

Kihívások és trendek mesterséges intelligencia alapú rendszerek tesztelésében

Majzik István

BME VIK Méréstechnika és Információs Rendszerek Tanszék

IIR Szoftvertesztelés 2019 konferencia

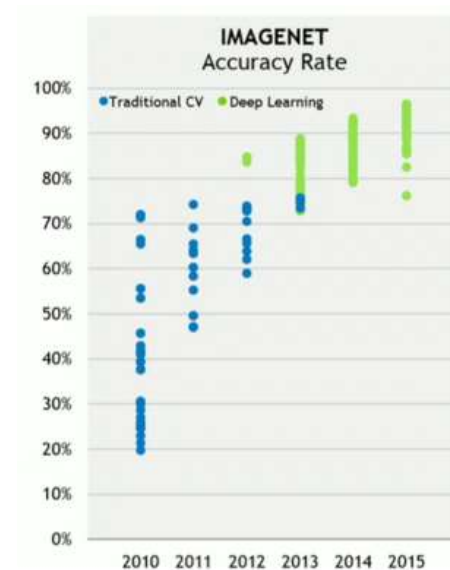
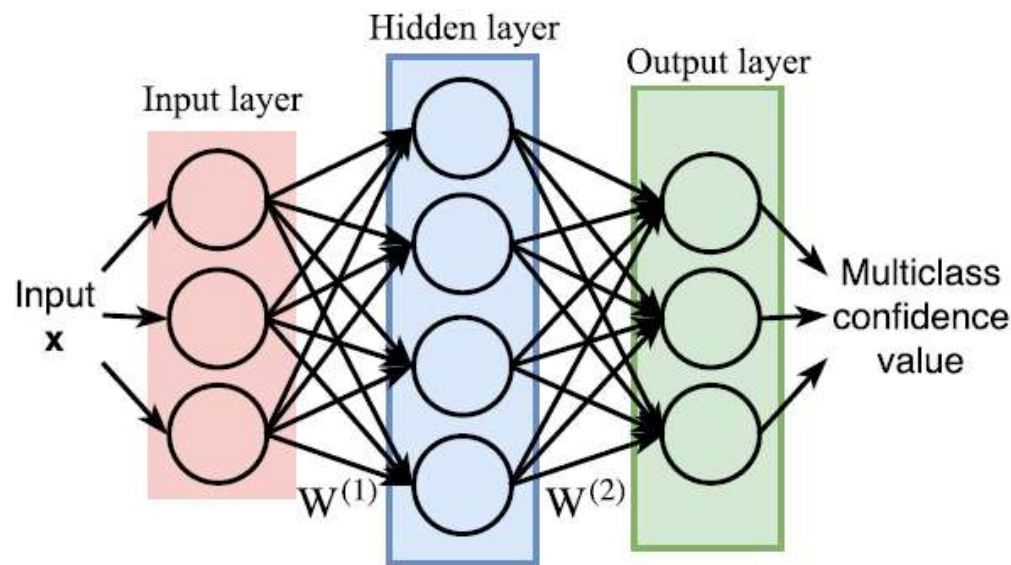
Budapest, 2019. május 29.

Mesterséges intelligencia alapú rendszerek

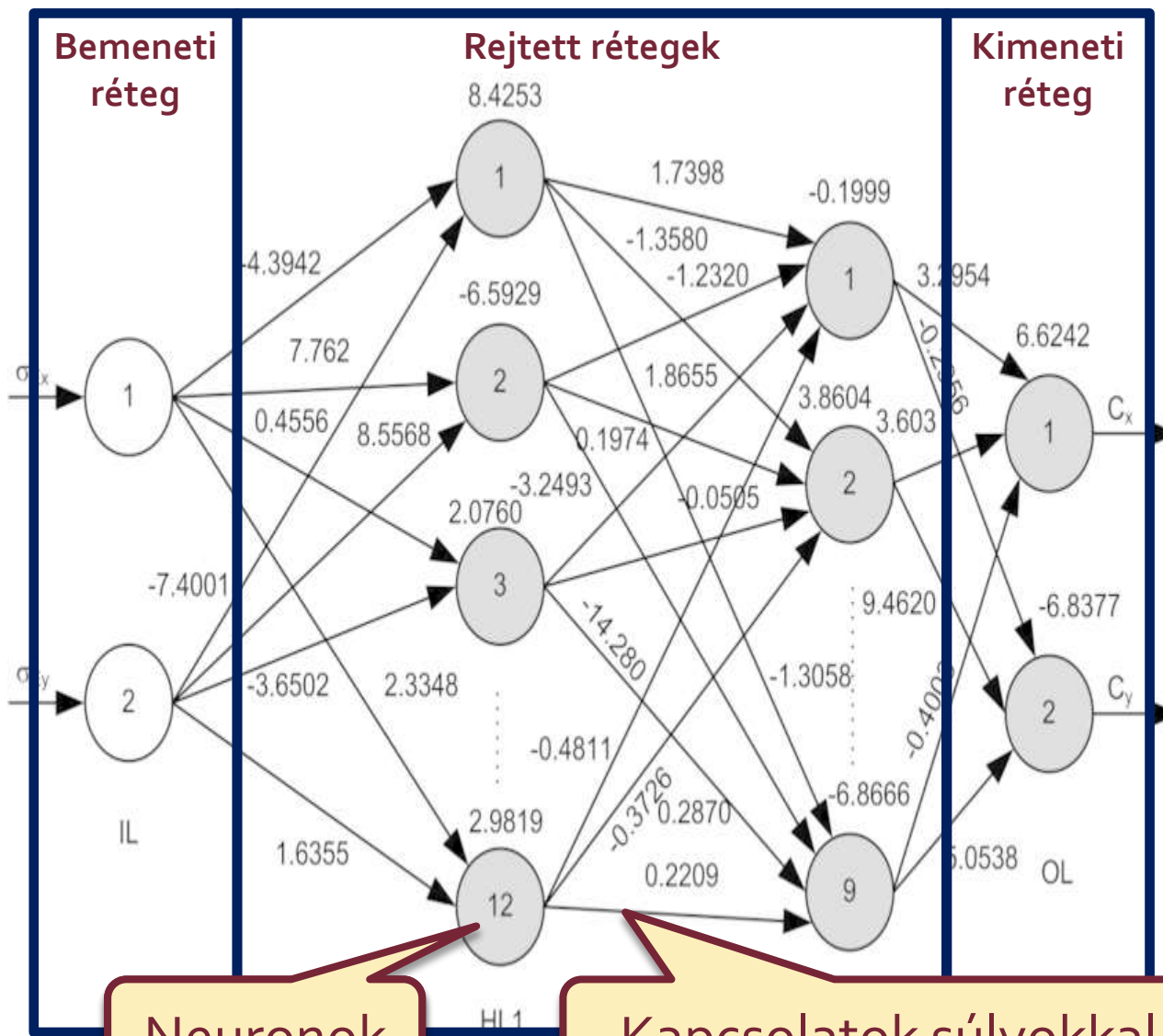
- **Mesterséges intelligencia (MI) elemek**
 - Tudásreprezentáció, keresés, következtetés, döntéshozatal, tanulás, tervekészítés, ...
- Jellegzetes **feladatok**
 - Felismerési feladatok: Objektumok, személyek, környezeti szituációk, adatsorozatok
 - Döntési feladatok: Akciók adott kontextusban
 - Előrejelzés: Megfigyelt viselkedés alapján
- Példa technológiák
 - Mély neurális hálók
 - Bayes-hálók
- Biztonságkritikus alkalmazásokban is
 - Autonóm járművek, robotok



Objektum felismerés neurális hálóval



Mély neurális háló (deep neural network)



Strucc

Neuronok

Kapcsolatok súlyokkal

Neurális hálók jellegzetességei

- Tanulás
 - Tanító halmaz + teszt halmaz
- Jellegzetes hiba: Rosszul felismert bemenet
 - „Ellenséges” bemenet (adversarial input)



Panda

(57.7% konfidencia)

+



=

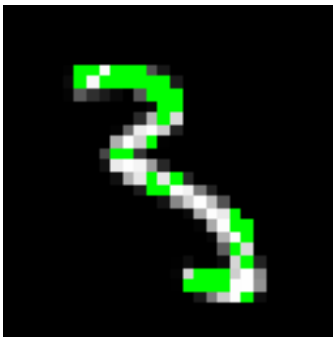


Gibbon

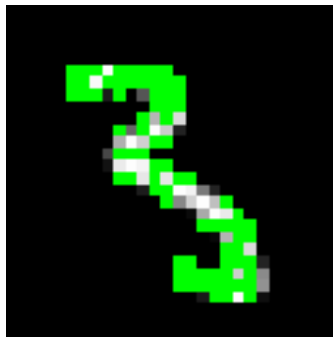
(99.3% konfidencia)

Neurális hálók jellegzetességei

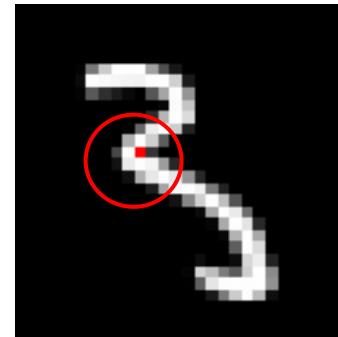
- Tanulás
 - Tanító halmaz + teszt halmaz
- Jellegzetes hiba: Rosszul felismert bemenet
 - „Ellenséges” bemenet (adversarial input)
- Debuggolás
 - Neuron aktivitás: milyen súllyal szerepel a kimenetben
 - Interpretálhatóság, megmagyarázhatóság



Felső 5%
fontos pixel
(3-hoz)



Felső 10%
fontos pixel
(3-hoz)



1 pixel
támadás
(nem 3 lesz)

Kihívások a tesztelés szempontjából

Követelmények:
Nem jól
specifikáltak

- Tanuló rendszerek: tanító halmaz áll rendelkezésre – mennyire jó?
- Mit garantálhatunk a többi bemenetre?

Részletes tervek:
Nehezen
megfoghatók

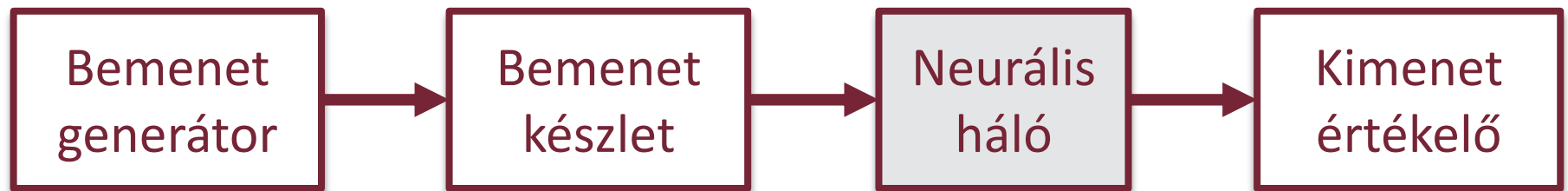
- Tudásreprezentáció: komplex struktúra
- Neurális háló rétegei: tanult paraméterek
- Bemenetek hatásossága neuronokra?

Teszt minőség:
Nehezen
mérhető

- Elégtelenek a klasszikus metrikák (követelmények, struktúra, forráskód fedettsége)
- Mi legyen a fókusz: a bemeneti domén?

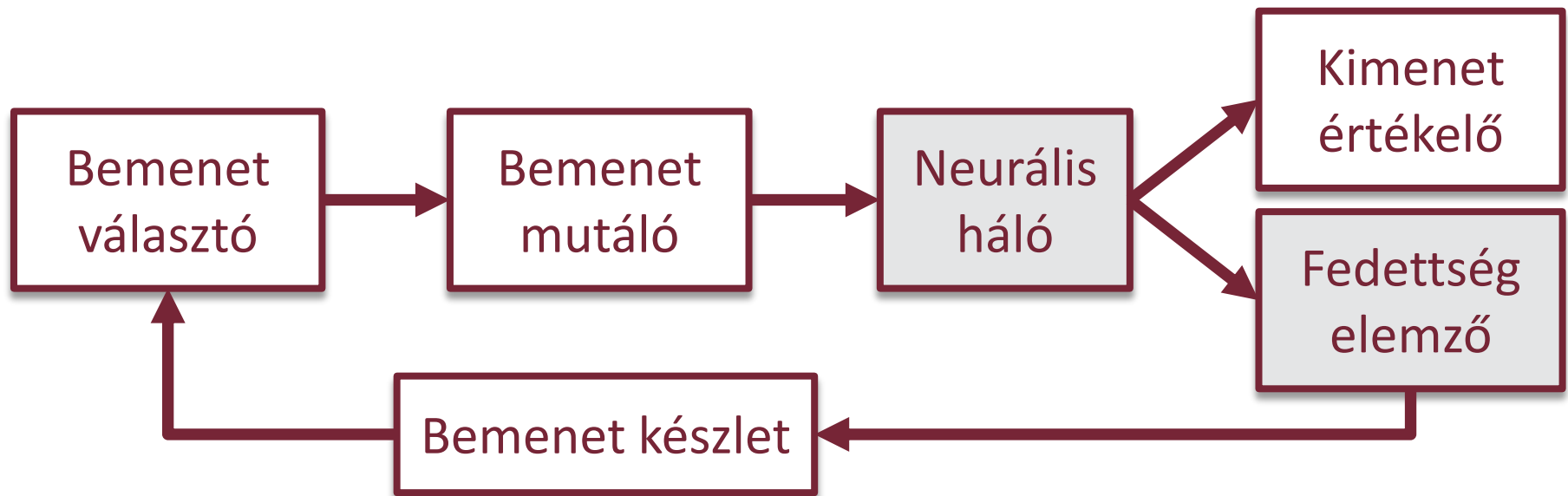
Fekete dobo tesztelés

- Bemenetek automatikus generálása
 - Készlet alapú, szisztematikus és mutációs módszerek
 - Nincs hozzáférés a háló belsejéhez



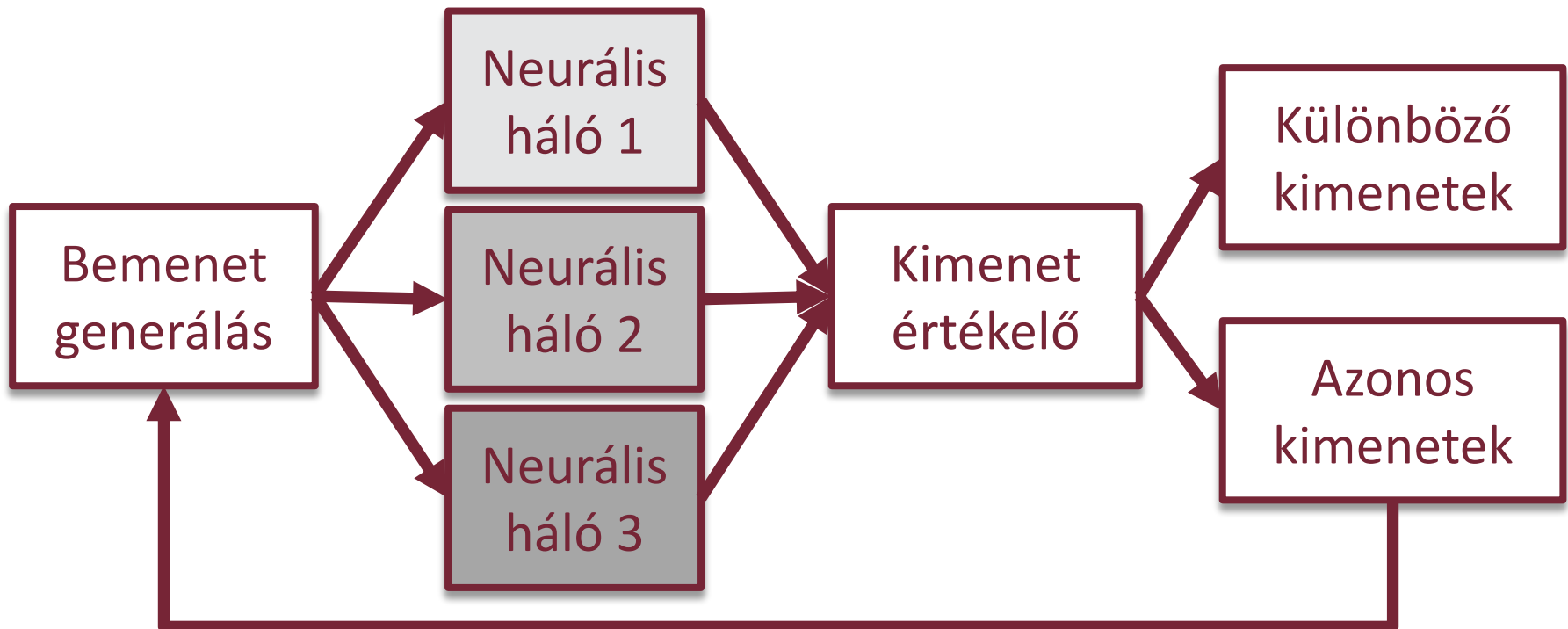
ostrich
spoonbill
beaver
armadillo
cradle
reel
macaw
...

- Bemenetek módosítása: Fuzzing
 - Zaj vagy előre definiált mutációk bemeneti készleten
 - Kiválasztás a fedettségi elemzés alapján: a neuronok kimenetének aktiváltsága



Differenciális tesztelés

- Ellenséges bemenetek keresése adott hálókhöz
 - Hálók képességeinek vizsgálata: Több, eltérő módon tanított hálóval, nagy neuron fedettséggel
 - Azonos bemenetre adott kimenetek értékelése



További megközelítések – A mélység titkai

■ DeepSafe

- A háló robusztusságát vizsgálja ellenséges bemenetekre
- Adat-vezérelt klaszterezéssel keresi a biztonságos régiókat



■ DeepXplore

- Rosszul felismert bemeneteket keres különféle módon betanított hálókat alkalmazva

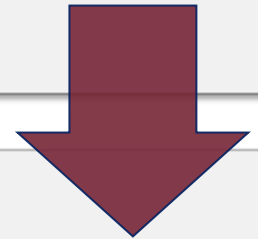
■ DeepTest

- Autonóm járművek tesztelésére fókuszál
- Kép transzformációkkal generál új bemeneteket (pl. köd, eső hatása)

Új szemlélet biztonságkritikus rendszerekben

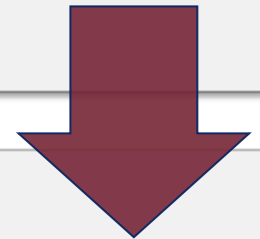
Biztonságkritikus rendszerek ellenőrzése

- Klasszikus: A **tervezett funkcionalitás** ellenőrzése (hibamódokra)
- Kiterjesztés: A **szándékolt funkcionalitás** szerinti ellenőrzés
Road vehicles – Safety of the Intended Functionality (SOTIF)



Fókuszálás új kockázatokra: Elégtelen funkcionalitás

- Komponensek, szenzorok **képességeinek** korlátai
- **Környezeti változások** kezelésének korlátai
- A funkcionalitás használata **nem validált környezetben**



Jellegzetes irányok az ellenőrzésben

- Tervek **finomításának** elfogadhatósága
- Determinisztikus működés → **Adaptív működés**
- Kimerítő tesztelés → **Bemenet szegmentálás**

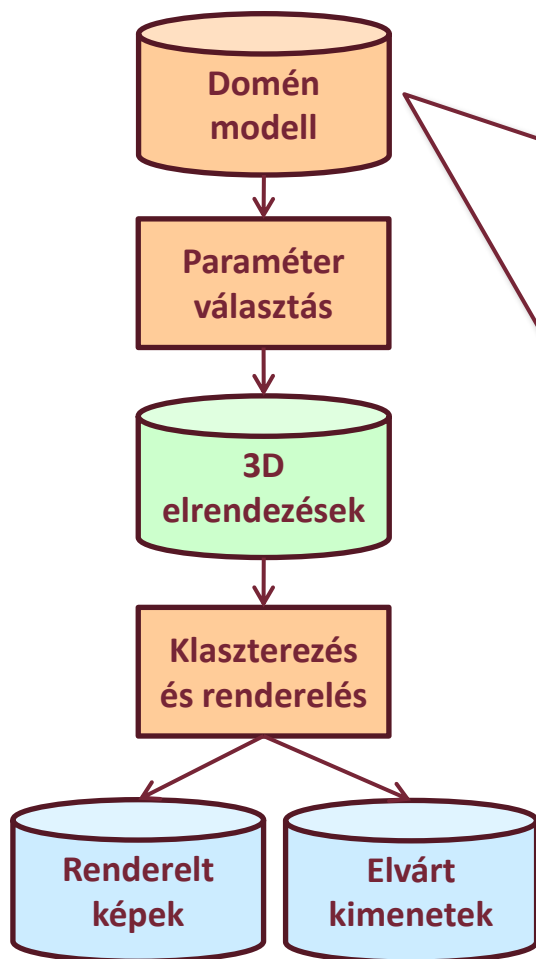
Esettanulmány: Objektum felismerés tesztelése



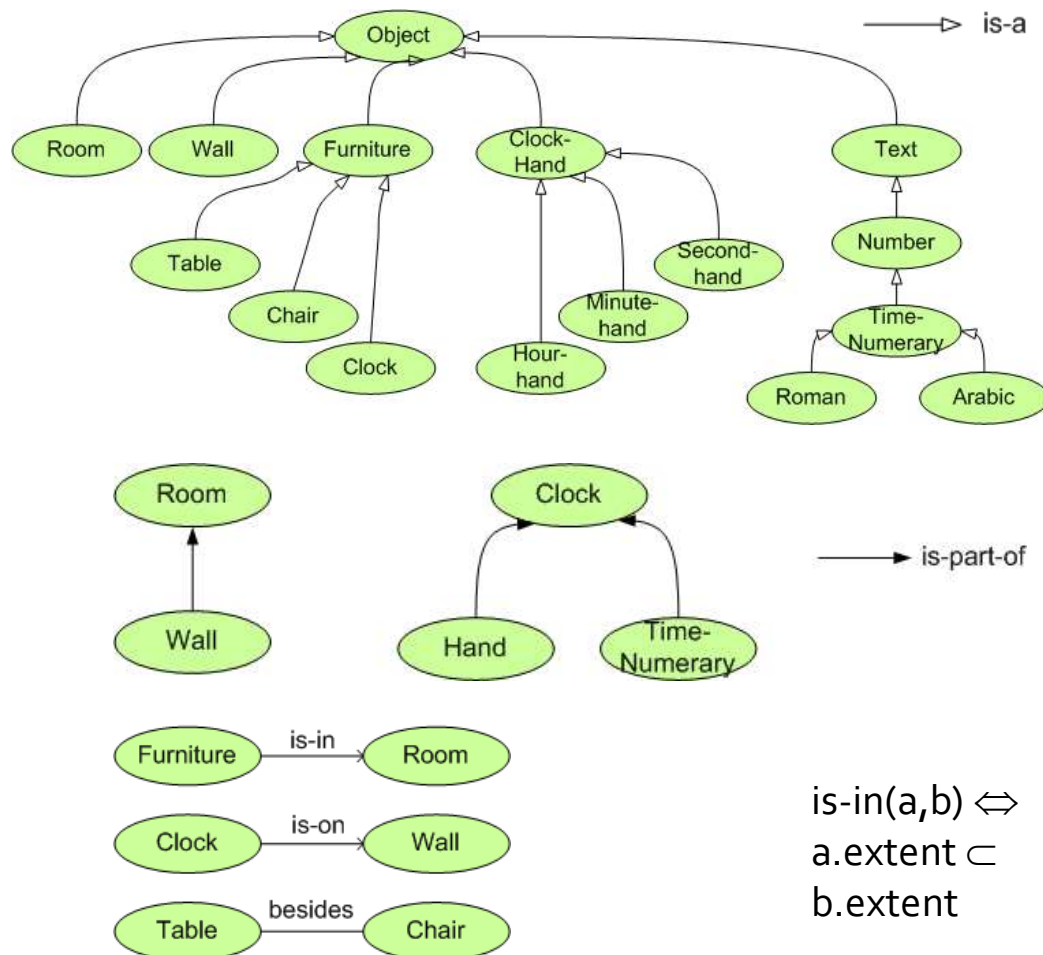
- Háztartási robot tesztelése
 - Kamera kép → Felismert objektumok
 - Teszt: Renderelt képek és elvárt eredmények; szisztematikus generálás
- Kihívások
 - Objektumok és nézetek sokfélesége
 - Minimális, reprezentatív tesztkészlet
- Megoldási ötletek
 - Domén analízis: Objektum kategóriák
 - Kritikus paraméterek megállapítása
 - Hasonló képek klaszterezése



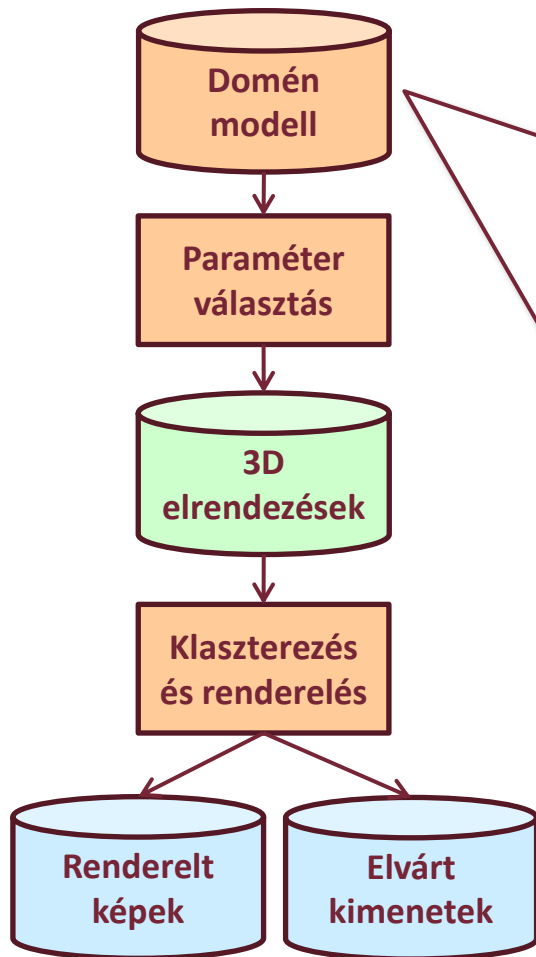
A tesztgenerálás folyamata



Objektum típusok, paraméterek, relációk, kényszerek

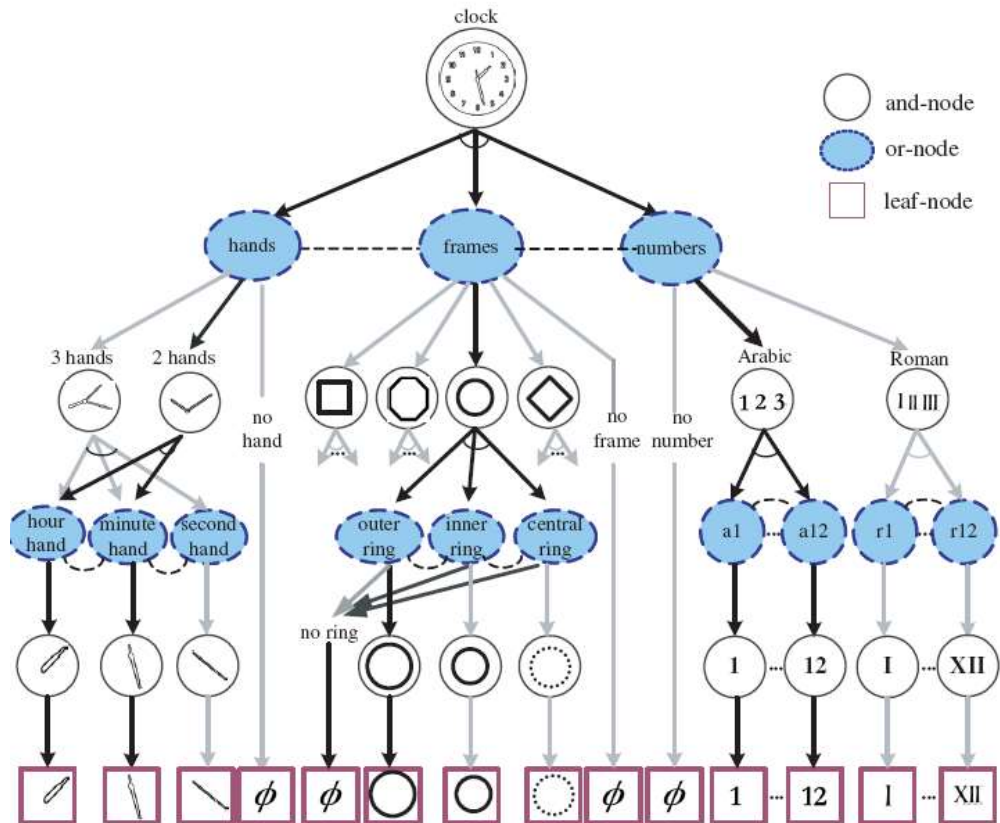


A tesztgenerálás folyamata

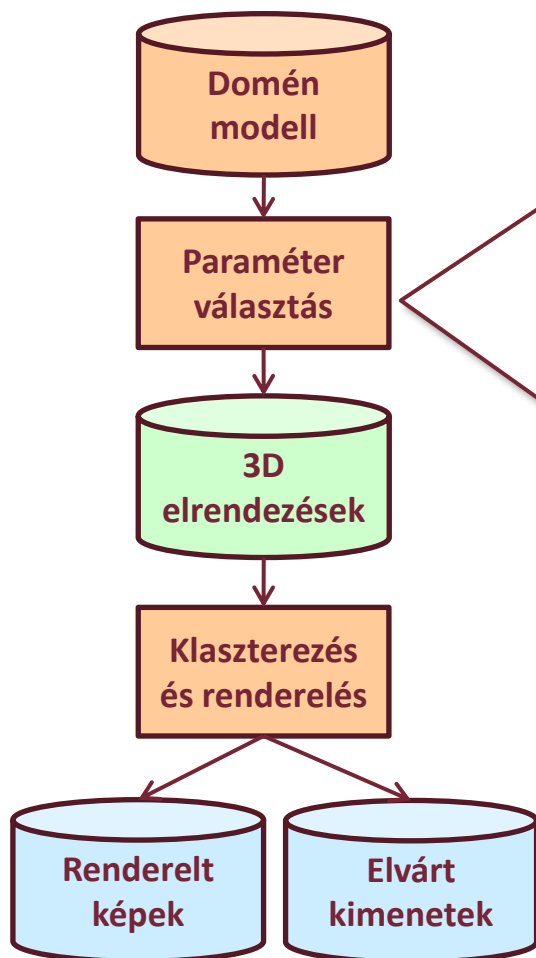


Objektumok variabilitása

- AND-OR struktúra



A tesztgenerálás folyamata



Generálható formák

- Prioritások a paraméterek hatása alapján

```
<generic_cup>
  <corners ptype="integer" min="3" max="32"/>
  <bottom ptype="enum" enum="extruded,depressed,regular"/>
  <handle>
    <type ptype="enum" enum="none,D_shaped,rectangular_shaped,S_shaped,triangle_shaped"/>
    <projected_height ptype="real" min="0.0" max="1.0"/>
  </handle>
  <handle_size ptype="integer" min="4" max="5"/>
  <bottom_diameter>
  <center_diameter>
  <top_diameter>
  <smoothing_pt>
  <xy_convexity>
  <xz_convexity>
  <extruded_top>
  <image_path ptype="string" min="1" max="100"/>
  <data_path ptype="string" min="1" max="100"/>
</generic_cup>
```

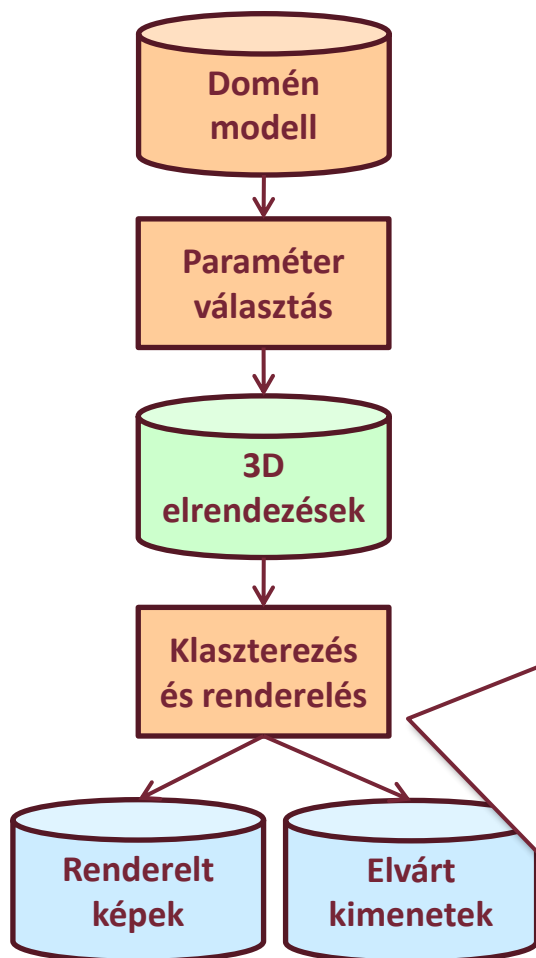


Generálható 3D elrendezések

- Kényszerek teljesítése (greedy algoritmus)
- Relációk közötti prioritások a hatásuk alapján

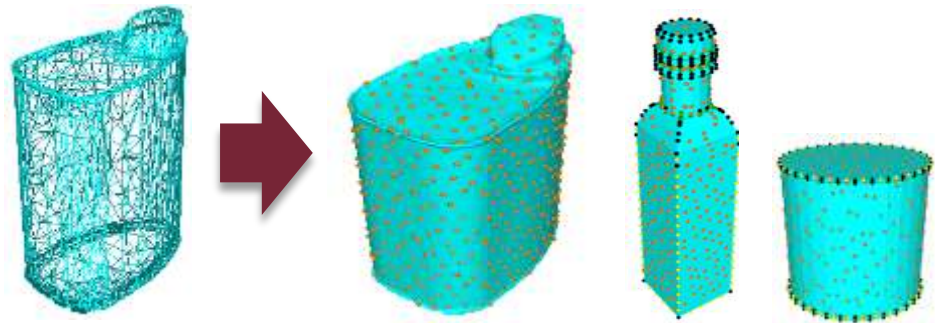


A tesztgenerálás folyamata

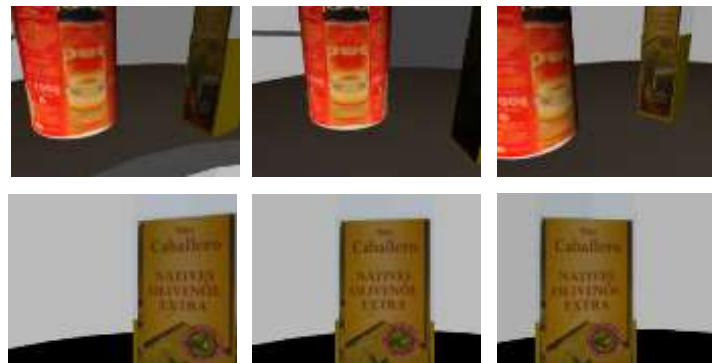


Tesztkészlet minimalizálás hasonlóság alapján

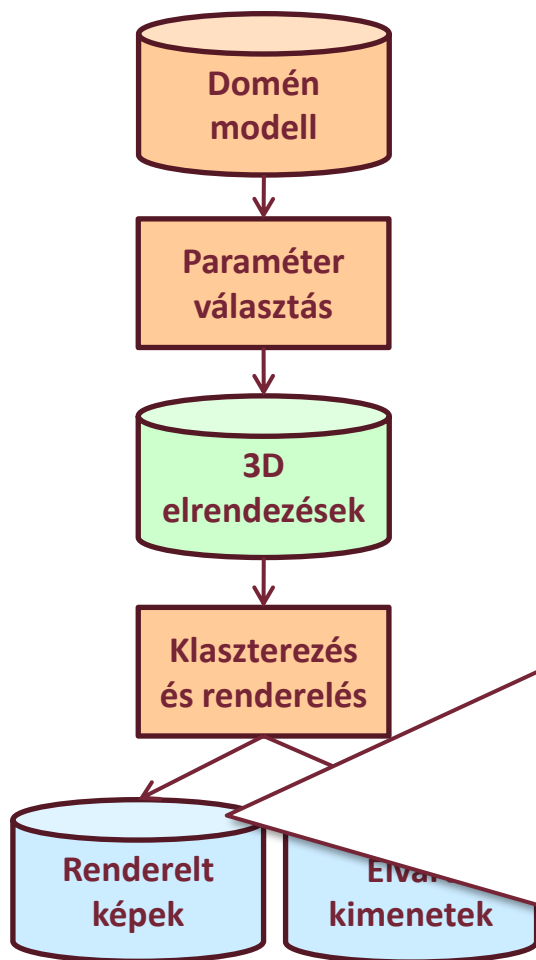
- Elrendezés jellemzése: trait points (Hammersley), trait tags (forma, szín, orientáció, árnyék, ...) → kvantitatív jellemző számítható



- Klaszterek képzése, ezekből reprezentatív eset választás



A tesztgenerálás folyamata



Generált képek 3D elrendezés alapján

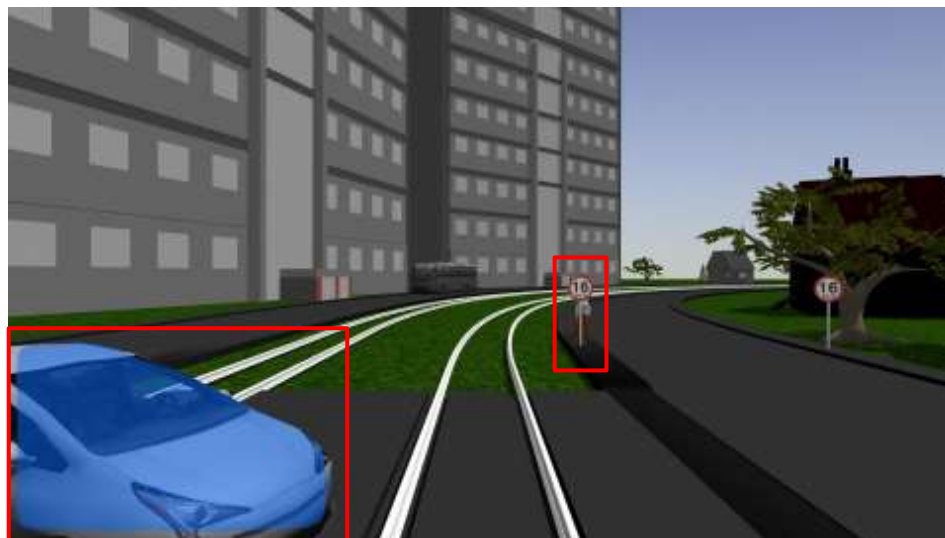


Paraméterek:

- Fényforrás: pozíció, felület, spektrum, intenzitás, polarizáció
- Közeg: átlátszóság, spektrum, textúra, részecskék
- Megfigyelő: helyzet, látószög
- Kamera hibák: tükröződés, lencsehibák, elmozdulás, ...



Esettanulmány: Környezetfüggő viselkedés tesztelése



Autonóm járművek és robotok tesztelése

- Nagyszámú releváns faktor

Működési terület: 8

Objektum felismerés: 10

Esemény felismerés: 7

Reagálás: 4

Rendszer-hibák: 8

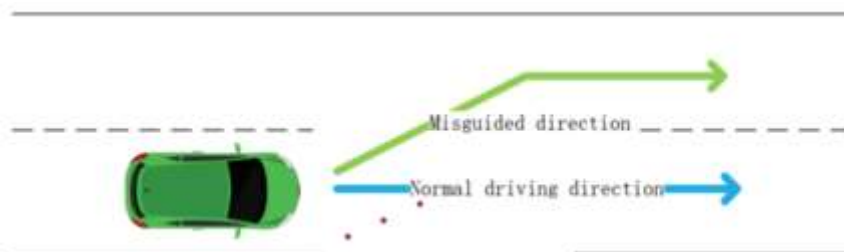
Hiba-kezelés: 6

Funkcionális korlátok: 8

- Sok hibalehetőség

Hacking street signs with stickers could confuse self-driving cars

Subtle or camouflaged optical hacks can change a stop sign into something else.



TESLA AUTOPILOT —

Researchers trick Tesla Autopilot into steering into oncoming traffic

Stickers that are invisible to drivers and fool autopilot.

Artificial ignorance: The 10 biggest AI failures of 2017

From self-driving car accidents to Face ID hacks, artificial intelligence didn't have a flawless year.

Megoldási lehetőségek



Valós környezetben végzett tesztelés

- Valós szituációk és problémás esetek összegyűjthetők
- Magas biztonságintegritási szinthez: sok teszt óra és megtett kilométer szükséges

- Emberi vezetés: 1,09 halálos baleset 100 millió mérföldre
- Autonóm jármű: 275 millió mérföldnyi futással tudja ezt 95%-os konfidenciaszinttel demonstrálni
- 100 autonóm jármű napi 24 órában 12,5 év alatt tudja ezt teljesíteni

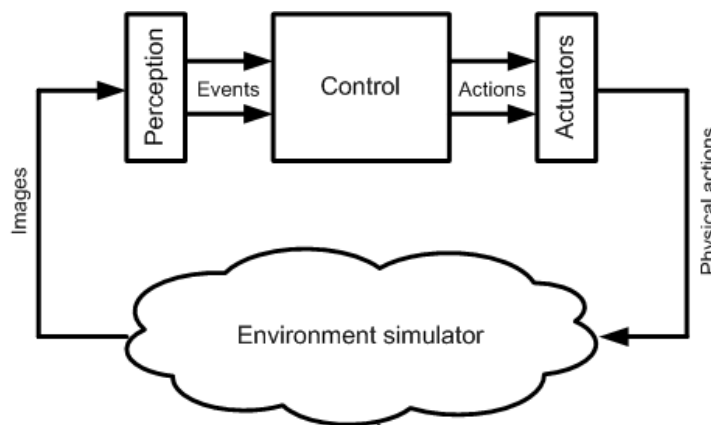
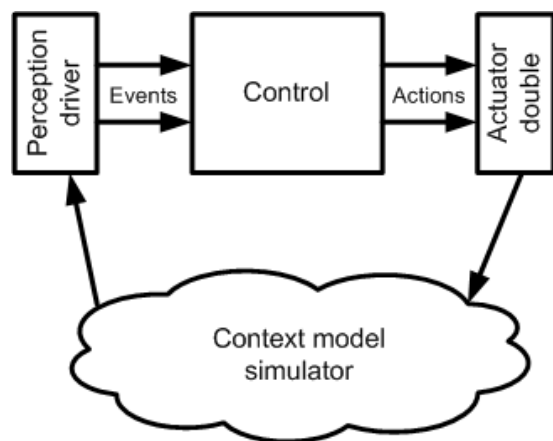


Szimulátorban történő tesztelés

- Szituációk szisztematikus generálása
- Biztonságos viselkedés monitorozása

Példa projekt: Célkitűzések

- Környezetfüggő viselkedés tesztelése autonóm robotok esetén



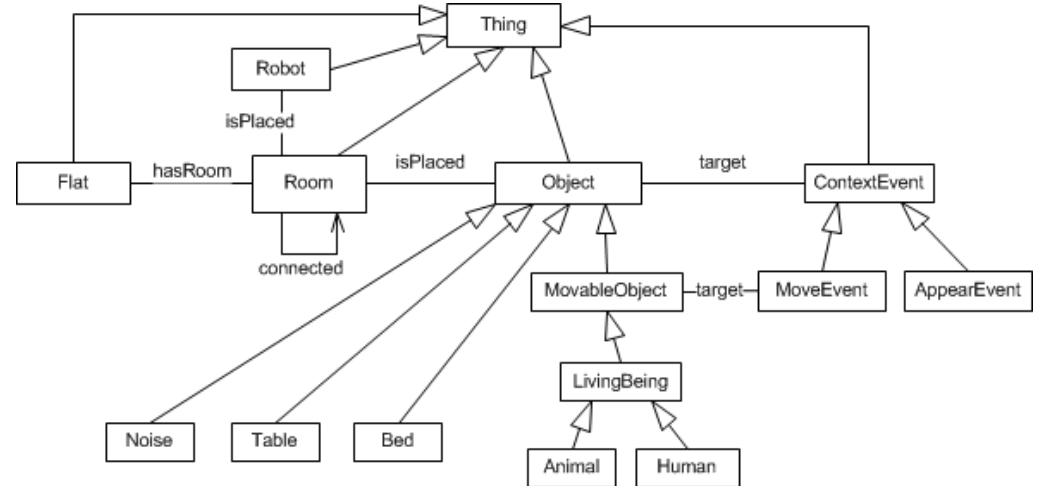
- **Teszt környezetek generálása** robot misszió végrehajtásához
- **Modell alapú megközelítés**
 - **Környezeti modell:** Mi lehetséges?
 - **Követelmény modell:** Mi az elvárás?
 - **Teszt stratégia:** Mit érdemes ellenőrizni?



Modellezés: Környezet és követelmények

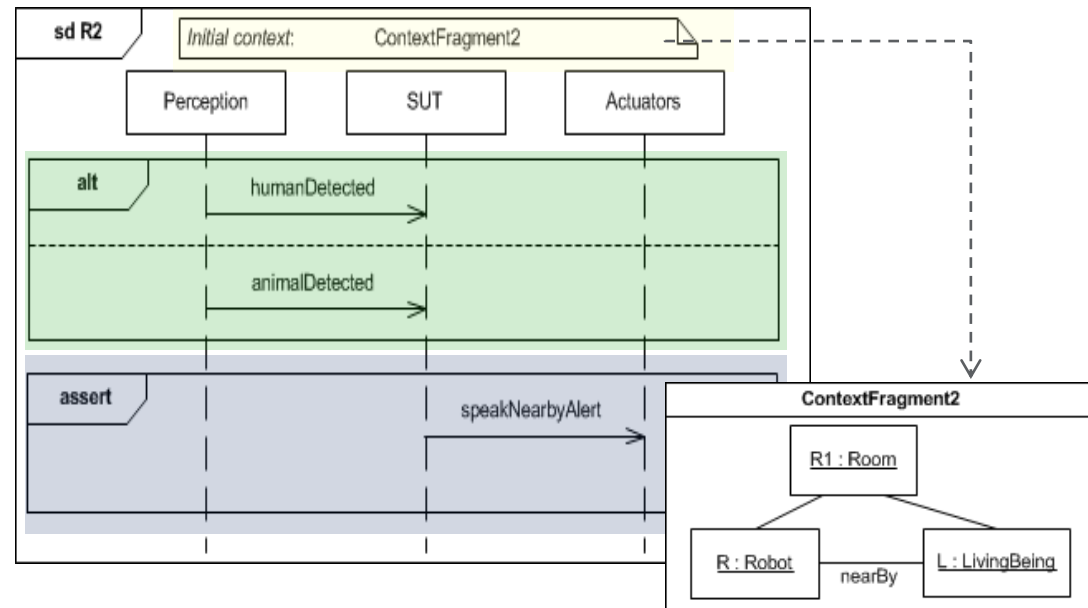
■ Környezeti metamodel

- Objektumtípusok, tulajdonságok, kapcsolatok
- Kényszerek: fizikai vagy alkalmazás-specifikus



■ Követelmény scenario modellek

- **Szituáció:** környezeti minta
- **Érzékelés:** szenzor bemenetek
- **Elvárt viselkedés:** akció szekvencia



Tesztelési célok robusztusság teszteléshez

Váratlan objektumok

- Szituációk kibővítése extra objektumokkal
- Automatikus környezeti modell generálás
- Fedettség: **Környezeti objektumok** $\times$ **szituációk**

Komplex szituációk

- Szituációk n-szeres kombinálása
(komplex szituációra több követelmény vonatkozik)
- Fedettség: **(Követelmény szituáció)ⁿ**

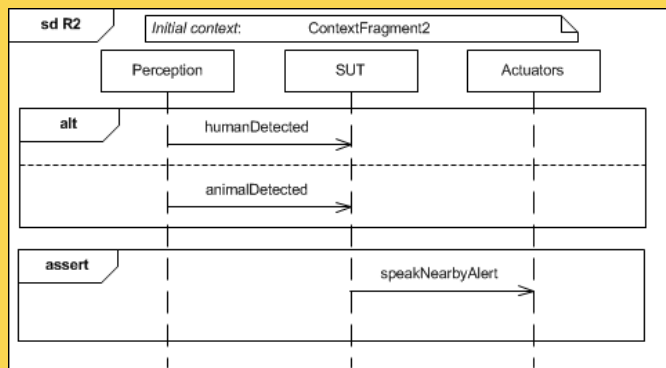
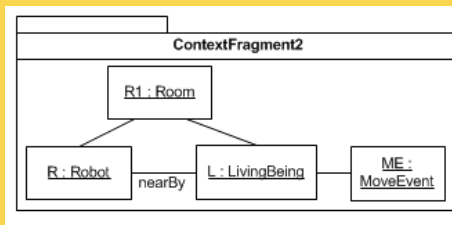
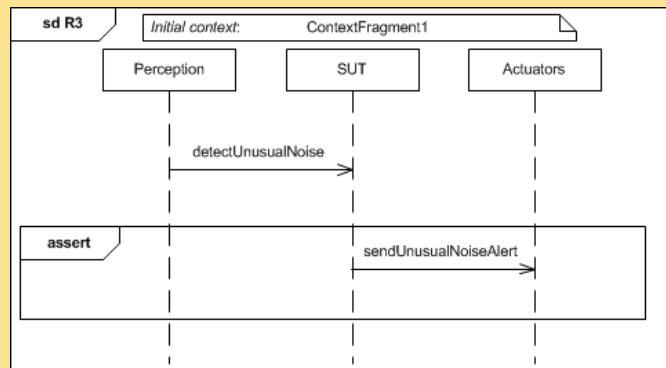
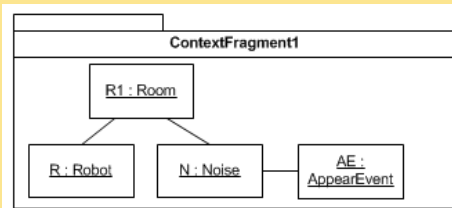
Extrém szituációk

- Szituációk mutálása: objektum tulajdonságok határértékei, kényszerek szélső értékei, megsértése
- Fedettség: **Szituációk** $\times$ **(tulajdonságok | kényszerek)**

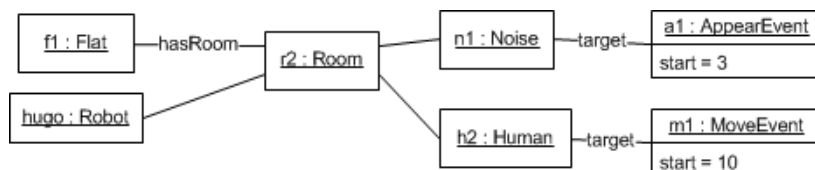
Modell alapú, szisztematikus tesztkészlet generálás

Példa: Komplex szituáció generálása

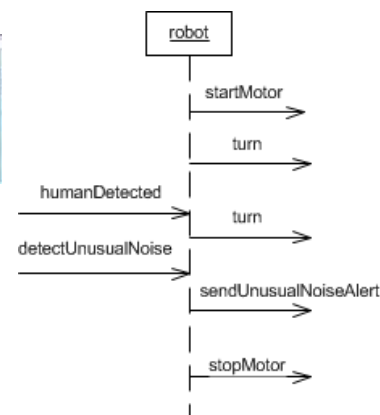
Követelmény scenario modellek:



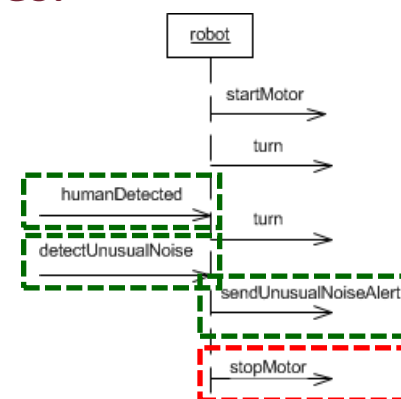
Generált teszt környezet:



Teszt lefutás:



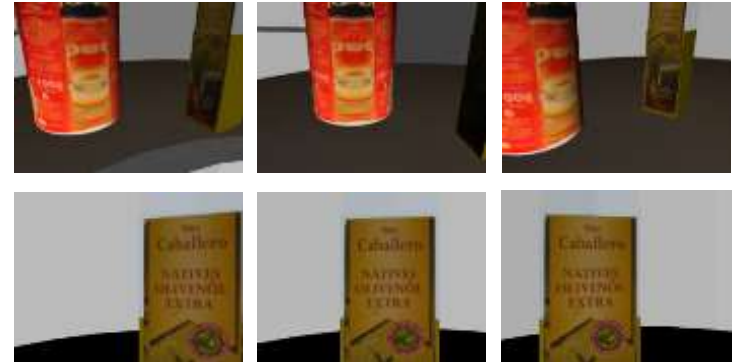
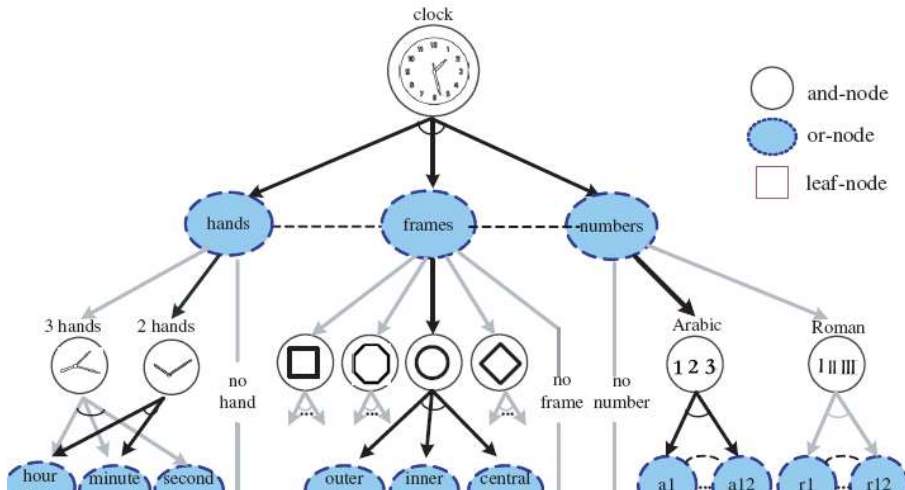
Kiértékelés:



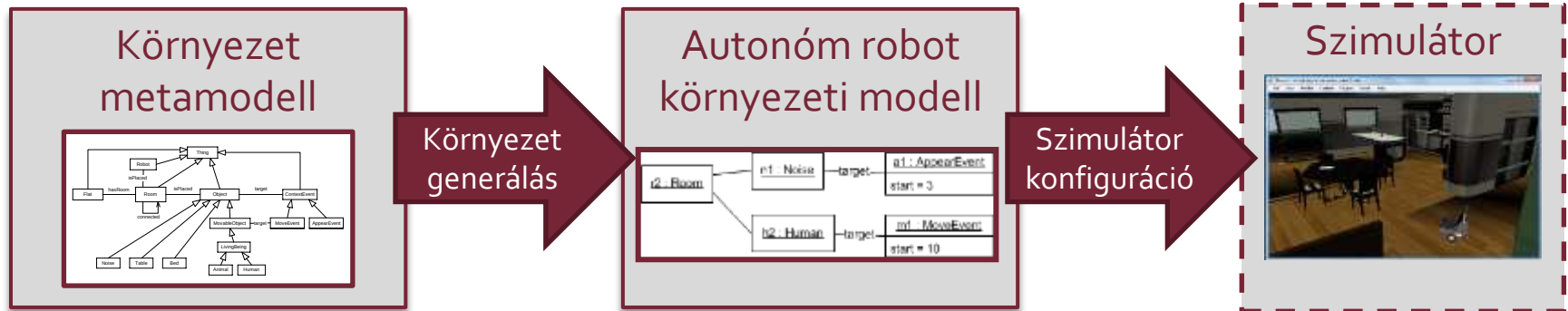
Test verdict:
R3 passed
R2 failed

Összefoglalás: Ötletek szisztematikus teszteléshez

- Modell alapú analízis, teszt bemenetek klaszterezése



- Teszt szituációk generálása fedettségi mértékekkel



- „Synthesizing robust adversarial examples”, A. Athalye, I. Sutskever, 2017.
- „Reluplex: An efficient SMT solver for verifying deep neural networks”, G. Katz, et al., 2017.
- „TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing”, A. Odena, I. Goodfellow, 2018.
- „Concolic Testing for Deep Neural Networks”, Y. Sun et al., 2018.
- „DeepXplore: Automated whitebox testing of deep learning systems”, K. Pei et al., 2017.
- „DeepSafe: A data-driven approach for checking adversarial robustness in neural networks”, D. Gopinath, G. Katz, C. S. Pasareanu, C. Barrett, 2017.
- „DeepTest: Automated testing of deep-neural-network-driven autonomous cars”, Y. Tian, K. Pei, S. Jana, B. Ray, 2017.
- „A Stochastic Grammar of Images”, S.-C. Zhu, D. Mumford, 2006.
- „How Many Operational Design Domains, Objects, and Events?”, P. Koopman, F. Fratrick, 2019.
- „Testing of Autonomous Systems - Challenges and Current State-of-the-Art”, P. Helle, C. Strobel, W. Schamai, 2016.
- „Testing Vision-Based Control Systems Using Learnable Evolutionary Algorithms”, N. Shiva, L. Briand, T. Stifter, 2018.
- „Driving to Safety – How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?”, N. Kalra, S. M. Paddock, 2016.
- „A Concept for Testing Robustness and Safety of the Context-Aware Behaviour of Autonomous Systems”, Z. Micskei, Z. Szatmári, J. Oláh, I. Majzik, 2012.
- „R3-COP: Resilient Reasoning Robotic Co-operating Systems”, ARTEMIS project, 2012.