

# Követelmény-specifikáció készítés és ellenőrzés

Majzik István

Budapesti Műszaki és Gazdaságtudományi Egyetem

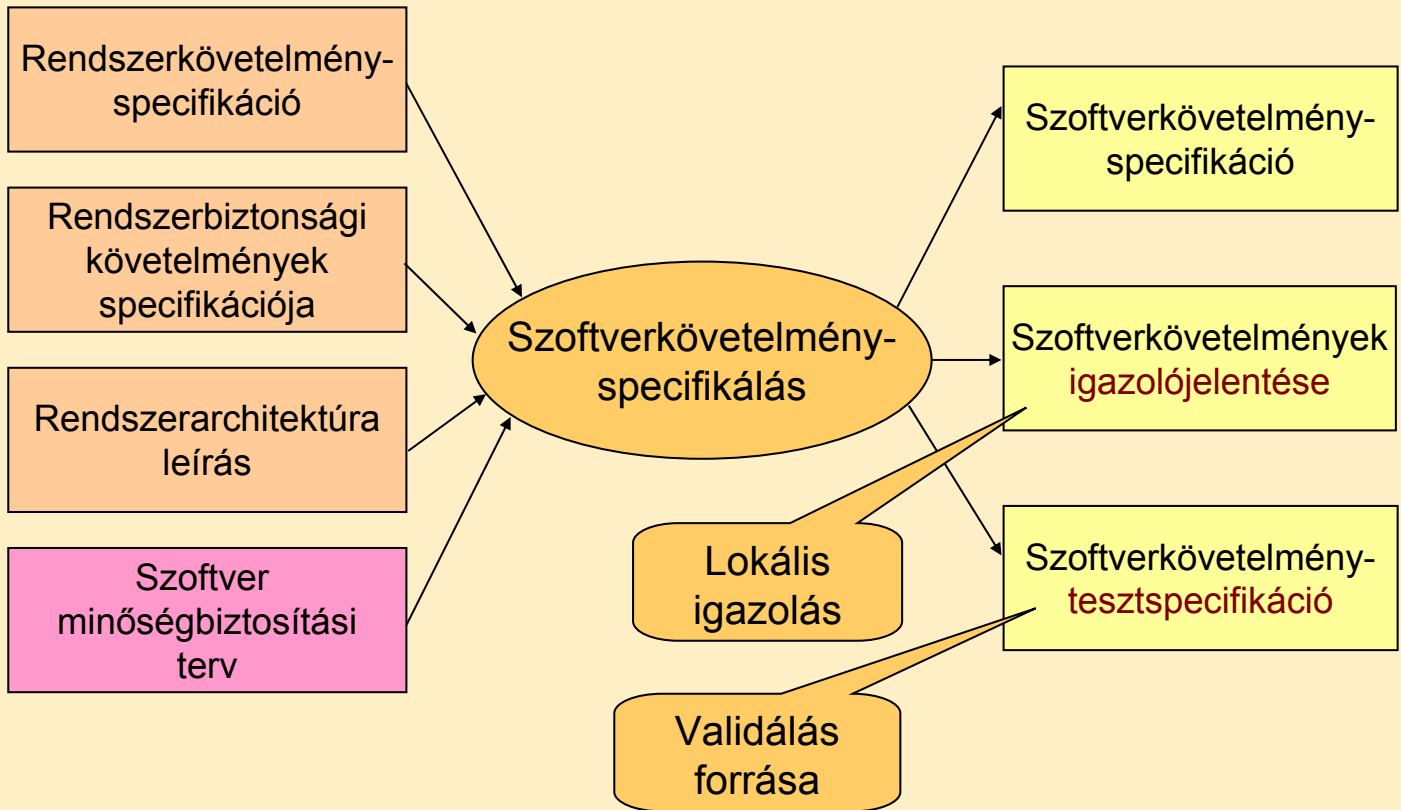
Méréstechnika és Információs Rendszerek Tanszék

<http://www.mit.bme.hu/~majzik/>

## Tartalomjegyzék

- A fázis ki- és bemenetei
- A szoftverkövetelmény-specifikáció elkészítése
  - Formális nyelvek (áttekintés)
  - Félformális és strukturált technikák
  - Példa: SysML
- Ellenőrzési feladatok
  - Általános szempontok és módszerek
  - Teljesség és ellentmondás-mentesség
- Követelménykezelés
  - Feladatok
  - Automatikus eszközök

## Kimenetek és bemenetek



## Szoftver minőségbiztosítási terv

- **Célkitűzés:**
  - Védelem a **szisztematikus hibák** ellen
  - Az összes technikai és irányítási tevékenység meghatározása, figyelése, ellenőrzése
- **Meghatározandó pontok:**
  - Tevékenységek, be/kimeneti kritériumok
  - Számszerű **minőségi elvárások** (pl. ISO/IEC 9126)
  - **Saját korszerűsítés / felülvizsgálat**
- **Külső beszállítók ellenőrzésére szolgáló eljárások**
  - Külső szállító minőségbiztosítási tervének megfelelősége
  - Szoftverek ellenőrzése, fejlesztési eszközök megfelelősége
- **Probléma bejelentés és javítás rögzítése**
  - Dokumentáció és visszajelzések
  - Elemzések (okok azonosítása)
  - Felderítés és megoldás (javítás) módszerei, megelőző tevékenységek
  - Javítás ellenőrzése, ismételt igazolás és érvényesítés

## Szoftverigazolási terv

- Biztonságkritikus rendszerekben külön terv
- Verifikációs tevékenységek tervezése
  - A szabvány szerinti választási lehetőségek kiaknázása
  - Kritériumok és eszközök meghatározása
- Általános igazolási célok:
  - „Lokális” helyesség: Teljesség, ellentmondás-mentesség
  - Megfelelőség: Az előző fázis követelményeinek
- Részletezés:
  - Eszközök (pl. teszt berendezés)
  - Eredmények kiértékelése (elfogadás feltételei), megbízhatósági követelmények kiértékelése
  - Résztvevők szerepe és felelőssége
  - Teszt lefedettség foka

## Szoftverkövetelmény-tesztspecifikáció

- Az összes követelmény igazolására illetve a kész szoftveren végzett tesztek leírására
  - Teszteléssel vizsgálható követelmény
  - Elemzéssel vizsgálható követelmény (pl. architektúra)
- Tesztesetek rögzítése:
  - Bemenetek és sorrendjük
  - Elvárt kimenetek és sorrendjük
  - Teljesítési feltételek (siker / kudarc ismérve)
- Nagy rendszereknél célszerű:
  - Felhasználói felületek forgatókönyv alapú ellenőrzése
  - Magas kockázatú elemek esetén gyors prototípus

# Szoftverkövetelmény-specifikáció

- **Követelmény (requirement):**
  - Bejövő igény, vízió, elvárás
    - Felhasználóktól (user)
    - Érdekeltektől (stakeholder: hatóság, vezetőség, operátor, ...)
  - Validáció alapja
- **Specifikáció (specification, requirement spec.):**
  - Tervezők, fejlesztők felé átalakított elvárások
  - A követelményelemzés (absztrakció, strukturálás, szűrés) eredménye
  - Sokféle típus
    - Rendszerspecifikáció, architektúra specifikáció, tervspecifikáció
  - Verifikáció alapja

## Tartalomjegyzék

- A fázis ki- és bemenetei
- **A szoftverkövetelmény-specifikáció elkészítése**
  - Formális nyelvek (áttekintés)
  - Félformális és strukturált technikák
  - Példa: SysML
- Ellenőrzési feladatok
  - Általános szempontok és módszerek
  - Teljesség és ellentmondás-mentesség
- Követelménykezelés
  - Feladatok
  - Automatikus eszközök

## Elvárások a specifikációval szemben

- A követelmények teljes lefedése
  - Funkcionális követelmények
  - Extra-funkcionális követelmények
- Megfogalmazás: Egyértelmű, igazolható, megvalósítható
- Javasolt megoldások:
  - Szigorú specifikációs nyelv (pl. formális)
  - Ellenőrzött „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!

## Formális nyelvek áttekintése

- Modell-orientált nyelvek (VDM, Z, B, ...)
- Algebrai nyelvek (ADT, OBJ, ...)
- Processz leíró nyelvek (CSP, CCS, ...)
- Logikai nyelvek (HOL, CTL\*, ...)
- Konstruktív nyelvek (NUPRL, ...)
- Hibrid illetve széles spektrumú nyelvek (CPN, E-LOTOS, ...)

## Félformális nyelvek

- **Struktúra leírás:**
  - Funkcionális blokkdiagramok
- **Adatáramlás leírása:**
  - Adatáramlási diagramok
- **Vezérlés leírása**
  - Állapotátmeneti diagram, állapottérkép
  - (Üzenet) szekvencia diagram
  - Petri-háló
- **Logikai feltételek leírása**
  - Igazságtáblázatok
  - Kényszer nyelvek (pl. OCL a struktúra alapján)

## Strukturált metodika

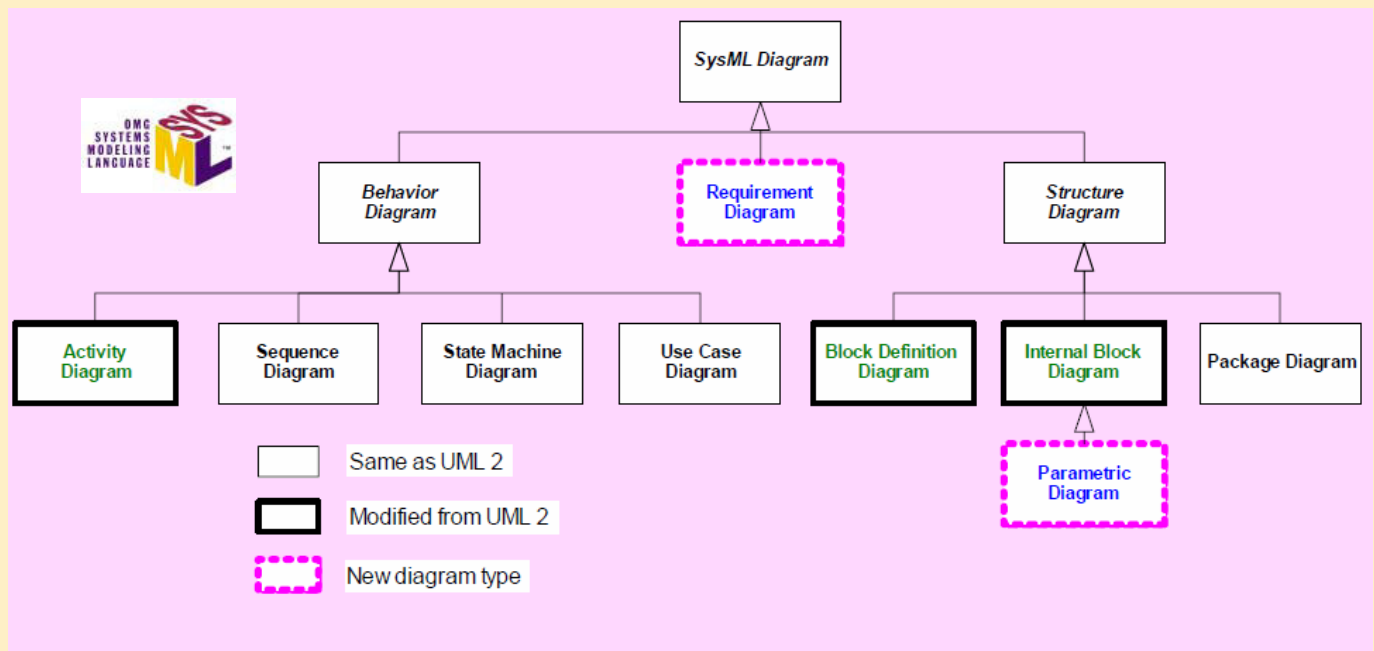
- **Jellemzők:**
  - Teljes rendszer + környezet leírása
  - Kezelhető részekre osztás (strukturáltság)
  - Intuitív, pragmatikus próbál lenni
  - Ellenőrző listák csatolhatók
  - Jelölésrendszer: Rendszerelemek (viselkedés általában informális)

## Strukturált metodika példák

- Jackson System Development (JSD)
  - Entitás struktúra: Entitások + akciók (sorrend) + processzek
  - Hálózati: Kommunikáló szekvenciális processzek
- Valósídejű Yourdon (Ward-Mellor)
  - Alapvető: Környezet (események) + viselkedés (válasz)
  - Konstrukciós: Folyamatok (processzorokon)
- SSADM
  - Logikai adatmodell (entitás relációs diagram)
  - Adatfolyam diagram (processzek, adattárak)
  - Entitás (life history) diagram
  - Entitás hatás diagram
- Structured Analysis and Design Technique (SADT)
  - Aktivitás-faktor diagram: Tevékenységek + faktor relációk; bemenet, vezérlés, erőforrás, kimenet
- ROOM: Real-Time Object-Oriented Modeling

## Félformális követelményspecifikáció: SysML

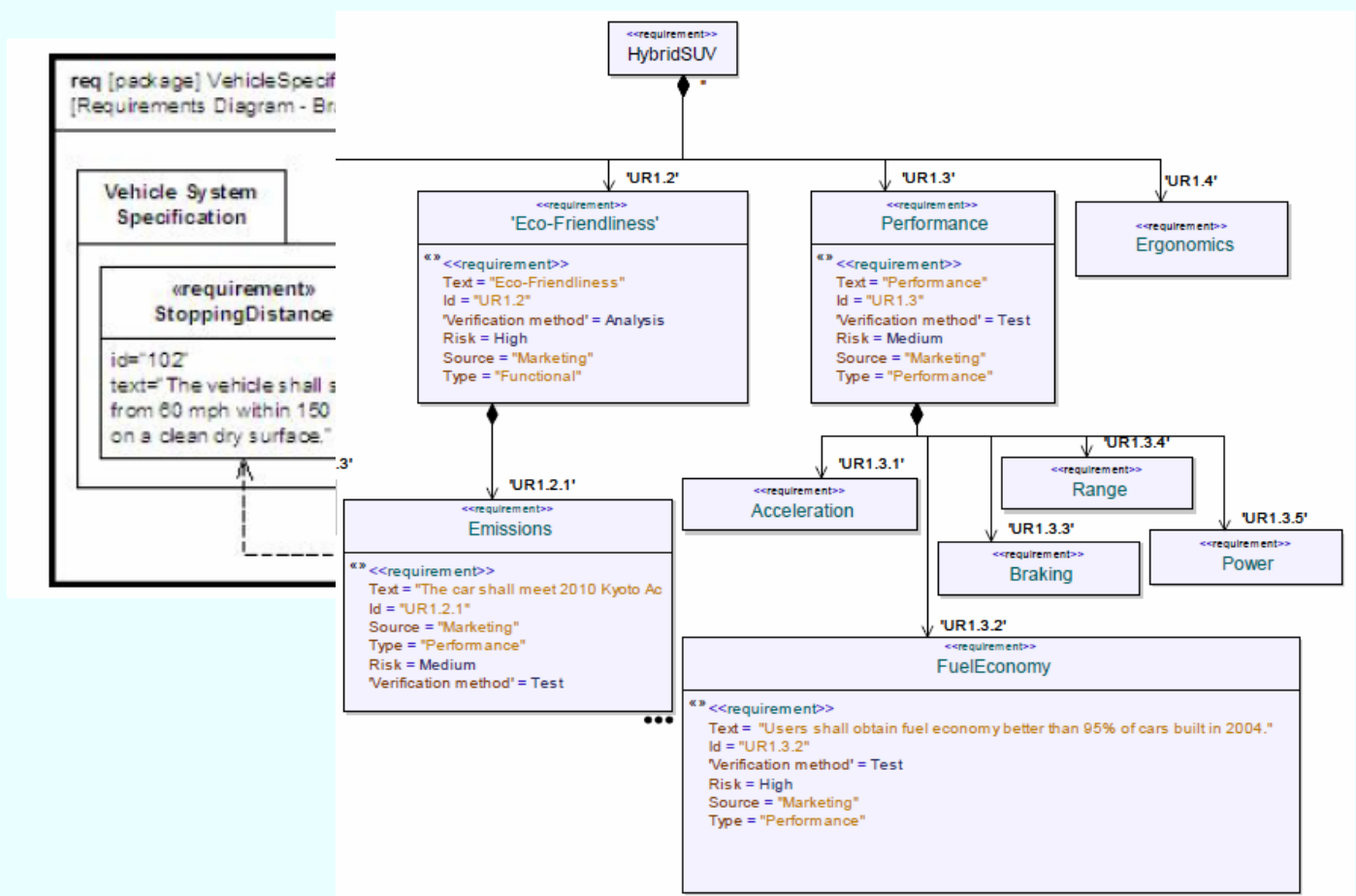
- Systems Modeling Language
  - UML részhalmoz egy kiterjesztése rendszertervezéshez
  - Fő újdonságok: Requirements és Parametric diagram



# Requirements diagram

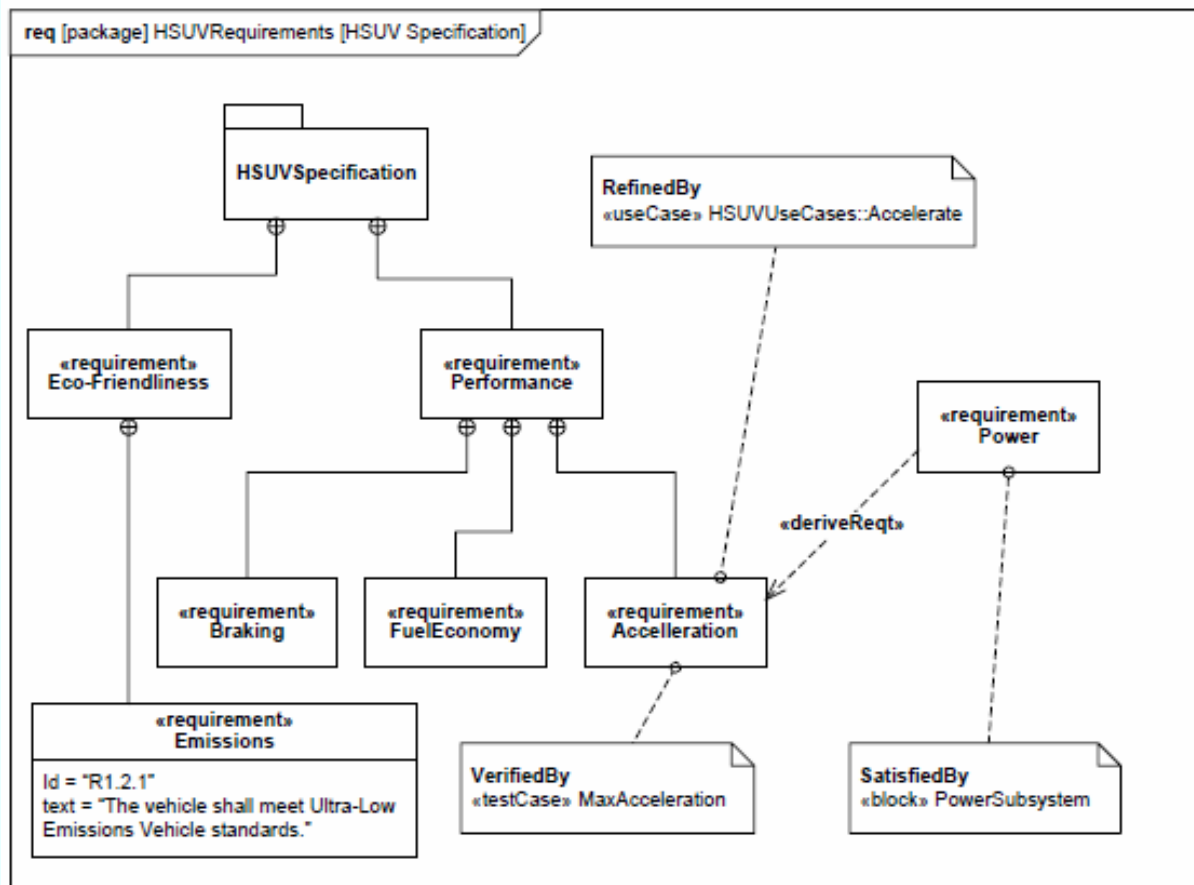
- **Követelmények (szöveges is) tárolása azonosítóval**
  - `<<requirement>>` stereotype
  - Id és text mezők
  - Felhasználói **attribútumok**: pl type, source, risk, ...
  - Táblázatos **forma** is támogatott
- **Követelmények hierarchikus csomagokba rendezhetők**
  - Funkcionális, teljesítmény, ... 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 terv vagy implementáció elemek között
  - **Verify**: követelmények és teszt elemek között

## Requirements diagram példa: Struktúra



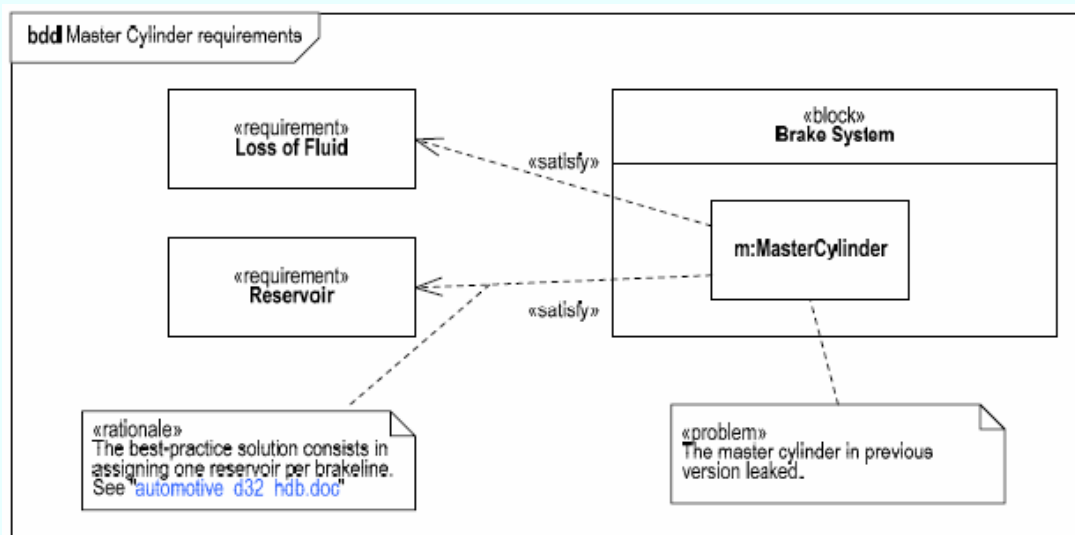


# Requirements diagram példa: Relációk



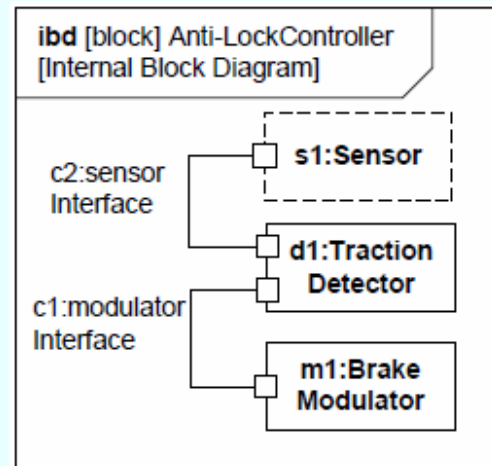
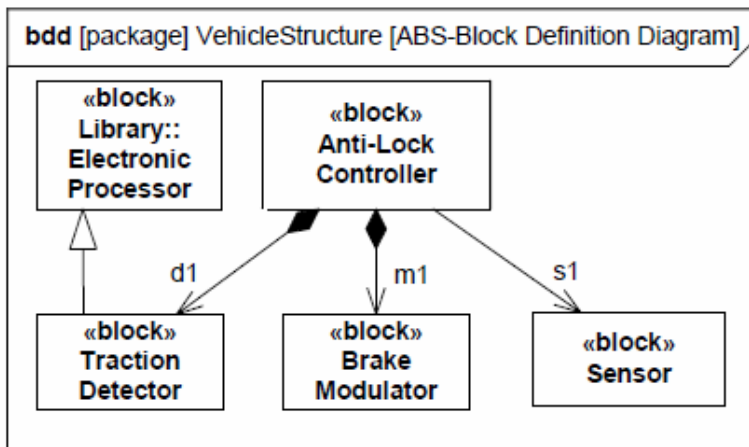
## Requirements diagram: Döntések

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



## Block diagram

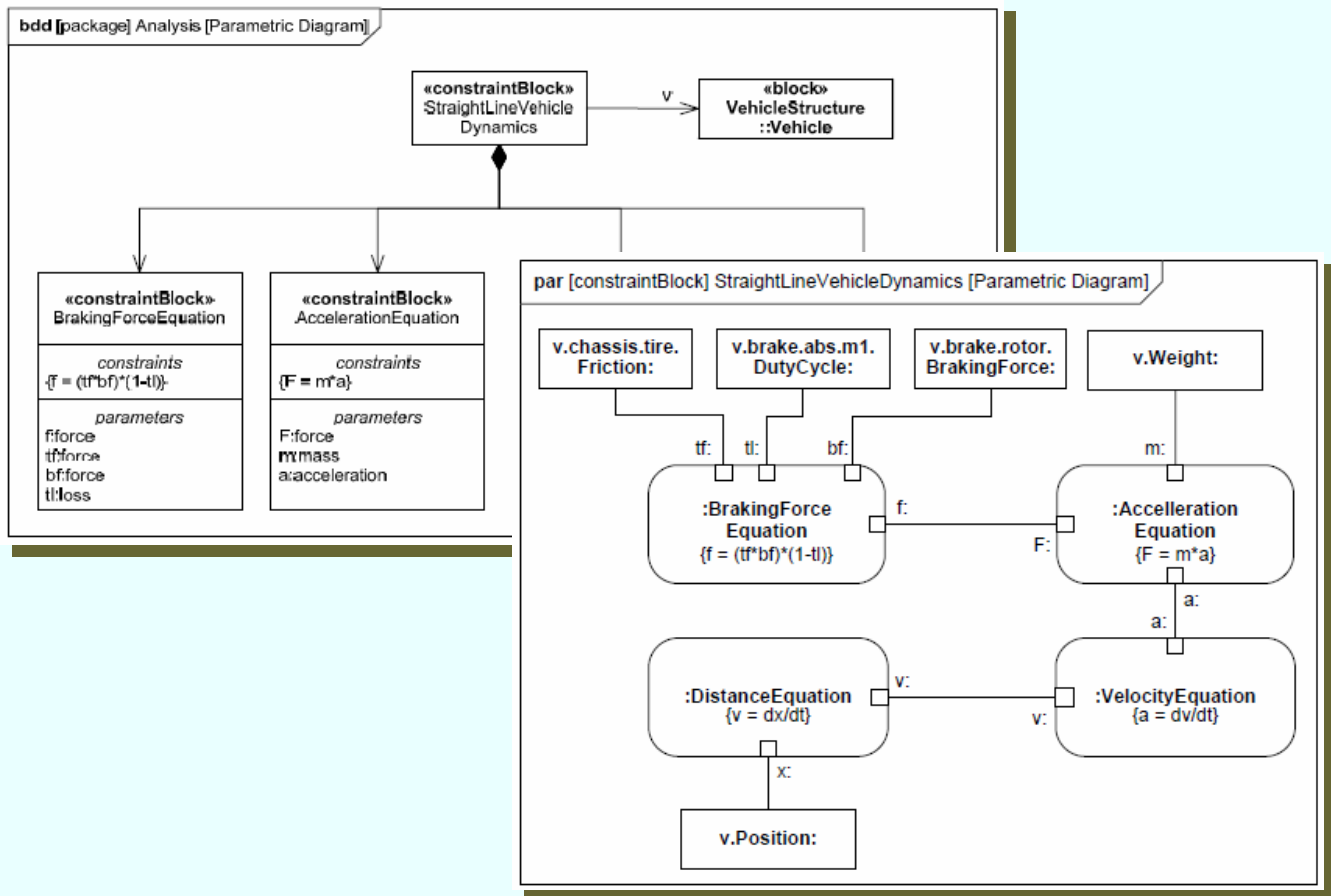
- Block: A struktúra eleme (fekete / üveg doboz)
  - Komponens (nem csak szoftver)
  - A SysML-ben az UML 2.0 osztályokon alapul
- Internal block diagram:
  - Konkrét szerepek; típust a Block adja meg



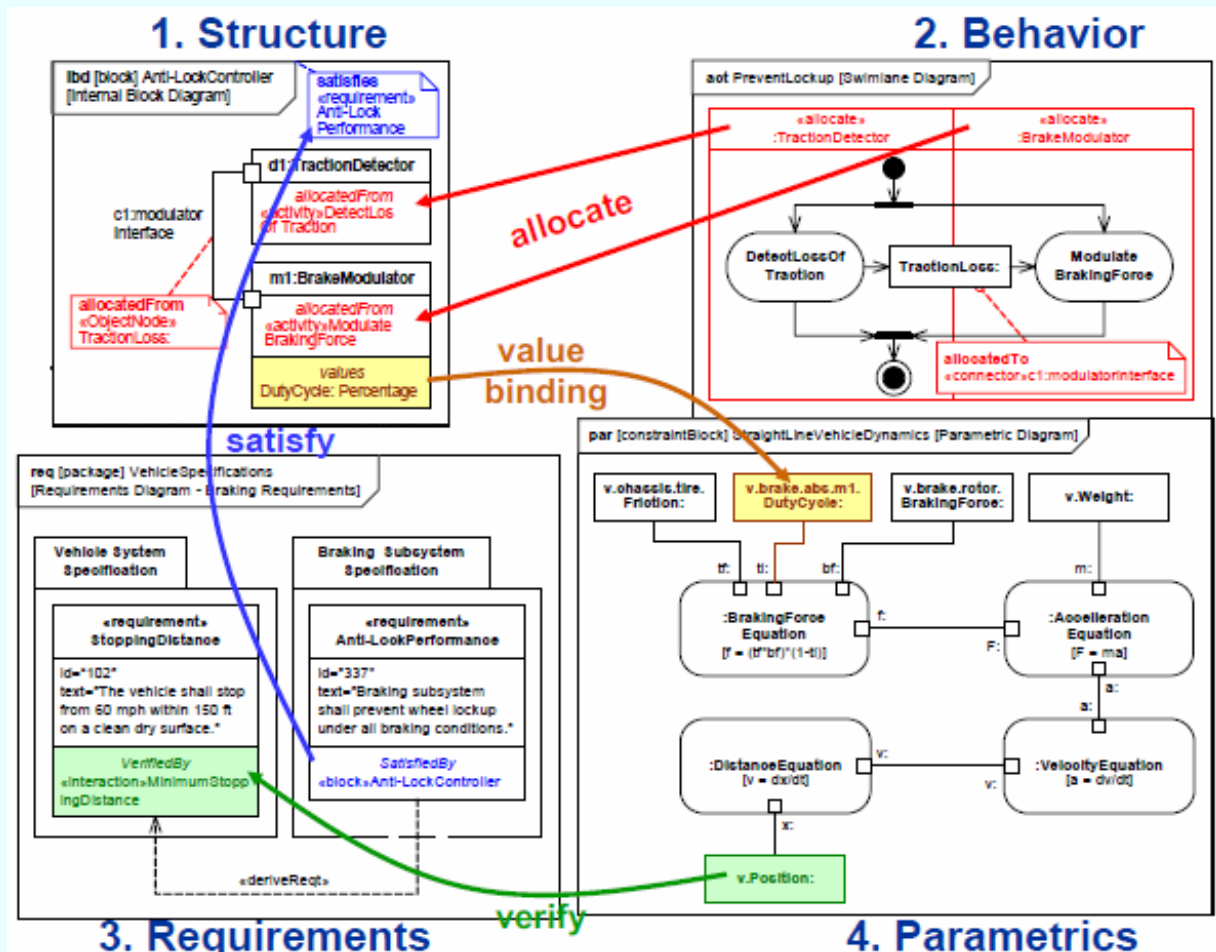
## Parametric diagram

- 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

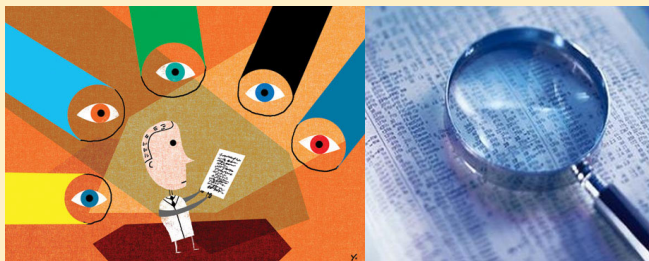


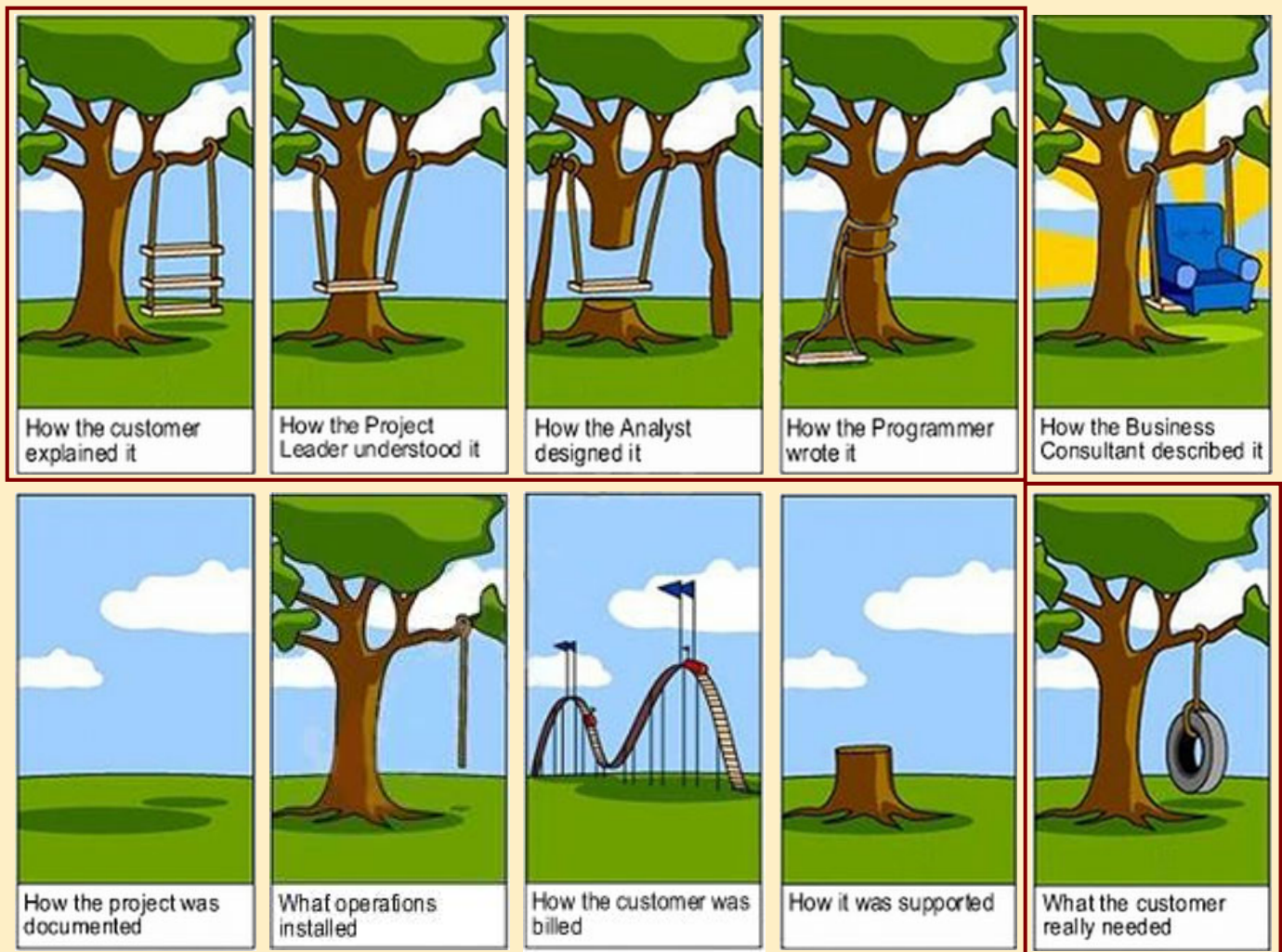
# Tartalomjegyzék

- A fázis ki- és bemenetei
- A szoftverkövetelmény-specifikáció elkészítése
  - Formális nyelvek (áttekintés)
  - Félformális és strukturált technikák
  - Példa: SysML
- Ellenőrzési feladatok
  - Általános szempontok és módszerek
  - Teljesség és ellentmondás-mentesség
- Követelménykezelés
  - Feladatok
  - Automatikus eszközök

## Verifikációs feladatok

- Szoftverkövetelmény-specifikáció ellenőrzése
  - Statikus analízis
    - Hiányosságok, ellentmondások kiszűrése végrehajtás nélkül
    - Analógia: Átolvasás „sorról sorra”
  - Ellenőrző listák
    - Tipikus hibák esetén hatékony (újra ne kövessük el)
    - Teljességet nem várhatunk
- Szoftverkövetelmény-tesztspecifikáció ellenőrzése
  - Végrehajtás (ki, mikor) és technika (hogyan) meghatározott-e?
  - Siker, kudarc ismérve adott-e?
- Megvalósítás:
  - Felülvizsgálat vagy egyenrangú átvizsgálás





## Általános ellenőrzési szempontok

- **Teljesség**
  - funkciók, hivatkozások, eszközök
- **Konzisztencia**
  - külső és belső
  - követhetőség
- **Megvalósíthatóság**
  - erőforrások
  - használhatóság
  - karbantarthatóság
  - kockázatok: költségbeli, technikai, környezeti
- **Tesztelhetőség**
  - specifikus
  - egyértelmű
  - számszerűsíthető

# A jó specifikáció jellegzetességei: IEEE Std 830-1998

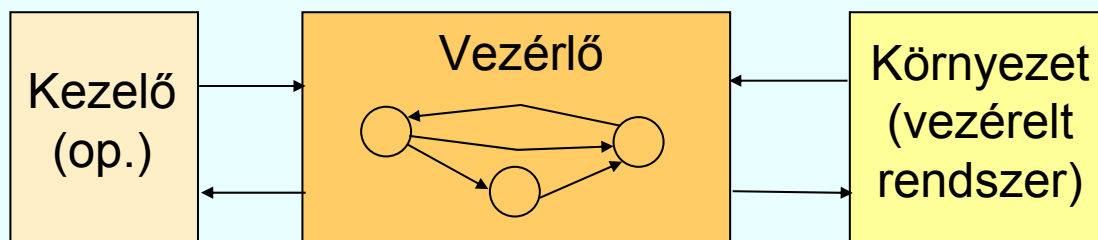
- Helyes
  - A szoftverre vonatkozó követelményeknek (elvárásoknak) megfelelő
  - Konzisztens a külső forrásokkal (pl. szabványok)
- Egyértelmű
  - Nem félreérthető, egy jelentése van
  - Hasznosak a formális, félformális specifikációs nyelvek
- Teljes
  - Minden (érvényes, érvénytelen) bemenetre van 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

## Teljesség és ellentmondás-mentesség

- Motiváció: Sok hiba visszavezethető hiányos vagy ellentmondásos specifikációra
  - Példa: Voyager és Galileo űrszondák szoftver tesztelése során felfedezett hibák statisztikája
    - 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
- Megoldás lehet:
  - Szigorú specifikációs nyelv
  - Ellenőrzött tervezési minták használata
  - Specifikáció ellenőrzése (verifikáció)

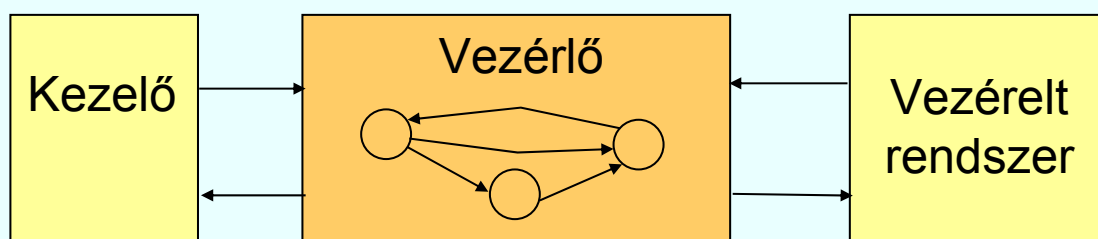
## Példa: Reaktív rsz. spec. ellenőrzése (N. Leveson)

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



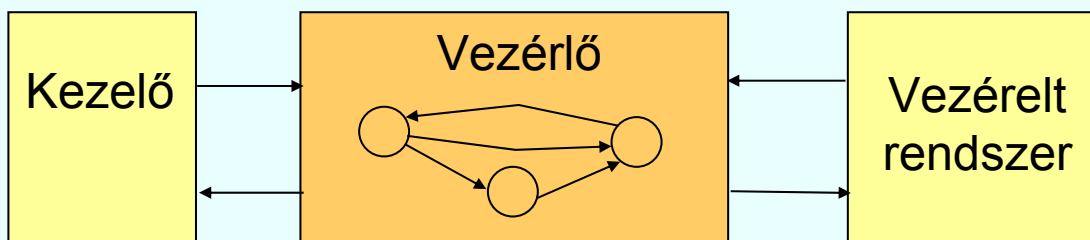
## Vizsgálati szempontok

- Állapotdefiníció
- Bemenetek (események) specifikációja
- Kimenetek specifikációja
  - Biztonságos kezdőállapot
  - Belső modell aktualizálása
- Kimenetek és trigger kapcsolata (kimaradó bemeneti események esetén time-out és nincs vezérlés)
- Állapotátmenetek specifikációja
- Ember-gép interfész



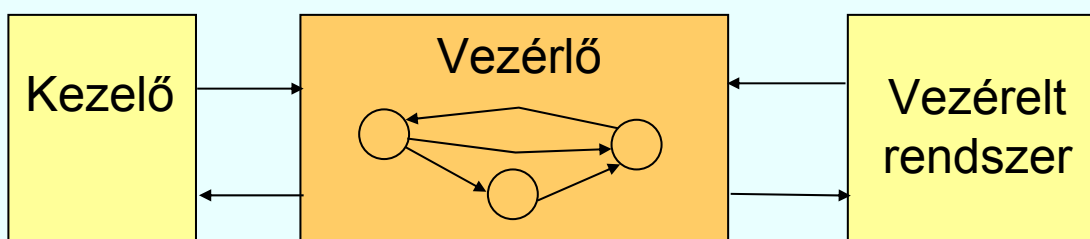
## Vizsgálati szempontok

- Állapotdefiníció
- Bemenetek (események)
- Kimenetek
- Kimenetek
  - Minden bemenetre van megadott reakció
- Kimenetek
  - Egyértelmű (determinisztikus) reakciók
- Állap
  - Van bemeneti ellenőrzés (érték, idő)
  - Hibás bemenet kezelése
- Embe
  - Megszakítások gyakorisága specifikált



## Vizsgálati szempontok

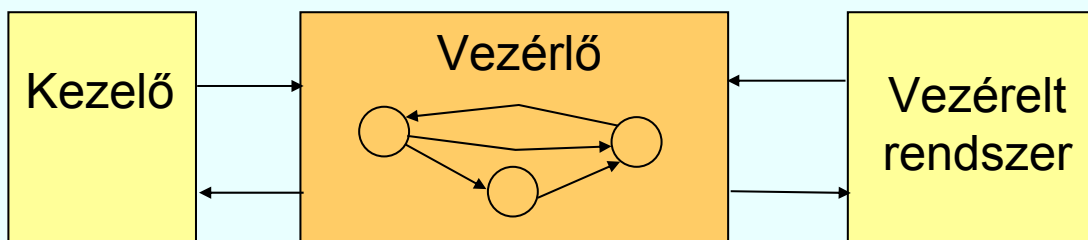
- Állapotdefiníció
- Bemenetek (események)
- Kimenetek
- Kimenetek
  - Hihetőség-vizsgálat kritériumai adottak
  - Fel nem használt kimenetek ellenőrzése
- Állap
  - Környezeti feldolgozó kapacitás betartva
- Embe





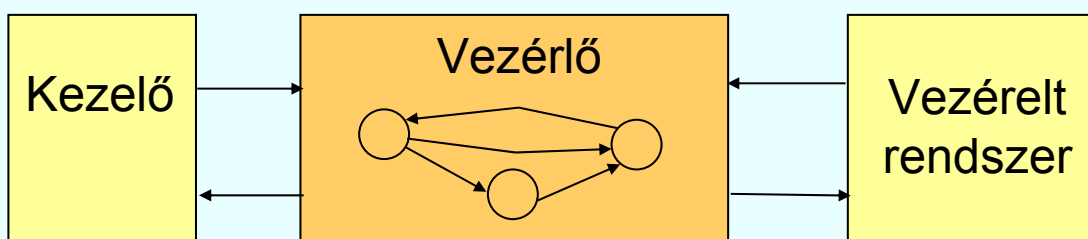
## Vizsgálati szempontok

- **Állapotdiagram** - Kimenetek hatásának ellenőrzése a bemeneteken keresztül
- **Bemenetek** - Szabályzási kör stabilitása
- **Kimenetek**
- Kimenetek és trigger kapcsolata
- **Állapotátmenetek**
- **Ember-gép interfész**



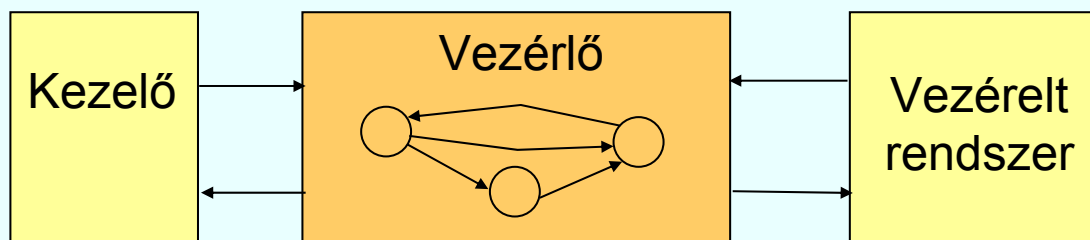
## Vizsgálati szempontok

- **Állapotdiagram** - Minden állapot statikusan elérhető
- **Bemenetek** - Állapotátmenetek visszafordíthatók
- **Kimenetek** - Több átmenet veszélyes állapotból biztonságosba
- **Kimenetek** - Megerősített átmenet biztonságosból veszélyes állapotba
- Állapotátmenetek
- **Ember-gép interfész**



## Vizsgálati szempontok

- Állapot
  - Be
  - Kir
  - Kin
  - Állapotát
  - Ember-gép interfész
- Kezelő felé kimenő események specifikációja:
- Sorrendezés (prioritás)
  - Frissítés
  - Időbeliség (kezelő terhelhetősége)



## Példa: UML állapottérkép modell vizsgálata

### Specifikációs hiányosságok ellenőrzése:

#### Teljesség:

1. Minden eseményre, minden állapotban van specifikált viselkedés (itt: **állapot-konfigurációk**)
2. Egy eseményre minden állapotban tautológiát képeznek az őrfeltételek (itt: **őrfeltétel kiértékelés**)

...

#### Ellentmondás-mentesség:

1. Egy eseményre minden állapotban csak egy átmenet lehet engedélyezett (itt: **állapot-konfigurációk**)
2. Minden állapotban van egy és csakis egy „time-out” triggerelt átmenet

...

# Tartalomjegyzék

- A fázis ki- és bemenetei
- A szoftverkövetelmény-specifikáció elkészítése
  - Formális nyelvek (áttekintés)
  - Félformális és strukturált technikák
  - Példa: SysML
- Ellenőrzési feladatok
  - Általános szempontok és módszerek
  - Teljesség és ellentmondás-mentesség
- **Követelménykezelés**
  - Feladatok
  - Automatikus eszközök

## A követelménykezelés feladatai (áttekintés)

- Követelmények hatékony, strukturált tárolása
  - Hierarchikus elrendezés, tulajdonságokkal
- Követelmények finomítása és kapcsolódása
  - Specifikáció -> Architektúra terv -> Modulterv -> Forráskód -> Teszt
- Követelmény életciklus támogatása
  - Felvétel, törlés, változás, finomítás, kapcsolatok megjelenése
- 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 / tesztelve?

Követhetőség  
(traceability)  
szükséges

## 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: Táblázatos forma
  - 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	Változások
Biztonsági megoldások	Sértetlenség

# Megvalósítás: Strukturált tárolás

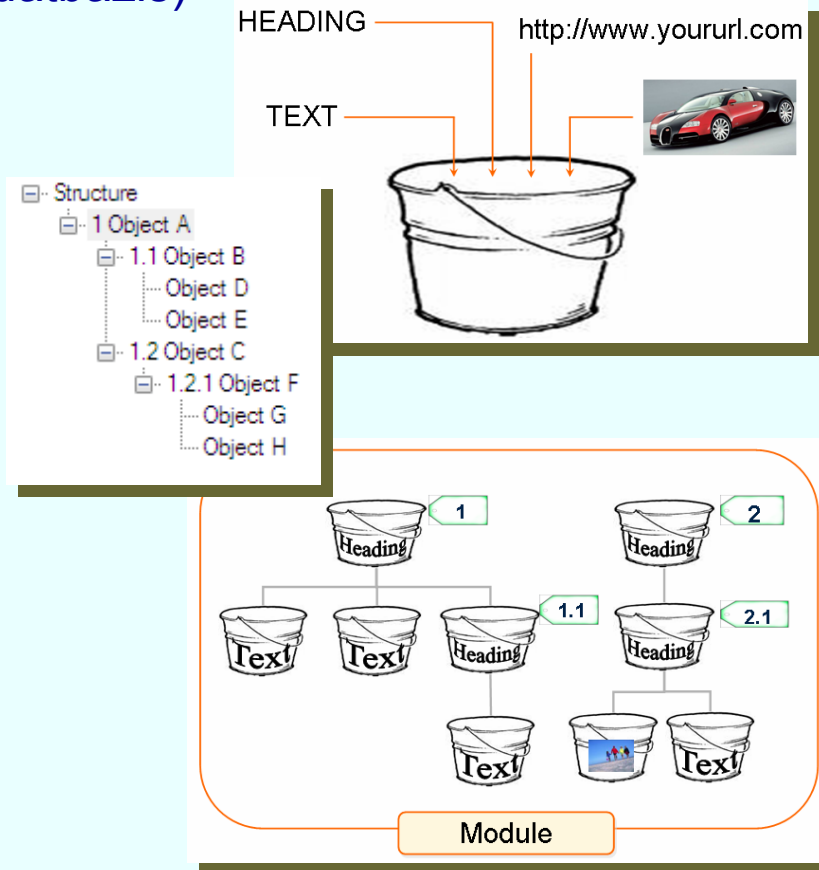
- Objektum modell (adatbázis)

- Általános „tároló”
- Egyedi azonosító
- Modulok

- Hierarchia

- Tulajdonságok

- Fejléc / szöveg
- Hozzáférési jogok
- Történet
- Attribútumok
  - Prioritás
  - Státusz
  - Költség
  - ...
- Linkek



## Példa: IBM Telelogic DOORS

ID	Last Modified By	Car user requirements	Priority	Percentage cost	Comments
TRN-CSR-1	Bill Young	<b>1 Introduction</b>	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		
TRN-CSR-3	Bill Young	<b>2 User types</b>	Desirable	1.370889	
TRN-CSR-4	Bill Young	<b>2.1 Nationalities</b>	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	<b>2.2 User sizes</b>	Mand		
TRN-CSR-7	Bill Young	People come in all shapes and sizes. The car must be suitable for people maximum and minimum sizes. ffgfg to 2 m weighing 25 kilograms to	Mand		

Objektum azonosító

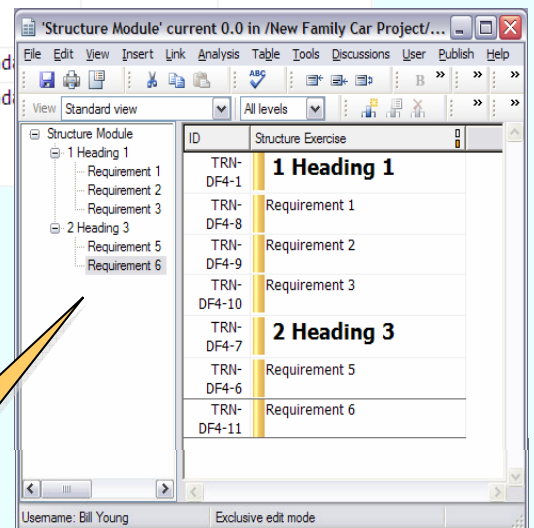
Változás-jelző

Fejléc objektum

Hierarchia

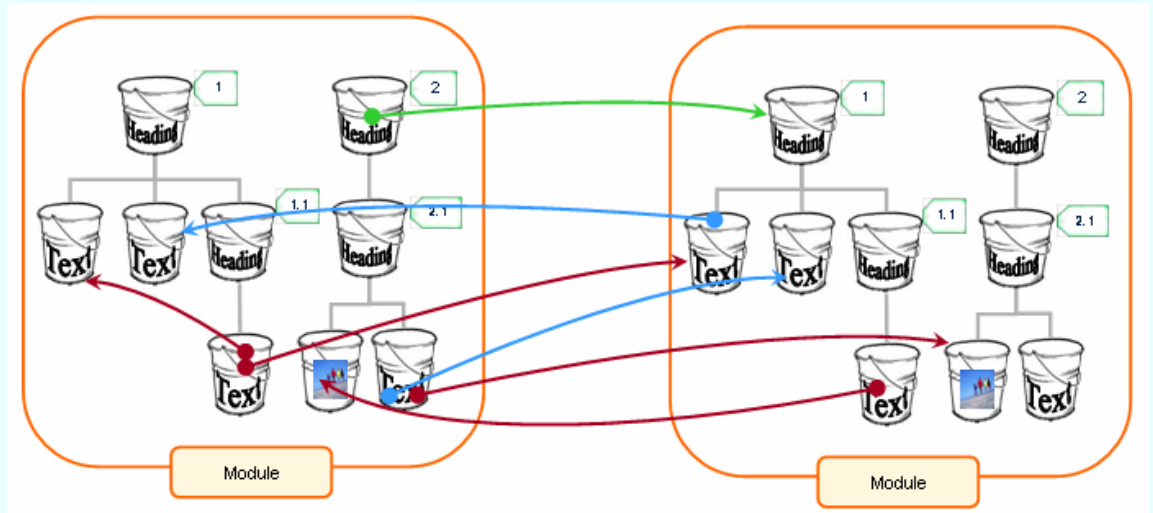
Szöveges objektum

Tulajdonságok

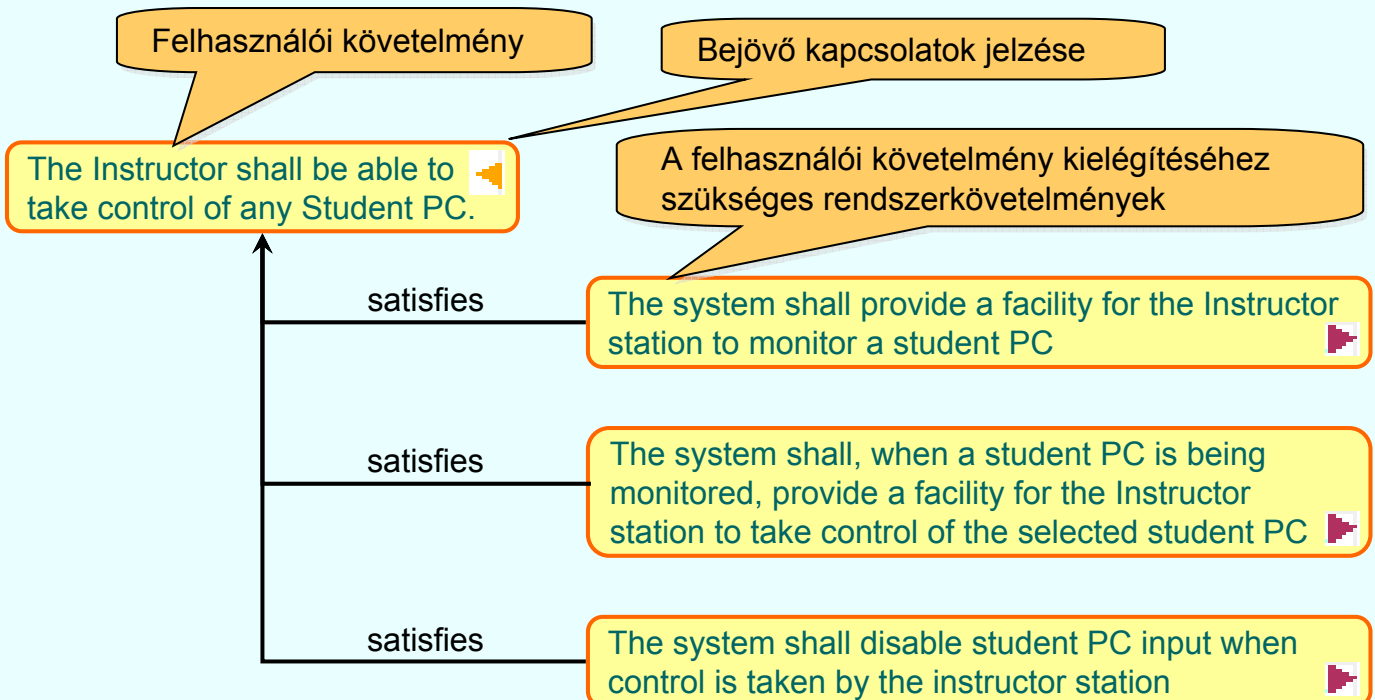


# Megvalósítás: Kapcsolatok (linkek)

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



## Példa: DOORS



## Megvalósítás: Követelmény életciklus

- **Objektum szintű változások**
  - Közvetlen szerkesztés (létrehozás, módosítás, törlés)
  - Megváltozott 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 **kijánlása**, 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 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 verifikációs eszközláncok

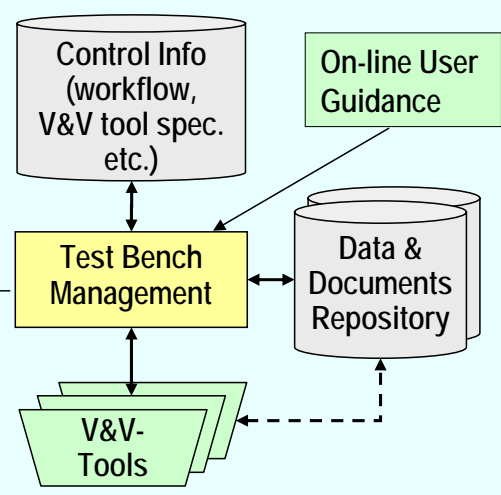
- **Verifikációs tevékenység felvétele a követelmények mellé**
  - Tervezett és rögzített verifikáció (pl. biztonságigazolás)
  - Ellenőrzések a követelmények és szabványok alapján
- **Verifikációs eszközláncok kialakítása (általában külső)**
  - **Analízis:** analízis modell generálása, analízis végrehajtása, eredmények visszacsatolása
  - **Tesztelés:** modell alapú tesztgenerálás, teszt végrehajtás, teszt eredmény kiértékelés
  - **Mérések:** mérési konfigurálása, végrehajtása, eredmények értékelése
- **Verifikációs eszközláncok indítása a követelménykezelőből**
  - Triggerek alapján; script nyelven programozható
- **Verifikáció státuszának rögzítése**
  - Ellenőrzött modell, sikeresen tesztelt követelmény
  - Eredmények tárolása: **adattár** (repository)



# Verifikáció „menedzselése” a követelménykezelővel

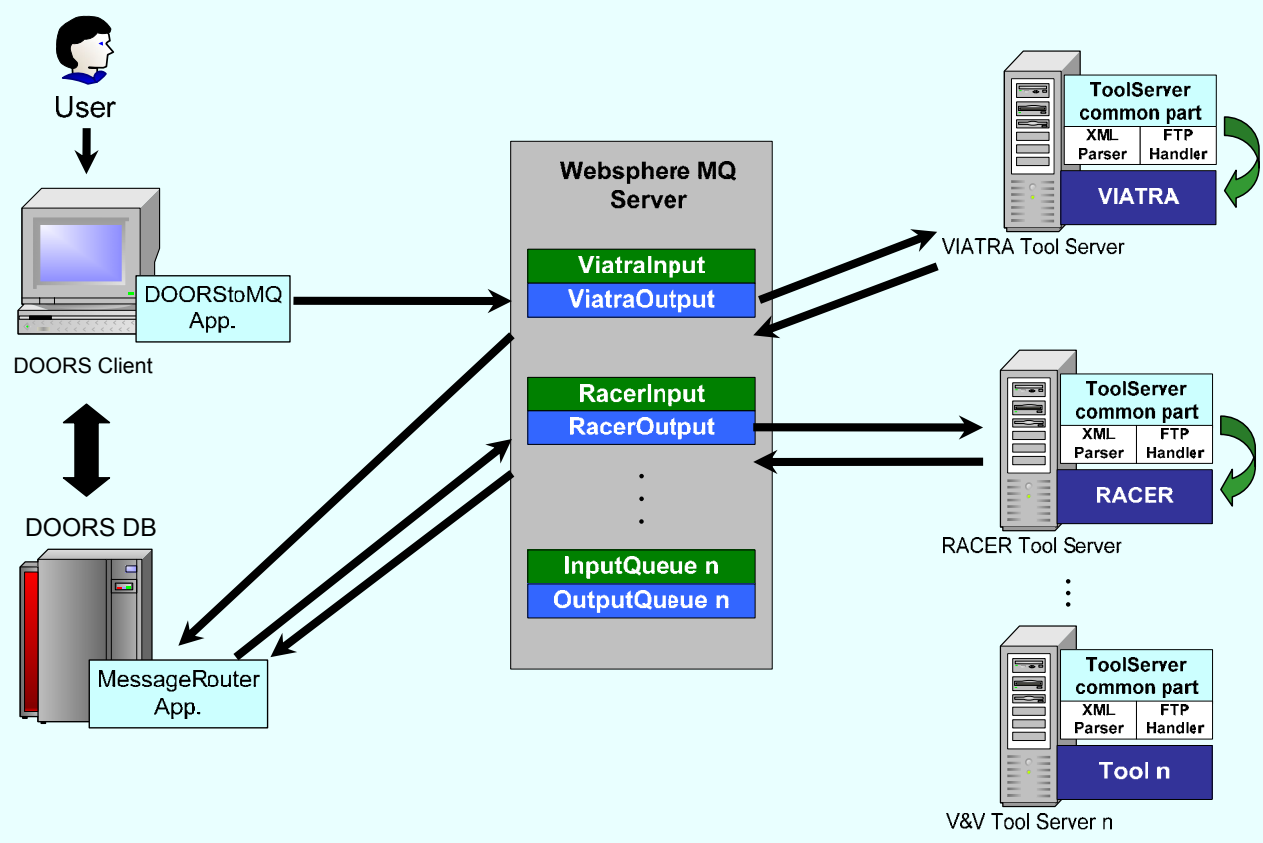
ID	Name	VWStatus	Type	Phase	VWActivity
<b>1 V-Plan Components and Middleware</b>					
VPCM1	Arch-gen-1	Not ready	Compound		
VPCM2	Arch-gen-core-1	Completed	Compound		
VPCM3	Arch-core-predictable-transport-1	Completed	Elementary		
VPCM4	Arch-core-ft-clock-sync-1	Completed	Elementary		
VPCM5	Arch-core-fault-isolation-1	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	Compound		
VPCM11	Arch-DECOS-high-level-service-1	Not ready	Compound		
VPCM12	Arch-DECOS-exec-1	Not ready	Compound		
VPCM13	Arch-DECOS-com-1	Not ready	Compound		
VPCM14	Arch-DECOS-1	Not ready	Compound		
VPCM15	Arch-DECOS-2	Not ready	Compound		

Implemented in DOORS™



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

## Példa végrehajtás: PIM ellenőrzés RACER-rel



# Egy más aspektus: Tervezői döntés adatbázis

