Aspektovo orientovaná syntaktická analýza

Diplomová práca

Autor: Marcel Tóth
Vedúci: Ján Kollár

Technická univerzita v Košiciach, November 2004

Obsah prezentácie

- Obsah prezentácie
- Ciele práce
- Analýza stavu AOP
- Pojmy AOP
- Popis riešenia (jazyk, tkáč, prekladač, interpreter, integrované prostredie)
- Námety pre ďalší rozvoj
- Záver

Ciele práce

- Analýza možností syntaktickej analýzy s AOP prístupom
- Definícia bodov spájania a vytvorenie modulu spájania (tkáč)
- Implementácia interpretátora jednoduchého jazyka
- Využiteľnosť produktu diplomovej práce pre ilustráciu možností AOP

Analýza stavu - potreba AOP

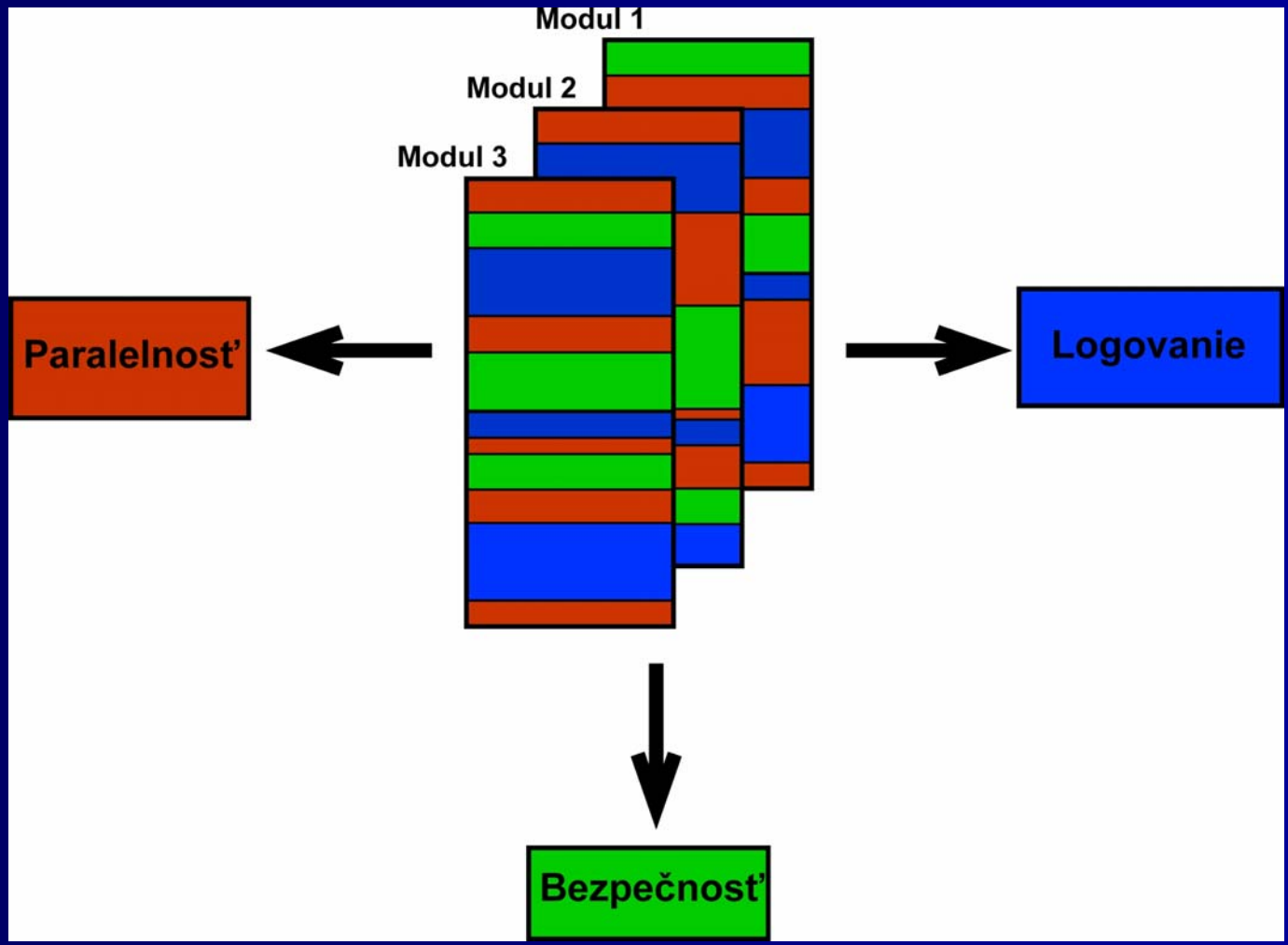
Nedostatky existujúceho spôsobu návrhu a implementácie SW systémov

- Prekrížené súvislosti
- Zmotaný a roztrúsený kód (nemodulárny)
- Sťažená udržiavateľnosť a rozšíriteľnosť rozsiahlych systémov

Riešenie:

- Modulárne vyjadrenie prekrížených súvislostí
- Odstránenie zmotaného a roztrúseného kódu (abstrakcia z iného pohľadu ako OOP)

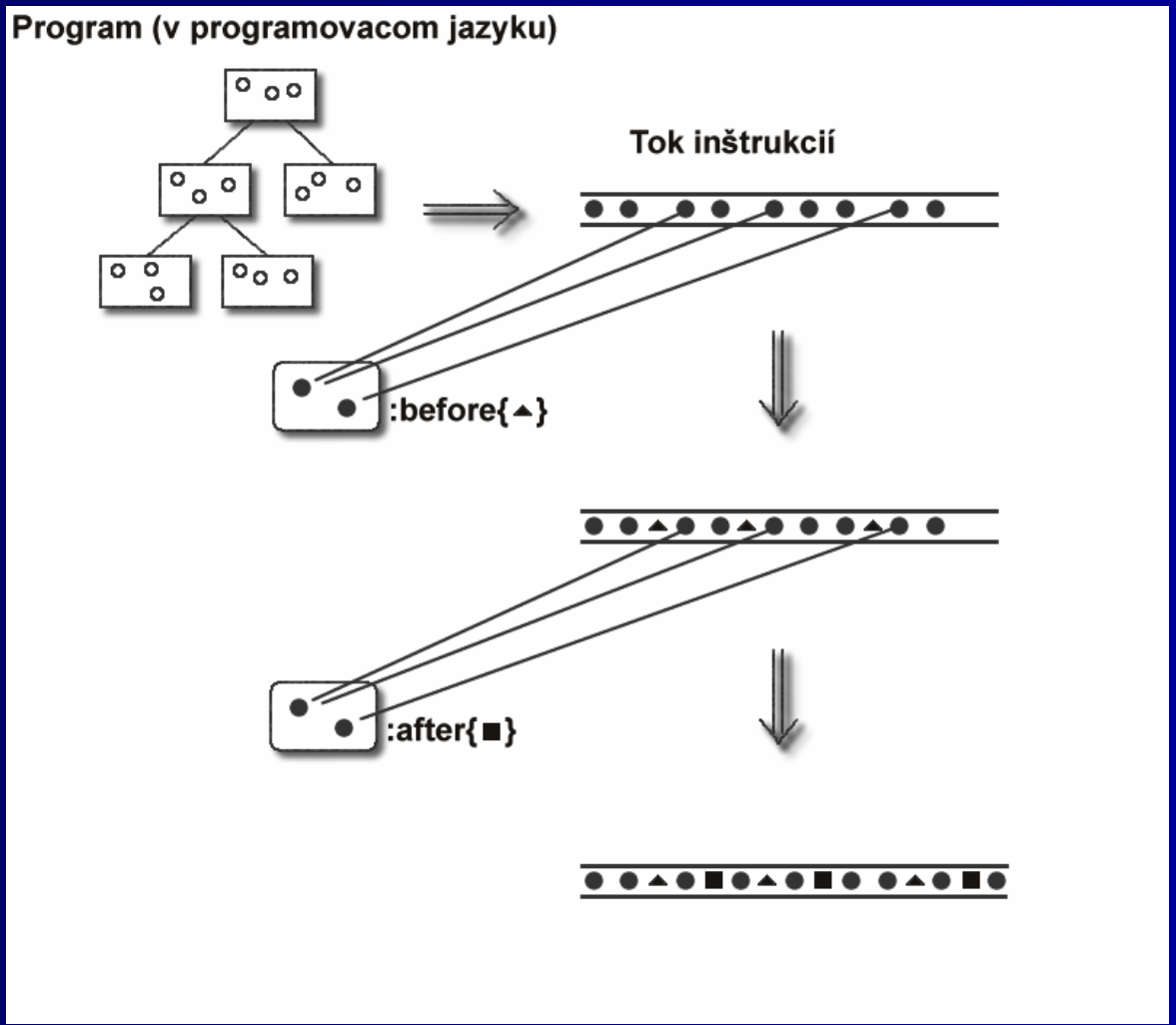
Analýza stavu – ilustrácia prekrížených súvislostí



Základné pojmy AOP

- Aspekt – Prostriedok modulárneho vyjadrenia pretínajúcich aktivít
- Bod spojenia (join point) – dobre definované miesto v kóde programu komponentu, v ktorom je možné pripojenie kódu rady aspektu
- Rada (advice) – definuje kód pripájaný v označených bodoch spojenia
- Označovač (pointcut) – vyberá (označuje) body spojenia, ku ktorým je pripojený kód rady
- Tkáč – program vykonávajúci spojenie (zotkanie) komponentov a aspektov do výsledného kódu
- Komponent – zdrojový kód, do ktorého sú votkané aspekty

Princíp AOP


Program (v programovacom jazyku)
Tok inštrukcií
:before{▲}
:after{■}

Pojmy AOP

- Preddefinované funkcie označovačov, napr.:
  - set
  - get
  - call
- Typ rady
  - Pred (before)
  - Po (after)
  - Namiesto (around)
- Tkanie (zotkávanie)
  - Statické
  - Dynamické

Popis jazyka Kresl

- Úlohou je ilustrovať princípy AOP
- Funkciou je vykresľovanie obrazcov
- Používa štandardné jazykové konštrukcie (deklarácie, vetvenia, cykly, funkcie, priradenia, ...)
- Obsahuje kľúčové slová pre prácu s interpretačnou plochou (pen, oval, quad, line, text, ...)

Aspetové rozšírenie jazyka Kresl

Obsahuje jazykové konštrukcie pre:

- Aspekty (aspect)
- Rady (advice)
- Kód rád (v jazyku Kresl)
- Typy rád (before, after, around)
- Označovače
- Preddefinované funkcie označovačov (set, get, call)

Príklad definície aspektu

```
aspect Obmedzenie
   {
   advice set('int x'):after
        {
        if(x<0||x>10)
                x=5;
        }
   }
```

ekvivalent triedy

ekvivalent metódy

Popis modulu “tkáč”

- Zotkávanie kódu aspektov a kódu komponentov
- Vstup: súbor v jazyku Kresl (komponenty .ck), súbor v aspektovom rozšírení jazyka Kresl (aspekty .ak)
- Výstup: zotkaný súbor v jazyku Kresl

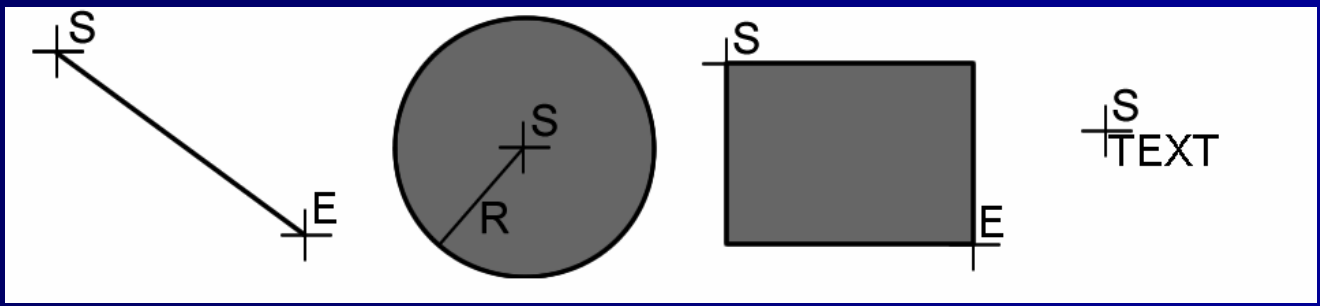
Popis modulu “prekladač”

- Vstup: v jazyku Kresl (.k)
- Výstup: v jazyku virtuálnych inštrukcií (.asm)
- Generovaný generátorom kompilátorov Lex-Yacc

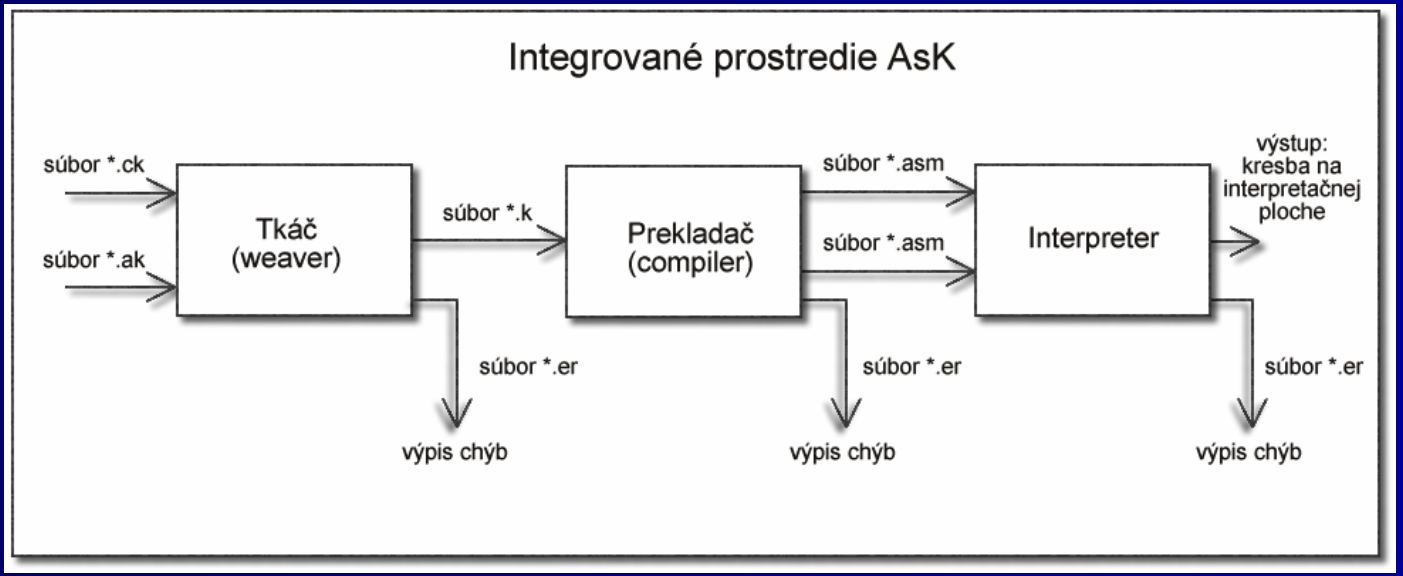
Interpreter

- Vykonáva virtuálne inštrukcie z preloženého súboru
- Zobrazuje výsledok interpretácie na interpretačnej ploche

Tvary zobraziteľné na interpretačnej ploche:



Integrované prostredie



- Grafické prostredie pre prácu s modulmi tkáča a prekladača
- Zjednodušuje spoluprácu modulov a prenášanie súborov medzi nimi

Námety pre ďalší rozvoj (problémy AOP)

- Odstránenie alebo zmenšenie viazanosti aspektov na text zdrojového kódu
- Zlepšenie rýchlosti a efektívnosti techník dynamického zotkávania
- Definovanie väčšieho počtu preddefinovaných funkcií označovačov, prípadne iný prístup

Záver

- Vytvorený systém tkáč – prekladač – inerpreter úspešne demonštruje základné princípy AOP na jednoduchom jazyku Kresl
- Pre AOP už exisujú použiteľné jazyky (AspectJ, AspectC++), napriek tomu je potrebný výskum pre odstránenie uvedených problémov.
- AOP je perspektívnym riešením niektorých programátorských problémov (modularizácia prekrížených súvislostí, zlepšenie čitateľnosti kódu)

Ďakujem za pozornosť