  - Irány kijelölhető (előre, hátra)
- Script nyelv használható
  - Bejárás, kigyűjtés
  - Tulajdonságok megváltoztatása
- Jelentések készítése
  - Hatás analízis: Előre navigálás alapján
  - Eredet analízis: Hátra navigálás alapján
  - Fedettség analízis
    - Pl.: Objektumok kigyűjtése: Nincs kapcsolat adott célhalmazig (nincs megvalósítás, nincs sikeres teszt)

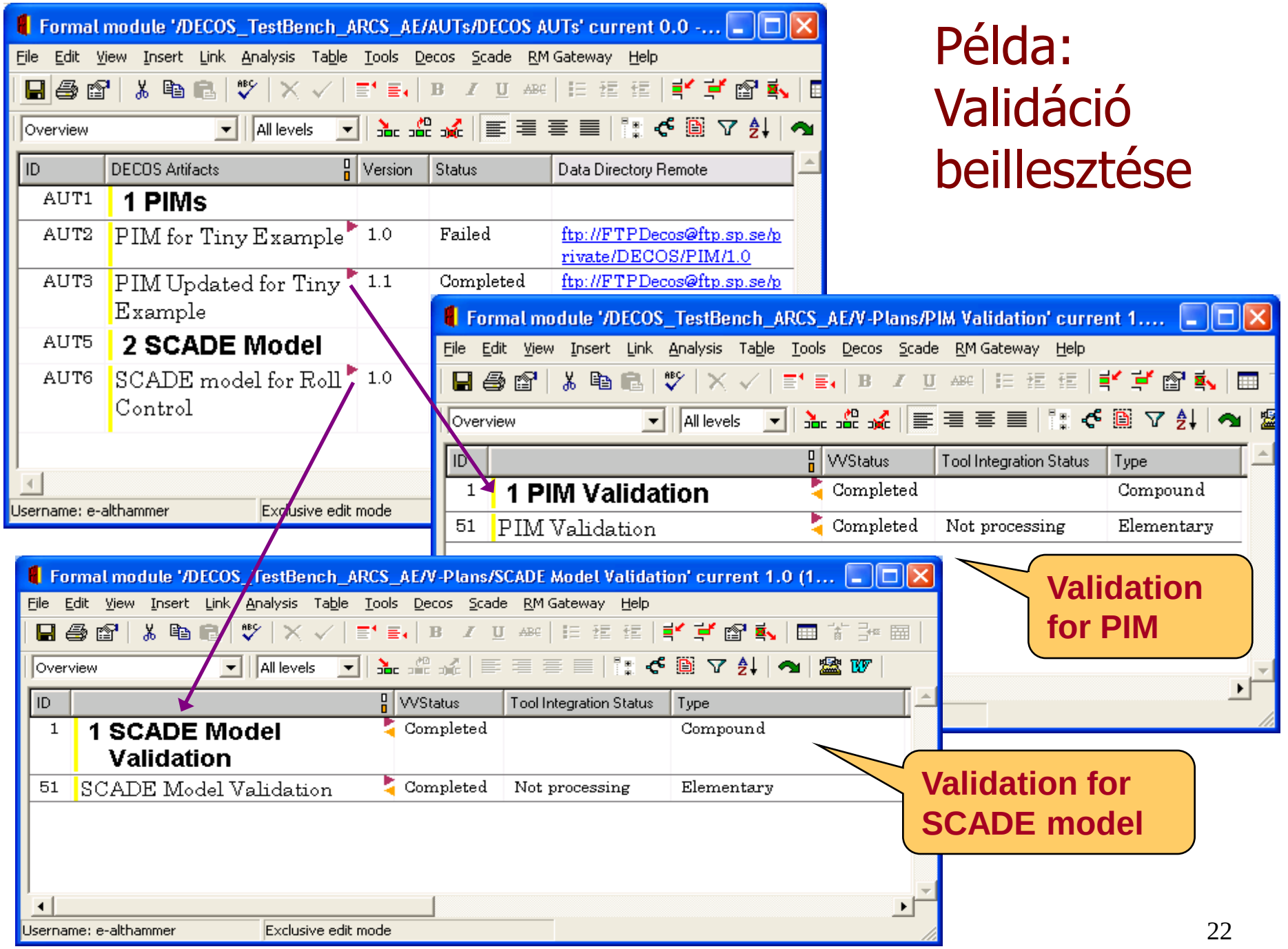
Megvalósítás: Járulékos funkciók

- Nézetek az objektum (követelmény) listában
  - Szűrés hierarchiaszintekre, tulajdonságokra, …
- Űrlapok a tulajdonságok szerkesztésére
- Megosztott szerkesztés (csoportmunka)
  - Objektum (követelmény) szintű zárolás
- Dokumentáció generálás
  - Hierarchia (fejezetek) alapján rendezett szöveges és egyéb objektumok
  - Exportálás: Strukturált külső dokumentumban (pl. Word) történt változtatások vissza is olvashatók (struktúra megőrzése)
  - Importálás: Előkészítés szükséges
- Webes hivatkozások (pl. e-mailben küldhető)
  - Adatbázis, projekt, objektum hivatkozás

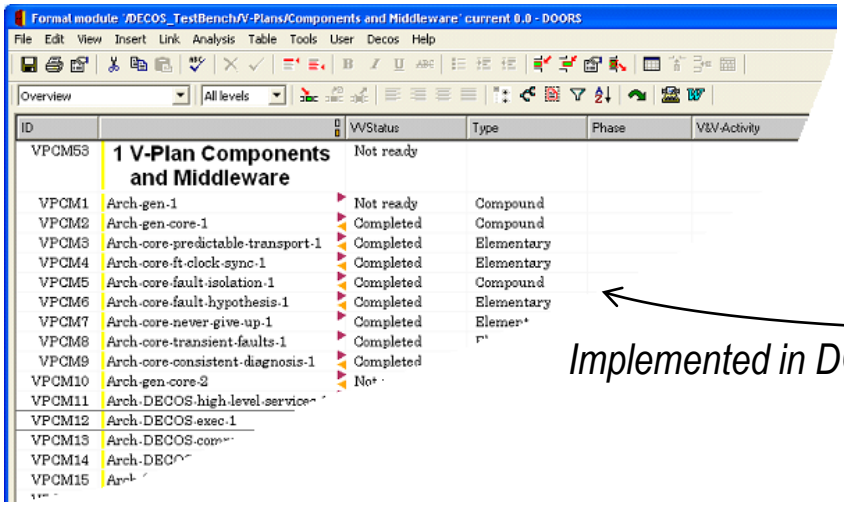
Követelmény alapú verifikáció

- Verifikációs tevékenység felvétele a követelmények mellé
  - Tervezett és rögzített verifikáció (pl. szabvány előírások)
  - Többféle tevékenység kombinálható
- Verifikációs eszközláncok kialakítása
  - Analízis: analízis modell generálása, analízis végrehajtása, eredmények visszacsatolása
  - Tesztelés: követelmény alapú tesztgenerálás, teszt végrehajtás, teszt eredmény kiértékelés
  - Mérések: mérés konfigurálás, végrehajtás, eredmények értékelése
- Verifikációs eszközláncok indítása a követelménykezelőből
  - Triggerek alapján; ezek script nyelven programozhatók
- Verifikáció státuszának rögzítése
  - Pl. ellenőrzött modell, sikeresen tesztelt követelmény

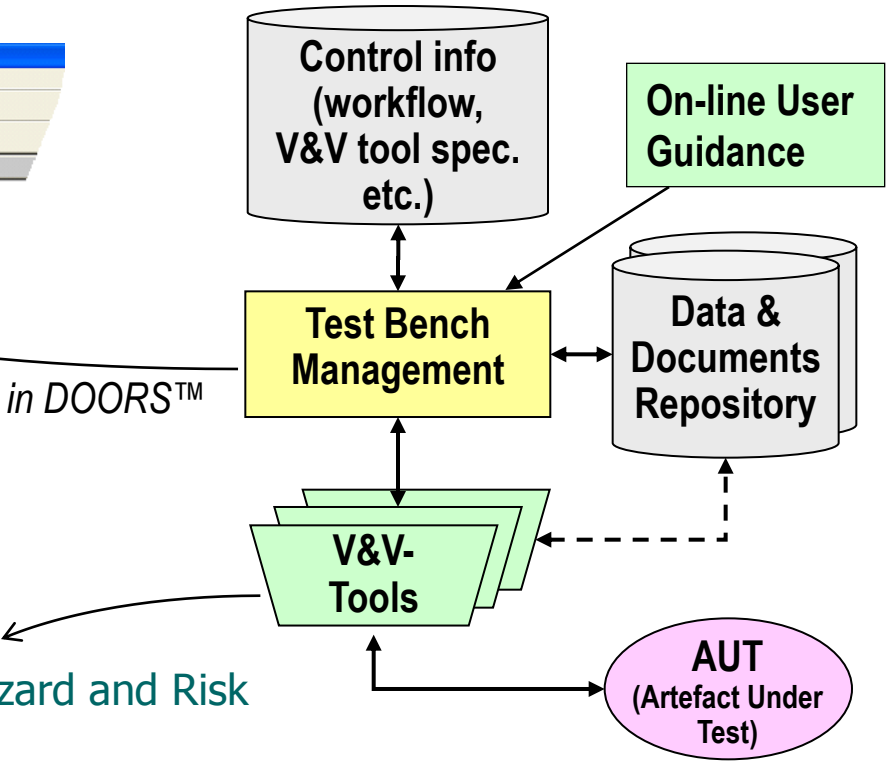
Példa: Validáció beillesztése



Példa: Verifikáció futtatása a követelménykezelőből



Implemented in DOORS™



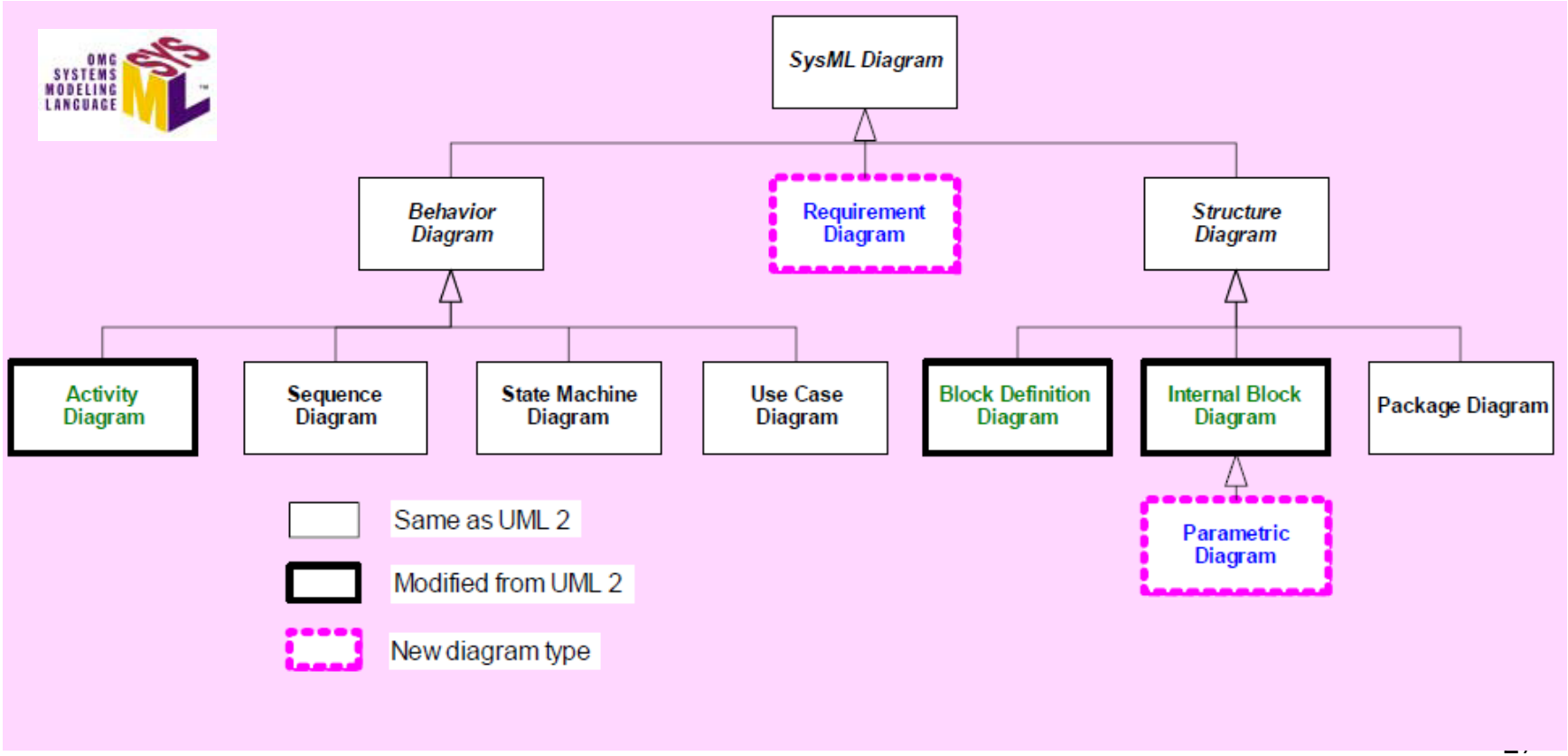
- ITEM (Hazard and Risk Analysis)
- RACER (Formal Verification)
- SCADE MTC (Simulation)
- LDRA (Testing)
- PROPANE (Fault Injection)
- EMI Test Bench

- Models
- Middleware
- Applications

Tartalomjegyzék

Félformális követelményspecifikáció: SysML

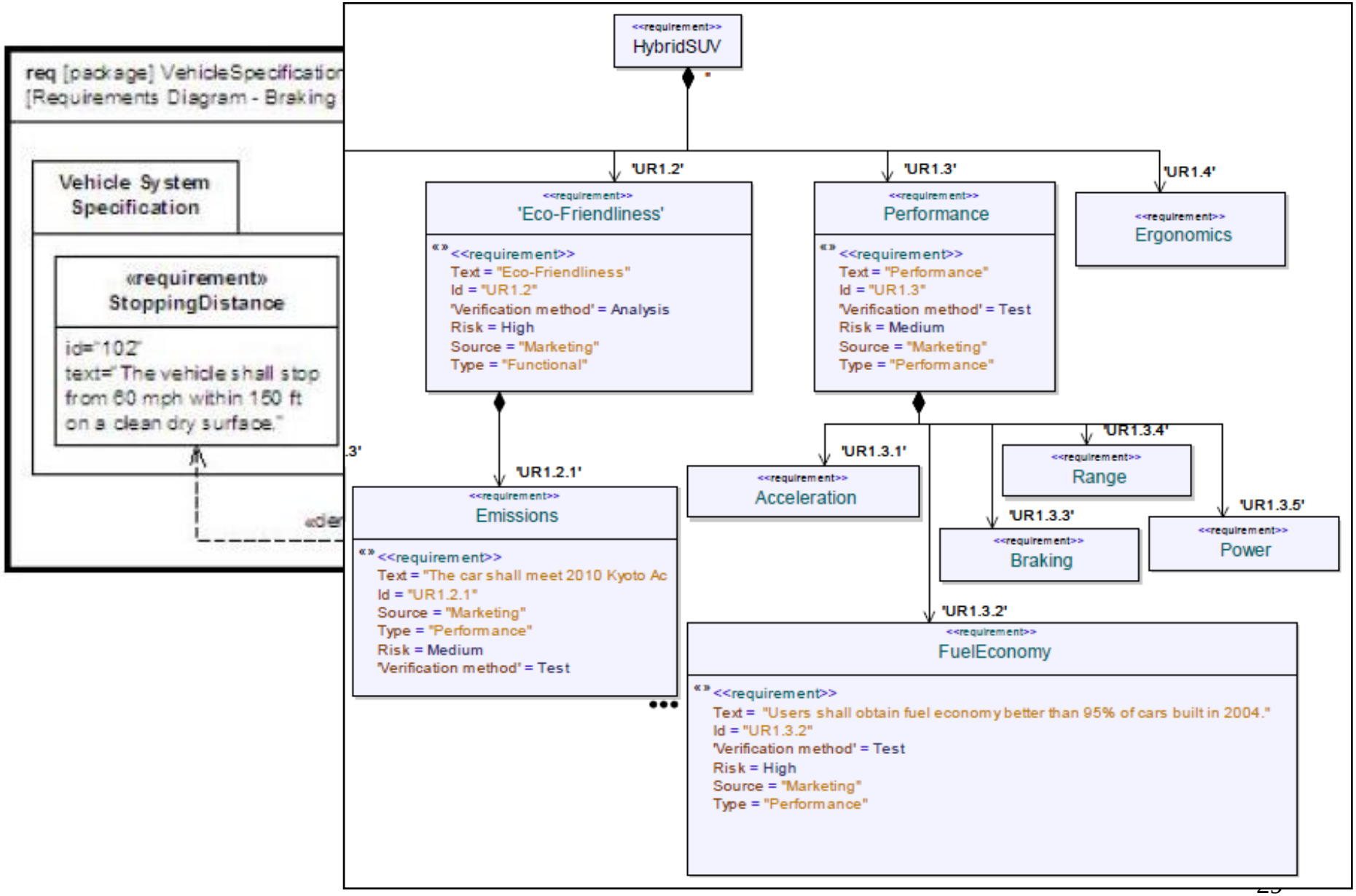
- Systems Modeling Language
  - UML részhalmaz egy kiterjesztése rendszertervezéshez
  - Fő újdonságok: Requirement és Parametric diagram



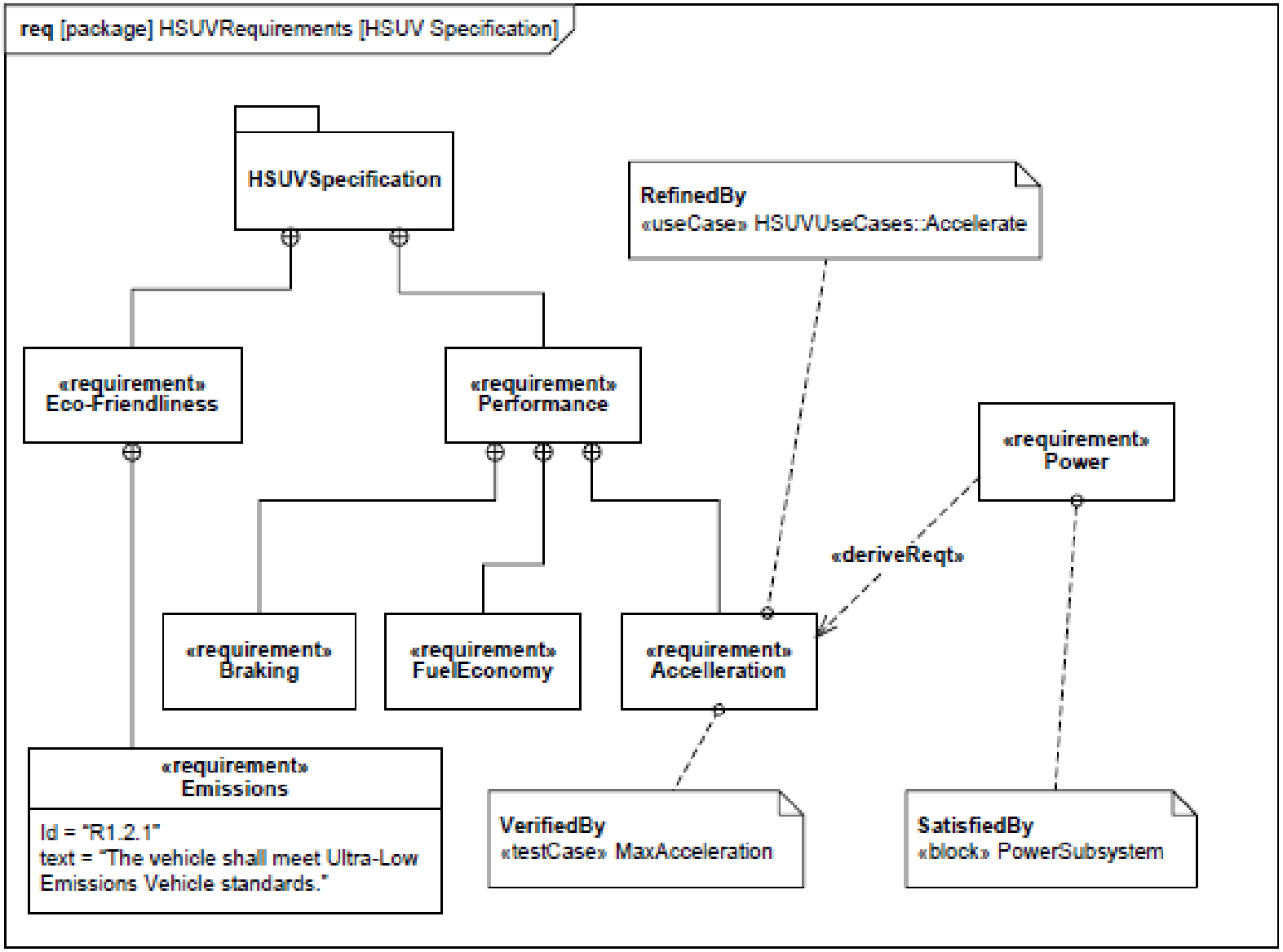
Requirement diagram

- Követelmények (szöveges is) tárolása azonosítóval
  - <<requirement>> stereotype
  - Id és text mezők
  - Felhasználói attribútumok: pl. type, source, risk, ...
  - Táblázatos forma is támogatott
- Követelmények hierarchikus csomagokba rendezhetők
  - Funkcionális, teljesítmény, ... kategóriák
- Követelmények közötti finomítás (~ subclass), kompozíció
- Relációk használhatók (callout: megjegyzésekben):
  - Copy: követelmények között (master – slave)
  - Trace: követelmények között (client – supplier)
  - DeriveReqt: követelmények között (forrás – származtatott)
  - Refine: követelmények között (finomítás)
  - Satisfy: követelmények és terv/implementáció elemek között
  - Verify: követelmények és teszt elemek között

Requirement diagram példa: Struktúra

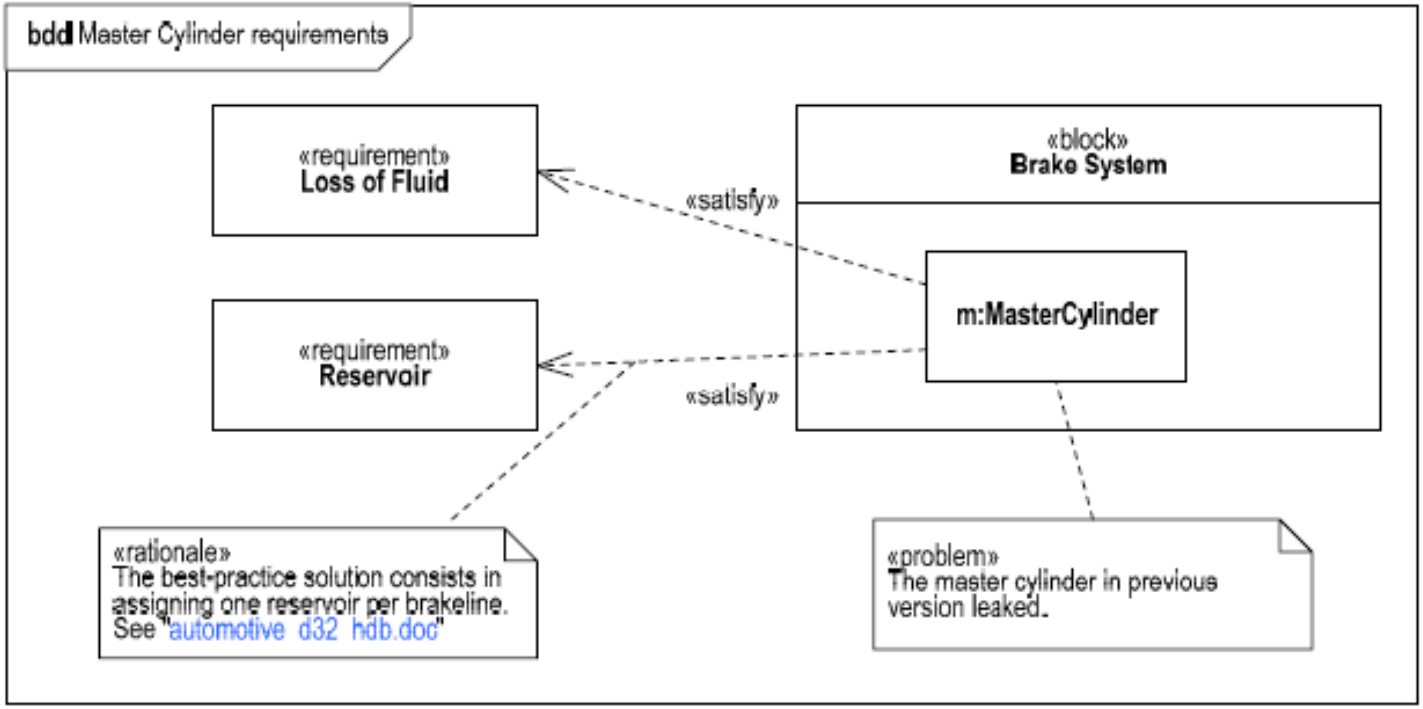


Requirement diagram példa: Relációk



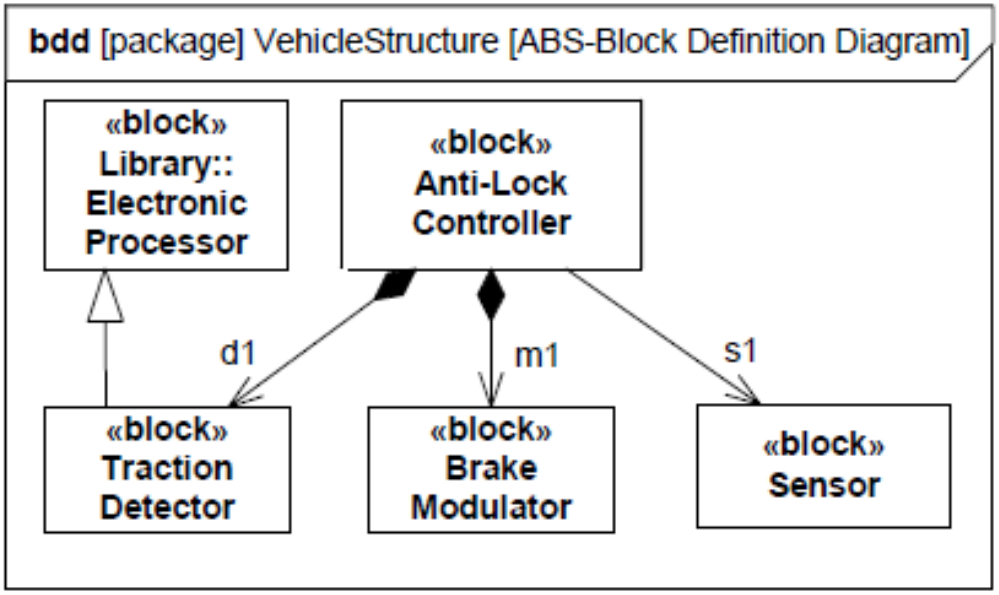
Requirement diagram példa: Tervezői döntések

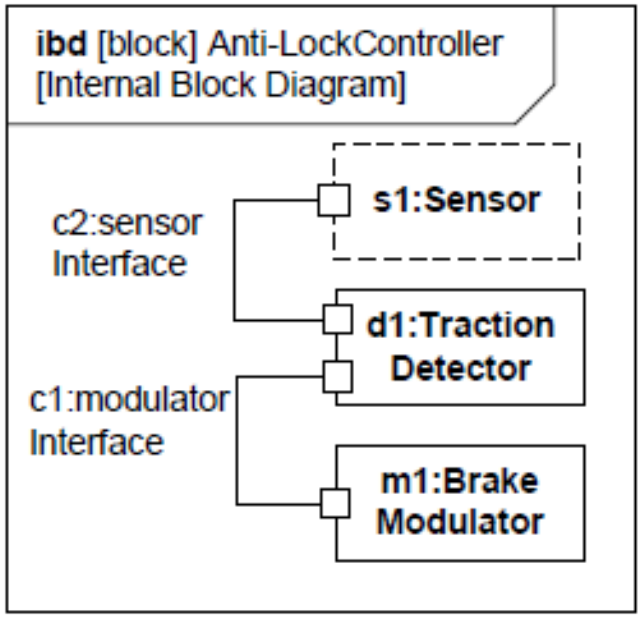
- Tetszőleges modell elemhez köthető megjegyzések (előredefiniált stereotype):
  - <<problem>>: Probléma, döntést igénylő felvetés
  - <<rationale>>: Megoldás, magyarázat



Block diagram

- Block Definition Diagram:
  - Block: A struktúra eleme (fekete / üveg doboz)
    - Komponens (nem csak szoftver)
    - A SysML-ben az UML 2.0 osztályokon alapul
- Internal Block Diagram:
  - Konkrét szerepek; típust a Block adja meg

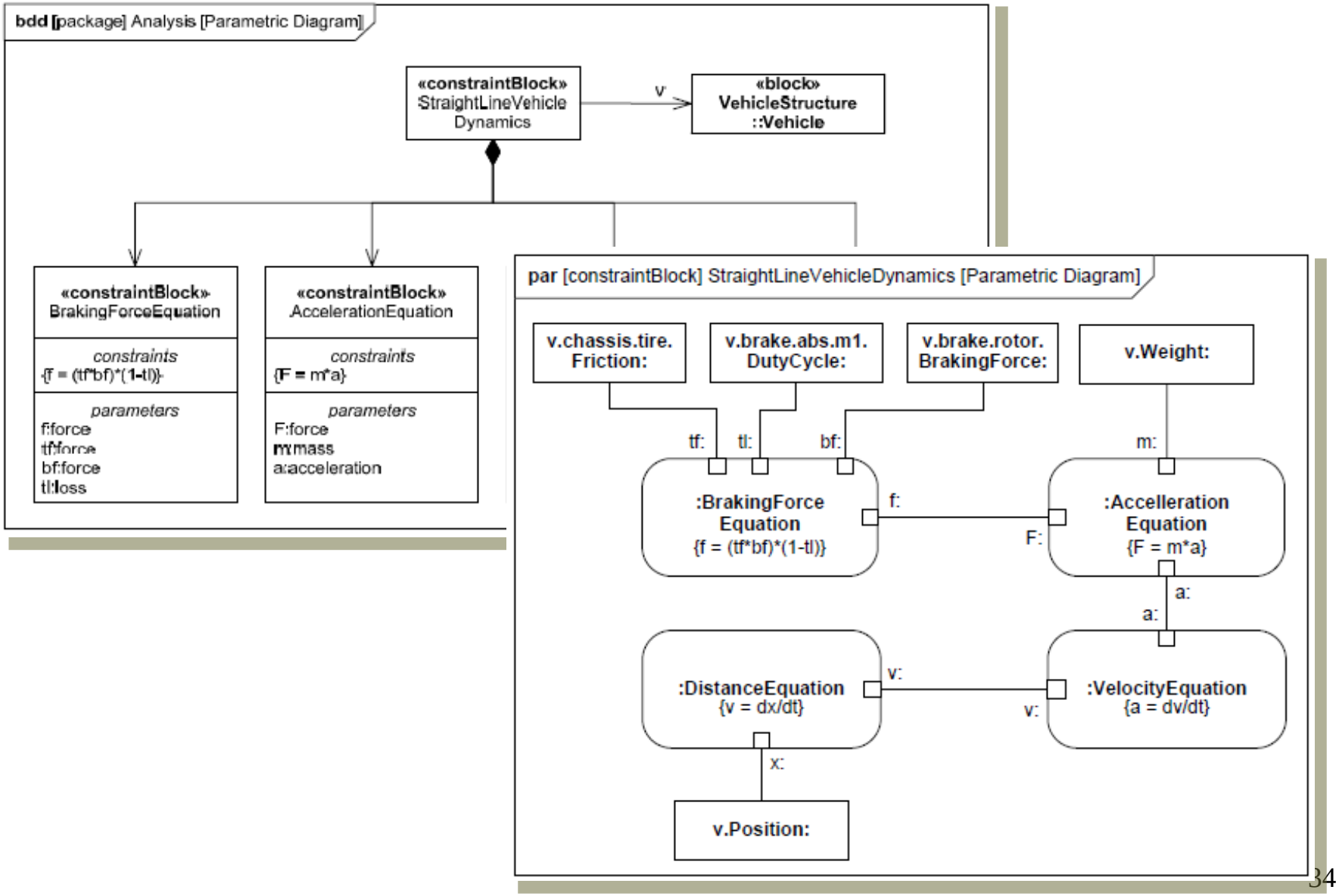




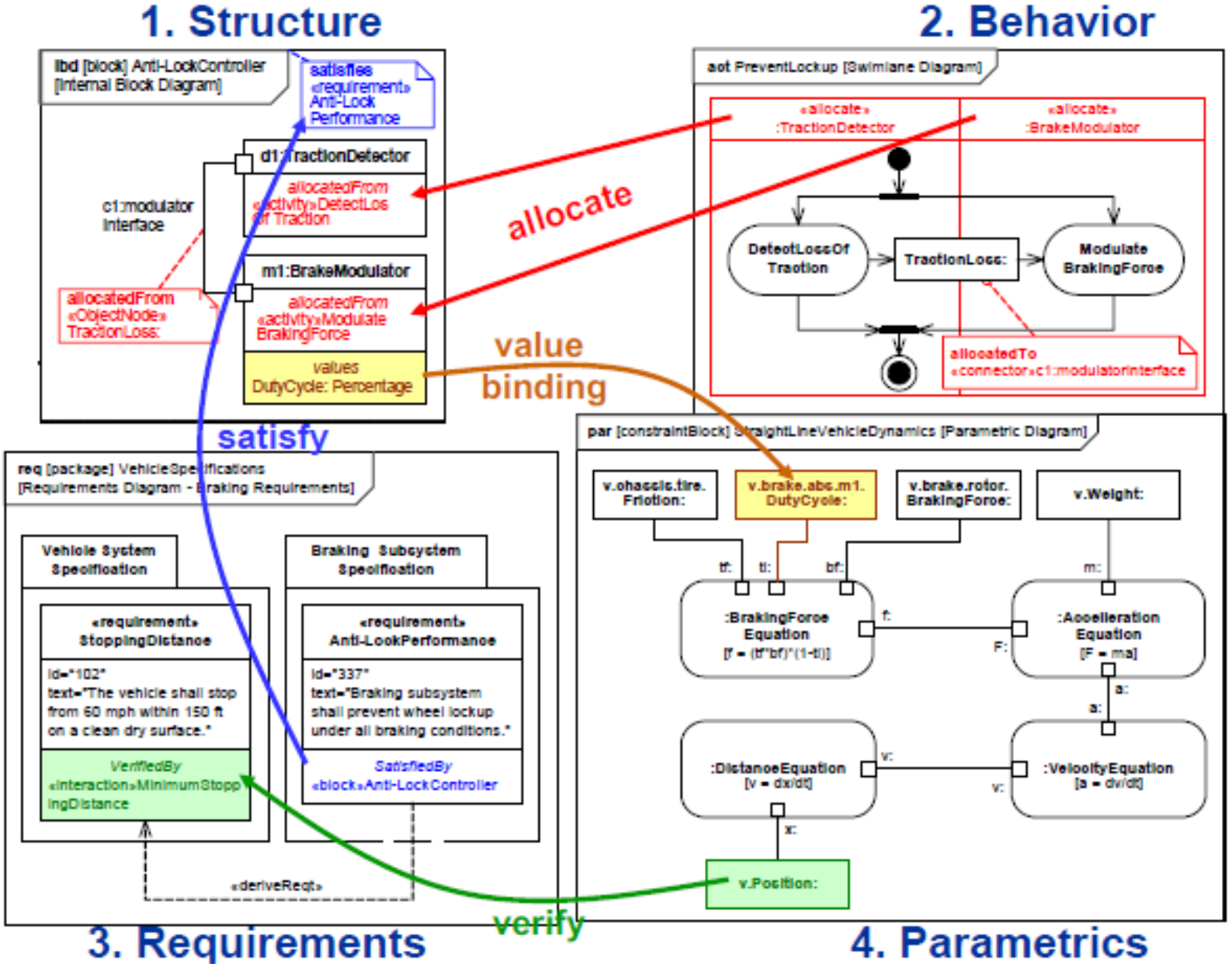
Parametric diagram

- Cél: Ellenőrizhető számszerű követelmények (kényszerek) megfogalmazása tulajdonságokra
  - Nem-funkcionális követelmények aspektusa
  - Analízis (pl. teljesítmény, megbízhatóság) támogatása
- ConstraintBlock: Összefüggések megadása
  - Formális (pl. MathML, OCL), vagy informális alakban
  - Analízis eszközhöz igazítható (nem SysML specifikus)
- Parametric diagram: Összefüggések alkalmazása
  - Az összefüggések (Constraint block) alkalmazása egy adott környezetben
  - Kötések értékek között

Parametric diagram példa



Relációk diagramok között: Követhetőség



Tartalomjegyzék

- Motiváció
  - Miért fontosak a tervezési folyamat ezen szakaszai?
  - Milyen elvárások vannak a specifikációval szemben?
  - Milyen módszerei vannak a specifikáció készítésnek?
- A követelménykezelés általános feladatai
  - Követelmények nyilvántartása
  - Követhetőség a verifikációhoz
- Félformális specifikáció
  - Specifikus technika: SysML
- A követelményspecifikáció verifikációja
  - Általános kritériumok
  - Példa: Specifikus kritériumok UML állapottérképekre

Tipikus vizsgálati szempontok

- Teljesség
  - Funkciók, hivatkozások, eszközök
- Konzisztencia (ellentmondásmentesség)
  - Külső és belső
  - Követhetőség
- Megvalósíthatóság
  - Erőforrások
  - Használhatóság
  - Karbantarthatóság
  - Kockázatok: költségbeli, technikai, környezeti
- Ellenőrizhetőség (tesztelhetőség)
  - Specifikus
  - Egyértelmű
  - Számszerűsíthető

Vizsgálati szempontok IEEE Std 830-1998 alapján

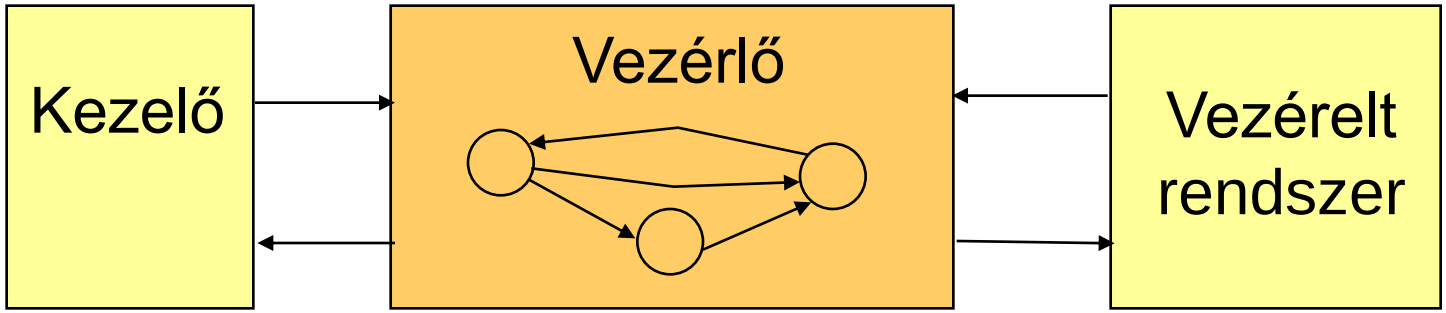
- Helyes
  - A szoftverre vonatkozó követelményeknek (elvárásoknak) megfelelő
  - Konzisztens a külső forrásokkal (pl. szabványok)
- Egyértelmű
  - Nem félreérthető, egy jelentése van
  - Hasznosak a formális, félformális specifikációs nyelvek
- Teljes
  - Minden (érvényes, érvénytelen) bemenetre van specifikált viselkedés
  - TBD csak indoklással és a feloldás módjával
- Konzisztens
  - Nincs belső ellentmondás, egységes a terminológia
- Fontosság és stabilitás szempontjából rendezett
  - Követelmények szükségessége, változatlansága felmérve
- Ellenőrizhető
  - Megállapítható egyértelműen, ha nem teljesül egy követelmény
- Módosítható
  - Nem redundáns, jól strukturált, jól elválasztott követelmények
- Követhető
  - Eredet becsatolható, további hatások hivatkozhatók

Vizsgálati szempontok IEEE Std 29148-2011 alapján

- Szükséges
  - Kihagyása hiányosságot jelent, más követelmények nem pótolják, nem fedik le.
- Megvalósítástól független
  - Nem jelent szükségtelen megkötéseket (csak azt írja le, hogy mi az elvárás).
- Egyértelmű
  - Csak egyféleképpen értelmezhető, könnyen érthető.
- Konzisztens
  - Nincs konfliktusban más követelményekkel.
- Teljes
  - Nincs szükség további kiegészítésekre az érintettek igényeinek megértéséhez.
- Egyedi
  - Csak egy követelményt ír le (nincs benne konjunkció).
- Megvalósítható
  - Illeszkedik a meglévő technikai, költségbeli, ütemezésbeli, szabályzási keretek és kényszerek közé.
- Követhető
  - Visszafelé a felhasználói igényekhez, előrefelé a rendszer további elemeihez (terv, implementáció, teszt stb.) köthető.
- Ellenőrizhető
  - Megállapítható, hogy a rendszer teljesíti-e a követelményt (erről bizonyosság szerezhető, pl. teszteléssel, elemzéssel, mérésekkel).

Példa: Vizsgálati szempontok reaktív rendszerekre (Leveson)

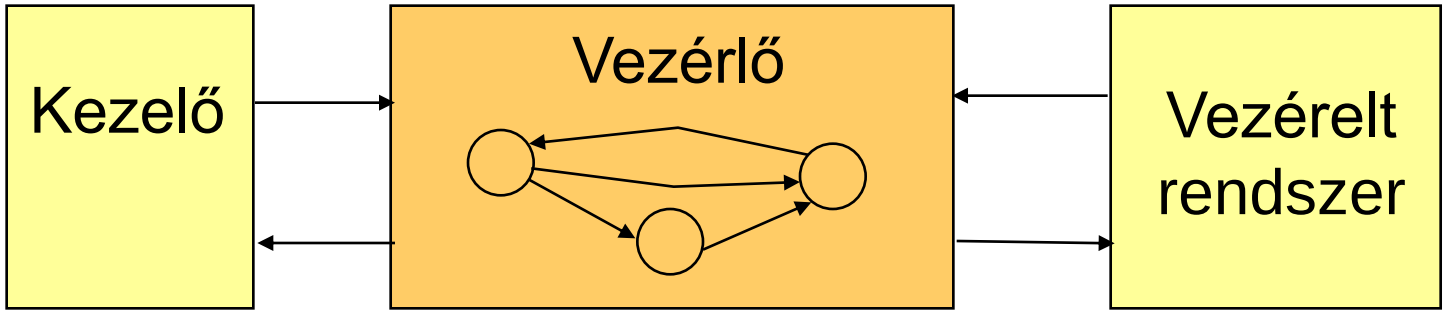
- Állapotdefiníció
- Bemenetek (események)
- Kimenetek
- Kimenetek és trigger kapcsolata
- Állapotátmenetek
- Ember-gép interfész



Példa: Vizsgálati szempontok reaktív rendszerekre

- Állapotdefiníció
- Beme
- Kime
- Kime
- Állap
- Ember-gép interfész

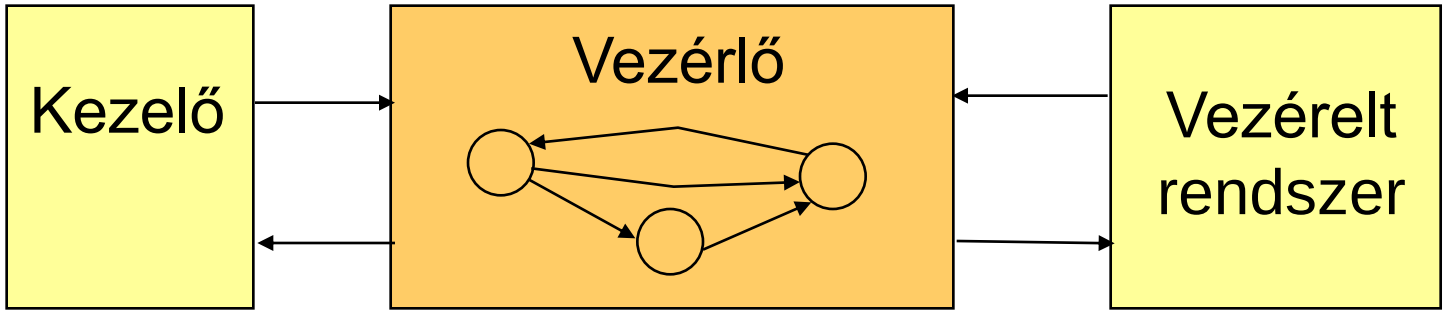
- Biztonságos a kezdőállapot
- Belső modell aktualizálva van a környezettel
(kimaradó bemeneti események esetén time-out van, és nincs a kimeneten akció)



Példa: Vizsgálati szempontok reaktív rendszerekre

- Állapotdefiníció
- Bemenetek (események)
- Kimenet
- Kimene
- Állapotá
- Ember-g

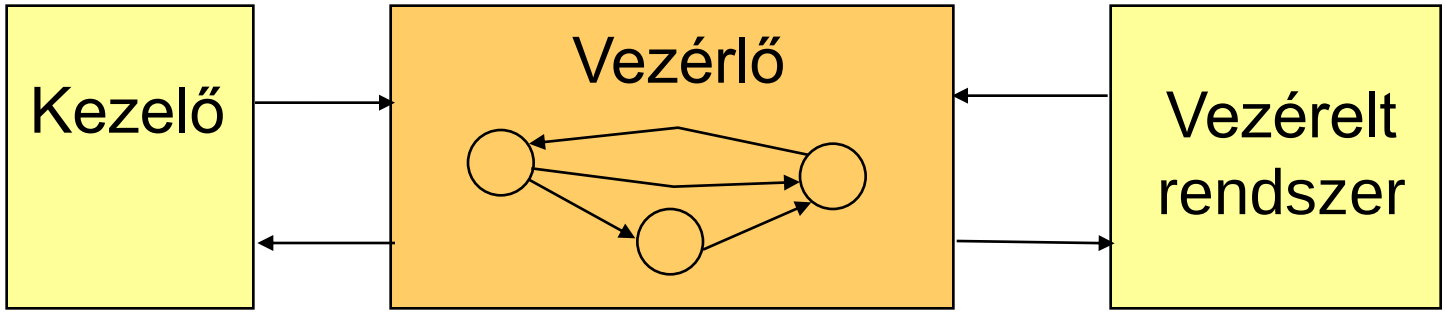
- Minden bemenetre, minden állapotban van specifikált viselkedés (reakció)
- Egyértelműek (determinisztikusak) a reakciók
- Van bemeneti ellenőrzés (értékbeli, időbeli)
- Hibás bemenet kezelése specifikálva van
- Megszakítások gyakorisága korlátozva van



Példa: Vizsgálati szempontok reaktív rendszerekre

- Állapotdefiníció
- Bemenetek (események)
- Kimenetek
- Kimen
- Állap
- Embe

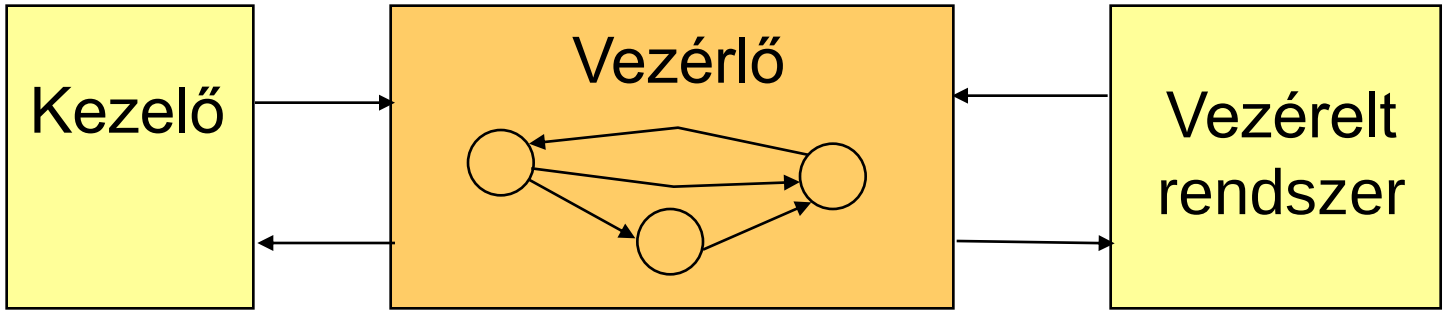
- Hihetőségvizsgálat kritériumai specifikáltak
- Nincsenek fel nem használt kimenetek
- Környezeti feldolgozóképesség be van tartva



Példa: Vizsgálati szempontok reaktív rendszerekre

- Állapotde
- Bemenet
- Kimenetek
- Kimenetek és trigger kapcsolata
- Állapotátmenetek
- Ember-gép interfész

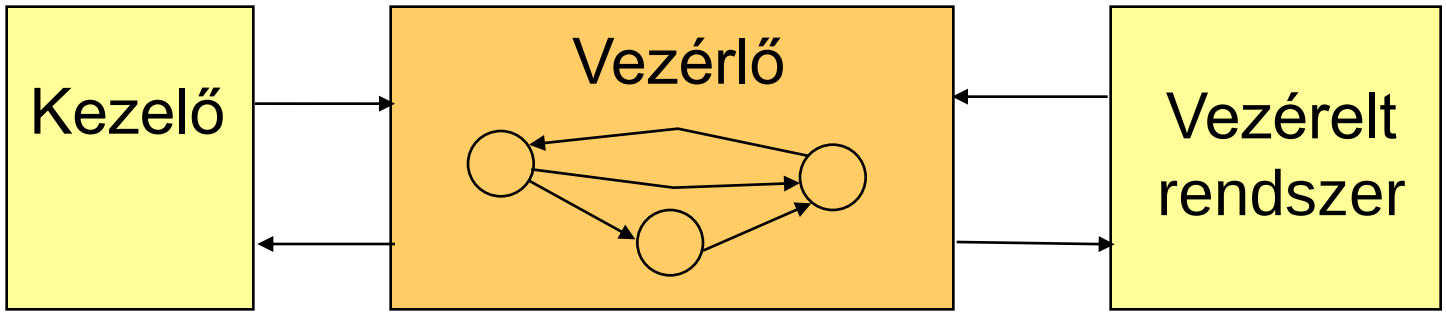
- Kimenetek hatása a bemeneteken keresztül ellenőrizve
- A szabályzási kör stabil



Példa: Vizsgálati szempontok reaktív rendszerekre

- Álla
- Ber
- Kin
- Kim
- Állapotátmenetek
- Ember-gép interfész

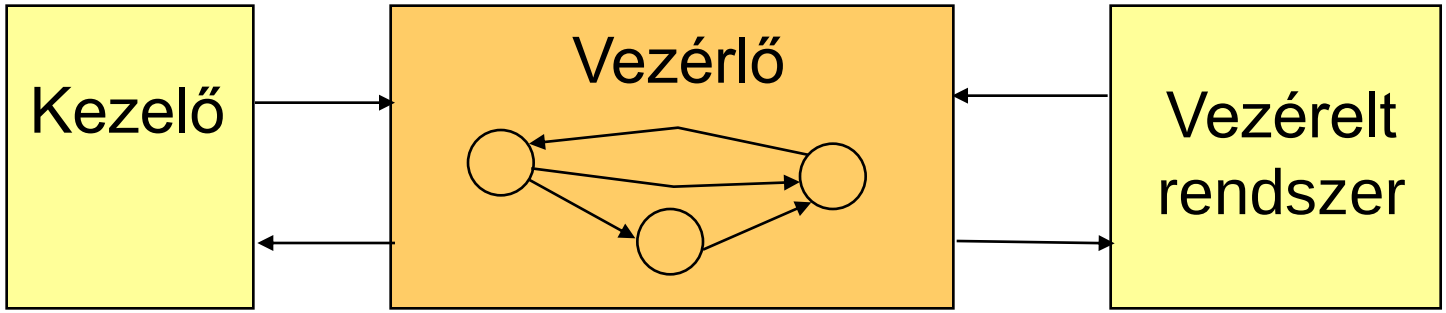
- Minden állapot elérhető statikusan
- Állapotátmenetek visszafordíthatók (visszaút van)
- Több átmenet van veszélyes állapotból biztonságosba
- Megerősített átmenet van biztonságos állapotból veszélyes állapotba



Példa: Vizsgálati szempontok reaktív rendszerekre

- Álla
- Be
- Kir
- Kin
- Állapotátr tek
- Ember-gép interfész

Kezelő felé kimenő események specifikációja:
- Sorrendezés előírt (prioritással)
- Frissítés előírt
- Gyakoriság korlátozott (kezelő terhelhetősége)



Ellenőrzési módszerek

- Módszer: Statikus analízis
  - Hiányosságok, ellentmondások kiszűrése a specifikáció (illetve terv, kód) végrehajtása nélkül
  - Ellenőrző listák alkalmazhatók (tipikus hibákra, jó megoldásokra; teljesség nem garantált)
- Eszközök:
  - Kézi átvizsgálás („átolvasás és értelmezés sorról sorra”)
  - Automatikus ellenőrzés: (fél)formális specifikáció esetén

Hogyan dokumentálható az ellenőrzés eredménye?

- Szoftverkövetelmények igazolójelentése
  - Megvalósítás dokumentálása
    - Felülvizsgálat
    - Egyenrangú átvizsgálás
  - Vizsgálati szempontok szerinti eredmények dokumentálása
    - Ellenőrző lista
    - Követhetőségi vizsgálatok
    - Statikus analízis
    - Formális verifikáció
    - …
  - Összefoglaló vélemény
    - Minőségi értékelés
    - Szükséges javítások előírása

Tartalomjegyzék

- Motiváció
  - Miért fontosak a tervezési folyamat ezen szakaszai?
  - Milyen elvárások vannak a specifikációval szemben?
  - Milyen módszerei vannak a specifikáció készítésnek?
- A követelménykezelés általános feladatai
  - Követelmények nyilvántartása
  - Követhetőség a verifikációhoz
- Félformális specifikáció
  - Specifikus technika: SysML
- A specifikáció verifikációja
  - Általános kritériumok
  - Példa: Specifikus kritériumok UML állapottérképekre

Példa: IAR VisualState eszköz

A statikus ellenőrzés UML állapottérképeken:

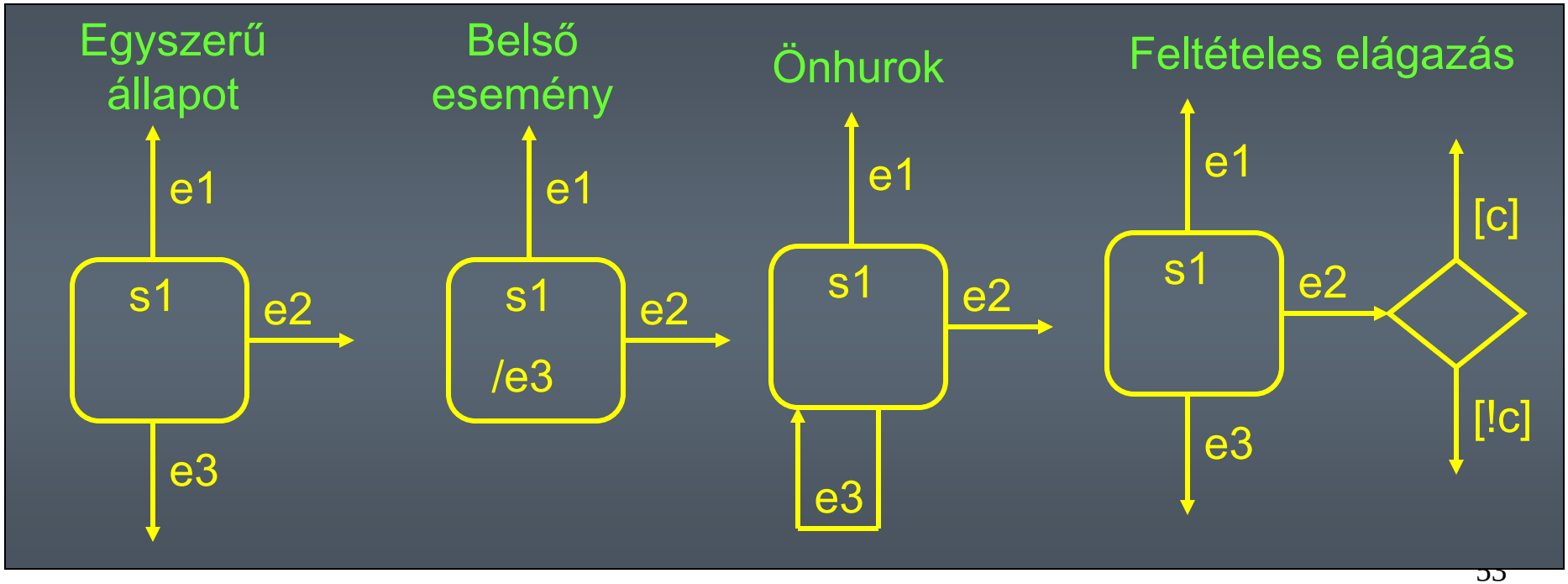
- Reset funkcionalitás
- Események, akciók felhasználása
- Állapotátmenetek statikus engedélyezettsége
- Állapotátmenet konfliktusok (azonos trigger)
- Állapotok statikus elérhetősége
- Nyelő állapot keresése (nincs kimenet)

Teljesség és ellentmondásmentesség ellenőrzése UML állapottérképek esetén

- Kritériumok:
  - Teljesség: Minden eseményre, minden állapot-konfigurációban van specifikált viselkedés (tranzíció vagy helybenmaradás)
  - Ellentmondásmentesség: Egy eseményre egy állapot-konfigurációban csak egy tranzíció lehet engedélyezett
- Kihívások:
  - Hierarchikus állapot-definíció: Állapot helyett állapot-konfiguráció ellenőrzése kell
  - A prioritások figyelembevétele az állapotátmenetek esetén
  - Konkurens régiók: Konkurens tranzíciók
    - Akciók determinisztikus sorrendje
  - Őrfeltételek használata: Kiértékelés
    - Teljesség: Bármely állapot-konfigurációban egy esemény által triggerelt tranzíciók őrfeltételeinek VAGY kapcsolata igaz értéket ad
    - Ellentmondásmentesség: Bármely állapot-konfigurációban az egy esemény által triggerelt tranzíciók őrfeltételei közül csak egy lehet igaz

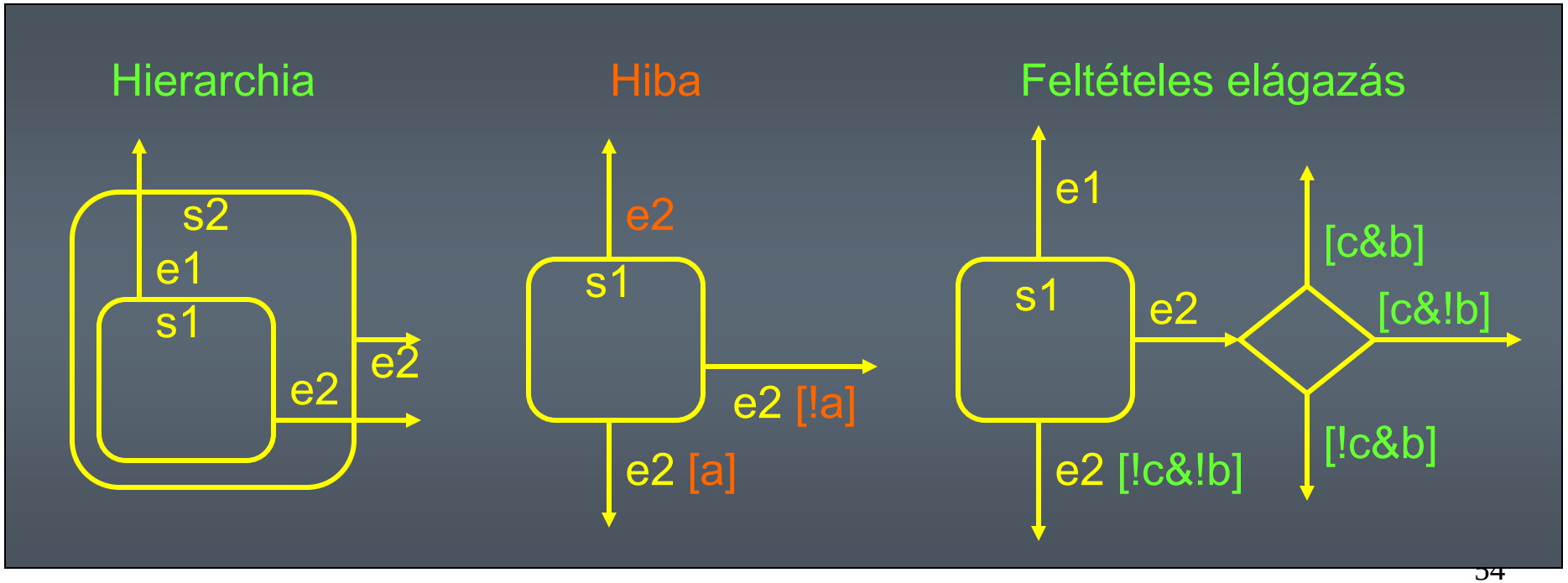
Teljesség

- Minden állapotkonfigurációból, minden eseményre vonatkozóan, minden őrfeltétel-kiértékelés esetén kell lennie definiált tranzíciónak



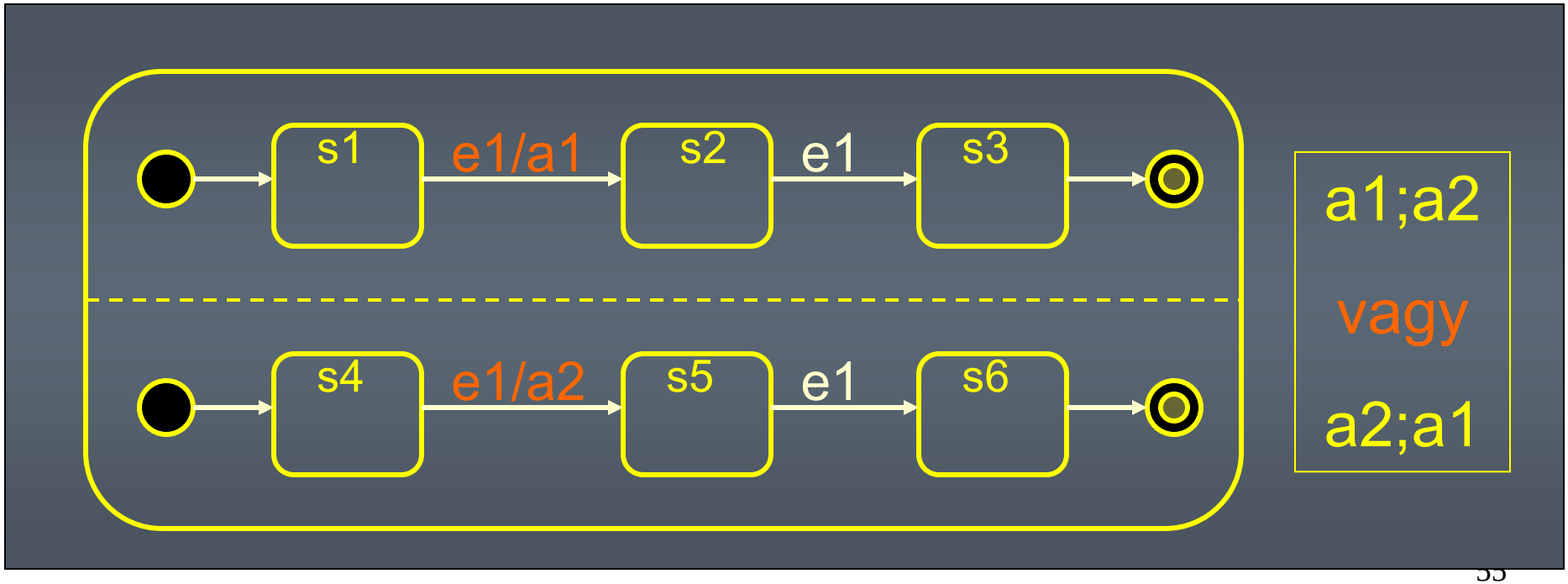
Egyértelműség I.

- Minden állapotkonfiguráció és minden esemény esetén az összes őrfeltétel-kiértékelés mellett egy hierarchia szinten belül egy időben csak egy tranzíció lehet engedélyezett



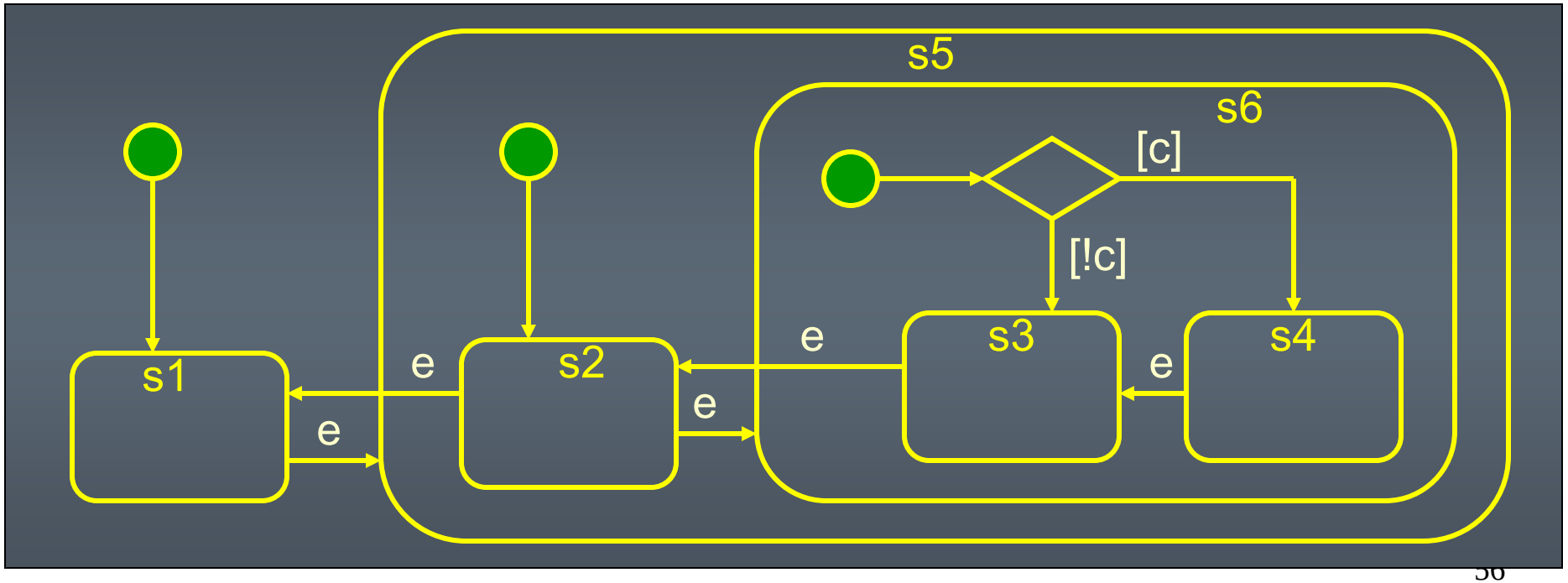
Egyértelműség II.

- Konkurens állapotgépeken belül egyazon eseményre csak az egyik állapotgépben legyen akció definiálva



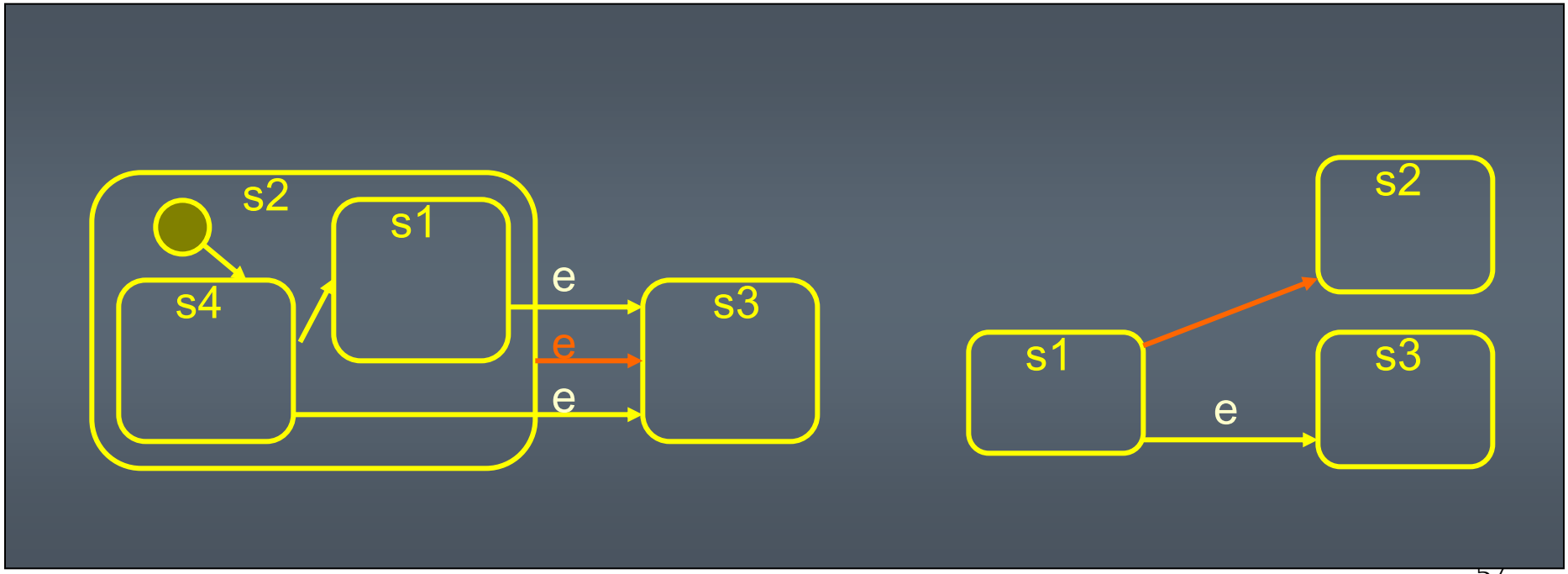
Indulási állapot I.

- Minden rész-automatában szerepelnie kell kezdőállapotnak,
  beleértve a legfelső szintű régiót is



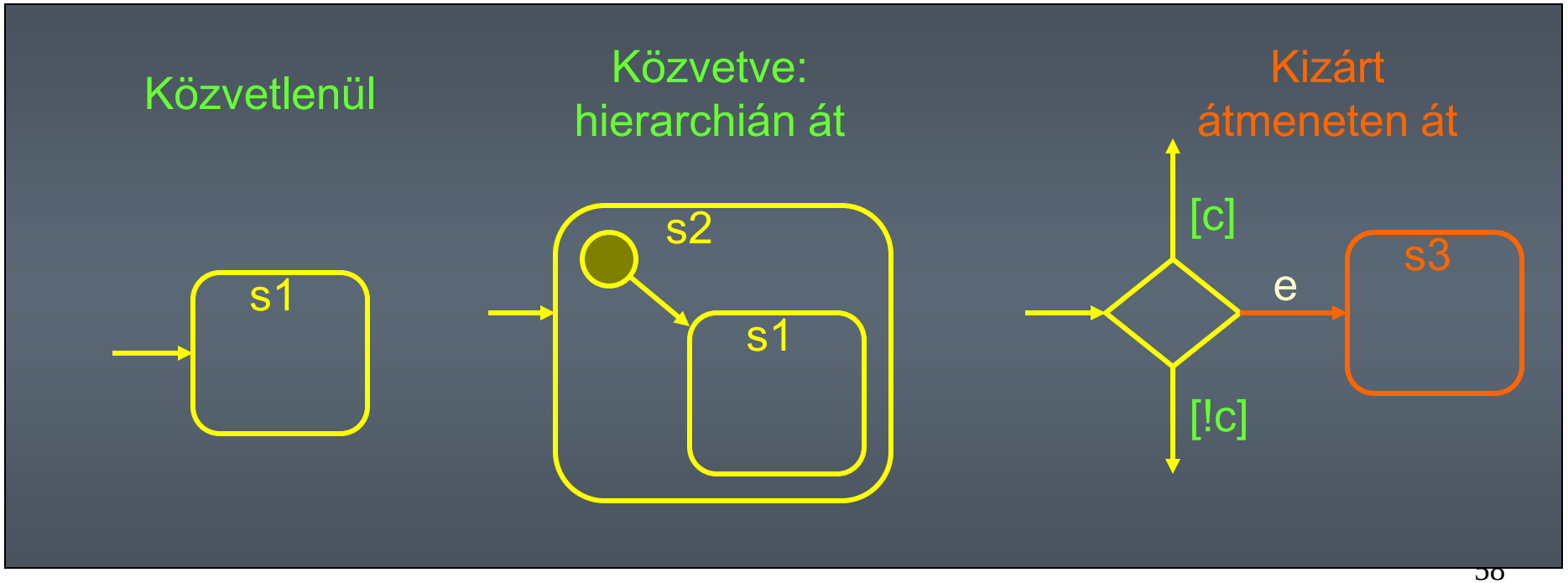
Tranzíciók takarása

- A hierarchia miatt tranzíciók takartak lehetnek
- Trigger nélküli tranzíció takarja az eseménnyel triggerelt tranzíciót



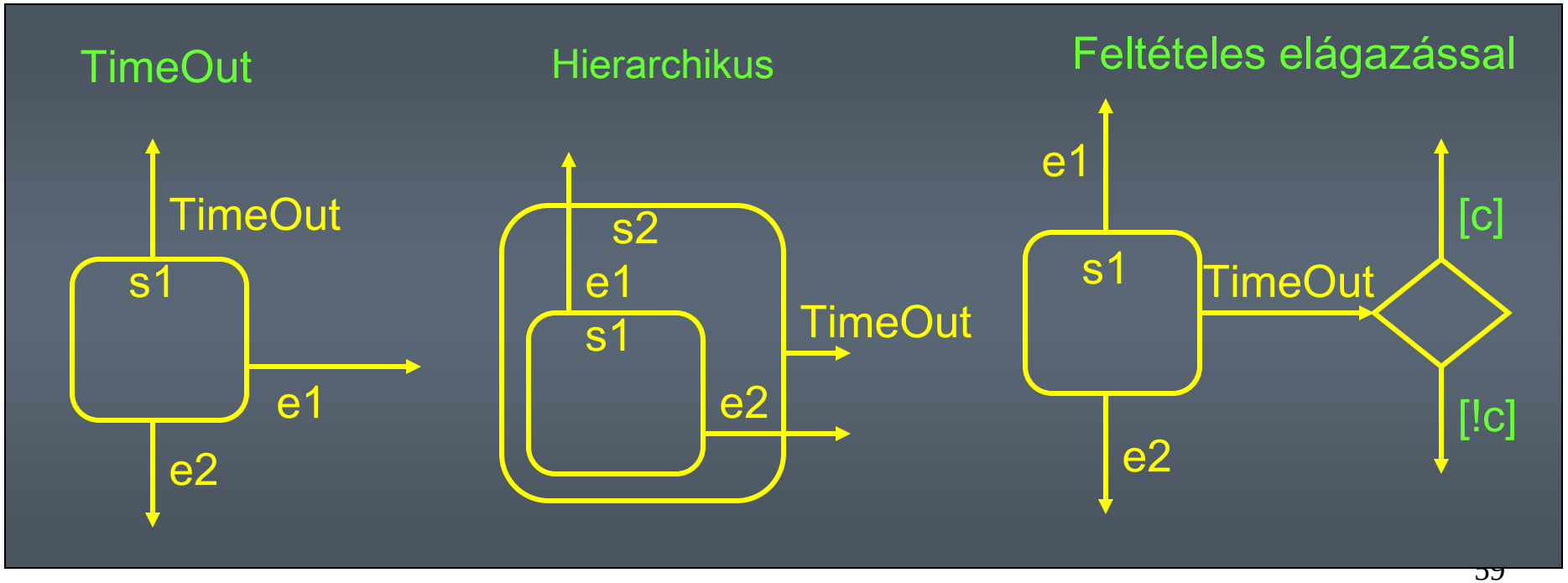
Elérhetőség

- A rendszer minden egyszerű állapotának elérhetőnek kell lennie vagy közvetlenül, vagy közvetve



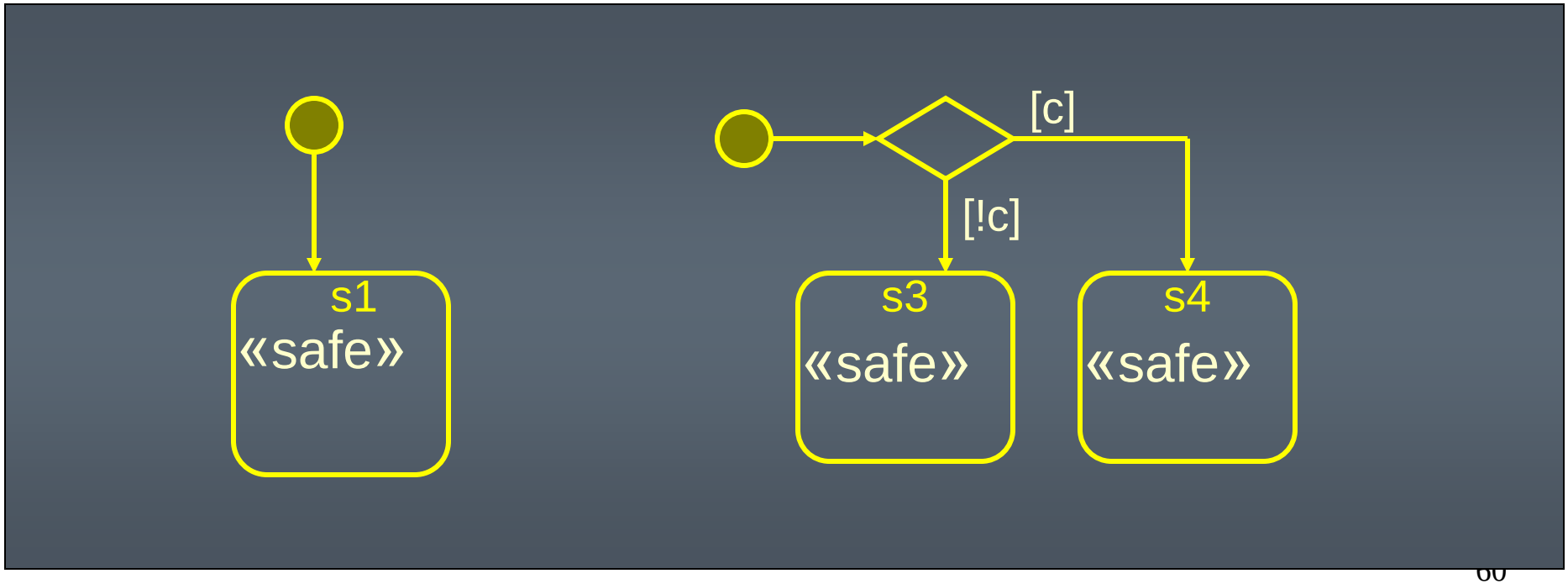
Beágyazott vezérlőkre: Időtúllépés

- Minden állapotkonfigurációra vonatkozóan definiálva kell lennie olyan tranzíciónak, mely a TimeOut nevű eseményre van triggerelve (lehet felsőbb szintű tranzíció is)



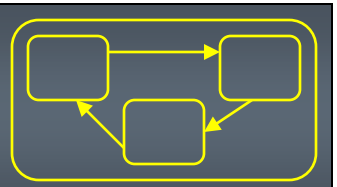
Beágyazott vezérlőkre: Biztonságos kezdőállapot

- A legfelső szintű régió kezdőállapotának biztonságosnak kell lennie:
  a kezdő pszeudoállapotból közvetlenül elérhető állapotok «safe» sztereotípiájúak kell legyenek

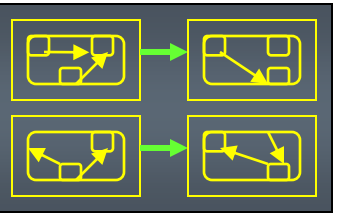


A statikus ellenőrzés menete

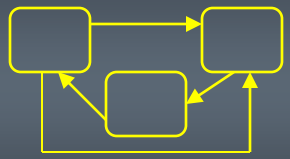
Eredeti modell



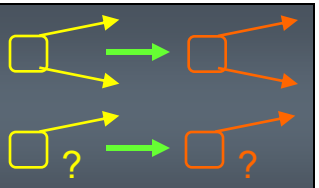
Transzformáció-sorozat



Redukált alak



Hibaminta illesztés

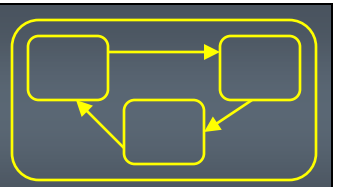


Hibaüzenet-lista

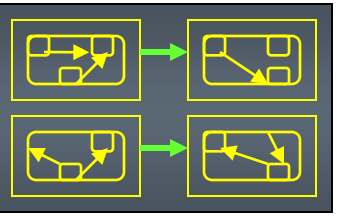
Hiba: definiálatlan
Hiba: kétértelmű…
Hiba: felesleges…
Hiba: hiányzik…
Hiba: elérhetetlen

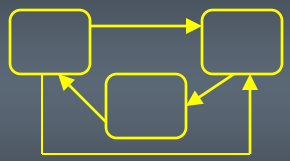
A statikus ellenőrzés menete

Eredeti modell

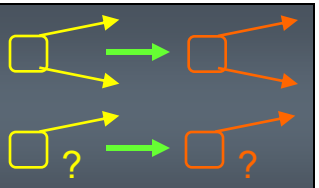


Transzformáció-sorozat





1. Események kigyűjtése
2. Átmeneti állapotok megszüntetése
3. Párhuzamos állapotok összerendelése
4. Hierarchia felbontása
5. Entry/exit áthelyezése
6. Belső akciók konvertálása önhurokká
7. Pszeudoállapotok, őrfeltételek konvertálása



Hiba: definiálatlan
Hiba: kétértelmű…
Hiba: felesleges…
Hiba: hiányzik…
Hiba: elérhetetlen

Miről volt szó?

- Motiváció
  - Miért fontosak a tervezési folyamat ezen szakaszai?
  - Milyen elvárások vannak a specifikációval szemben?
  - Milyen módszerei vannak a specifikáció készítésnek?
- A követelménykezelés általános feladatai
  - Követelmények nyilvántartása
  - Követhetőség a verifikációhoz
- Félformális specifikáció
  - Specifikus technika: SysML
- A specifikáció verifikációja
  - Általános kritériumok
  - Példa: Specifikus kritériumok UML állapottérképekre