

Követelménykezelés

A követelményspecifikáció ellenőrzése

Majzik István

Egyes ábrák: Pap Zsigmond, Polgár Balázs

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

Tartalomjegyzék

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



How the customer explained it



How the Project Leader understood it



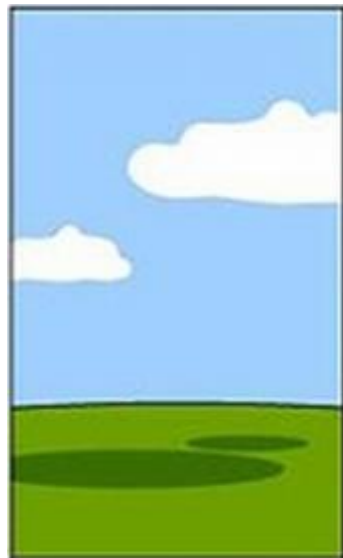
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



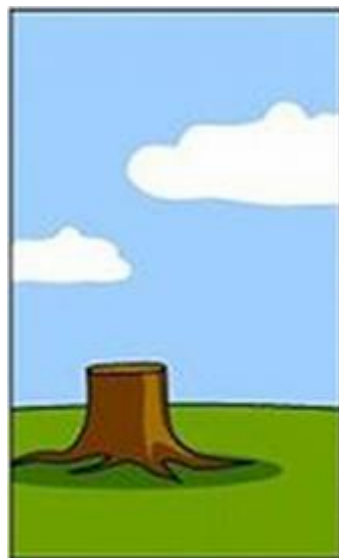
How the project was documented



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: 203 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% időzíteni 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 (specification, requirement specification, system specification):
 - Tervezők, fejlesztők felé átalakított elvárások
 - Elemzés (absztrakció, strukturálás, szűrés) eredménye
 - Sokféle típus
 - Rendszerspecifikáció, biztonsági specifikáció, ...
 - Verifikáció alapja

Elvárások a specifikációval szemben

- A követelmények teljes lefedése
 - Funkcionális követelmények
 - Extra-funkcionális követelmények
- Megfogalmazás: Egyértelmű, igazolható, megvalósítható
- Javasolt 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

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

A követelménykezelés feladatai (áttekintés)

- Követelmények hatékony, strukturált tárolása
 - Hierarchikus elrendezés, tulajdonságokkal
- Követelmény **életciklus** támogatása
 - Felvétel, törlés, változás, kapcsolatok megjelenése
- Követelmények **finomítása és követhetősége**
 - Specifikáció -> 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

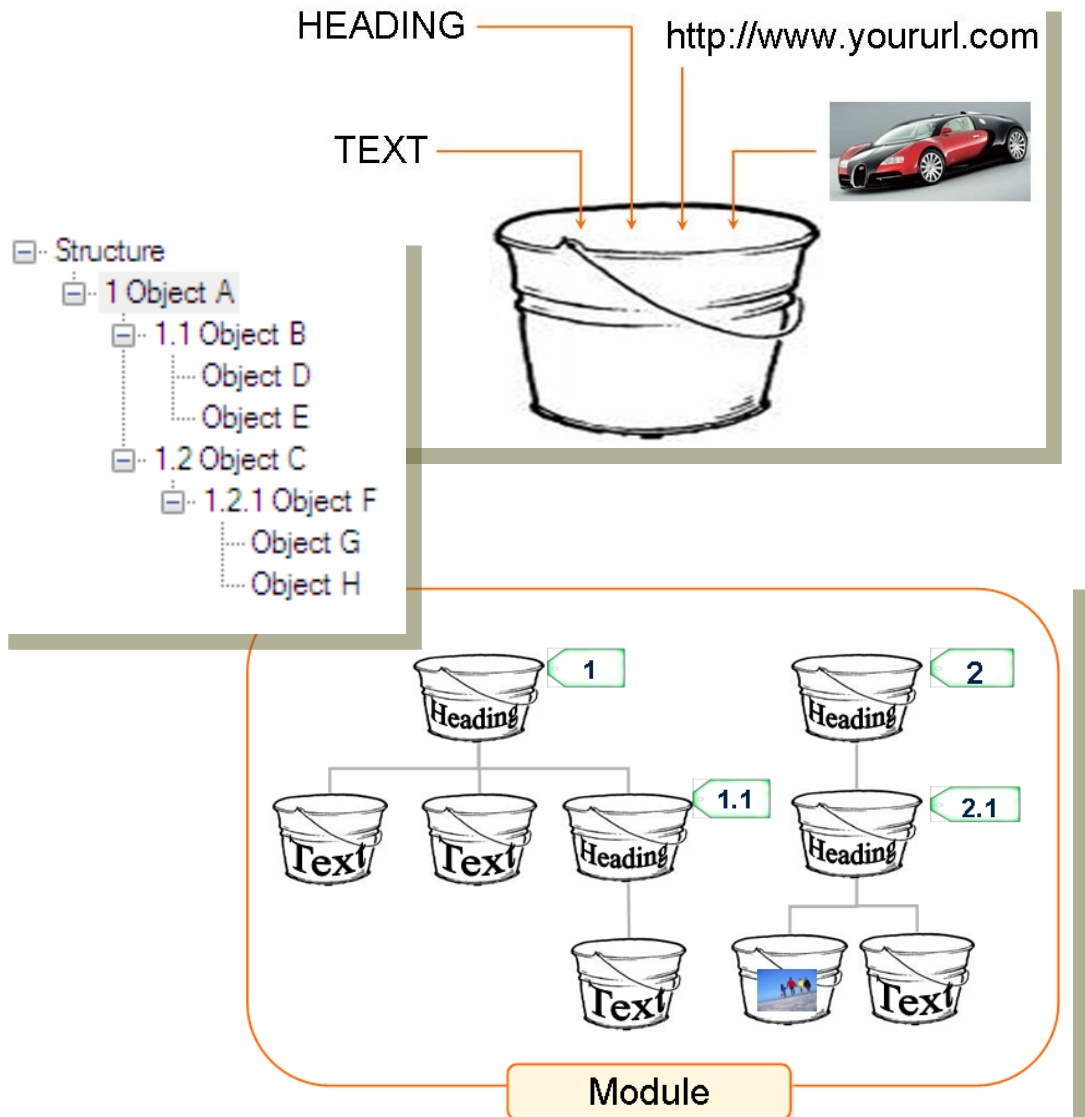
- **Természetes nyelvű követelmény dokumentum**
 - Strukturálás adott (fejezetek, alfejezetek)
 - Követelményazonosítók felvétele
- **Követelményfinomítás: Táblázatos nyilvántartás**
 - Követelményazonosítók szerepelnek
 - Különböző dokumentumokból (SRS, SA, MDS, MTS, ...)
 - Analízis makrókkal támogatható
 - Üres, többszörös, ... mezők kikeresése
- **Követhetőségi mátrix: Kapcsolatok megjelenítése**
 - Követelmények azonosítói
 - Kódrészlet azonosítók (funkció szint tipikus)
 - Teszt azonosítók
 - Sikeres/sikertelen teszteredmény bejelölése

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

- Objektum adatbázis
 - Általános „tároló”
 - 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		... form text
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	a text field.
TRN-CSR-6	Bill Young	2.2 User sizes	Mand		
TRN-CSR-7	Bill Young	People come in all shapes and sizes. The car must be suitable for people maximum and minimum sizes of fgfg to 2 m weighing 25 kilograms to	Mand		

Tulajdonságok

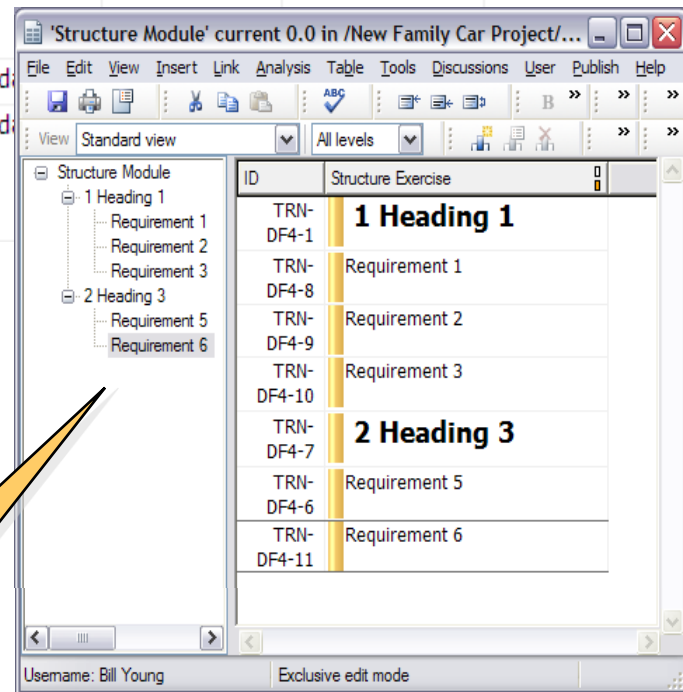
Objektum azonosító

Változás-jelző

Fejléc objektum

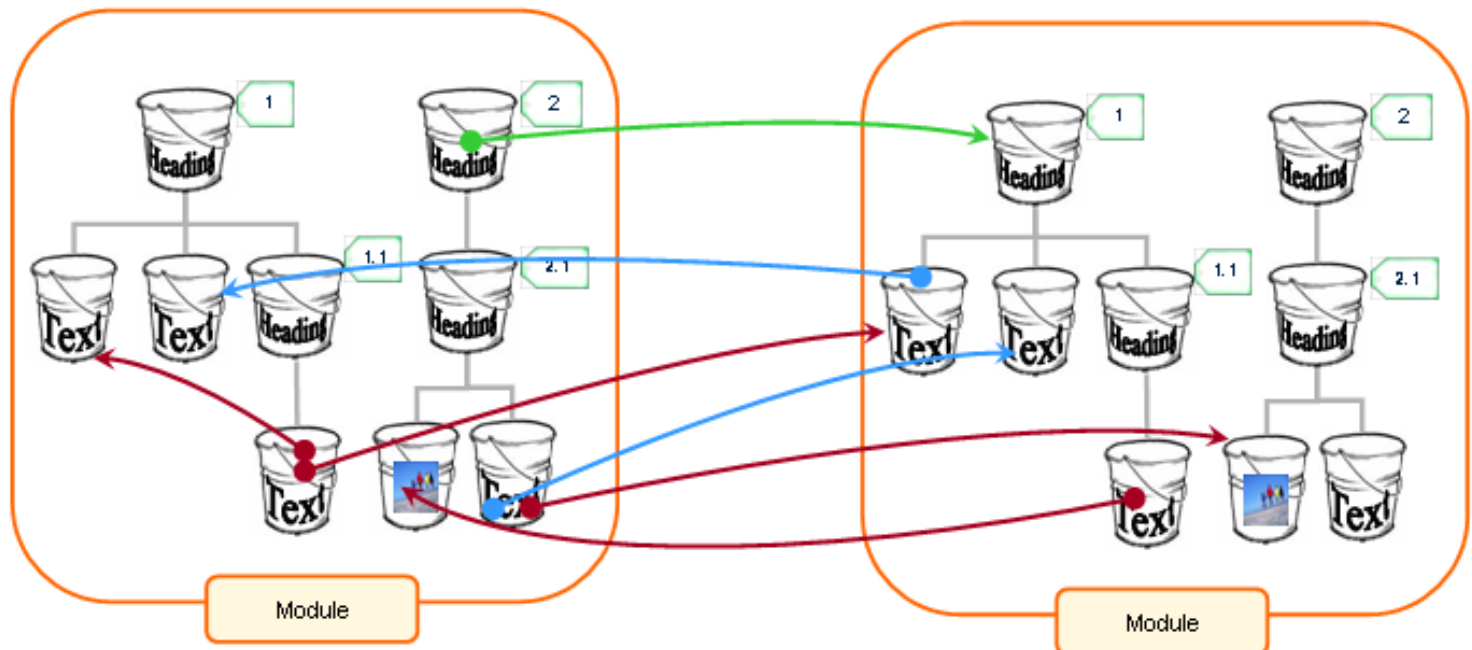
Szöveges objektum

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

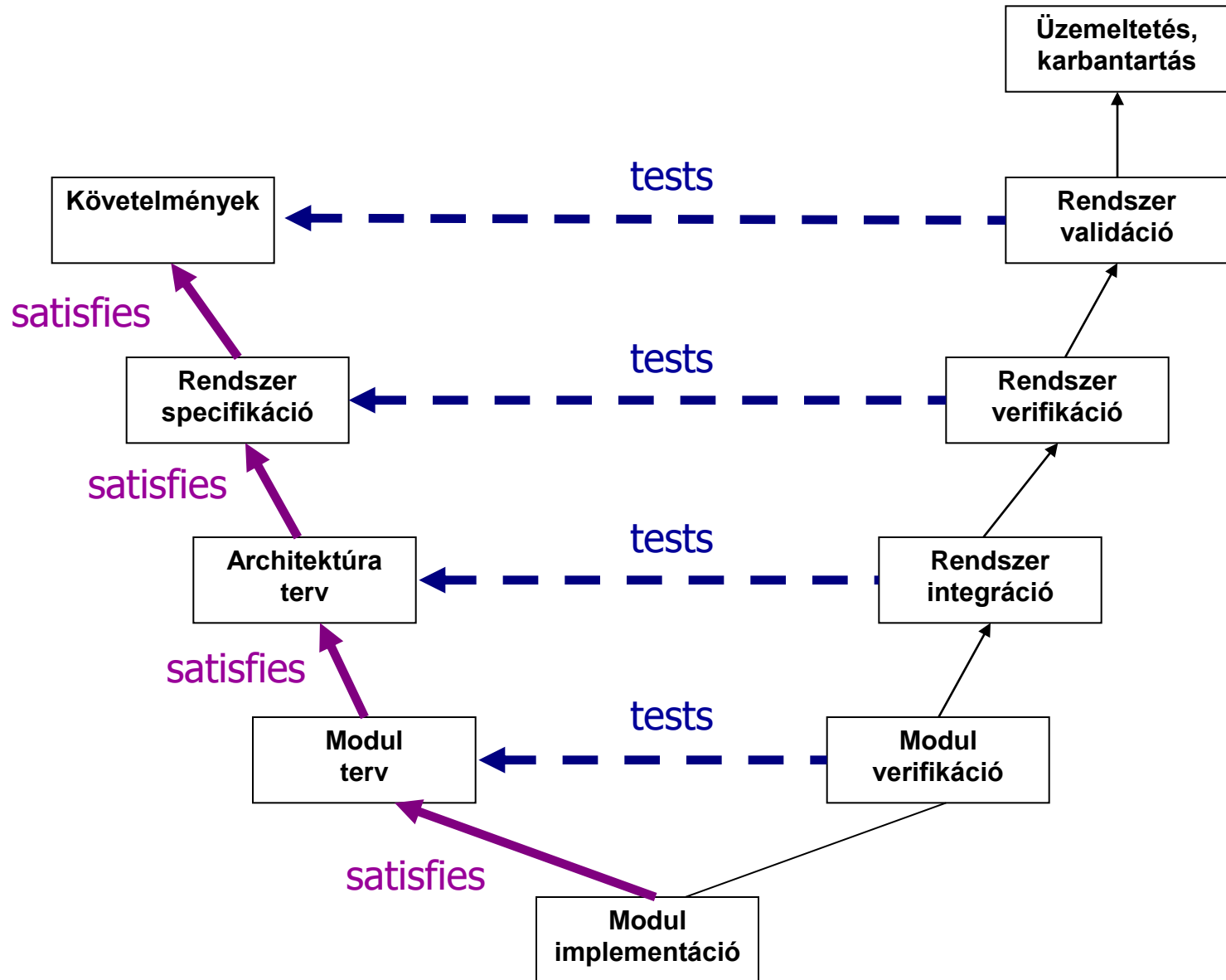
satisfies

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-modelben



Megvalósítás: Követelmény életciklus

- **Objektum 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
 - Szűrés: Navigálás kód objektumokig, teszt objektumokig
 - Objektumok kigyűjtése: Nincs kapcsolat adott célhalmazig
 - Pl. 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 eszközláncok

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

Példa: Validációs követelmények beillesztése

ID	DECOS Artifacts	Version	Status	Data Directory Remote
AUT1	1 PIMs			
AUT2	PIM for Tiny Example	1.0	Failed	ftp://FTPDecos@ftp.sp.se/private/DECOS/PIM/1.0
AUT3	PIM Updated for Tiny Example	1.1	Completed	ftp://FTPDecos@ftp.sp.se/
AUT5	2 SCADE Model			
AUT6	SCADE model for Roll Control	1.0		

ID	VVStatus	Tool Integration Status	Type	
1	1 PIM Validation	Completed	Compound	
51	PIM Validation	Completed	Not processing	Elementary

ID	VVStatus	Tool Integration Status	Type	
1	1 SCADE Model Validation	Completed	Compound	
51	SCADE Model Validation	Completed	Not processing	Elementary

V-Plan for
PIM
validation

V-Plan for
SCADE model
validation

Példa: Eszközláncok indítása a DOORS-ból

Eszközök végrehajtása:

- **Belső** eszköz: Pl. egyszerű ellenőrző lista
 - Megvalósítható a DOORS keretein belül
- **Helyi** eszközvégrehajtás: pl. PROPANE (SWIFI)
 - Automatikus indítás (scriptből) és eredmény becsatolás
- **Távoli** eszközvégrehajtás: pl. RACER
(metamodellek és modellek ontológia alapú vizsgálata)
 - Indítás távoli hozzáféréssel (pl. üzenet alapú)
- **Kézi** eszközvégrehajtás: pl. EMI Hardware Test Bench
 - Eszköz indítási és eredmény beviteli dialógus

Eredmények tárolása:

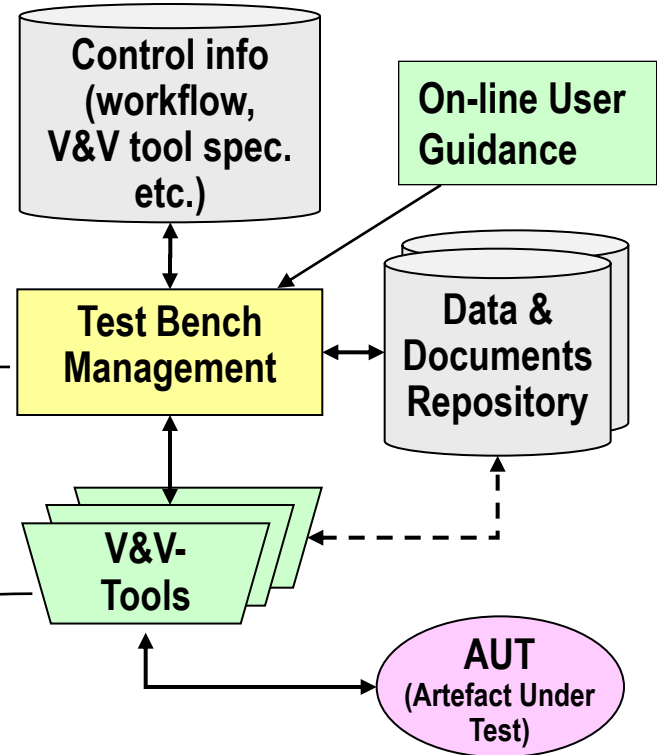
- Távolról is elérhető **adattár** (repository)

Példa: Verifikáció futtatása a követelménykezelőből

Formal module: /DECOS_TestBench/V-Plans/Components and Middleware: current 0.0 - DOORS

ID		VVStatus	Type	Phase	V&V-Activity
VPCM59	1 V-Plan Components and Middleware	Not ready			
VPCM1	Arch-gen-1	Not ready	Compound		
VPCM2	Arch-gen-core-1	Completed	Compound		
VPCM3	Arch-core-predictable-transport-1	Completed	Elementary		
VPCM4	Arch-core-ft-clock-sync-1	Completed	Elementary		
VPCM5	Arch-core-fault-isolation-1	Completed	Compound		
VPCM6	Arch-core-fault-hypothesis-1	Completed	Elementary		
VPCM7	Arch-core-never-give-up-1	Completed	Elementary		
VPCM8	Arch-core-transient-faults-1	Completed	Elementary		
VPCM9	Arch-core-consistent-diagnosis-1	Completed	Elementary		
VPCM10	Arch-gen-core-2	Not ready			
VPCM11	Arch-DECOS-high-level-service-1				
VPCM12	Arch-DECOS-exec-1				
VPCM13	Arch-DECOS-com-1				
VPCM14	Arch-DECOS-com-2				
VPCM15	Arch-DECOS-com-3				

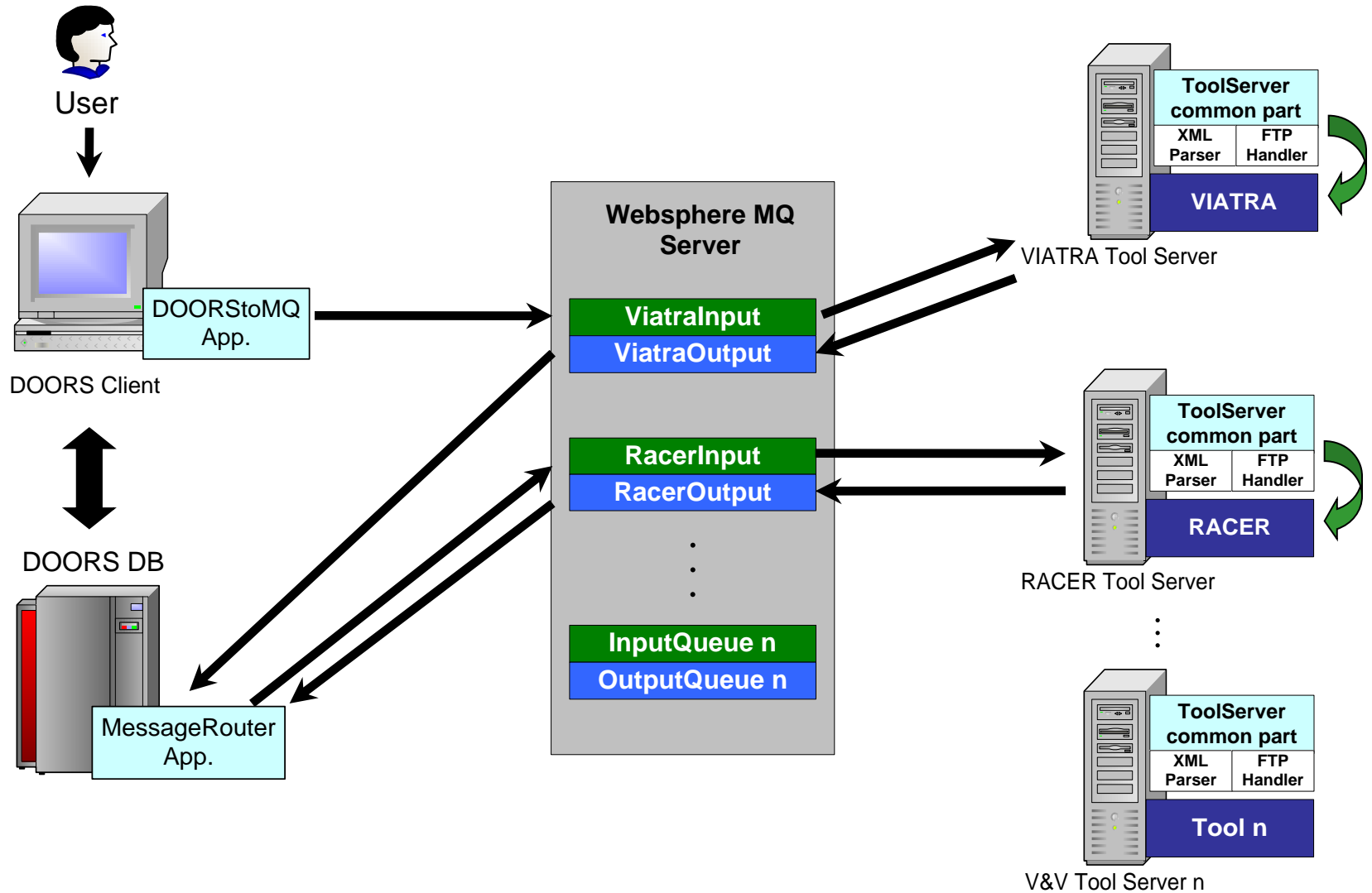
Implemented in DOORS™



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

- **Models**
- **Hardware**
- **Middleware**
- **Applications**

Példa: Eszközlánc végrehajtása



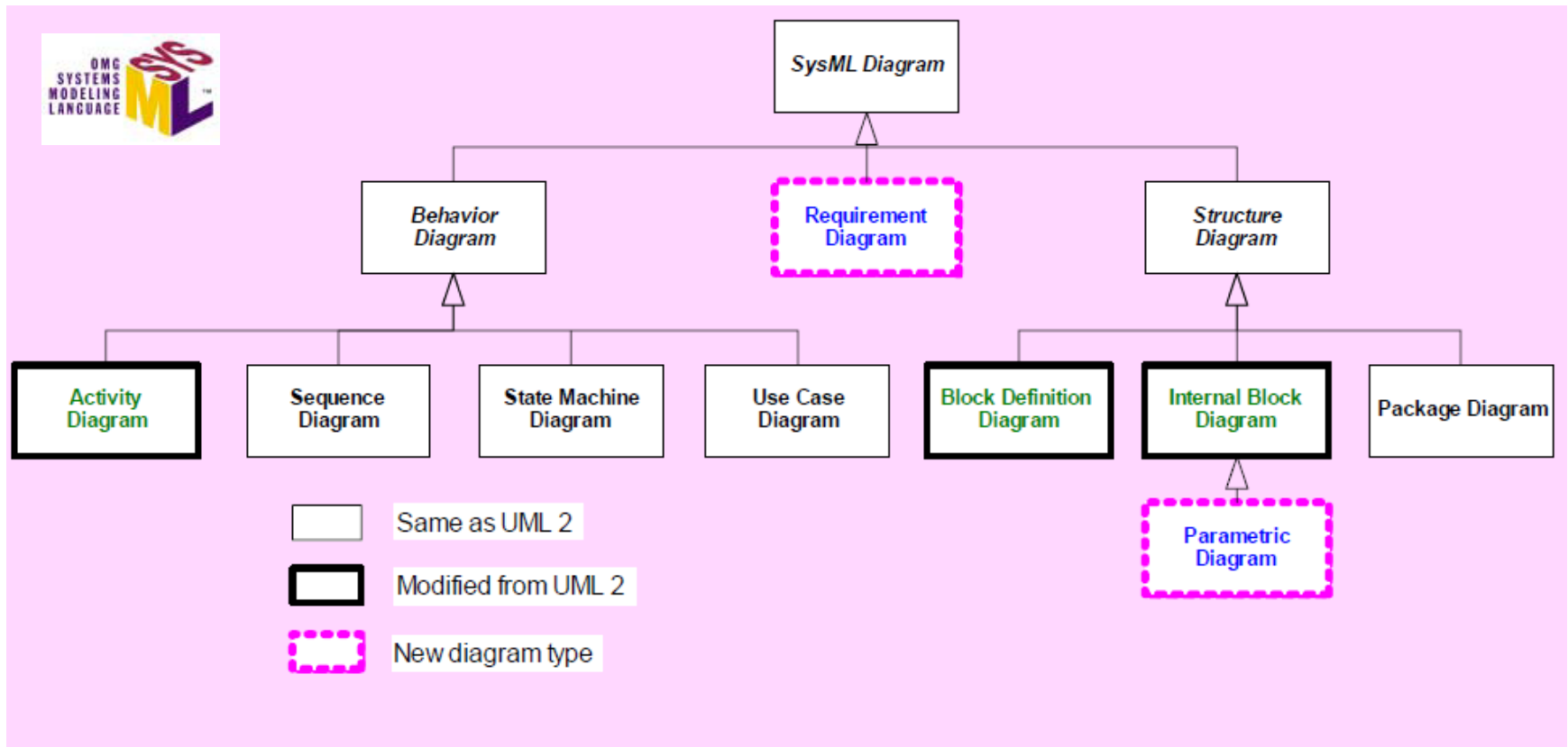
Tartalomjegyzék

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

Félformális követelményspecifikáció: SysML

- Systems Modeling Language

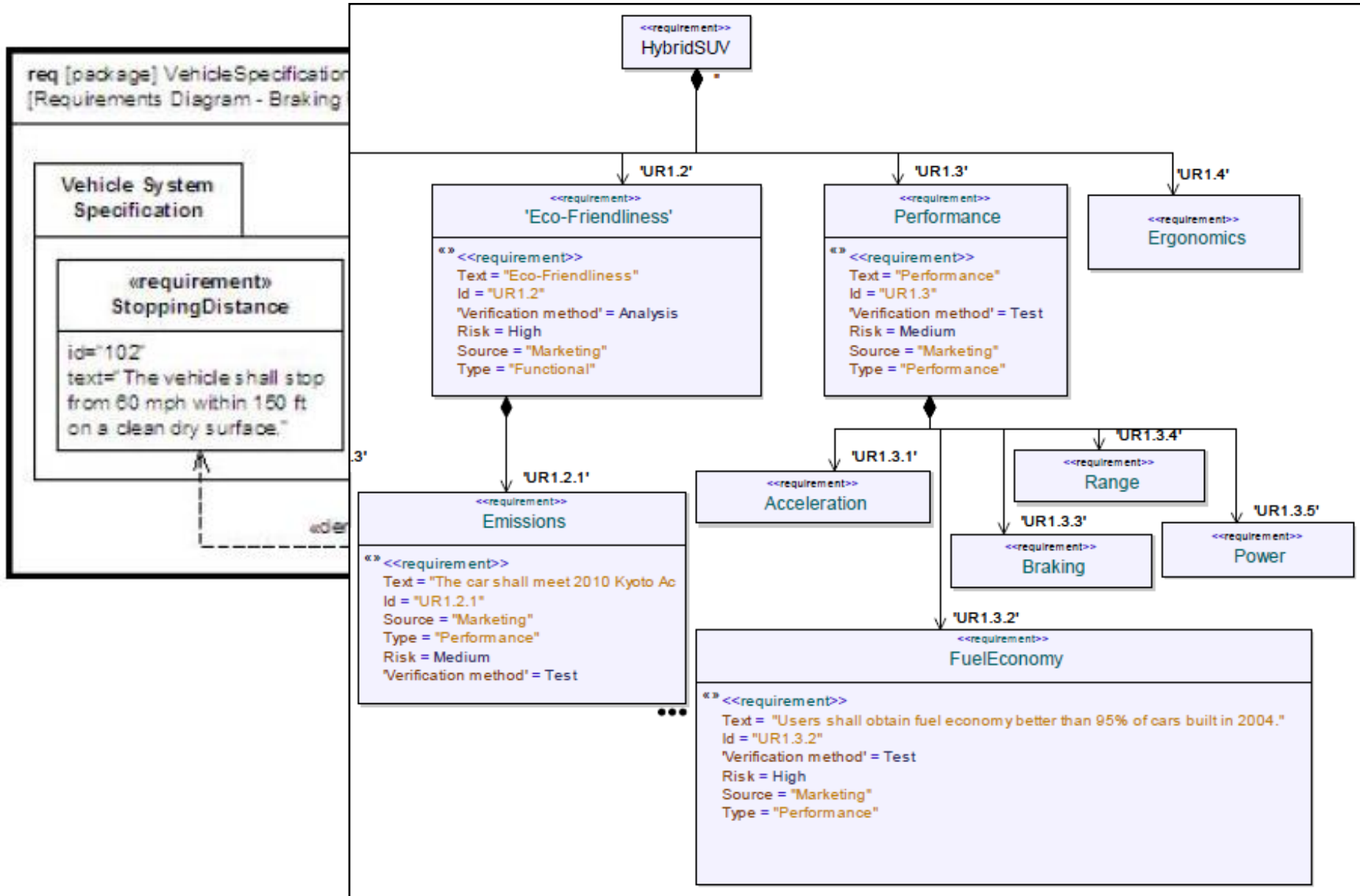
- UML részhalmoz egy kiterjesztése rendszertervezéshez
- Fő újdonságok: Requirement és Parametric diagram



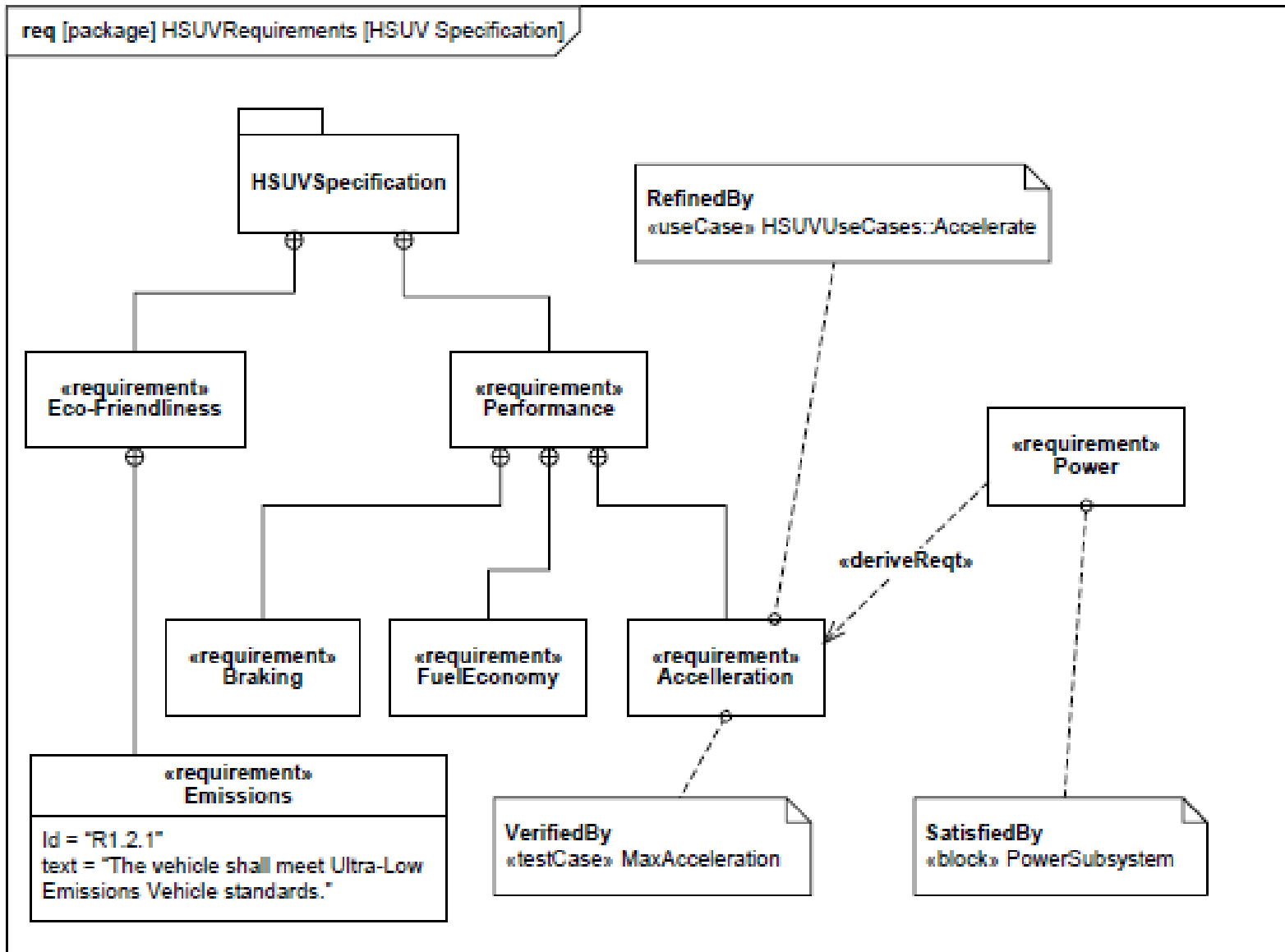
Requirement diagram

- **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)
 - **DeriveReq:** követelmények között (forrás – származtatott)
 - **Refine:** követelmények és terv elemek között (pl. szövegeshez)
 - **Satisfy:** követelmények és implementáció elemek között
 - **Verify:** követelmények és teszt elemek között

Requirement diagram példa: Struktúra

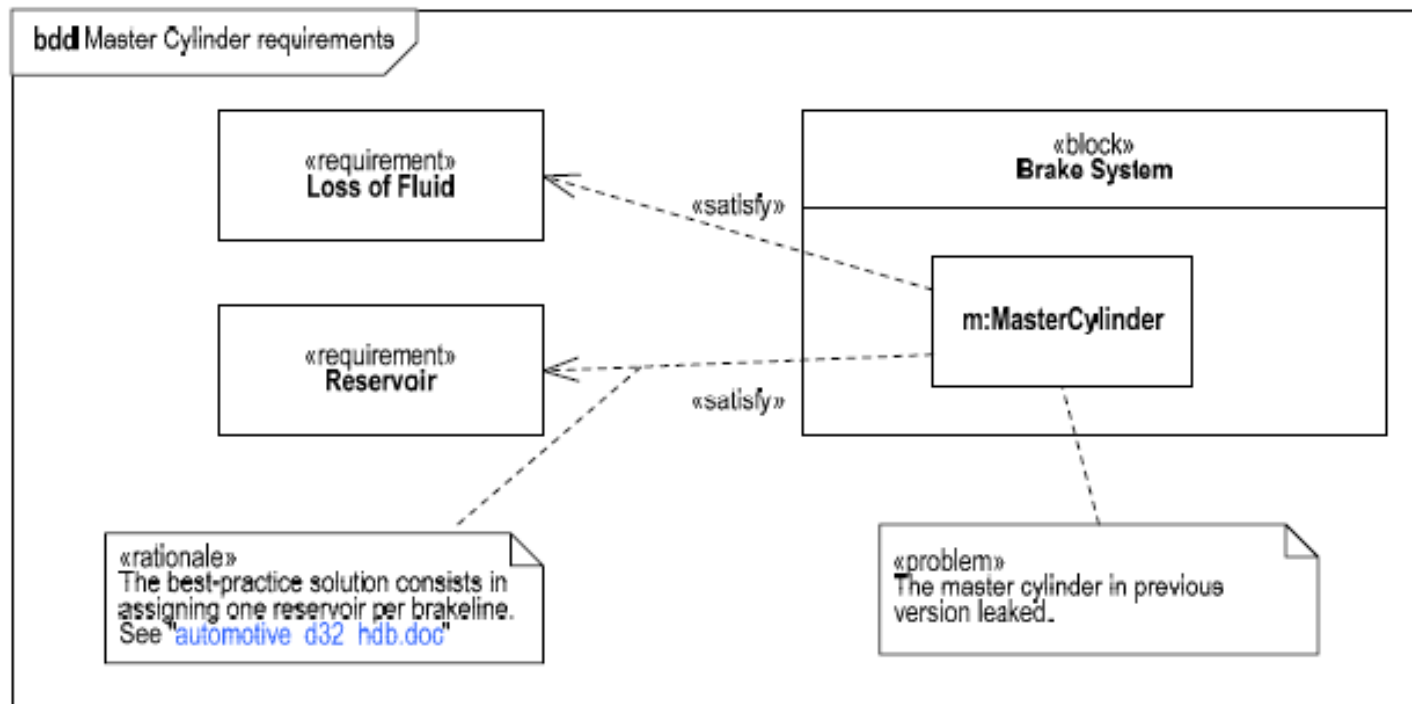


Requirement diagram példa: Relációk



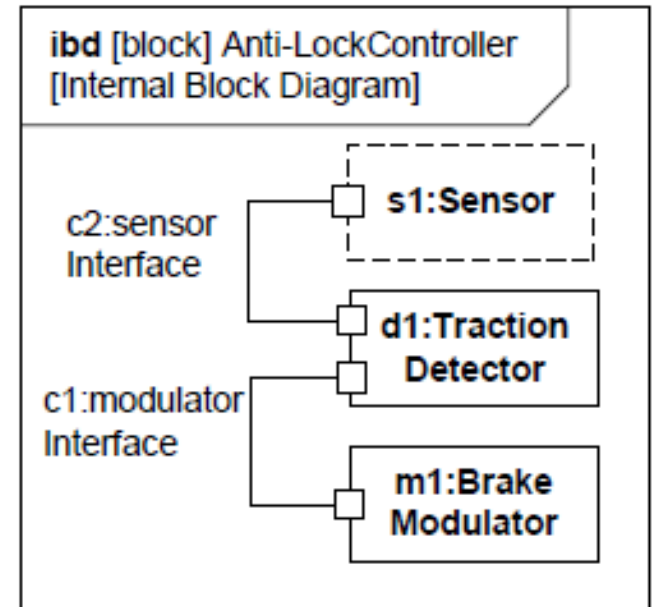
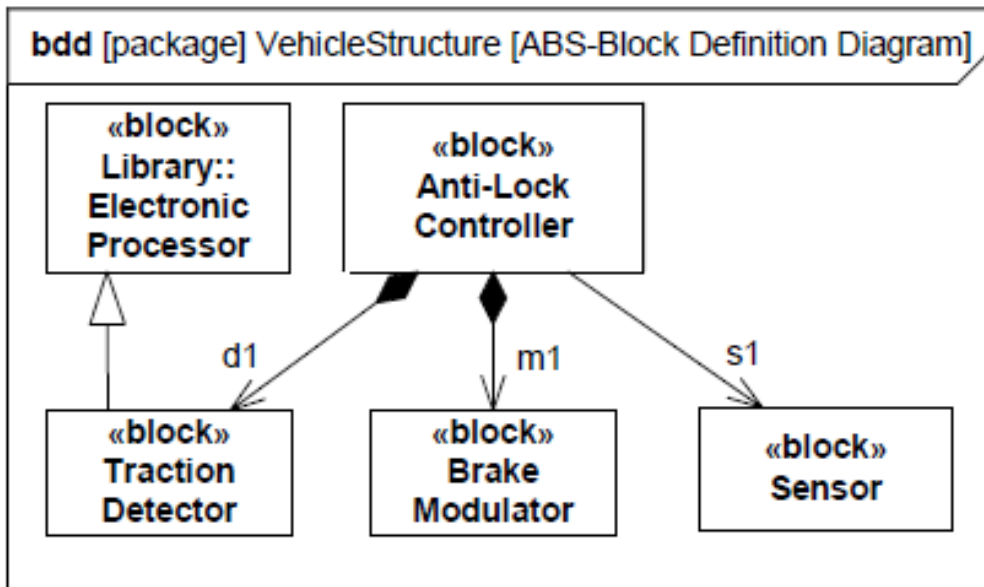
Requirement diagram példa: Tervezői döntések

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



Block diagram

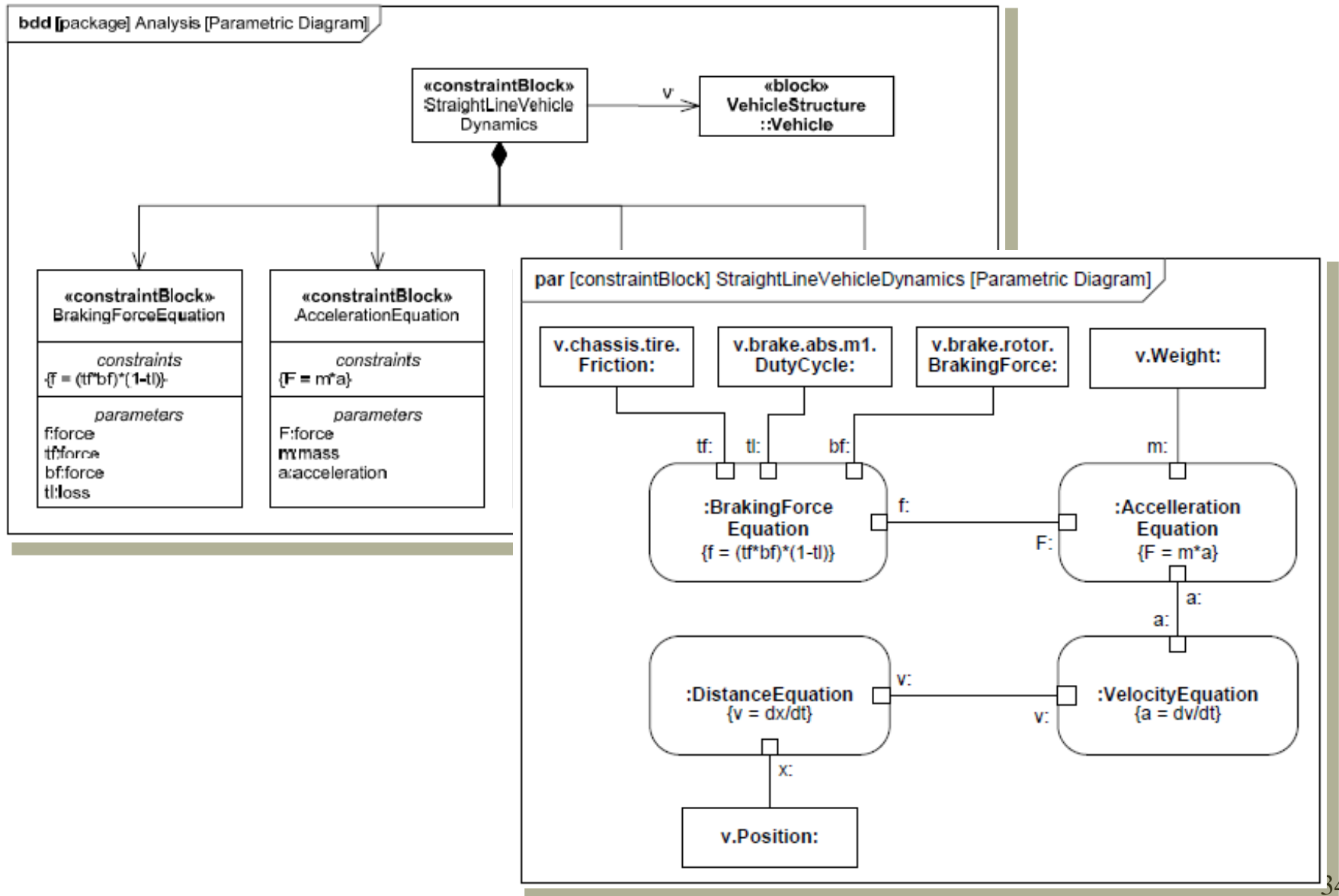
- **Block Definition Diagram:**
 - Block: A struktúra eleme (fekete / üveg doboz)
 - Komponens (nem csak szoftver)
 - A SysML-ben az UML 2.0 osztályokon alapul
- **Internal Block Diagram:**
 - Konkrét szerepek; típust a Block adja meg



Parametric diagram

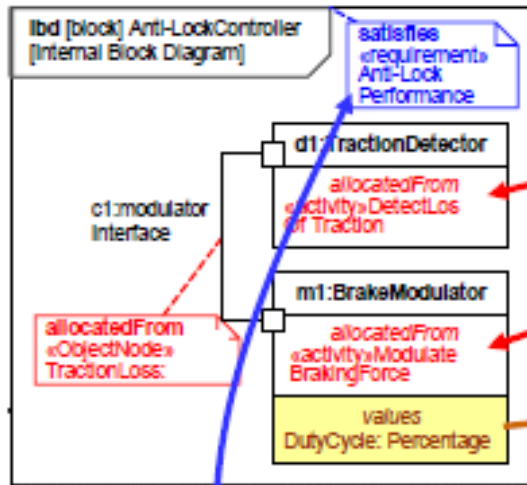
- Cél: Ellenőrizhető számszerű követelmények (kényszerek) megfogalmazása tulajdonságokra
 - Nem-funkcionális követelmények aspektusa
 - **Analízis** (pl. teljesítmény, megbízhatóság) támogatása
- ConstraintBlock: Összefüggések megadása
 - **Formális** (pl. MathML, OCL), vagy **informális** alakban
 -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

Parametric diagram példa

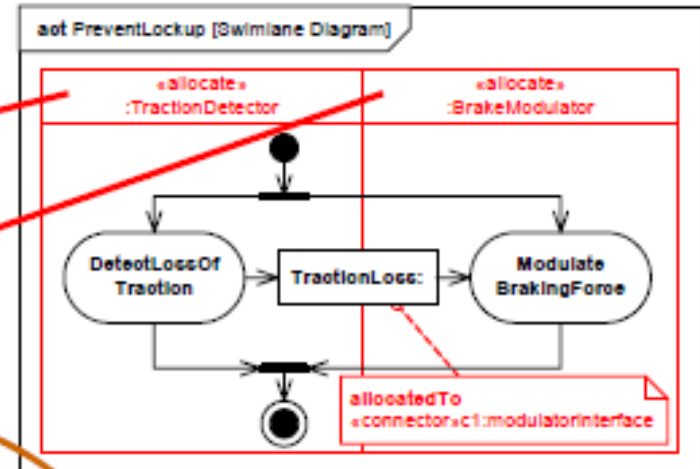


Relációk diagramok között: Követhetőség

1. Structure



2. Behavior

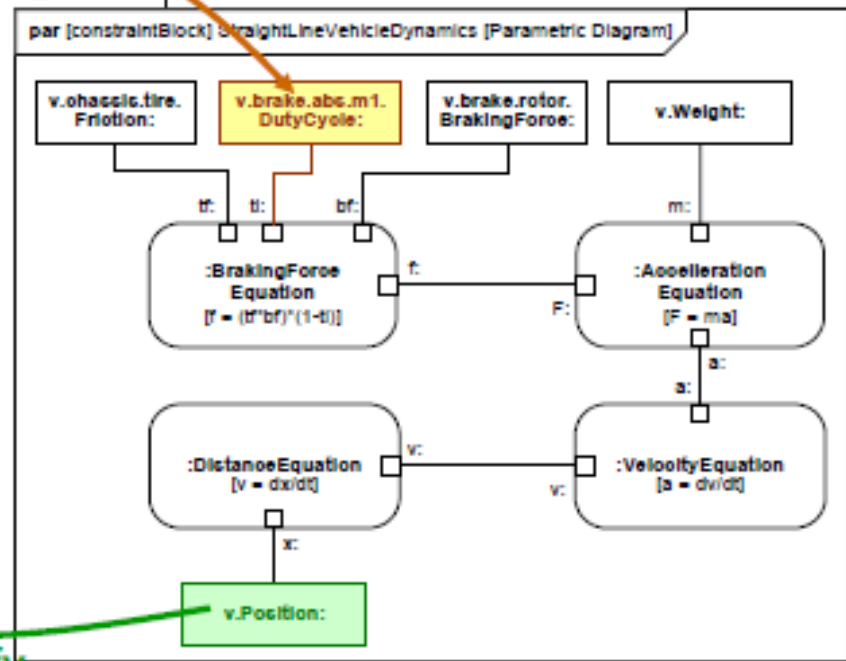
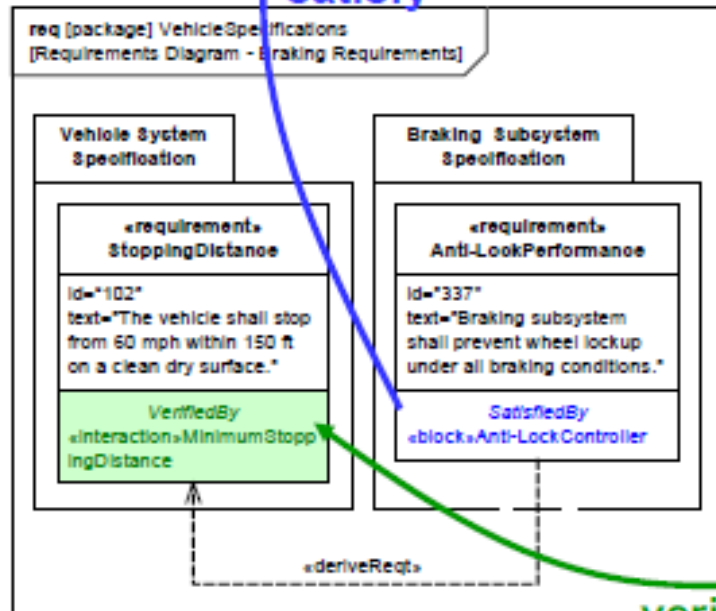


allocate

value binding

satisfy

verify



3. Requirements

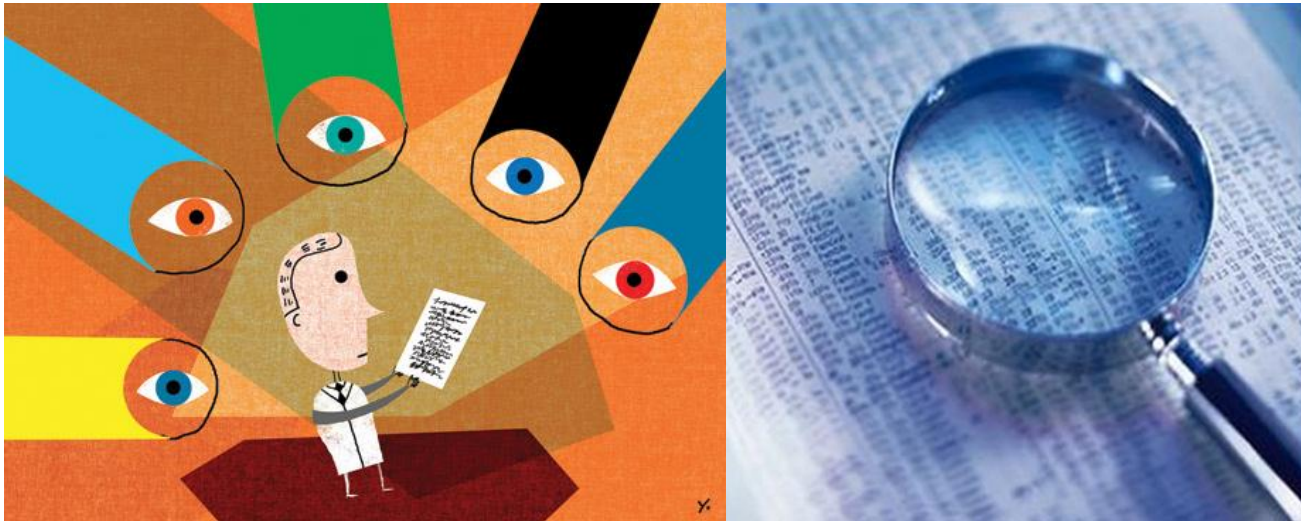
4. Parametrics

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
 - Specifikus kritériumok UML állapottérképekre (mintapélda)

Ellenőrzési módszerek

- Ellenőrző listák
 - Tipikus hibák esetén hatékony (újra ne kövessük el)
 - Teljességet nem várhatunk
- Statikus analízis
 - Hiányosságok, ellentmondások kiszűrése a specifikáció (ill. terv, kód) végrehajtása nélkül
 - Analógia: Átolvasás „sorról sorra”
 - Szerepek: Szerző, átvizsgáló, tesztelő



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

Vizsgálati szempontok biztonságkritikus rendszerekben

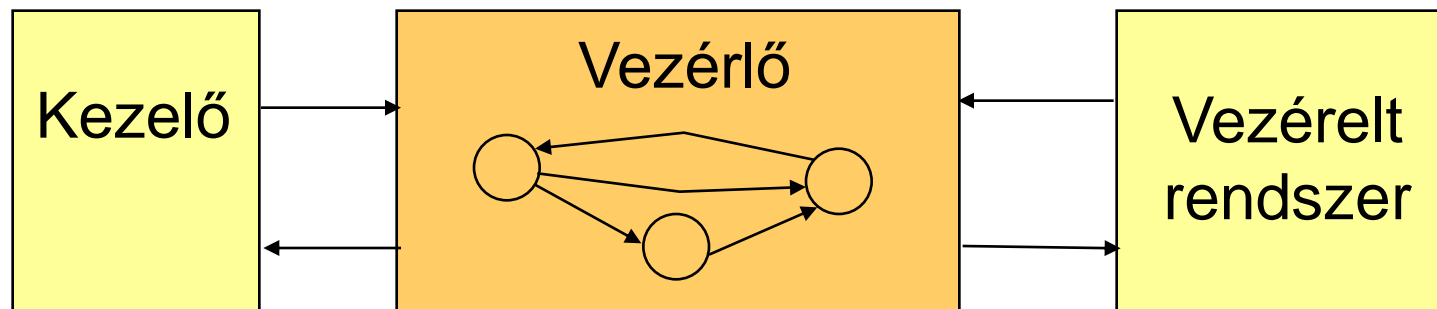
- **Teljesség**
 - Funkciók, hivatkozások, eszközök
- **Konzisztencia (ellentmondásmentesség)**
 - Külső és belső
 - Követhetőség
- **Megvalósíthatóság**
 - Erőforrások
 - Használhatóság
 - Karbantarthatóság
 - Kockázatok: költségbeli, technikai, környezeti
- **Tesztelhetőség**
 - Specifikus
 - Egyértelmű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

Példa: Vizsgálati szempontok reaktív rendszerekre (Leveson)

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



Példa: Vizsgálati szempontok reaktív rendszerekre

- Állapotdefiníció

- Bemeneti események

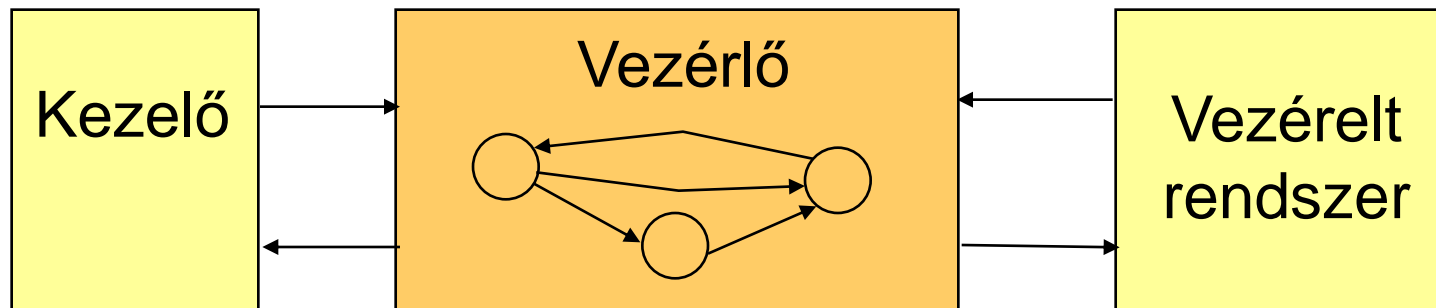
- Kimeneti események

- Kimeneti események (kimaradó bemeneti események esetén time-out van, és nincs a kimeneten akció)

- Állapot

- Ember-gép interfész

- Biztonságos a kezdőállapot
- Belső modell aktualizálva van a környezettel



Példa: Vizsgálati szempontok reaktív rendszerekre

- **Állapotdefiníció**

- **Bemenetek (események)**

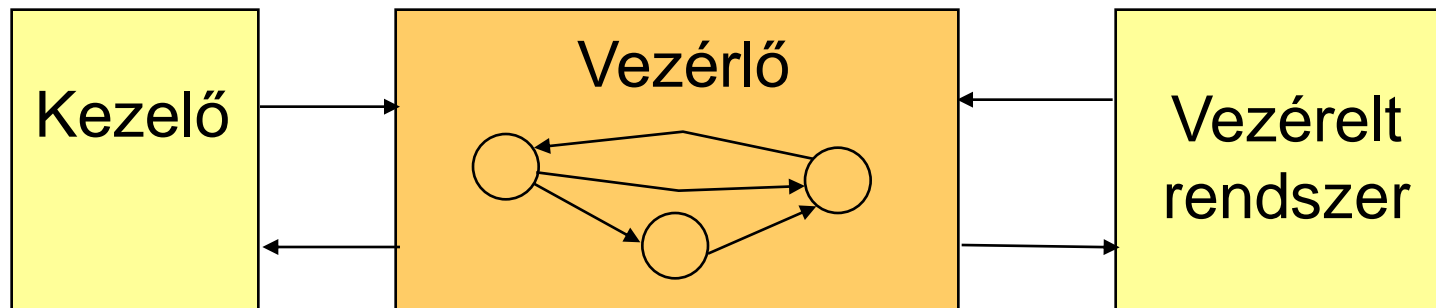
- **Kimenetek**

- **Kimenetek**

- **Állapotát**

- **Ember-gé**

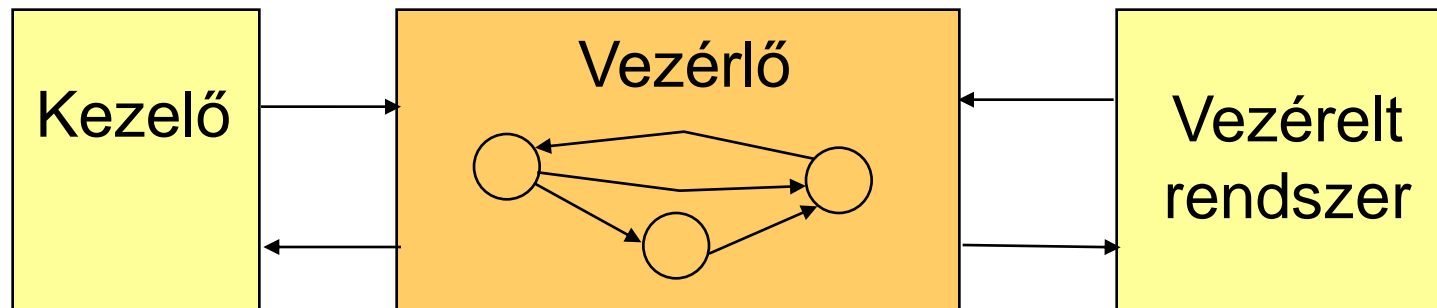
- Minden bemenetre, minden állapotban van specifikált viselkedés (reakció)
- Egyértelműek (determinisztikusak) a reakciók
- Van bemeneti ellenőrzés (értékbeli, időbeli)
- Hibás bemenet kezelése specifikálva van
- Megszakítások gyakorisága korlátozva van



Példa: Vizsgálati szempontok reaktív rendszerekre

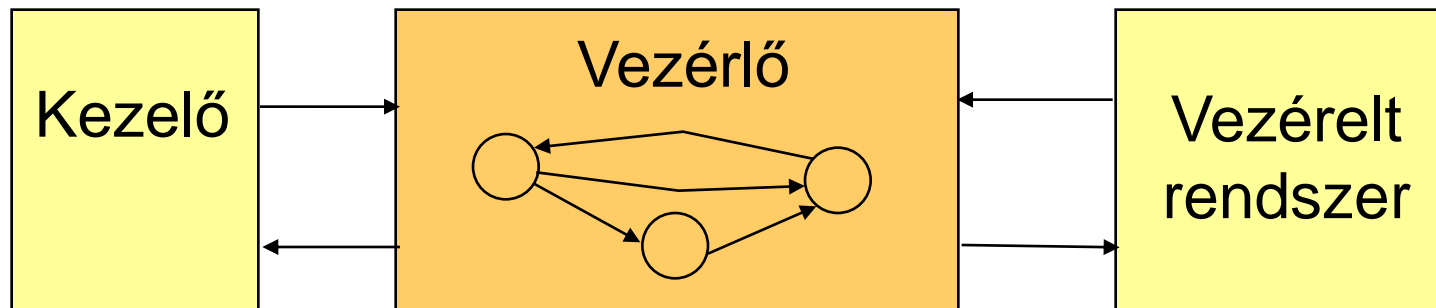
- **Állapotdefiníció**
- **Bemenetek (események)**
- **Kimenetek**
- **Kimer...**
- **Állap...**
- **Embe...**

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



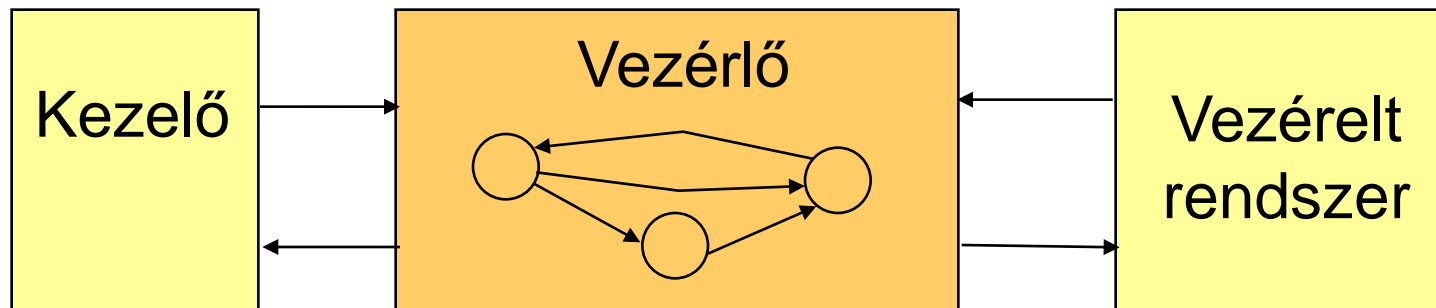
Példa: Vizsgálati szempontok reaktív rendszerekre

- **Állapotde** - Kimenetek hatása a bemeneteken keresztül ellenőrizve
- **Bemenet** - A szabályzási kör stabil
- **Kimenetek**
- Kimenetek és trigger kapcsolata
- **Állapotátmenetek**
- **Ember-gép interfész**



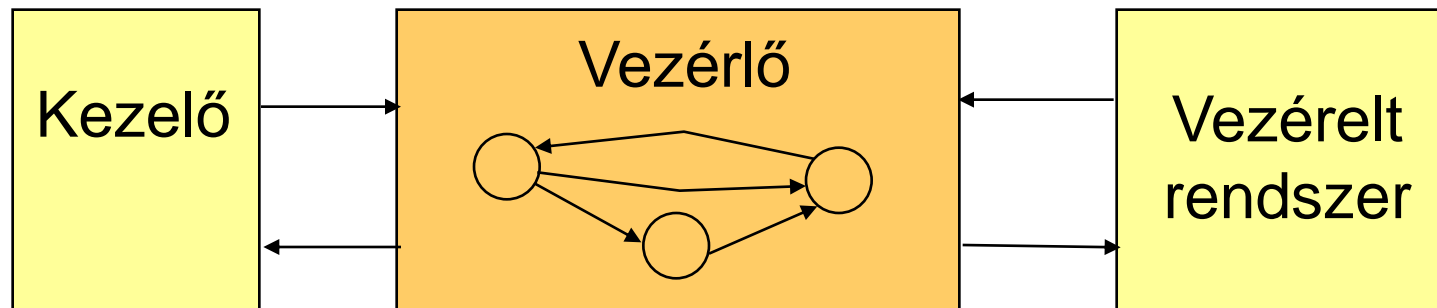
Példa: Vizsgálati szempontok reaktív rendszerekre

- **Állap** - Minden állapot elérhető statikusan
- **Be** - Állapotátmenetek visszafordíthatók (visszaút van)
- **Kin** - Több átmenet van veszélyes állapotból biztonságosba
- **Kim** - Megerősített átmenet van biztonságos állapotból veszélyes állapotba
- Állapotátmenetek
- Ember-gép interfész



Példa: Vizsgálati szempontok reaktív rendszerekre

- **Állapotr**
 - **Be**
 - **Kir**
 - **Kin**
 - **Állapotát**
 - **Ember-gép interfész**
- Kezelő felé kimenő események specifikációja:
- Sorrendezés előírt (prioritással)
 - Frissítés előírt
 - Gyakoriság korlátozott (kezelő terhelhetősége)



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
 - **Specifikus kritériumok UML állapottérképekre (mintapélda)**

Példa: IAR VisualState eszköz

A statikus ellenőrzés UML állapottérképeken:

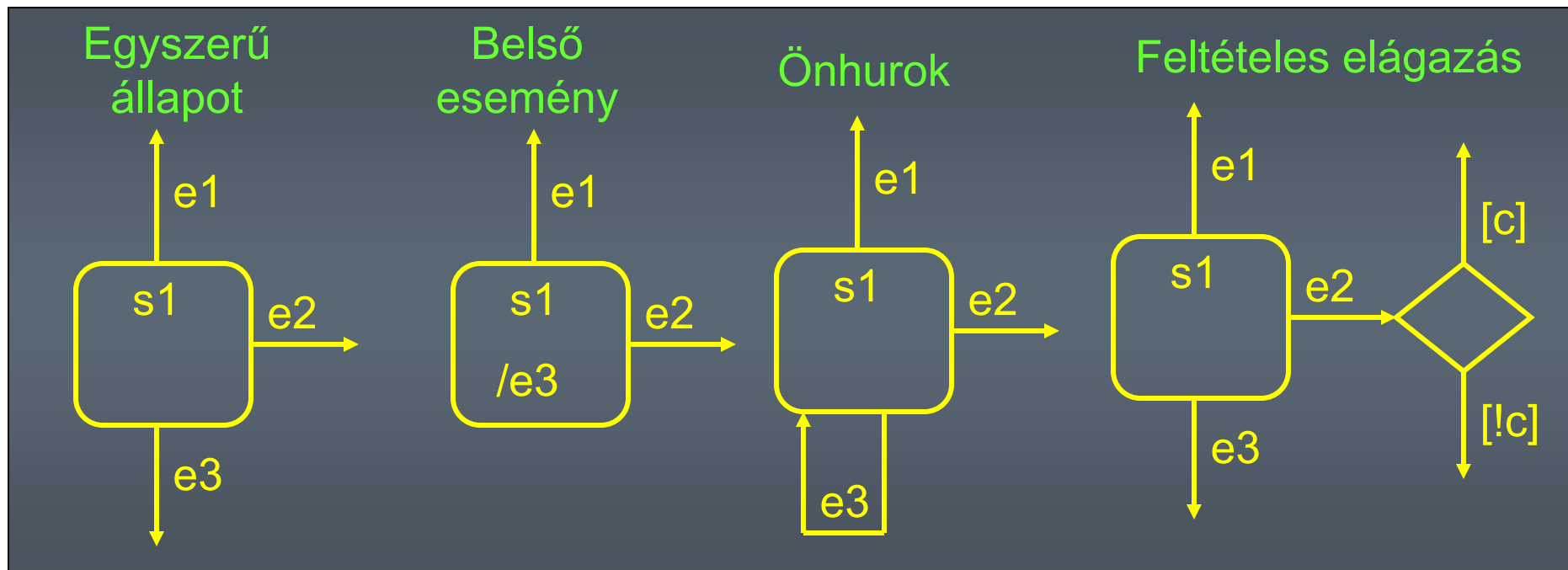
- Reset funkcionalitás
- Események, akciók felhasználása
- Állapotátmenetek statikus engedélyezettsége
- Állapotátmenet konfliktusok (azonos trigger)
- Állapotok statikus elérhetősége
- Nyelő állapot keresése (nincs kimenet)

Teljesség és ellentmondásmentesség UML állapottérképek esetén

- Hierarchikus állapot-definíció:
 - (1) Állapot helyett állapot-konfiguráció ellenőrzése kell
 - (2) A prioritásokat is figyelembe kell venni
 - Teljesség: Minden eseményre, minden állapot-konfigurációban van specifikált viselkedés (tranzíció vagy helybenmaradás)
 - Ellentmondásmentesség: Egy eseményre egy állapot-konfigurációban csak egy tranzíció lehet engedélyezett
- Konkurens régiók: Konkurens tranzíciók
 - Ellentmondásmentesség: Akciók determinisztikus végrehajtása
- 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

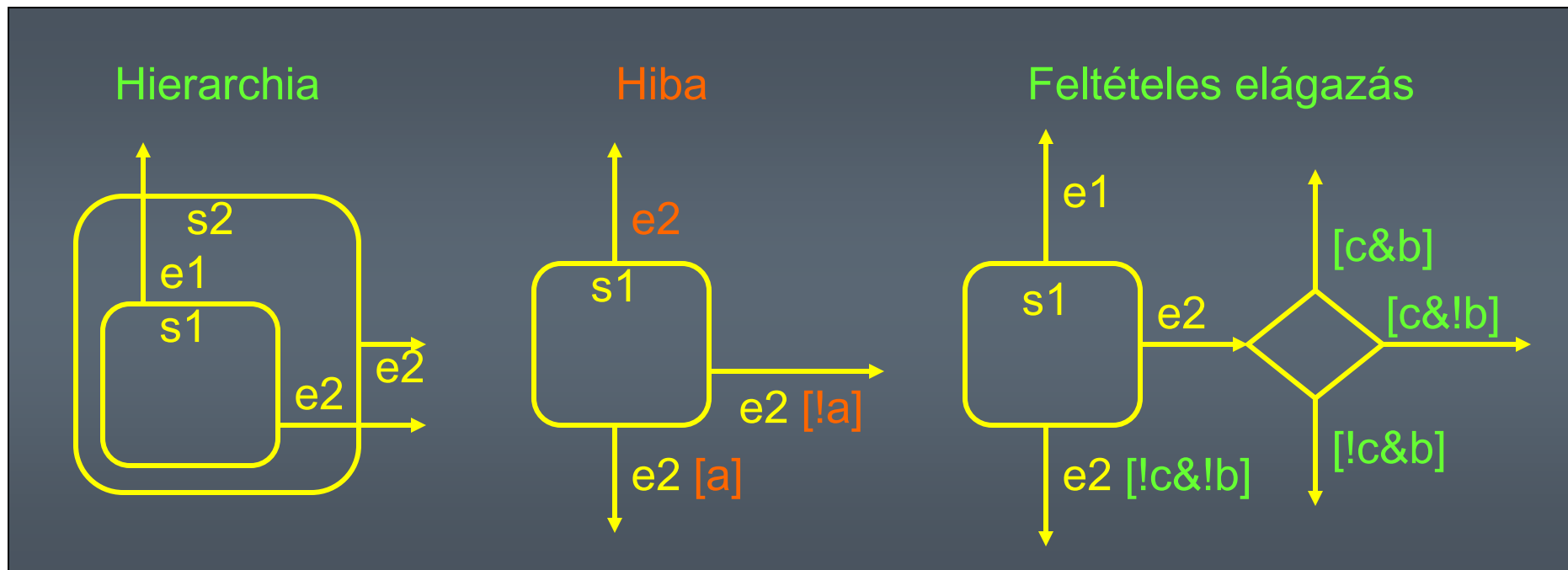
Teljesség

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



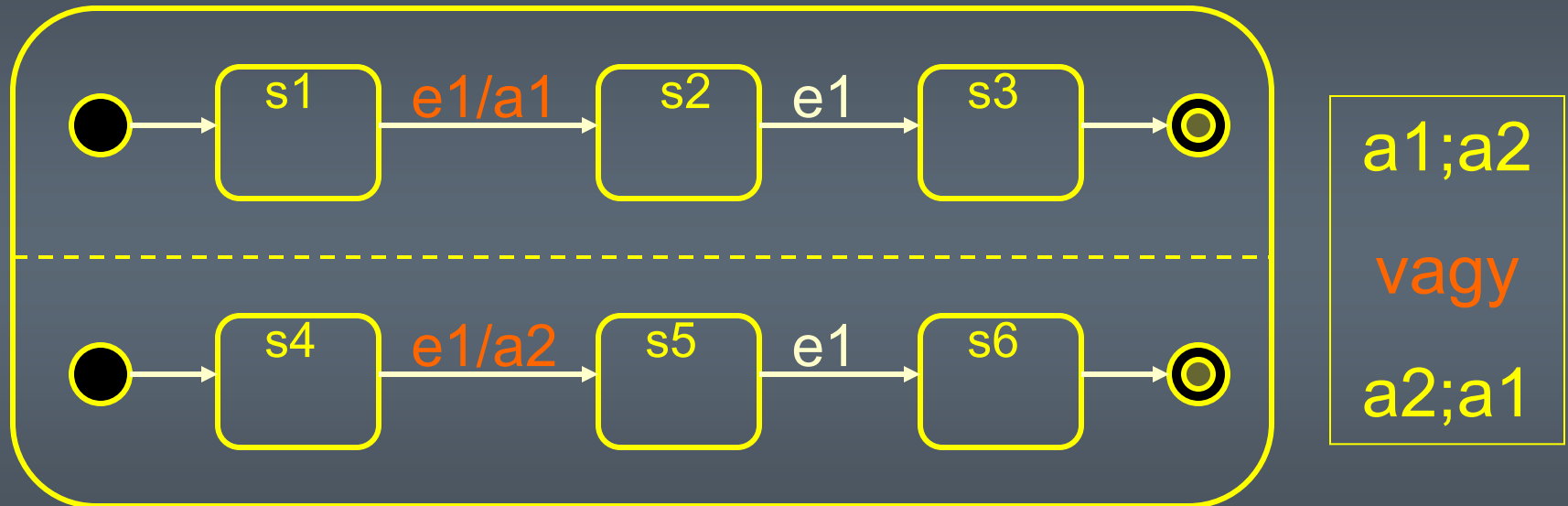
Egyértelműség I.

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



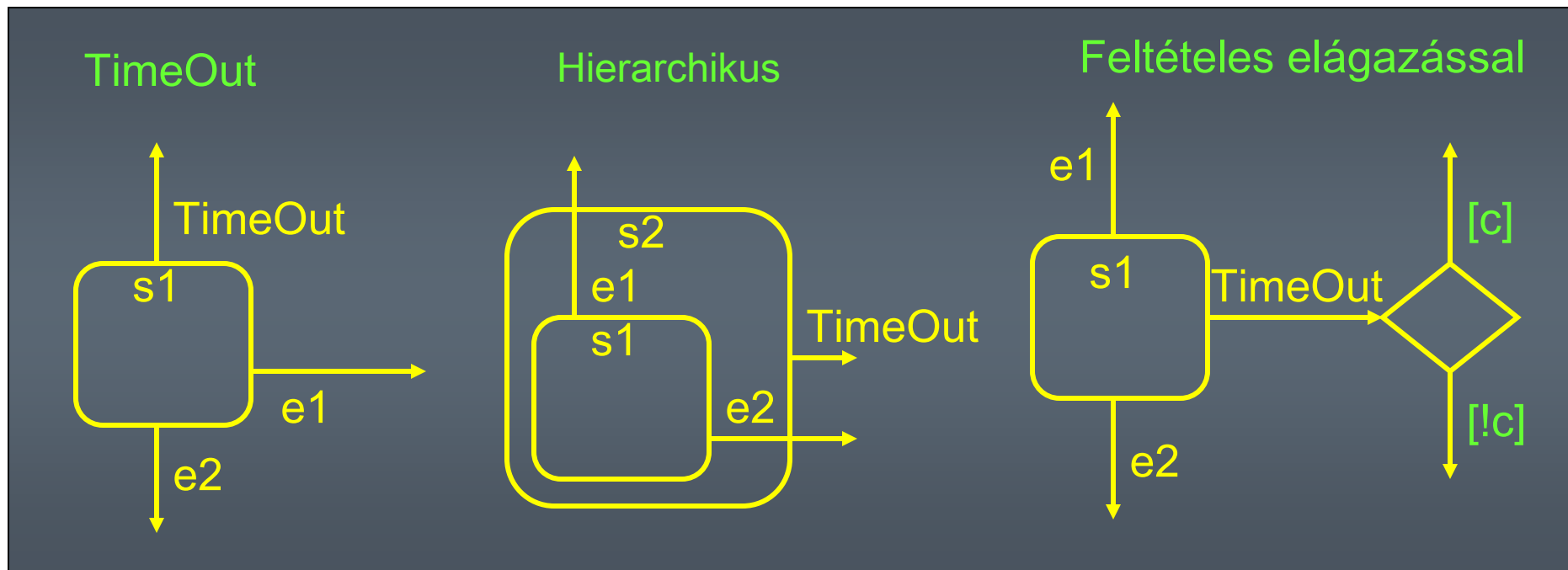
Egyértelműség II.

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



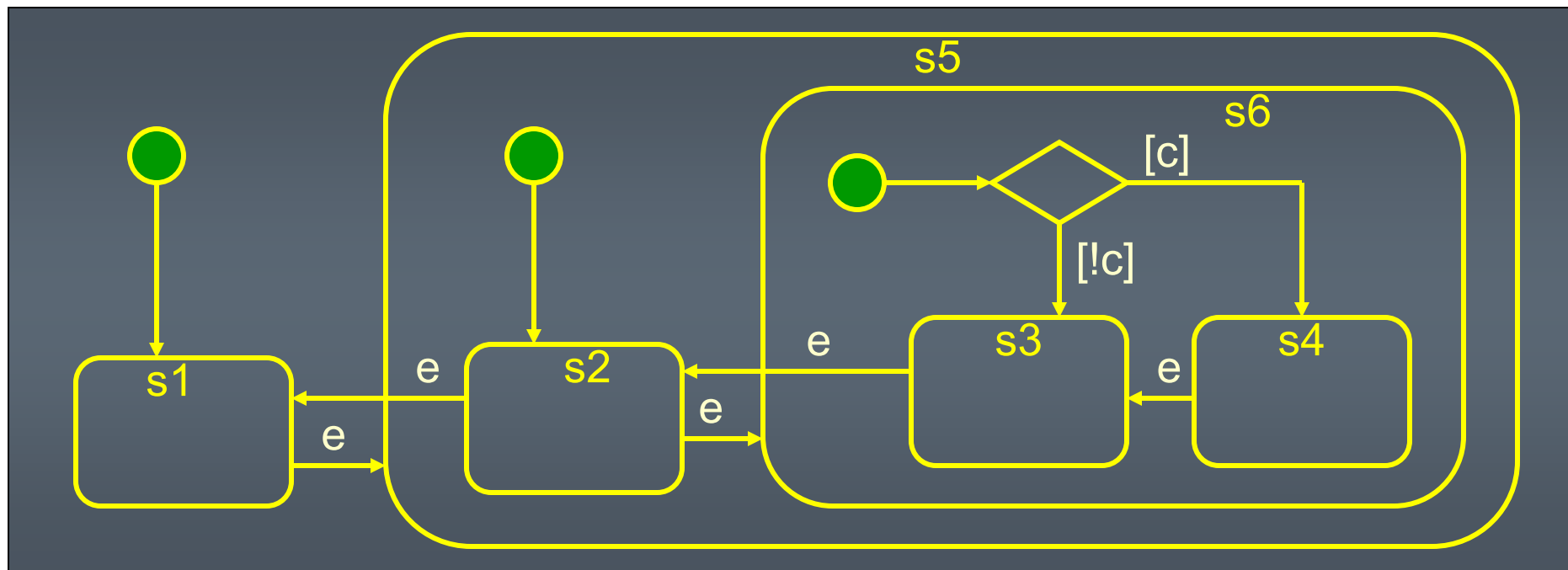
Időtűllépés

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



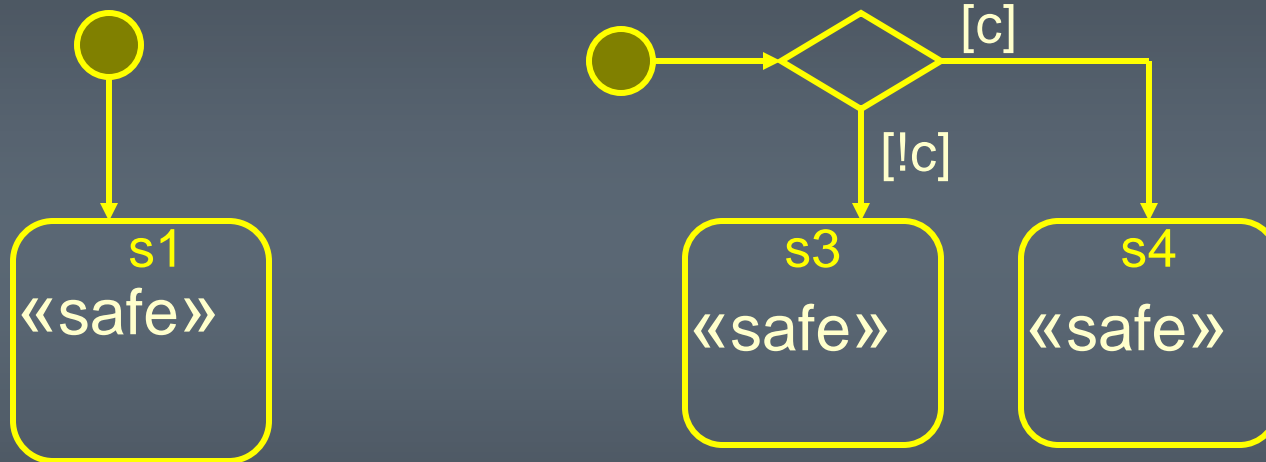
Indulási állapot I.

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



Indulási állapot II.

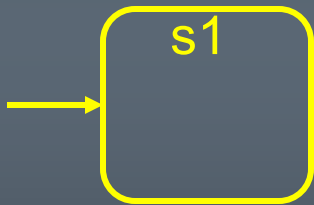
- A legfelső szintű régió kezdőállapotának biztonságosnak kell lennie:
a pseudoállapotból közvetlenül elérhető állapotoknak «safe» sztereotípiával jelöltnek kell lennie



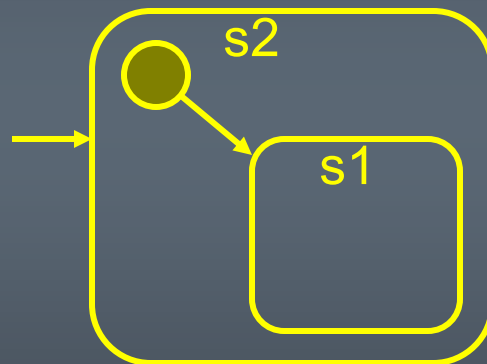
Elérhetőség

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

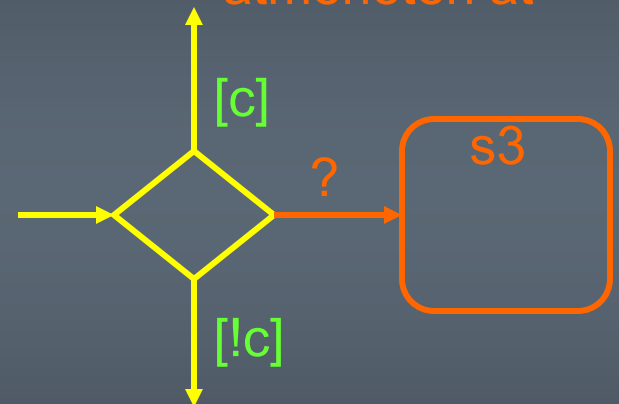
Közvetlenül



Közvetve:
hierarchián át

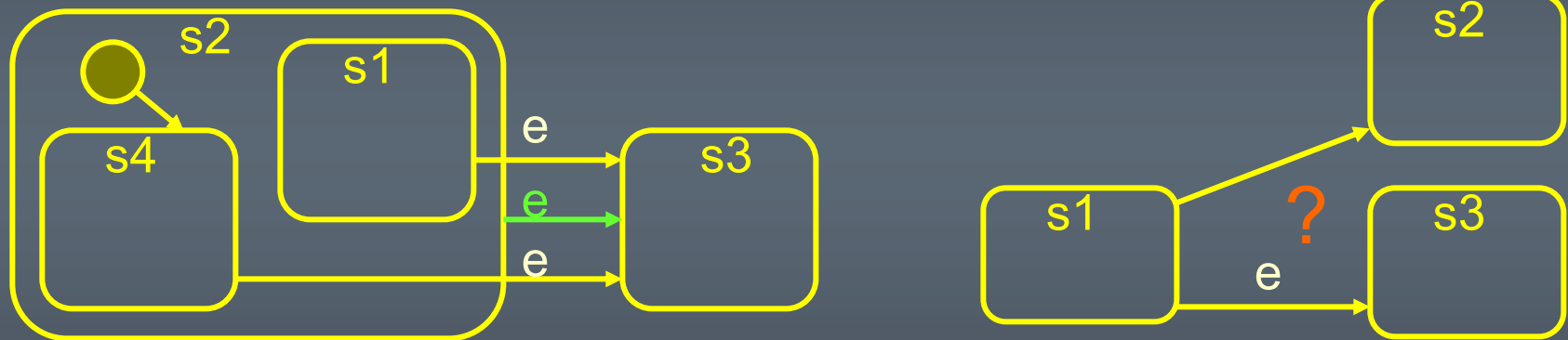


Kizárt
átmeneten át



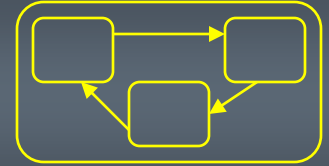
Takarás, completion

- A hierarchia miatt tranzíciók takartak lehetnek
- Completion és eseménnyel triggerelt tranzíció nem keverhető

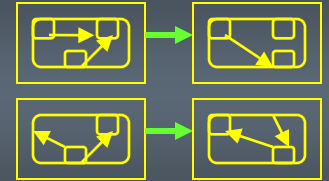


A statikus ellenőrzés menete

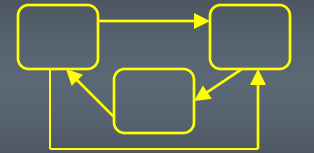
Eredeti modell



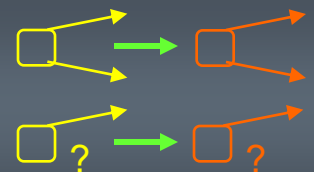
Transzformáció-sorozat



Redukált alak



Hibaminta illesztés

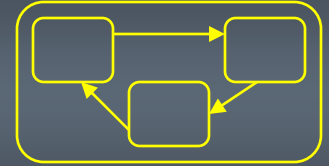


Hibaüzenet-lista

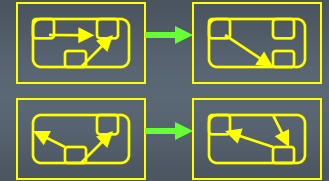
Hiba: definiálatlan
Hiba: kétértelmű...
Hiba: felesleges...
Hiba: hiányzik...
Hiba: elérhetetlen

A statikus ellenőrzés menete

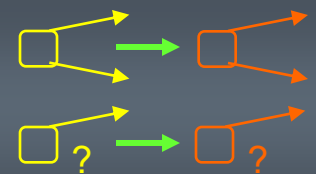
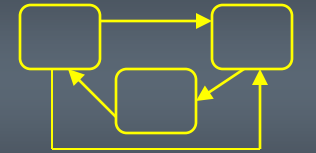
Eredeti modell



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
 - Specifikus kritériumok UML állapottérképekre (mintapélda)