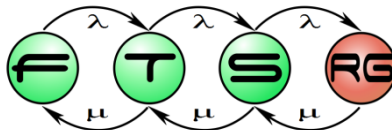


Konkrét szintaxis tervezés

Modellalapú szoftvertervezés

Ujhelyi Zoltán



Modellező nyelvek tervezése

■ Metamodellek részei

- Absztrakt szintaxis
 - Pl. EMF
- **Konkrét szintaxis**
 - **Szöveges vagy grafikus jelölés (pl. GMF, Graphiti, Xtext)**
- Jólformáltsági szabályok
 - Kiértékelés (pl. EMF-IncQuery)
- Dinamikus szemantika
 - Modelltranszformációk (pl. VIATRA2, ATL, ETL, ...)
- Leképezések más nyelvekre
 - Kódgenerálás (pl. Acceleo, Xtend)

Konkrét szintaxis tervezés

- Modell felhasználó által látható része
 - Megbízhatóság
 - Usability kérdések!
- Szerkesztő készítése
 - Hasonló probléma programozási nyelveknél
 - Integrált fejlesztőeszköz bővítése szükséges

- Eclipse Modeling Tools
 - Sokféle kapcsolódó technika
 - Mindegyik csak egy-egy aspektussal foglalkozik
 - Ma ilyenekről lesz szó
- Microsoft Visual Studio 2010 Visualization & Modeling SDK
 - Microsoft DSL modellező keretrendszer
 - Saját metamodellező mag
 - Grafikus editorokra kihegyezve

Grafikus szerkesztők

Grafikus modellezés

- Modell
 - Tipikusan gráf modell
 - Esetünkben EMF
- Alapötlet
 - Ábrázoljuk és szerkesszük gráf formában

Példa: Petri háló szerkesztő

r2init2r.vpml

Select
Marquee

New Place
New Transition
New Token
New OutArc
New InArc
Delete element
Delete Token

pn1

```
graph LR; p1((p1  
2)) --> t1[t1]; p1 --> t2[t2]; t1 --> p2((p2  
0)); t2 --> p3((p3  
0));
```

Outline

PetriNet model elements

- pn0
- pn1
 - p1
 - p1_t1
 - p1_t2
 - token0
 - token1
 - p2
 - p3
 - t1
 - t1_p2
 - t2
 - t2_p3

PetriNet diagrams

- Example Petri net [PetriNetDiagram]
- Example Petri net [PetriNetRoot]
 - pn1 [PetriNetFigure]
 - p1 [PlaceFigure]
 - token0 [TokenFigure]
 - token1 [TokenFigure]
 - p2 [PlaceFigure]
 - p3 [PlaceFigure]
 - t1 [TransitionFigure]
 - t2 [TransitionFigure]
 - p1_t1 [OutArcFigure]
 - p1_t2 [OutArcFigure]
 - t1_p2 [InArcFigure]
 - t2_p3 [InArcFigure]

Petri net State machine

Példa: Közösségi háló szerkesztő

The screenshot shows the Eclipse IDE interface with a social network editor. The main canvas displays a hierarchical structure: 'Foo Club' contains 'Bar Society', which contains 'Baz Community'. 'J. Random' is connected to 'Jane Doe', who is connected to 'John Doe'. 'John Doe' is also connected to 'Qux Fellowship'. The left sidebar shows the Project Explorer and Outline. The bottom right shows the Properties window for 'John Doe'.

Project Explorer:

- DSE
 - default.socialnetwork
 - Social Network
 - Acquaintance friendship
 - Acquaintance marriage
 - Community Foo Club
 - Community Qux Fellowship
 - Person J. Random
 - Person Jane Doe
 - Person John Doe

Outline:

- John Doe
- Jane Doe
- Foo Club
 - Bar Society
 - Baz Community
- Qux Fellowship
- J. Random

Properties Window:

Property	Value
Name	John Doe
Sex	male
X	677
Y	240

Megvalósítás

- Megjelenítés
 - Rajzvászonra rajzolva
 - Vektorgrafikus könyvtár használata (GEF/Draw2d)
- Modellezés/szerkesztés
 - EMF Edit technikákra építve
 - EMF modellmódosító parancsok
 - EMF tranzakciók
 - Minden módosítás tranzakció
 - Visszavonható
 - Megismételhető

Megvalósítás 2.

■ Nézeti modellek

- Megjelenítésspecifikus információk eltárolása
 - Koordináta
 - Méret
 - ...
- Gyakran külön fájlban
 - Miért?

Technológia 1. - GEF

- Graphical Editing Framework (GEF)
 - “Alacsony szintű” eszköz
 - Nem EMF-specifikus
- Model-View-Controller megközelítés
- Általános szerkesztési támogatás
 - Beleértve visszavonás kezelését
- Gyakorlatilag mindent kézzel kell lekódolni

Technológia 2. – GMF

- Graphical Modeling Framework
- EMF-alapú modellezés
- GEF alapú
- Domain és nézeti modellek elkülönítése
 - A két modellt a GMF tartja szinkronban
- Viszonylag idős technológia
 - Széles körben elterjedt
 - Ugyanakkor elég komplex

- Modell-alapú fejlesztőkörnyezet
 - Alapszintű szerkesztő összeállítható modellekből
 - Összetett elemek könyvtára
 - Szerkesztőeszközök modellje
 - Kapcsolómodell
 - Szerkesztő generálható
 - Utólagos testreszabás ilyenkor nehézkes lehet
 - Lehetséges ténylegesen kézzel is csinálni

Technológia 3. - Graphiti

- Friss grafikus felület szerkesztő technika
 - EMF és GEF alapú
 - DE: nem GMF - más megközelítés
 - Egyszerűsített fedő API
 - Kézi kódolás
 - Ötlet
 - Szerkesztők legyenek hasonlóak
 - Kinézet
 - Viselkedés

Technológia 3. - Graphiti

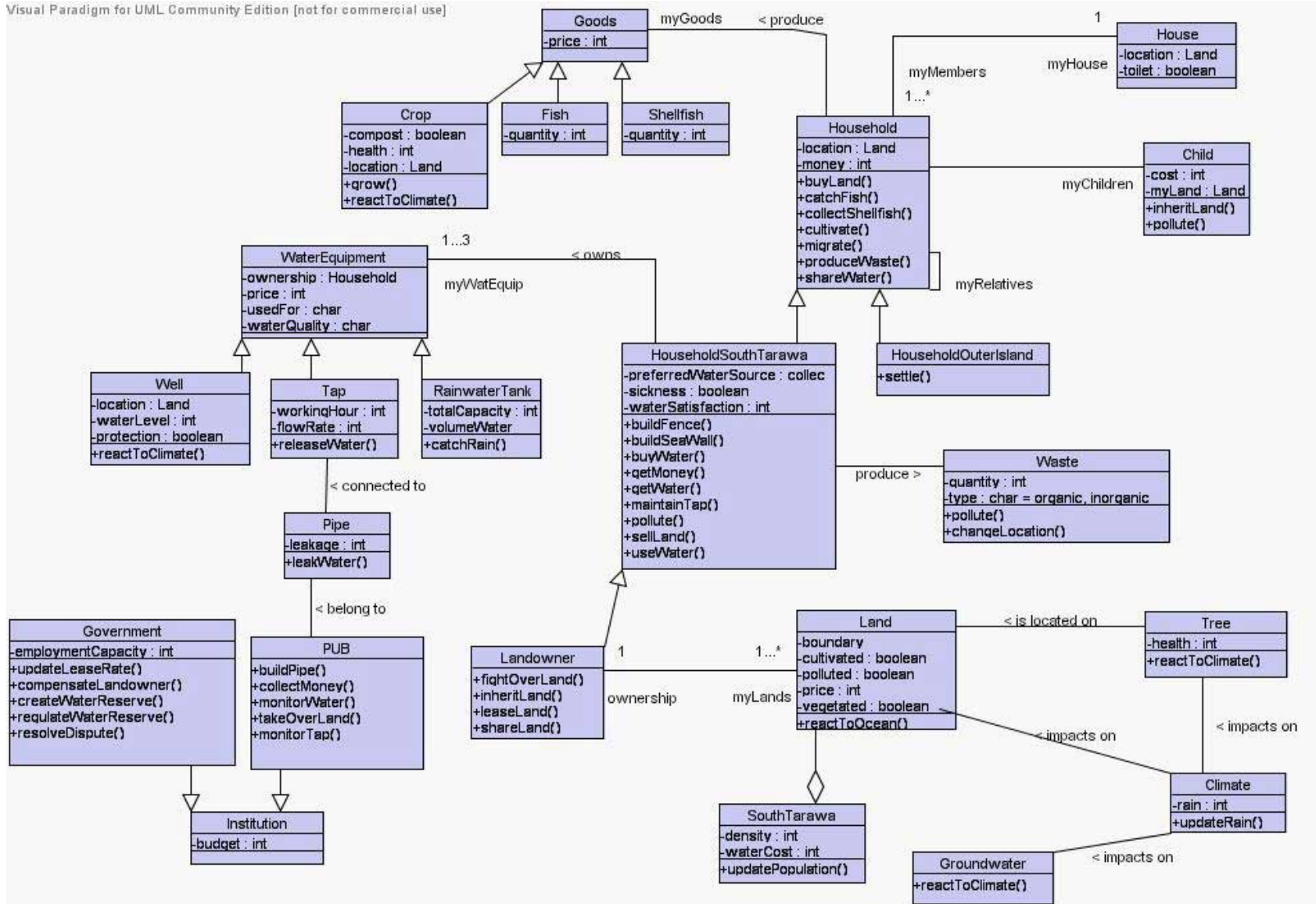
- Hivatalos fejlesztési módszer
 - Java kódolás
 - Ismétlődő kódrészletek
- Spray projekt
 - Szöveges modellező környezet
 - Graphiti kódot állít elő

Problémák grafikus szerkesztéssel

- Nehézkes szerkesztés
 - “Középre beszúráshoz” át kell rendezni
- Átláthatóság, olvashatóság
 - Biztosítható
 - De szerkesztéskor is oda kell erre figyelni
- Nagy (20+ elem...) modellek kezelése
 - Ez nem igazán működik

Nagyobb modell

Visual Paradigm for UML Community Edition [not for commercial use]



Szöveges modellezési nyelvek

Szöveges domain-specifikus nyelvek

- Ötlet
 - Modelleket szövegfájlként írjuk le
- Szöveges fejlesztőeszközök
 - Hosszú történet (30+ év)
 - Kiforrott elméleti háttér
 - Kiforrott eszköztámogatás

Szöveges nyelvek feldolgozása

- Természetes nyelvek
- XML feldolgozás
- Nyelvtan alapú feldolgozás
 - Kézzel kódolt elemző
 - Bonyolult
 - Reguláris kifejezések
 - Generált elemző (formális nyelvek)

Természetes nyelvek feldolgozása

- “Nehéz” probléma
 - Kontextus információk
 - Nem (jól) definiált szemantika



XML feldolgozás

- XML: szabványos szöveges szintaxis
 - Sémák definiálhatóak - metamodel
- Jó tooling
 - Többféle beolvasási technika
 - Validáció
- Nehéz írni
 - Lezáró címkék
 - Entitások (pl. `>`; `„`)
- Nehéz olvasni
 - Szigorú fastruktúra
 - Hivatkozásfeloldás nem automatikus

Reguláris kifejezések

- Mintaillesztés karakterláncokban
 - Jó támogatás
 - Legtöbb programozási nyelven
 - Nagyjából közös szintaxis
 - Illeszkedés számítás/visszaadás
- Felhasználható szöveg elemzésére?

RegExp: Email validáció

[illegible]

This regular expression will only validate addresses that have had any comments stripped and replaced with whitespace

További probléma

- Kimenet egyetlen logikai változó
 - Illeszkedik-e vagy sem
- Mi hiányzik?

További probléma

- Kimenet egyetlen logikai változó
 - Illeszkedik-e vagy sem
- Mi hiányzik?

Hibajelzés!

Formális nyelv

■ Fogalmak

- Nyelv: lehetséges mondatok halmaza
- Mondat: szimbólumok sorozata
- Ábécé: lehetséges szimbólumok halmaza

Példa: Természetes számok

Példa: Természetes számok

- Ábécé: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Egyjegyű számok nyelve:
 - «0», «1», «2», «3», «4», «5», «6», «7», «8», «9»
- 15-nél kisebb páros számok nyelve:
 - «0», «2», «4», «6», «8», «10», «12», «14»
- Pozitív páros számok nyelve:
 - «0», «2», «4», «6», «8», «10», «12», «14», «16», «18», «20», «22», «24», «26», «28», ...

Példa: Természetes számok

- Ábécé: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Egyjegyű számok nyelve:
 - «0», «1», «2», «3», «4», «5», «6», «7», «8», «9»
- 15-nél kisebb páros számok nyelve:
 - «0», «2», «4», «6», «8», «10», «12», «14»
- Pozitív páros számok nyelve:
 - «0», «2», «4», «6», «8», «10», «12», «14», «16», «18», «20», «22», «24», «26», «28», ...

Példa: Természetes számok

- Ábécé: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Egyjegyű számok nyelve:
 - «0», «1», «2», «3», «4», «5», «6», «7», «8», «9»
- 15-nél kisebb páros számok nyelve:
 - «0», «2», «4», «6», «8», «10», «12», «14»
- Pozitív páros számok nyelve:
 - «0», «2», «4», «6», «8», «10», «12», «14», «16», «18», «20», «22», «24», «26», «28», ...

Példa: Természetes számok

- Ábécé: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Egyjegyű számok nyelve:
 - «0», «1», «2», «3», «4», «5», «6», «7», «8», «9»
- 15-nél kisebb páros számok nyelve:
 - «0», «2», «4», «6», «8», «10», «12», «14»
- Pozitív páros számok nyelve:
 - «0», «2», «4», «6», «8», «10», «12», «14», «16», «18», «20», «22», «24», «26», «28», ...

Nyelvek megadása

- Elemek felsorolása
 - Csak véges elemszám esetén működik
 - Ld. páros számok nyelve
- Véges algoritmus felhasználása
 - Lépések, melynek a kimenetei a nyelv mondatai

Példa: Nevek felsorolása

- Nevek felsorolása:
 - Neveket vesszővel válasszuk el;
 - Kivéve az utolsót, ott és legyen közben
 - Pl.: Zoltán, István és Dániel
 - Ismétlődés megengedett
 - Pl.: Zoltán, Zoltán és Dániel

Példa: Nevek felsorolása

Példa: Nevek felsorolása

■ Algoritmus

- ① Dániel, István, Zoltán nevek
- ② Egy név érvényes mondat
- ③ Egy mondat, amelyet vessző és egy név követ, értelmes mondat
- ④ Ha a befejezés előtt a mondat “, «név»” alakú, cseréljük ki “ és «név»” formájúra

Példa: Nevek felsorolása

■ Algoritmus

- ① Dániel, István, Zoltán nevek
- ② Egy név érvényes mondat
- ③ Egy mondat, amelyet vessző és egy név követ, értelmes mondat
- ④ Ha a befejezés előtt a mondat “, «név»” alakú, cseréljük ki “ és «név»” formájúra

«név» nem fog szerepelni a szövegben!

Terminális és nem-terminális szimbólumok

- Kétféle szimbólum
 - Megjelenhet a szövegben (ábécéhez tartozik)
 - Terminális szimbólum
 - Nem jelenhet meg a szövegben
 - Nem-terminális szimbólum
- Nem-terminálisok
 - Cserélhetőek más szimbólumokkal ($\rightarrow$)
 - Algoritmus vége
 - Csak terminális szimbólum van
 - Nem cserélhetünk

Módosított algoritmus

- Ábécé
 - Dániel, István, Zoltán, és, “,” , sorvége
- Nem-terminális szimbólumok
 - «Név», «Mondat»

- ① Dániel $\rightarrow$ «Név»; István $\rightarrow$ «Név»;
Zoltán $\rightarrow$ «Név»
- ② «Mondat» $\rightarrow$ «Név»
- ③ «Mondat» $\rightarrow$ «Mondat», «Név»
- ④ «Mondat» végén , «Név» $\rightarrow$ és «Név»
- ⑤ Kiindulás: «Mondat»

Módosított algoritmus

- Ábécé
 - Dániel, István, Zoltán, és, “,” , sorvége
- Nem-terminális szimbólumok
 - «Név», «Mondat»

① Dániel → «Név»; István → «Név»;
Zoltán → «Név»

② «Mondat» → «Név»

③ «Mondat» → «Mondat», «Név»

④ «Mondat» végén , «Név» → és «Név»

⑤ Kiindulás: «Mondat»

Bonyolult
elemzési szabály

Példa: CF nyelvtan a nevekhez

- Ábécé
 - Dániel, István, Zoltán, és, “,” , sorvége
- Nem-terminális szimbólumok
 - «Név», «Mondat», «Lista»

«Név» ::= Zoltán | István | Dániel

«Mondat» ::= «Név» | «Lista» és «Név»

«Lista» ::= «Lista», «Név» | «Név»

Nyelvtanok alkalmazása

- Szabályok alapján behelyettesítés
- Indeterminisztikus választás
 - Elemzési algoritmusok leköthetik!
- Cél:
 - Szabályok sorozatával előállítani a bemenetet

Példa: Levezetés

«Név» ::= Zoltán | István | Dániel

«Mondat» ::= «Név» | «Lista» és «Név»

«Lista» ::= «Lista», «Név» | «Név»

Mondatszerű forma	Felhasznált szabály
«Mondat»	

Példa: Levezetés

«Név» ::= Zoltán | István | Dániel

«Mondat» ::= «Név» | «Lista» és «Név»

«Lista» ::= «Lista», «Név» | «Név»

Mondatszerű forma	Felhasznált szabály
«Mondat»	
«Lista» és «Név»	«Mondat» -> «Lista» és «Név»

Példa: Levezetés

«Név» ::= Zoltán | István | Dániel

«Mondat» ::= «Név» | «Lista» és «Név»

«Lista» ::= «Lista», «Név» | «Név»

Mondatszerű forma	Felhasznált szabály
«Mondat»	
«Lista» és «Név»	«Mondat» -> «Lista» és «Név»
«Lista», «Név» és «Név»	«Lista» -> «Lista», «Név»

Példa: Levezetés

«Név» ::= Zoltán | István | Dániel

«Mondat» ::= «Név» | «Lista» és «Név»

«Lista» ::= «Lista», «Név» | «Név»

Mondatszerű forma	Felhasznált szabály
«Mondat»	
«Lista» és «Név»	«Mondat» -> «Lista» és «Név»
«Lista», «Név» és «Név»	«Lista» -> «Lista», «Név»
«Név», «Név» és «Név»	«Lista» -> «Név»

Példa: Levezetés

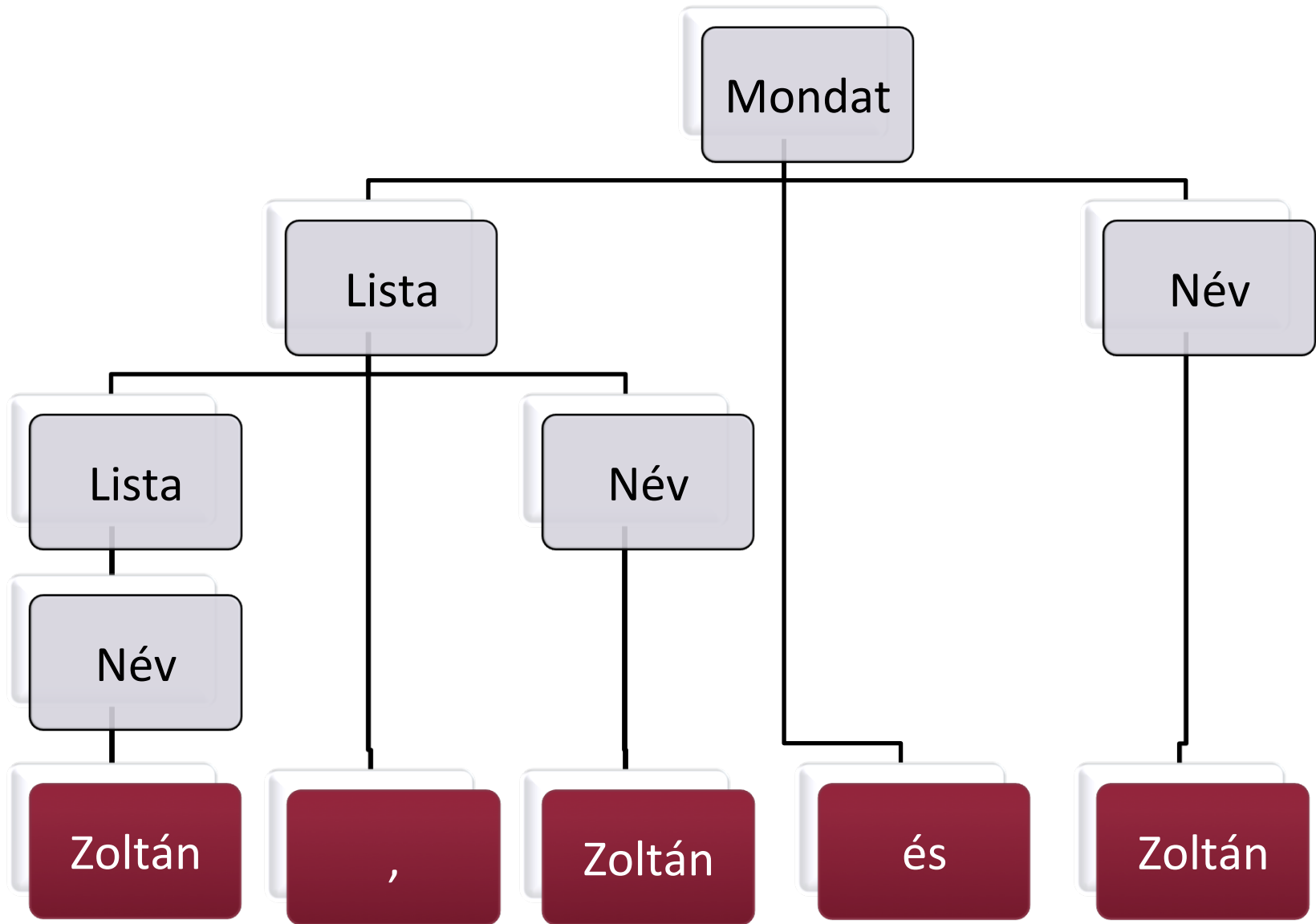
«Név» ::= Zoltán | István | Dániel

«Mondat» ::= «Név» | «Lista» és «Név»

«Lista» ::= «Lista», «Név» | «Név»

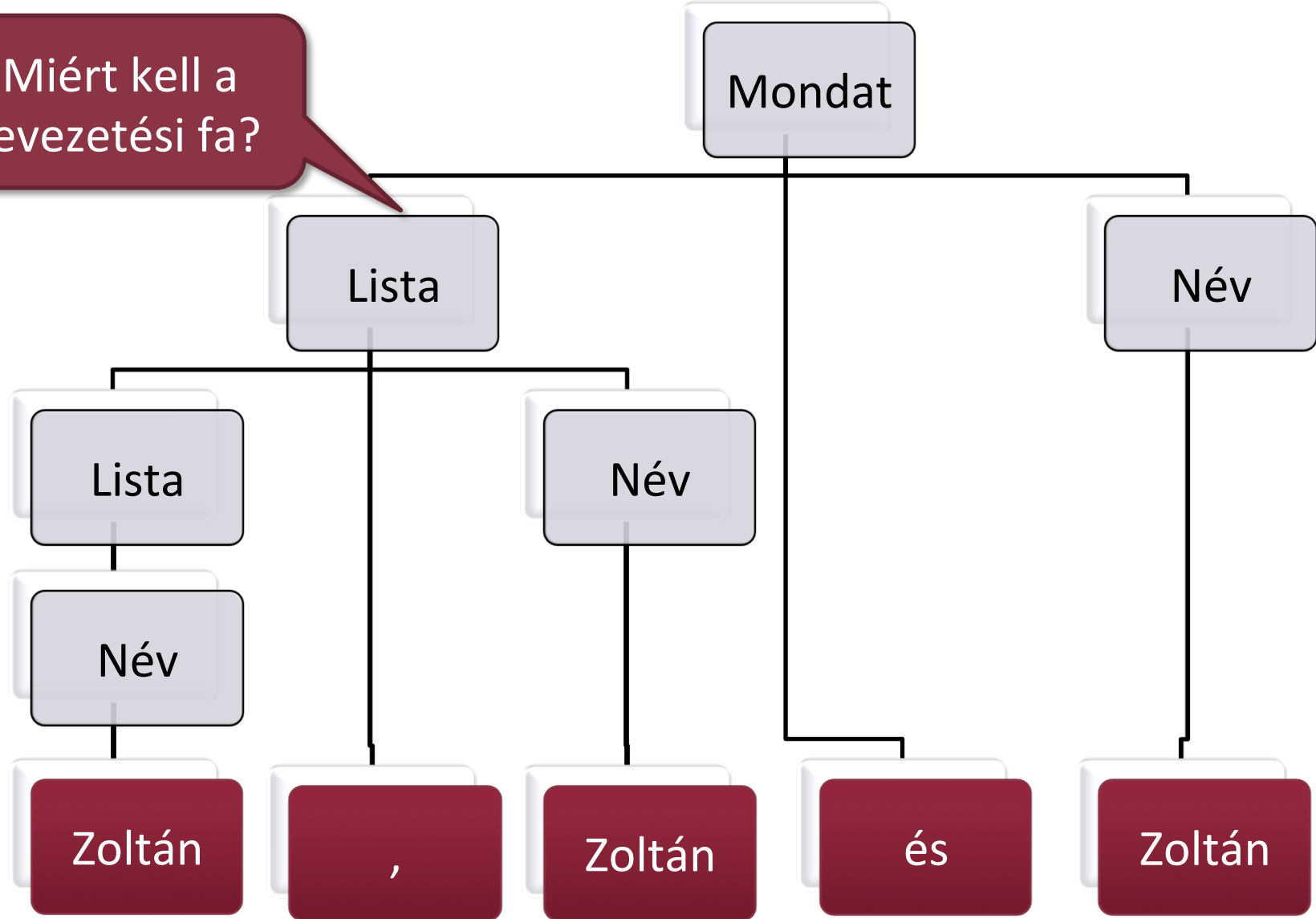
Mondatszerű forma	Felhasznált szabály
«Mondat»	
«Lista» és «Név»	«Mondat» -> «Lista» és «Név»
«Lista», «Név» és «Név»	«Lista» -> «Lista», «Név»
«Név», «Név» és «Név»	«Lista» -> «Név»
Zoltán, Zoltán és Zoltán	

Levezetési fa

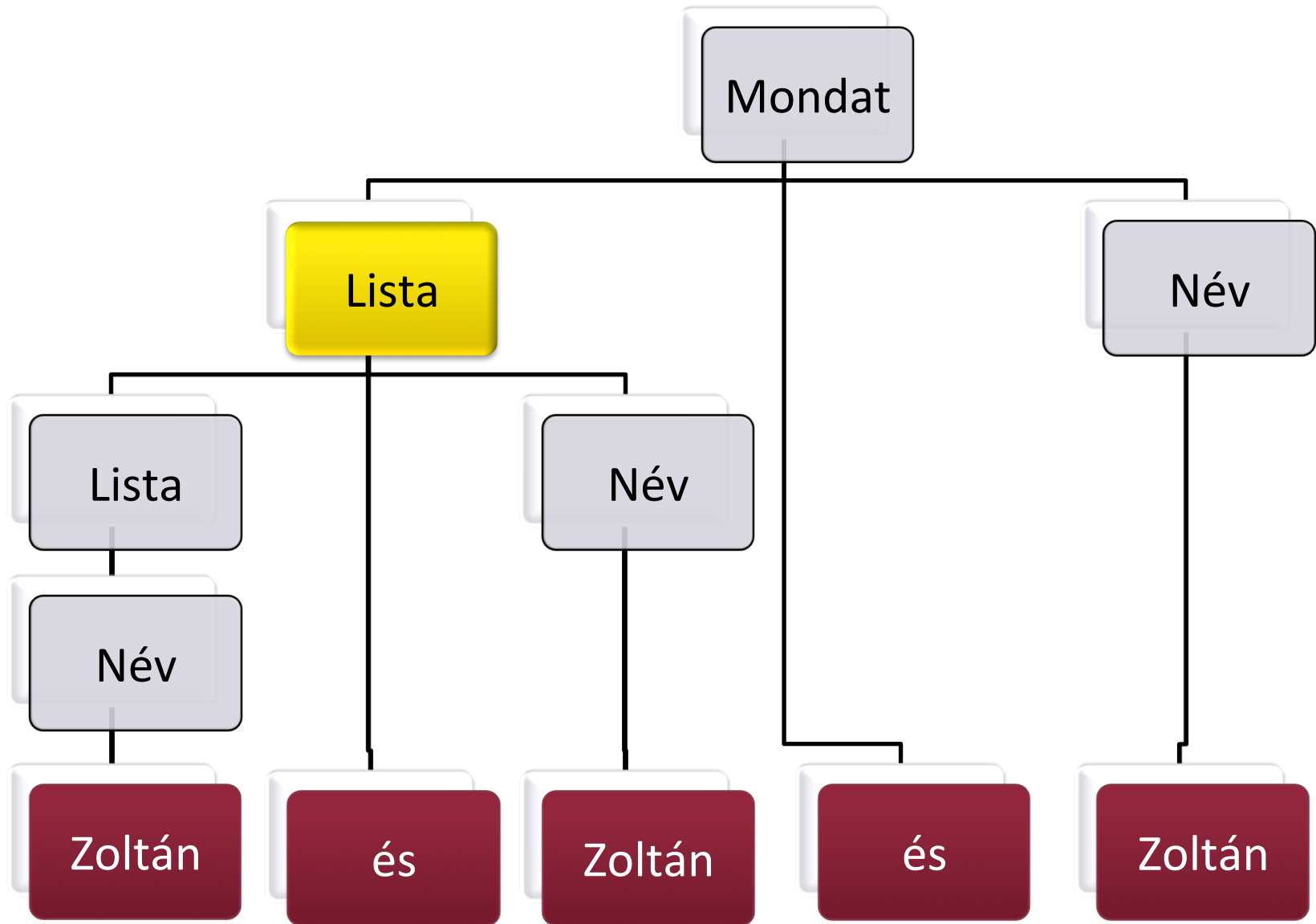


Levezetési fa

Miért kell a levezetési fa?



Hibajelzés



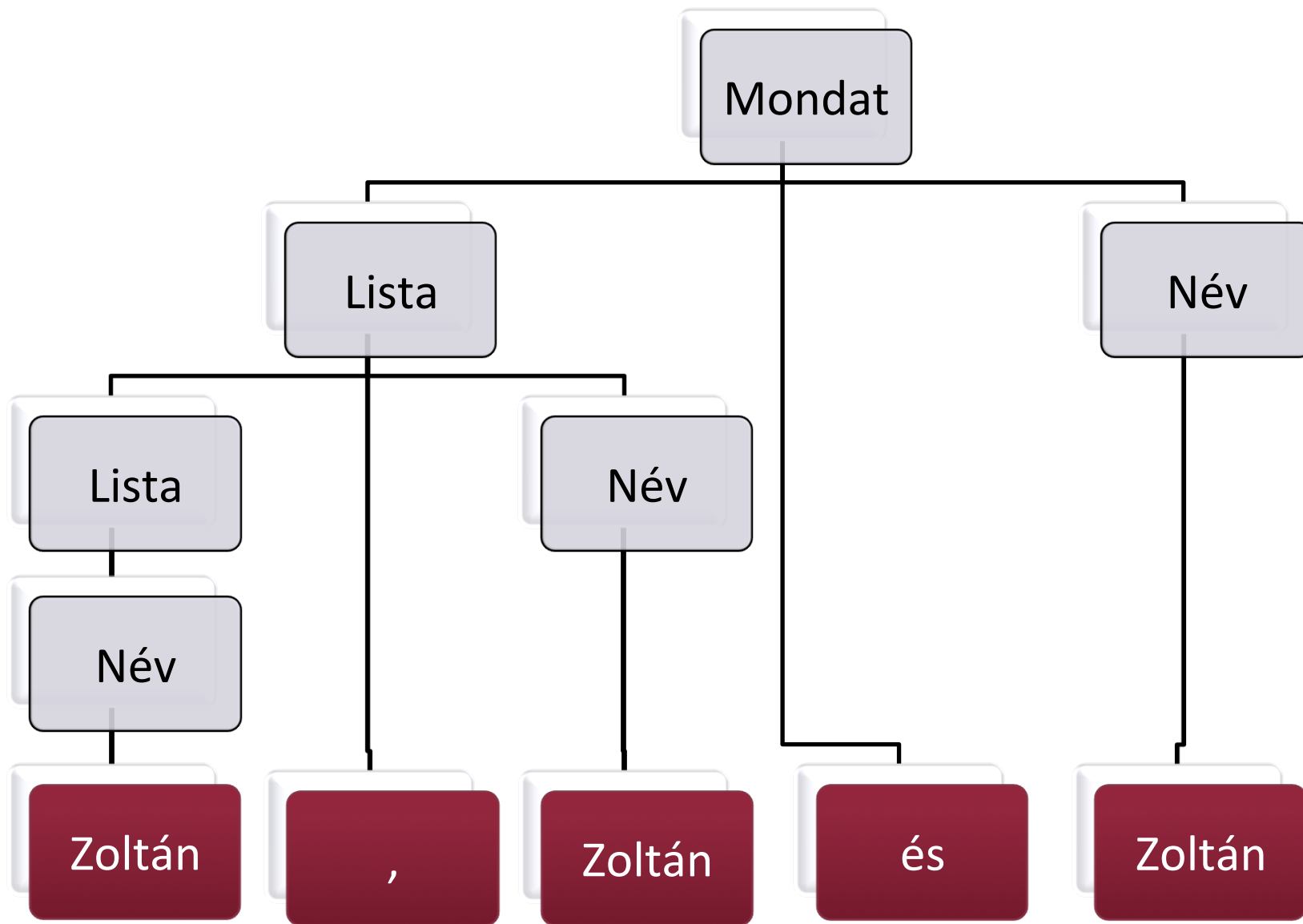
Áttekintő

- Eddig
 - Nyelvtan
 - Elemző
- Mi hiányzik még?

Áttekintő

- Eddig
 - Nyelvtan
 - Elemző
- Mi hiányzik még?
 - Bemeneti karakterfolyam kezelés
 - Változók bevezetése
 - Magasabb szintű elemzések

Bemenet



Bemenet

Tényleges bemenet:

'Z' 'o' 'l' 't' 'á' 'n' , 'Z'
'o' 'l' 't' 'á' 'n' 'é' 's'
'Z' 'o' 'l' 't' 'á' 'n'

Zoltán

'

Zoltán

és

Zoltán

Bemenet kezelése

Bemenet kezelése

- Bemenet:
 - Karakterfolyam
- Elemző bemenete
 - Magasabb szintű tokenek
 - «Név», ‘,’ vagy “és”
 - Kettő közötti kapcsolat: lexer
- Miért hasznos ez az indirekció?
 - Hibakezelés
 - Teljesítmény
 - Probléma dekompozíció

Lexer

- Cél:
 - Karakterfolyam tokenizálás
 - Hasonlít elemzők problémájához
 - De egyszerűbb – tipikusan reguláris kifejezés
 - Csak szavak/tokenek azonosítása
 - (Kommentek elhagyása)
 - Egyszerűsíti az elemző dolgát

Változók bevezetése

```
a=3;
```

```
System.out.println(a);
```

■ Változó

- Futási időben
 - Érték behelyettesítés
- Szerkesztési/elemzési időben
 - Hivatkozás
 - AST más részére

Változó kezelés

- Változó hivatkozás (*variable reference*)
 - Egy változó felhasználása
- Változó definíció (*variable definition*)
 - Egy változó megadása
- Egyedi név (jólformáltsági kényszer)
 - Változó definícióknak azonosíthatónak kell lennie
 - Külön fázis elemzés után

Változó kezelés

- Elemző biztosítja
 - Érvényes változónév-e (szintaktikailag)
- Feloldás ellenőrzi
 - Definíció létezése
 - Hatókör probléma (*scope*)
 - Definíció egyedisége

Hatókör probléma

```
private int value;
```

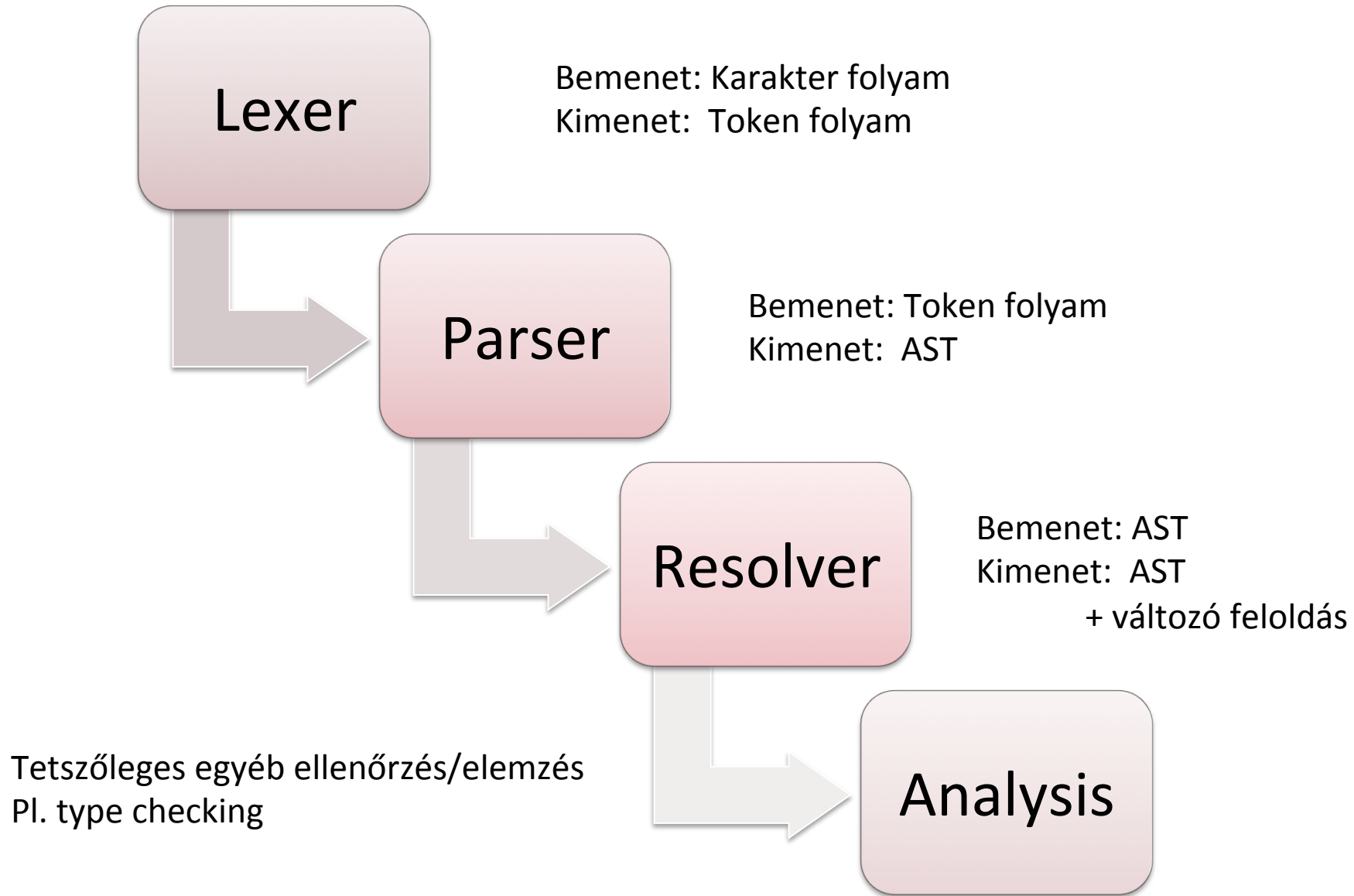
```
public void setValue(int value) {  
    this.value = value;  
}
```

Mire hivatkozik a
'value' referencia?

Hatókör probléma

- Megoldás
 - Feloldó szintjén definiálандó
- Lehetséges megközelítések
 - Legspecifikusabb előfordulás
 - Hibajelzés konfliktus esetén
 - Minősített hivatkozások (*qualified name*)
 - ...

Elemzés menete



- IDE Meta-tooling platform
 - Cél:
 - nyelvi támogatás fejlesztése
 - Parser-generátor független technológia
 - Sokat kell kódolni
 - Nagyon kevésbé fejlesztik

- Szerkesztő származtatás
 - EMF modell alapján
 - Több stílusú nyelvtan származtatható
 - Utána módosítható még
- Syntax Zoo
 - Sokféle mintanyelvtan
 - Beleértve Java nyelv is!

Technológiák 3. – Xtext

- Szerkesztő és EMF modell generálás
 - Nyelvtan alapján
 - Opció: EMF metamodel itt is lehet bemenet
- Jól testreszabható
- Új fejlesztési irány:
 - JVM felett futó DSL-ek készítése

Szöveges vagy grafikus nyelvek?

Összehasonlítás

Szöveges nyelvek	Grafikus nyelvek
Egyszerű, kényelmes szerkesztés	Körülményesebb szerkesztés
Kapcsolatok csak string referenciaként	Kapcsolatok vizuális megjelenítése
Inkonzisztens modellek lehetségesek szerkesztés közben	Mindig szintaktikusan helyes modellek
Szövegformázás	Gráf rajzoló eljárások

EMF technológiákkal mindkettő fedhető

Szerkesztés menete



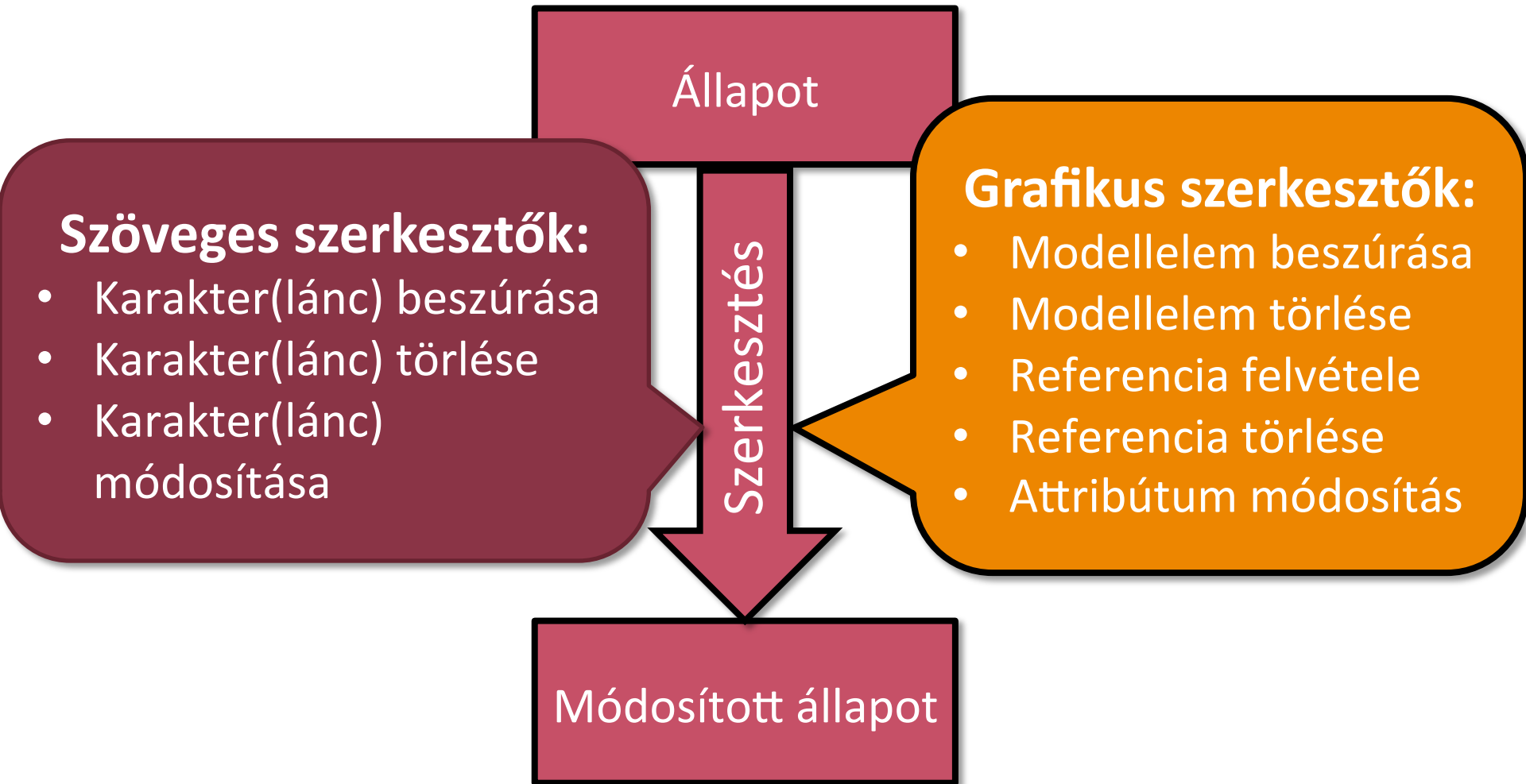
Szerkesztés menete

Szöveges szerkesztők:

- Karakter(lánc) beszúrása
- Karakter(lánc) törlése
- Karakter(lánc) módosítása



Szerkesztés menete



- Szöveges vagy grafikus nyelv?
 - Mindkettőt 😊
 - Hol az egyik, hol a másik jobb
 - Ökölszabályok
 - Viselkedésleírás inkább szöveges
 - Struktúraleírás inkább grafikus
 - Egyszerű nyelvek esetén
 - Form alapú szerkesztés

Xtext és GMF szerkesztő

The image displays the Xtext and GMF editor interface for a social network model. The left pane shows the Xtext source code, and the right pane shows the corresponding GMF diagram.

test.socialnetwork

```
SocialNetwork {  
  Person Ujhelyi {  
    male  
    memberships BME, VVEC  
  }  
  Person Horvath {  
    male  
    memberships FTSRG  
  }  
  Community BME {  
    Community FTSRG {  
      Community test  
    }  
  }  
  Person Test {  
    female  
    memberships test  
  }  
  Community VVEC  
  Person Proba {  
    male  
  }  
  Community Pr2  
  Person valaki {  
    male  
  }  
  
  Ujhelyi is friend of Horvath  
  Test is married to Ujhelyi  
}
```

test.socialnetwork_diagram

The diagram visualizes the model structure. It includes nodes for individuals (Test, Proba, Ujhelyi, Horvath, valaki) and communities (BME, VVEC). The BME community contains the FTSRG community, which in turn contains the test community. Relationships are indicated by lines connecting the nodes.

Palette

- Community
- Person
- Acquaintance
- Membership

Grafikus nézet származtatás

- Xtext Generic Viewer komponens
 - Xtext fejlesztők projektje
 - Független a fő fejlesztéstől
 - +1 nyelv
 - Grafikus megjelenítés definiálása

- Integrált környezet
 - Magas szintű szerkesztési támogatás
 - Áttekintő nézet
 - Dokumentáció megjelenítés
 - Sablonok
 - Content assist
 - Hibajelzések/automatikus javítások
 - További elvárt szolgáltatások
 - Projekt struktúra
 - Kódgenerátor

Összefoglalás

Konkrét szintaxis tervezés

- Sokféle technika
 - Szöveges és/vagy grafikus jelölés
- Sok munkát és tesztelést igényel
 - Felhasználók látják/használják!
 - Legyen kényelmes nekik