

Tvorba RBAC modelů na základě UML diagramů

Pavel Gavlovský Martin Lasoň Miroslav Beneš

Katedra informatiky, FEI
VŠB-TU Ostrava

DATAKON 2005



Motivace

Hlavní cíl

Vytvoření základního modelu přístupových práv na základě existujících UML diagramů

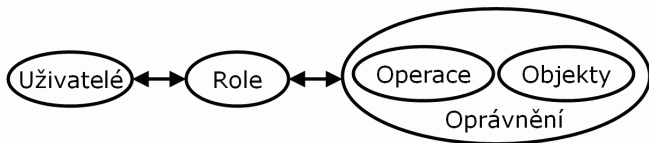
Výhody

- Úspora času
- Prevence chyb
- Snížení nákladů



Role-Based Access Control

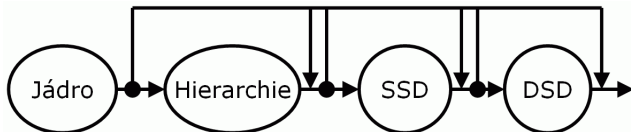
- Model omezující přístup do systému autorizovaným uživatelům
- Snižuje složitost vazeb a náklady na správu
- Práva jsou vázána na role (např. uživatelské profese)
- Flexibilita – snadná změna práv při změně profese
- Základními prvky jsou *uživatelé*, *role* a *práva* (tvořená *operacemi* nad *objekty*)



Komponenty RBAC

RBAC se skládá ze 4 funkčních komponent

- 1 Jádru RBAC modelu
- 2 Hierarchický model
- 3 Statická omezení (SSD)
- 4 Dynamický omezení (DSD)



Unified Modeling Language (UML)

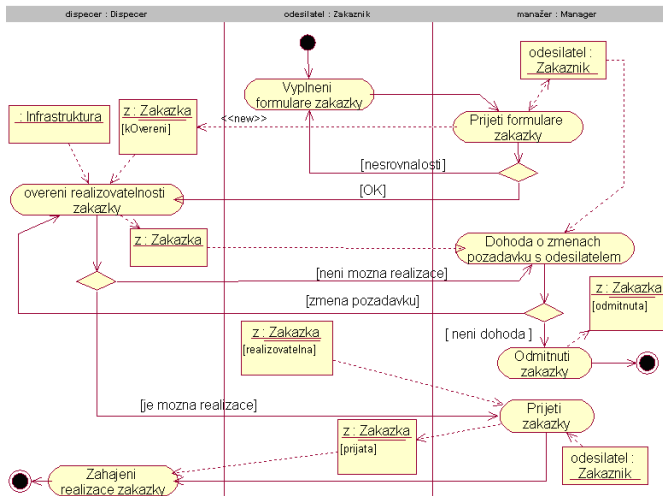
UML

Standardní grafický jazyk používaný při analýze a návrhu softwaru.

- **Diagramy případů užití** – aktéři a případy užití
- **Diagramy aktivit** – popis činností a odpovědností aktérů
- **Diagramy tříd** – vazby mezi třídami



Diagram aktivit – příklad



Definování rolí

- **Uživatelé** – při vytváření softwaru ještě nejsou známi → nevytvářejí se
- **Role** odpovídají aktérům byznys procesu
 - Diagramy aktivit – dráhy odpovědnosti
Sdružení akcí vykonávaných jednou entitou
 - Případy užití – aktéři



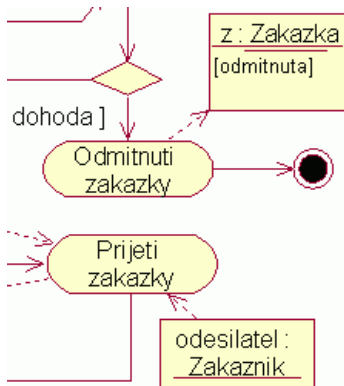
Definování oprávnění – objekty

- *Oprávnění* – operace nad objekty
- *Objekt* – zdroj dat dostupný v systému
Objekt v akt. diagramu → RBAC objekt
- *Operace* reprezentována objektovým tokem
Objektový tok → operace nad objektem
- Oprávnění jsou přiřazena aktérovi (roli) v jehož dráze odpovědnosti se nachází akce pracující s objektem



Definování oprávnění – objekty

Diagram aktivit – operace nad objekty



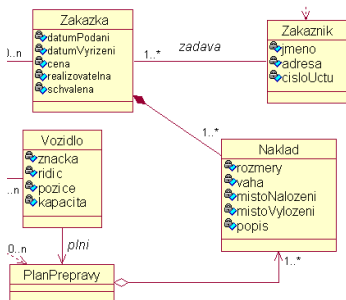
Operace

- **čtení** – tok z objektu do akce
- **zápis** – tok z akce do objektu
- **vytvoření** – stereotyp «new»
- **zrušení** – stereotyp «destroy»



Definování oprávnění – objekty

Třídní diagram – vztahy mezi objekty



Nastavování práv

- **kompozice** – práva nastavena všem částem
- **agregace** – částem nastaven *kandidát na právo*



Definování oprávnění – operace

Diagram aktivit – provádění akcí

- Nastavení *oprávnění* k provádění akcí
- Oprávnění jsou přiřazena aktérům (rolím) podle:

Dráhy zodpovědnosti

Oprávnění provést vlastní akce

Řídící toky

Kandidát na oprávnění k vyvolání akce jiného aktéra

- Zdroj toku – vlastník práva
- Rozdělení toků → vytvoření dalších kandidátů na právo
- Identifikace návratových hodnot

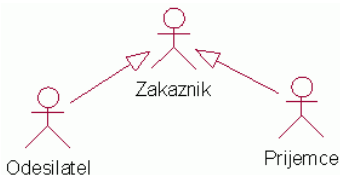


Definování hierarchie rolí

Definice

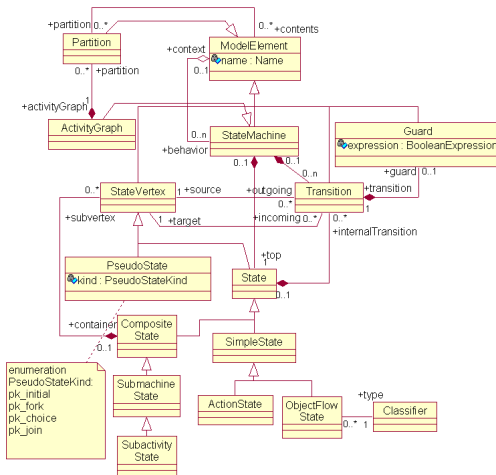
Jedna role dědí z druhé role pokud všechna práva druhé role jsou také právy první role a pokud všichni uživatelé první role jsou také uživateli druhé role.

- Hierarchie generována z diagramů případů užití
- Dědičnost mezi aktéry → dědičnost mezi rolami



Reprezentace diagramů

Podmnožina metamodelu UML

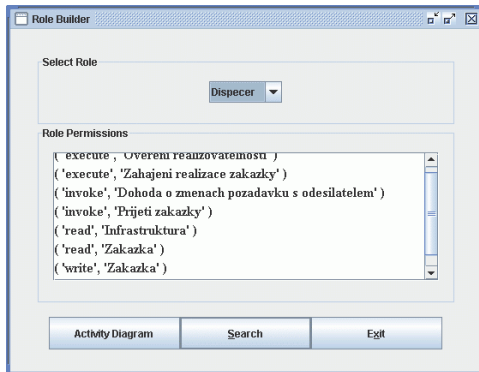


Algoritmus

- 1 Identifikace rolí na základě existujících aktérů (objekty `Partition`)
- 2 Identifikace práv k vyvolání akcí (`ActionState`)
 - vyvolání vlastních akcí (*execute*)
 - vyvolání akce jiného aktéra (*invoke*)
- 3 Identifikace práv k operacím nad objekty (`ObjectFlowState`)
 - kontrola směru toku (*read, write*)
 - kontrola stereotypu (*create, delete*)



Výstup



Závěr

- Vytvořen postup, jak definovat základní role a přiřadit těmto rolím přístupová práva.
- Využívá existujících UML diagramů
- Urychluje proces vytváření rolí a práv
- Předchází lidským chybám
- Automatizace celého procesu

