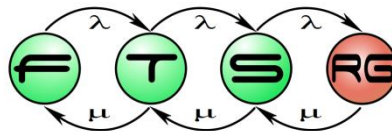


Rendszermodellezés

Modellezési alapismeretek

**Budapest University of Technology and Economics
Fault Tolerant Systems Research Group**



Tartalom

Modell és
modellezés

Mire
használunk
modelleket?

Alapfogalmak

Modellek
osztályozása

Illusztratív
példák

A
modellezés
alkalmazása

Modell és
modellezés

Mire
használunk
modelleket?

Alapfogalmak

Modellek
osztályozása

Illusztratív
példák

A
modellezés
alkalmazása

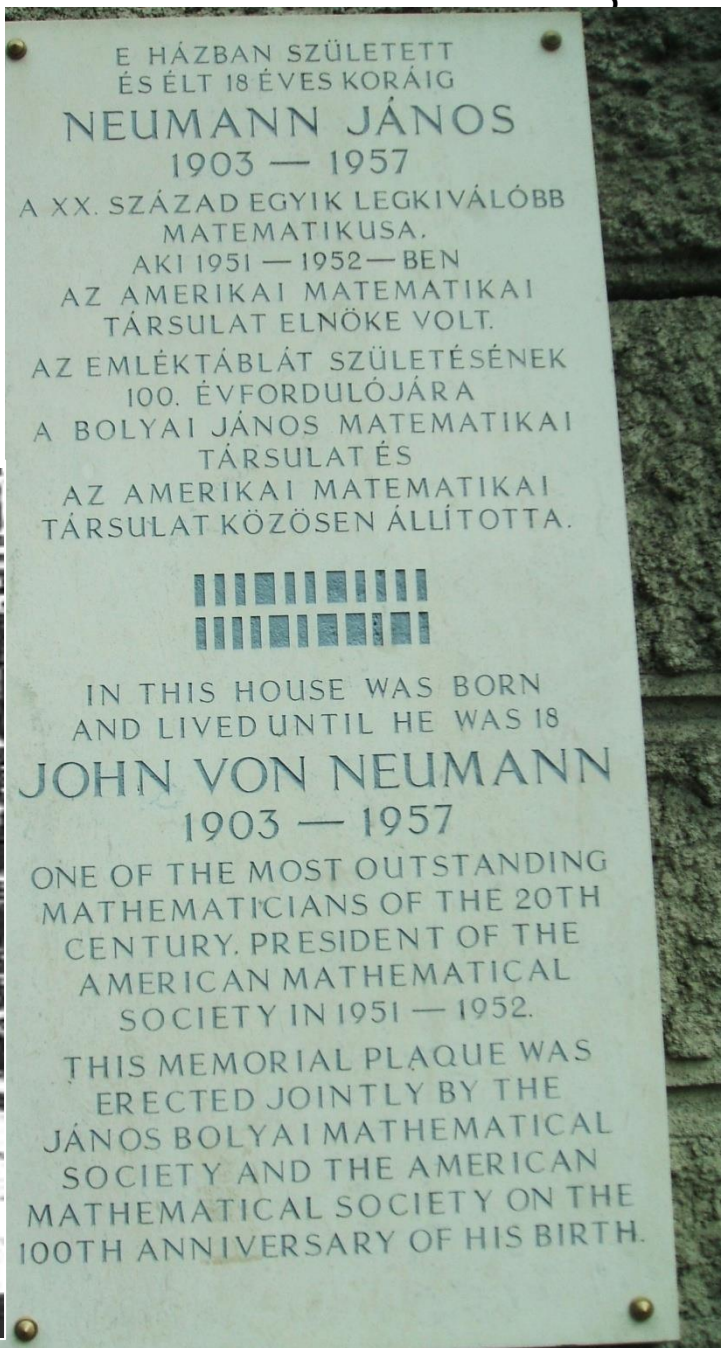
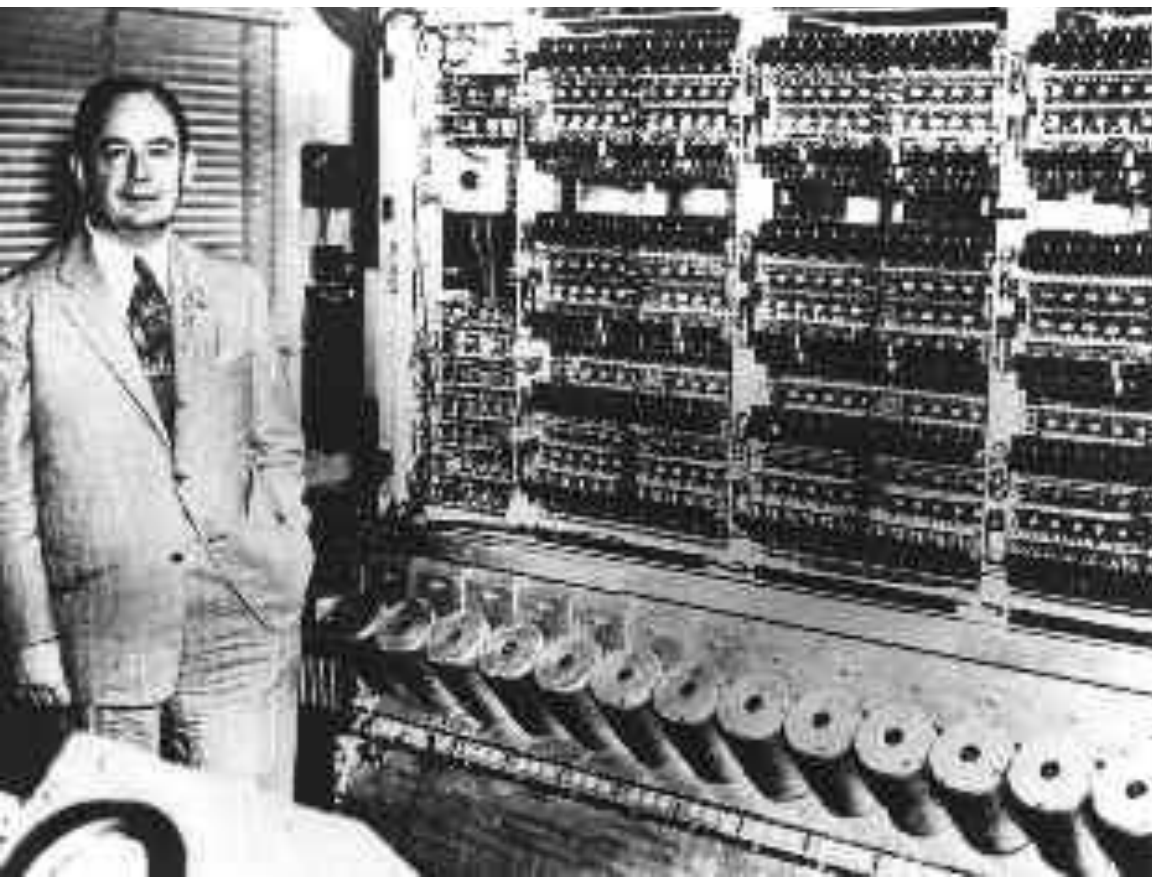
MODELL ÉS MODELLEZÉS

Mi a modell?

- "The sciences
 - do not try to explain,
 - they hardly even try to interpret,
 - they mainly make models.
- By a model is meant
 - a mathematical construct which,
 - with the addition of certain verbal interpretations,
 - describes observed phenomena.
- The justification of such a mathematical construct is solely and precisely that it is expected to work.,,

Neumann János

Birth house of John von Neumann



Mi a modell?

- Egy valós vagy hipotetikus világ (a „rendszer”) egy **részének egyszerűsített képe**, amely a rendszert **helyettesíteni tudja** bizonyos megfontolásokban

- **Döntések:**

- **A világ mely része?**
- **Mit hanyagol el?**
- **Hogy feleltethető meg a világnak?**

**Mikor lehet
és érdemes
felhasználni?**

- **Haszna**

- kisebb (véges)
- áttekinthetőbb



abcgallery.com - Internet's biggest art collection

Modell vs. valóság



Matematikai modell vs. valóság

- Minden modell:
zárt világ
 - Hatások, faktorok
 - Paraméterek
 - Érvényesség
- A modell bizonytalan,
működésű ezen a világon kívül
- Nem minden fejezhető ki előre
 - *Emberi döntés*
 - *Generált modellek*
- *Megoldás validációja*

- Normál működés
 - Peremfeltételek:
 - Van elég anyag
 - **Minden** rendelés határidőre
 - Célfüggvény:
 - Költségminimum
- Rendkívüli eset
 - Peremfeltétel
 - Anyaghiány
 - Célfüggvény:
 1. **Minél több** rendelés határidőre
 2. Költségminimum

Mi értelme van modellezni?

- Én szoftvert készítek. Kell-e modelleznem is?
 - Már így is ezt teszed!
 - (A szoftver forráskódja is egyfajta modell...)
 - Ami fontosabb: **mentális modellek**
- Mikor kell kifejezetten *lejegyezni* a modelleket?
 - Szerepe: **kommunikáció**
 - Ember → ember
 - Ember → gép
 - Gép → gép
 - Ember → saját maga, kicsivel később
 - Pl. évekkel később emlékezni kéne mérnöki döntések indokaira...

Modellezési nyelvek

- A cél a kommunikáció
 - Modell megértése szükséges
 - **Modellezési nyelvek**
- Szintaxis
 - „Matematikai struktúra”: absztrakt szintaxis
 - Jelölés: konkrét szintaxis
 - rajzjelek / szöveges formátum
- Szemantika
- Kényszerfeltételek, korlátozások
 - Szintaktikai helyesség, jólformáltság
 - Tervezési konvenciók (csapatonként változhat)

Modell és
modellezés

Mire
használunk
modelleket?

Alapfogalmak

Modellek
osztályozása

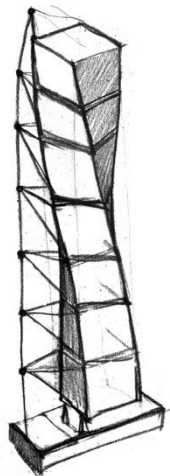
Illusztratív
példák

A
modellezés
alkalmazása

MIRE HASZNÁLUNK MODELLEKET?

Models in a design workflow

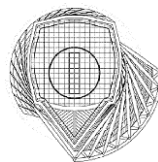
Esquiza ③ forma discoidal



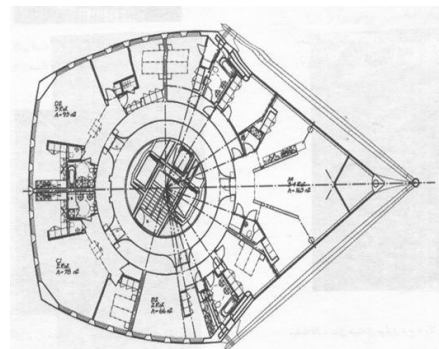
Concept



Specification



Structure



Implementation



Operation

FORMS/FORMAT 2010
Braunschweig

Rendszertervezési folyamat

Management

- Acquisition and Supply
- Planning, Assessment, Control

ANSI/EIA 632
szabványból

Mérnöki feladatok

System Design

- Requirements Definition
- Solution Definition

Product Realization

- Implementation
- Transition to Use

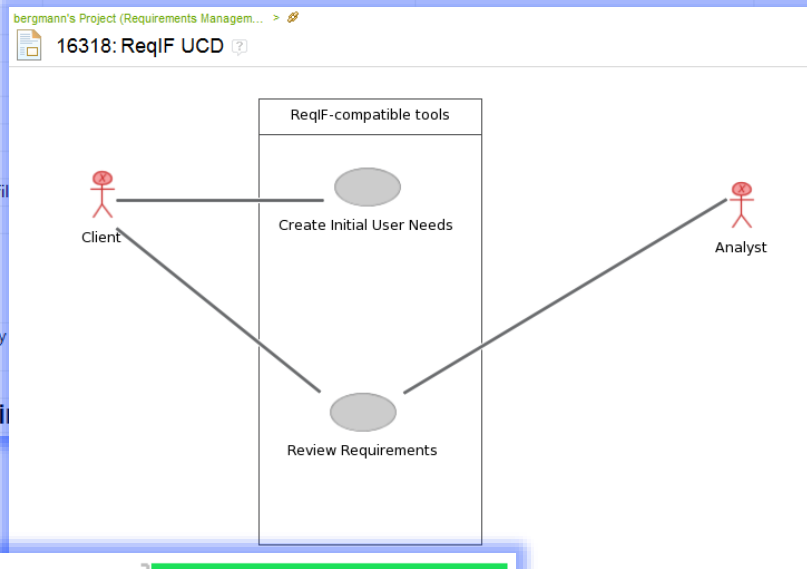
Technical Evaluation

- Systems Analysis
- Requirements Validation
- System Verification
- End Products Validation

Ezeket *tipikus*
modellekkel
támogatni

Példa: követelménymodellezés

ID	Title	Description	Rationale	Notes	Author
8	Operational Requirements				
8.1	Expected Technological Environment				
8.1.1	Support current and previous Eclipse Release	RMF shall support the current and the previous Eclipse Release	maximize the supported platforms		Mark Brörkens
8.1.2	Executable with oldest Java version required by Eclipse targets	RMF shall run with the oldest Java version that is required by the previous Eclipse release	maximize the supported platforms		Mark Brörkens
8.1.3	Executable without Eclipse Runtime	The RMF metamodel, serialization and validation shall run without Eclipse Runtime	allow usage of RMF for adding ReqIF support to Non-Eclipse Tools or simple generator scripts		Mark Brörkens
8.1.4	Execution Environment Headless Eclipse				Mark Brörkens
8.1.5	Execution Environment Eclipse with UI				Mark Brörkens
8.1.6	Focus on common EMF techniques			Avoid need for support of FeatureMaps.	
8.1.7	Enable usage in many contexts				
8.1.8	may expect validating and namespace aware parser				
8.2	Partner Applications				
8.2.1	usage with sample reflective editor				
8.2.2	usage of tool extensions with provided schema				
8.2.3	usage of tool extensions with provided emf classes				
8.2.4	usage of tool extensions with provided emf package as.ecore file				
8.2.5	XHTML as string				
8.2.6	allow usage of RMF parts in standard contexts				
8.2.7	provide set of standard EMF plugins that allow reuse and many contexts				
8.2.8	Allow alternative storages				
9	Maintainability and Portability Requirements				



```

void normalizePosition(Position* p) {
    if (p->x > 100) {
        p->x = 100;
    } if
    if ([p->y] -> implements Landing > 100) {
        p->y = 100;
    } if
} normalizePosition (function)
  
```

-> implements FasterThan100

-> implements InitialNoPoints

Mire használunk modelleket?

- Kommunikáció, dokumentáció
- Gondolkodás, tervezés támogatása
- Analízis
- Származtatás
- Szimuláció
- ...belefér egy számítógépbe / emberi agyba

Felhasználás – Dokumentáció

- A modell egyszerűbb
 - könnyebben elmondható, mint a teljes valóság
 - fokozatosan finomítható (ld. később)
- Kommunikáció, szemléltetés
 - demonstráció (ld. később)
 - érthető szöveges nyelv
 - szemléletes diagram
- Gondolkodás, tervezés támogatása
 - hasonlóak a szempontok
 - „kommunikáció magunkkal”

Felhasználás - Analízis

- Emberi erővel vagy (részben) automatizáltan
- Módszer
 - Felületes, statikus elemzés
 - Dinamikus állapottér bejárással – modellellenőrzés
 - Formális állítások bizonyításával
- Cél
 - Ellenőrzés, hibák keresése (best effort)
 - Szolgáltatásbiztonsági kritériumok igazolása (erősebb!)
 - Jellemzők számítása, tervezése (pl. ütemezés)

Felhasználás - Származtatás

- Emberi erővel vagy (részben) automatizáltan
- Eredmény
 - programkód, analizálható nyelv, stb. generálása
 - másik modell
 - finomítás, következő tervezési fázis
 - részaspektus
 - modellek integrációja
- Lehet tulajdonságmegőrző

Felhasználás - Szimuláció

- Validáció
 - „Jól építettem fel?”
- Demonstráció
 - A kommunikáció eszközeként
- Kísérlet
 - Tulajdonságok elemzésére
 - Mérések
 - A valóságban költségesen kipróbálható
 - Elméleti úton előre meg nem határozható

Modell és
modellezés

Mire
használunk
modelleket?

Alapfogalmak

Modellek
osztályozása

Illusztratív
példák

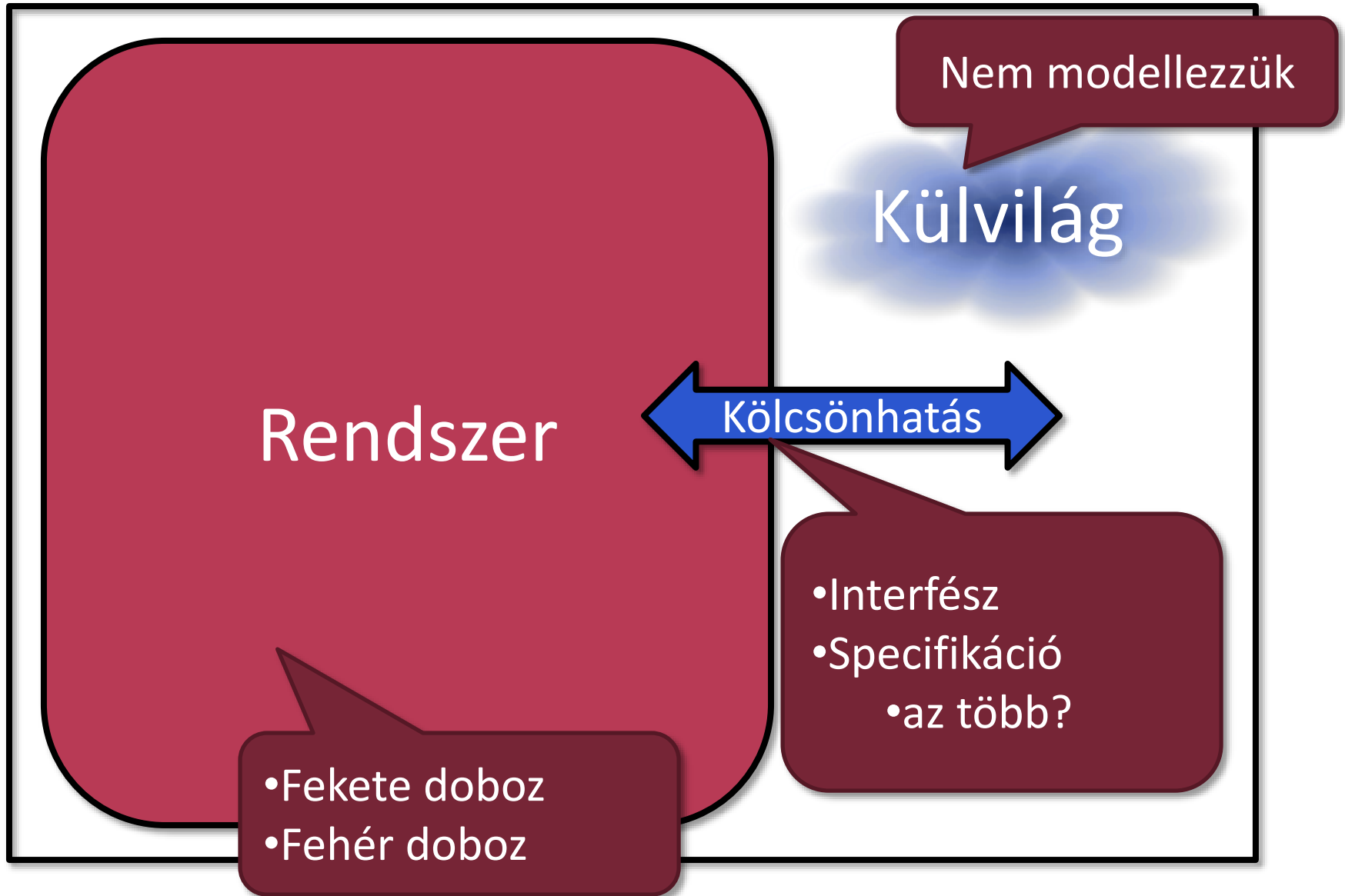
A
modellezés
alkalmazása

MODELLEZÉSI ALAPFOGALMAK

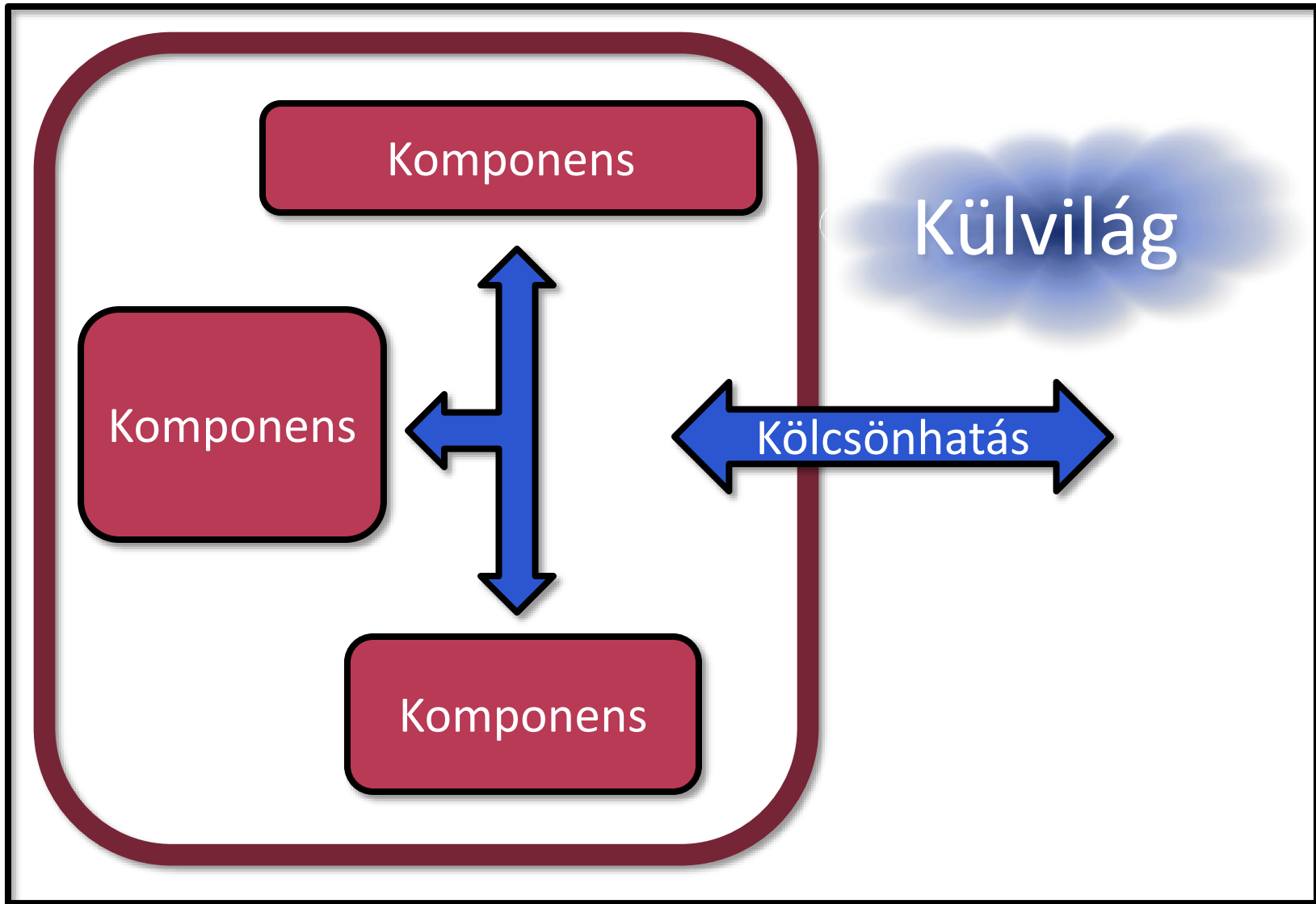
Modellezési alapfogalmak

- Rendszer és külvilág, komponensek
- Finomítás, absztrakció
- Metamodellezés

Alapfogalmak – rendszer és külvilág



Alapfogalmak – rendszer és külvilág

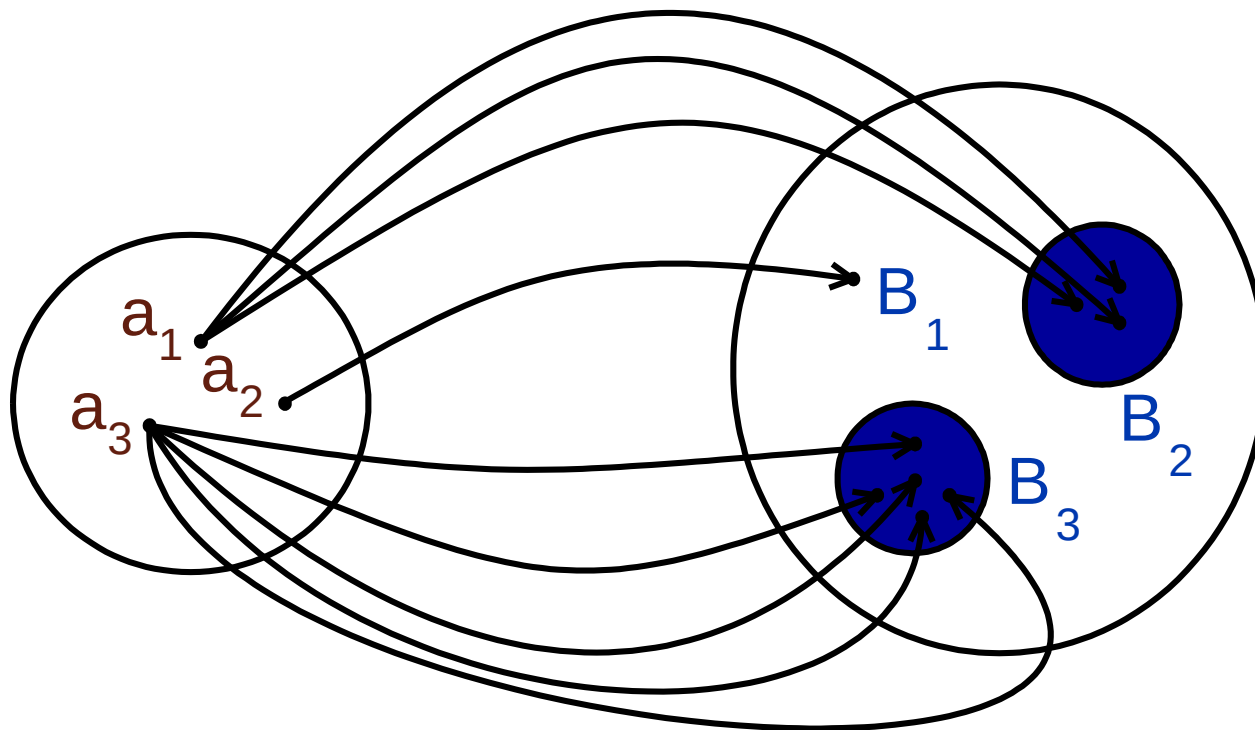


Alapfogalmak - Finomítás

- *Finomítás*: a modell gazdagítása részletekkel...
- ...hogyan az eredeti modell absztrakció maradjon
- Inverze: *(vertikális) absztrakció*
- Az előbbi dián egy *hierarchikus finomítás* volt
 - „dobozok kibontása”
- Finomítható más is...
 - Pl. Halmazfinomítás: változók értékkészlete
 - Jó / rossz helyett
 - Gyors / átlagos / lassú / hiányos / veszélyes

Halmazfinomítás

Diszjunkt részhalmazok hozzárendelése elemekhez

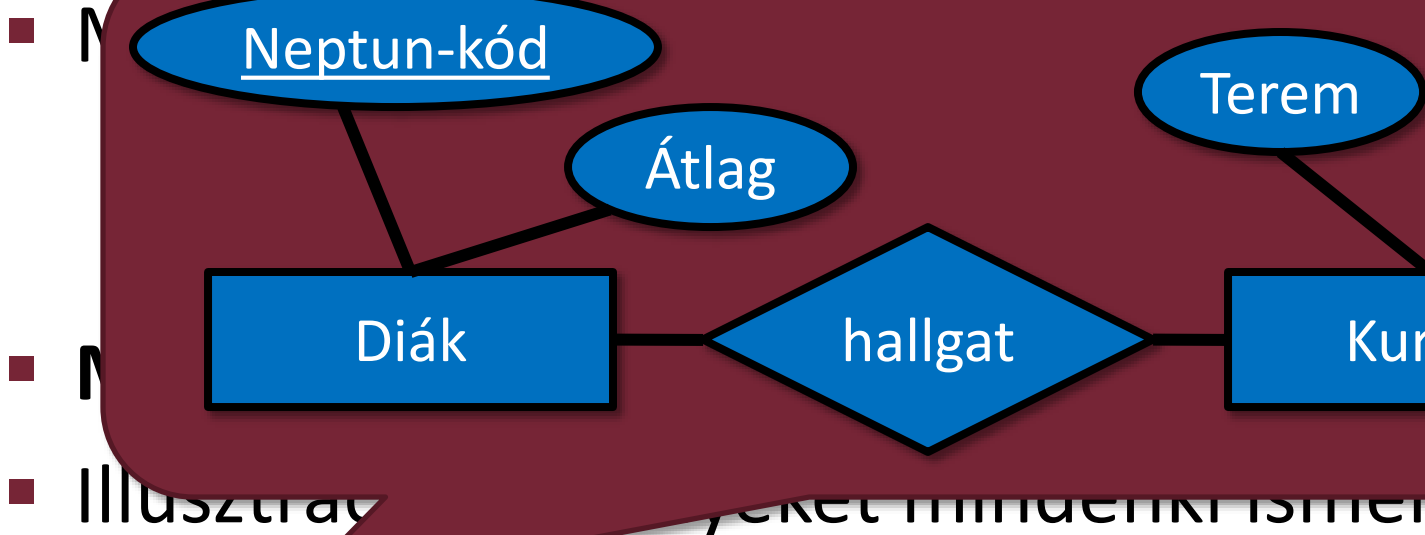


$\forall a_i, \in A, R(a_i) \subset B$ úgy, hogy $R(a_i) \cap R(a_j) = \emptyset \quad \forall i, j$

Alapfogalmak - Metamodellezés

- Modellezési nyelv: milyen típusú elemei vannak?
 - ...és milyen kapcsolatban állhatnak ezek az elemek?
 - ...és ezeknek a típusoknak mik a viszonya egymáshoz?
- **Metamodell** = egy modellezési nyelv modellje
- Illusztrációk, amelyeket mindenki ismer
 - Egyed-kapcsolat (ER) modell
 - UML objektum diagram → osztálydiagram
 - Adatbázis tábla → relációs adatbázisséma
 - XML dokumentum → XML séma (vagy DTD)
 - ...

Alapfogalmak - Metamodellezés



- Egyed-kapcsolat (ER) modell
- UML objektum diagram → osztálydiagram
- Adatbázis tábla → relációs adatbázisséma
- XML dokumentum → XML séma (vagy DTD)
- ...

Modell és
modellezés

Mire
használunk
modelleket?

Alapfogalmak

Modellek
osztályozása

Illusztratív
példák

A
modellezés
alkalmazása

MODELLEK OSZTÁLYOZÁSA

Osztályozási szempontok

- Felépítési vs. viselkedési modellek
- Matematikai-formális vs. informális
- Folytonos vs. diszkrét változók és idő szerint
- Végrehajtható vs. deklaratív
- ...

Osztályozás: felépítési vs. viselkedési

■ Felépítési (*structural*)

- Statikus
- Rész és egész, összetevők
- Kapcsolatok, összeköttetések

Egy kosárban hat bájos kismacska van

■ Viselkedési (*behavioral*)

- Dinamikus
- Időbeli lefolyás
- Állapot, folyamat
- Reakciók a külvilágra

A kismacskák, amikor ébren vannak, tudnak inni, nyávogni és karmolni

Egy kosárban hat bájos kismacska van, akik tudnak inni, nyávogni és karmolni

■ Nem fed le mindent, nem válik élesen szét...

Osztályozás: formális vs. informális

- Mennyi matematikai állítást tartalmaz?
 - Mennyire tereli mederbe a modellezett rendszert?
- Folytonos skála, nem válik élesen szét
 - Differenciálegyenletekkel leírt közegáramlás
 - Állapotgép
 - Szekvencia diagram
- Nem biztos, hogy mindig a szigorúbb a jó
 - Néha nehéz beletenni azokat a matematikai állításokat
 - Szemléletesség...

Osztályozás: formális vs. informális

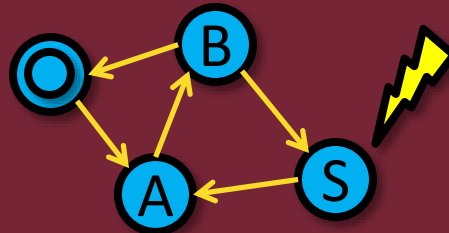
■ Mechanika

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \nabla \cdot (\mu \cdot (\nabla \mathbf{v} + (\nabla \mathbf{v})^T)) + \nabla (\lambda \nabla \cdot \mathbf{v}) + \rho \mathbf{g}$$

szert?

■ Folytonos mechanika

- Differenciálegyenletek
- Állapotgép
- Szekvencia diagram

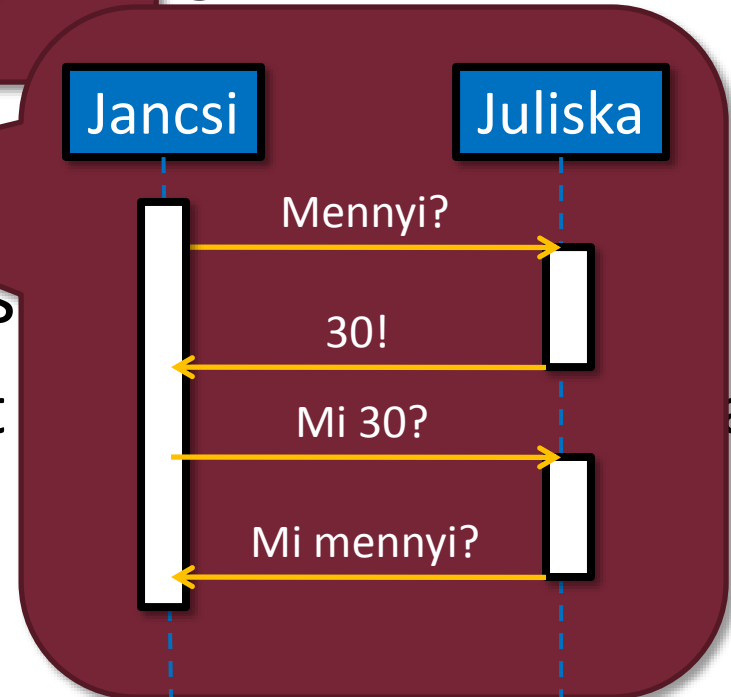


n szét

egáramlás

■ Nem biztos, hogy mindig a s

- Néha nehéz beletenni azokat
- Szemléletesség...



at

Osztályozás: két további szempont

- Folytonos vs. diszkrét
 - Időben
 - Értékben
- Viselkedési modell végrehajtható?
 - Teljes eseménysor determinisztikusan rekonstruálható
 - Eseménysor sztochasztikusan definiált
 - Nemdeterminisztikusan végrehajtható
 - Részben korlátozza a lehetséges eseményteret
 - Pl. csak ellenőrizhető kritériumok
 - Nem is viselkedési modell

Modell és
modellezés

Mire
használunk
modelleket?

Alapfogalmak

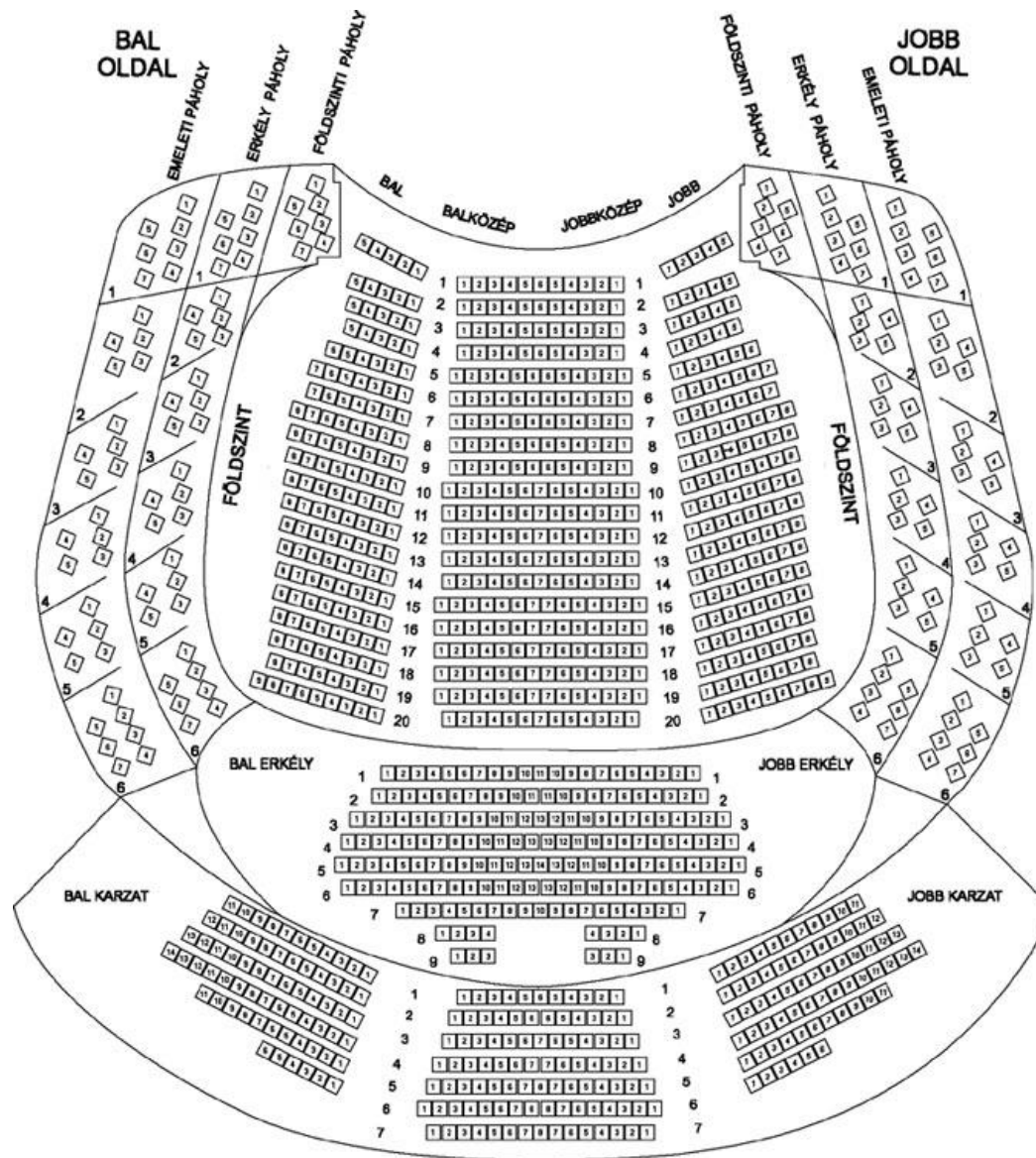
Modellek
osztályozása

Illusztratív
példák

A
modellezés
alkalmazása

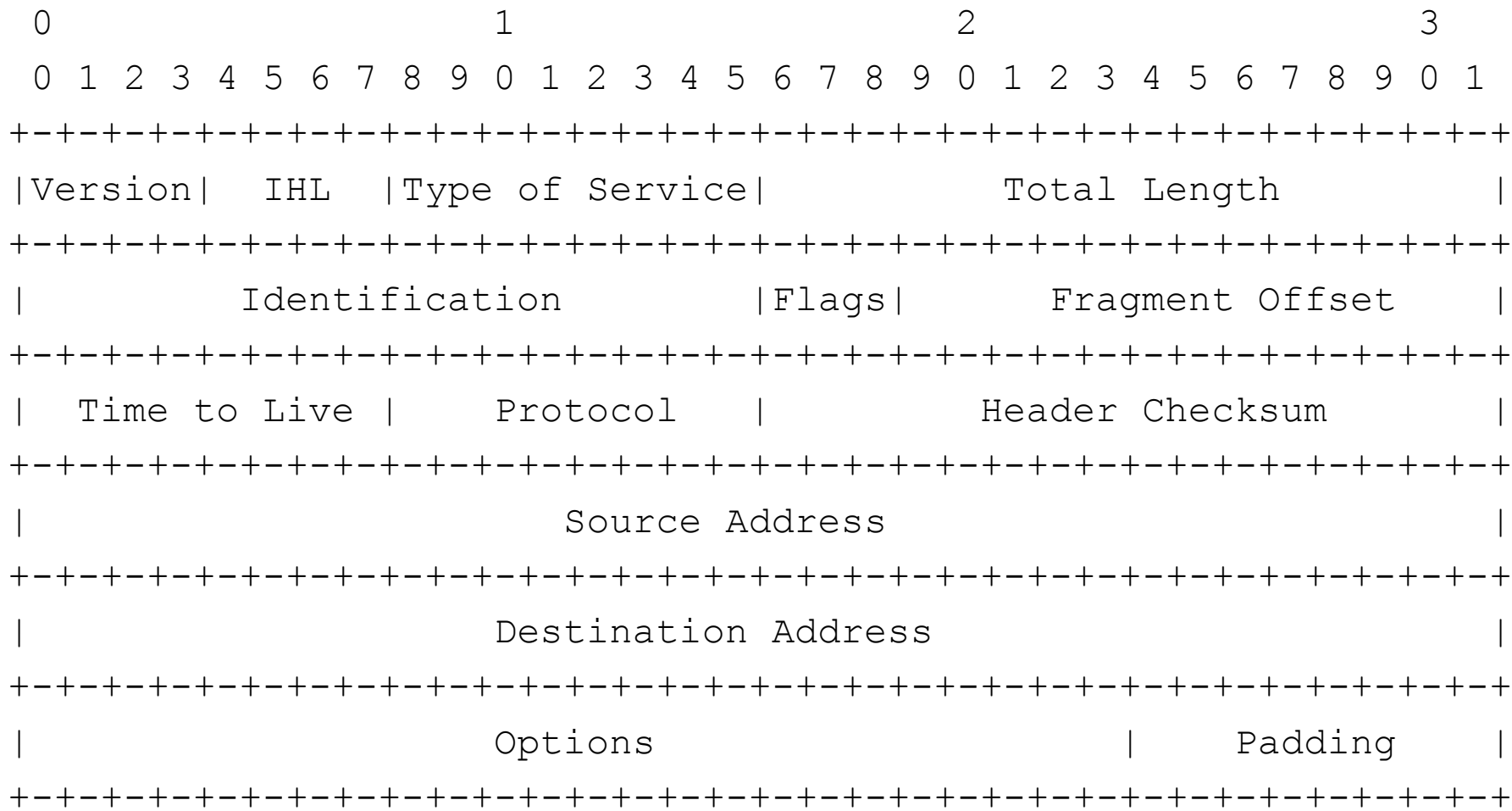
ILLUSZTRATÍV PÉLDÁK

Illusztráció – Felépítési modellek



Illusztráció – Felépítési modellek

Keretszerkezet



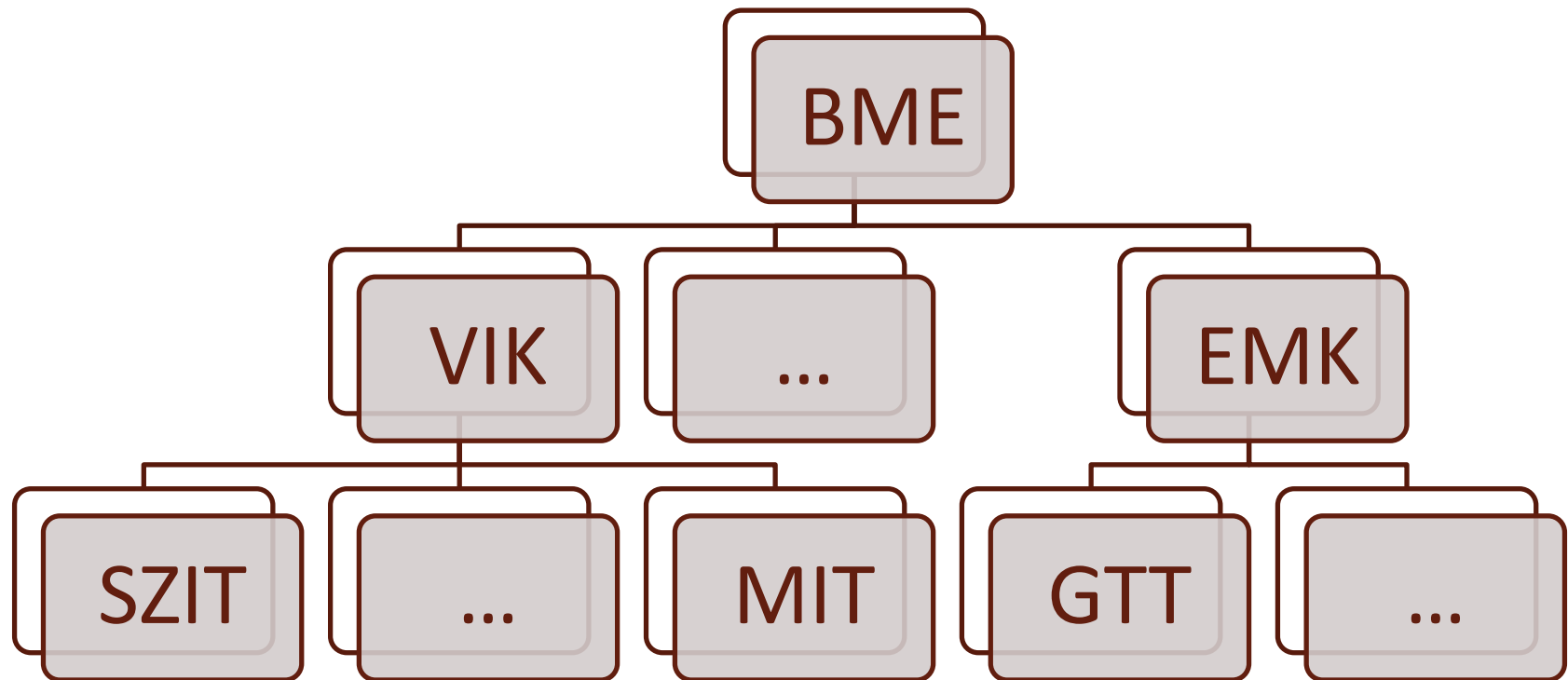
Illusztráció – Felépítési modellek

Tartalmazási hierarchia

- /
 - bin/
 - home/
 - lib/
 - usr/
 - share/
 - lib/
 - var/
 - log/
 - tmp/

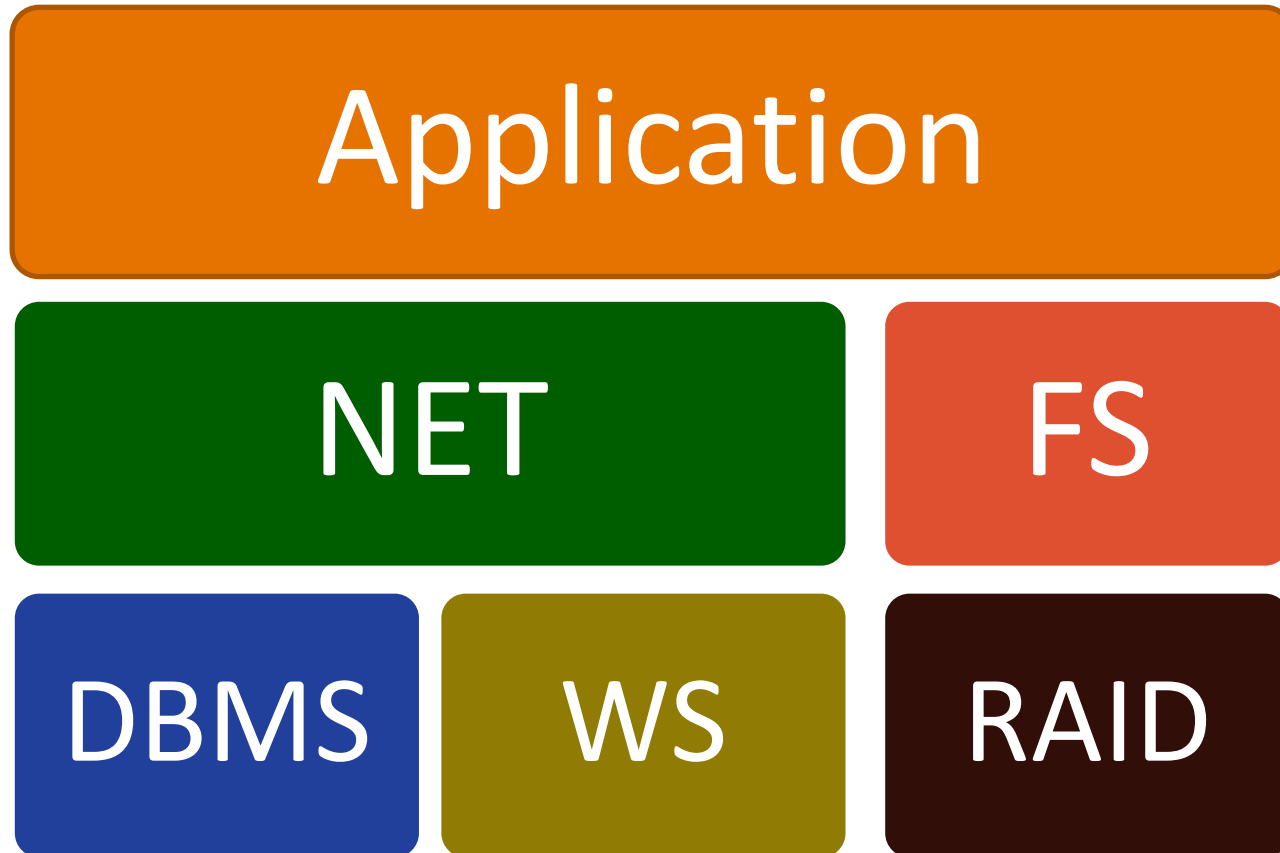
Illusztráció – Felépítési modellek

Szervezeti felépítés (ld. tártalmazási hierarchia)



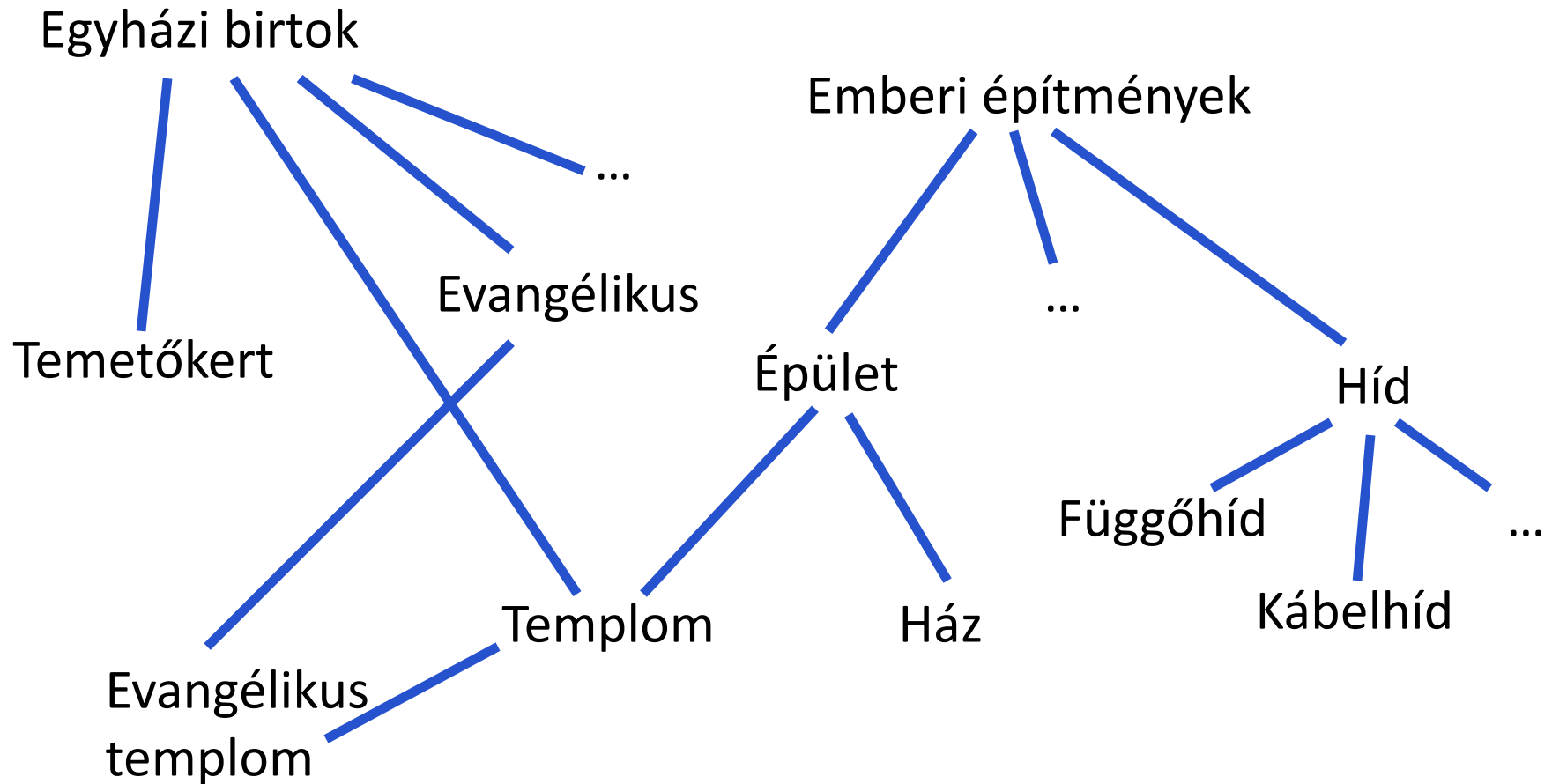
Illusztráció – Felépítési modellek

Architektúra modell (blokkdiagram)



Illusztráció – Felépítési modellek

Taxonómia jellegű



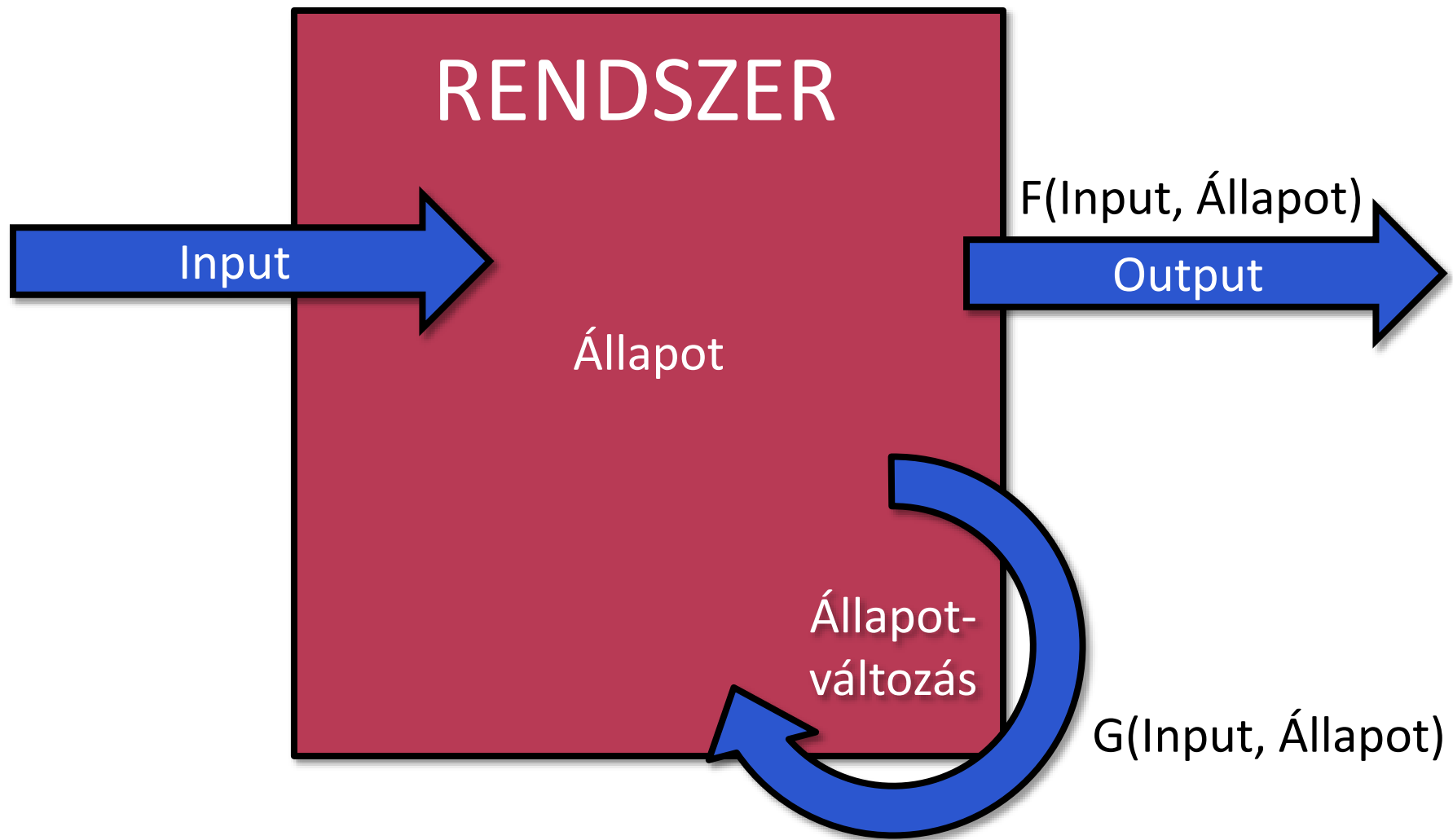
Illusztráció – Felépítési modellek

■ Ontológia

- Fogalmak és kapcsolatok taxonómiája
- Fogalmi- és viszonyállítások összekapcsolódnak
 - „Férfiak és nők között definiált a házasság”
 - „Anya: az a nő, akinek van legalább egy gyermeke”
 - „A halott férfi élő házastársa: özvegyasszony”
 - ...ez persze a konkrét formalizmustól függ
- Példák
 - WordNet
 - Dublin Core
 - Orvosi, gyógyszeripari, bioinformatikai ontológiák

Illusztráció – Viselkedési modellek

Klasszikus rendszerelméleti automata-modell



Illusztráció – Viselkedési formalizmusok

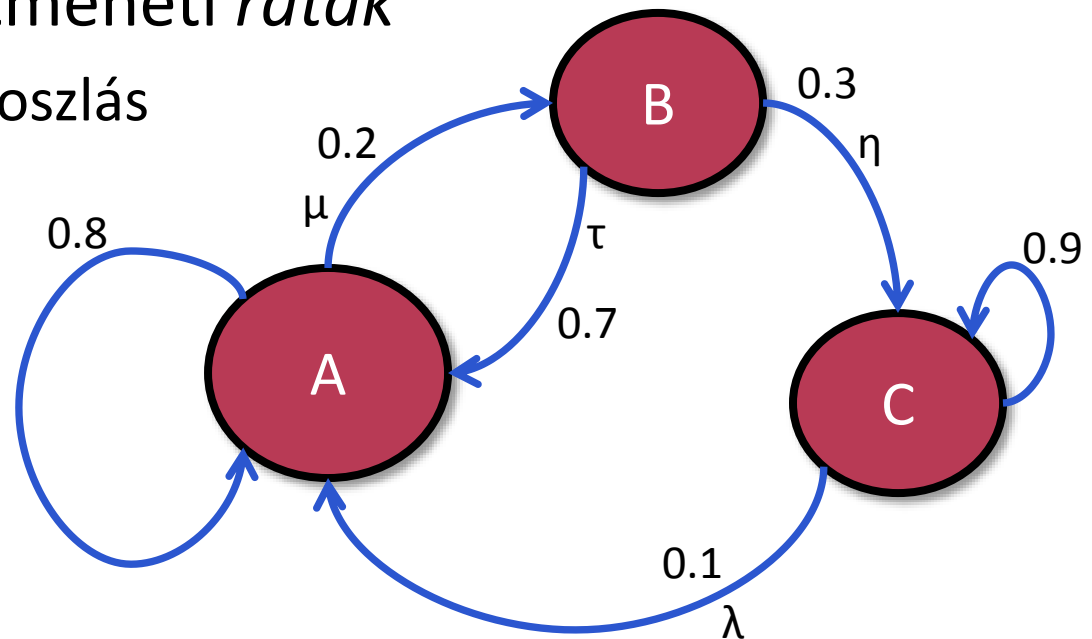
- Állapottérkép
 - Automaták szinkron és aszinkron kompozíciója
 - Címkezett állapotok, átmenetek
- Vezérlési folyam
 - Kiterjesztés : konkurencia
 - Kiterjesztés: erőforrásmodell
- Adatfolyamháló (Dataflow Network, DFN)
 - Nagyon általános formalizmus
 - Megvalósítható vele az előző kettő

Illusztráció – Viselkedési formalizmusok

- Petri-háló
 - Erőforráskorlátok
 - Kompozíció, szinkronizáció, konkurencia, deadlock...
 - Potenciálisan végtelen állapottér
 - Erős matematikai háttér
 - Kiterjesztés: színes PN, hasonlít a DFN-re
- És még...
 - Processz-algebrák
 - Gráfnyelvtanok
 - ...

Illusztráció – Viselkedési formalizmusok

- Sztochasztikus (Markov-) modellek
 - Sok viselkedési modell kiegészíthető valószínűségekkel
 - Matematikai háttér, származtatható jellemzők
 - Diszkrét idő: átmeneti valószínűségek
 - Folytonos idő: átmeneti *ráták*
 - Exponenciális eloszlás



Modell és
modellezés

Mire
használunk
modelleket?

Alapfogalmak

Modellek
osztályozása

Illusztratív
példák

A
modellezés
alkalmazása

A MODELLEZÉS ALKALMAZÁSA

Modellezés a tárgyban

Rendszermodellezés

- Építés
- Szimuláció
- Mérés

HF

(Üzleti-) Folyamatmodell

- Vezérlés
- Erőforrások
- Teljesítmény

Informatikai
szolgáltatás
modellezése
üzleti
folyamatként

Példák

Modellalapú szoftvertervezés:

Precíz mérnöki
modellek és
transzformációk
segítségével
történő tervezés,
fejlesztés és
projektvezetés.

Szolgáltatásbiztos rendszerek:

Megbízható, nagy
rendelkezésre állású,
kritikus rendszerek
tervezése,
megvalósítása,
mérése, felügyelete.
Cloud.

Verifikáció és validáció:

Ellenőrzések a
tervezési,
megvalósítási és
futási fázisban.
Formális módszerek
alkalmazása a
verifikáció során.

ÜZLETI ALKALMAZÁSOK

- Tudásmodellezés
- Gyártásoptimalizás
- Hatékonyságelemzés
- Integrált rendszer

Az egységes európai szállítmányozási rendszer

EmeraldWebVisualizer - Windows Internet Explorer

http://localhost:8080/EmeraldWebVisualizer/?perspective=ftsrc&locale=en

EmeraldWebVisualizer v. 1.0-b920-ftsrc

Perspective: Document

Document data

Highlighted data All data

Document versions

Language: hu

Table of contents

Source document

2) IMO number (imo)

3) Call sign

4) Voyage number

2. Port of arrival / Port of departure

3. Date and time of arrival / Date and time of departure

4. Flag State of ship

5. Name of master

6. Last port of call / Next port of call (nextPort)

7. Certificate of registry (Port; date; number)

8. Name of ship's agent and Contact details of ship's agent

9. Gross tonnage

10. Net tonnage

11. Position of the ship in the port (berth or station)

12. Brief particulars of voyage (previous and subsequent ports of call; underline where remaining cargo will be discharged)

13. Brief description of the cargo

14. Number of crew

15. Number of passengers

16. Remarks

Attached documents (indicate number of copies)

Entity hits: Previous << 0 / 0 >> Next

Page 1 of 1

No data to disp

Terminology PersonalBrain View Comments

Comment [imo]

Mandatory if MMSINumber not given. IMO number has to be checked if not existing in the reference database.

Comment [nextPort]

This attribute indicates the actual next port of call, e.g. if the port of Oostende is sending this notification, then this NextPortOfCall attribute must be the location code of Oostende (BEOST) and not the next port of call after Oostende. The cancellation of a previous port notification is merely a new port notification for which the NextPortOfCall attribute value is "ZZCAN". E.g. during the voyage, the master of the ship receives new orders from his head office. The new destination is now port of Stockholm instead of Amsterdam. Upon receiving this information, the Master of the ship has the obligation to inform Amsterdam about these changes but he has not obligation to indicate his new port of destination. In this case, Amsterdam must transmit a new port notification message to SSN, cancelling the one sent previously, by indicating that the next port of call is "ZZCAN" and then ETA and ETD attributes are not given. The "next port of call" attribute cannot be unknown ("ZZUKN"). The "next port of call" attribute must only be the LOCODE of the specific port of call or its dependent port's LOCODEs.

Tudás

- Feltárása
- Modellezése
- Formalizálása

Egy nagy magyar lakossági szolgáltató

Megoldandó probléma: heterogén (üzleti) adatok alapján vizsgáljuk meg a cég belső jutalékrendszerét, keressünk érdekes összefüggéseket.

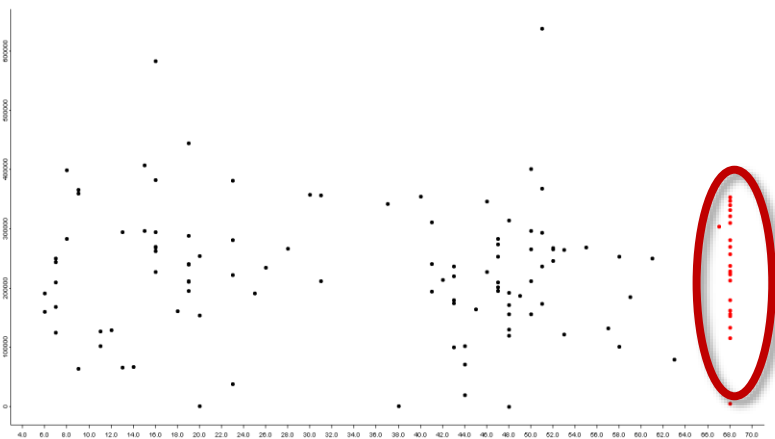
Kihívások:

- Definiálatlan folyamatok
- Össze nem kapcsolt adatforrások
- Piszkos adatok, kivételek



Főbb eredmények

Adatkezelési és elemzési folyamatok, javaslatok a hatékonyság növelésére

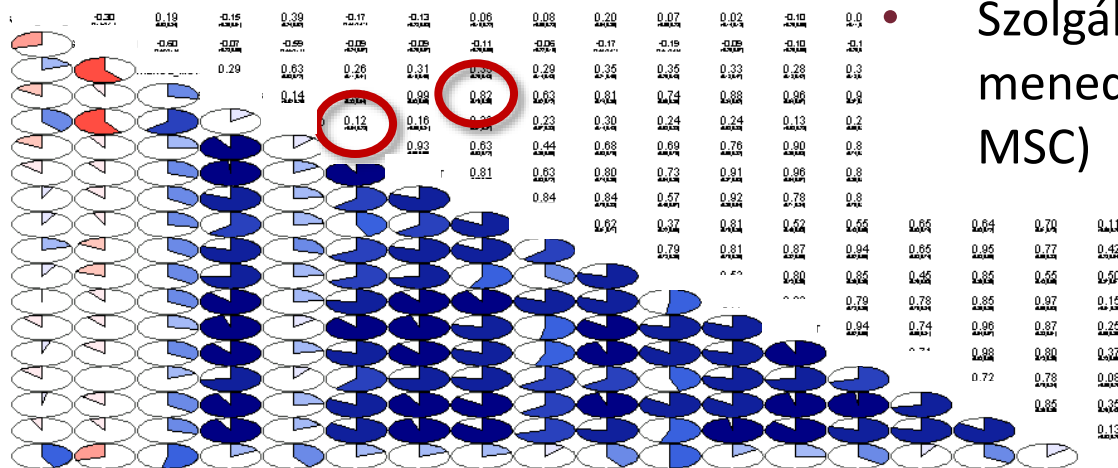


Megtanultuk:

- KNIME, R
- Adatkáosz és értelmezés
- Agilis projektmenedzsment (?)

Oktatás:

- Rendszermodellezés (BSc)
- Szolgáltatásfejlesztés és –menedzsment a gyakorlatban (GaiN MSC)

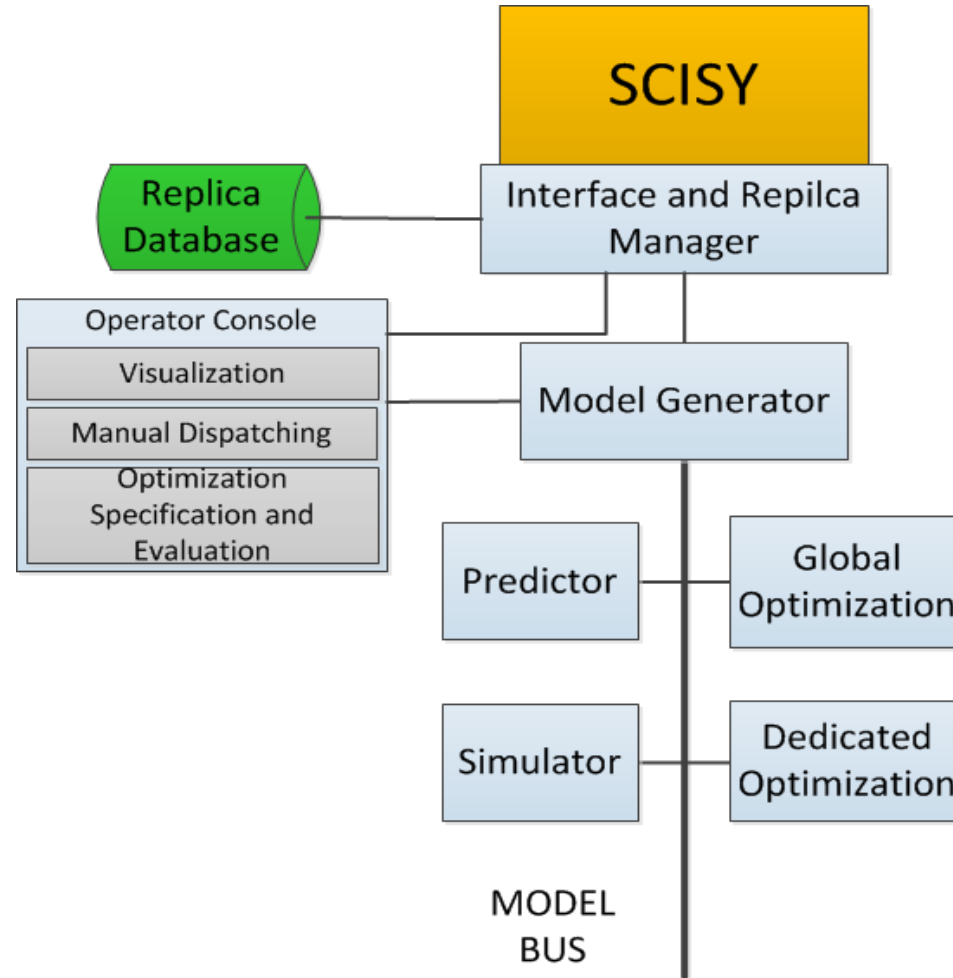


IBM Data Storage Systems

Megoldandó probléma: időben változó és előre nem ismert rendelésállomány anyagbeszerzés+gyártás optimalizálása

Kihívások:

- Több milliós konfiguráció szám
* ~1500 rendelés
KOMBINATORIKUS ROBBANÁS
- Kevés és zajos adatból jóslás
ADAPTÍV PREDIKCIÓ



Példa folyamat – tesztelési lépések

Kiegészítés egy létező konfigurációhoz

Új gép gyártása

Újrakonfigurálás

Gyártási folyamat

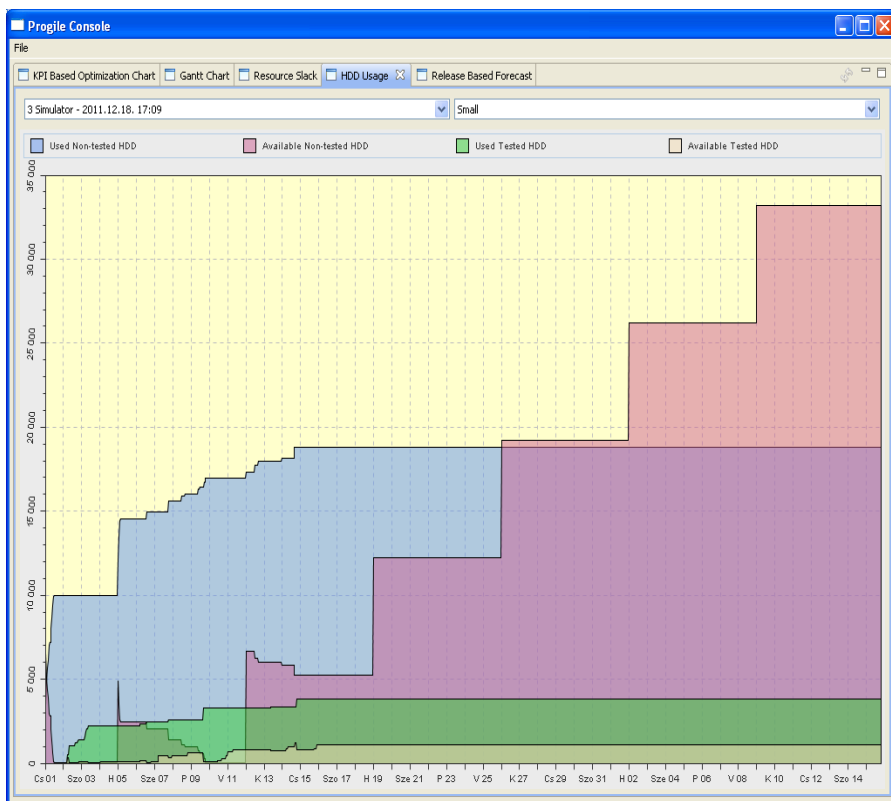
- Modellezése
- Szimuláció
- Optimalizálás

Főbb eredmények

Működő rendszer élesben használva: 40 perc futási idő 1 negyedévre

European Supply-Chain Technology Award (2012)

Revolution R: „Applications of R in Business” competition, Honorable Mention (2012)



Megtanultuk:

- IBM CPLEX
- IBM SPSS
- IBM JViews

Oktatás:

- Rendszermodellezés (BSc)
- Üzleti rendszerek modellezése (Gazdinfo)

Mérnöki vs. matematikai modellezés

- Tacit tudás modellezése!
- Nem kell matematikai optimum
- Heurisztikus gyorsítás
- Darabolással jól csökkenthető az állapottér
- Józan ésszel több nagyságrend komplexitás absztrahálható el

Smart Metering projekt

Feladat: egy szolgáltató okos mérős mintaprojektje

- 3000 okos mérőóra (villany, víz, gáz) telepítése
- 15 perces mérési adatok gyűjtése, ellenőrzése, becslése, számlázásra bocsátása
- Kapcsolódó rendszerek (mérők, MDMS, SAP stb.) integrációja

Eredmények:

- Új szakterület, IEC szabványok ismerete
- Nagyméretű ipari rendszerintegrációs példa

Oktatás:

- Szolgáltatásintegráció (MSc)

Lehetőségek:

Smart Grid elemek fejlesztése

- Hardver

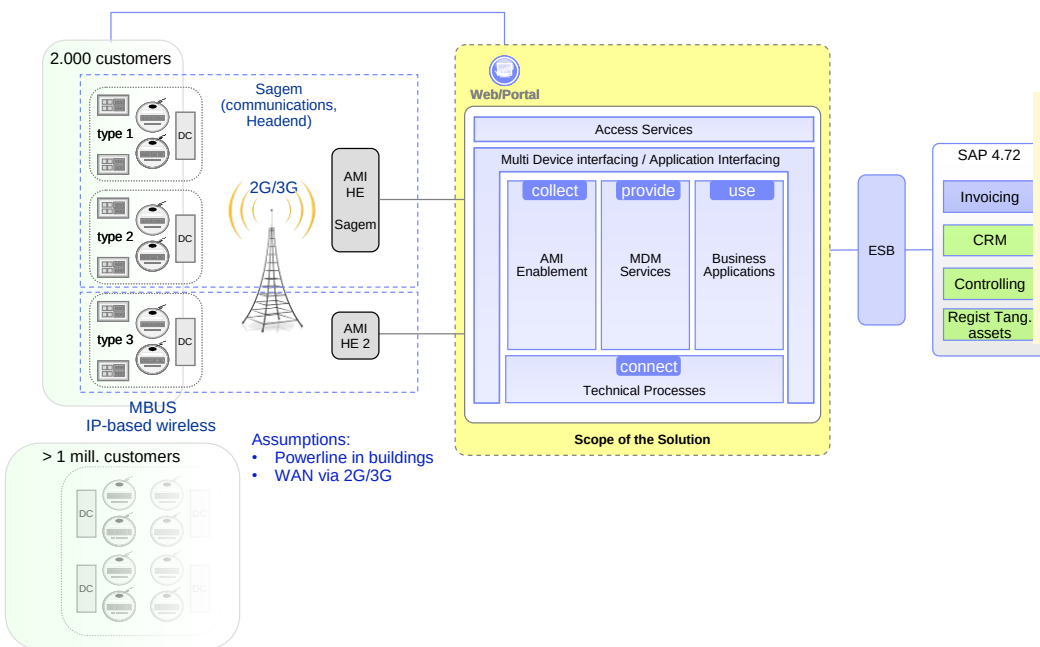
Tudás

- Eszközfüggetlen reprezentálása
- Adatbázis tervezés
- Alkalmazástervezés

- Szabványos kommunikáció, integráció

Potenciális hasznosítás:

- Intézményen belüli okos mérős rendszer
- Komponensek elosztói hálózatokba



Assumptions:
• Powerline in buildings
• WAN via 2G/3G

IT INFRASTRUKTÚRA

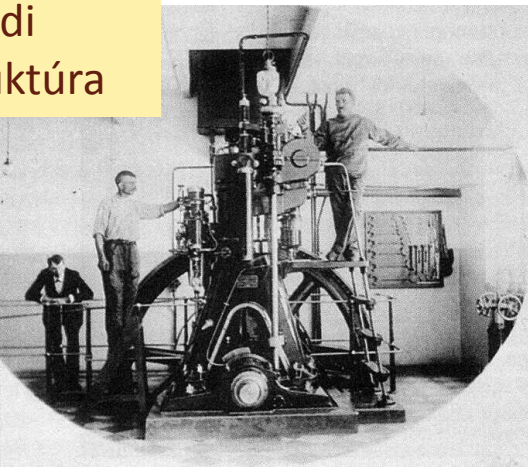
Teljesítmény- és szolgáltatásbiztonság elemzése

- **Cloud computing**
- **Rendszerfelügyelet és management**

A felhő számítástechnika

Kezdet: a fogyasztó magának termel energiát

Csúcsterhelésre
méretezett
egydi
infrastruktúra



A felhő számítástechnika ugyanez

- Energia=számítási teljesítmény
- Erőmű= szerver
- Távvezeték=Internet
- Elosztás, védelem, mérés, szabályozás



Energiaszolgáltatás

Hatékony termelés



Csúcserőmű



Konfigurálás, védelem,
redundancia



Tehereelosztás,
Monitorozás
Szabályozás



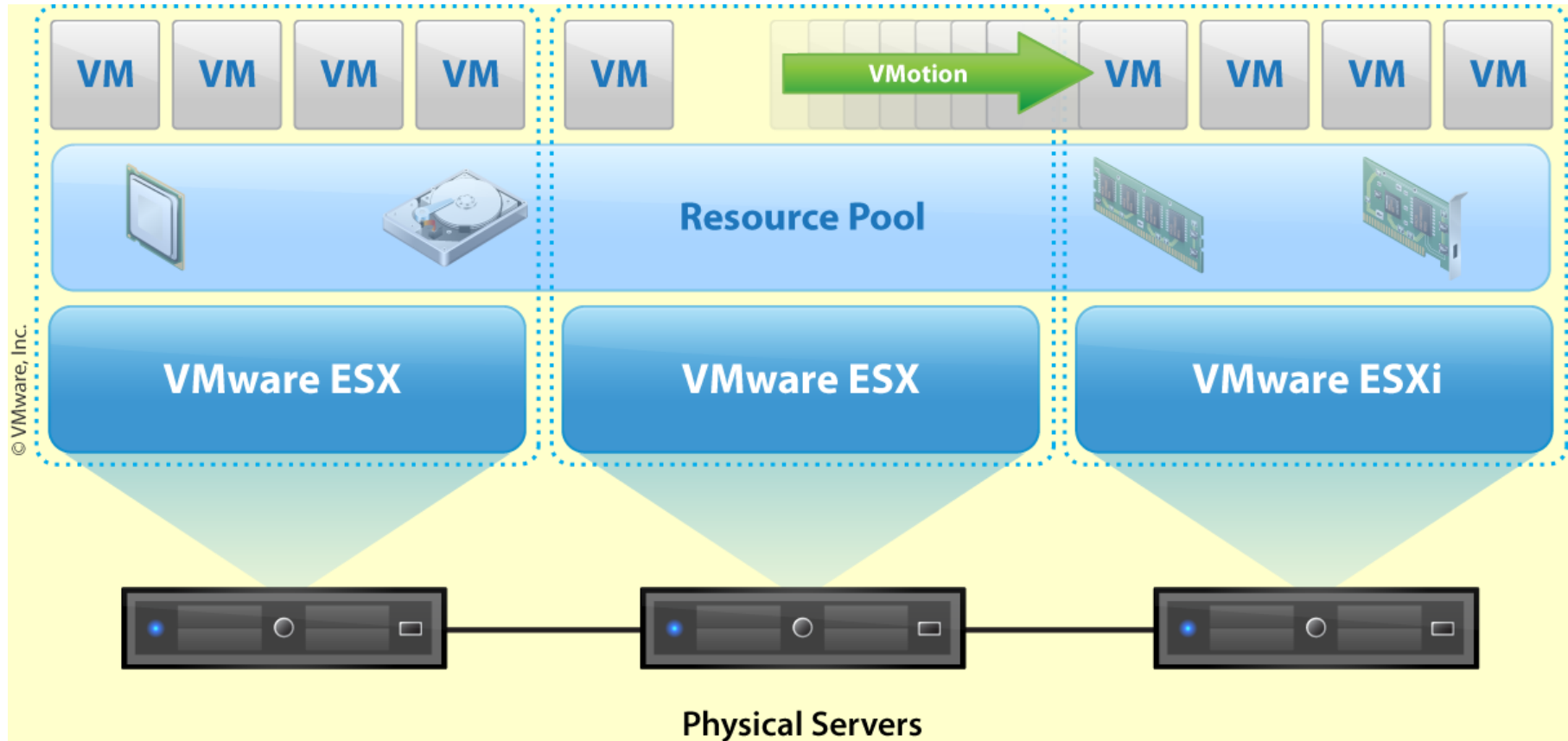
Hatékony szállítás



Fogyasztás
SLA



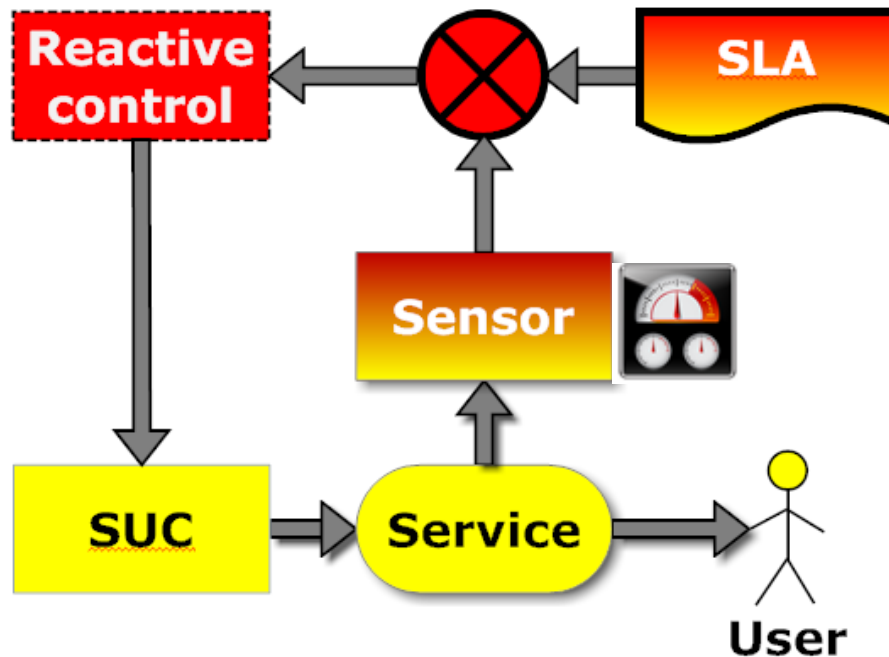
Virtualizáció



Proaktív felügyelet és szabályozás

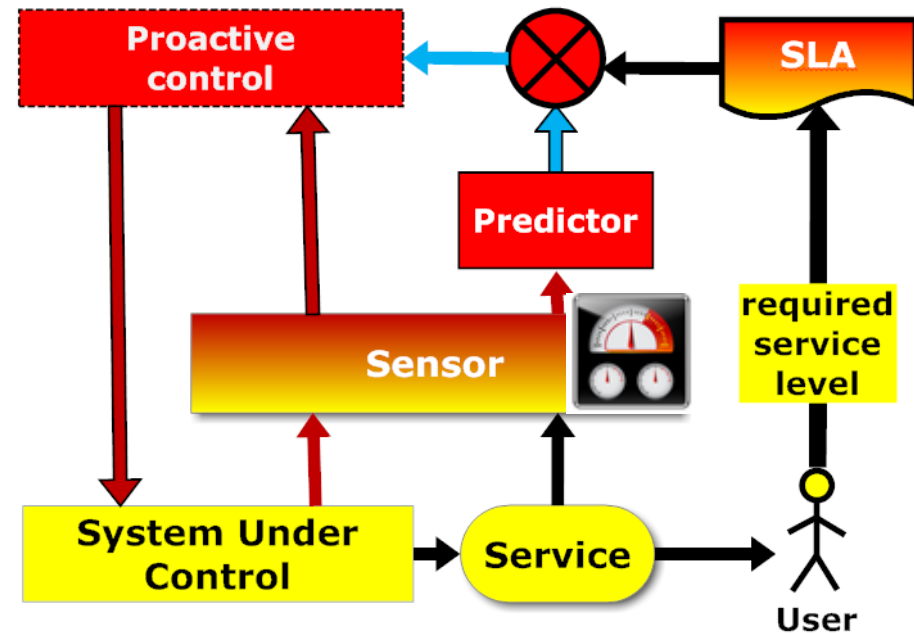
■ Reaktív szabályozás

- Egy veszélyes szituációra **reagál** ahelyett hogy elkerülné vagy uralná



■ Proaktív szabályozás

- Egy szituációt már a **bekövetkezése előtt** ural



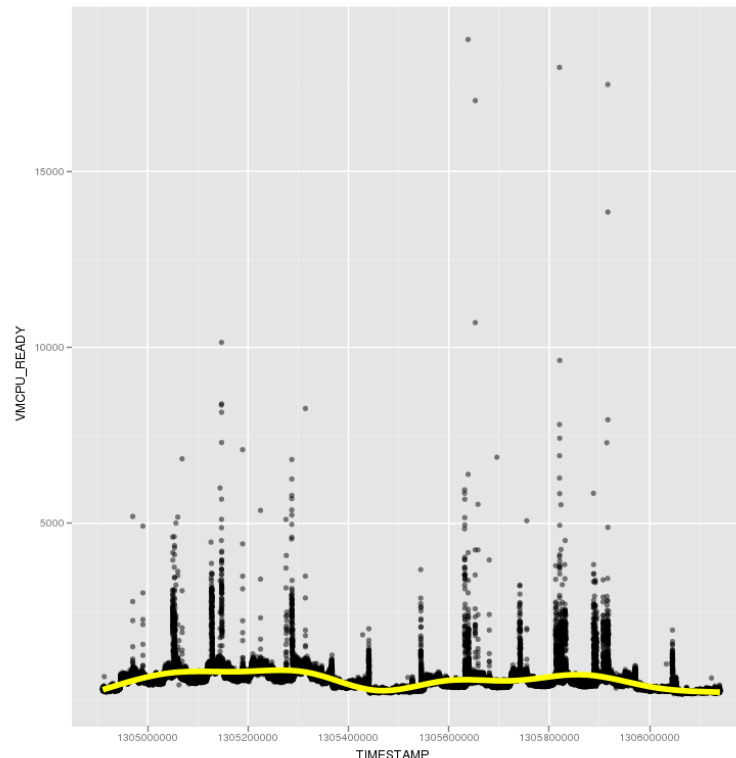
Egy gazdasági világcég

Megoldandó probléma: egy sokfelhasználós, privát cloud alapú rendszerben a felhasználói gépek időnként „csuklanak”

Statikusan is kevés a kapacitás, vagy rossz az elosztás dinamikája?

Kihívások:

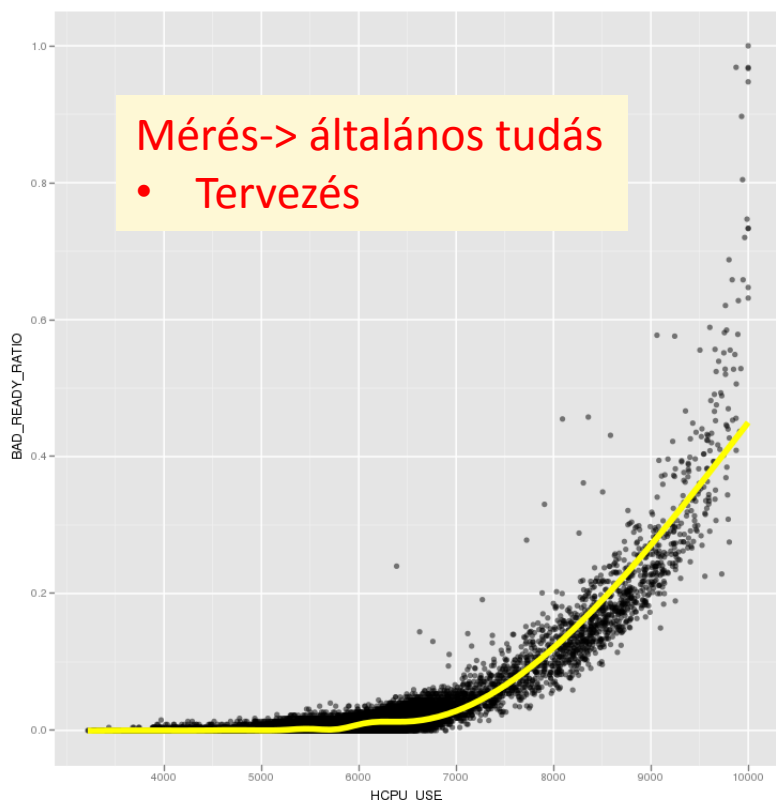
- Adatsor: 180 millió x 20 ezer
BIG DATA
- Ebből 6 ezer hibára utaló
RITKA ESEMÉNY ANALÍZIS
- Mérési hibák
ADATTISZTÍTÁS



Főbb eredmények

Erőforráskezelést kell finomítani és nem szerverkapacitást növelni (\$\$\$!!!)

IEEE DSN/PFARM meghívott előadás (Hong Kong 2011)



Megtanultuk:

- Vizuális analízis, R statisztikai analízis
- Virtualizációs környezet mérése

Oktatás:

- IDA (PhD)
- Rendszermodellezés (BSc)
- Autonóm és hibatűrő rendszerek (MSc)

Információs rendszerek mérés- és szabályozástechnikája

- Klasszikus megközelítések adaptálhatóak
- 30 évvel ezelőtti mérés technikai tudás nemzetközi novum

Egy telco technológiai világcég

Megoldandó probléma: QoS-re tervezés cloud környezetben

Kérdés: Mit nyújt a cloud platform az alkalmazásomnak?

Kihívások:

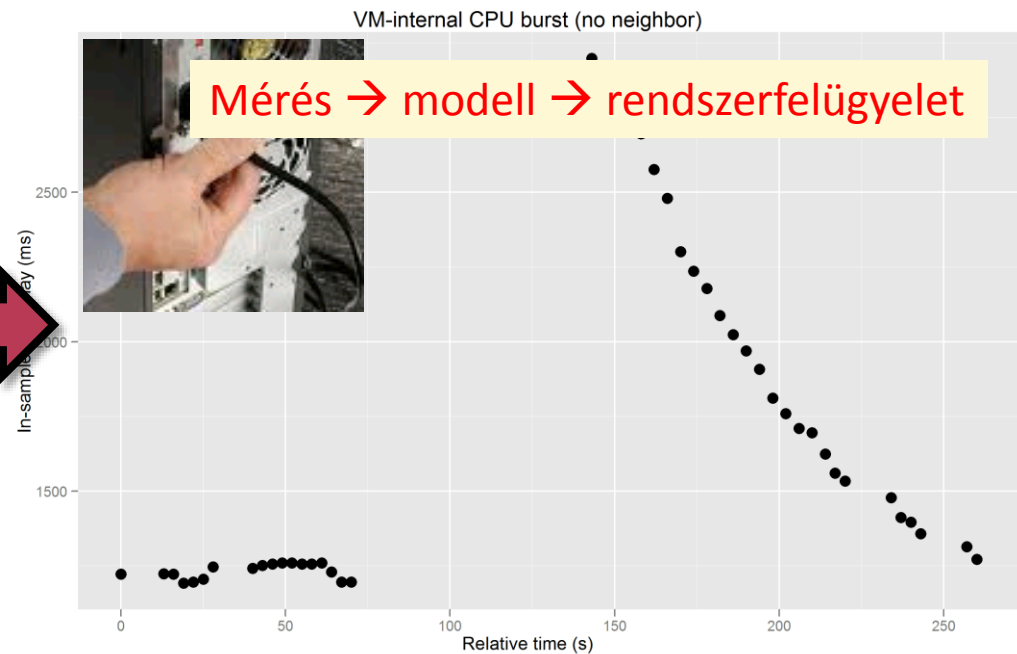
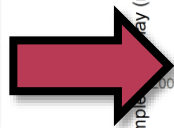
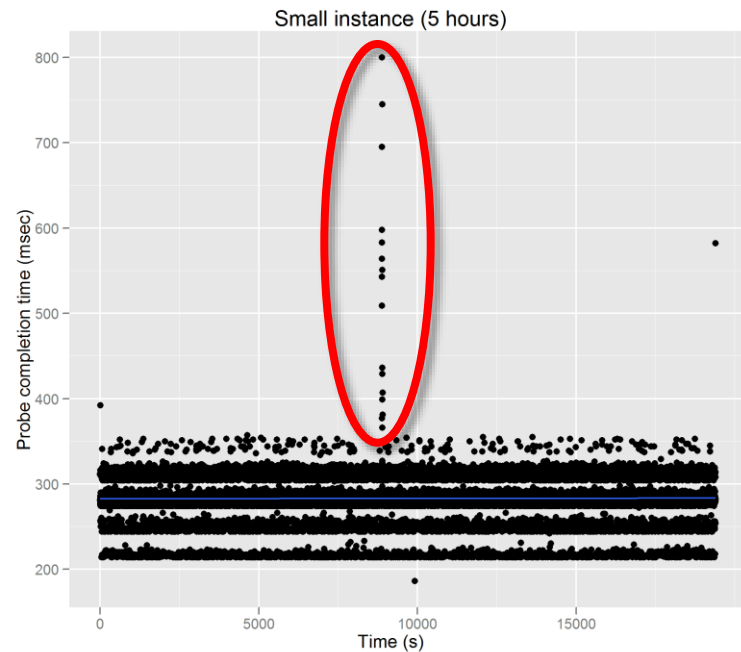
- Nehezen kezelhető, nagy alkalmazások
- Minden cloud más
- Mérhető jellemzők megmérése és relevánsak kiválasztása



QoS - Főbb eredmények

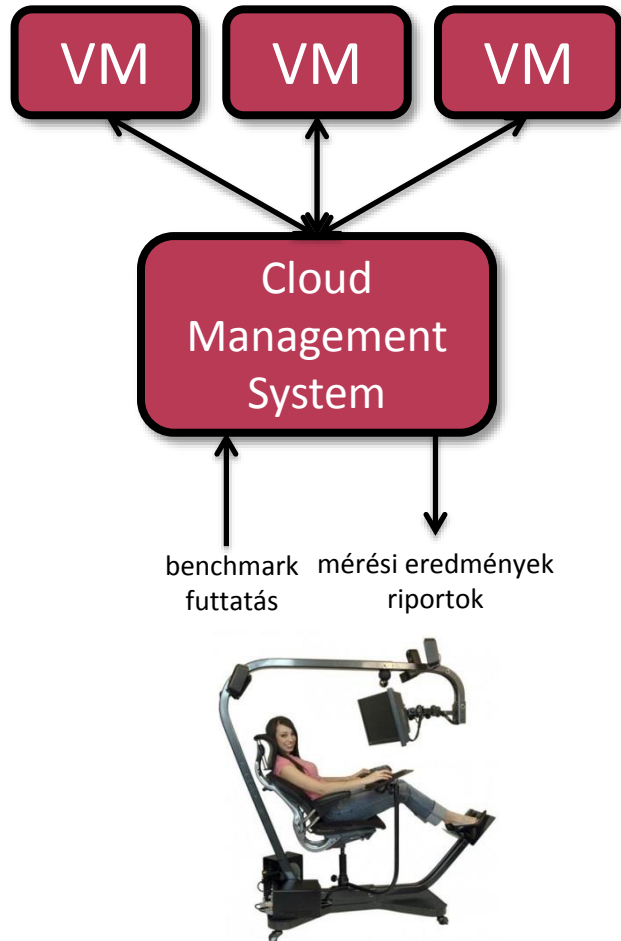
Főbb eredmények:

- „**Próbavásárló**”: finom felbontású, hosszú ideig futó, paraméterezhető intenzitású mikrobenchmarkok
- QoS-érzékenységvizsgálat: statisztikai eszközök
- Hibatűrési **minták** tervezhetővé/paraméterezhetővé válnak



Főbb eredmények

A megrendelő eszközébe integrált automatikus mérő- és kiértékelő környezet kifejlesztése



Megtanultuk:

- Cloud-specifikus benchmarkok

Oktatás:

- IRF (BSc)
- Rendszermodellezés (BSc)
- Autonóm és hibatűrő (MSc)

Információs rendszerek QoS becslése

- A létező benchmarkok eredményeinek részletes analízise kell
- Minél több, lehetőleg alkalmazás specifikus tulajdonság kimérése szükséges (pl: késleltetések terhelésfüggősége)

SZOFTVERFEJLESZTÉS

- Kritikus szolgáltatások
- Beágyazott rendszerek, robotok
- Nyílt forráskódú projektjeink

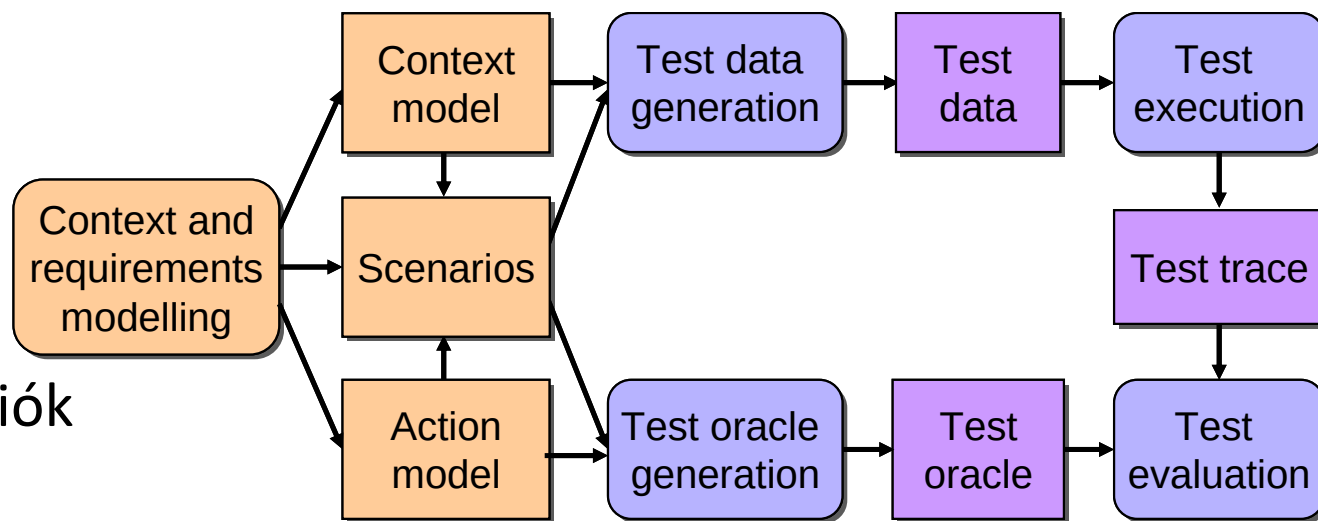
R3-COP: Robusztusság tesztelés részfeladat

Megoldandó probléma: Környezetfüggő, adaptív, autonóm robotok biztonságos és robusztus működésének tesztelése.

- Milyen környezetben sért a viselkedés biztonsági elvárásokat?

Kihívások:

- Elvárások precíz rögzítése (szcenáriók)
- Teszt konfigurációk (környezetek) szisztematikus generálása
- Teszt kiértékelés automatizálása

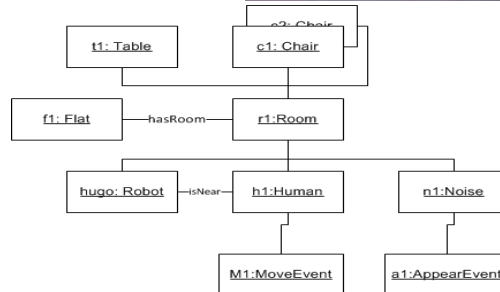
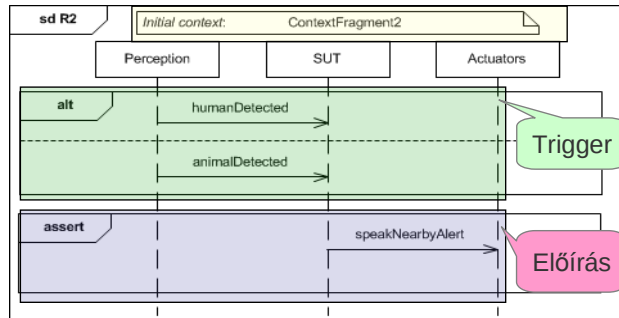


Modell=specifikáció → teszt



R3-COP: Robusztusság tesztelés eredmények

- Leíró nyelv teszt követelmények megadására
- Algoritmusok kidolgozása és megvalósítása
 - Teszt konfiguráció generálás, teszt lefutás kiértékelés
 - Mutációs tesztek a kommunikáció robusztusságának teszteléséhez



Újdonságok:

- Környezet ontológiák használata
- Keresés alapú algoritmusok (SBSE)

Demonstrátorok:

- Önjáró villástargonca (E80)
- Háztartási robot (Fraunhofer)

Oktatás:

- Szoftverellenőrzési technikák (MSc)
- Szoftver verifikáció és validáció (PhD)

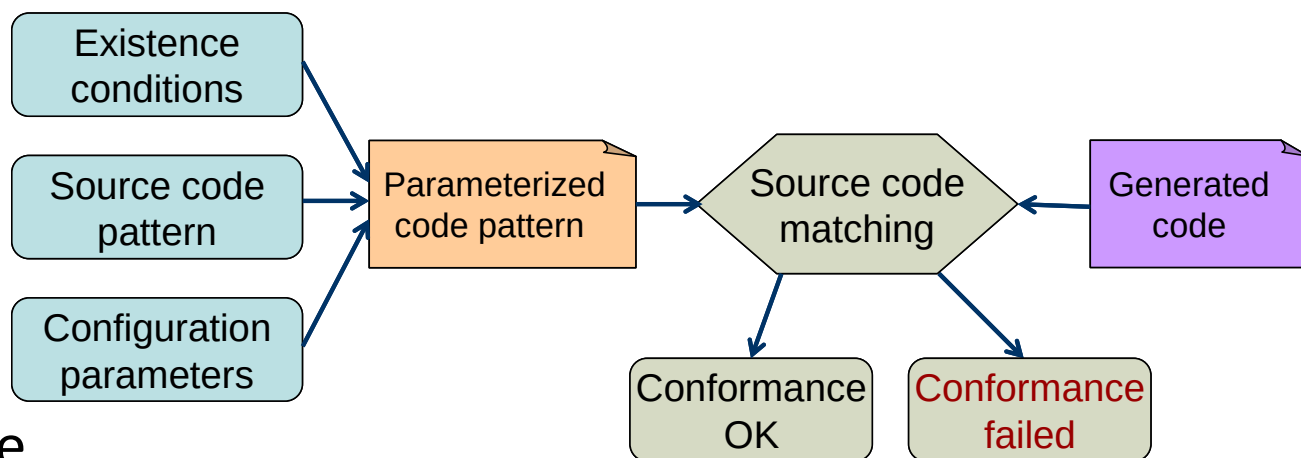
Forráskód ellenőrzés

Megoldandó probléma: Verifikációs eszköz tervezése konfigurációs paraméterek alapján, automatikusan generált szoftver forráskód helyességének ellenőrzésére

- Biztonságkritikus környezet: Helyes-e a generált kód?

Kihívások:

- Autóipari szoftver környezet
- Forráskód feldolgozás
- Kódminták keresése



Eredmények:

- Rendszerterv

Fejlesztési folyamatok értékelése

Megoldandó probléma: SIL-4 vasúti jelátviteli alkalmazás fejlesztési lépéseinek és ezek eredményeinek ellenőrzése a vasúti szoftverekre vonatkozó szabvány (EN50128) alapján

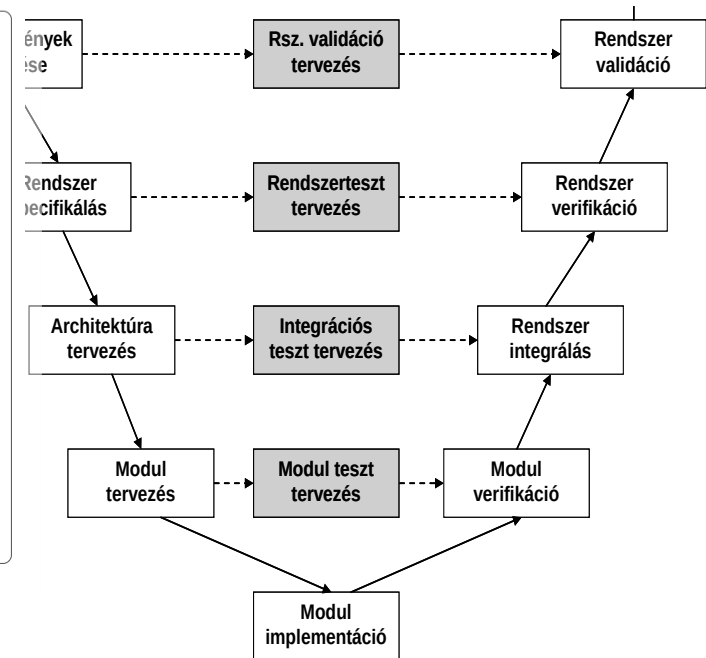
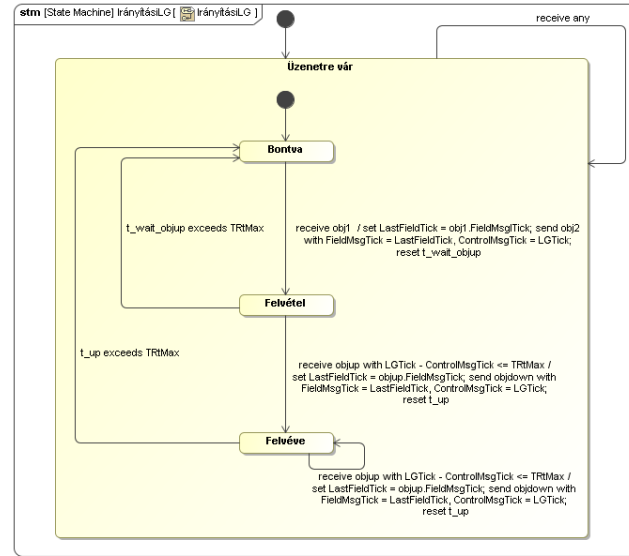
- Megfelelnek-e az eredmények a szabvány előírásainak?

Kihívások:

- Szisztematikus felülvizsgálat
- Modellézés és formális verifikáció

Oktatás:

- Kritikus beágyazott rendszerek (MSc)
- Formális módszerek (MSc)



Folyamat → modell → minőségbiztosítás

CECRIS: Kritikus rendszerek tanúsítása

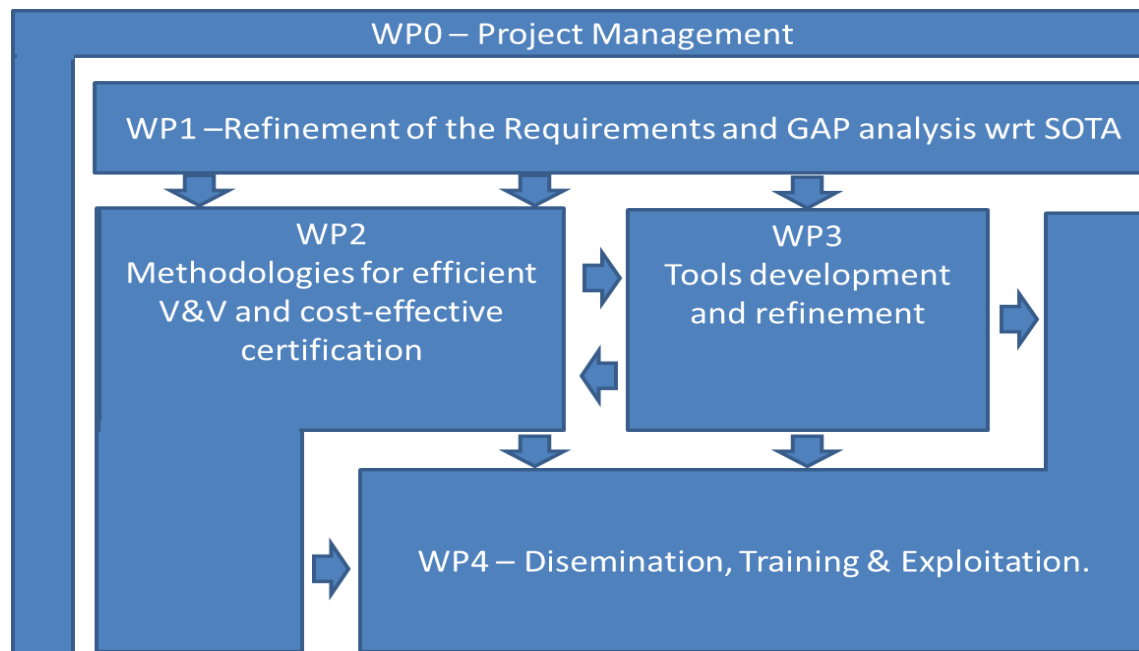
Megoldandó probléma: Módszerek és eszközök kidolgozása kritikus rendszerek verifikációjához és validációjához. Tudástranszfer ipari és akadémiai partnerek között.

Kihívások:

- Komponensek újrafelhasználása
- Modell alapú tervezés és implementáció
- Költséghatékony automatikus eszközök
- Funkcionális és adatbiztonság együttese

Oktatás:

- Szolgáltatásbiztos rendszertervezés



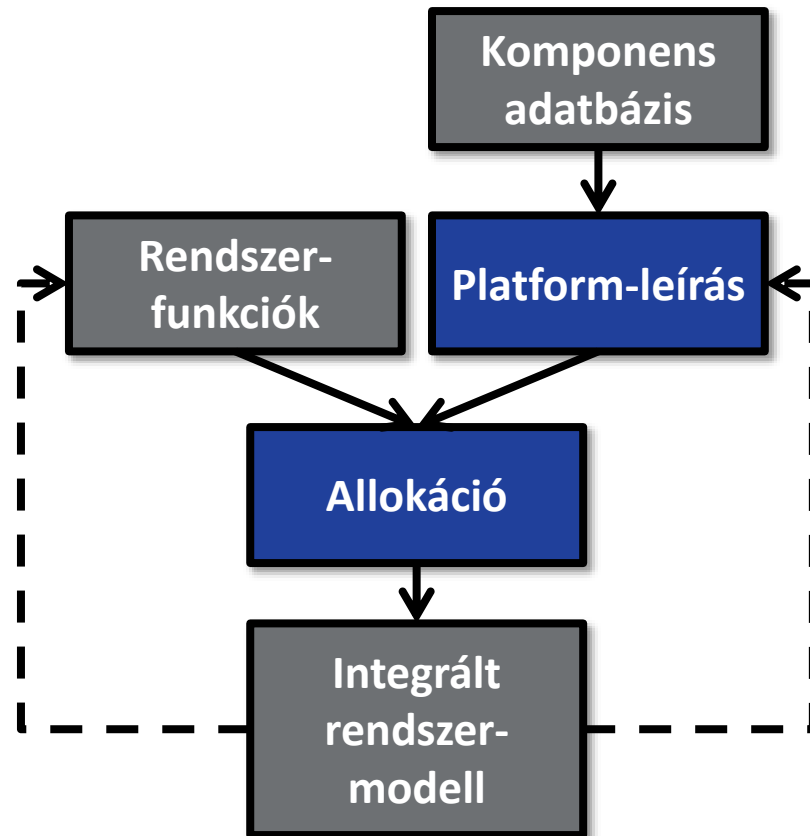
TRANS-IMA (Embraer)

Megoldandó probléma: szoftverkomponensek allokációja egyedileg tervezett (ARINC 653 szabvány kompatibilis) integrált moduláris repülőgép-platformra

Embraer: a világ 3. legnagyobb polgári repülőgépgyártója

Kihívások:

- Létező komponens adatbázisok támogatása
- Kommunikációs architektúra generálása
- Nyomonkövethetőség
- MATLAB Simulink – Eclipse integráció

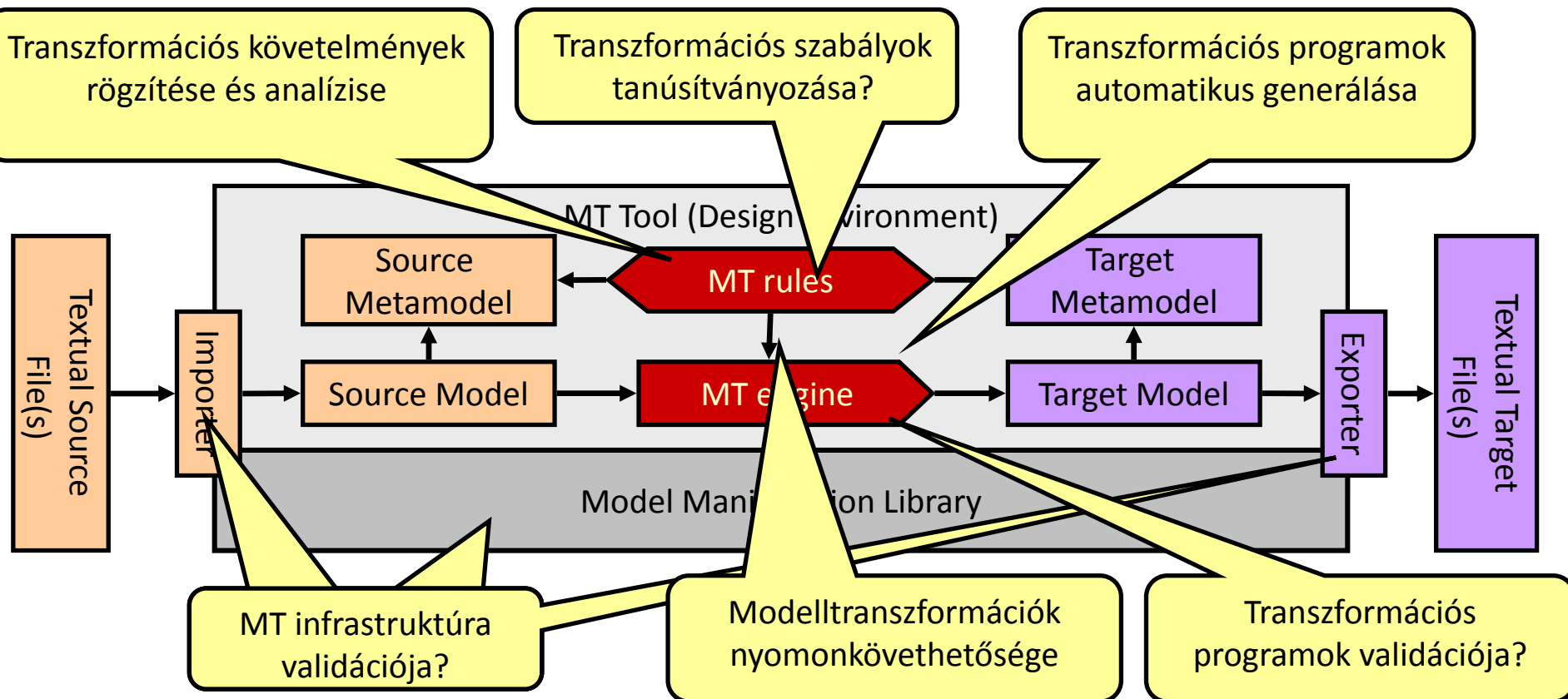


CERTIMOT

Megoldandó probléma:

olyan modelltranszformációk szisztematikus tervezése és analízise, amely összehangolható a kritikus beágyazott rendszerek tanúsítványozási folyamatainak követelményeivel

Modell → bizonyított helyesség



Nyílt forráskódú Eclipse projektek (EMF-IncQuery)

Megoldandó probléma: jólformáltsági kényszerek inkrementális validációja ipari méretű modellek felett Eclipse (EMF) környezetben

Eclipse: a beágyazott rendszerek modellezésére és tervezésére legelterjedtebben használt ipari szoftver keretrendszer

Kihívások:

- AUTOSAR, ARINC 653: >500 kényszer
- Modellek mérete: > 1 millió elem
- Válaszidő: < 1 sec
- Újrafelhasználható kényszerdefiníciók?
- Eszközök összehasonlítása

Eredmények:

- RETE algoritmus adaptációja
- Válaszidő: < 100 msec
- Modellméret: > 4 millió

