

# Doménové inžinierstvo

Valentino Vranić

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

ÚISI FIIT STU

Architektúra softvérových systémov

Prednáška 24. novembra 2005

- Dnes prevláda tvorba individuálnych systémov
  - Špecifický systém pre špecifického zákazníka a kontext
- Doménové inžinierstvo — organizovaný prístup k znovupoužitiu
- Vývoj jedného softvérového systému vs. vývoj rodiny softvérových systémov
- Modelovanie domény: zachytenie variability
- Tvorba komponentov
- Implementácia jednotlivých členov rodiny

# Prehľad prednášky

- Doménové inžinierstvo
- Rady softvérových výrobkov
- Analýza domény
- Modelovanie vlastností

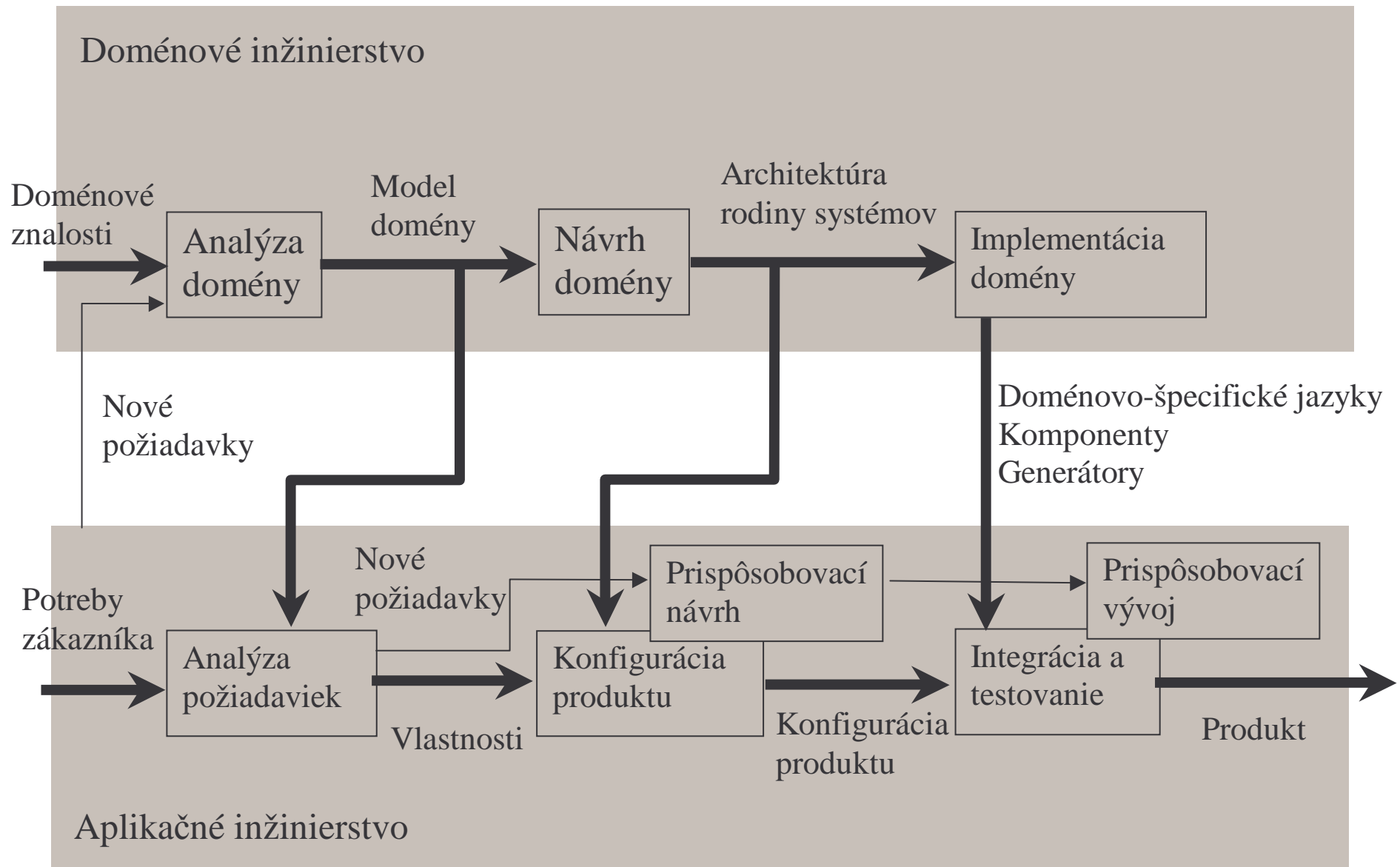
- Doménové inžinierstvo (*domain engineering*)  
Aktivita zbierania, organizovania a ukladania skúseností v stavbe systémov alebo ich častí v určitej doméne v tvare znovupoužiteľných prvkov, ako aj poskytovania adekvátnych prostriedkov na znovupoužitie týchto prvkov pri stavbe nových systémov.
- Znovupoužitie sa uskutočňuje v aplikačnom inžinierstve
- Softvérové systémy v doméne sú podobné

---

<sup>a</sup>K. Czarnecki, U. Eisenecker. *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley, 2000.

- Dva pohľady:
  - Doména ako "reálny svet" (napr. bankovníctvo)
  - *Doména ako množina systémov* (napr. bankovníctvo ako také + softvérové systémy na jeho podporu)
- Doména: oblasť vedomostí
  - S rozsahom určeným tak aby uspokojil požiadavky zainteresovaných
  - Zahŕňa množinu pojmov a terminológiu zrozumiteľnú pôsobiacim v danej oblasti
  - Zahŕňa vedomosti o tom ako tvoriť softvérové systémy (alebo ich časti) v danej oblasti

# Proces doménového inžinierstva



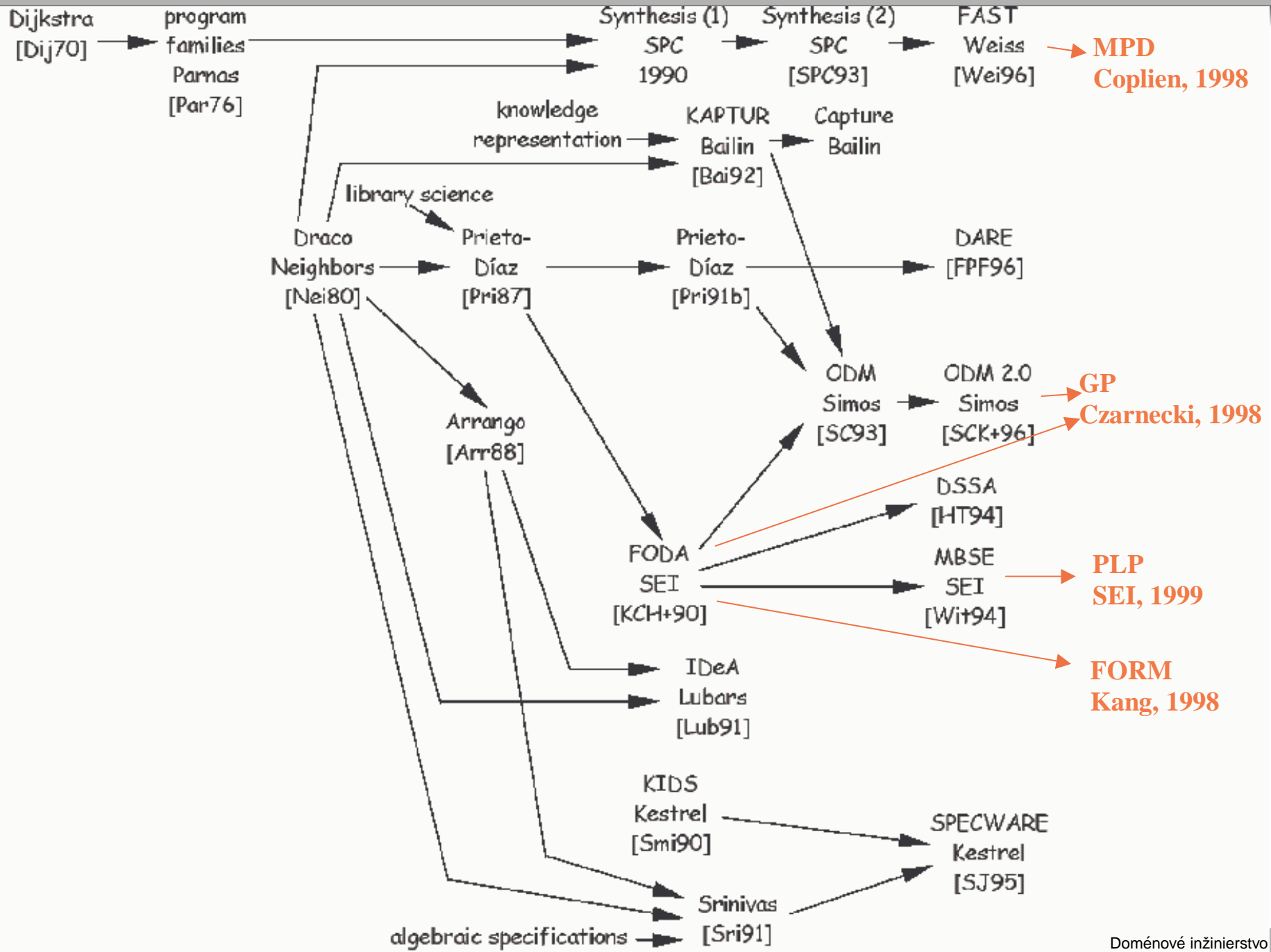
# Analýza domény

- Tvorba modelu domény
- Aktivity:
  - Vymedzenie domény
    - Vyplýva zo záujmov zainteresovaných
    - Ekonomický zvládnuteľné
  - Modelovanie domény
    - Definícia domény
    - Slovník domény
    - Modely konceptov
    - *Modely vlastností*
- Každá reálna doména sa dá zjemňovať donekonečna

# Návrh a implementácia domény

- Návrh domény
  - Tvorba architektúry rodiny systémov
  - Tvorba plánu výroby
- Implementácia domény
  - Architektúra
  - Komponenty
  - Plán výroby

# Vývoj doménového inžinierstva



# Rady softvérových výrobkov

- *Software product-lines*
- *A Framework for Software Product Line Practice*
- Iniciatíva SEI (*Software Engineering Institute*)
  - <http://www.sei.cmu.edu/plp/>
- Rieši záležitosti manažmentu a organizácie ohľadom zriadenia procesu doménového a aplikačného inžinierstva

# Zavedenie radu softvérových výrobkov<sup>a</sup>

- Evolučne / revolučne
- Jestvujúci rad výrobkov / nový rad výrobkov

⇒ Štyri možnosti:

- Postupný vývoj radu výrobkov z jestvujúcej množiny výrobkov
- Nahradenie jestvujúcej množiny výrobkov radom výrobkov
- Postupný vývoj nového radu výrobkov
- Vývoj nového radu výrobkov (pred vyrobením prvého produktu)

---

<sup>a</sup>J. Bosch. *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*. Addison-Wesley, 2000.

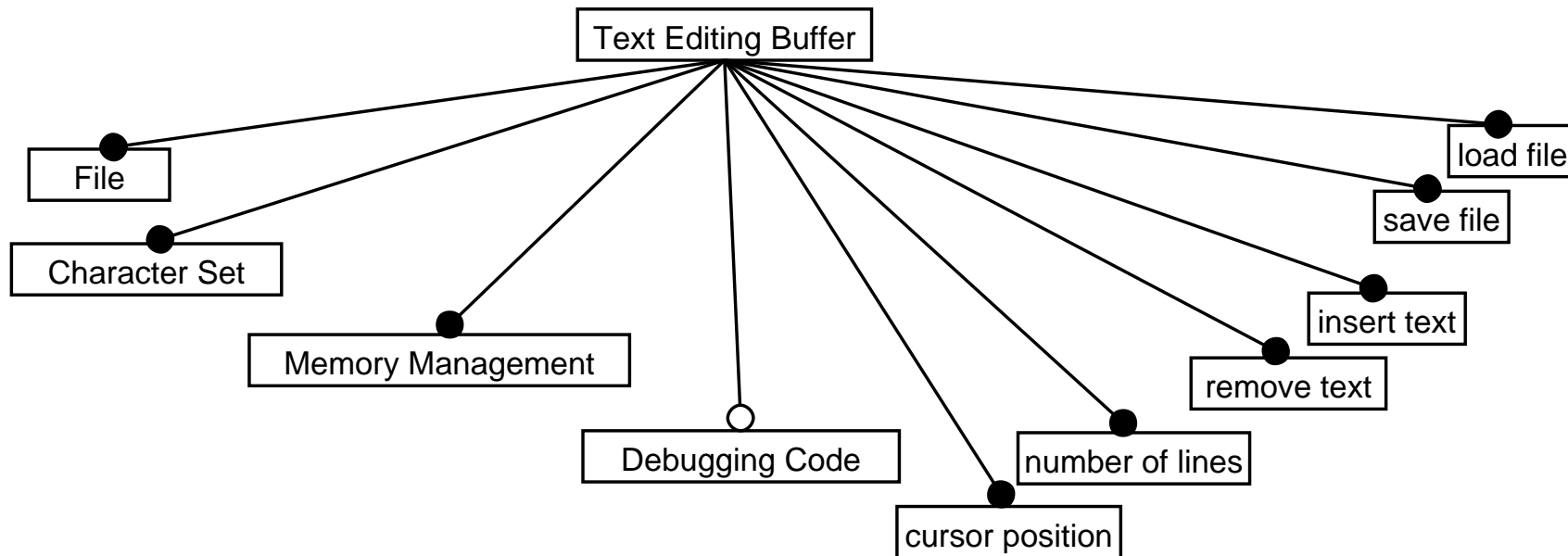
# Modelovanie vlastností (*feature modeling*)

- Zachytáva prepojenia medzi vlastnosťami a variabilitu
- Model vlastností: množina diagramov vlastností plus ďalšie informácie
- Založené na pojmoch *domény, konceptu a vlastností*
  - Vlastnosti: spoločné a premenlivé
  - Inštancie konceptov: špecializácie konceptov
- Rôzne notácie, napr. FODA, ODM, Czarneckého-Eiseneckerova a *modelovanie vlastností pre multiparadigmový návrh<sup>a</sup>*

---

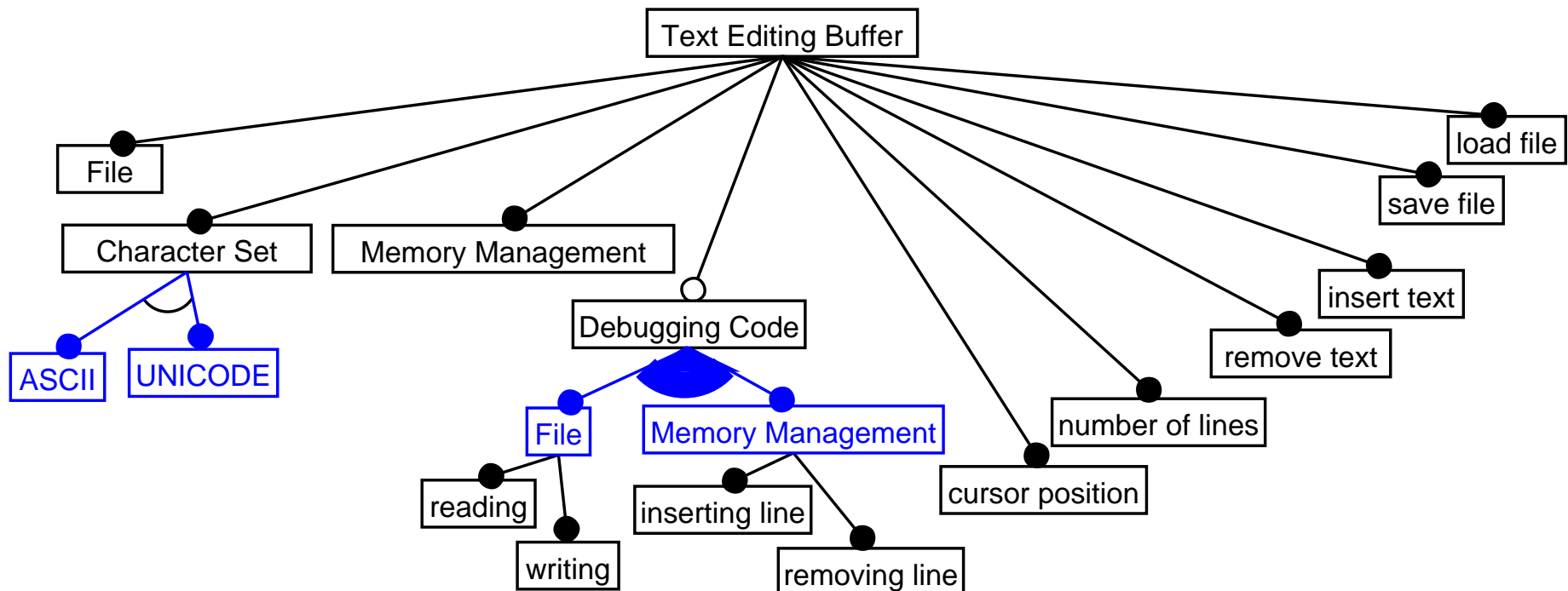
<sup>a</sup>V. Vranić. Reconciling Feature Modeling: A Feature Modeling Metamodel. In Mathias Weske and Peter Liggesmeyer, editors, *Proc. of 5th Annual International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World (Net.ObjectDays 2004)*, Erfurt, Germany, September 2004. Springer.

# Variabilita vlastností



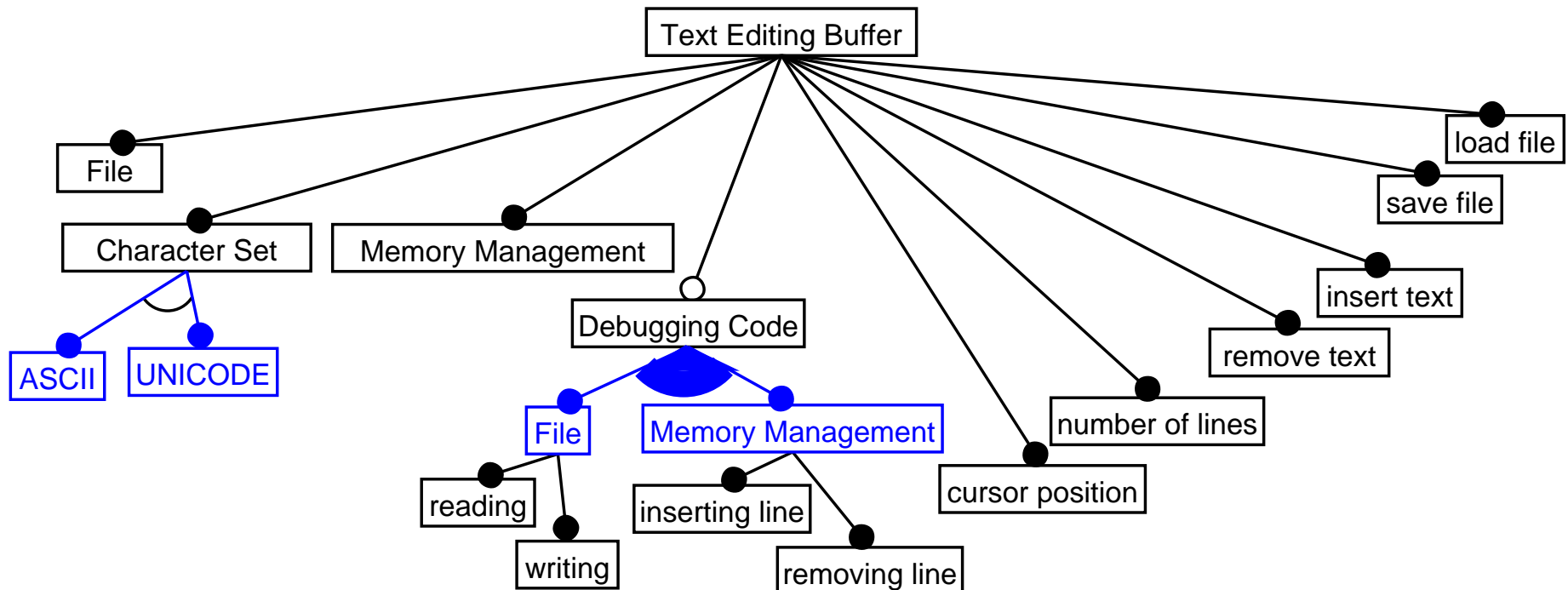
- *Povinné* vlastnosti (hrany ukončené vyplnenými krúžkami)
- *Voliteľné* vlastnosti (hrany ukončené prázdnyimi krúžkami)

# Variabilita vlastností



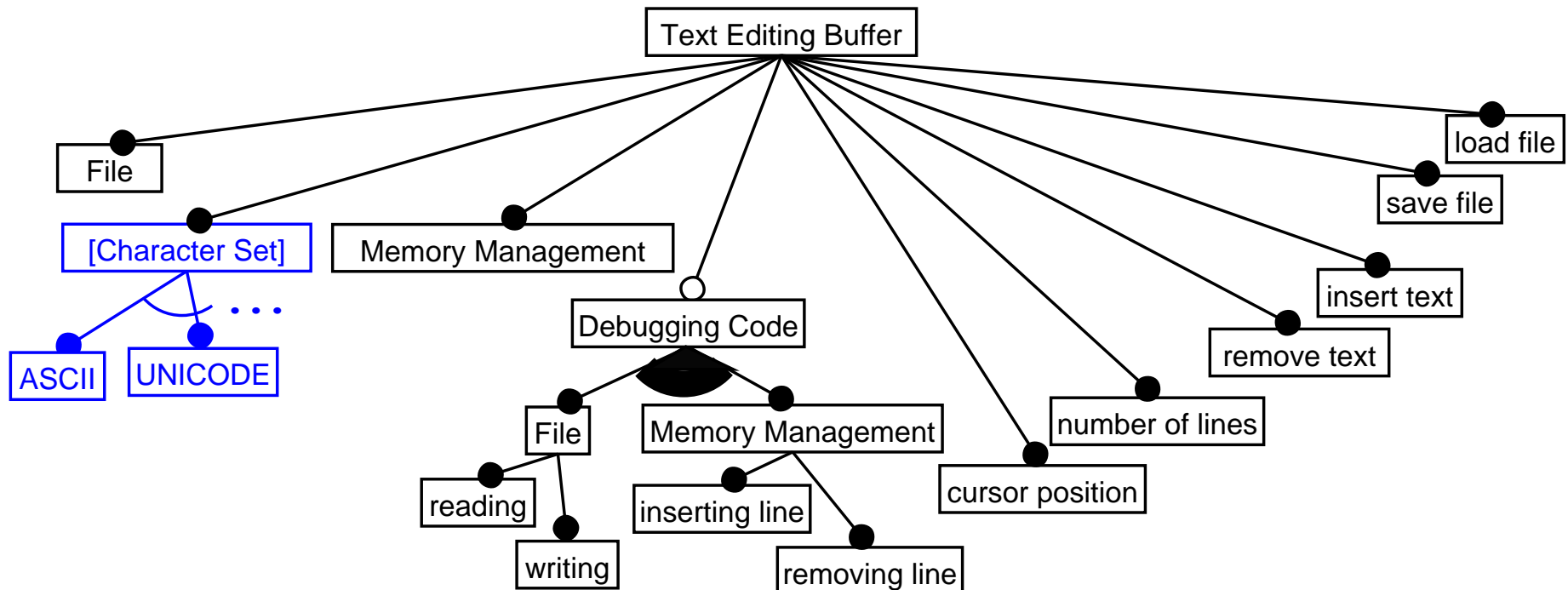
- *Alternatívne vlastnosti (prázdny oblúk)*
- *Alebo-vlastnosti (vyplnený oblúk)*

# Variabilita vlastností



- Oblúky upravujú význam hrán
  - Povinné alternatívne / voliteľné alternatívne vlastnosti
  - Povinné alebo- / voliteľné alebo-vlastnosti

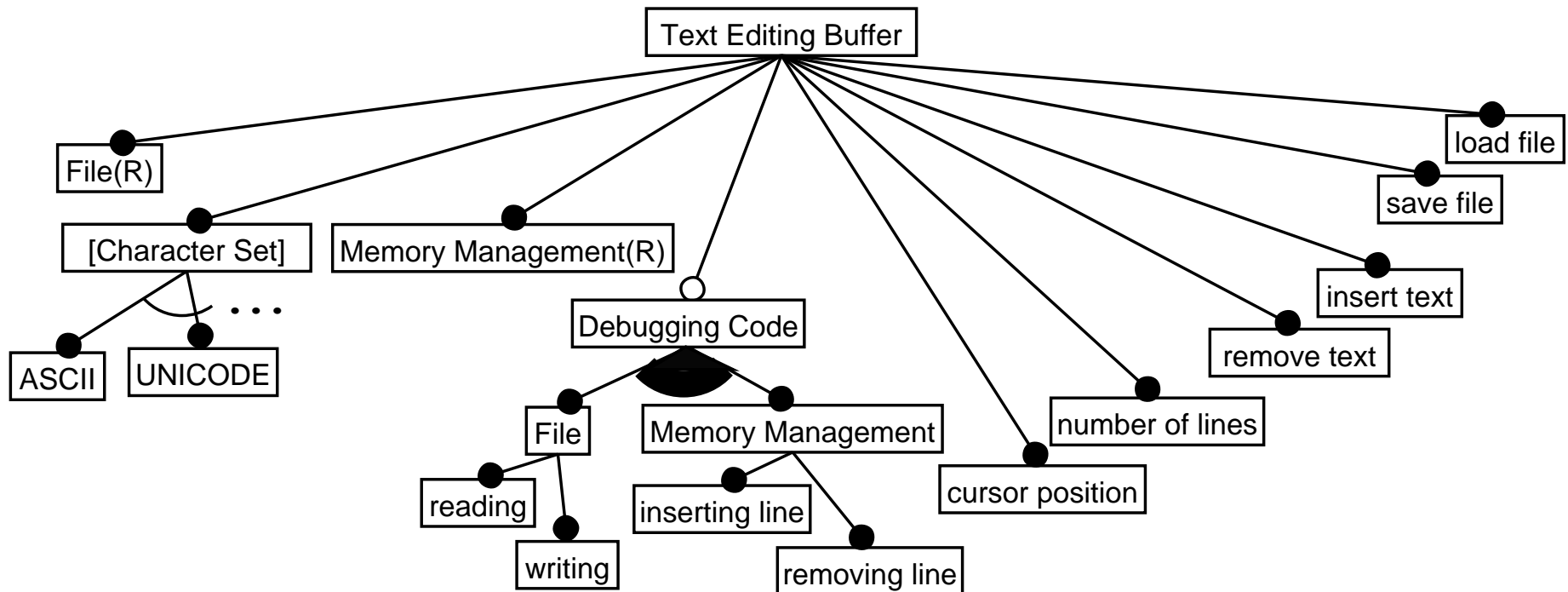
# Variabilita vlastností



## • Otvorené vlastnosti

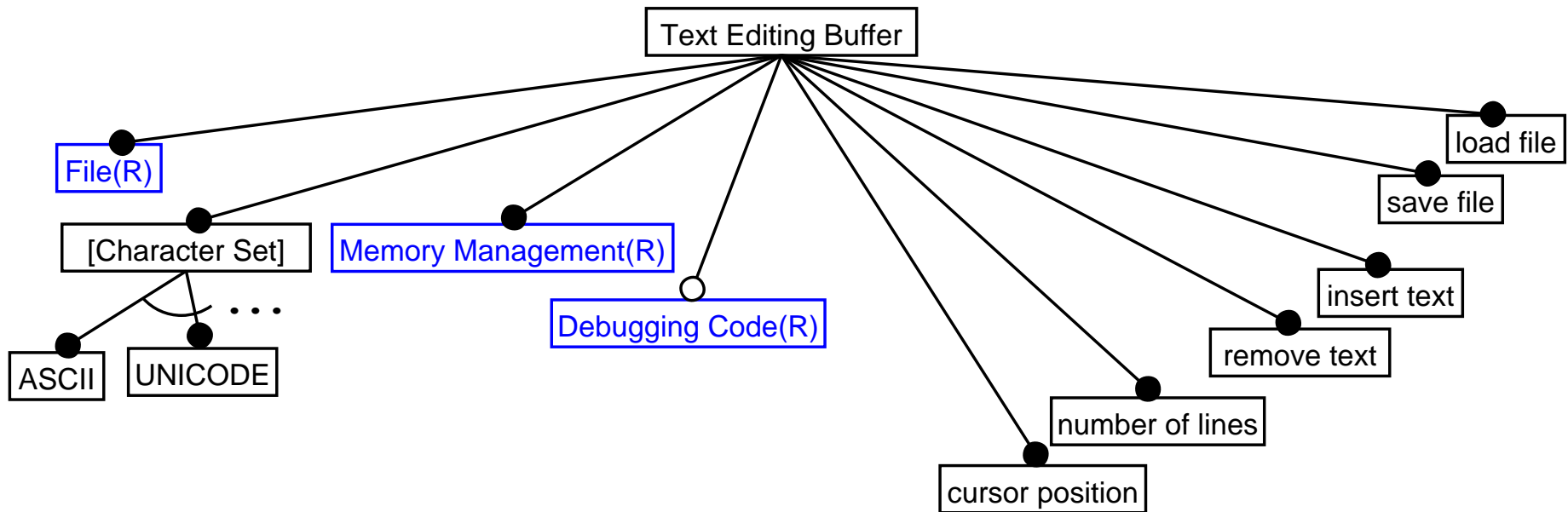
- Očakávajú sa ďalšie premenlivé podvlastnosti
- Označuje sa hranatými zátvorkami; prípadne sa upresní tromi bodkami

# Variabilita vlastností



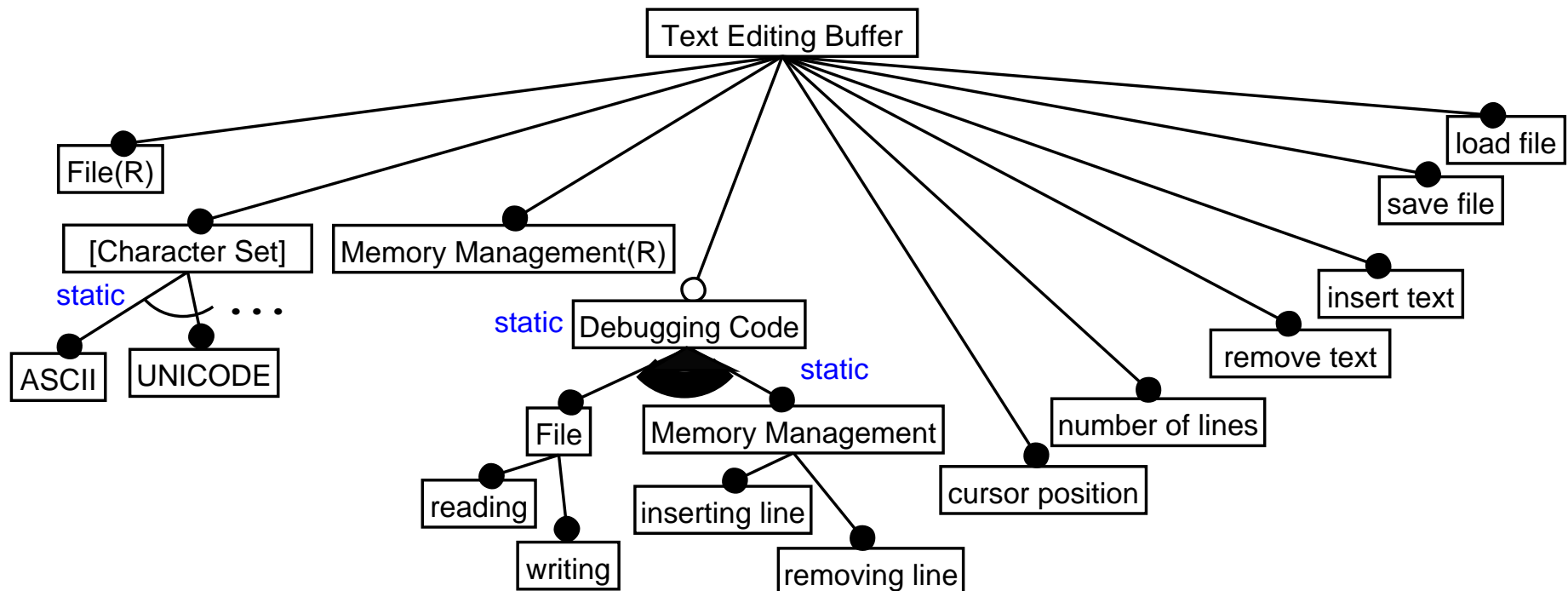
- Zahrnutie vlastnosti v inštancii konceptu je podmienené zahrnutím jej rodiča
- Vlastnosti hocijakého typu variability sa môžu vyskytovať na hocijakej úrovni

# Referencie konceptov



- Označujú sa symbolom ® (v diagramoch ako (R) z technických dôvodov)
- Môžno ich rozvinúť v prípade potreby

# Doba/mód viazania

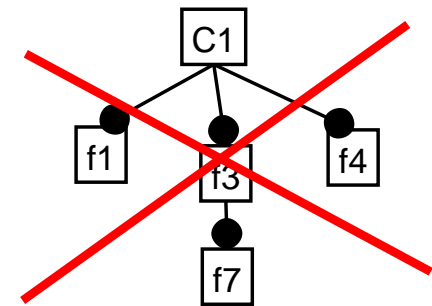
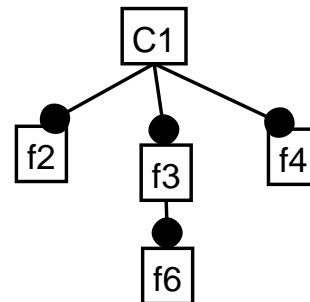
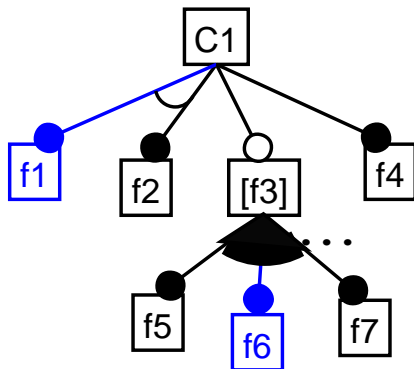


- Kedy/ako sa vlastnosť bude viazať
- Obvykle doby viazania: tvorba zdrojového kódu, preklad, spájanie, načítavanie a vykonávanie
- Mód viazania: statický alebo dynamický

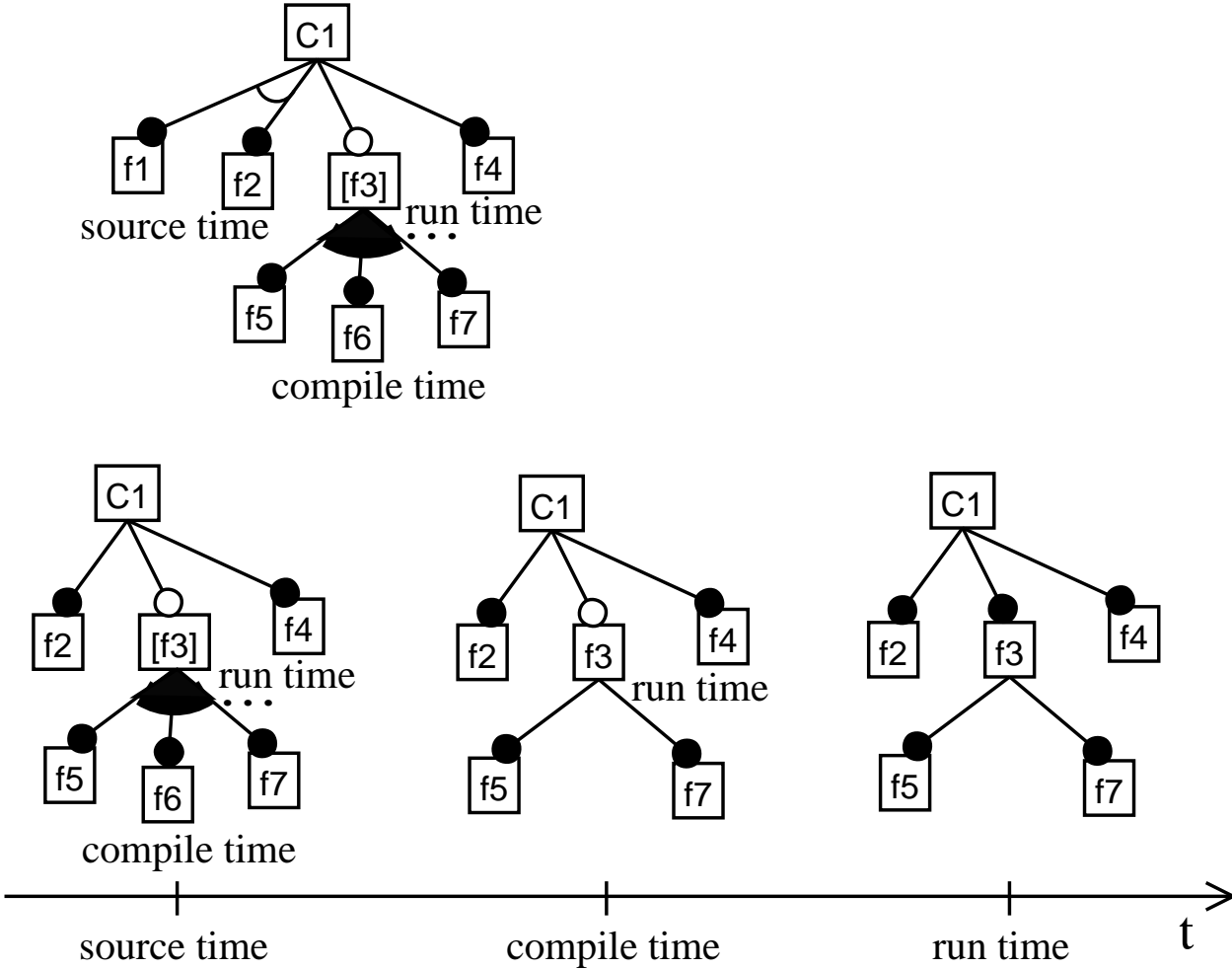
# Ďalšie informácie v modeloch vlastností

- Informácie pridružené konceptom a vlastnostiam
  - Textuálne informácie: opis, dôvod prítomnosti, dôvod zahrnutia, poznámka
  - Doba/mód viazania
- Obmedzenia a pravidlá preddefinovaných závislostí
  - Príklad obmedzenia:

$f1 \Rightarrow f6$



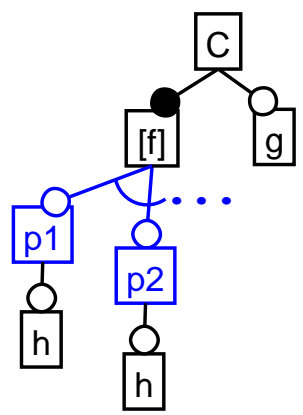
# Inštanciácia konceptov



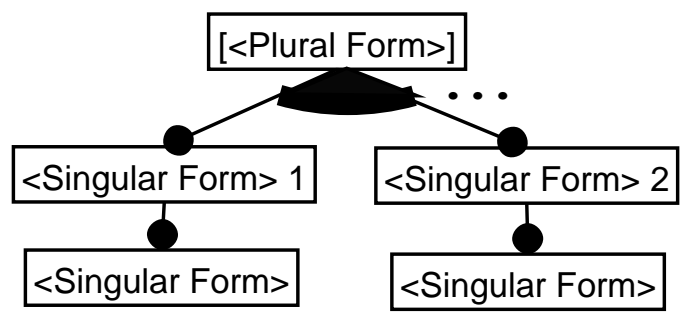
# Parametrizácia v modeloch vlastností

- Parametrizované názvy vlastností a konceptov

Obmedzenie:  $\forall \langle i \rangle \in N \ p_{\langle i \rangle}.h \underline{\vee} g$



- Parametrizované koncepty



# Softvérová podpora radov výrobkov

- Captain Feature — podporuje aj základnú Czarneckého-Eiseneckerovu notáciu, aj tú založenú na kardinalitách
- ASADAL — podporuje metódy FORM a FODA
- fmp a fmp2rsm
- pure::variants

- fmp<sup>a</sup> je plug-in do Eclipse
- Podporuje Czarneckého-Eiseneckerovu notáciu založenú na kardinalitách
- „folder view“
- Umožňuje zapísať dodatočné obmedzenia
- Umožňuje vytváranie špecializácií konfigurácií konceptov
- Validácia konfigurácie

---

<sup>a</sup><http://gp.uwaterloo.ca/fmp/>

- fmp2rsm<sup>a</sup> je ďalší plug-in do Eclipse
- Umožňuje konfigurovanie modelov v UML pomocou modelu vlastností z fmp
- Vyžaduje ďalší, komerčný plug-in Rational Software Modeler<sup>b</sup>

---

<sup>a</sup><http://gp.uwaterloo.ca/fmp2rsm/>

<sup>b</sup><http://www-128.ibm.com/developerworks/rational/products/rsm/>

# pure::variants

- Komerčný produkt na podporu vývoja radov softvérových výrobkov<sup>a</sup>
- Ako plug-in do Eclipse, ale aj ako nezávislá aplikácia
- Založený na modeloch vlastností
- Implementácia radu softvérových výrobkov je definovaná v špeciálnom modeli rodín komponentov (component family model)
- Tento model sa konfiguruje výberom vlastností z modelu vlastností

---

<sup>a</sup><http://www.pure-systems.com/>

# Zhrnutie prednášky

- Úvod do problematiky doménového inžinierstva
- Doménové inžinierstvo ako organizovaný prístup k znovupoužitiu
- Rady softvérových výrobkov a ako ich zaviesť v praxi
- Modelovanie vlastností ako technika na zachytenie variability
- Konfigurovanie ostatných modelov prostredníctvom modelu vlastností