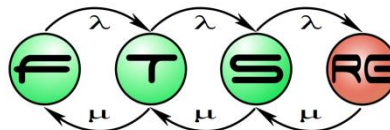


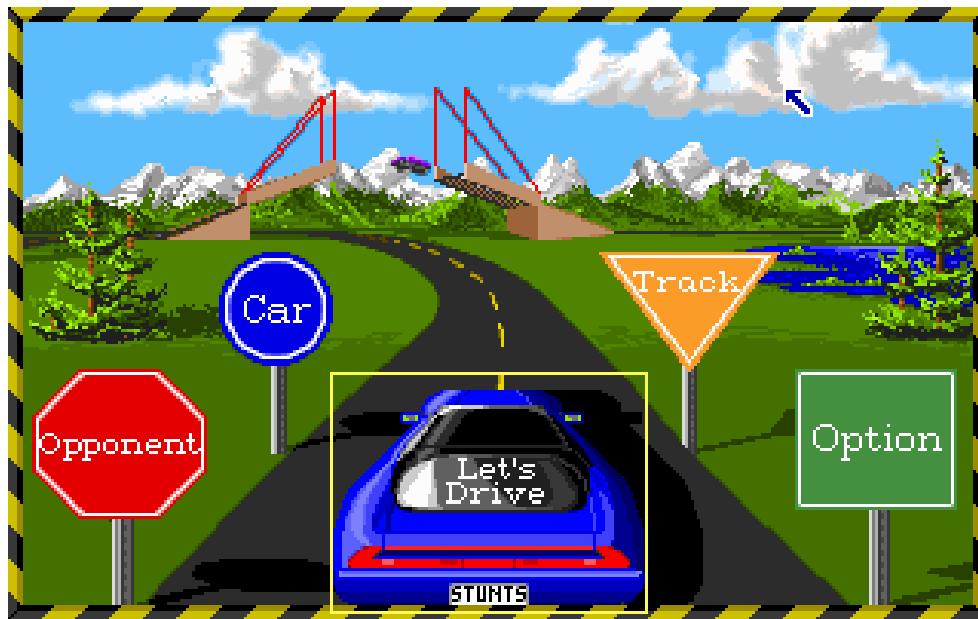
Modellezési alapismeretek

Rendszermodellezés

Budapesti Műszaki és Gazdaságtudományi Egyetem
Hibatűrő Rendszerek Kutatócsoport



“Motiváció”



- Stunts autóverseny, 1990.
 - Distinctive Games/Brøderbund Software
 - [https://en.wikipedia.org/wiki/Stunts_\(video_game\)](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?

“Valóságghű”-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

Modellezés alapfogalmai

Modell és modellezés



Mire használunk modelleket?

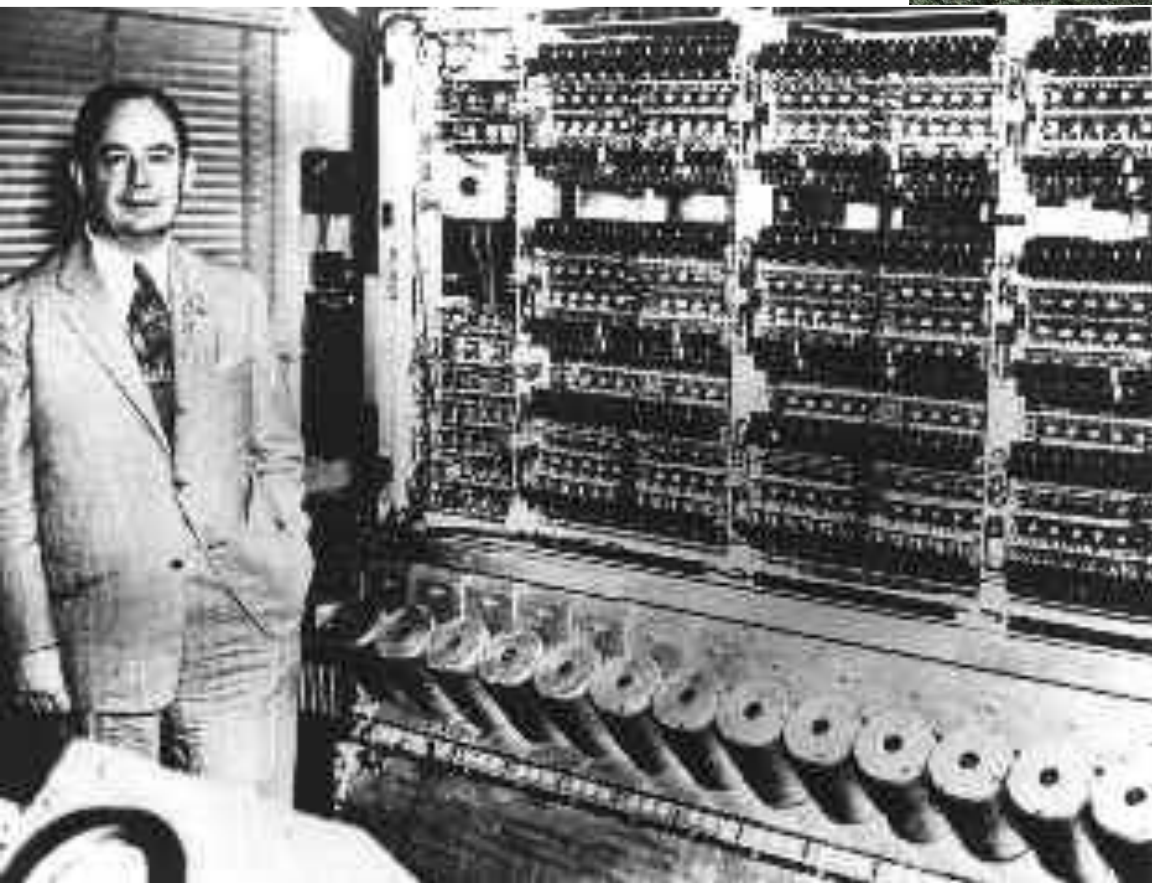


Alapfogalmak

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
MATEMATICIANS 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.

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



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

Matematikai modell vs. valóság

■ Minden modell:

zárt világ

- Hatások, faktorok
- Paraméterek
- Érvényesség

■ A modell

bizonytalan

ezen

■ Nem

előre

- **Ember**
- **Generál**

■ *Megoldás* *valóság*

■ Normál működés

- Perem

ridőre

**A modell egy(néhány)
kérdés megválaszolásának
segédeszköze !**

- **Elmélet**

- Anyaghiány

- Célfüggvény:

1. **Minél több** rendelés
határidőre
2. Költségminimum

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)

Kerék a levegőben

Kerék csúszik



- **A valóság: nyílt világ $\leftrightarrow$**
A modell: zárt világ
- **Valósághűség:**
 - Valószínű + kritikus esetek

Modell minősége

- A valóság: nyílt világ ←

A modell: zárt világ

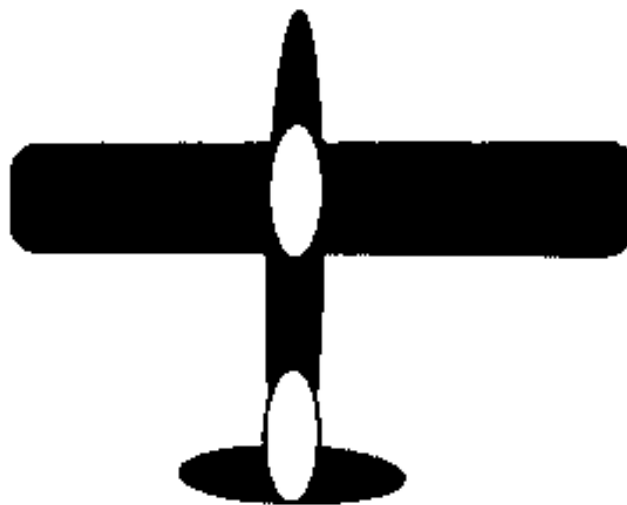
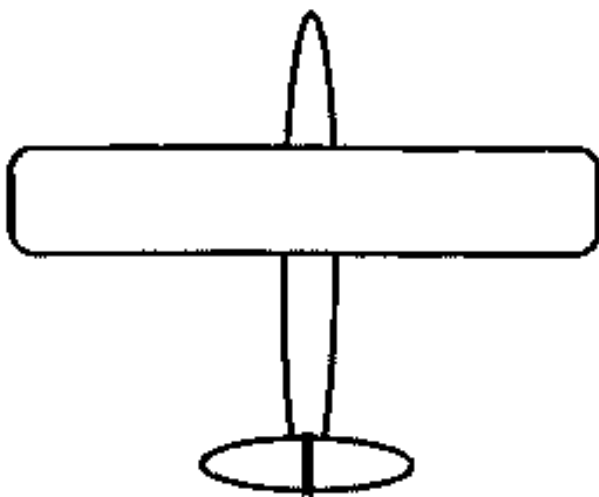
- Valóságban



ese

**Rossz modell tökéletes
implementációja halálos!**

Páncélozás? → Wald Abraham



Páncélozás? → Wald Abraham



**A modell a környezettel
együtt értelmes!**



Modell és modellezés



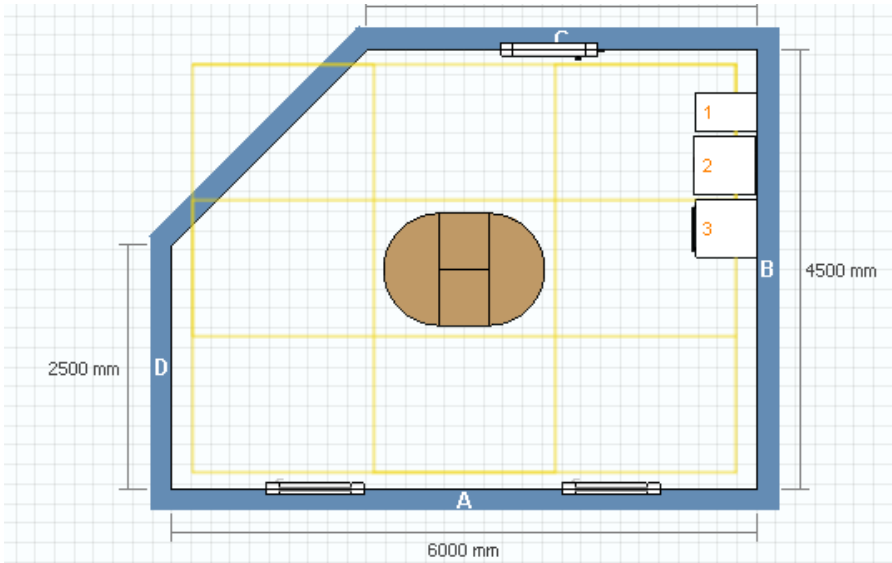
Mire használunk modelleket?



Alapfogalmak

Modellezés a gyakorlati életben?

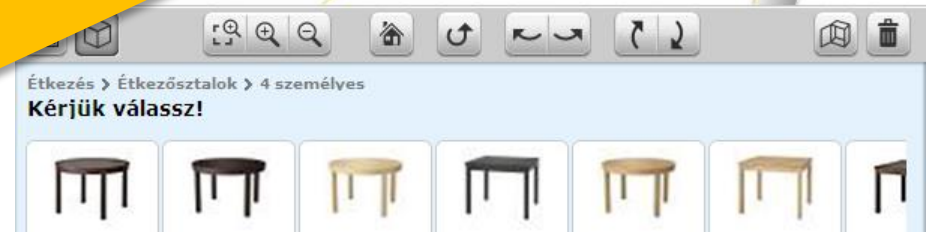
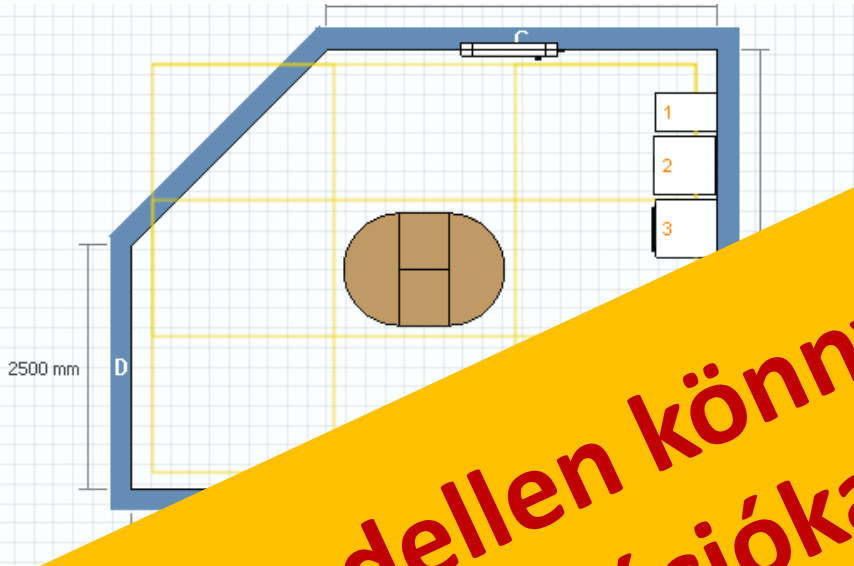
Pl.: [svéd cég] webes konyhatervezője



Modellezés a gyakorlati életben?

Pl.: [svéd cég] webes konyhatervezője

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

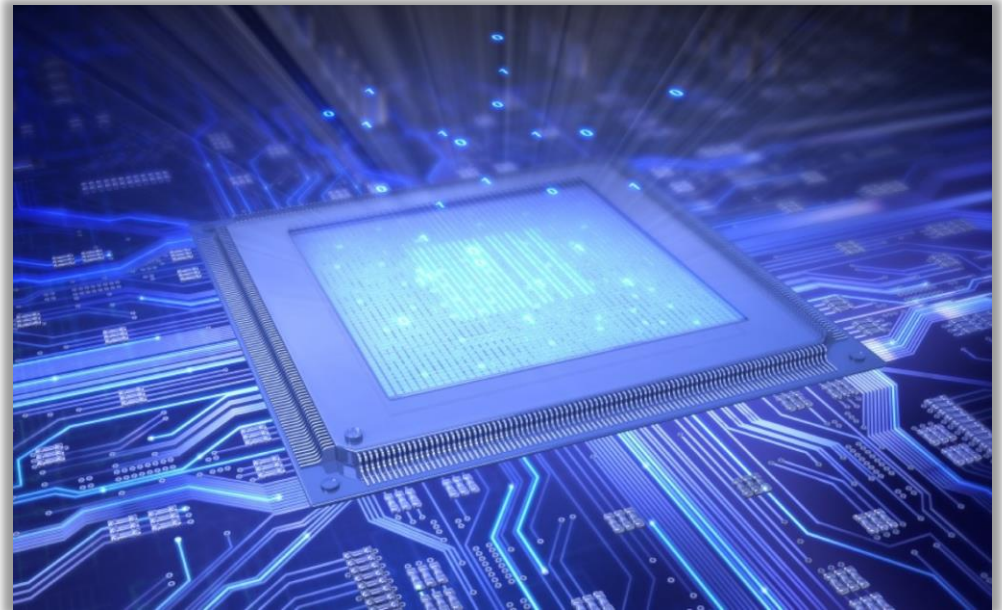


Ez is modellezési nyelv! 😊

- Verilog – szakterület specifikus, hardver leíró

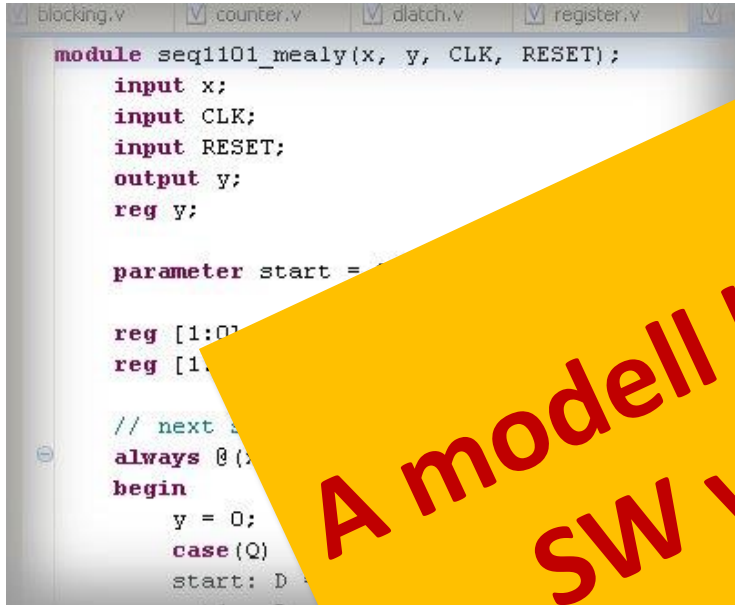
```
blocking.v counter.v dlatch.v register.v  
  
module seq1101_mealy(x, y, CLK, RESET);  
    input x;  
    input CLK;  
    input RESET;  
    output y;  
    reg y;  
  
    parameter start = 2'b00, got1 = 2'b01, got1  
  
    reg [1:0] Q;    // state variables  
    reg [1:0] D;    // next state 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



```
blocking.v counter.v latch.v register.v  
  
module seq1101_mealy(x, y, CLK, RESET);  
    input x;  
    input CLK;  
    input RESET;  
    output y;  
    reg y;  
  
    parameter start = 0;  
  
    reg [1:0] Q;  
    reg [1:0] D;  
  
    // next state logic  
    always @ (CLK) begin  
        y = 0;  
        case (Q)  
            start: D = 0;  
            00: D = 1;  
            01: D = 0;  
            10: D = 1;  
            11: D = 0;  
        endcase  
    end  
  
    y <= D;  
endmodule
```

**A modell lehet rendszer, HW,
SW vagy bármi más...**

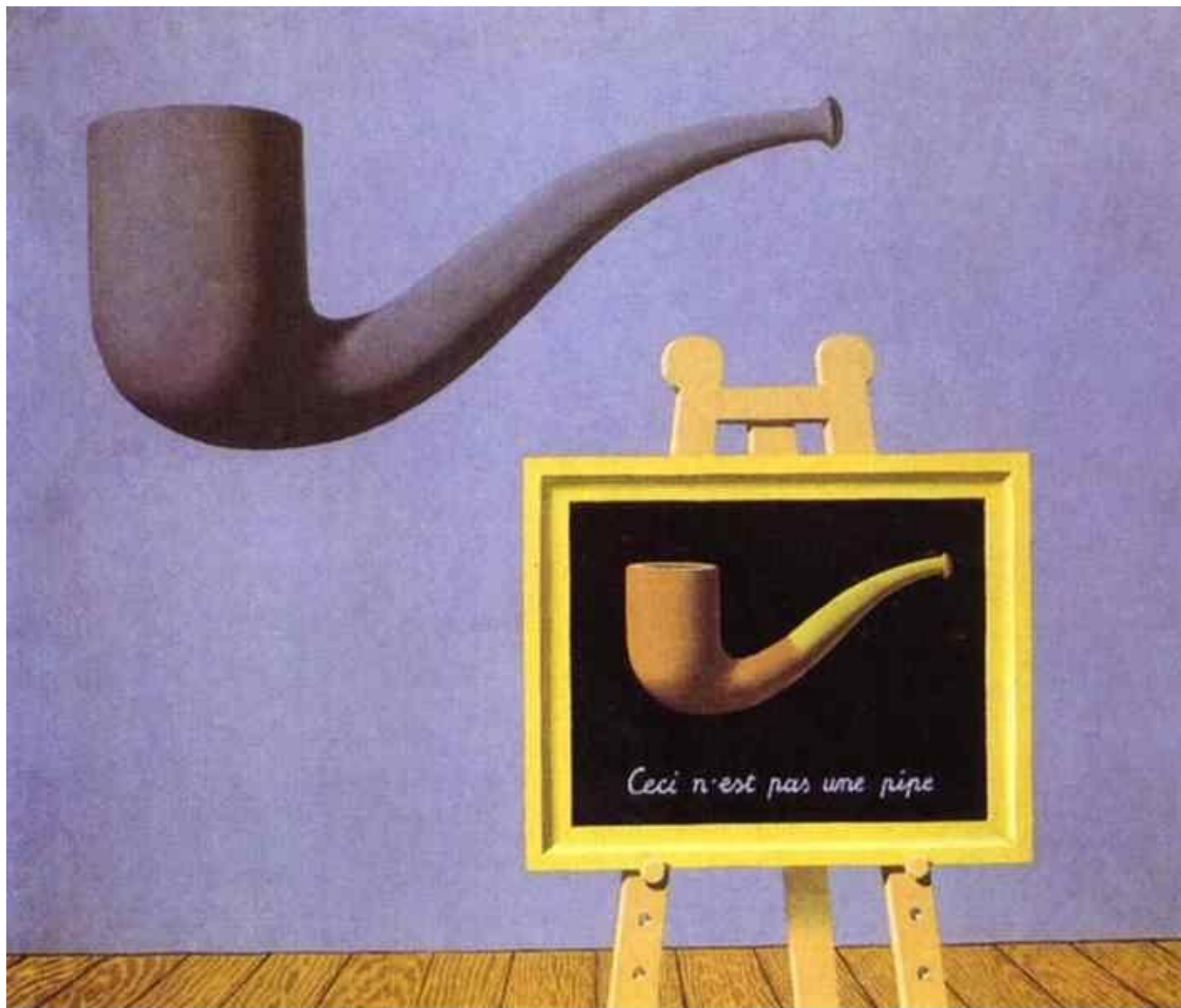
Mi értelme van modellezni?

- 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. SystemC”***
 - „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)

A modell modellje



abcgallery.com - Internet's biggest art collection

Alapfogalmak - Metamodellezés

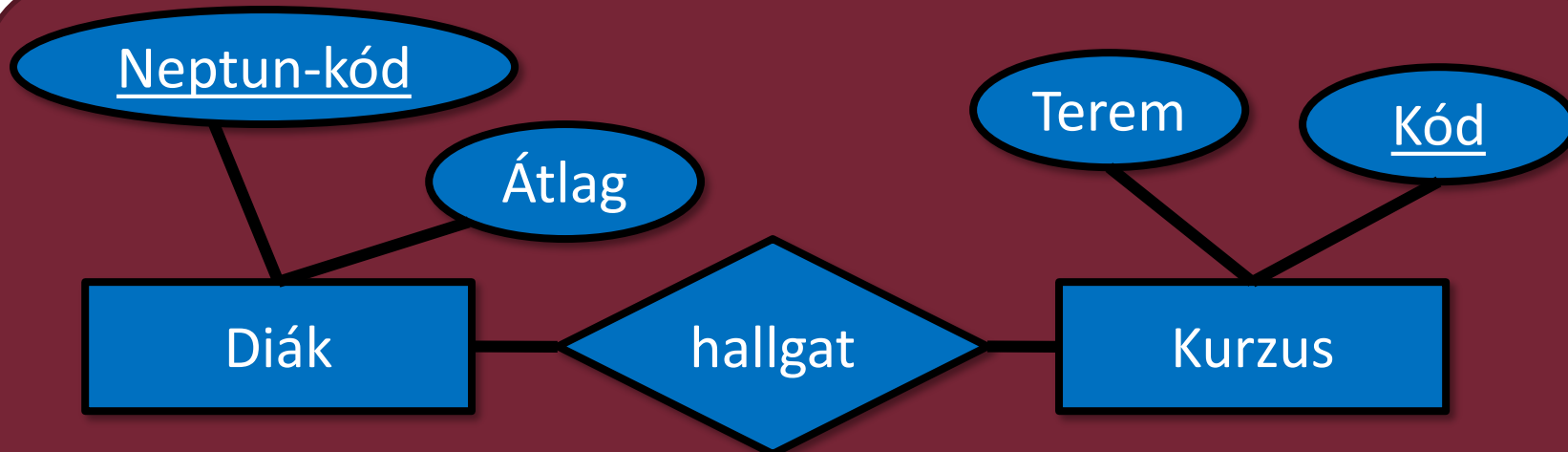
- Modellezési nyelv: milyen típusú elemei vannak?
 - ...és milyen kapcsolatban állhatnak ezek az elemek?
 - ...és ezeknek a típusoknak mik a viszonya egymáshoz?
- **Metamodell** = egy modellezési nyelv modellje
- Illusztrációk, amelyeket mindenki ismer
 - Adatbázis tábla
 - XML dokumentum
 - Relációs adatbázisséma
 - UML objektum diagram
 - ...

Alapfogalmak - Metamodellezés

■

■

■



Illusztrációk, amelyekkel minden

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

Rendszertervezési folyamat

Menedzsment

- Beszerzés, ellátás
- Tervezés, vezetés, értékelés

ANSI/EIA 632
szabványból

Rendszer
tervezés

- Követelmények definiálása
- Megoldás definiálása

Mérnöki feladatok

Termék
Elkészítése

- Implementáció
- Használhatóság

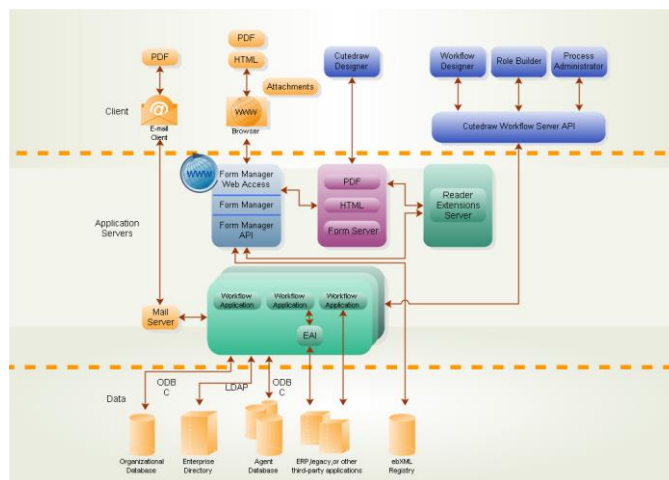
Kiértékelés

- Rendszer analízis
- Követelmény validáció
- Rendszer verifikáció
- Végtermék validáció

Ezeket *tipikus*
modellekkel
támogatni

Felhasználás – Dokumentáció

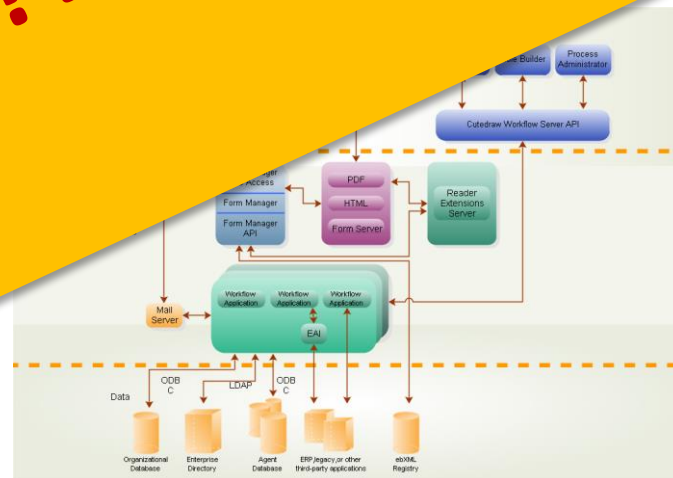
- A modell egyszerűbb
 - könnyebben elmondható, mint a teljes valóság
 - fokozatosan finomítható (ld. később)
- Kommunikáció, szemléltetés
 - demonstráció (ld. később)
 - érthető szöveges nyelv
 - szemléletes diagram
- Gondolkodás, tervezés támogatása
 - hasonlóak a szempontok
 - „kommunikáció magunkkal”



Felhasználás – Dokumentáció

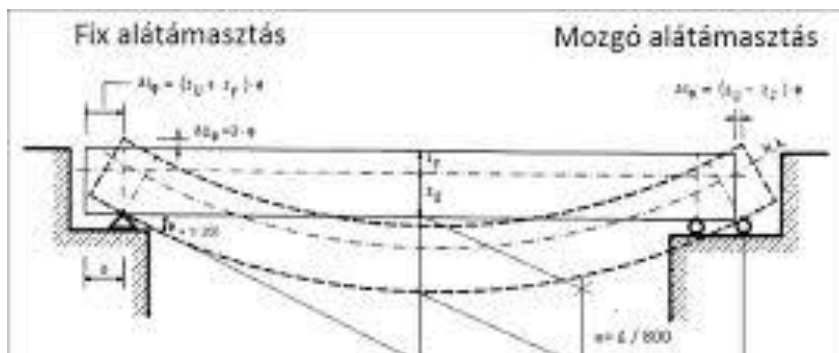
- A modell egyszerűbb
 - könnyebben elmondható, mint a teljes valóság
 - fokozatosan finomítható (ld később)
- Kommunikáció
 - dem...
- Gondolkodási folyamatok zés támogatás

Egy rajz = ??? oldal szöveg?



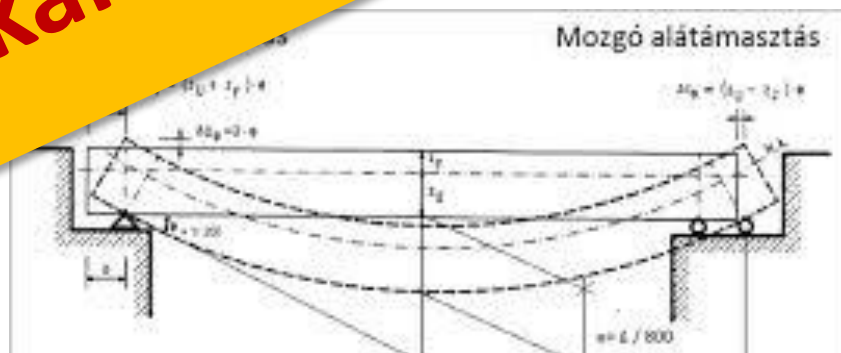
Felhasználás - Analízis

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



Felhasználás - Analízis

- Emberi erővel vagy (részben) automatizáltan
 - Cél
 - Előrejelzés
 - Módszer
 - Felületes, statikus elemzés
 - pl. hiányzó pontos
 - Dinamikus elemzés
 - bejárás
- Modell és matematikai helyességbizonyítás
- számítása, vezérlése (pl. ütemezés)



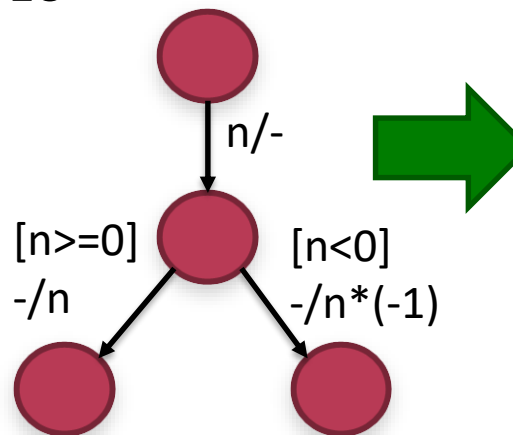
Felhasználás: szimuláció

- 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

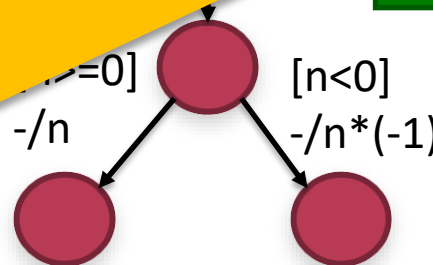


```
int n;  
scanf("%d", &n);  
if (n >= 0) {  
    printf("%d", n);  
} else {  
    n = n * (-1);  
    printf("%d", n);  
}
```

Felhasználás - Származtatás

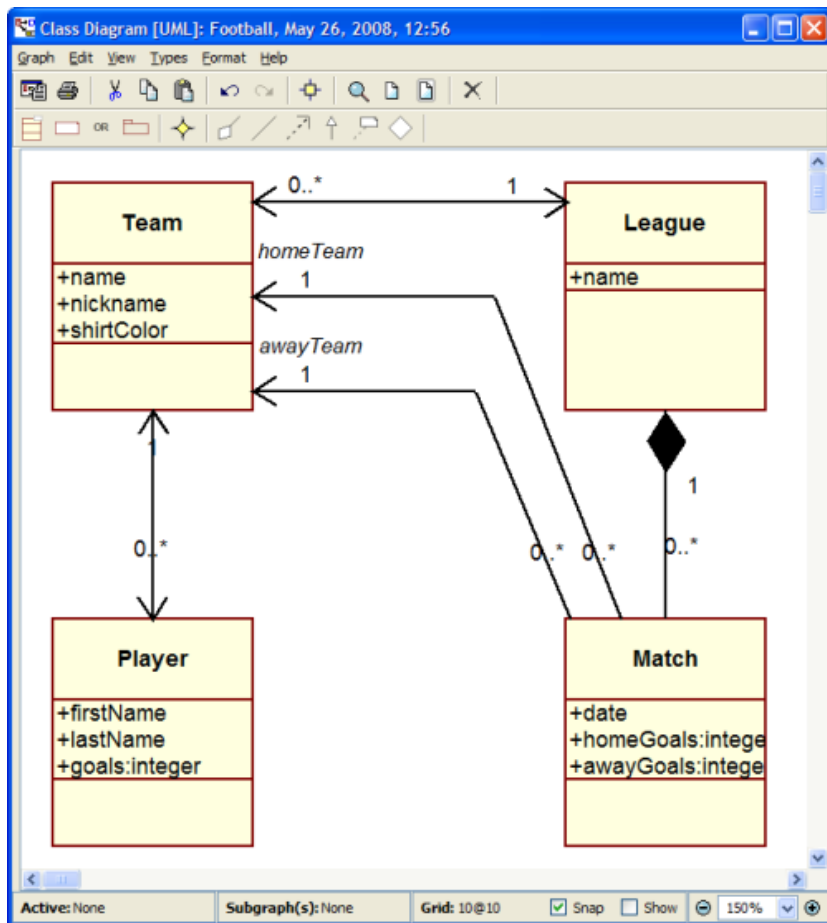
- Emberi erővel vagy (részben) automatizáltan
- Eredmény
 - programkód, analízálható nyelv
 - másik modell
 - finomítás, követelmény
 - részaspék
 - m

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



League [Gears] - Mozilla Firefox

File Edit View History Bookmarks Tools Help

League [Gears]

Enter a League to store in the database:

name

Teams name

Matches date

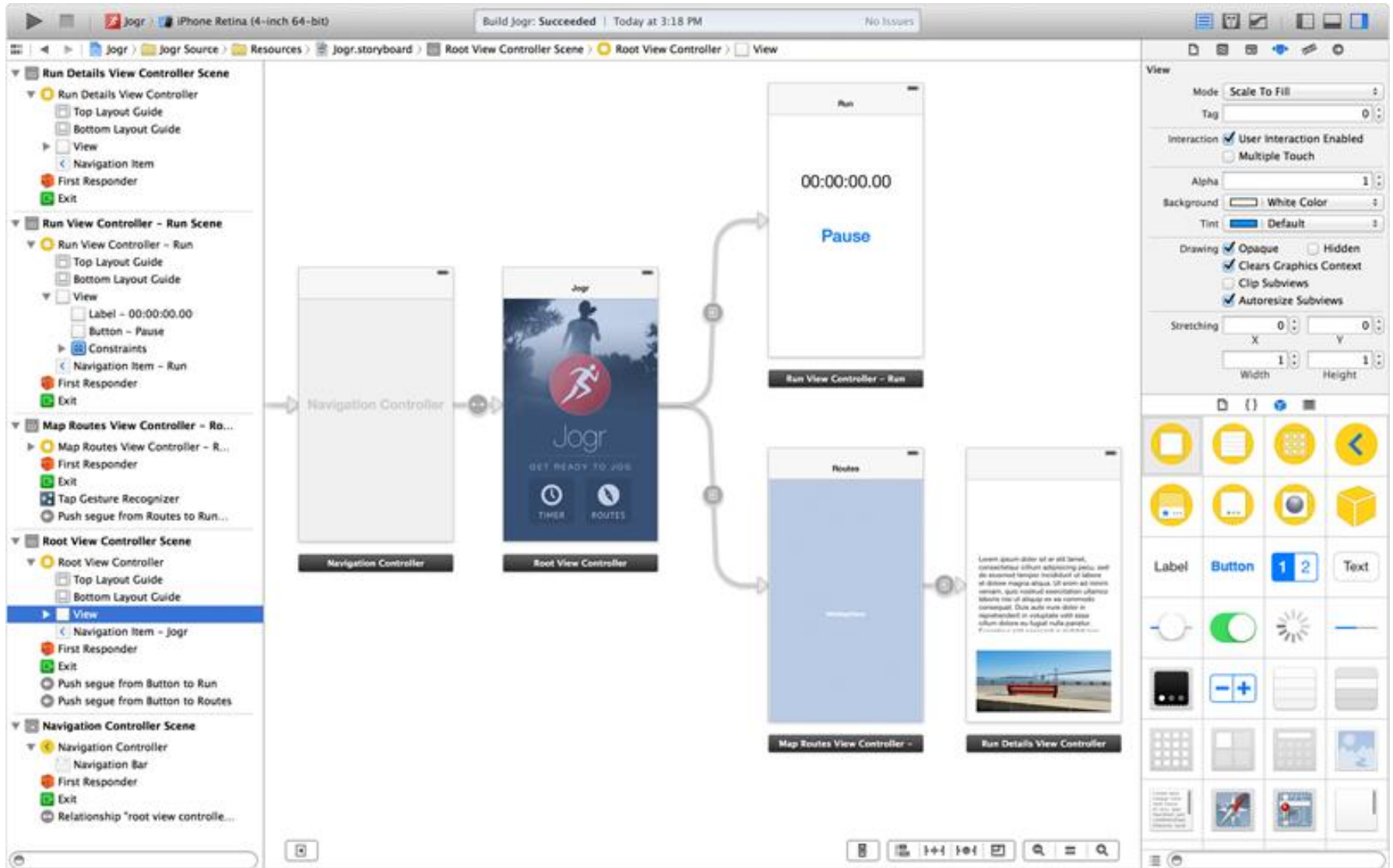
4 most recently edited League entries:

name	
Edit	Premiership Delete

[Back to top page of Football application.](#)

This page uses Gears to record your entries on the local disk. If you navigate away and revisit this page, all your data will still be here. Try it!

Okostelefon alkalmazás fejlesztése



Reverse engineering

- Modell visszafejtése a rendszer alapján

League [Gears] - Mozilla Firefox

File Edit View History Bookmarks Tools Help

League [Gears]

Enter a League to store in the database:

name

OK

Teams name

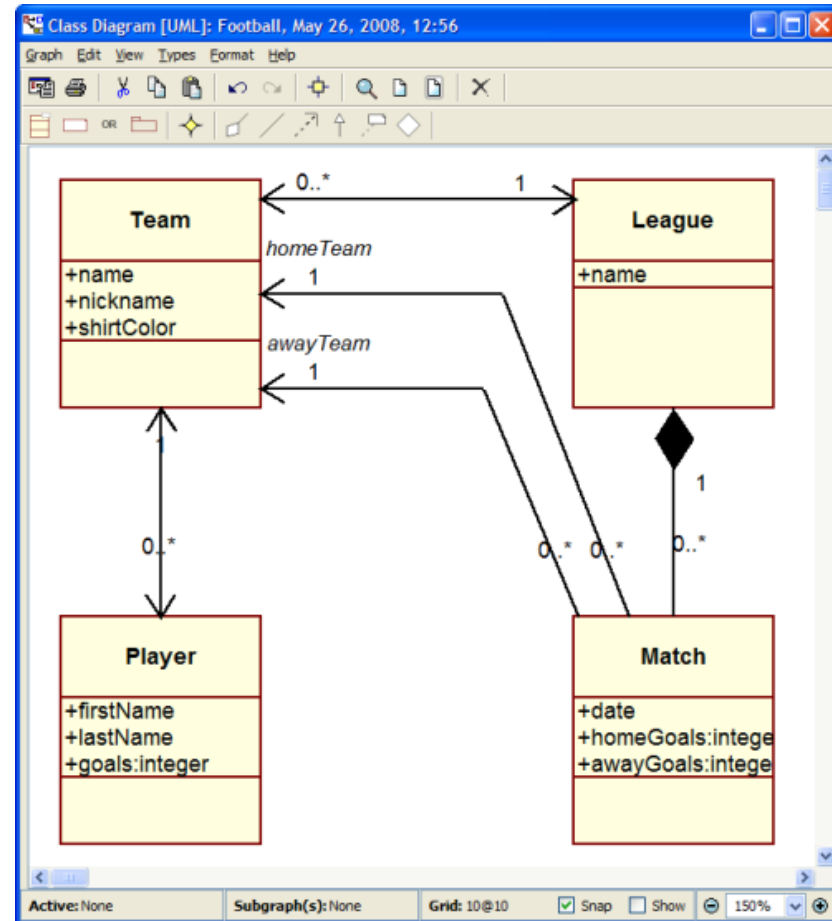
Matches date

4 most recently edited League entries:

name	
Edit	Premiership Delete
New	View All Delete All

[Back to top page of Football application.](#)

This page uses Gears to record your entries on the local disk. If you navigate away and revisit this page, all your data will still be here. Try it!



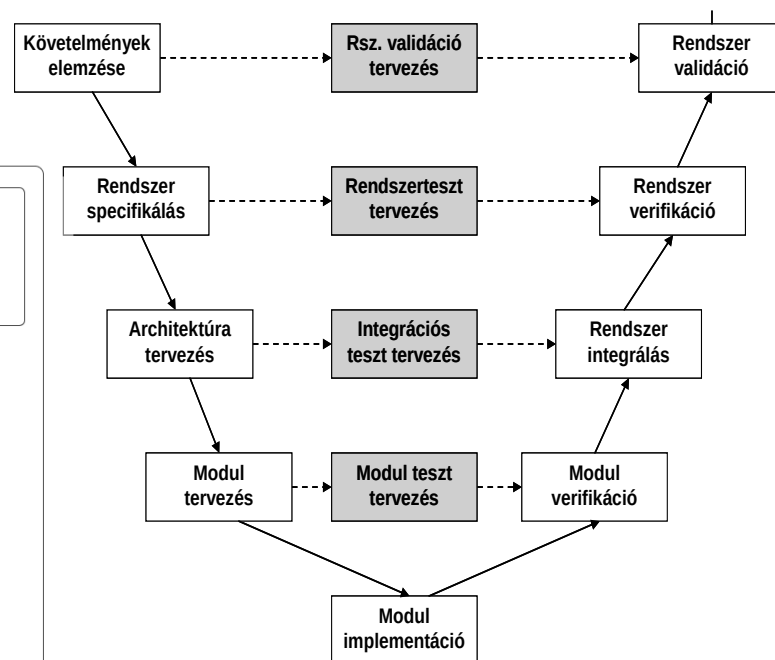
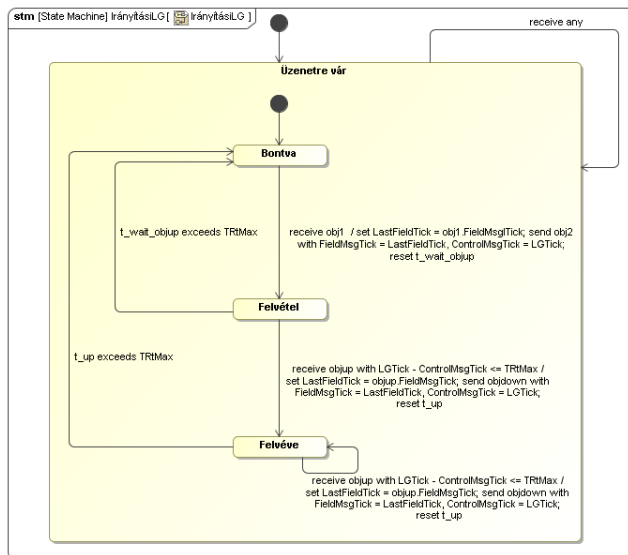
Vasúti szoftverek fejlesztése

Megoldandó probléma:

SIL-4 vasúti alkalmazás fejlesztési lépéseinek és ezek eredményeinek ellenőrzése

Kihívások:

- Szisztematikus felülvizsgálat
- Modellelés és formális verifikáció



Folyamat → modell → minőségbiztosítás

Vasúti szoftverek fejlesztése

Megoldandó probléma:

SIL-4 vasúti alkalmazás fejlesztési lépések és ezek eredményeinek ellenőrzése

Kihívások:

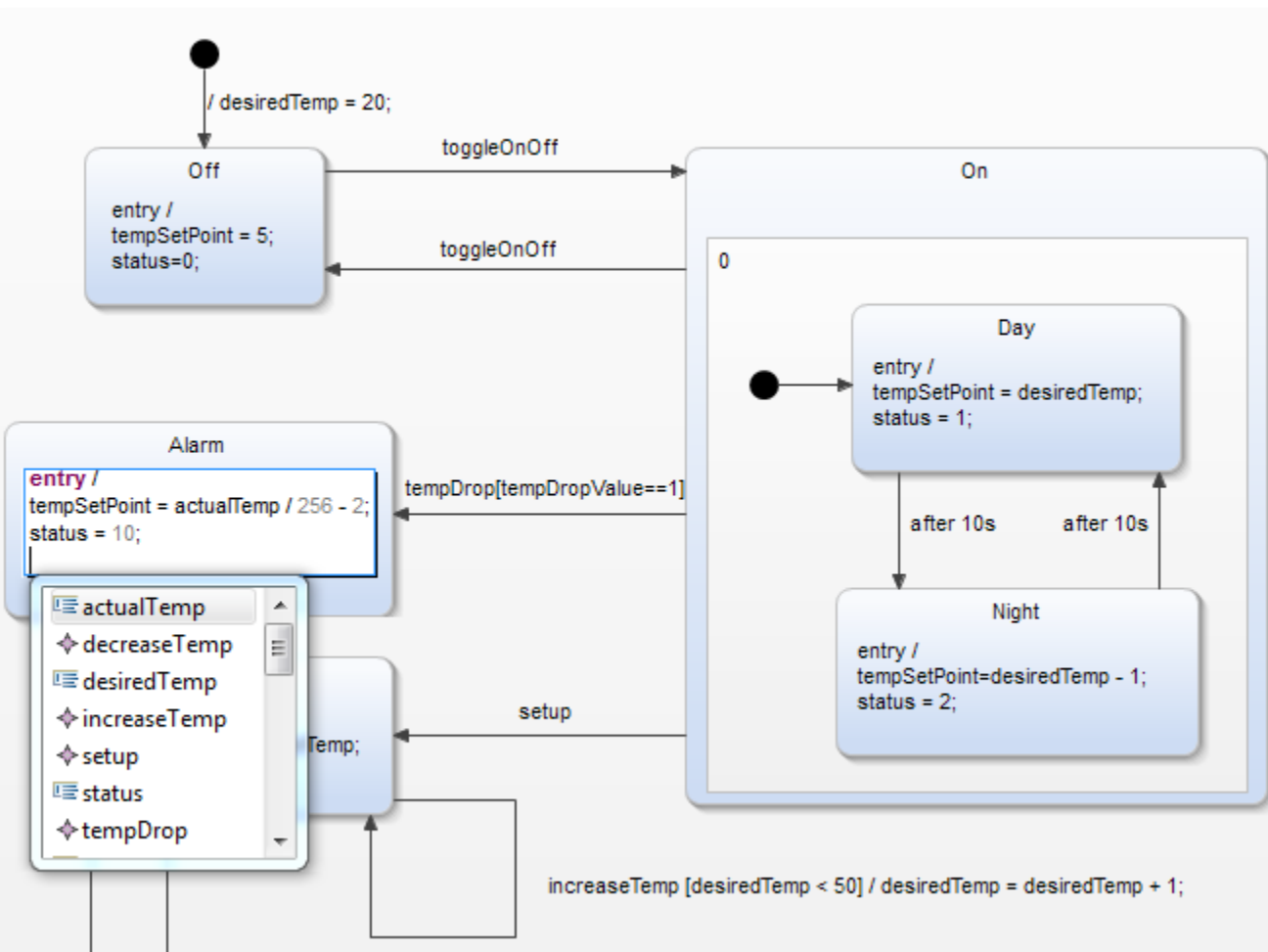
- Szisztematikus felülvizsgálat
- Modell formális verifikáció

A kritikus alkalmazások fejlesztése modellvezérelt



Folyamat → modell → minőségbiztosítás

Yakindu:állapotdiagram



```

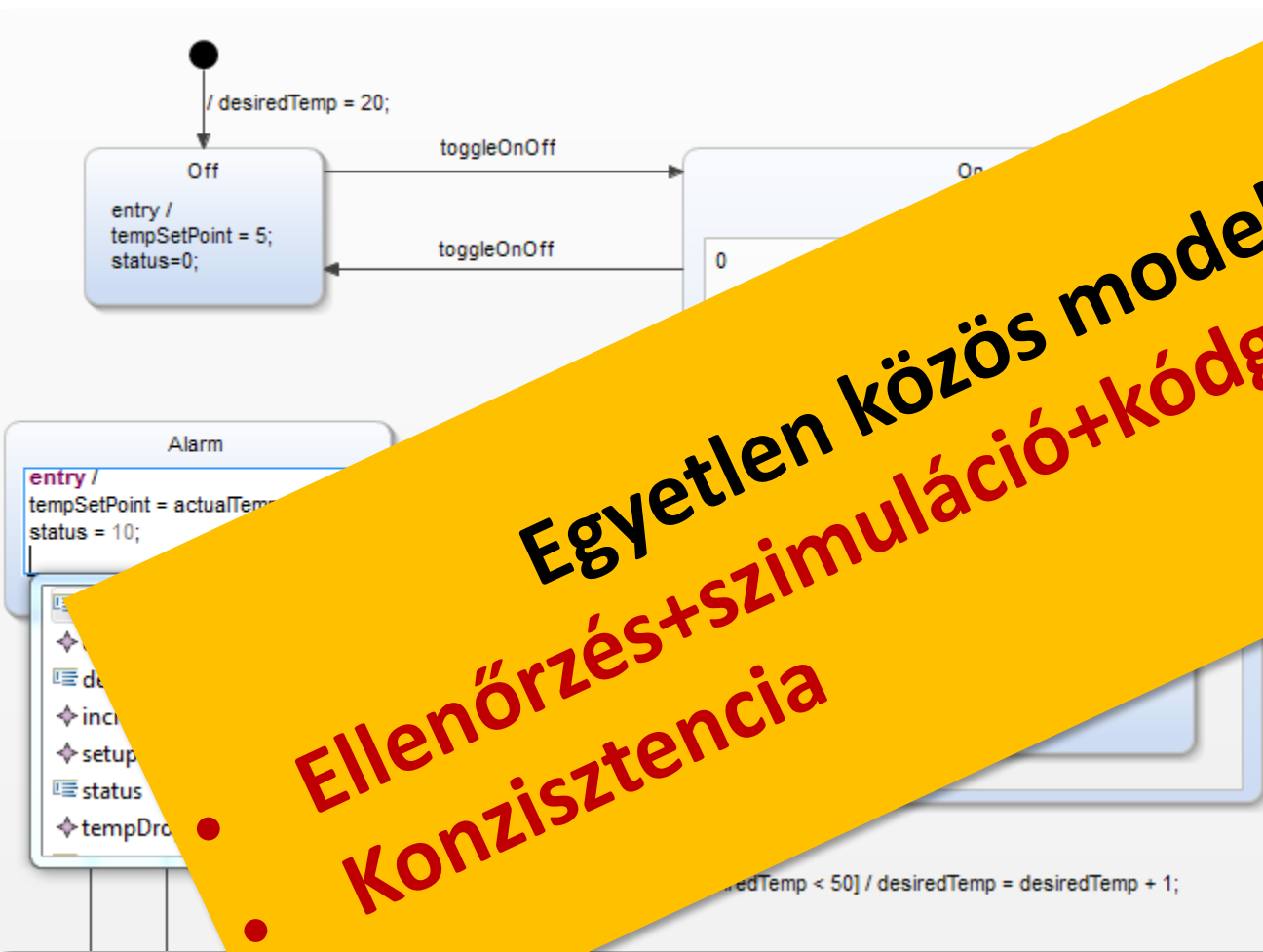
import java.io.*;
import java.net.*;
import java.security.*;
import protection;

public class Client {
    public void sendAuthentication(String username, String password) throws IOException {
        OutputStream outStream = new DataOutputStream(new DataOutputStream(new FileOutputStream("out.txt")));
        long t1 = Math.random();
        double q1 = Math.random();
        byte[] protected1 = ProtectionUtil.protect(q1, t1);
        long t2 = Math.random();
        double q2 = Math.random();
        byte[] protected2 = ProtectionUtil.protect(q2, t2);
        out.writeUTF(username);
        out.writeInt(protected1.length);
        out.write(protected2);
        out.flush();
    }
}

public static void main(String[] args) {
    String host = args[0];
    int port = 7999;
    String user = "John";
    String password = "shh";
    Socket s = new Socket(host, port);
    Client client = new Client();
    client.sendAuthentication(user, password);
}

```

Yakindu:állapotdiagram

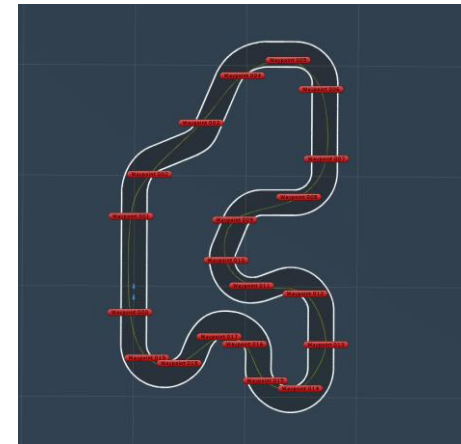
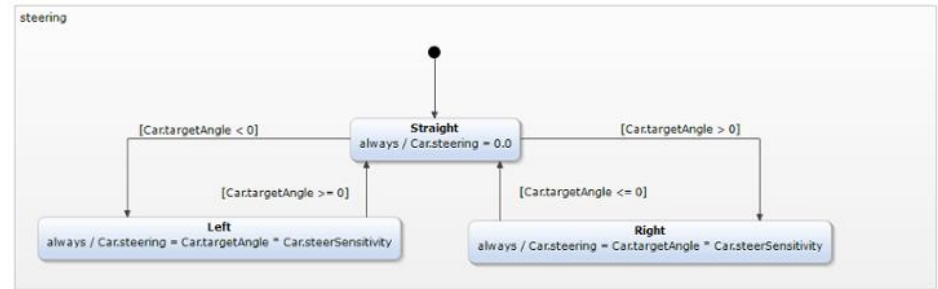
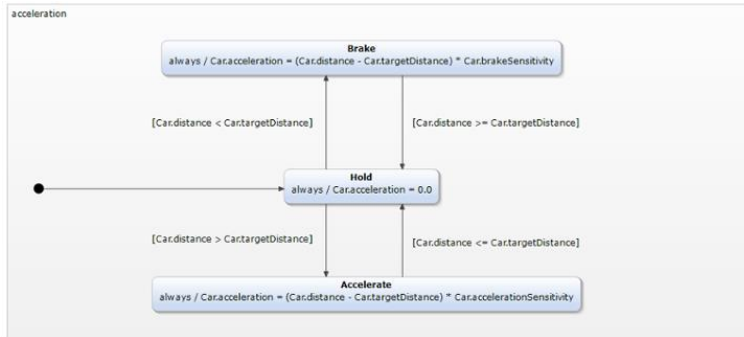


Egyetlen közös modell
Ellenőrzés+szimuláció+kódgenerálás

```
void main(st  
    int argc = args[0];  
    int port = 7999;  
    String user = "John";  
    String password = "sh  
    Socket s = new Socket  
  
    Client client = new  
    client.sendAuthent
```


Yakindu → 3D szimuláció

Járműkövető/távolságtartó rendszer interaktív 3D szimulációja



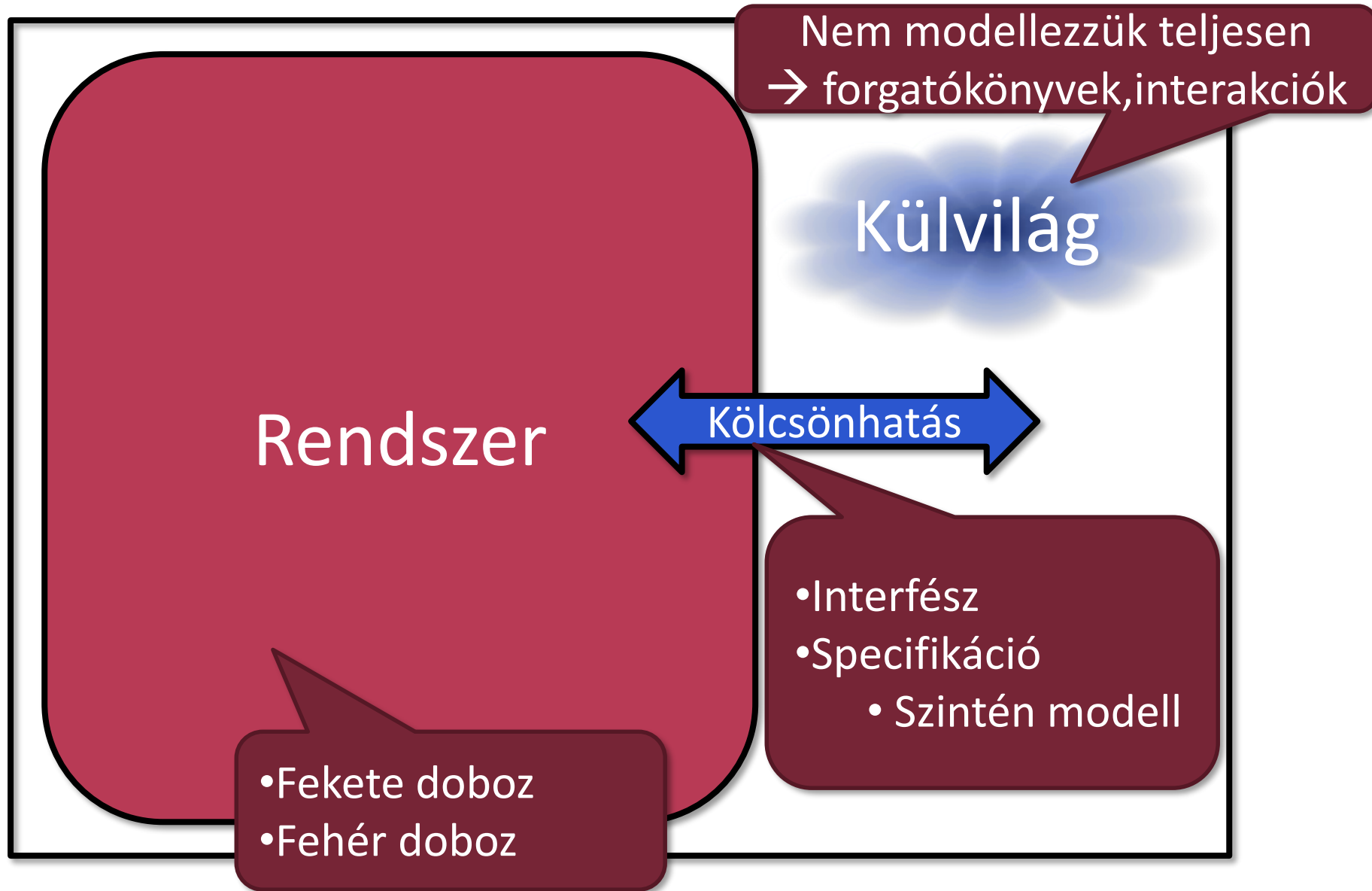
Interactive 3D Visualization and Simulation with State Machines, Benjamin Bolte, 2017.
<https://blogs.itemis.com/en/interactive-3d-visualization-and-simulation-with-state-machines>

Modell és modellezés

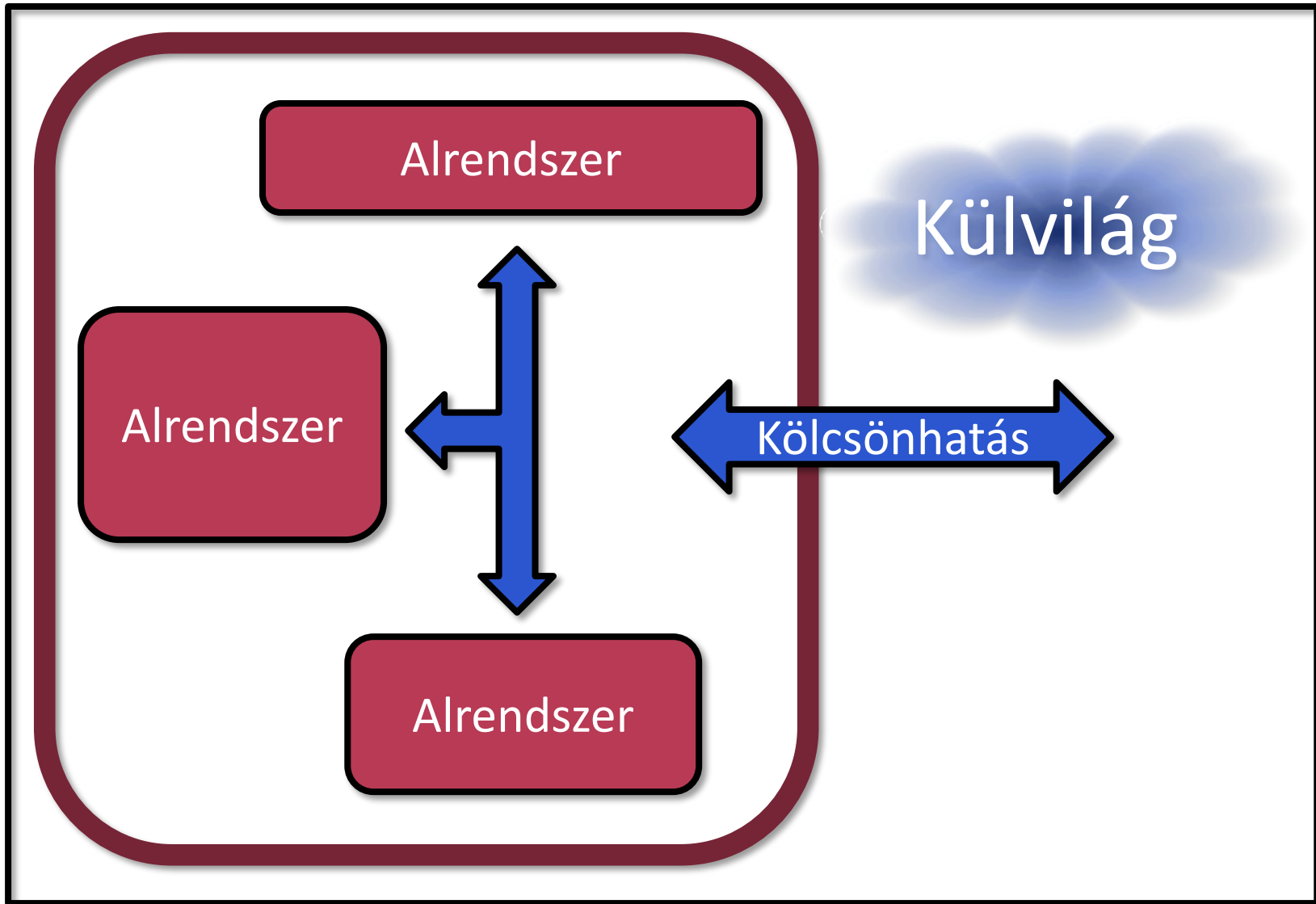
Mire használunk modelleket?

Alapfogalmak

Alapfogalmak – rendszer és külvilág



Alapfogalmak – rendszer és külvilág



Alapfogalmak – Finomítás/Absztrakció

- *Finomítás*: a modell gazdagítása részletekkel...
Rendszer felbontása alrendszerekre
- ...hogyan az eredeti modell absztrakció maradjon
Nincsenek ellentmondó részletek
- Inverze: (*vertikális*) *absztrakció*
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

Alapfogalmak – finomítás

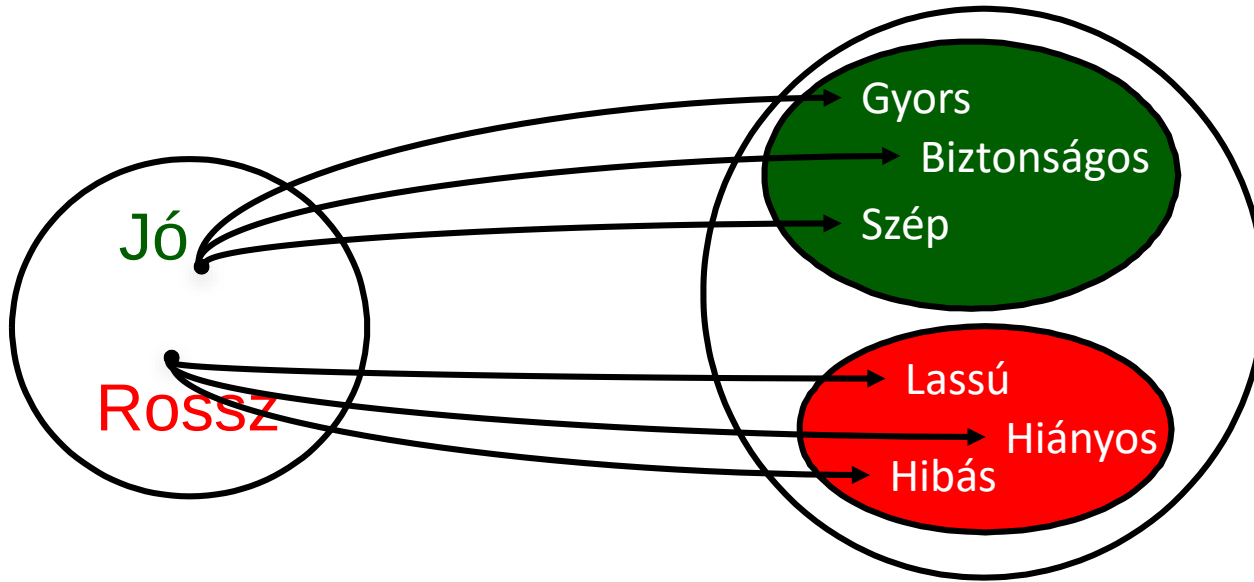


Alapfogalmak- finomítás



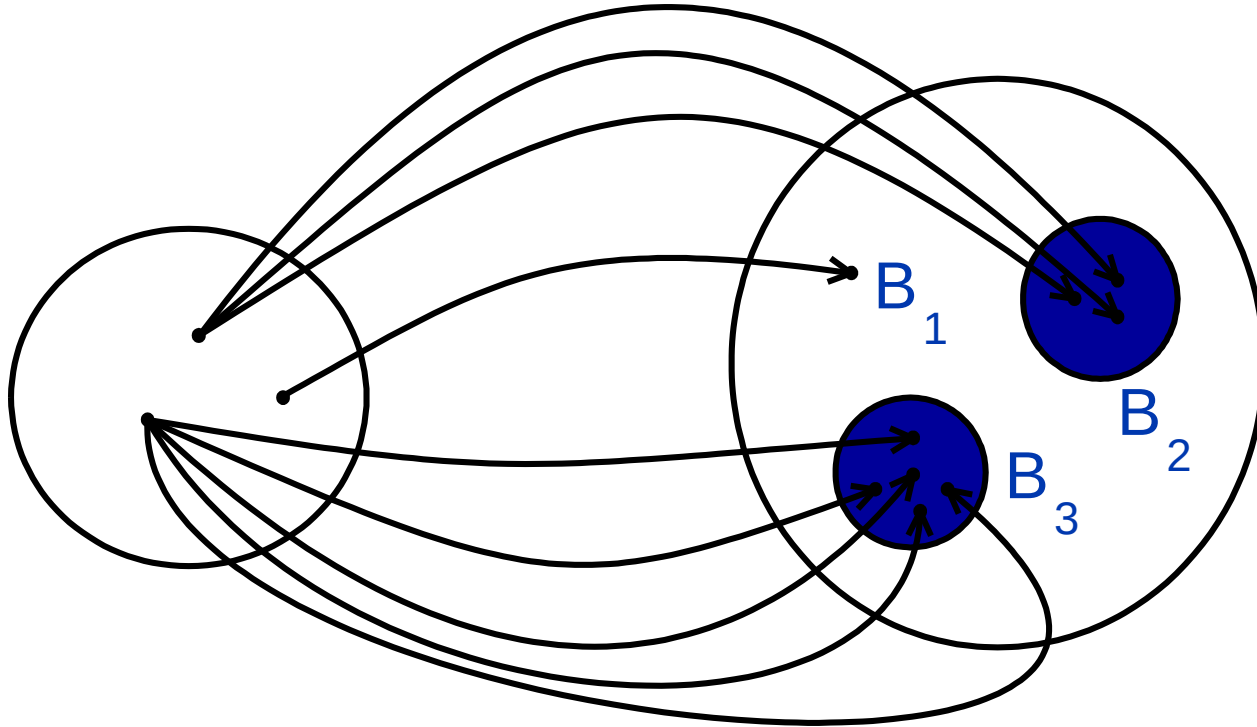
Halmazfinomítás

Diszjunkt részhalmazok hozzárendelése elemekhez



Halmazfinomítás

Diszjunkt részhalmazok hozzárendelése elemekhez



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

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.)

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?

→ Kódduplikáció elkerülése

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?

→ Kódduplikáció elkerülése

Absztrakció szerepe a kódolásban

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

Modellezés a tárgyban

