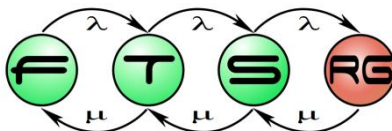


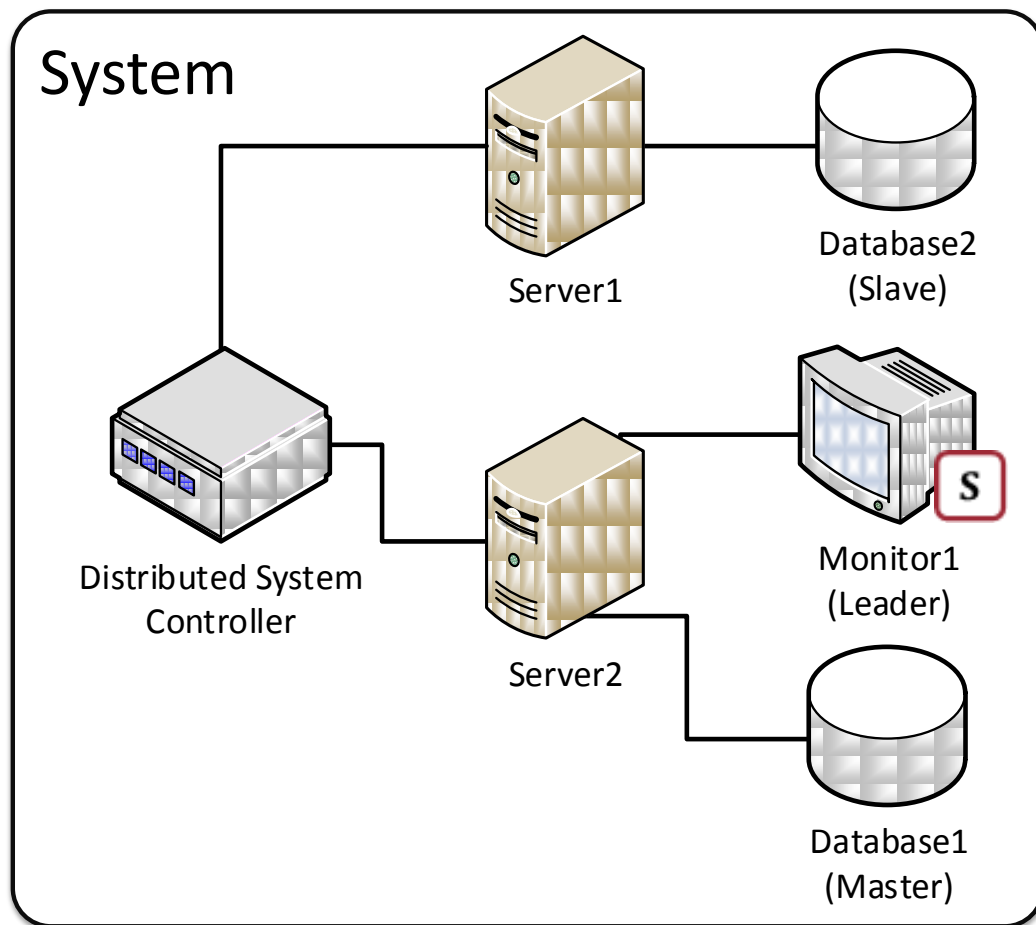
Modeltranszformációk tesztelése logikai megoldókkal

Semeráth Oszkár



Motiváció

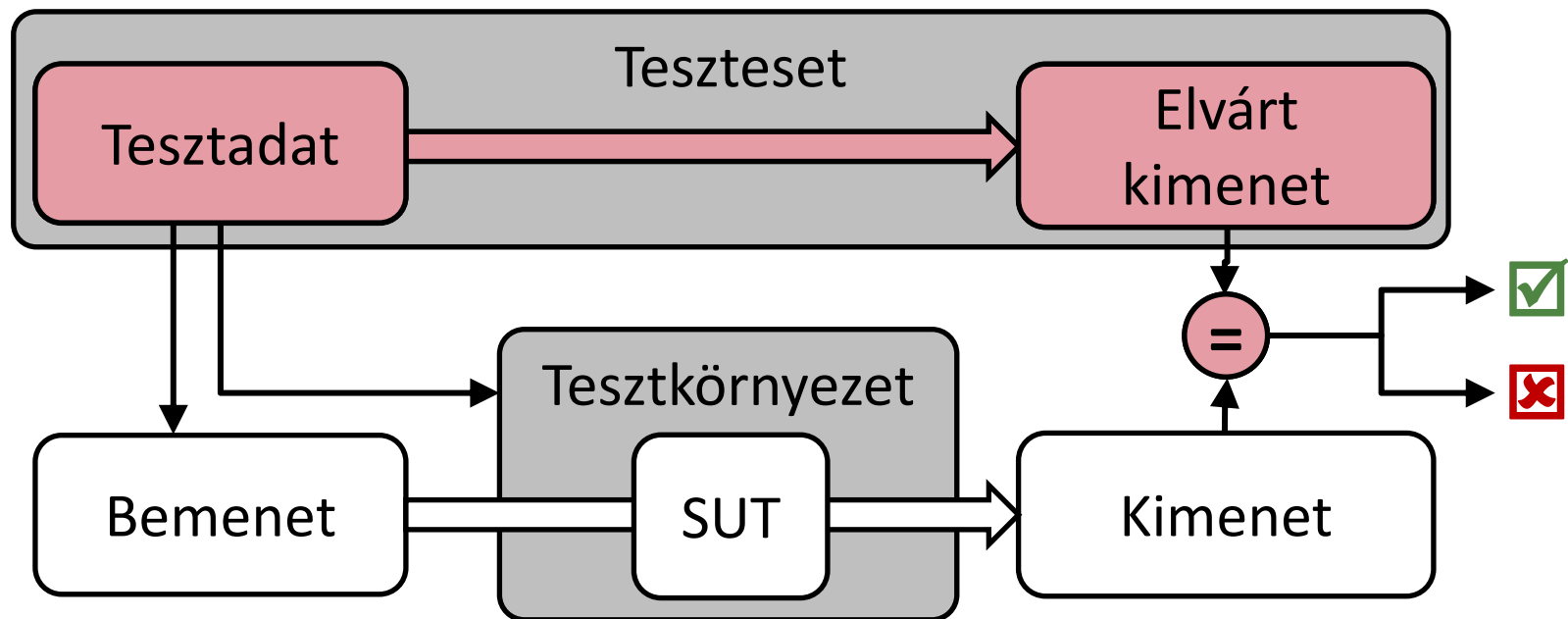
- Modellvezérelt tervezés
- Modelltranszformáció
= fő fejlesztési elem
- Ellenőrzés fontos
- Esettanulmány:
Egyszerűsített **CUDB**
Elosztott architektúra
Master – Slave
Leader – Follower
Hiba → Helyes
- Ericsson + R5Cop



Kihívások

„Verifying a transformation that transforms one artifact into another is fundamentally more complex than verifying an individual artifact itself” [1,2]

1. Változatos, értelmes tesztbemenetek + tesztkörnyezet
2. Elvárt kimenet előállítása
3. Tesztorákulum

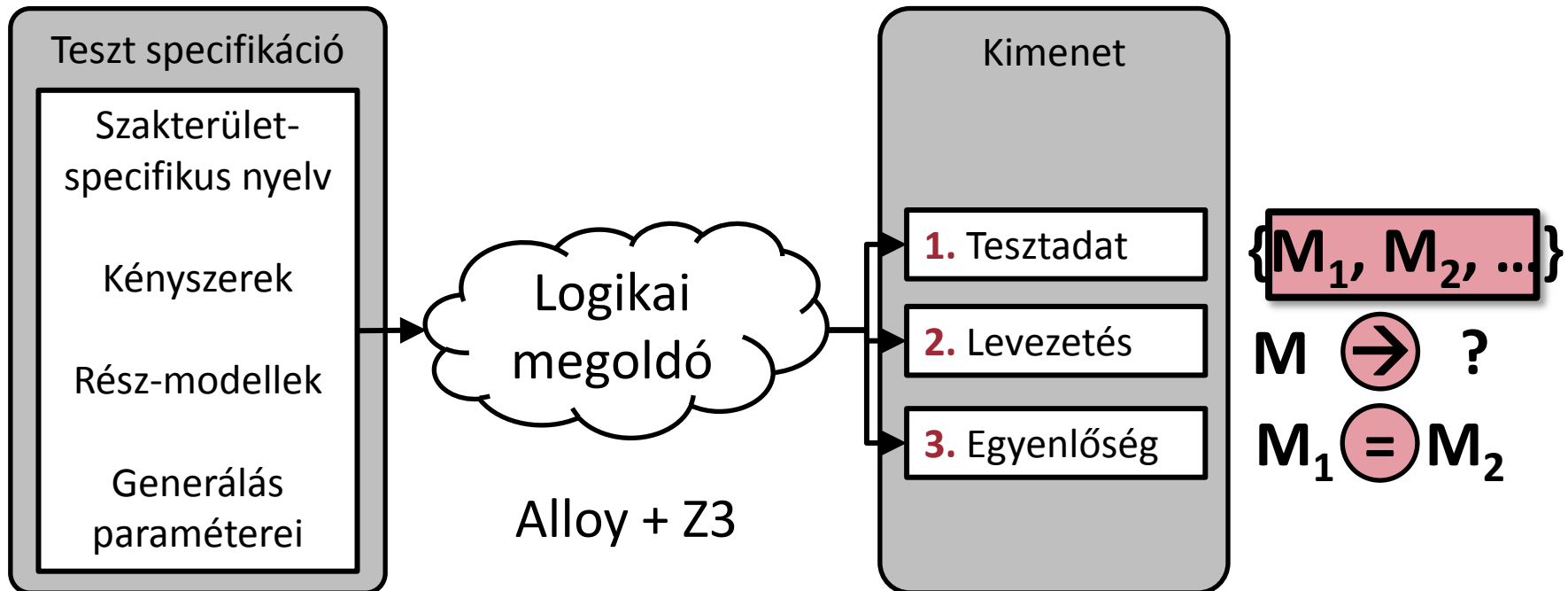


Megközelítés: Logikai megoldók

Modellezés → Probléma → Bizonyítás

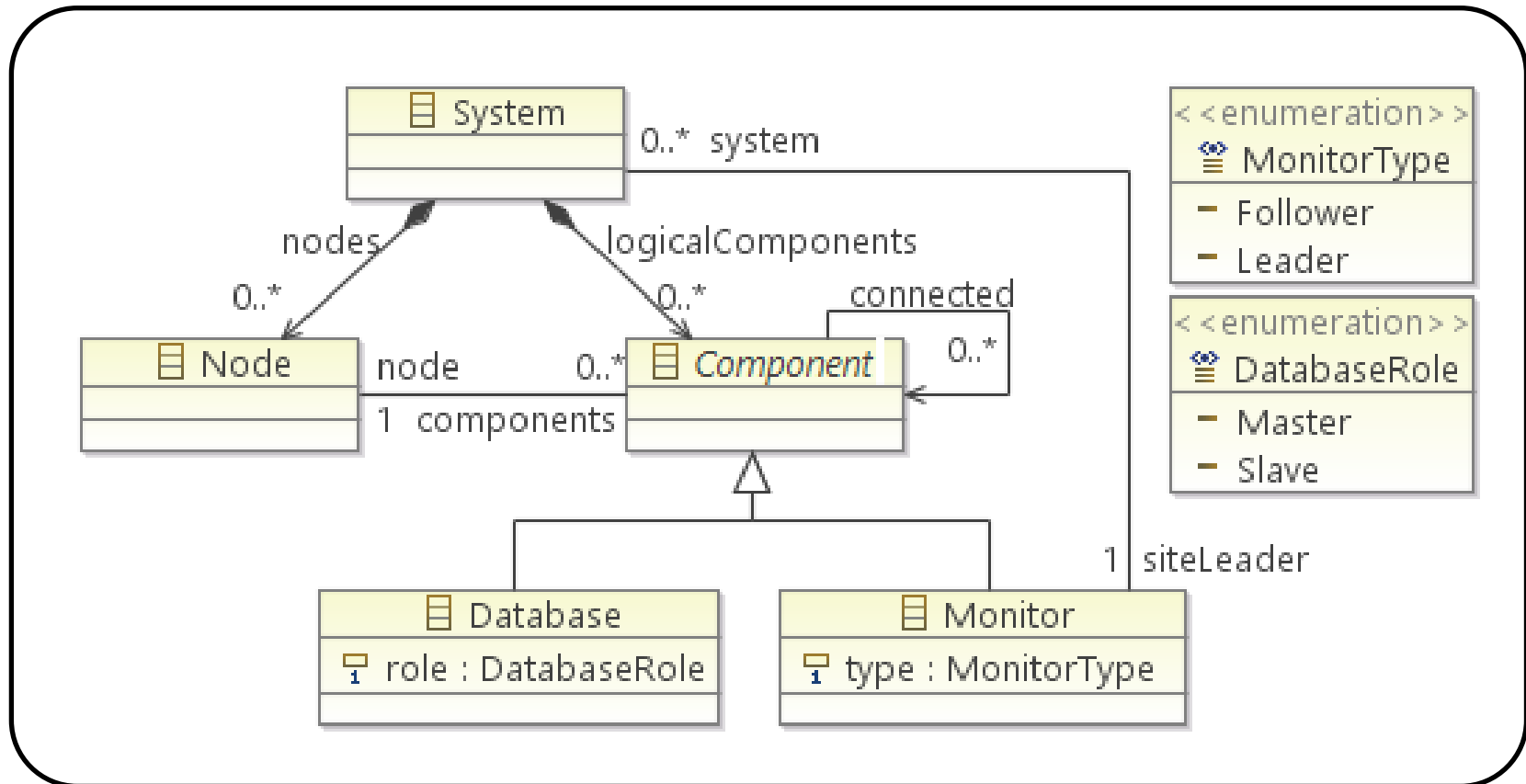
Példa / Ellenpélda

Bizonyítás / Cáfolat

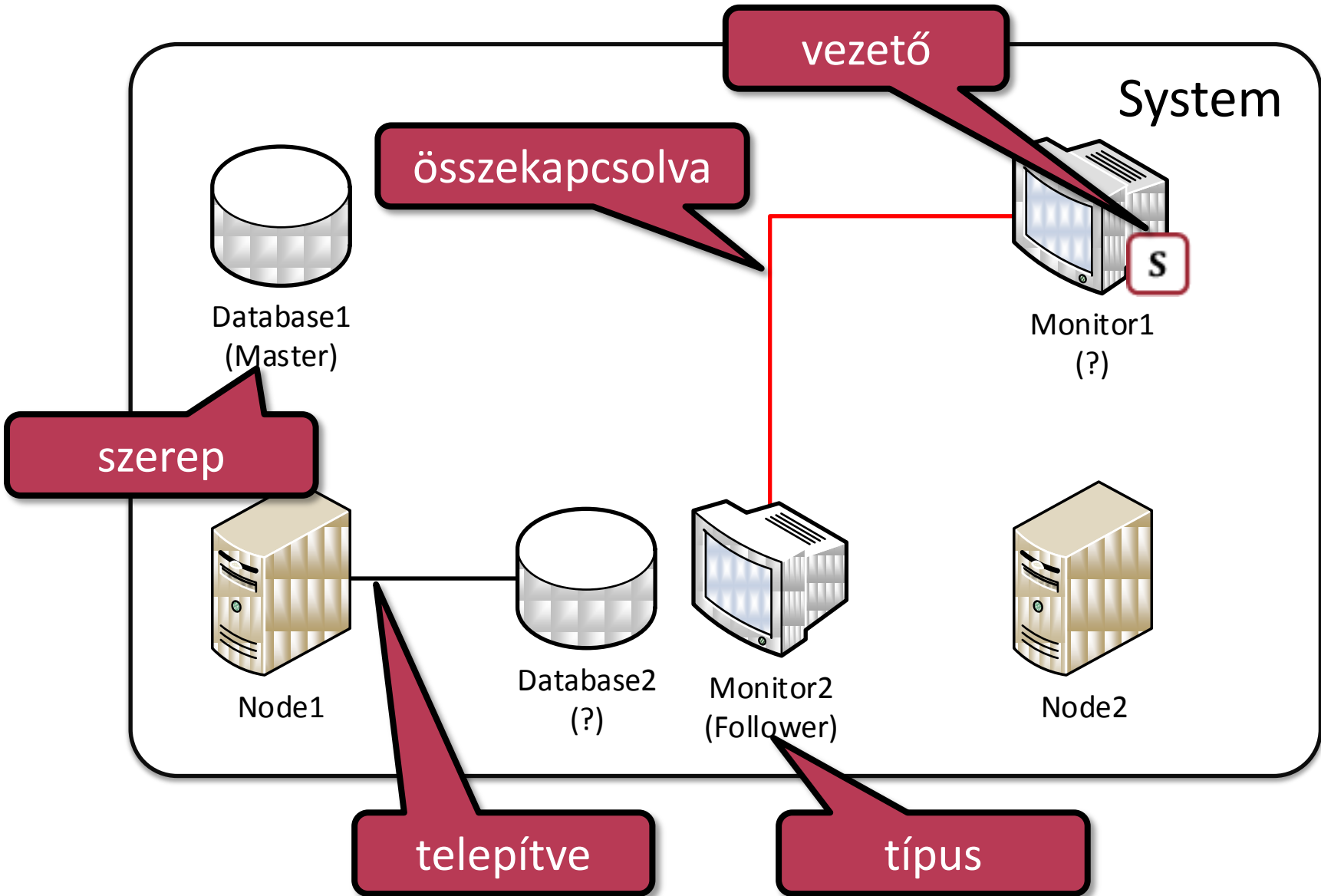


Metamodell

- A tesztelendő rendszer struktúráját, fogalmait, relációit leíró modell



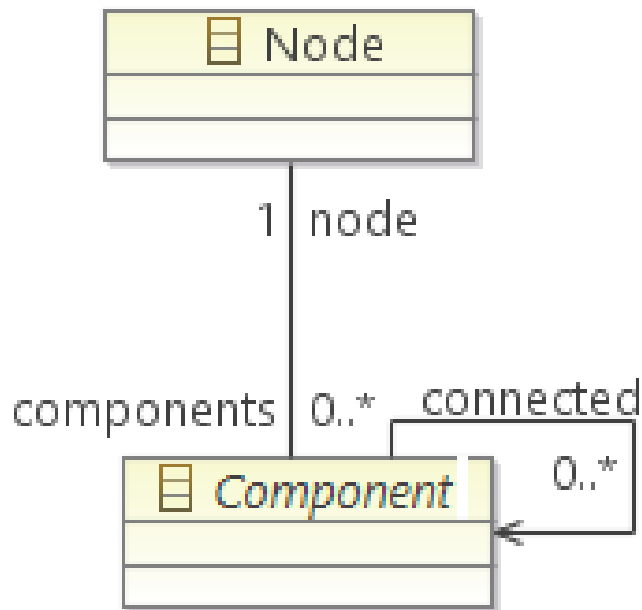
Kiindulási modell (Kezdeti konfiguráció)



Kényszerek (connOCL)

- OCL nyelven megfogalmazott
- Érvényes tesztadatok szűkítése

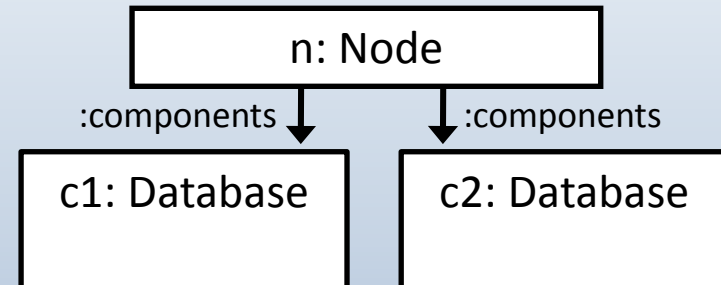
Nyelvi szint



OCL kényszer

- **connected** referencia

connected(c1, c2)

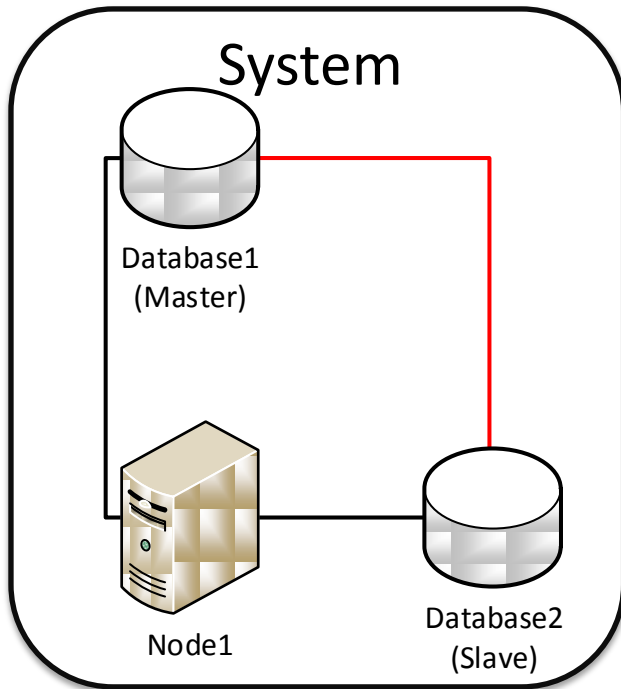


=> connected(c1, c2)

Kényszerek (connOCL)

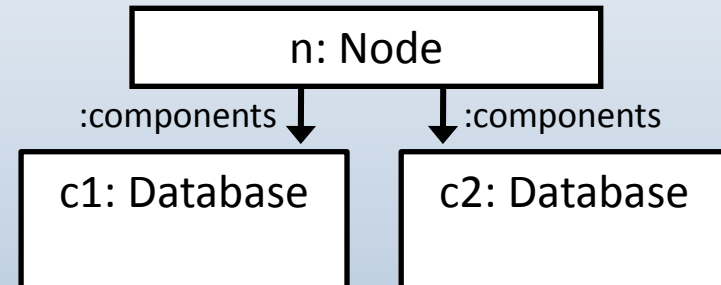
- OCL nyelven megfogalmazott
- Érvényes tesztadatok szűkítése

Példány szint



```
constraint connected= OCL invariant in "System"  
context "Component.allInstances()->forall(c1, c2 |  
if(not(c1=c2) and (c1.node=c2.node)) then  
    (c1.connected->includes(c2))  
else not(c1.connected->includes(c2)) endif )";
```

connected(c1, c2)

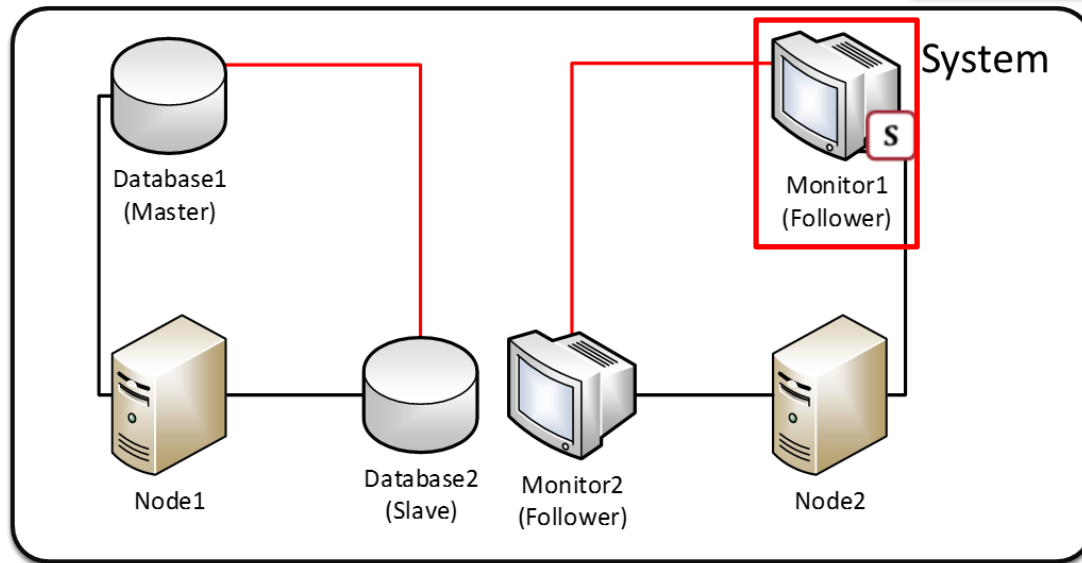


=> connected(c1, c2)

Kihívás I.: Értelmes Tesztesetek

- Modell méret: 7 (lehet tetszőleges is)

- Teszt kritérium: $\mathbf{KM} \wedge \mathbf{connOCL} \wedge$



- Kitekintés: nincs ellenpélda: Igazolja

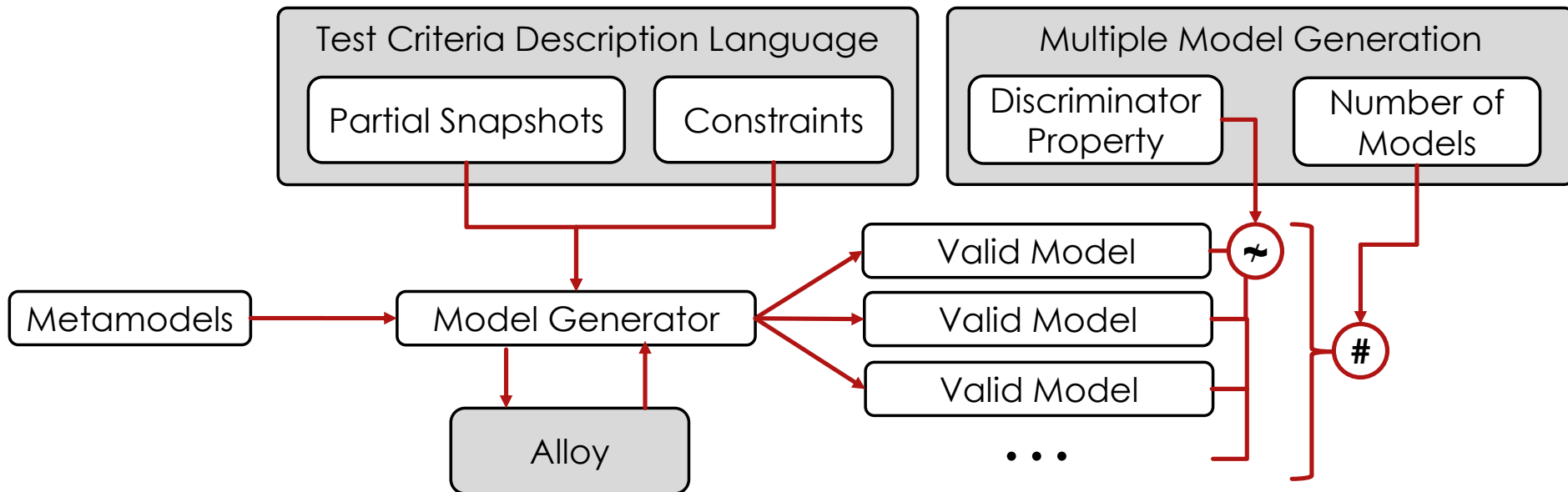
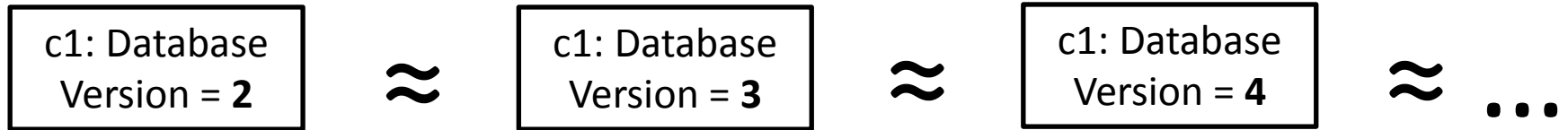
Pl: Részmodell \rightarrow OCL,

IncQuery \leftrightarrow OCL,

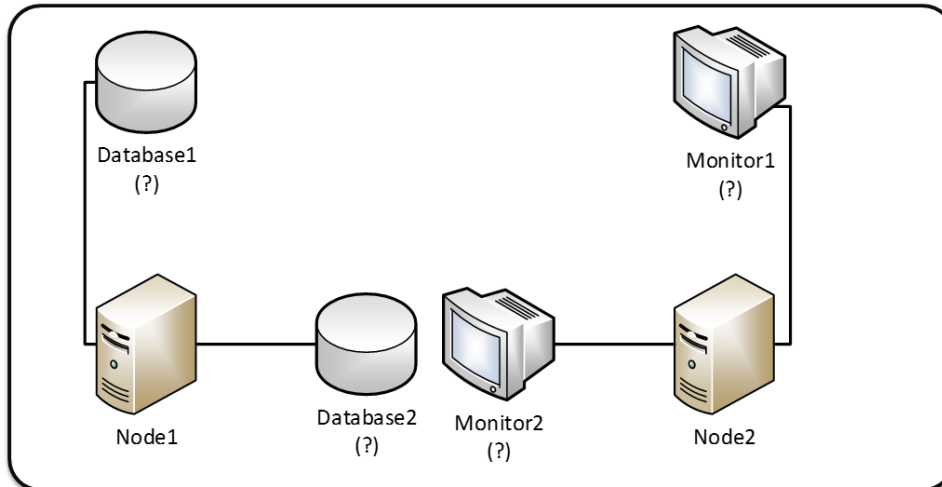
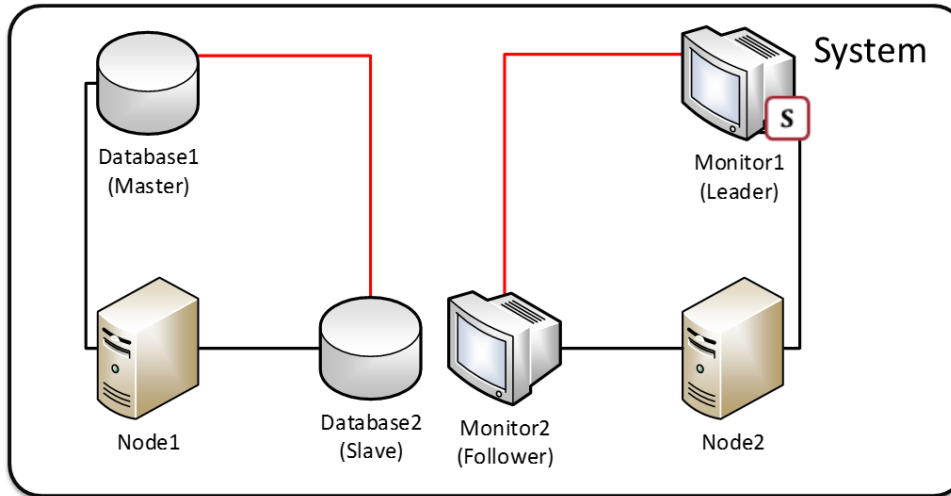
Derived_feature \wedge helytelen

Kihívás I: Változatos Tesztesetek

Tesztadat = példány modell (gráf + attribútumok)



Kihívás I: Változatos Tesztesetek



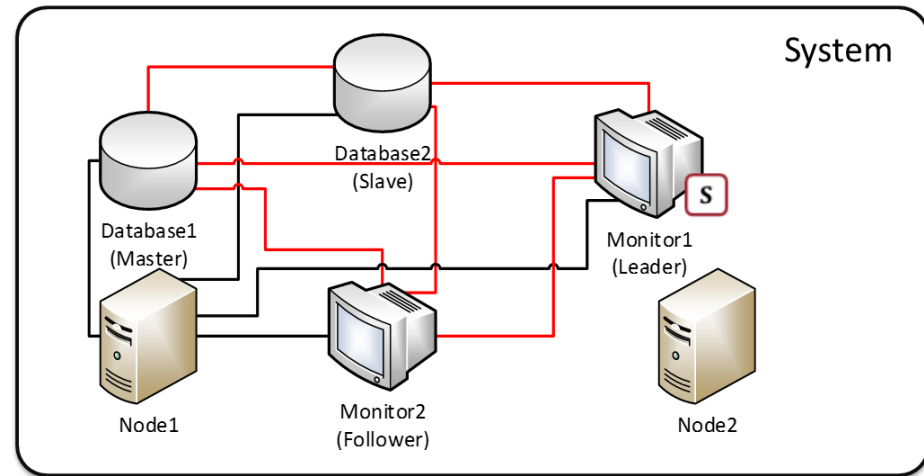
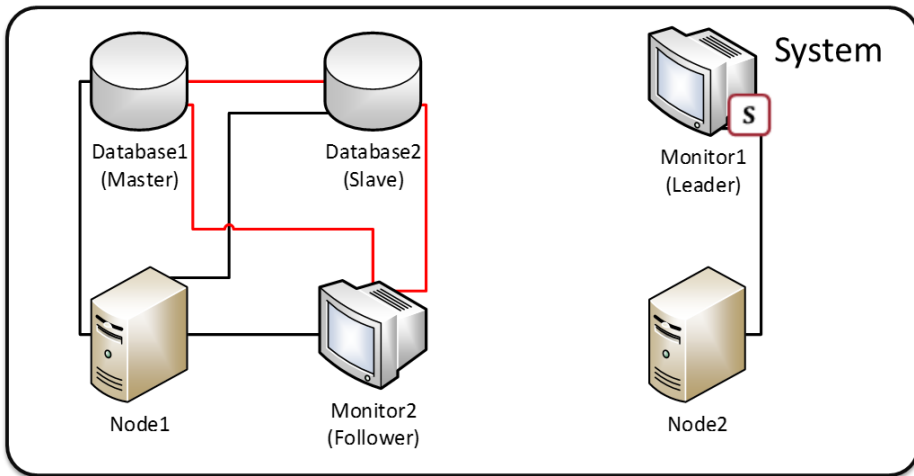
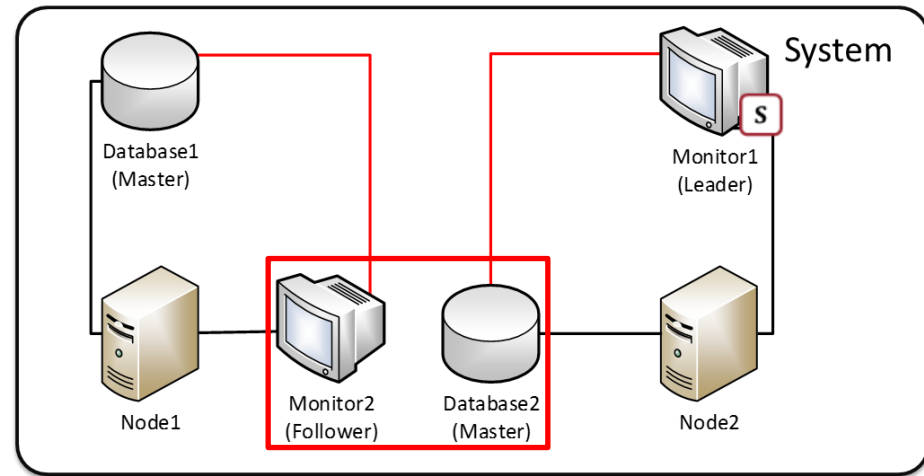
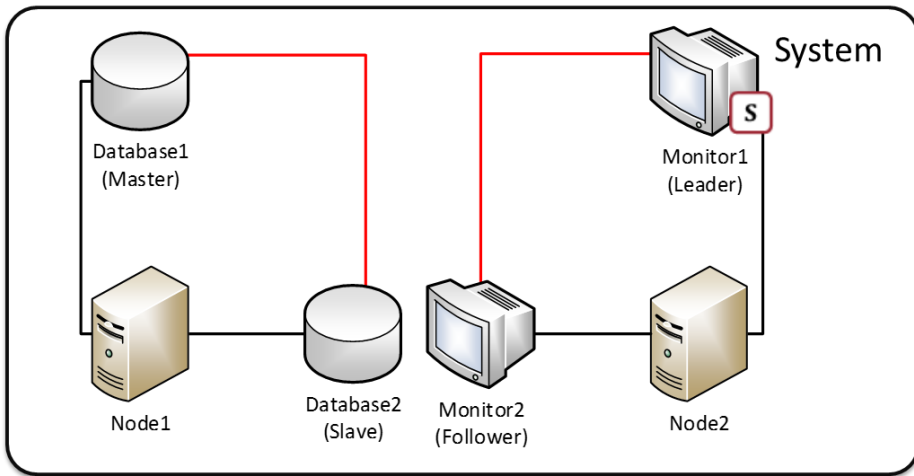
Ekvivalenciapartícionálás:
Minden ekvivalenciaosztályból egy példány

Változatossági kritérium:
Lényegtelen részletek elhagyásával is különbözzön

```
model generation =  
generate 4 different {  
    omitted: "System";  
    omitted: "siteLeader";  
    omitted: "system";  
    omitted: "connected";  
    omitted: "role";  
    omitted: "type";  
}
```

Lehetőségek:
Attribútumok
Referenciák
Egy típus összes eleme
Típusáltalánosítás
(Adatbázis → Szoftver)

Kihívás I: Változatos Tesztesetek



Többlépéses tesztgenerálás

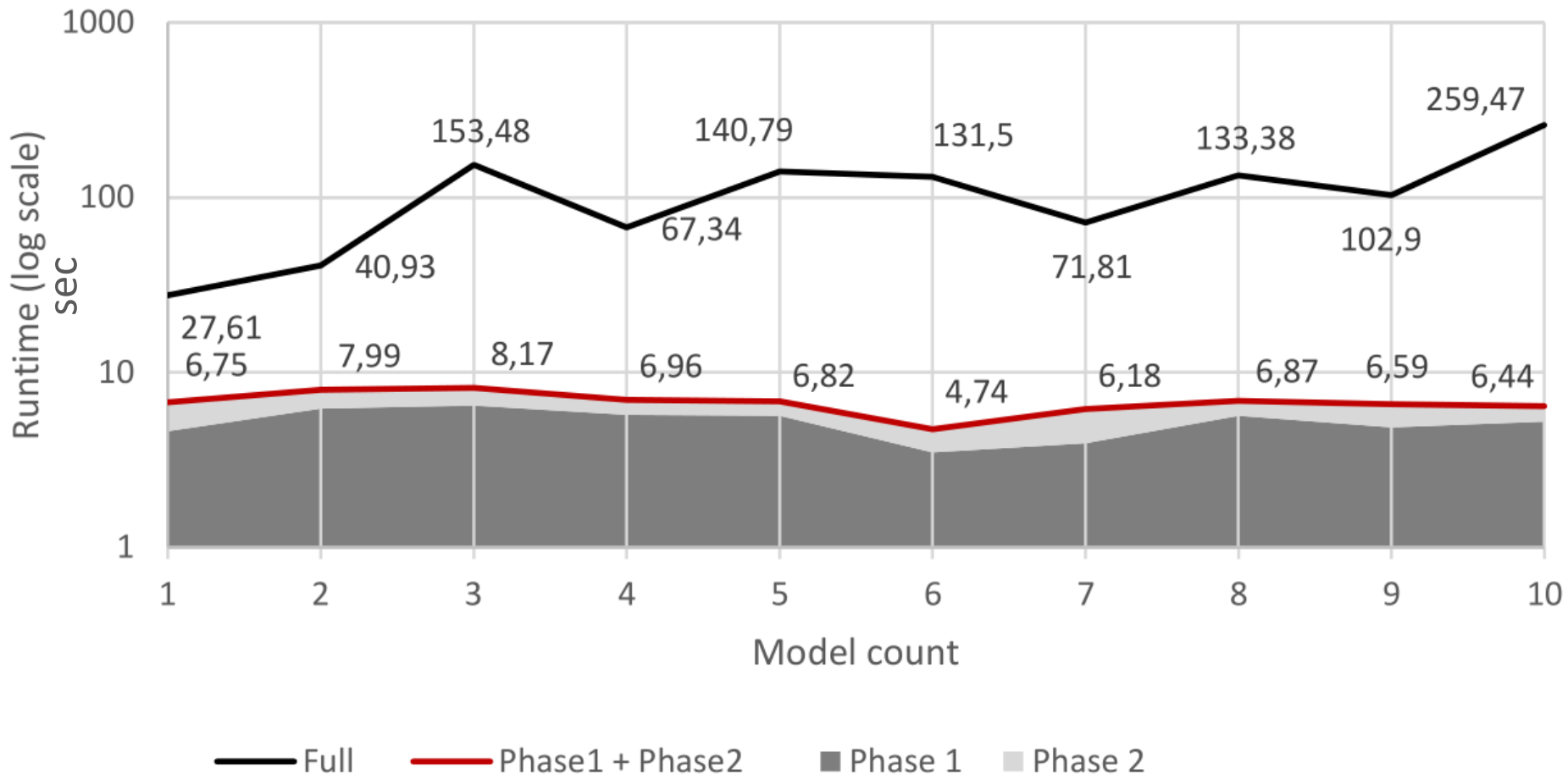
Egyik modellgenerálálás kimenet \rightarrow Másik bemenet

- Párhuzamosítás
- Egyék ebben jobb, másik abban \rightarrow
generálják külön
- Összefűzés: $A + B + C$ Több kicsi \rightarrow Nagy
- Kiegészítés: $A \subset B \subset C$ Kicsi+Kicsi+... \rightarrow Nagy
- Finomítás: $A \Leftarrow B \Leftarrow C$ Kicsi+Aspektus+...
- Optimalizálás: $M(A) < M(B) < M(C)$

Többlépéses tesztgenerálás

R3Cop Példa:

Kétlépéses finomítás \leftrightarrow egyben generálás



Kihívás II. és III.

- Transzformáció számítása:

Megvalósítás = Követelmény

- IncQuery minták illeszkedéseinek megadása
- Összehasonlítás az implementációval

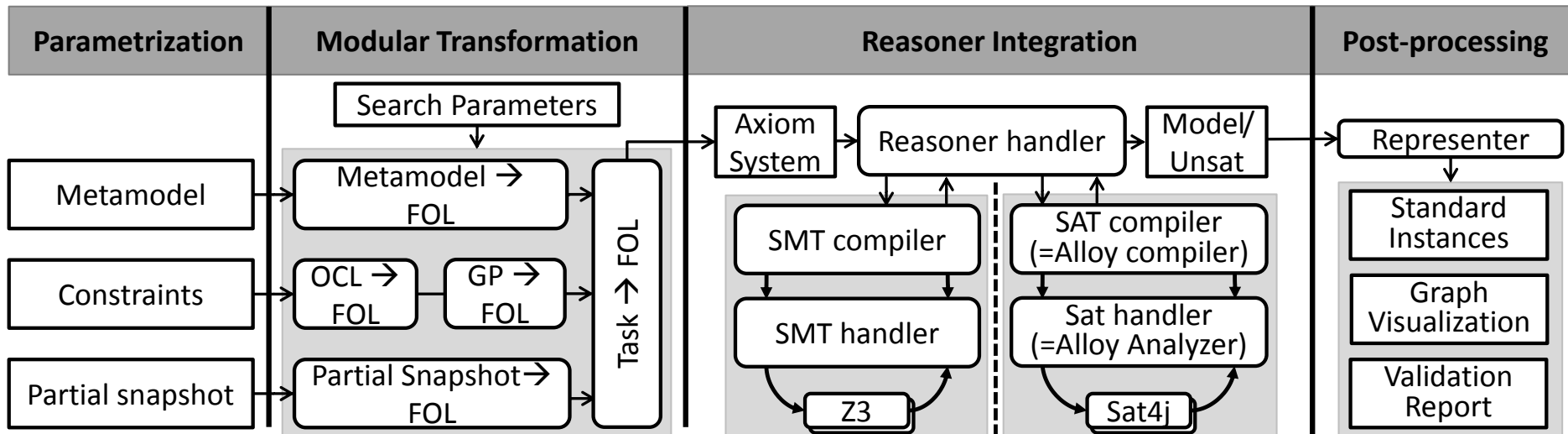
- Tesztorákulum:

Összehasonlítás $\in NP$, ha izomorfia Különben szemantikai ellenőrzés

- Eredmény helyességének ellenőrzése
- Helyesség egyértelműségének ellenőrzése

Megvalósítás

- Eclipse modellező eszközhöz integrált
- Szabványos modellek, lekérdezések
- Csak modellezési fogalmak → Tesztmérnök 😊
- Mérési kampányok, vizualizálás, jelentések



Kapcsolódó munkák

- Imperatív kódokra (**Erre nem hatékony**)

java: Korat <http://korat.sourceforge.net/>

C#: PEX <http://research.microsoft.com/en-us/projects/pex/>

- Hasonló megközelítések:

Sen, S., Baudry, B., Mottu, J.-M.: On combining multi-formalism knowledge to select models for model transformation testing. In: 1st International Conference on Software Testing, Verification, and Validation, pp. 328–337. IEEE (2008)

- Formula – Modellező és bizonyító nyelv

<http://research.microsoft.com/en-us/projects/formula/>

- Alloy Analyser – SAT megoldó

<http://alloy.mit.edu>

- Z3 – SMT megoldó

online: <http://rise4fun.com/Z3>, <http://rise4fun.com/Z3Opt>

Tapasztalatok, Eredmények

- Metamodell + Modell + OCL + IncQuery → Modell

Z3: SMT solver	Alloy: SAT solver
Korlátfüggetlen	Korlátos modellek
Bizonyítás	Példa/Ellenpélda
Attribútumok	Gráfstruktúra
20 elem	80 elem

- Változatos, értelmes tesztek → modell transzformációk tesztelése
- Gráfmintaillesztés (IncQuery) végrehajtása
- Tesztmérnök által paraméterezhető
- Esettanulmányok: EMBRAER, R3COP, CUDB
Jövőben: R5COP

Hivatkozások

1. Baudry, B., Dinh-trong, T., Mottu, J.-M., Simmonds, D., France, R., Ghosh, S., Fleurey, F., LeTraon, Y.: Model transformation testing challenges. In: ECMDA Workshop on Integration of Model Driven Development and Model Driven Testing (2006)
2. Goos, G.: Compiler verification and compiler architecture. Electron. Notes Theor. Comput. Sci. 65(2), 1 (2002). COCV'02, Compiler Optimization Meets Compiler Verification (Satellite Event of ETAPS 2002)