Multiparadigmový návrh s modelovaním vlastností

Valentino Vranić

www.fiit.stuba.sk/~vranic, vranic@fiit.stuba.sk

Ústav informatiky a softvérového inžinierstva

Fakulta informatiky a informačných technológií

Slovenská technická univerzita

Obhajoba dizertačnej práce — 3. september 2004

Úvod

- Proces vývoja softvéru: mapovanie domény aplikácie (problému) na doménu riešenia
- Paradigma vývoja softvéru: ako vyjadriť koncepty domény aplikácie v zmysle konceptov domény riešenia
- Koncepty domény riešenia zodpovedajú mechanizmom programovacích jazykov
- Výber zodpovedajúcej paradigmy je dôležitá záležitosť
- Jednotlivé koncepty domény riešenia (napr. trieda v Jave) možno považovať za paradigmy

Prehľad prezentácie

- Ciele dizertačnej práce
- Prínosy dizertačnej práce
- Pojem paradigmy
- Multiparadigmový návrh s modelovaním vlastností (MPD_{FM})
- Modelovanie paradigiem v MPD_{FM}
- Transformačná analýza v MPD_{FM}
- Vyhodnotenie MPD_{FM}
- Zhodnotenie a ďalšia práca

Ciele dizertačnej práce

- Hlavný cieľ: vylepšenie multiparadigmového návrhu použitím modelovania vlastností
- Analýza pojmu paradigmy
- Vhodná reprezentácia paradigmy
  - Prispôsobenie modelovania vlastností
- Vyhodnotenie metódy
  - Vytvorenie modelu paradigiem jazyka AspectJ
  - Transformačná analýza domény modelovania vlastností pomocou modelu paradigiem jazyka AspectJ

Prínosy

- Metóda multiparadigmového návrhu s modelovaním vlastností (MPD_{FM})
- MPD_{FM} pre AspectJ
- Vylepšenia modelovania vlastností
- Model vlastností domény modelovania vlastností
- Aplikácia MPD_{FM} pre AspectJ na túto doménu

Pojem paradigmy

- Pôvodný význam: príklad alebo vzor
- Vedecká paradigma[a]
- Paradigmy programovania a vývoja softvéru[b]
  - Podstata určitého procesu vývoja softvéru
  - “Populárny význam slova”: paradigmy vo veľkom[c]
  - Procedurálna, logická, funkcionálna, objektovo-orientovaná paradigma. . .

---

[a] T. S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago, 1970.

[b] R. W. Floyd. The paradigms of programming. *Communications of the ACM*, 22(8), 1979.

[c] J. O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.

Paradigmy v malom

- Z pohľadu programovacieho jazyka
- Konfigurácie spoločného a premenlivého
- Analýza rozsahu, spoločného, premenlivého a vzťahov (SCVR)[a]
- Príklad: paradigma procedúra

  Rozsah: zoskupenie podobných častí kódu

  Spoločné: kód spoločný pre všetky časti

  Premenlivé: “nespoločný” kód; rozdiely ošetrené parametrami procedúry alebo prispôsobovacím kódom

[a] J. O. Coplien et al. Commonality and variability in software engineering. *IEEE Software*, 15(6), Nov. 1998.

Multi-Paradigm Software Development

- Dve záležitosti:
  - Zabezpečenie dostupnosti viacerých paradigiem: multiparadigmové jazyky (napr. Leda[a])
  - Výber vhodnej paradigmy pre problém ktorý sa rieši: multiparadigmový návrh
- Metódy multiparadigmového návrhu
  - Metóda multiparadigmového návrhu pre jazyk Leda[b]
  - Multiparadigmový návrh (pre C++)[c]

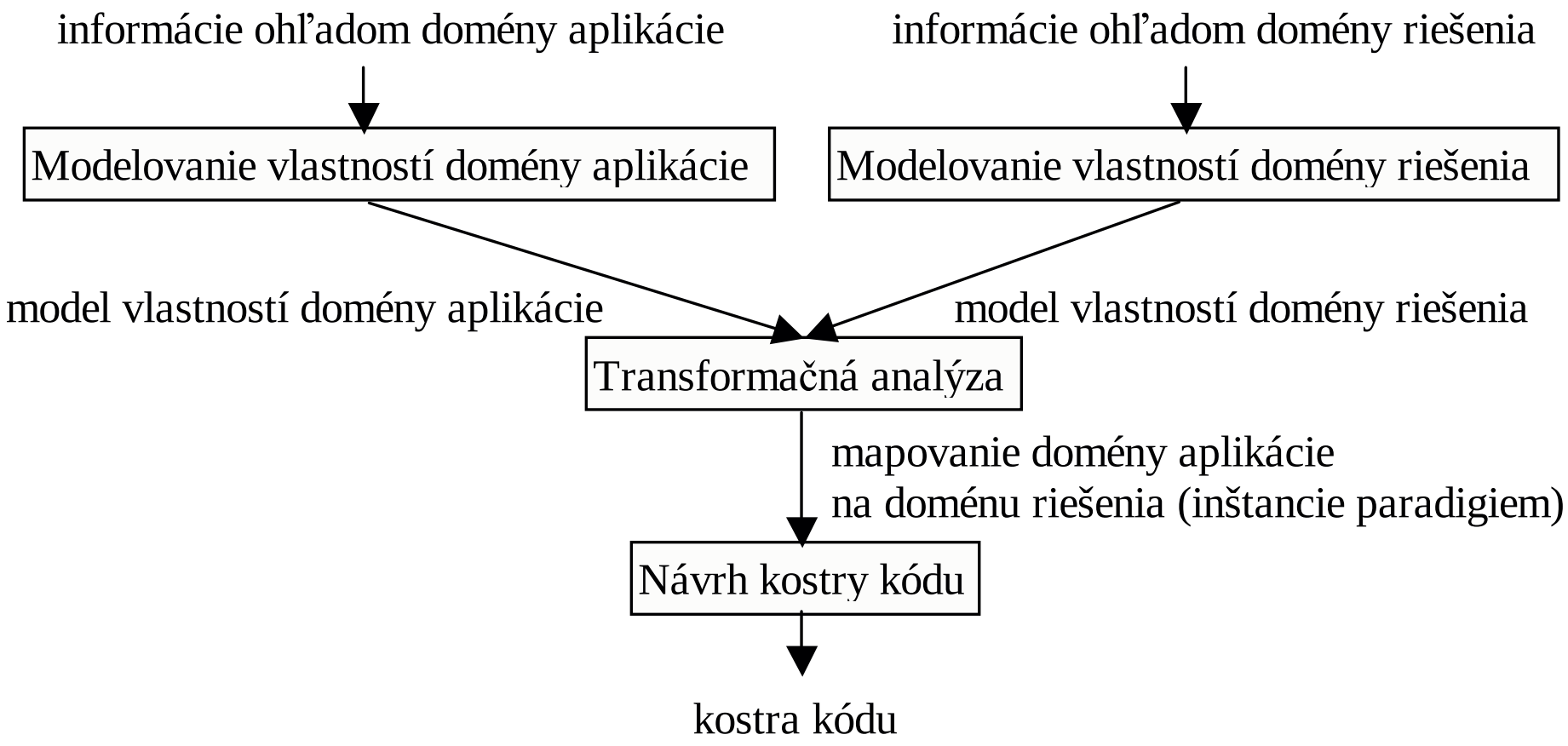
---

[a] T. A. Budd. *Multiparadigm Programming in Leda*. Addison-Wesley, 1995.

[b] C. D. Knutson et al. Multiparadigm design of a simple relational database. *ACM SIGPLAN Notices*, 35(12), Dec. 2000.

[c] J. O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.

Aktivity v MPD_{FM}

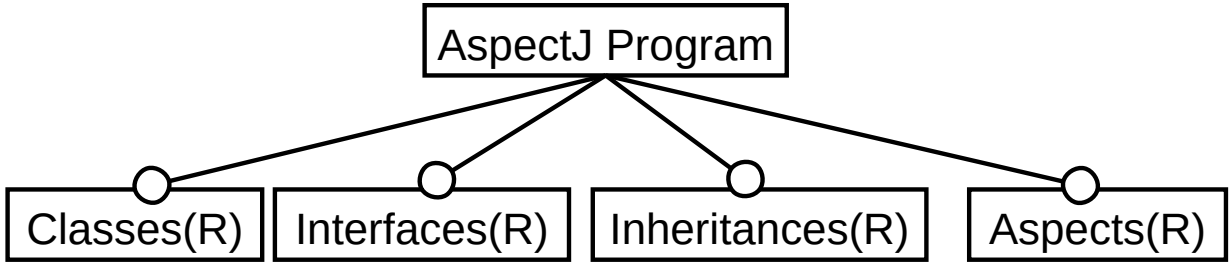


Modelovanie paradigiem v MPD_{FM}

- Identifikácia paradigiem
  - Priamo a nepriamo použiteľné paradigmy
  - Hierarchia paradigiem
- Identifikácia dôb viazania
  - Postupnosť dôb viazania ktoré poskytuje doména riešenia
  - Obvyklé doby viazania: tvorba zdrojového kódu, preklad, spájanie, načítavanie a vykonávanie
  - Príklad z jazyka AspectJ: telo metódy — viazanie v dobe vykonávania
- Prvá úroveň modelu paradigiem
- Modelovanie jednotlivých paradigiem

Prvá úroveň modelu paradigiem

- Koncept riešenia
- Pozostáva z priamo použiteľných paradigiem
  - Podkoncepty konceptu riešenia
  - Uvádzajú sa ako referencie konceptov (zvyčajne v množnom čísle)
  - Treba stanoviť ich variabilitu a dobu viazania
- Príklad: prvá úroveň modelu paradigiem jazyka AspectJ



Modelovanie jednotlivých paradigiem

- Každá paradigma sa uvádza vo zvláštnom diagrame vlastností
  - Koncepty domény riešenia
  - Môžu sa odkazovať jedny na druhé
- Pomocné koncepty
  - Koncepty na ktoré sa paradigmy odkazujú
  - Ale nepokladajú sa za paradigmy ako také
- Doba viazania (premenlivé vlastnosti)
- Inštanciácia sa modeluje pomocou vlastností

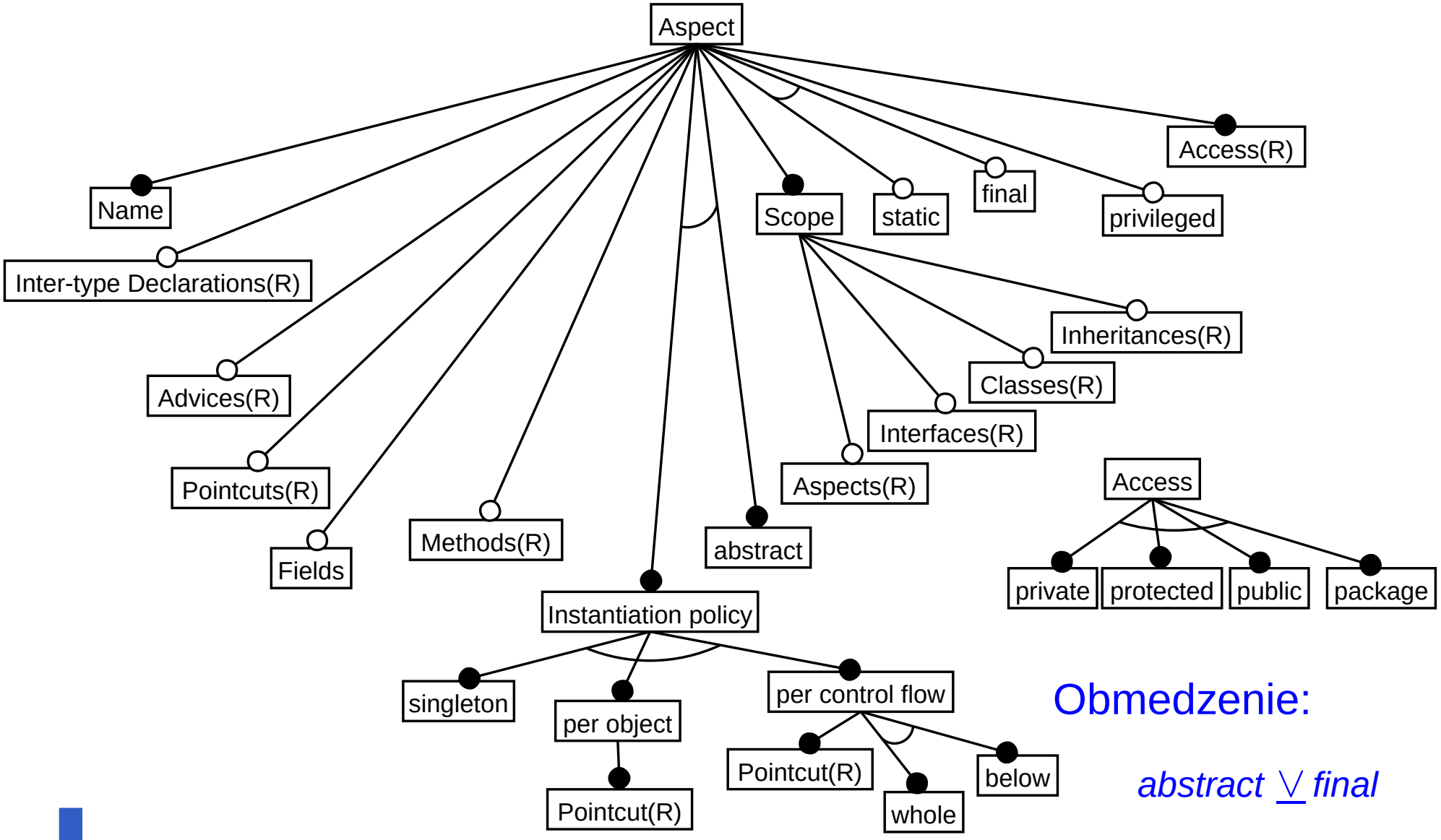
Štruktúry a vzťahy

- Štrukturálne paradigmy zodpovedajú hlavným konštrukciám (štruktúram) programovacieho jazyka
- Paradigmy vzťahu sa týkajú vzťahov medzi štruktúrami programovacích jazykov
- V transformačnej analýze uzol v modeli vlastností domény aplikácie
  - Môže zodpovedať koreňu štrukturálnej paradigmy
  - Ale nemôže zodpovedať koreňu paradigmy vzťahu

Aspektovo-orientované paradigmy

- Aspektovo-orientované programovanie
  - Modularizácia pretínajúcich záležitostí
  - Vo všeobecnosti užitočné pre ladenie, sledovanie a synchronizáciu
  - Aplikačne špecifické aspekty
- Aspektovo-orientované paradigmy jazyka AspectJ
- Paradigma aspekt:
  - Štrukturálna paradigma (modularizácia)
  - Obsahuje rady, bodové rezy a medzitypové deklarácie — paradigmy vzťahov (pretínajúce záležitosti)

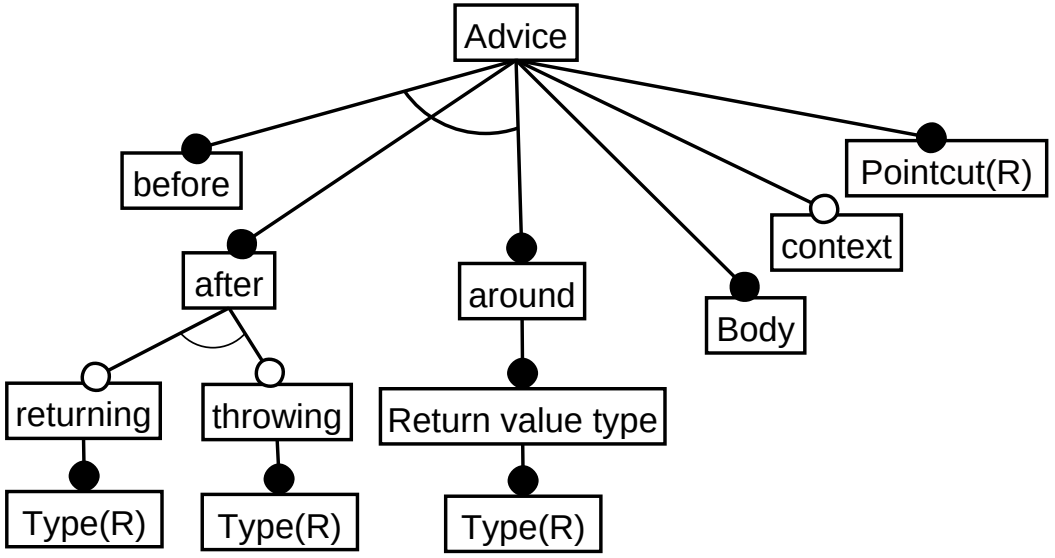
Aspect

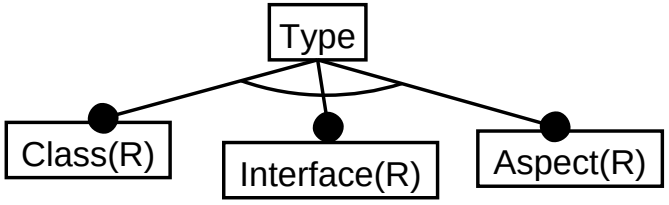


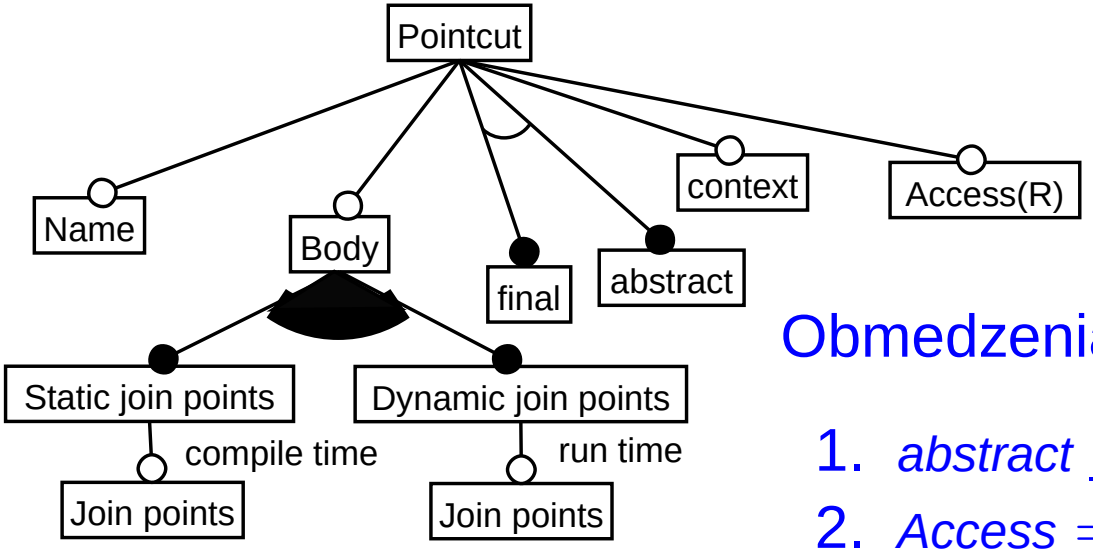
Obmedzenie:

$abstract \veebar final$

Advice / Pointcut







Obmedzenia:

1. *abstract* $\veebar$ *Body*
2. *Access* $\Rightarrow$ *Name*

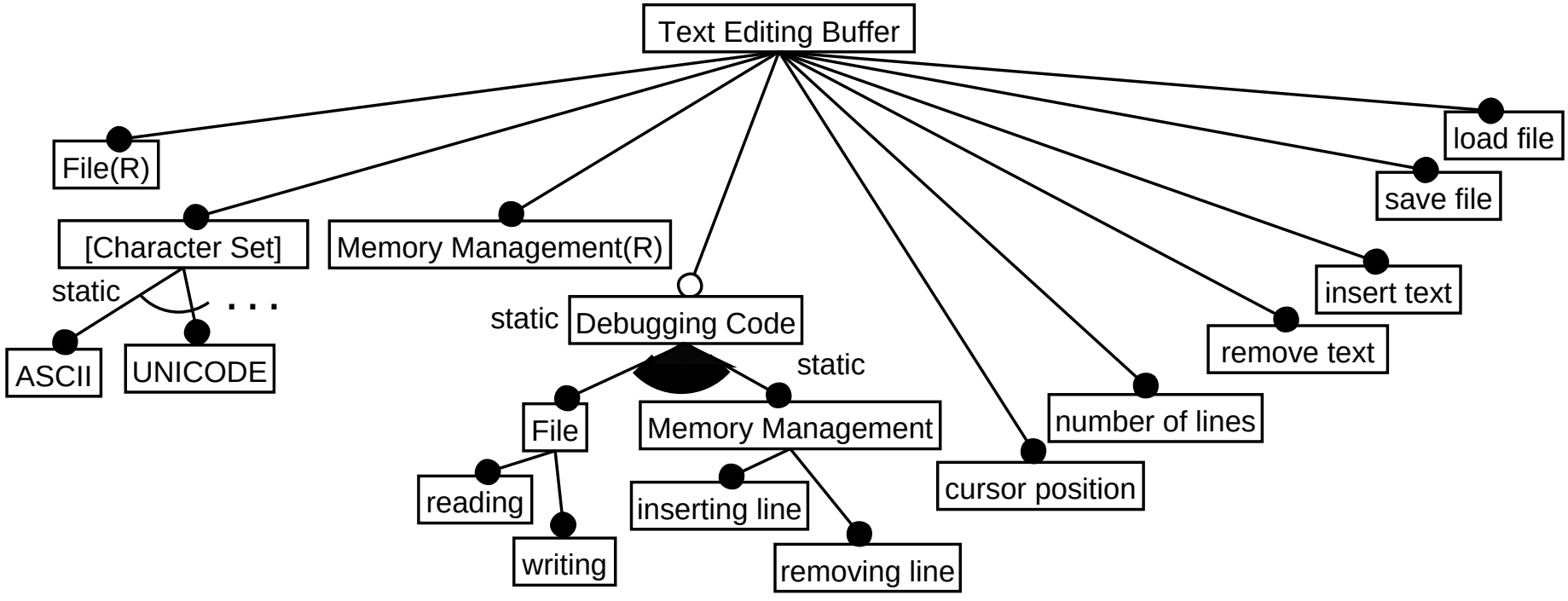
Transformačná analýza v MPD_{FM}

- Založená na inštanciácii paradigiem nad konceptami domény aplikácie v dobe tvorby zdrojového kódu
- O konceptoch domény aplikácie sa uvažuje po jednom
  1. Určí sa štrukturálna paradigma konceptu domény aplikácie
  2. Určia sa zodpovedajúce paradigmy vzťahov pre každý nenamapovaný vzťah v ňom
- Tvorivý proces

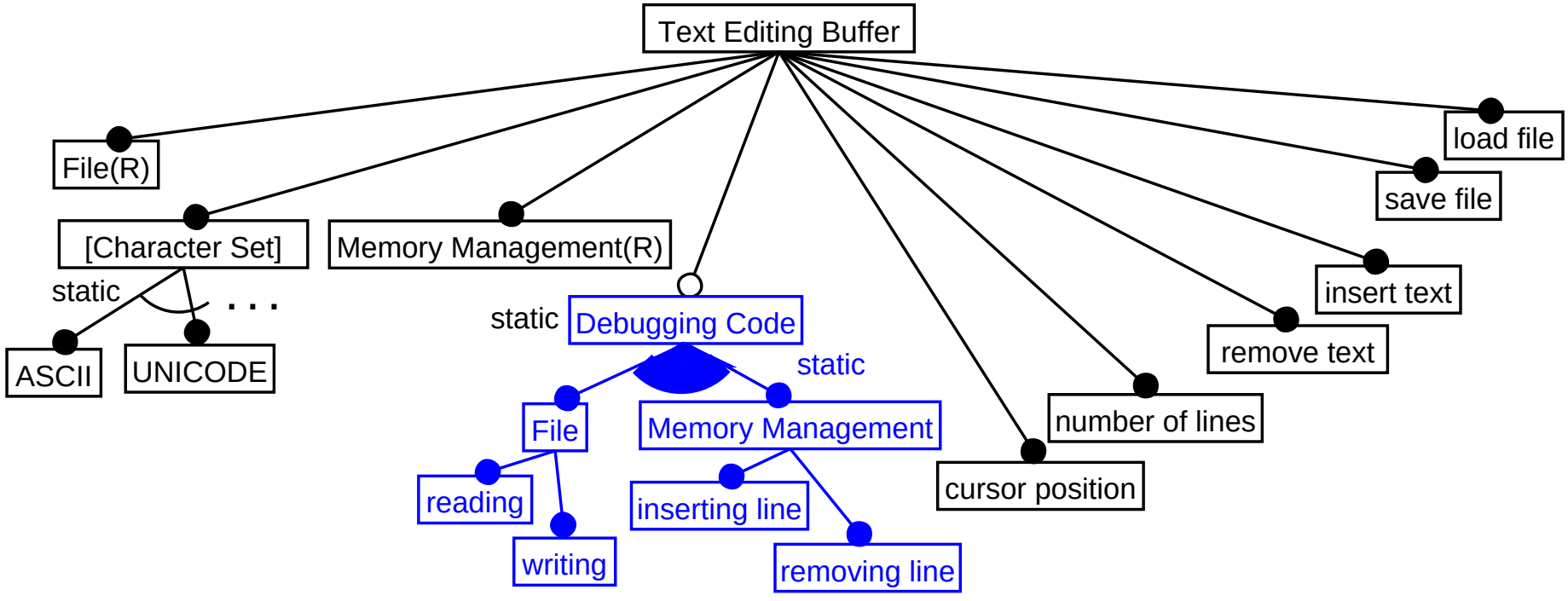
Inštanciácia paradigiem v MPD_{FM}

- Inštanciácia konceptov v MPD_{FM}
  - Chápe sa ako špecializácia konceptov
  - Inštancie konceptov sa reprezentujú pomocou diagramov vlastností
  - Berie sa do úvahy doba viazania
- Inštanciácia zdola-nahor
- Zahrnutie uzlov paradigmy podmienené mapovaním uzlov konceptov domény aplikácie
  - Konceptuálny súlad
  - Súlad dôb viazania

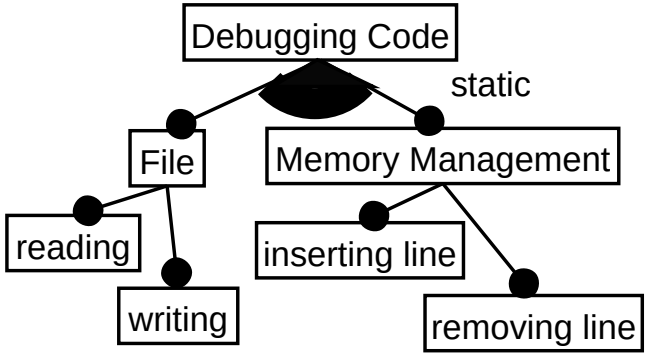
Príklad transformačnej analýzy



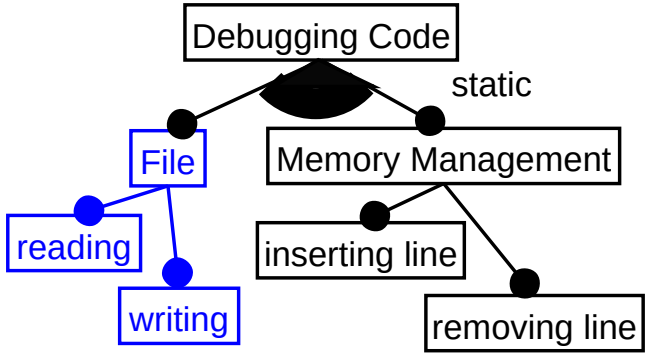
Príklad transformačnej analýzy



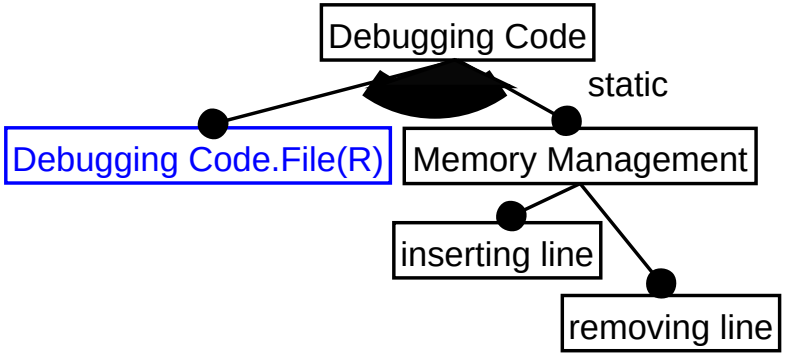
Príklad transformačnej analýzy

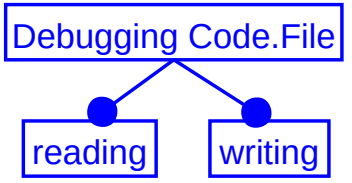


Príklad transformačnej analýzy

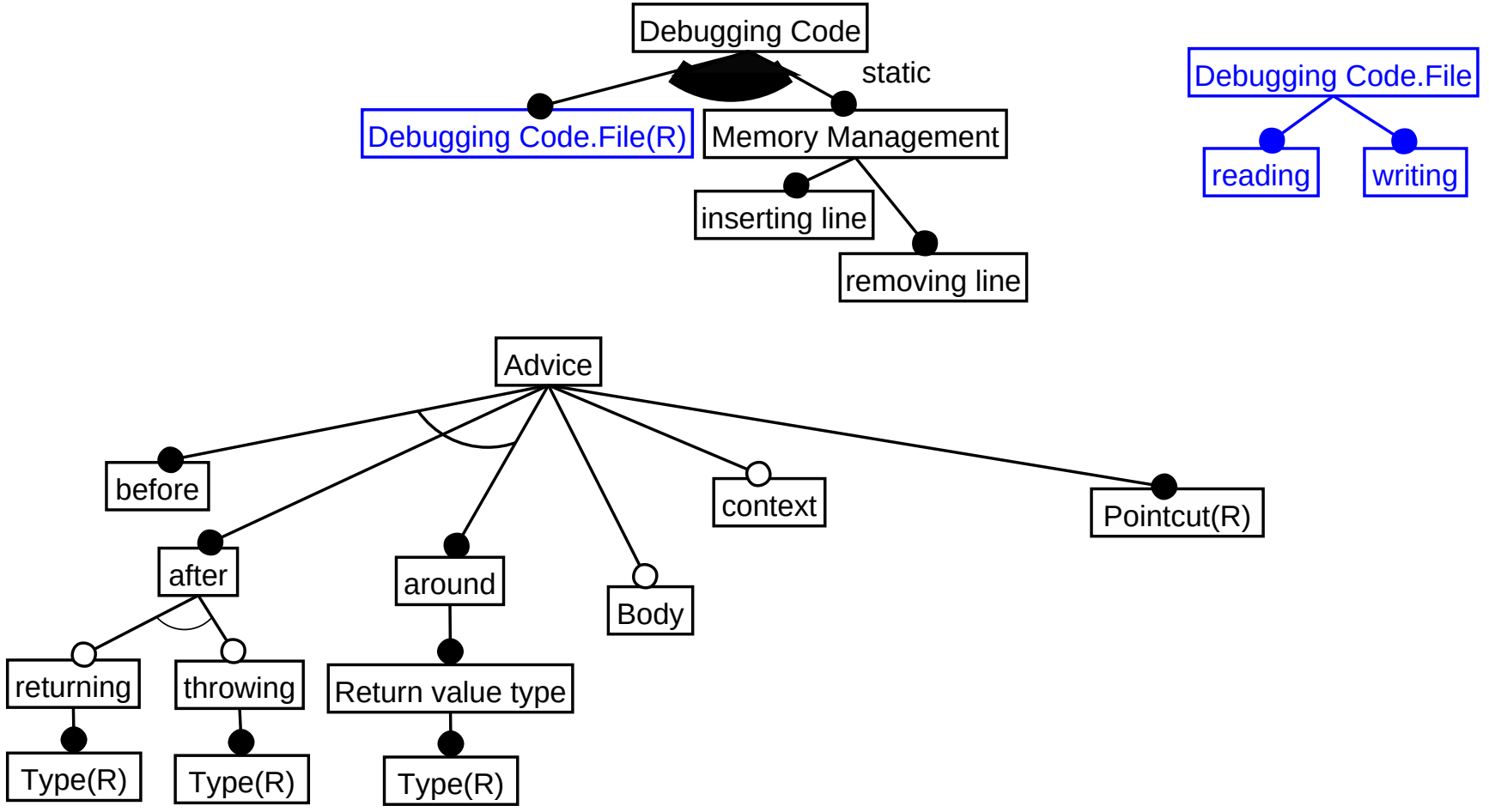


Príklad transformačnej analýzy

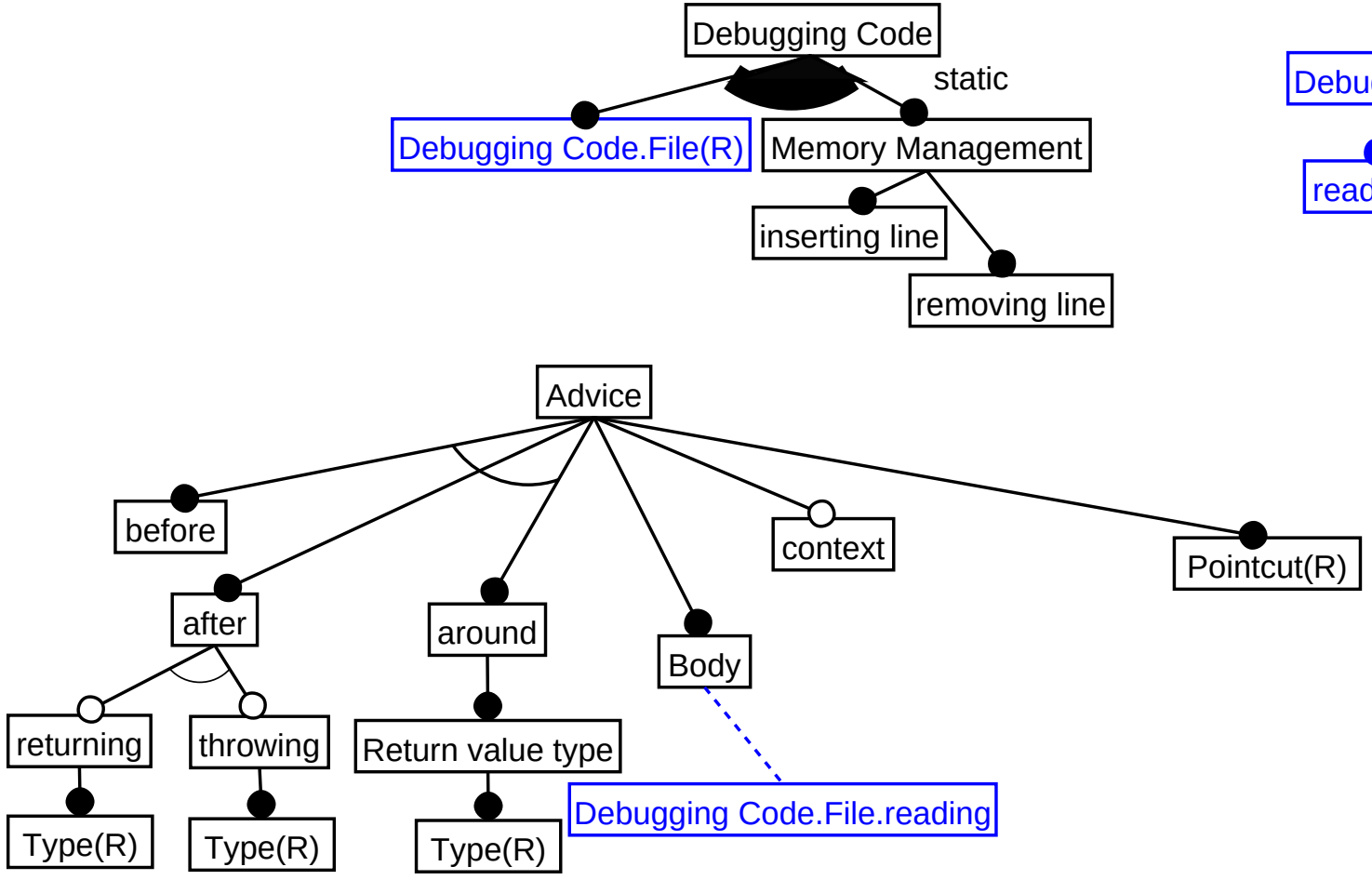




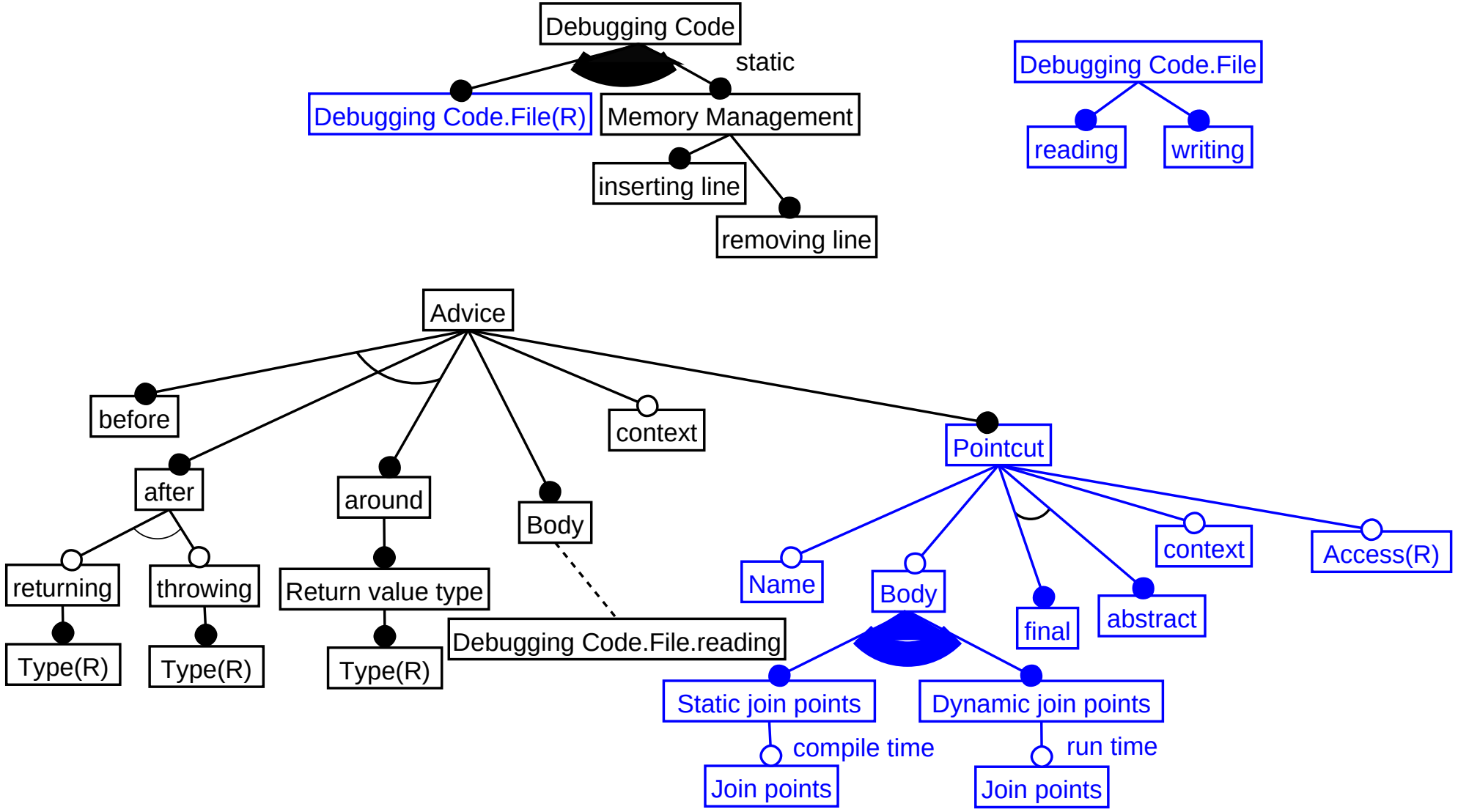
Príklad transformačnej analýzy



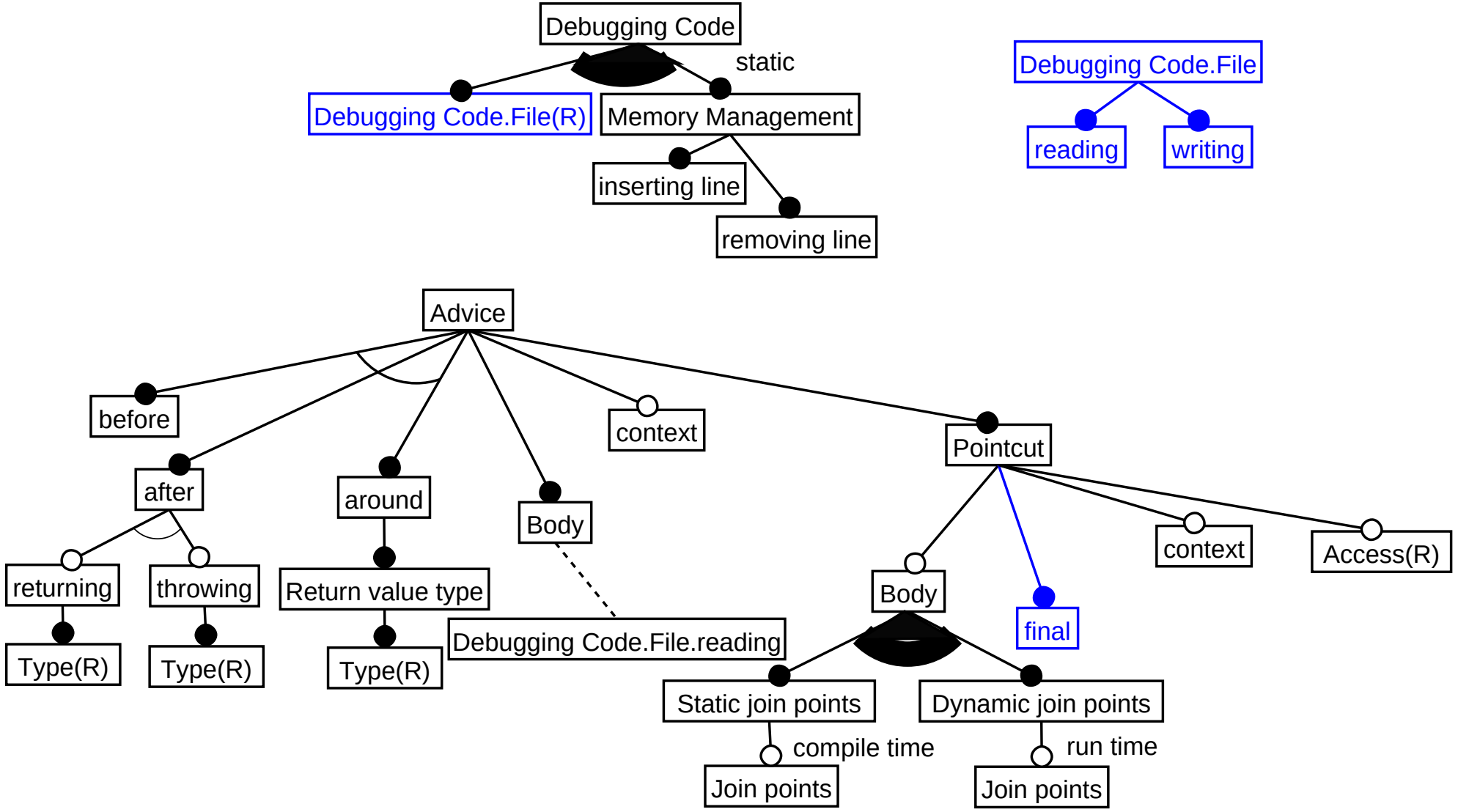
Príklad transformačnej analýzy



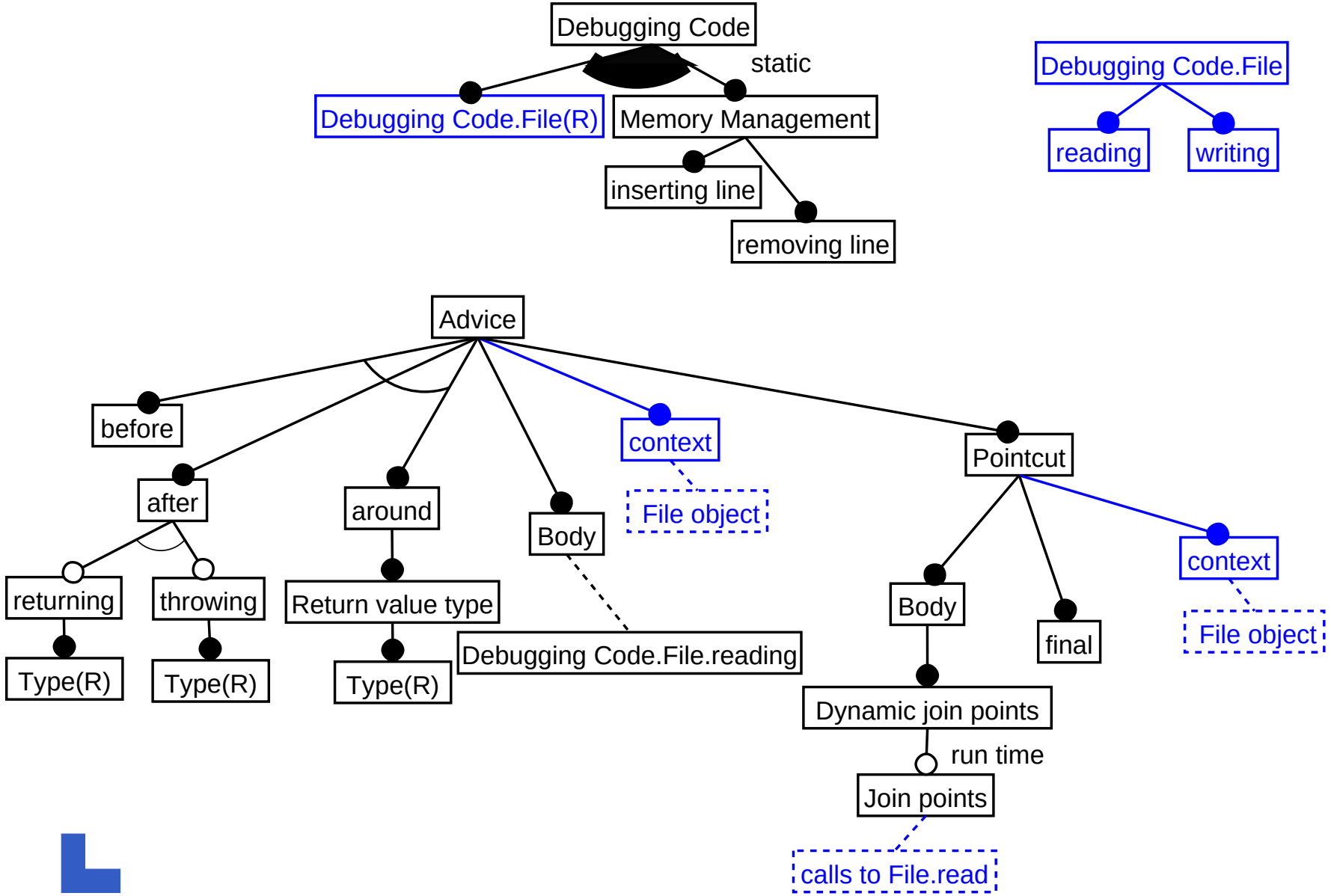
Príklad transformačnej analýzy



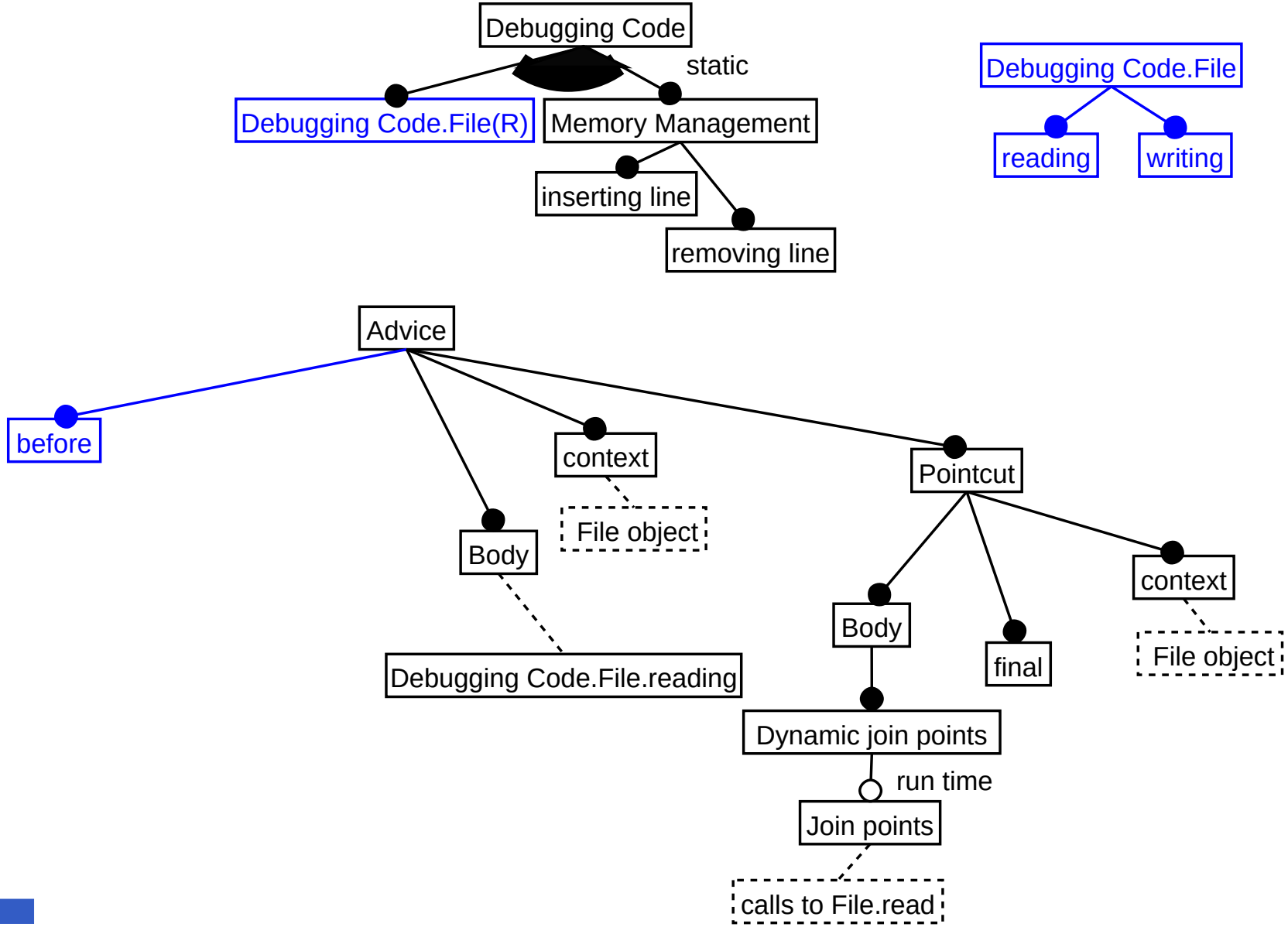
Príklad transformačnej analýzy



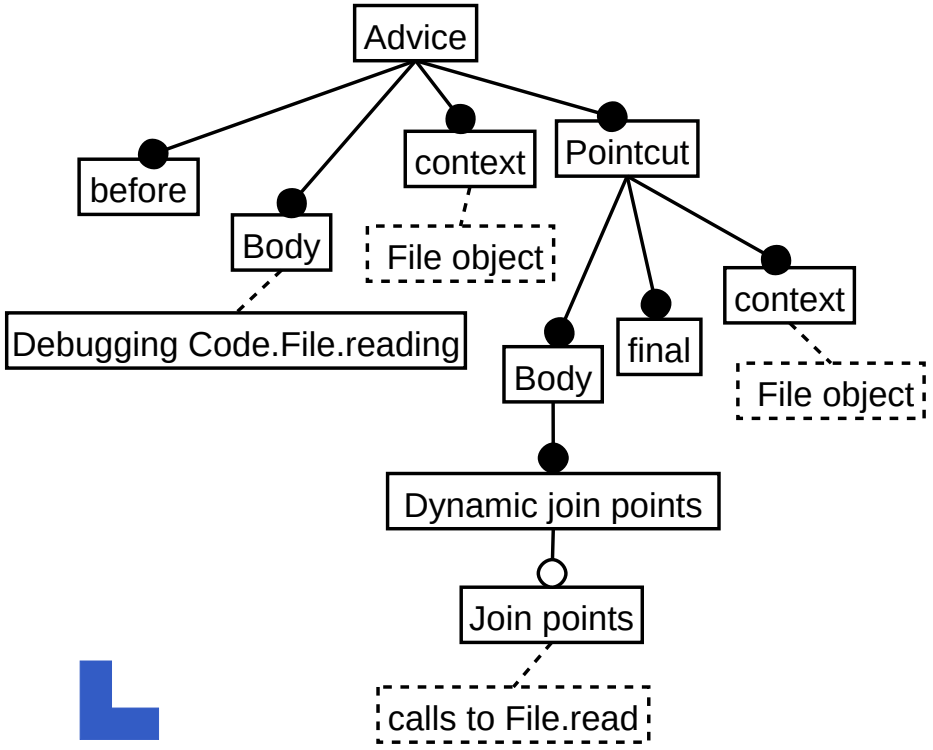
Príklad transformačnej analýzy



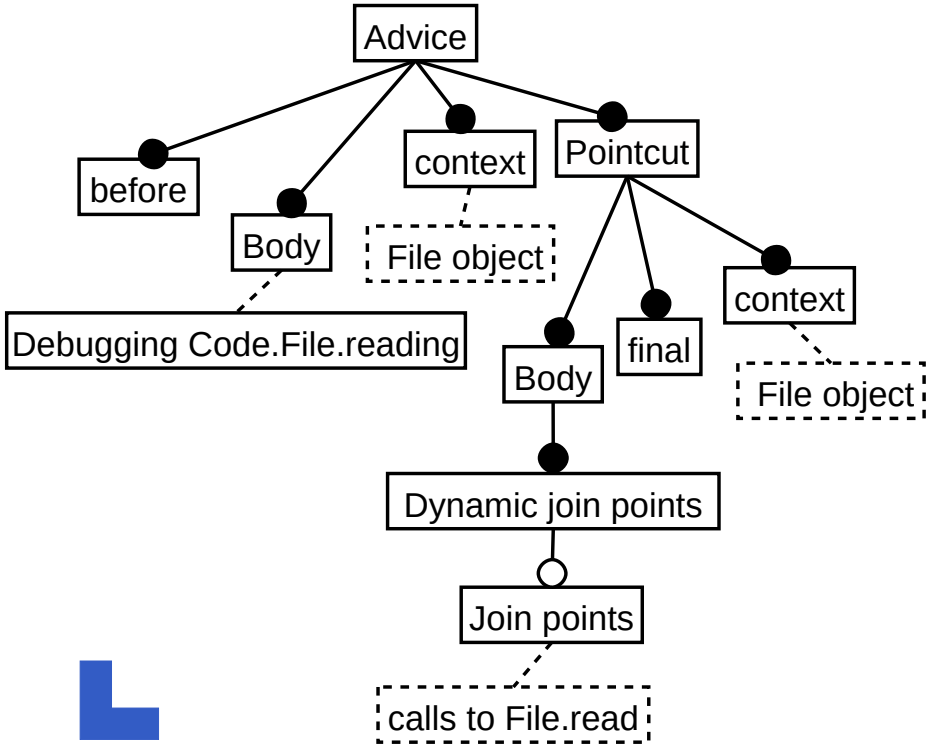
Príklad transformačnej analýzy

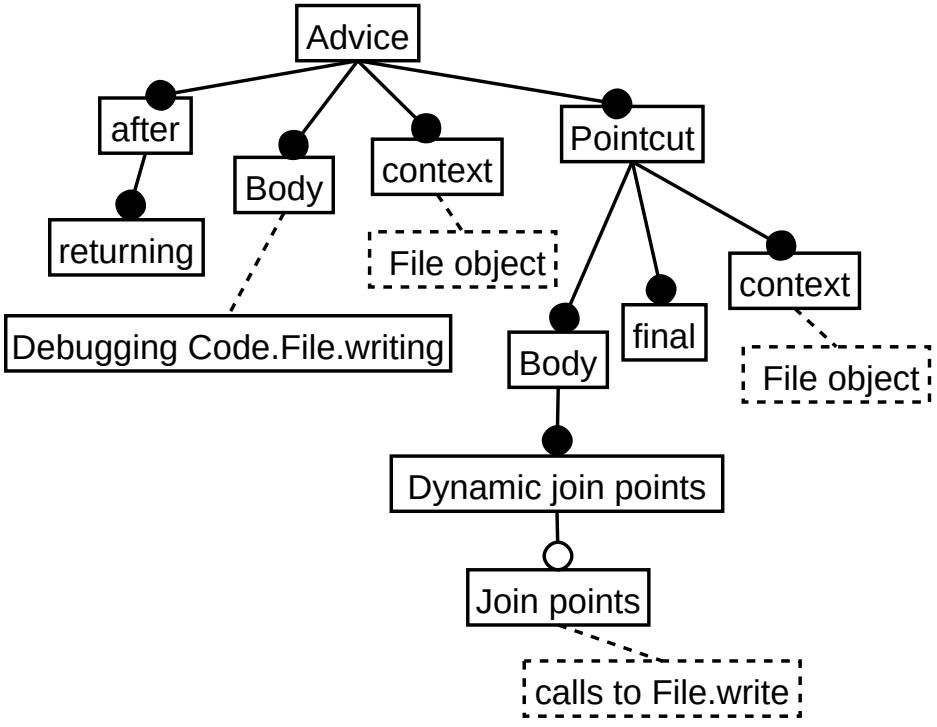


Príklad transformačnej analýzy

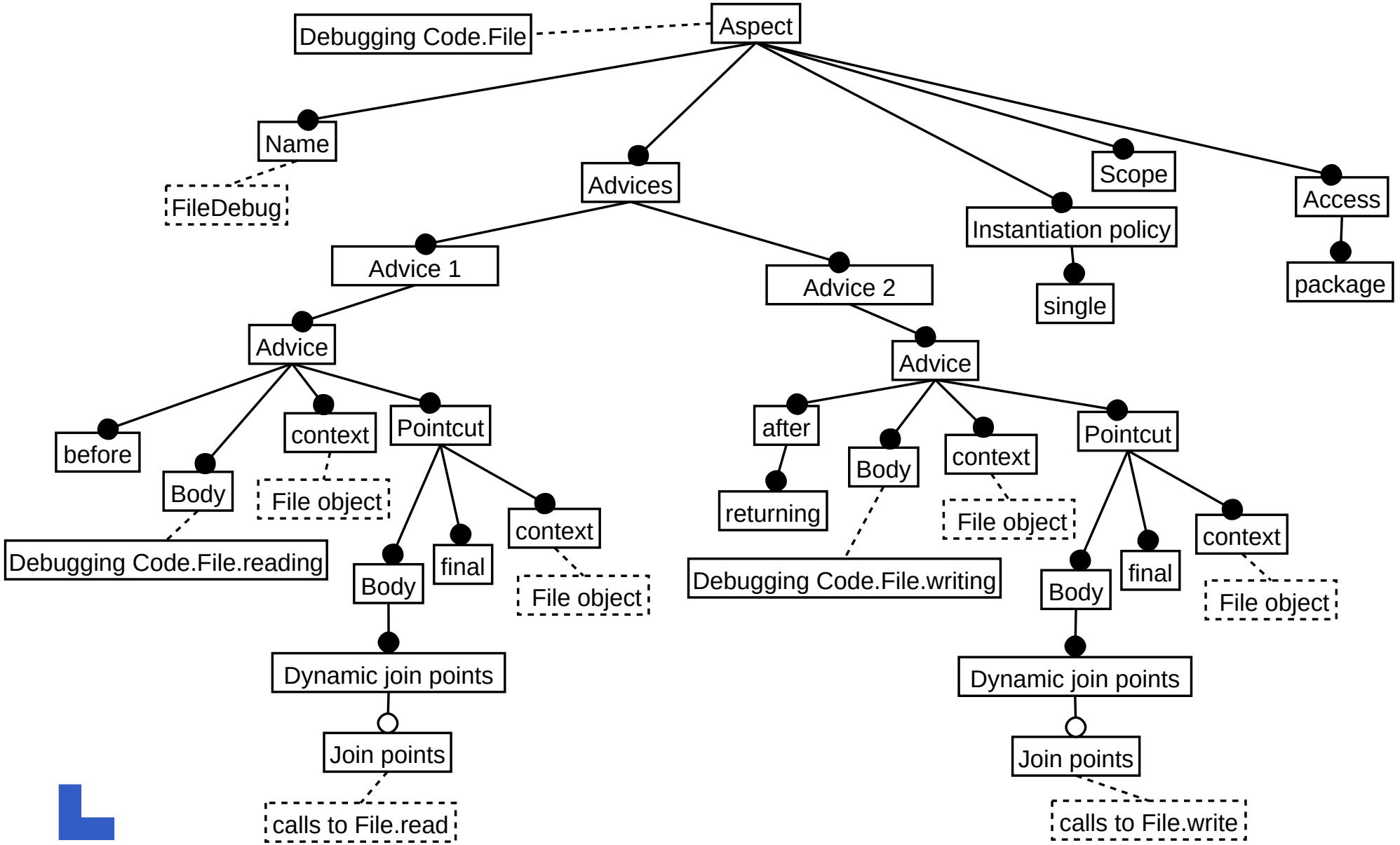


Príklad transformačnej analýzy





Príklad transformačnej analýzy



Návrh kostry kódu

- Uskutočňuje sa prechádzaním stromami inštancií paradigiem
- Najprv inštancie štrukturálných paradigiem
- Príklad: aspekt ladenia kódu práce so súbormi

```
aspect FileDC {
     before(File f):
         target(f) && call(* File.read(..)) {
             . . .
         }

     after(File f):
         target(f) && call(* File.write(..)) {
             . . .
         }
}
```

Vyhodnotenie MPD_{FM}

- Modelovanie vlastností
  - Modelovanie vlastností domény aplikácie aplikované na samotnú doménu modelovania vlastností
  - Modelovanie vlastností domény riešenia aplikované na jazyk AspectJ: MPD_{FM} pre AspectJ
- Transformačná analýza a návrh kostry kódu
  - MPD_{FM} pre AspectJ použitý v transformačnej analýze návrhu kostry kódu domény modelovania vlastností
  - Modely domény aplikácie a riešenia sa môžu nezávisle od seba znovupoužívať v transformačnej analýze

Zhodnotenie (1)

- Multiparadigmový návrh s modelovaním vlastností (MPD_{FM}):
  - Aj doména aplikácie, aj doména riešenia reprezentované ako modely vlastností
  - Transformačná analýza plne založená na modelovaní vlastností
- MPD_{FM} pre AspectJ
  - Model paradigiem jazyka AspectJ
  - Použitie ukázané v transformačnej analýze vyrovnávacej pamäti editora textu
  - Úspešne aplikovaný na doménu modelovania vlastností

Zhodnotenie (2)

- Vylepšenia modelovania vlastností:
  - Inštanciácia konceptov zohľadňujúca dobu inštanciácie
  - Parametrizácia v modeloch vlastností
  - Obmedzenia a pravidlá preddefinovaných závislostí pomocou logických výrazov
  - Referencie konceptov
  - Bodková konvencia umožňujúca jednoznačné odvolávanie sa na koncepty a vlastnosti
  - Parametrizovaný koncept na reprezentáciu kardinality v modelovaní vlastností

Ďalšia práca

- Čiastkové znovupoužitie modelov vlastností
  - Prekrývajúce sa domény
  - Zovšeobecňovanie podobných konceptov z rôznych domén
- Špecializácia MPD_{FM} na domény riešenia iné než programovacie jazyky

Otázky oponentov

- prof. Ing. Milan Krokavec, PhD.
  Fakulta elektrotechniky a informatiky TU Košice
- doc. Ing. Karel Richta, CSc.
  Fakulta elektrotechniky ČVUT Praha
- assoc. prof. dr. Tatjana Welzer Družovec
  Univerzita v Maribore, Slovinsko

prof. Krokavec — otázky (1)

Mohol dizertant pre svoje modifikované modelovanie vlastností použiť niektorý z podporných nástrojov ktoré spomína na strane 44 a 45?

- ASADAL nevhodný lebo vnucuje notáciu FORM
- AmiEddi sa v zásade dá použiť
  - Nepodporuje však referencie konceptov
  - Pridružené informácie sa nedajú exportovať
  - Chýbajú lepšie možnosti exportu diagramov vlastností
- Captain Feature
  - Lepší ako AmiEddi
  - Fatálne problémy s ukladaním súborov

prof. Krokavec — otázky (2)

Prečo sa metóda nazýva . . . s modelovaním vlastností a nie . . . s modelovaním čŕt resp. "rysov", čo korešponduje anglickému názvu?

- Feature (Czarnecki-Eisenecker):

  A property of a domain cocncept, which is relevant to some domain stakeholder and is used to discriminate between concept instances.

- Feature sa teda definuje ako property, čo je vlastnosť
- *Mať vlastnosť* vs. *mať črtu (rys)*

doc. Richta — otázky (1)

Peter Coad publikoval metodiku nazývanou Feature Driven Development. Zajímal by mne vztah mezi touto metodikou a autorovým modelem.

- FDD nie je založený na modelovaní vlastností
- Nezohľadňuje explicitne variabilitu — cieľom je vývoj jedného systému, nie rodiny
- Pojem vlastnosti menej všeobecný než v MPD_{FM}
  - Len vlastnosti z pohľadu používateľa
  - Rozsah vlastnosti je obmedzený

doc. Richta — otázky (2)

Nebylo by možné pro reprezentaci modelů vlastností používat standardní diagramy UML?

- Stereotypy — nie celkom korektné riešenie
- Rozšírenie metamodelu

doc. Richta — otázky (3)

Proč autor nedovedl v práci uváděný příklad vyrovnávací paměti textového editoru až do fáze kódu? Lze nějakým způsobem změřit, či odhadnout efektivitu řešení? Nebo srovnat jeho výsledek s jiným řešením?

- Zámer bol demonštrovať jednotlivé kroky metódy
- Dizertačná práca obsahuje kompletný príklad aplikácie metódy na doménu modelovania vlastností
- Efektivita riešenia by sa dala hodnotiť na úrovni kódu (zohľadňujúc jeho znovupoužiteľnosť a prispôsobivosť)
- MPD_{FM} umožňuje lepšiu kontrolu nad procesom výberu paradigiem ako multiparadigmový návrh a metóda multiparadigmového návrhu pre jazyk Leda

doc. Richta — otázky (4)

V práci jsem nenašel žádnou zmínku o projektech, ve kterých by byly myšlenky práce využívány. Existují takové?

- Aplikácia MPD_{FM} na doménu modelovania vlastností
- Študentské projekty

doc. Richta — otázky (5)

Jak dopadl jeho příspěvek do CAI?

- Bol zamietnutý
- Príspevok sa teraz posudzuje v *Transactions on Aspect-Oriented Programing*
- Bol však prijatý príspevok:

  Valentino Vranić. Reconciling Feature Modeling: A Feature Modeling Metamodel. In Matias Weske and Peter Liggsmeyer, editors, *Proc. of 5th Annual International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World (Net.ObjectDays 2004)*, LNCS 3263, Erfurt, Germany, September 2004. Springer.

assoc. prof. Welzer — otázky (1)

During the defense I would like that the candidate point out the role of conceptual modeling in his research as well as reuse. Especially interesting is similarity of individual concepts.

- Konceptuálne modelovanie a MPD_{FM}
  - MPD_{FM} je založený plne na modelovaní vlastností
  - MPD_{FM} podporuje znovupoužitie modelov domén aplkácie a riešenia v transformačnej analýze
- Podobnosť konceptov
  - Parametrizácia konceptov
  - Zovšeobecňovanie podobných konceptov z rôznych domén — ďalšia práca

assoc. prof. Welzer — otázky (2)

I am also interested in the relationship between the concept and the paradigm.

- Vzťah konceptu a paradigmy
  - Koncept: vnímanie triedy alebo kategórie prvkov v určitej doméne
  - Paradigma: koncept domény riešenia