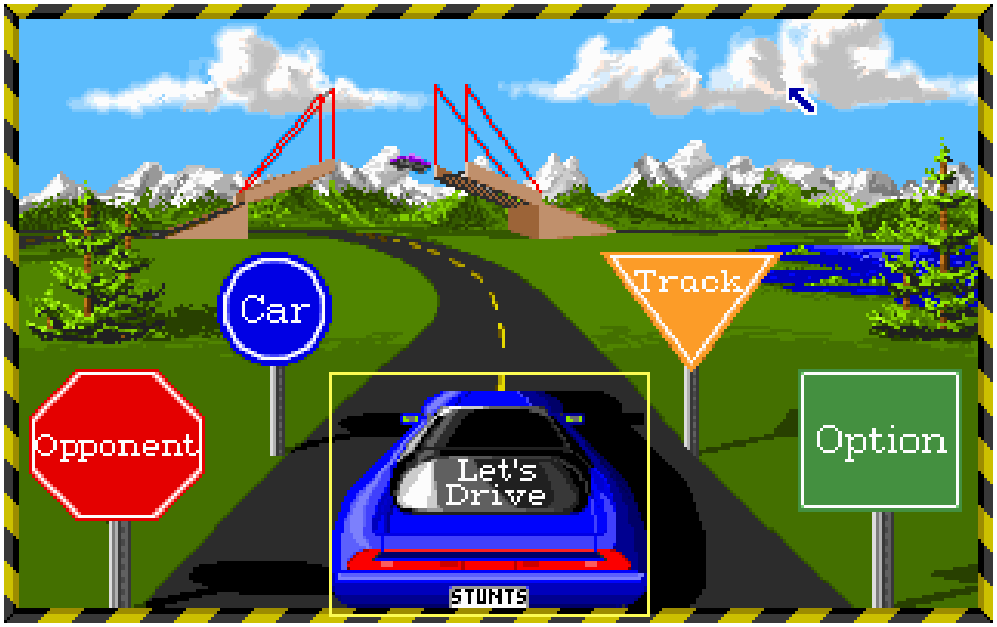
Modellezési alapismeretek

Rendszermodellezés

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

“Motiváció”



- Stunts autóverseny, 1990.
  - Distinctive Games/Brøderbund Software
  - https://en.wikipedia.org/wiki/Stunts_(video_game)
- Versenyzés + pályaszerkesztés (!)
  - Tkp. egy “szakterület specifikus modell”

“Motiváció”: mire jó egy modell?

Végig lehet-e menni egy pályán?
Ha igen, melyik kocsival?
Egyértelmű-e a pálya meghatározása?

Milyen elemek kombinálhatóak?
Milyen pályák állíthatóak elő?

Járható-e egy pálya fordított irányban is?

“Valósághű”-e egy pálya?
Alkalmas-e versenyzésre?

Szeretnék új típusú pályaelemet, mi kell ehhez?

Kép: http://www.abandonia.com/games/73/Stunts.htm

Tartalom

Modell és modellezés

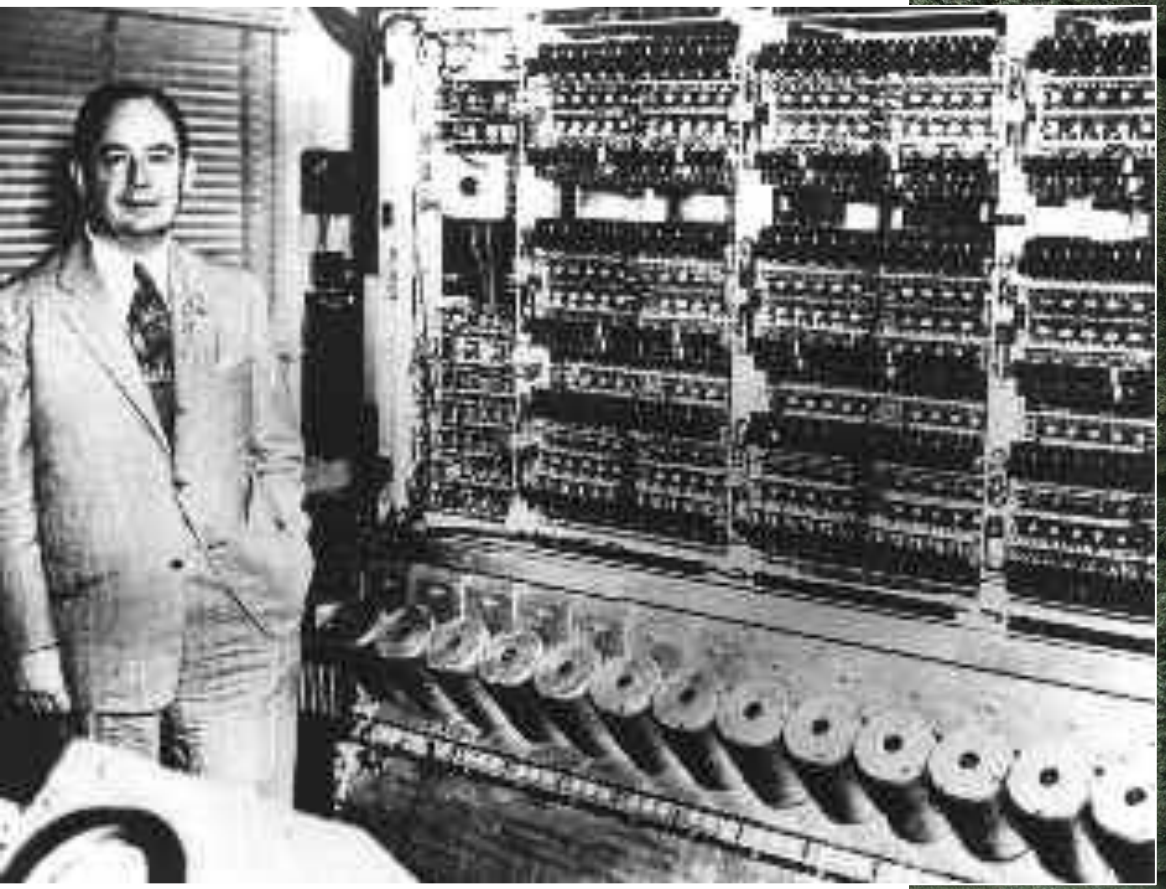
Mire használunk modelleket?

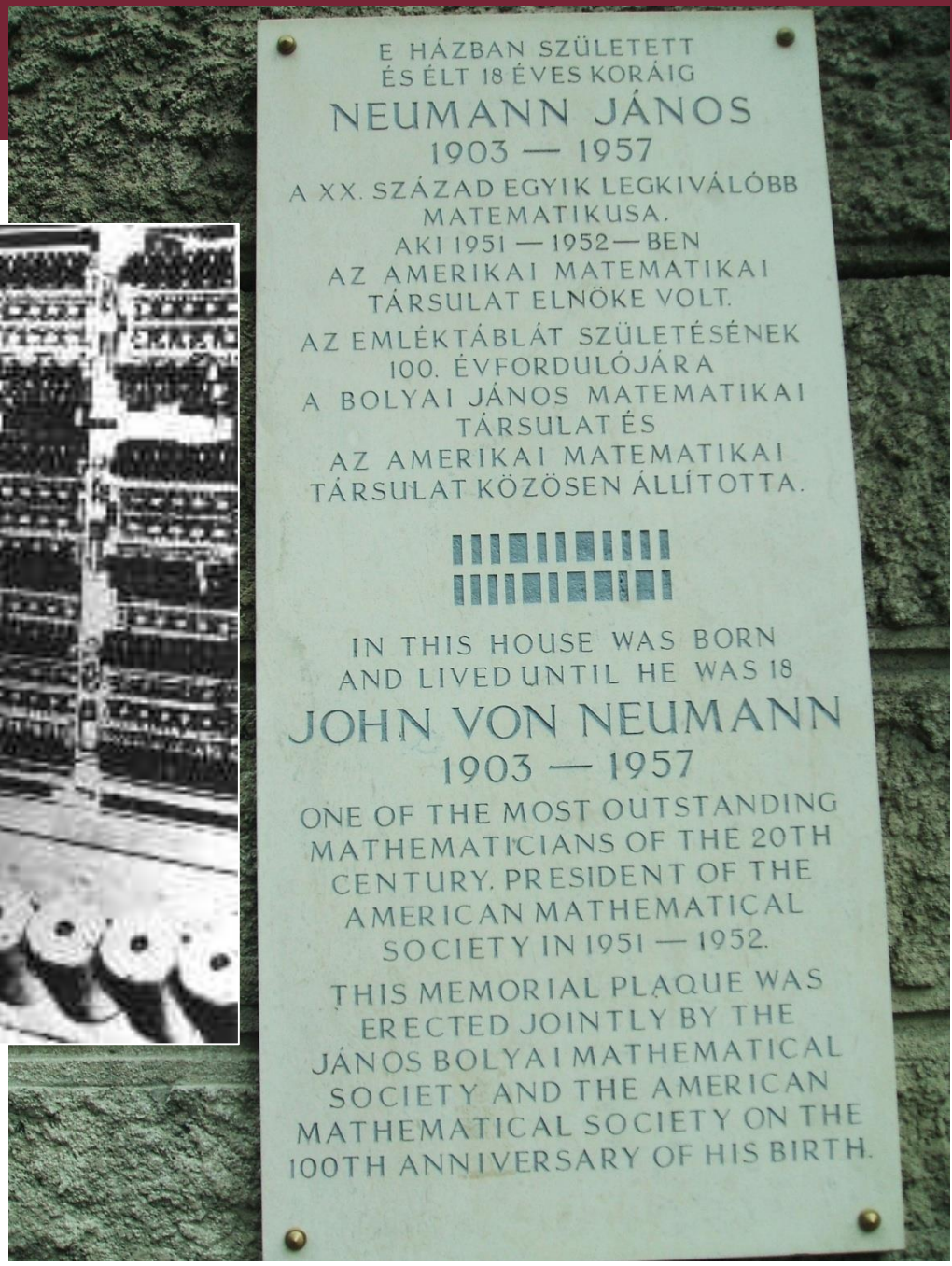
Alapfogalmak

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

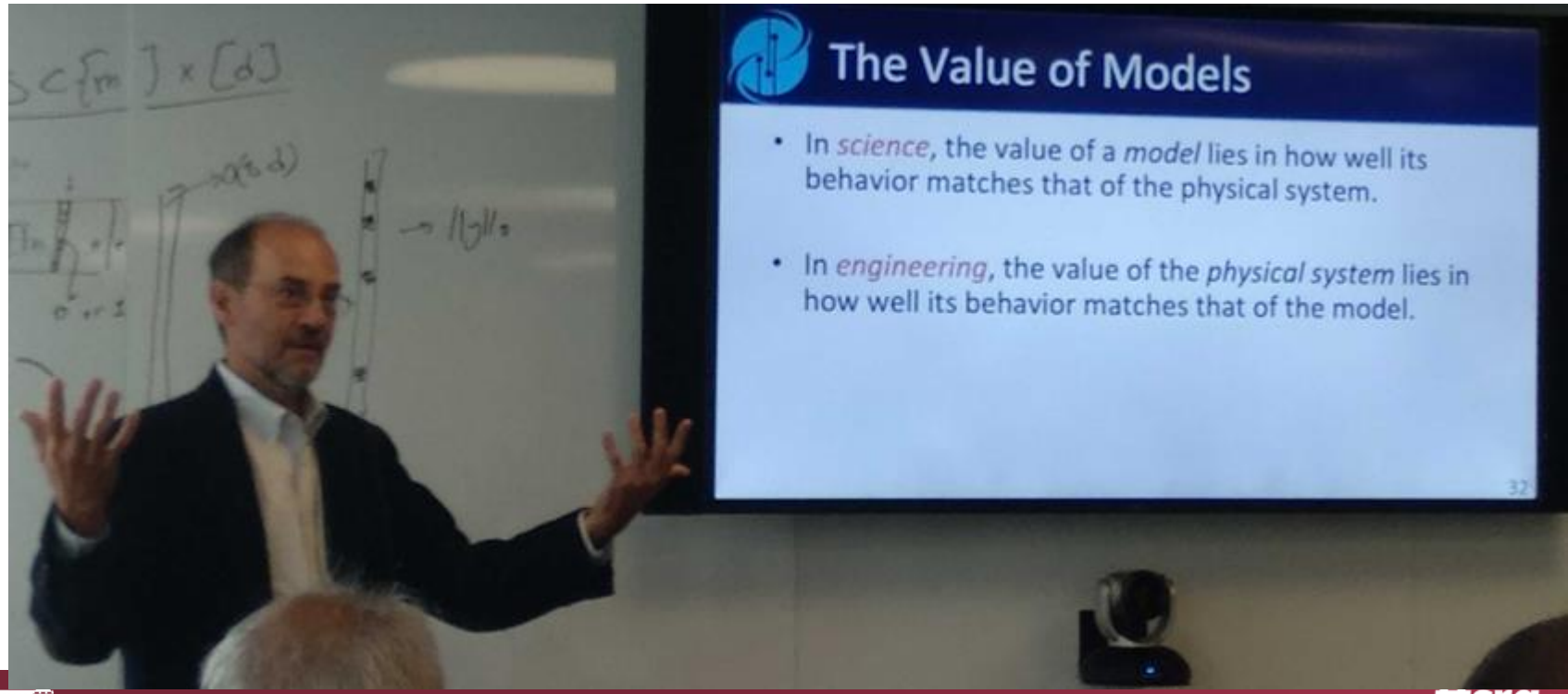



E HÁZBAN SZÜLETETT
ÉS ÉLT 18 ÉVES KORÁIG
NEUMANN JÁNOS
1903 — 1957
A XX. SZÁZAD EGYIK LEGKIVÁLÓBB
MATEMATIKUSA.
AKI 1951 — 1952 — BEN
AZ AMERIKAI MATEMATIKAI
TÁRSULAT ELNÖKE VOLT.
AZ EMLÉKTÁBLÁT SZÜLETÉSÉNEK
100. ÉVFORDULÓJÁRA
A BOLYAI JÁNOS MATEMATIKAI
TÁRSULAT ÉS
AZ AMERIKAI MATEMATIKAI
TÁRSULAT KÖZÖSEN ÁLLÍTOTTA.
IN THIS HOUSE WAS BORN
AND LIVED UNTIL HE WAS 18
JOHN VON NEUMANN
1903 — 1957
ONE OF THE MOST OUTSTANDING
MATHEMATICIANS OF THE 20TH
CENTURY. PRESIDENT OF THE
AMERICAN MATHEMATICAL
SOCIETY IN 1951 — 1952.
THIS MEMORIAL PLAQUE WAS
ERECTED JOINTLY BY THE
JÁNOS BOLYAI MATHEMATICAL
SOCIETY AND THE AMERICAN
MATHEMATICAL SOCIETY ON THE
100TH ANNIVERSARY OF HIS BIRTH.

Tudományos és mérnöki modellek

- Tudományos modell - deskriptív (leíró)
  - Newton törvénye nem illeszkedik fekete lyukra

  → rossz modell
- Mérnöki (tervezési) modell - preskriptív
  - Nem illeszkedik → "valóság" rossz



Mi a modell?

- Egy valós vagy hipotetikus világ (a „*rendszer*”)
  - egy **részének**
  - **egyszerűsített képe**, amely
  - a rendszert **helyettesíti** bizonyos megfontolásokban
- Döntések:
  - **A világ mely része**?
  - **Mit hanyagol el**?
  - **Hogy feleltethető meg a világnak**?
- Haszna
  - kisebb (véges)
  - áttekinthetőbb

Mikor *lehet* és *érdemes* felhasználni?

Mi NEM a modell?

- A modell nem a valóság!

- A modell nem a diagram.
  - az csak egy nézet…

Modell vs. valóság: értelmezhetőség?

Modell-
analízis

Modell-
analízis

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:
    - Gyár, ahol mindig van elég anyag
    - Minden rendelés elkészül határidőre
  - Célfüggvény:
    - Költségminimum
- Rendkívüli eset
  - Peremfeltétel
    - Anyaghiány
  - Célfüggvény:
    1. Minél több rendelés határidőre
    2. Költségminimum

Matematikai modell vs. valóság

- Minden modell:
  zárt világ
  - Hatások, faktorok
  - Paraméterek
  - Érvényesség
- A modell
  bizonytalan
  ezen a
- Nem
  előre
  - **Embe**
  - **Gener**
- *Megoldá*

- Normál műkö
  - Perem
    elég
    l
    set
    feltétel
    Anyaghiány
  - Célfüggvény:
    1. Minél több rendelés határidőre
    2. Költségminimum

A modell egy(néhány) kérdés megválaszolásának segédeszköze !

Példa: biztonságkritikus SW

- Repülőgép fékezése: kerékfék + sugárfordító

1993 Varsó: Lufthansa 2904

- (SW) védelem:

(*mindkét főfutó terhelt*)

OR

(*egy kerék gyorsan forog*)

→ (*gép a leszállópályán*)

→ **(PILÓTA FÉKEZHET)**

Modell minősége

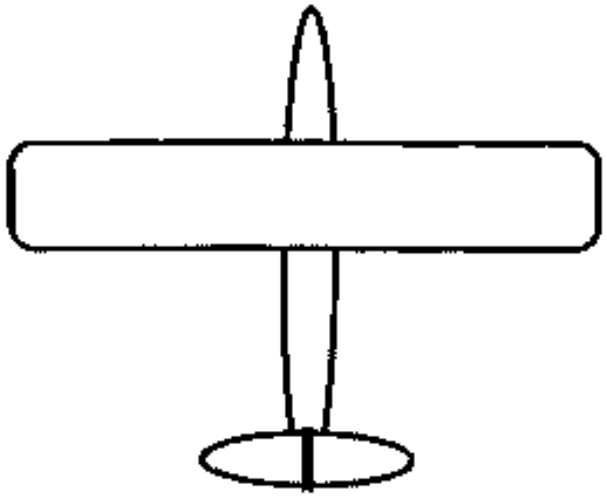
- A **valóság**: nyílt világ ←→
  A **modell:** zárt világ
- Valósághűség:
  - Valószínű + kritikus esetek

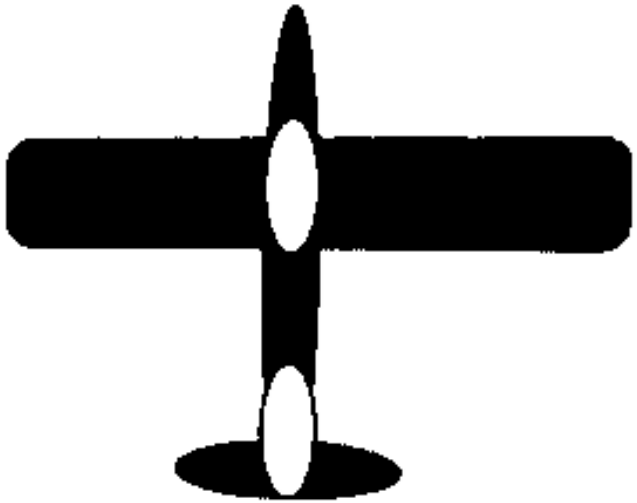
Modell minősége

- A **valóság**: nyílt világ ←→
  A **modell:** zárt világ
- Valósághűség:
  - Valószínű + kr
    esetek

Rossz modell tökéletes implementációja halálos!

Páncélozás? → Wald Abraham





Páncélozás? → Wald Abraham

A modell csak a környezettel együtt hasznos!

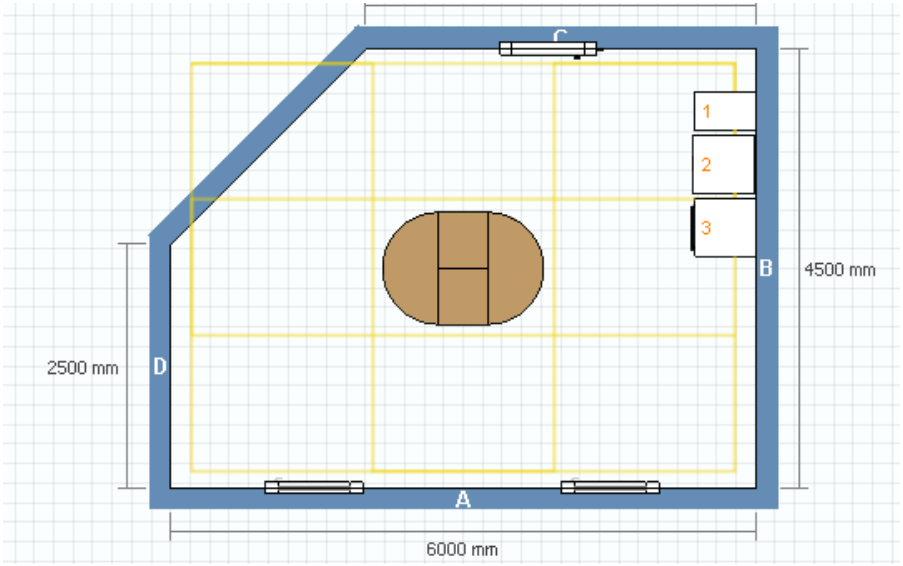
Modell és modellezés

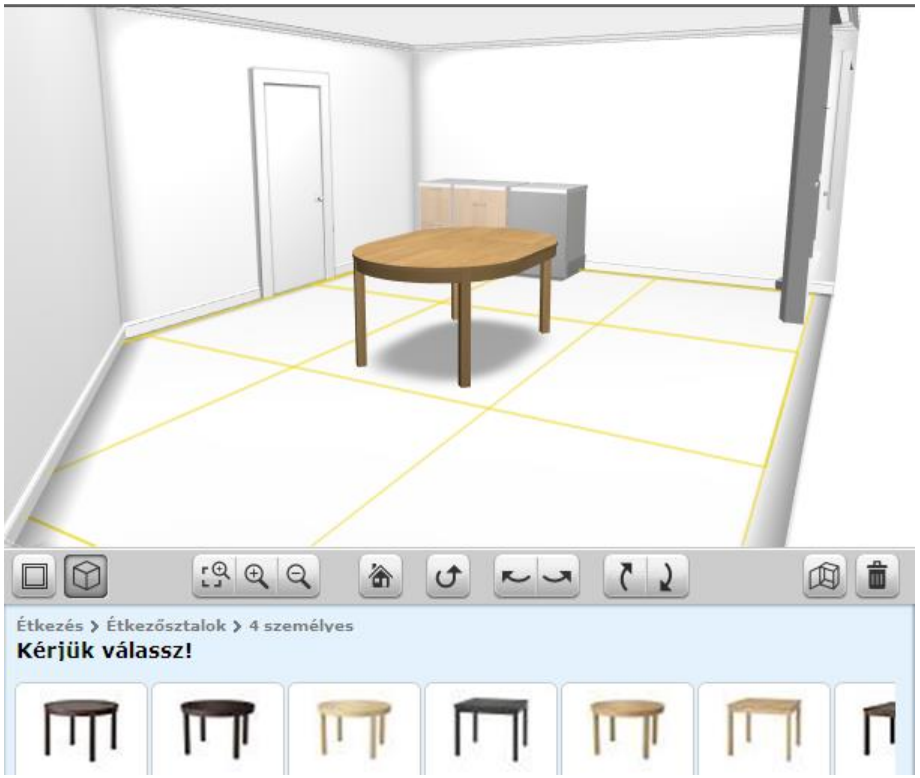
Mire használunk modelleket?

Alapfogalmak

Modellezés a gyakorlati életben?

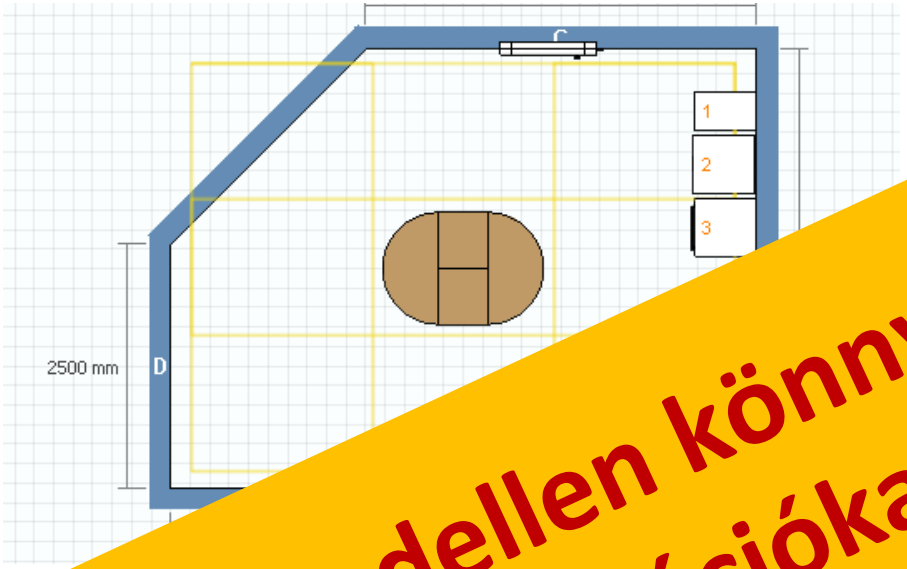
Pl.: [svéd cég] webes konyhatervezője





Modellezés a gyakorlati életben?

Pl.: [svéd cég] webes konyhatervezője

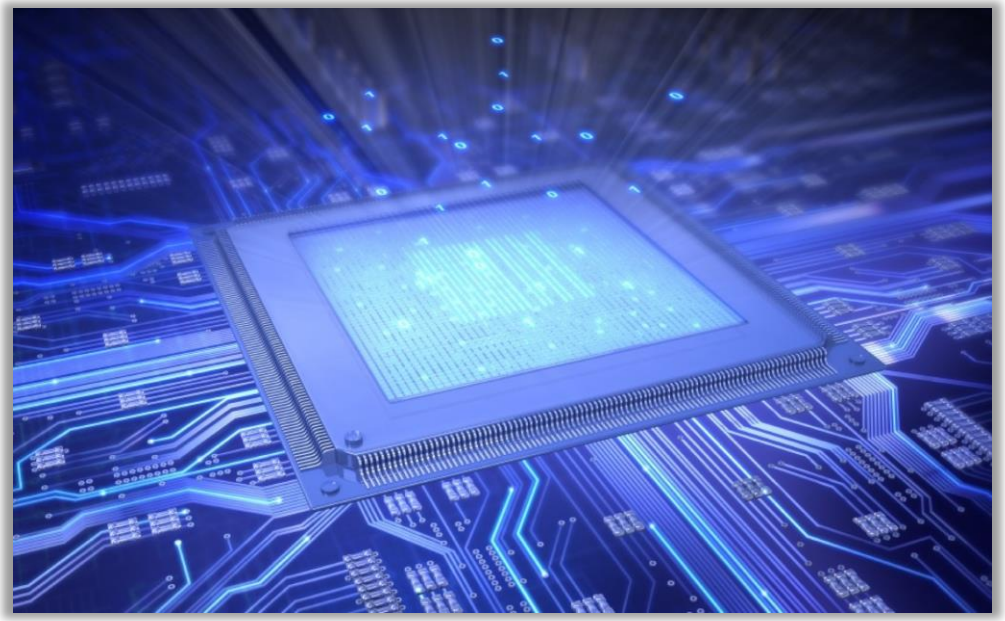


Modellen könnyebb különböző variációkat kipróbálni, (mint pl. bútortologatással)

Ez is modellezési nyelv! ☺

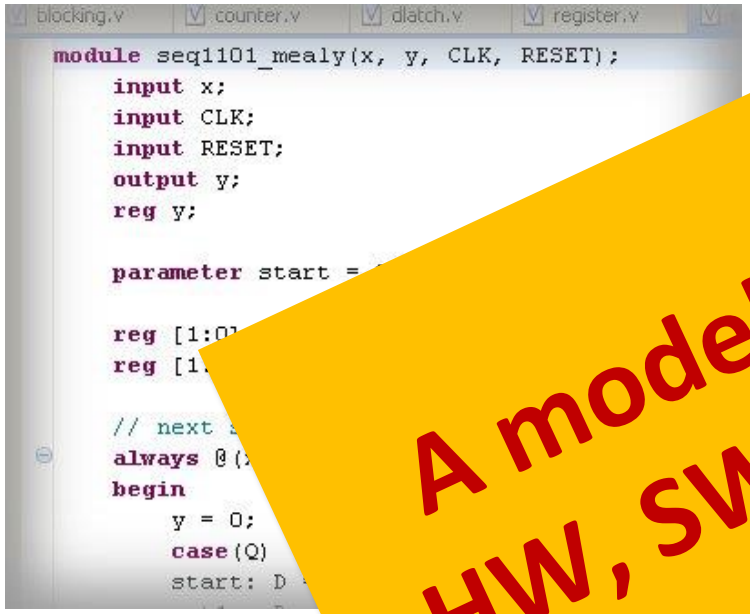
- Verilog – szakterület specifikus, hardver leíró

```
blocking.v | counter.v | dlatch.v | register.v
module seq1101_mealy(x, y, CLK, RESET);
    input x;
    input CLK;
    input RESET;
    output y;
    reg y;

    parameter start = 2'b00, got1 = 2'b01, got1

    reg [1:0] Q;     // state variables
    reg [1:0] D;     // next state logic output

    // next state logic
    always @(x or Q)
    begin
        y = 0;
        case(Q)
        start: D = x ? got1  : start;
```



Ez is modellezési nyelv…

- Verilog – szakterület specifikus, hardver leíró



A modell lehet rendszer, HW, SW vagy bármi más…

Mi értelme van modellezni?

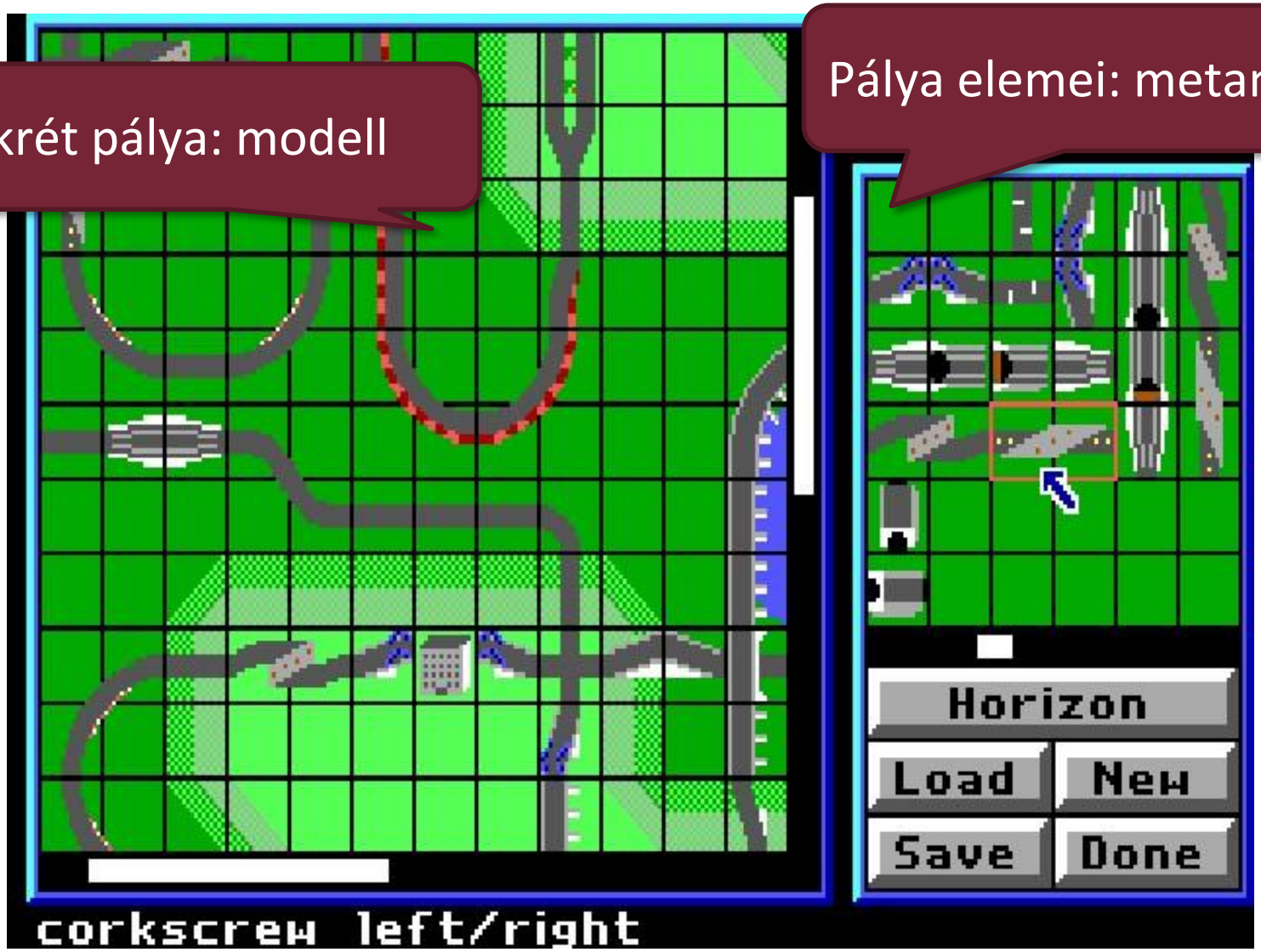
- Én szoftvert készítek. Kell-e modelleznem is?
  - Már így is ezt teszed!
    - (A szoftver forráskódja is egyfajta modell…)
    - Ami fontosabb: **mentális modellek**
- Mikor kell kifejezetten *lejegyezni* a modelleket?
  - Szerepe: **kommunikáció**
    - Ember → ember
    - Ember → gép
    - Gép → 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: ***hogyan írom le a modellt? "pl. C nyelven"***
  - „Matematikai struktúra": absztrakt szintaxis
  - Jelölés: konkrét szintaxis
    - rajzjelek / szöveges formátum
- Szemantika: ***mit jelent a modell? "i++"***
- Kényszerfeltételek, korlátozások
  - Szintaktikai helyesség, jólformáltság
  - Tervezési konvenciók (csapatonként változhat)

Biztos ez a legjobb?

A modellezési nyelv modellje



Alapfogalmak - Metamodellezés

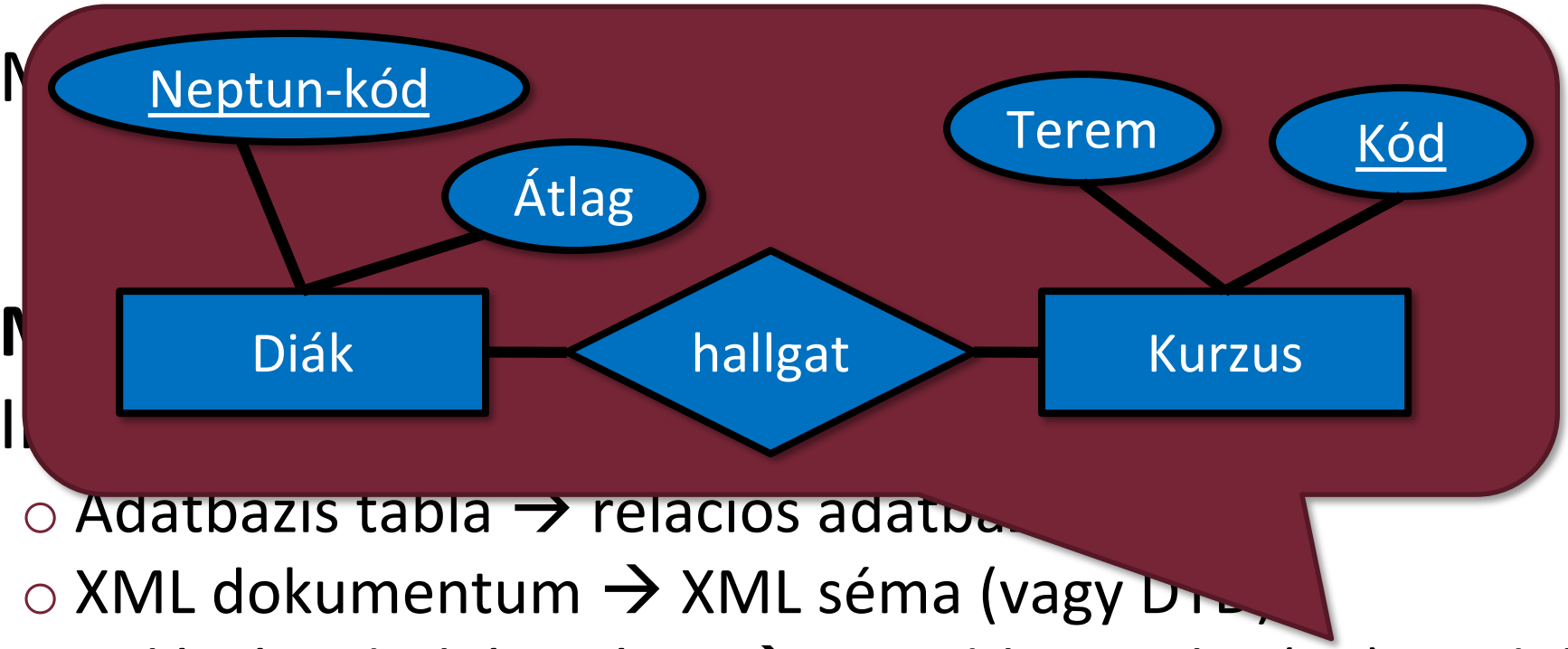
- Modellezési nyelv: milyen típusú elemei vannak?
  - …és milyen kapcsolatban állhatnak ezek az elemek?
  - …és ezeknek a típusoknak mik a viszonya egymáshoz?
- **Metamodell** = egy modellezési nyelv modellje
- Illusztrációk
  - Adatbázis tábla
  - XML dokumentum
  - Relációs adatbázisséma
  - C++ objektumok
  - C++ program

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
  - Adatbázis tábla → relációs adatbázisséma
  - XML dokumentum → XML séma (vagy DTD)
  - Relációs adatbázisséma → Egyed-kapcsolat (ER) modell
  - C++ objektumok → C++ osztályok
  - C++ program → C++ programnyelv specifikáció

Alapfogalmak - Metamodellezés

- M
- M
- I
  - Adatbázis tábla → relációs adatb
  - XML dokumentum → XML séma (vagy D
  - Relációs adatbázisséma → Egyed-kapcsolat (ER) modell
  - C++ objektumok → C++ osztályok
  - C++ program → C++ programnyelv specifikáció



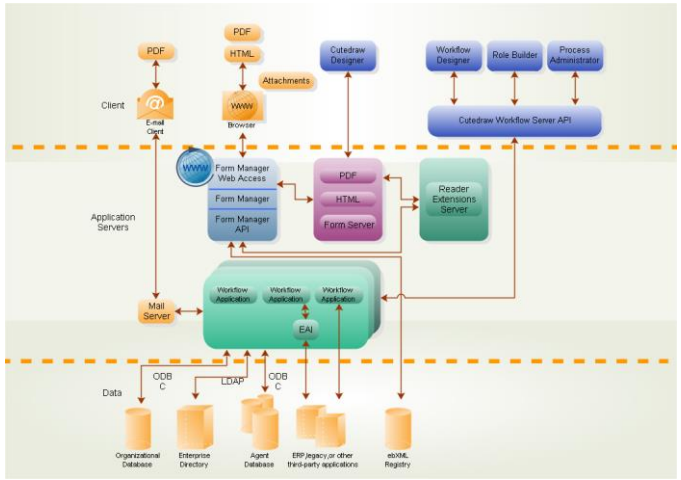
Alapfogalmak - Metamodellezés

- Modellezési nyelv: milyen típusú elem…ak?
  - …és milyen kapcsolatban állhatnak…?
  - …és ezeknek a típusoknak m…?
- **Metamodell** = egy m…
- Illusztrációk
  - Adatb…
  - …pcsolat (ER) modell
  - …yok
  - …ogramnyelv specifikáció

Az általunk használt programok többsége modellezőeszköz

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”

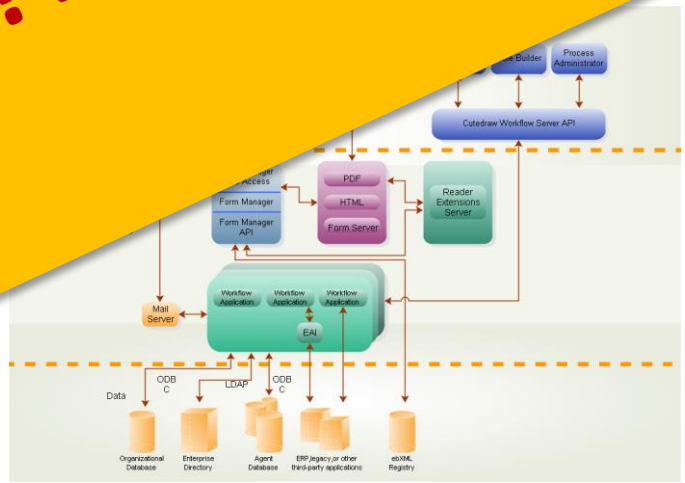


Felhasználás – Dokumentáció

- A modell egyszerűbb
  - könnyebben elmondható, mint a teljes valóság
  - fokozatosan finomítható (ld. később)
- Kommunikáció, szemléltet
  - de

- Gondolkod zés támo ok

Egy rajz = ??? oldal szöveg ?



„A kód dokumentáció is egyben”

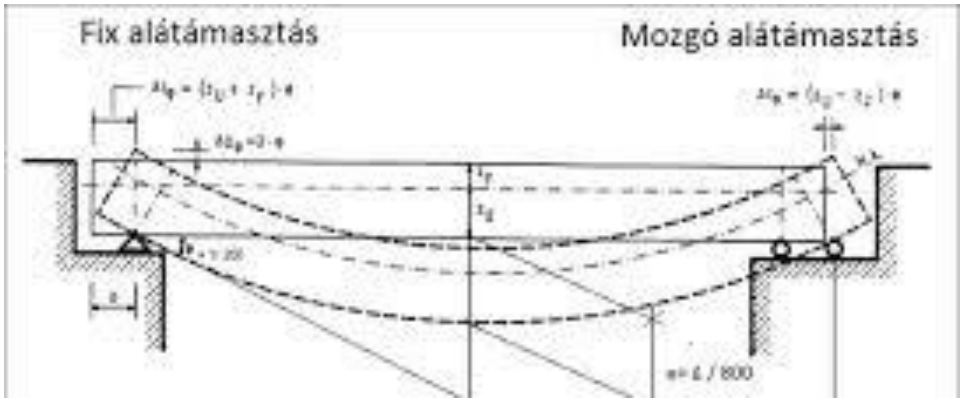
Margaret Hamilton

Felhasználás - Elemzés

- Emberi erővel vagy (részben) automatizáltan
- Módszer
  - Statikus ellenőrzés
  - Tesztelés támogatása
  - Dinamikus modellellenőrzés
  - Formális állítások bizonyításával
- Célok
  - Megengedett és elvárt viselkedés vizsgálata

  → Modell helyes legyen
  - Jellemzők számítása (pl. ütemezés)

  → Rendszer követelményei

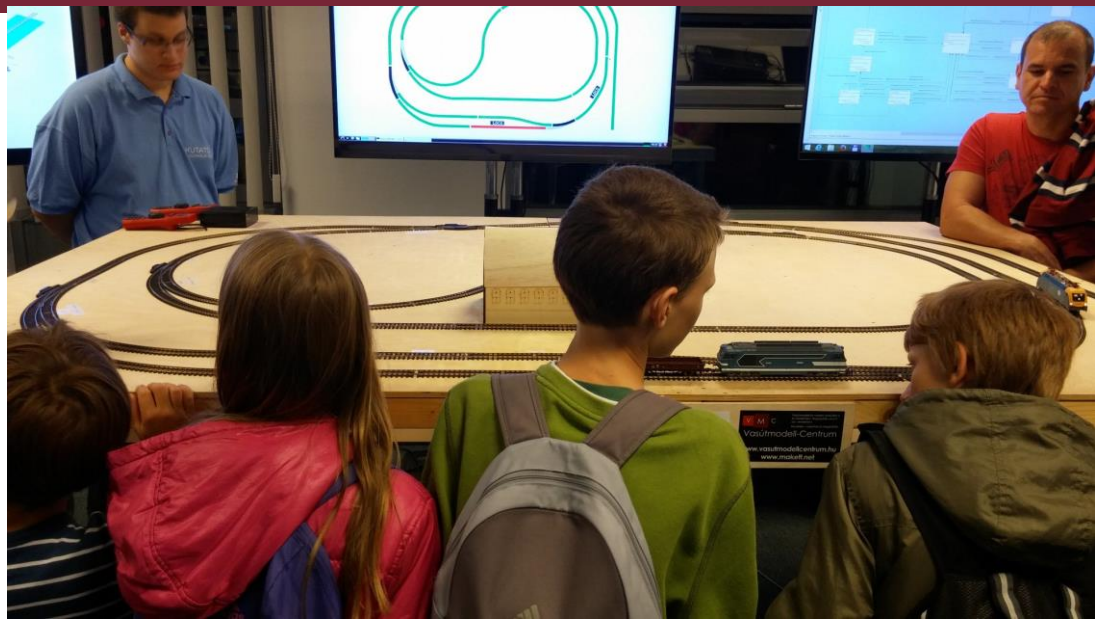


Felhasználás - Elemzés

- Emberi erővel vagy (részben) automatizáltan
- Módszer
  - Statikus ellenőrzés
  - Tesztelés támogatása
  - Dinamikus mod
  - Formális
    bi
- Célok
  - Meg
    yei

Modell és matematikai helyességbizonyítás

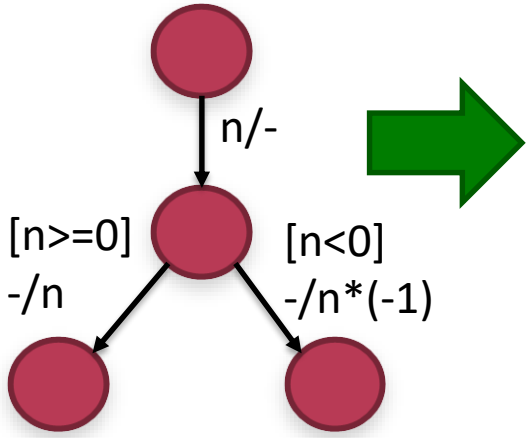
Felhasználás: szimuláció



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

Felhasználás - Származtatás

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

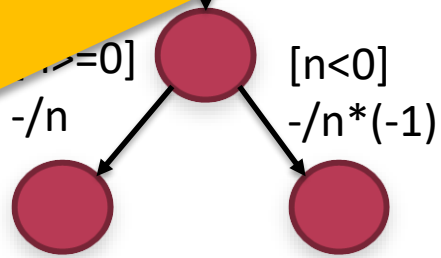



```
int n;
scanf("%d",&n);
if (n>=0) {
  printf("%d",n);
} else {
  n=n*(-1);
  printf(„%n",a);
}
```

Felhasználás - Származtatás

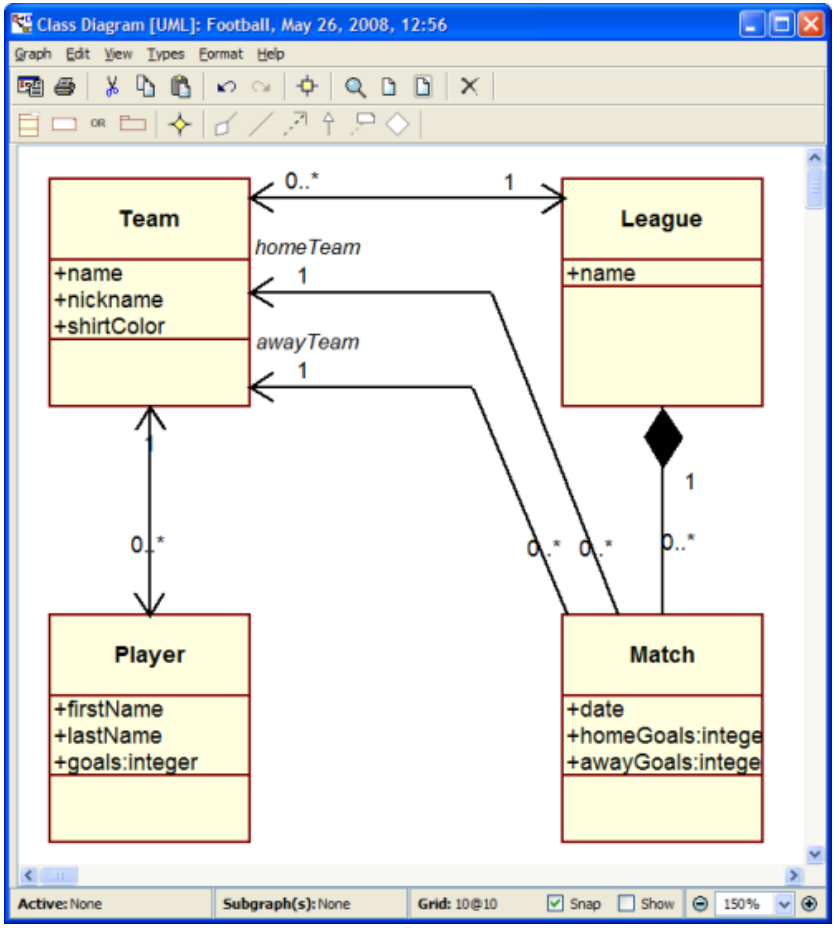
- Emberi erővel vagy (részben) automatizáltan
- Eredmény
  - programkód, analizálható nyelv, st
  - másik modell
    - finomítás, következő t
    - részaspektus
    - modellek i
  - Lehet
- P

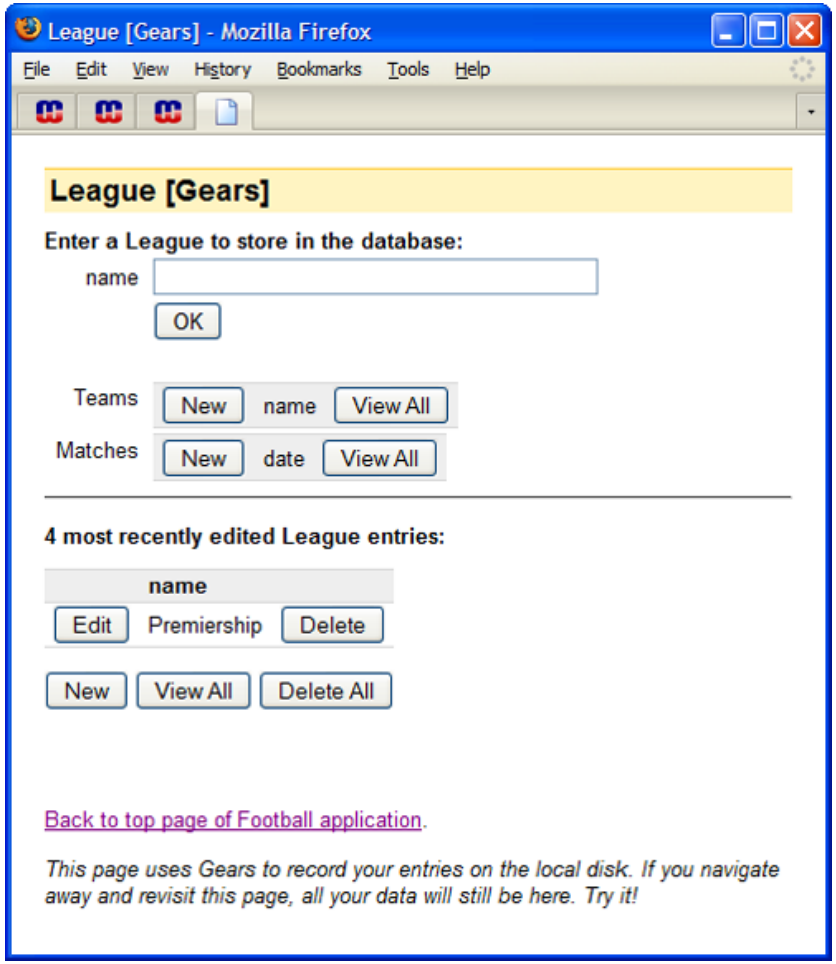
Rutin lépések automatizálhatóak
1-1 nagyságrend
termelékenység+minőség



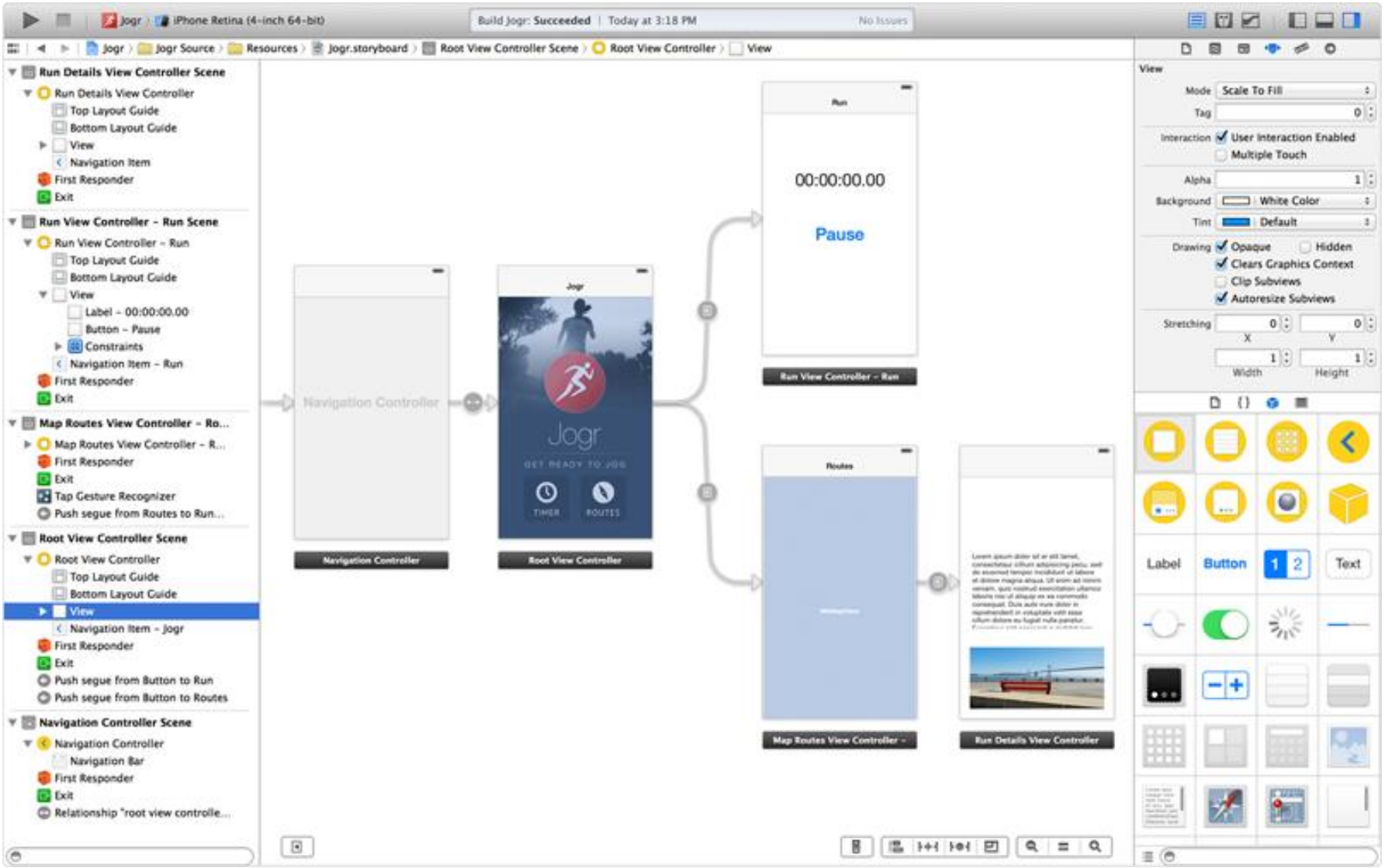

```
int n;
scanf("%d",&n);
if (n>=0) {
  printf("%d",n);
} else {
  n=n*(-1);
  printf(„%n",a);
}
```

Webalkalmazás fejlesztése



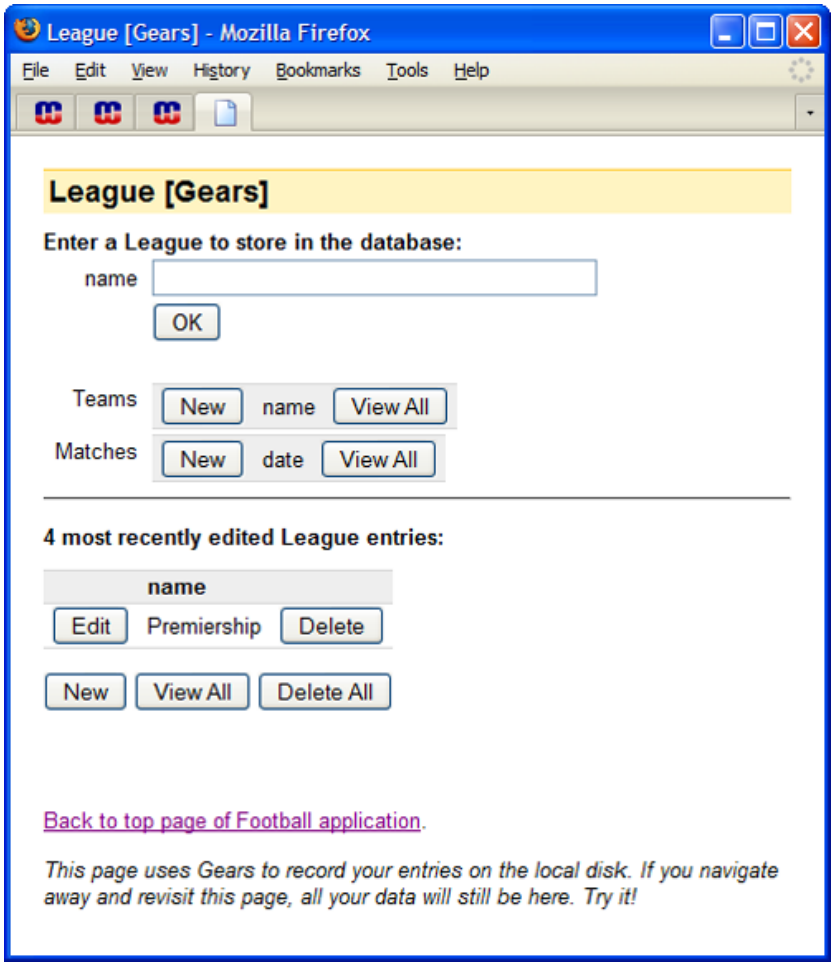


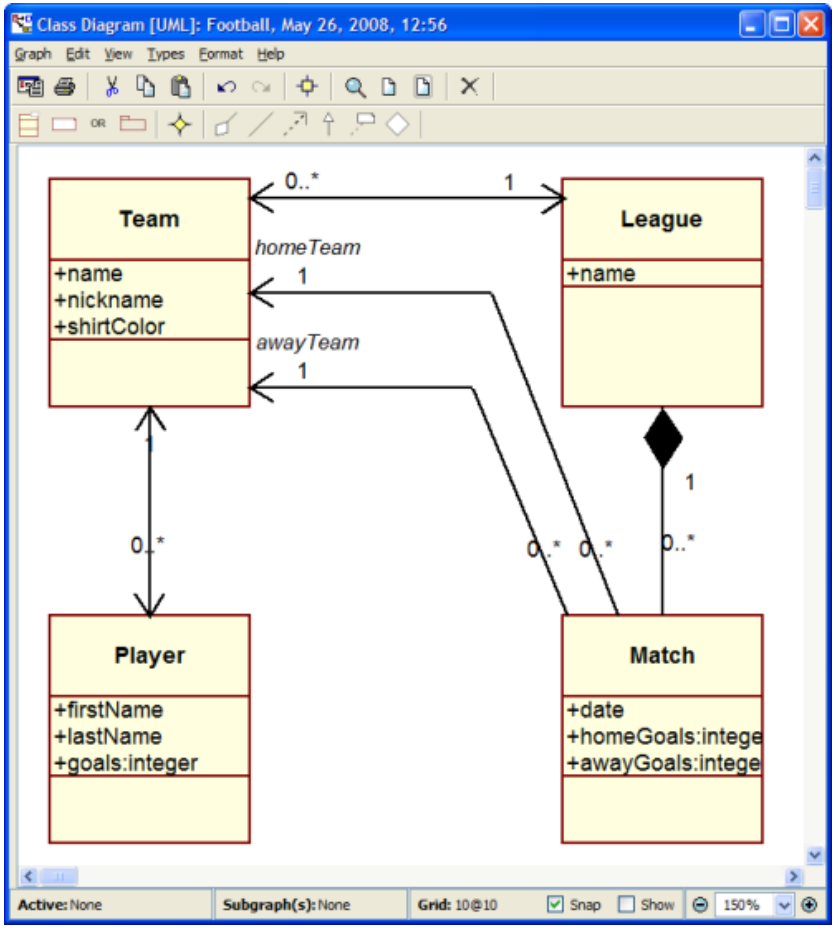
Okostelefon alkalmazás fejlesztése



Reverse engineering

- Modell visszafejtése a rendszer alapján





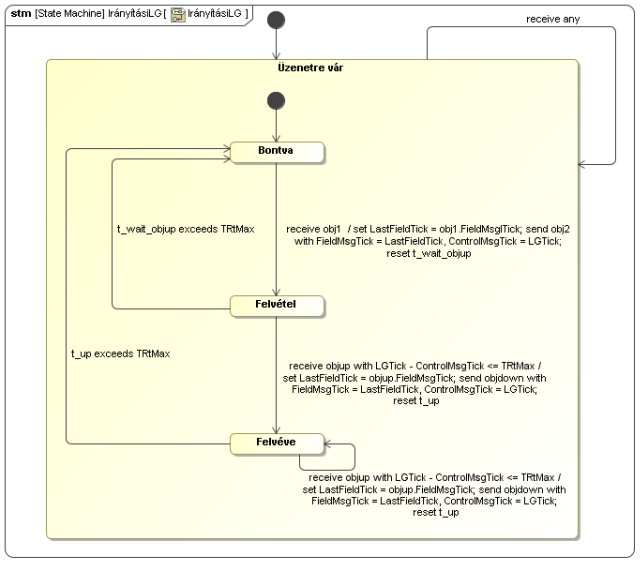
Vasúti rendszerek fejlesztése

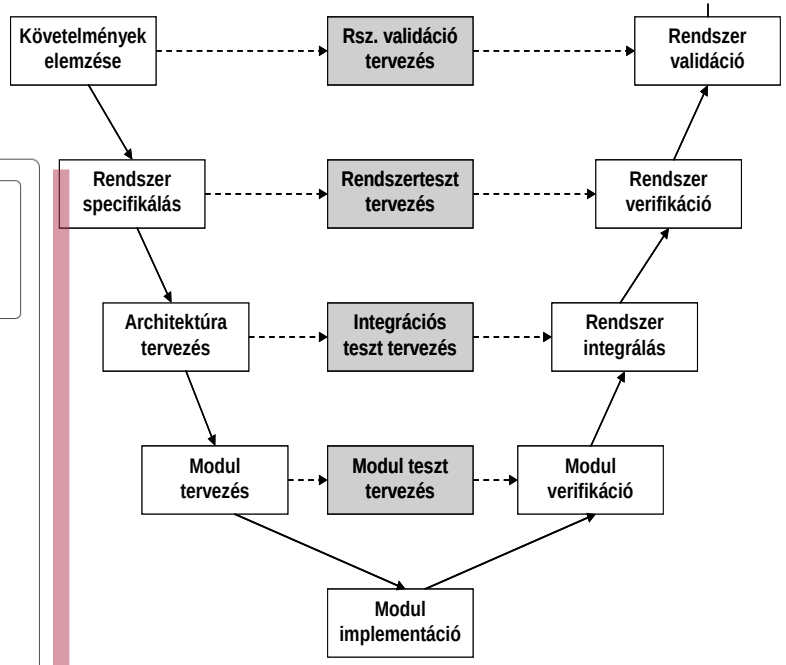
Megoldandó probléma:

SIL-4 vasúti alkalmazás fejlesztési lépéseinek és ezek eredményeinek ellenőrzése

Kihívások:

- Szisztematikus felülvizsgálat
- Modellezés és formális verifikáció





Folyamat → modell → minőségbiztosítás

Vasúti rendszerek fejlesztése

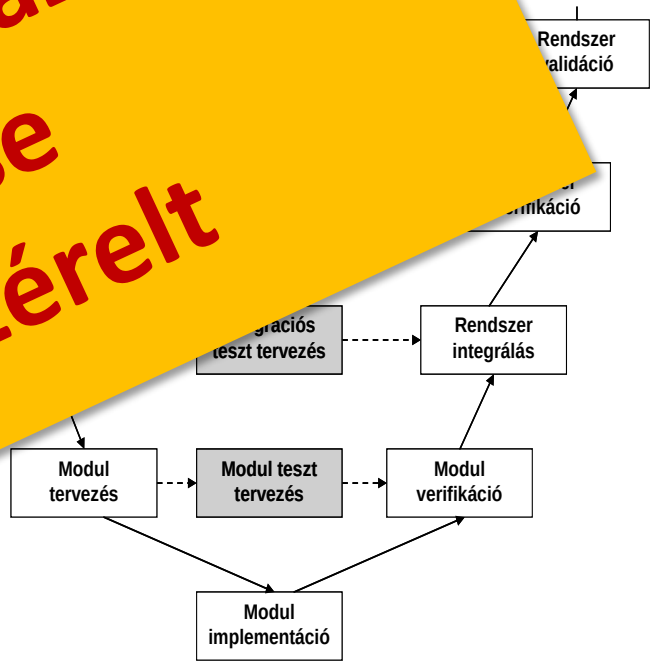
Megoldandó probléma:

SIL-4 vasúti alkalmazás fejlesztési lép… …s ezek eredményeinek ellenőrzése

Kihívások:

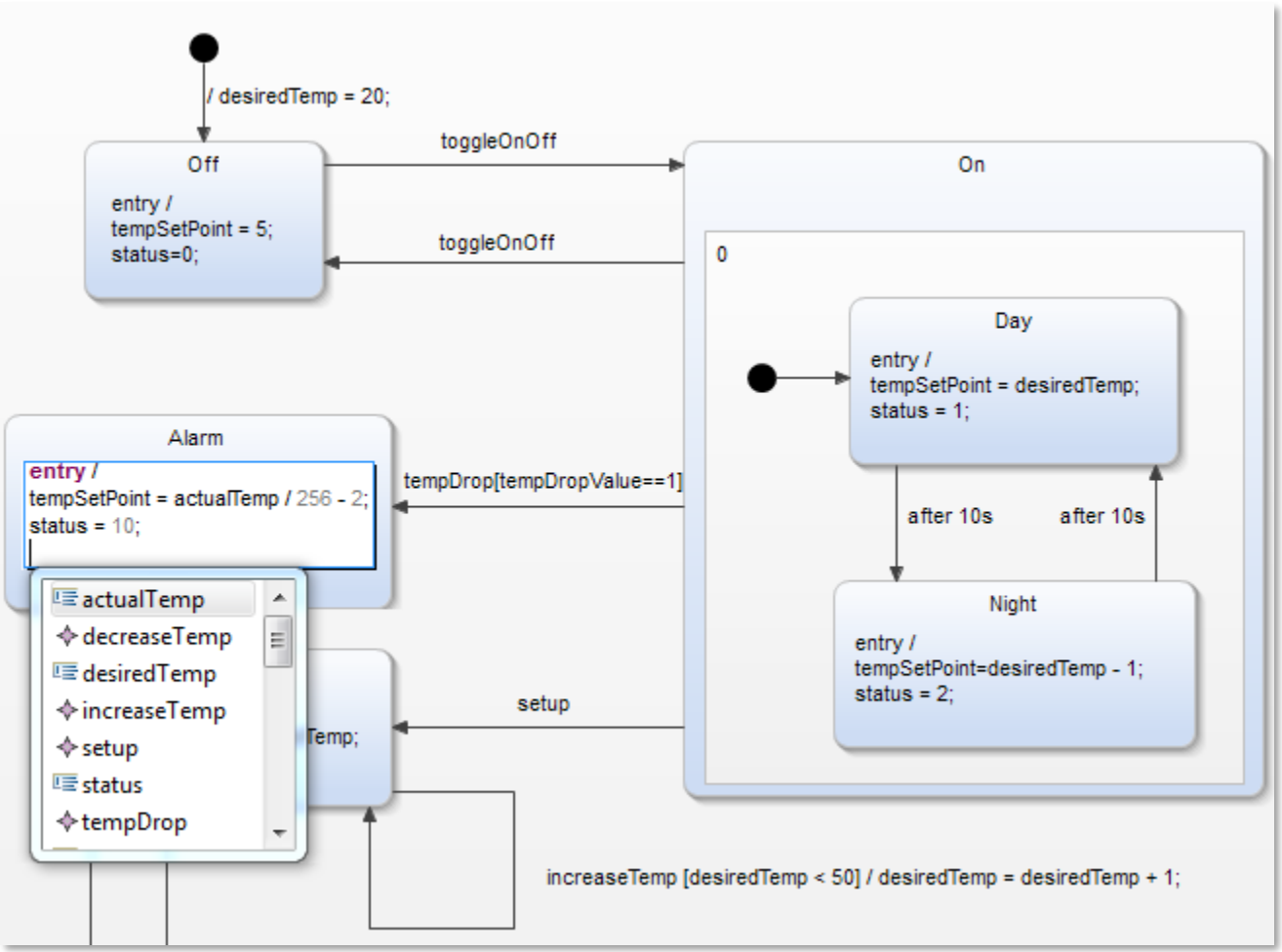
- Szisztematikus felülvizsgálat
- Modell… formális verifikáció

A kritikus alkalmazások fejlesztése modellvezérelt

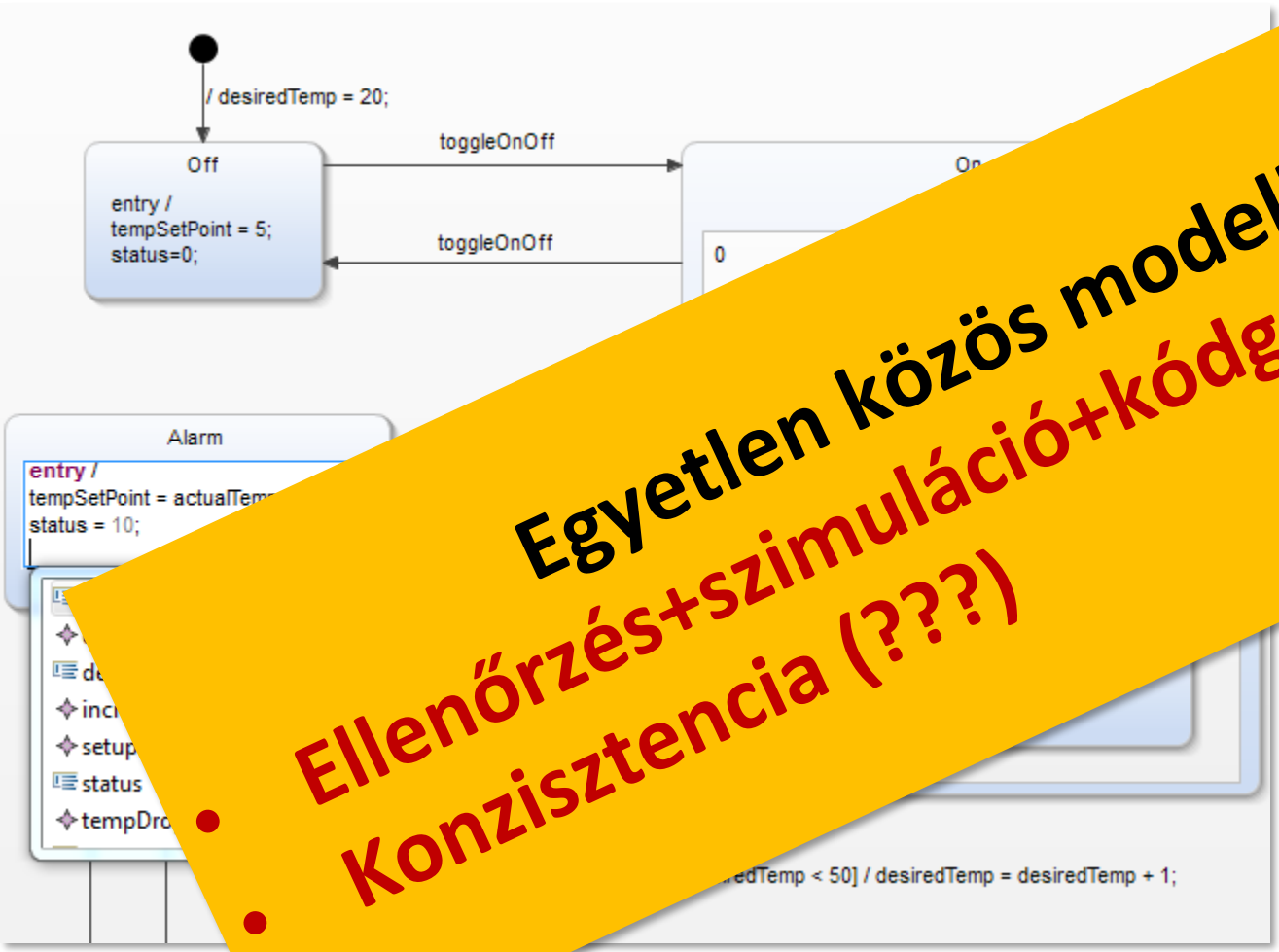


Folyamat → modell → minőségbiztosítás

Yakindu:állapotdiagram



Yakindu:állapotdiagram

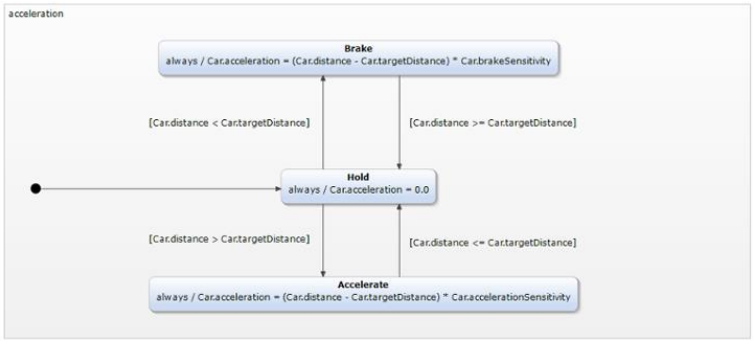


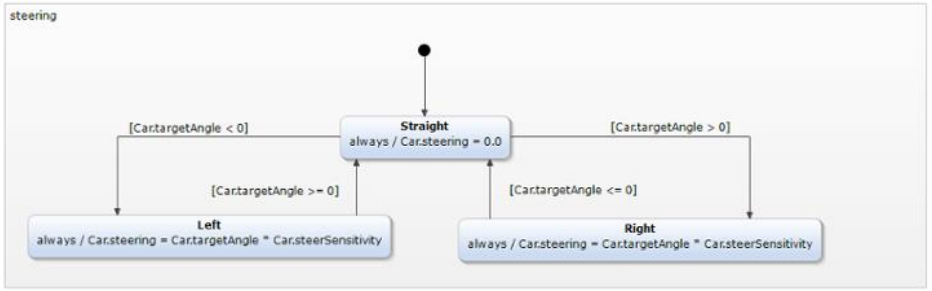
Egyetlen közös modell

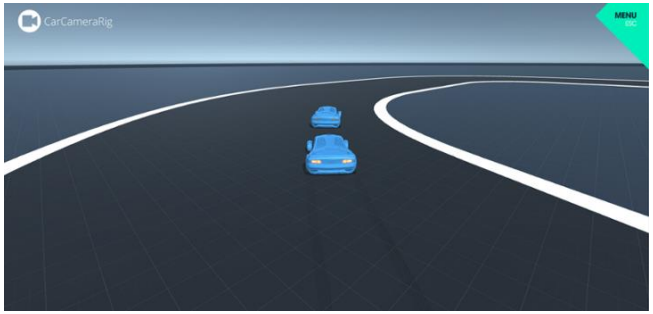
- **Ellenőrzés+szimuláció+kódgenerálás**
- **Konzisztencia (???)**

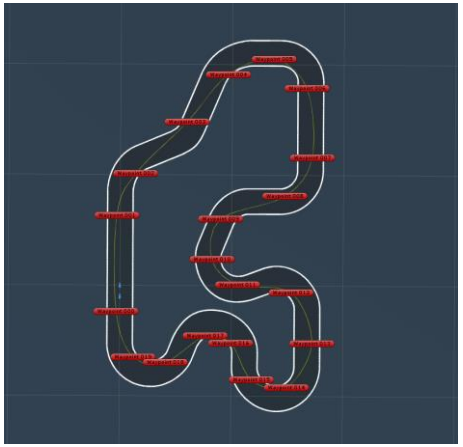
Yakindu →3D szimuláció

Járműkövető/távolságtartó rendszer interaktív 3D szimulációja









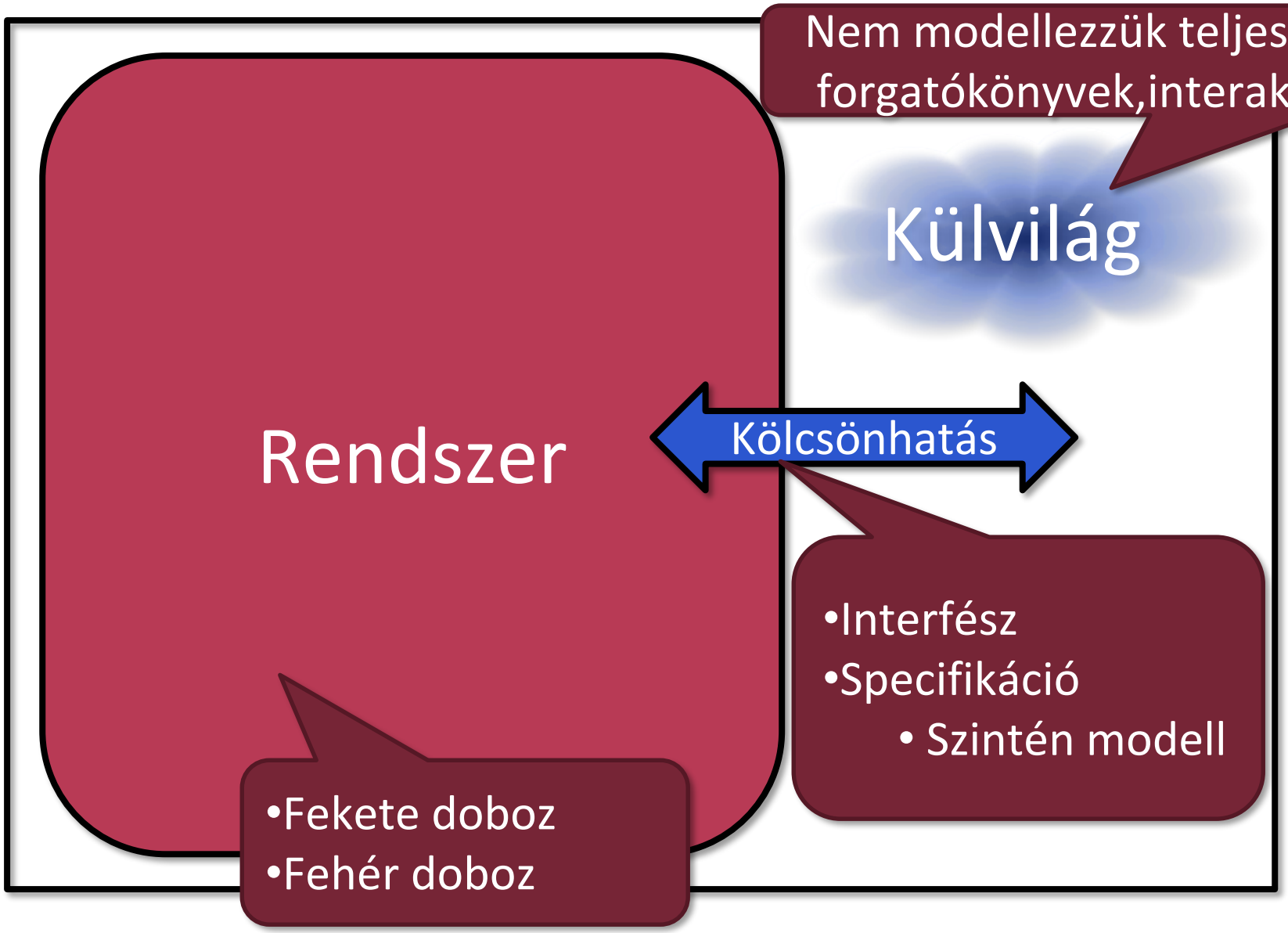
Interactive 3D Visualization and Simulation with State Machines, Benjamin Bolte, 2017.
https://blogs.itemis.com/en/interactive-3d-visualization-and-simulation-with-state-machines

Modell és modellezés

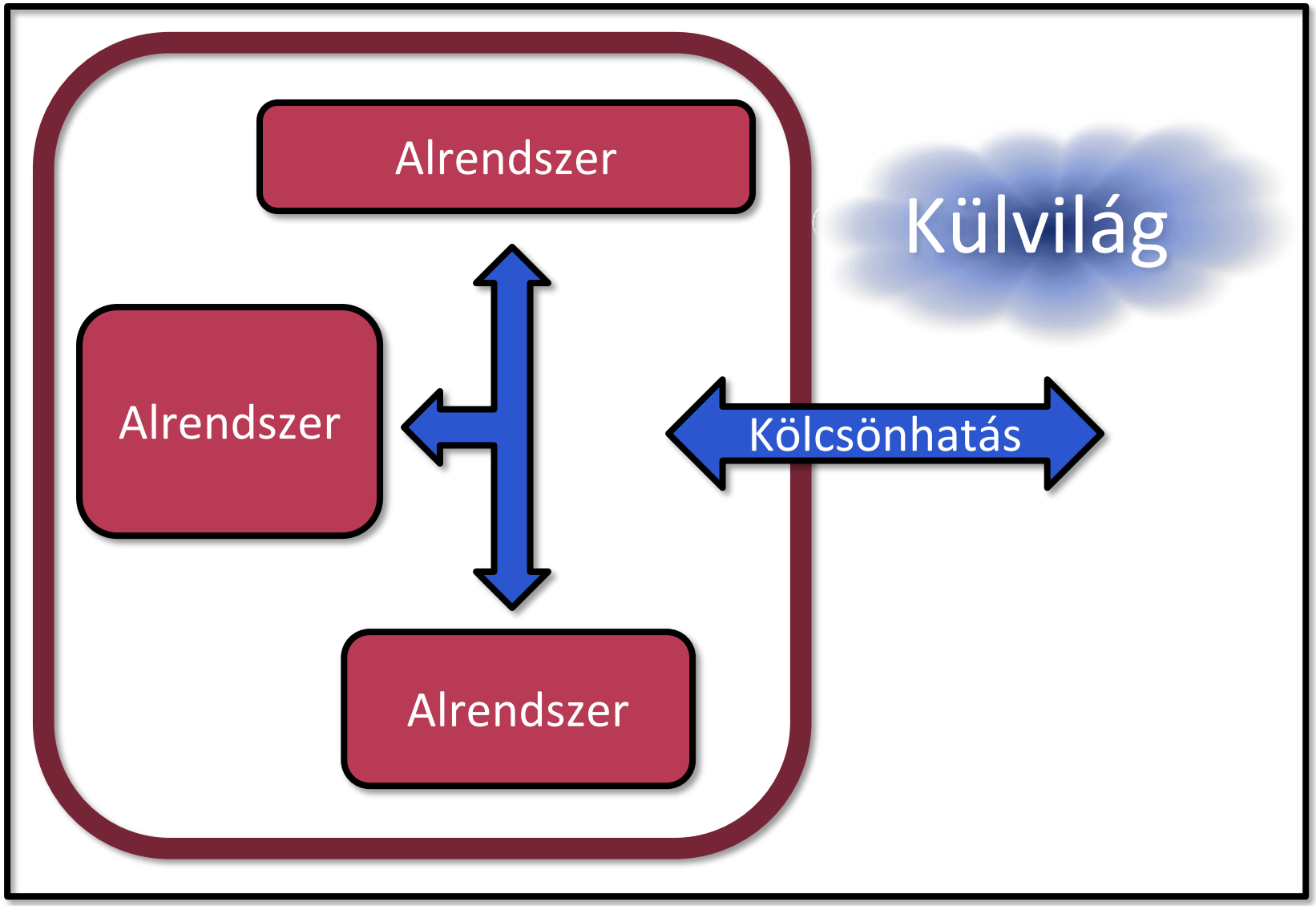
Mire használunk modelleket?

Alapfogalmak

Alapfogalmak – rendszer és külvilág



Alapfogalmak – rendszer és külvilág



Alapfogalmak – Finomítás/Absztrakció

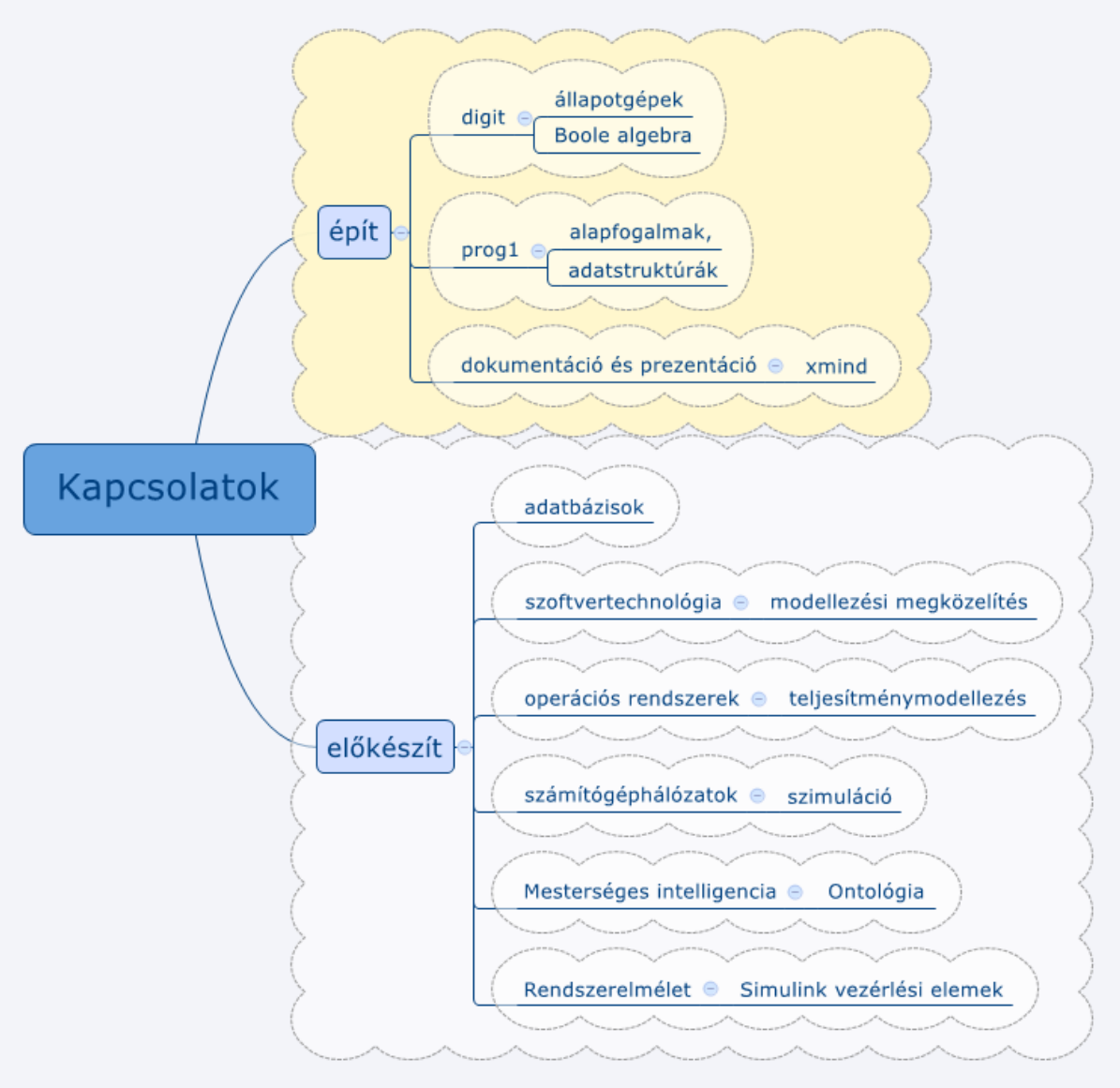
- *Finomítás*: a modell gazdagítása részletekkel…

  Pl. rendszer felbontása alrendszerekre
- …hogy az eredeti modell absztrakció maradjon

  Pl. az alrendszerekre együtt ugyanazokat az elvárásokat teljesítik
- Inverze: *absztrakció*

  Pl. alrendszerek összevonása
- Az előbbi dián egy *hierarchikus finomítás* volt
  - „dobozok kibontása”
- Finomítható más is...
  - Pl. Halmazfinomítás: változók értékkészlete
    - **Jó** / **rossz** helyett
    - **Gyors** / **átlagos** / **lassú** / **hiányos** / **veszélyes**

Modellezés a képzésben



IMSC KITEKINTÉS: ABSZTRAKCIÓ PONTOSABBAN

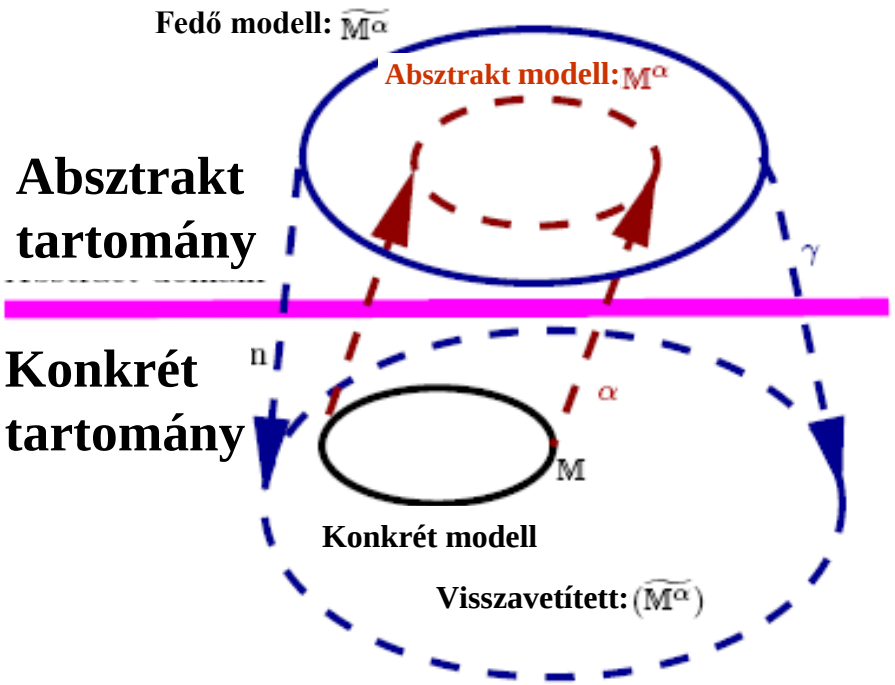
Absztrakció

- Az absztrakció **helyes**, ha az eredeti rendszer minden eleméhez hozzárendelhető az absztrakt rendszer egy eleme.
- Az absztrakció **teljes** *a vizsgált tulajdonságra nézve*, ha a rendszer összes jellemzője megjelenik, amely a vizsgált tulajdonságot befolyásolja.

(Több definíció lehetséges.)

Galois kapcsolat

- Galois kapcsolat (r)
  - $(A, \subseteq), (B, \subseteq)$ részben rendezett halmazok
  - $f: A \rightarrow B, g: B \rightarrow A$
  - $f(a) \subseteq b \leftrightarrow a \subseteq g(b) \leftrightarrow r(a, b)$
- Absztrakció
  - $\subseteq$ – tartalmazás
  - Befoglaló rész visszavetít
    - tartalmazza az eredetit



Galois kapcsolat - példa

- Jogosultsági szintek (több szint is elképzelhető)

Olvasás: Teljes jogosultság > Részleges jogosultság > Nincs jogosultság

Bizonyos attribútumok (pl. ID) elrejtése

Írás: Teljes jogosultság > … > Nincs jogosultság

- Mely jogosultságok kompatibilisek egymással?
  - Ha egy mezőt tudunk írni, olvasni is tudnunk kell!
  - X mezőt írnám → Részleges olvasási jog kell (de teljes is jó)
  - Részleges olvasási jogom van → Legfeljebb részleges írási jogom lehet (vagy semmi)
- Kompatibilitás: Galois reláció

Refactor példa

- Számok rendezése
  - Növekvő sorban: 18,23,131,247,1925
  - Csökkenő sorban: 1925,247,131,23,18
  - ABC rendben: 131,18,1925,23,247
- Kód (minimumkiválasztás):

```
void rendezes_novekvo(int* tomb, int meret){
    int i, j;
    for(i = 0; i<meret; i++){
        int minimumhely = i;
        for(j= i + 1; j<meret; j++){
            if(tomb[j] < tomb[minimumhely])
                minimumhely= j;
        }
        int tmp = *i;
        *i = *minimumhely;
        *minimumhely = tmp;
    }
 }
```

Miben különbözik a másik két sorrendhez tatozó kód?

Refactor példa

- Számok rendezése
  - Növekvő sorban: 18,23,131,247,1925
  - Csökkenő sorban: 1925,247,131,23,18
  - ABC rendben: 131,18,1925,23,247
- Kód (minimumkiválasztás):

```
void rendezes_novekvo(int* tomb, int meret){
    int i, j;
    for(i = 0; i<meret; i++){
        int minimumhely = i;
        for(j= i + 1; j<meret; j++){
            if(tomb[j] < tomb[minimumhely])
                minimumhely= j;
        }
        int tmp = *i;
        *i = *minimumhely;
        *minimumhely = tmp;
    }
 }
```

Miben különbözik a másik két sorrendhez tatozó kód?

Refactor példa

- Ötlet: Az összehasonlítást tegyük külön függvénybe

```
void rendezes(int* tomb, int size, bool (*hasonlit)(int,int)){
    int i, j;
    for(i = 0; i<meret; i++){
        int minimumhely = i;
        for(j= i + 1; j<meret; j++){
            if( (*hasonlit)(tomb[j],tomb[minimumhely]) )
                minimumhely= j;
        }
        int tmp = *i;
        *i = *minimumhely;
        *minimumhely = tmp;
    }
 }
```

Mit lehet még újrafelhasználni ebből a kódból?

Refactor példa

- Ötlet: Az összehasonlítást tegyük külön függvénybe

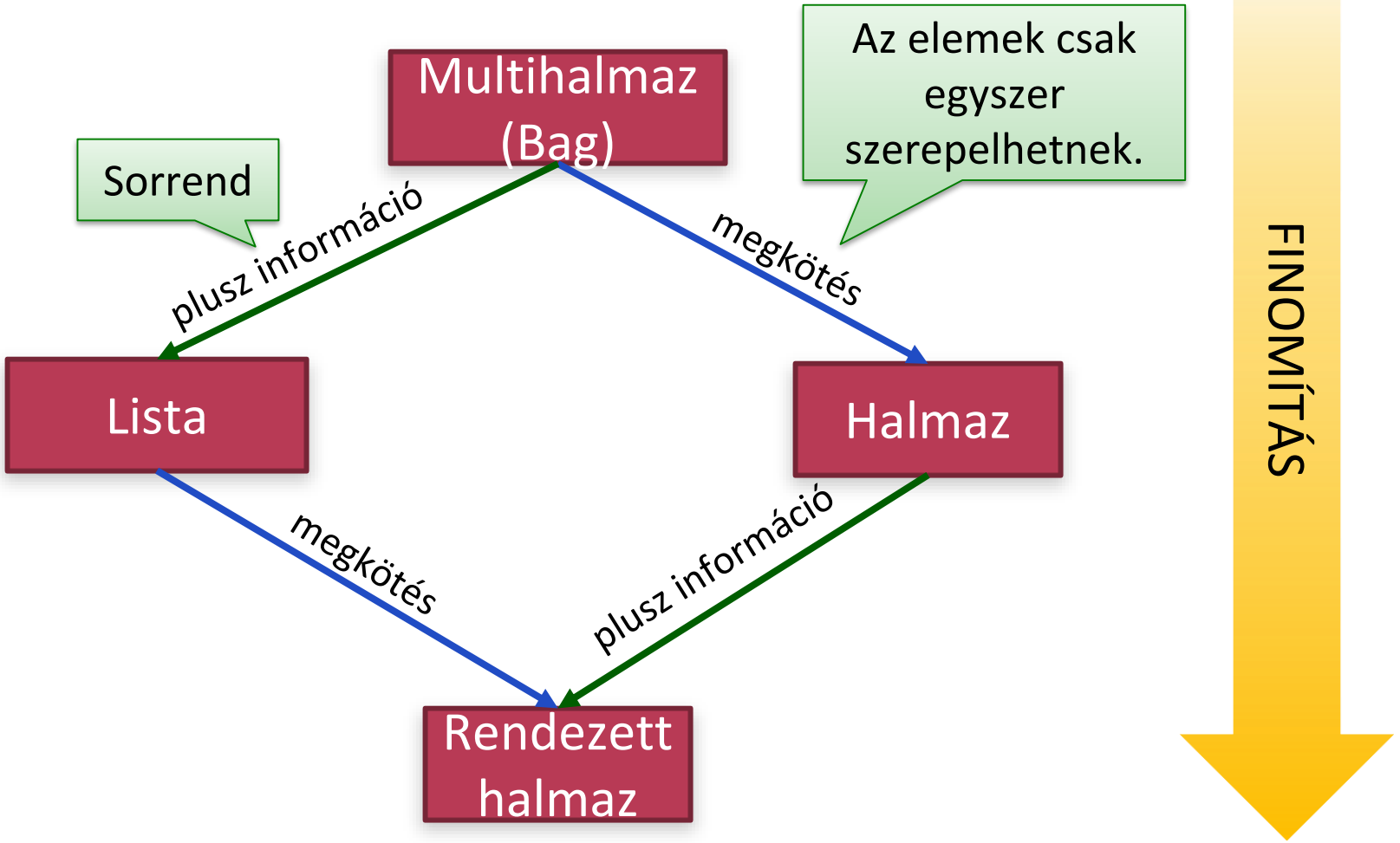
```
void rendezes(int* tomb, int size, bool (*hasonlit)(int,int)){
    int i, j;
    for(i = 0; i<meret; i++){
        int minimumhely = i;
        for(j= i + 1; j<meret; j++){
            if( (*hasonlit)(tomb[j],tomb[minimumhely]) )
                minimumhely= j;
        }
        int tmp = *i;
        *i = *minimumhely;
        *minimumhely = tmp;
    }
}
```

Mit lehet még újrafelhasználni ebből a kódból?

Absztrakció szerepe a kódolásban

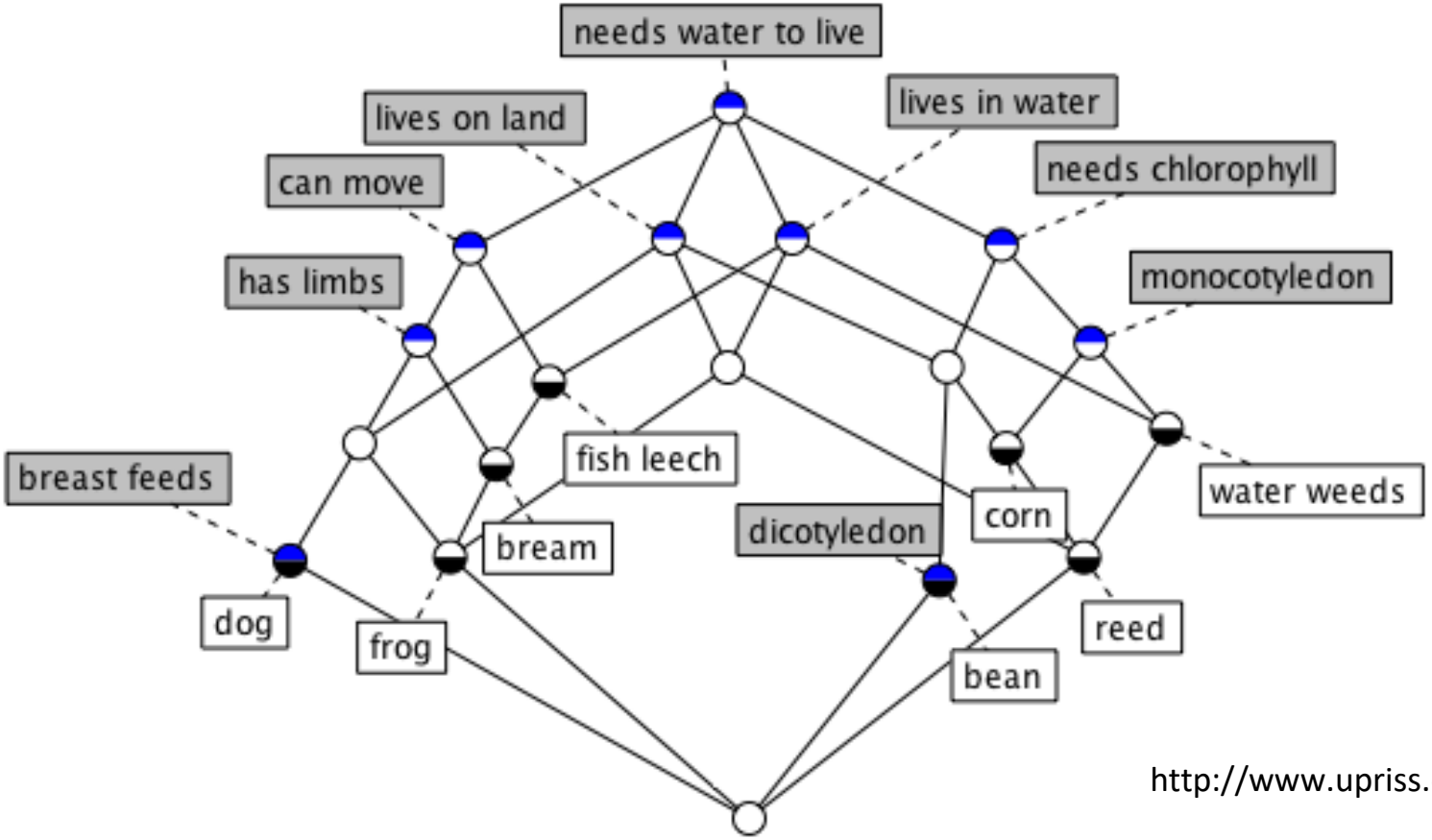
- Észrevétel:
  - Kiválasztásos rendezőalgoritmus: absztrakció
  - Különböző hasonlító függvények (→ sorrendek): konkretizálás
- Jó kód:
  - Algoritmusok paraméteresen
  - Különböző célok → külön példányosítás
  - Pl. ez az algoritmus használható szavak sorrendezésére

Absztrakciófinomítás



FCA-k, hálók

- Háló (lattice)
- FCA (Formal Concept Analysis)



http://www.upriss.org.uk/fca/examples