Sem vložte zadání Vaší práce.

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra softwarového inženýrství

Diplomová práce

Analýza a vizualizace závislostí uživatelských rolí v IDM ČVUT

Bc. Jan Matys

Vedoucí práce: Ing. Michal Valenta, Ph.D.

30. června 2015

Poděkování

Rád bych poděkoval vedoucímu diplomové práce, Ing. Michalu Valentovi, Ph.D., za jeho rady a trpělivost, kterou měl v celém průběhu zpracování této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 30. června 2015 .....

České vysoké učení technické v Praze
Fakulta informačních technologií
© 2015 Jan Matys. Všechna práva vyhrazena.
Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Matys, Jan. *Analýza a vizualizace závislostí uživatelských rolí v IDM ČVUT*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato práce obsahuje analýzu rolí IDM ČVUT. Dále popisuje návrh, implementaci, nasazení a testování aplikace pro vizualizaci a kontrolu rolí, která byla úspěšně nasazena ve Výpočetním informačním centru ČVUT.

Klíčová slova IDM, analýza rolí, Neo4j, vizualizace grafu

Abstract

This thesis contains roles analysis of CTU's IDM. Further, it describes design, implementation, deployment and testing of an application for roles analysis and roles check which was successfully deployed in Computing and Information Centre of CTU.

Keywords IDM, roles analysis, Neo4j, graph visualization

Obsah

Seznam obrázků

Seznam tabulek

Úvod

České vysoké učení technické v Praze (ČVUT) je velká organizace, která má několik tisíc zaměstnanců a několik desítek tisíc studentů. Téměř každá z těchto osob má uživatelský účet v některém informačním systému ČVUT. Ke správě svých informačních systémů používá ČVUT Identity Manager (IDM). IDM umožňuje centrální řízení uživatelských oprávnění, přiřazených systémů, pomocí uživatelských rolí. Z důvodu velkého počtu zmíněných osob a příslušných uživatelských účtů je potřeba spravovat přes 30 000 rolí. Protože tyto systémy obsahují často velmi citlivé údaje, je nutné, aby v rolích bylo minimální množství chyb.

Cílem této práce je analyzovat role z IDM, a poté navrhnout, implementovat, nasadit a pilotně otestovat aplikaci, která bude:

1. na základě vstupu popisujícího vztahy mezi rolemi tyto vztahy dynamicky zobrazovat a filtrovat dle požadavků uživatele

2. vyhledávat nesystematičnosti (výjimky z pravidel, chyby, zacyklení apod.) mezi závislostmi rolí

Přínosem této aplikace by měla být možnost získání lepšího přehledu o vztazích mezi rolemi a snížení počtu chyb v rolích.

Tato práce začíná kapitolou popisující technologie, jejichž použití plyne ze zadání. Tedy technologie IDM a grafová databáze Neo4j. Další kapitola se věnuje analýze IDM ČVUT a rolí. Ostatní kapitoly se týkají návrhu, implementace, nasazení a testování výsledné aplikace.

Kapitola 1

Zadané technologie a související pojmy

Tato kapitola popisuje technologie, jejichž použití plyne ze zadání. První technologií, která je zdrojem rolí, je IDM. Druhá technologie je grafová databáze Neo4j, která bude sloužit k uložení rolí v grafu. Dále budou popsány některé související pojmy s těmito technologiemi.

1.1 IDM

IDM (Identity Manager), do češtiny lze přeložit jako správce identit, je informační systém, který spravuje identity přiřazených systémů. IDM obsahuje:

- správu identit
- řízení přístupu

Než bude možné blíže vysvětlit správu identit a řízení přístupu je třeba vysvětlit pojmy entita, identita a uživatelský účet.

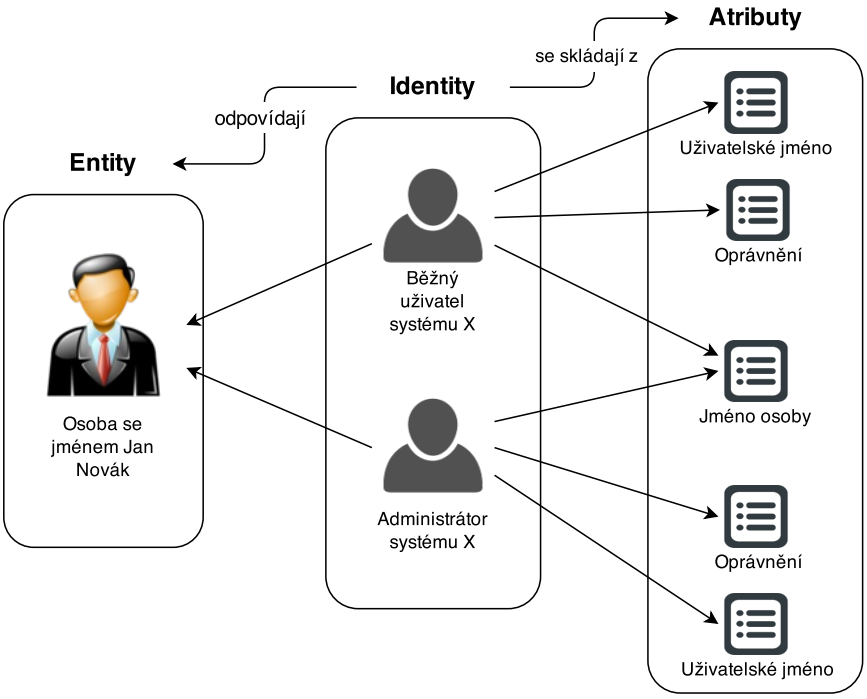
1.1.1 Entita

Entita má význam reálné osoby, firmy nebo věci. Entita je to, co zavádíme do IDM pomocí identit.

1.1.2 Identita

Identita je atribut nebo sada atributů nějaké entity, které ji jednoznačně určují v rámci systému [2]. Na obrázku 1.1 je znázorněn koncept identit. Entita, v tomto případě osoba se jménem Jan Novák, může mít velké množství identit, a to nejen v různých systémech, ale i v rámci jednoho systému. V jedné době může osoba Jan Novák v systému X vystupovat pod identitou běžného

uživatele a v druhé době pod identitou administrátora. Přestože reálná osoba je stále jedna, pod identitou běžného uživatele nemá Jan Novák oprávnění, které by měl pod identitou administrátora.



Obrázek 1.1: Koncept identit

1.1.3 Uživatelský účet

V informačním systému je identita většinou reprezentována pomocí uživatelského účtu. Uživatelský účet může obsahovat sadu různých atributů, které se týkají dané entity (jméno, věk) nebo uživatele informačního systému (heslo, přístupová oprávnění, datum vytvoření účtu). Nejdůležitějším atributem uživatelského účtu je uživatelské jméno, které jednoznačně identifikuje uživatele v rámci celého informačního systému.

1.1.4 Správa identit

Správa identit zahrnuje tři hlavní operace:

- vytvoření identity

- aktualizace identity
- smazání identity

Pokud chceme zavést entitu do systému, tak ji musíme vytvořit identitu. Tato identita může být založena na reálných atributech (jméno a příjmení osoby) nebo na fiktivních atributech (vygenerované unikátní (v rámci systému) osmimístné celé číslo). Při změně reálných atributů nebo při určitých jiných událostech je potřeba provést aktualizaci identity. Pokud je potřeba odebrat entitu ze systému, tak je třeba provést smazání identity.

1.1.5 Řízení přístupu

Řízení přístupu zahrnuje autentizaci a téměř vždy i autorizaci. Autorizace musí vždy následovat až po autentizaci.

Autentizace je proces ověření identity dané entity. Obecný postup je následující:

1. Entita prohlásí, že má identitu X.
2. Systém požádá o důkaz.
3. Entita prokáže svojí identitu, např. pomocí hesla nebo čipu.

Autorizace je proces ověření, zda má daná entita přístup k určité operaci. Obecný postup je následující:

1. Entita s identitou X požádá o přístup k operaci Y.
2. Systém ověří, jestli entita s identitou X má právo na operaci Y.
3. Pokud systém rozhodne, že entita s identitou X má právo na operaci Y, pak systém vydá povolení a entita s identitou X vykoná operaci Y. Pokud systém rozhodne, že entita s identitou X nemá právo na operaci Y, pak systém neumožní provést entitě s identitou X operaci Y.

1.1.6 RBAC

Velmi rozšířeným principem řízení přístupu v IDM je RBAC (Role-Based Access Control), což lze do češtiny přeložit jako řízení přístupu na základě rolí. RBAC využívá opakovaně přidělitelné objekty – **role** [3]. Uživateli se tedy nepřidělí konkrétní oprávnění, ale role, která v sobě zahrnuje sadu konkrétních oprávněních.

Role, resp. její oprávnění, by ideálně měla představovat reálnou roli ve firmě, resp. její reálná oprávnění. Například účetní ve firmě zapisuje a tiskne

faktury, takže se vytvoří role s názvem „účetní“, která obsahuje práva vytvářet a tisknout faktury v účetním informačním systému. Tato role se pak přidělí k uživatelskému účtu účetního.

Role může obsahovat, kromě přístupových oprávnění, mnoho dalších atributů. Velmi často používanými atributy jsou:

Schvalovatel Schvaluje přiřazení role k uživatelskému účtu.

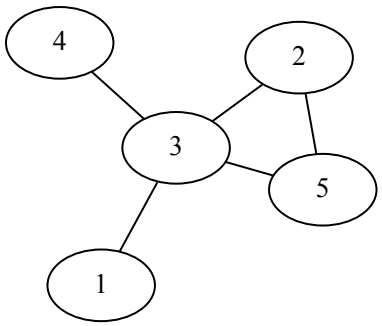
Vlastník Uživatel, který vytvořil příslušnou roli a většinou jedině on má právo měnit její přístupová práva.

1.2 Graf

V této sekci budou uvedeny definice neorientovaného a orientovaného grafu a několika dalších pojmů z teorie grafů.

1.2.1 Neorientovaný graf

Neorientovaný graf je uspořádaná trojice $G = \langle H, U, \varrho \rangle$, kde H je množina **hran** grafu G, U je množina **uzlů** grafu G disjunktní k H a ϱ je zobrazení $H \longrightarrow U \otimes U$ (neuspořádaná dvojice uzlů), které nazýváme incidence grafu G [4]. Příklad neorientovaného grafu lze vidět na obrázku 1.2.

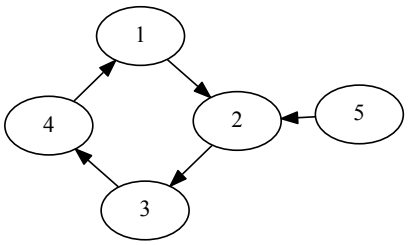


Obrázek 1.2: Neorientovaný graf

1.2.2 Orientovaný graf

Orientovaný graf je uspořádaná trojice $G = \langle H, U, \sigma \rangle$, kde H je množina **orientovaných hran** grafu G, U je množina **uzlů** grafu G disjunktní k H

a σ je zobrazení $H \longrightarrow U \times U$ (uspořádaná dvojice uzlů), které nazýváme incidence grafu G. Příklad orientovaného grafu lze vidět na obrázku 1.3.



Obrázek 1.3: Orientovaný graf

1.2.2.1 Spojení

Pokud pro dvojici uzlů u_{poc} a u_{kon} orientovaného grafu $G = \langle H, U, \sigma \rangle$ existuje posloupnost uzlů a hran

$$S = \langle u_0, h_1, u_1, h_2, \ldots, u_{n-1}, h_n, u_n \rangle$$

, pro kterou platí $\sigma(h_i) = (u_{i-1}, u_i)$, kde $h_i \in H$ a $i \in \{1, 2, \ldots, n\}$, $u_i \in U$, kde $i \in \{0, 1, 2, \ldots, n\}$, $u_0 = u_{poc}$ a $u_n = u_{kon}$, pak tuto posloupnost nazýváme **spojením** grafu G. Uzel u_{poc} je počátečním uzlem a u_{kon} je koncovým uzlem spojení S.

1.2.2.2 Orientovaný tah

Orientovaný tah orientovaného grafu $G = \langle H, U, \sigma \rangle$ je takové spojení

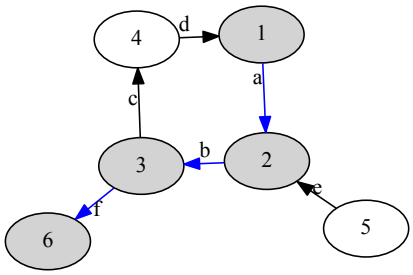
$$S = \langle u_0, h_1, u_1, h_2, \ldots, u_{n-1}, h_n, u_n \rangle$$

, ve kterém pro všechny $i, j \in \{1, 2, \ldots, n\}$ platí, že pokud $i \neq j$, pak $h_i \neq h_j$. Tedy všechny hrany ve spojení S musí být různé.

1.2.2.3 Orientovaná cesta

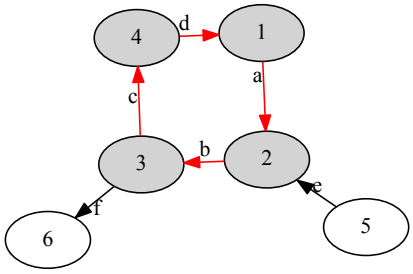
Orientovaná cesta je orientovaný tah T, ve kterém každý uzel inciduje maximálně se dvěma hranami tahu T.

Otevřená orientovaná cesta Otevřená orientovaná cesta je orientovaná cesta, která nemá stejný počáteční a koncový uzel nebo má jen jeden uzel. Tedy $\langle u_0, h_1, u_1, h_2, \ldots, u_{n-1}, h_n, u_n \rangle$, kde $n \geq 1$ a $u_0 \neq u_n$, nebo $\langle u_0 \rangle$. Příklad otevřené orientované cesty, která je tvořena posloupností $\langle 1, a, 2, b, 3, f, 6 \rangle$, lze vidět na obrázku 1.4.



Obrázek 1.4: Otevřená orientovaná cesta

Uzavřená orientovaná cesta Uzavřená orientovaná cesta je orientovaná cesta $\langle u_0, h_1, u_1, h_2, \ldots, u_{n-1}, h_n, u_n \rangle$, pro kterou platí $n \geq 1$ a $u_0 = u_n$. Příklad uzavřené orientované cesty, která je tvořena posloupností $\langle 1, a, 2, b, 3, c, 4, d, 1 \rangle$, lze vidět na obrázku 1.5.



Obrázek 1.5: Uzavřená orientovaná cesta

1.2.2.4 Cyklus

Cyklus je uzavřená orientovaná cesta.

1.2.3 Podgraf

Nechť existují grafy $G = \langle H, U, \sigma \rangle$ a $G' = \langle H', U', \sigma' \rangle$. Pokud platí

$$(H' \subseteq H) \wedge (U' \subseteq U) \wedge (\forall h \in H'.\ \sigma'(h) = \sigma(h))$$

, pak graf G' nazýváme **podgraf** grafu G.

1.3 Neo4j

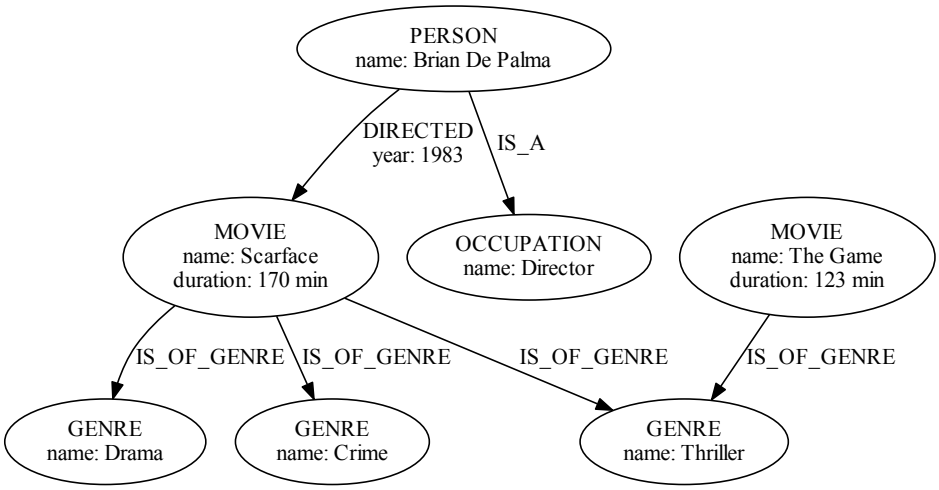
Neo4j je grafová databáze implementována v programovacích jazycích Java a Scala a běžící v JVM (Java Virtual Machine).

Neo4j patří mezi NoSQL databáze [5]. NoSQL (Not only SQL), do češtiny lze přeložit jako nejenom SQL (Standard Query Language), je databázový koncept nepoužívající relační model. Relační model je základem relačních databází, používajících jako dotazovací jazyk SQL, které jsou už řadu let nejpoužívanějšími databázemi [6]. Označení NoSQL vyjadřuje, že pro některá data existují a jsou vhodnější jiné databáze než relační.

1.3.1 Labeled Property Graph

Neo4j používá datový model s názvem Labeled Property Graph [7]. Labeled Property Graph je orientovaný graf (viz sekce 1.2.2), kde uzel i hrana mají své označení (label) a můžou obsahovat libovolné množství vlastností (properties). Takováto hrana se v kontextu Neo4j nazývá vztah, a proto se v dalším textu, v souvislosti s Neo4j databází, bude používat termín vztah. Vlastnost má jedinečný klíč v rámci daného uzlu, resp. vztahu, a hodnotu. Vlastnost s klíčem propertyKey je nazývána vlastnost *propertyKey*.

Příklad modelu Labeled Property Graph je na obrázku 1.6. Na obrázku je graf, který modeluje filmovou databázi. Nejvíce nahoře lze vidět uzel s označením „PERSON“, který má vlastnost *name* s hodnotou „Brian De Palma“. Z tohoto uzlu vede vztah s označením „DIRECTED“, který má vlastnost *year* s hodnotou „1983“, do uzlu s označením „MOVIE“, který má vlastnost *name* s hodnotou „Scarface“ a vlastnost *duration* s hodnotou „170 min“. Výše zmíněné dva uzly a jeden vztah mají následující význam: osoba se jménem Brian De Palma zrežírovala v roce 1983 film s názvem Scarface, jehož délka je 170 minut.



Obrázek 1.6: Labeled Property Graph – příklad

1.3.2 Transakce v Neo4j

V Neo4j musí být všechny operace, které přistupují ke grafu, indexům nebo schématu, prováděny uvnitř transakce [8]. Pokud některá z těchto operací není uvnitř transakce, pak je vyhozena výjimka *NotInTransactionException*.

Neo4j transakce splňuje ACID vlastnosti:

Atomicita (Atomicity) Transakce je atomická, buď se provede celá nebo vůbec. [9]

Konzistence (Consistency) Transakce převede databázi z jednoho konzistentního stavu do druhého.

Izolovanost (Isolation) Transakce nevidí provedené změny ostatních nedokončených transakcí.

Trvanlivost (Durability) Změny, provedené úspěšně dokončenou transakcí, jsou uloženy v databázi trvale (i v případě selhání systému).

1.3.3 Server a embedded mód

Neo4j může pracovat ve dvou různých módech: server a embedded.

1.3.3.1 Server mód

Server mód znamená, že je spuštěn Neo4j server nad konkrétní databází, který přijímá požadavky od klientských aplikací. Požadavky, i odpovědi, jsou posílány přes síť pomocí REST (Representational State Transfer) API (Application Programming Interface). Požadavky mohou být zpracovávány paralelně, ale jediný, kdo přistupuje přímo k databázi (databázovým souborům) je Neo4j server.

Výhodou server módu oproti embedded módu je možnost paralelního zpracování požadavků, nevýhodou je horší výkon kvůli přidané komunikaci po síti.

1.3.3.2 Embedded mód

Embedded mód znamená, že aplikace přistupuje ke konkrétní databázi přímo přes Neo4j API (viz [8]). Aplikace zamkne databázi provede potřebné transakce, a poté databázi odemkne. V době, kdy je zamčená databáze, nemůže jiná aplikace přistupovat k databázi. Tímto může vzniknout zbytečné zdržení, když první aplikace zamkne databázi a čeká dlouhou dobu na některá vstupní data a další aplikace už má vše připraveno, ale stejně musí čekat na uvolnění zámku.

Výhodou embedded módu oproti server módu je vyšší výkon (nemusí se komunikovat po síti), nevýhodou je, že není možné paralelně zpracovávat požadavky od více aplikací.

Níže je uvedena zdrojový kód v programovacím jazyku Java, který provede spuštění databáze (včetně implicitního zamčení databáze), provedení transakce a nakonec vypnutí databáze (včetně implicitního odemčení databáze).

```
1  String dbPath = "C:\\Neo4j\\default.graphdb";
2  GraphDatabaseService graphDatabaseService =
3      new GraphDatabaseFactory().newEmbeddedDatabase(dbPath);
4
5  try (Transaction tx = graphDatabaseService.beginTx()) {
6      Label label = DynamicLabel.label("PERSON");
7      Node node = graphDatabaseService.createNode(label);
8      node.setProperty("name", "Brian De Palma");
9
10     tx.success();
11 }
12
13 graphDatabaseService.shutdown();
```

1.3.4 Cypher

Cypher je deklarativní grafový dotazovací jazyk inspirovaný SQL [10]. Struktura jazyka Cypher je stejná jako struktura jazyka SQL. Dotaz je sestaven

z různých klauzulí, které si mezi sebou mohou předávat výsledné množiny dat. Cypher také podporuje všechny CRUD operace:

Vytvoření (Create) pomocí klauzule *CREATE* (podobné jako *INSERT* v SQL)

Čtení (Read) pomocí klauzule *MATCH*.

Aktualizace (Update) pomocí klauzule *SET* (podobné jako *SET* v SQL)

Vymazání (Delete) pomocí klauzule *DELETE* (podobné jako *DELETE* v SQL)

1.3.4.1 Vzor

Pomocí vzoru (pattern) se v jazyku Cypher popisuje hledaný (požadováný) podgraf. Následující text popisuje syntaxi vzoru.

Uzel Uzel se zapisuje pomocí dvou závorek takto:

```
()
```

Identifikátor nalezeného uzlu, pomocí nějž se můžeme na daný uzel odkazovat v jiných částech dotazu, lze vytvořit přidáním písmene n mezi závorky následujícím způsobem:

```
(n)
```

Písmeno n se tedy stane **odkazem** na daný uzel.

Vztah Vztah mezi dvěma uzly se zapisuje pomocí šipky takto:

```
()-->()
```

Odkazy na oba nalezené uzly i vztah vytvoříme následovně:

```
(n)-[r]->(m)
```

Pokud nezáleží na směru vztahu, ale chce se pouze vyjádřit vztah mezi dvěma uzly, pak je možný tento zápis:

```
()--()
```

Cesta Cestu mezi dvěma uzly lze zapsat takto:

```
(n)-[*]->(m)
```

Tento zápis vyjadřuje, že počáteční uzel má identifikátor n, koncový uzel má identifikátor m a počet uzlů a vztahů v cestě mezi těmito dvěma uzly je neomezený.

Označení Uzel s označením „PERSON“ lze zapsat takto:

```
(n:PERSON)
```

Obdobně se zapíše vztah s označením „DIRECTED“:

```
(n:PERSON)-[r:DIRECTED]->(m:MOVIE)
```

Vlastnost Vlastnost uzlu s klíčem *name* a hodnotou „Brian De Palma“ lze zapsat takto:

```
(n:PERSON {name: "Brian De Palma"})
```

Obdobně se zapíše vlastnost vztahu s klíčem *year* a hodnotou „1983“:

```
(n:PERSON)-[r:DIRECTED {year: "1983"}]->(m:MOVIE)
```

1.3.4.2 Klauzule

V této sekci budou popsány základní klauzule jazyka Cypher a uvedeno několik příkladů jejich použití. Dotazy jsou určeny pro graf, který je znázorněn na obrázku 1.6.

CREATE Klauzule *CREATE* slouží k vytvoření uzlů a vztahů. Vytvoření uzlu s označením „PERSON“ a vlastností *name* s hodnotou „David Fincher“ lze provést následujícím dotazem:

```
CREATE (n:PERSON {name: "David Fincher"})
```

MATCH Klauzule *MATCH* umožňuje definovat vzor, který se má vyhledat. Vyhledání všech filmů, jejichž žánr je drama (tedy uzlů s označením „MOVIE“, které mají vztah, s označením „IS_OF_GENRE“, s uzlem s označením „GENRE“ a vlastností *name* s hodnotou „Drama“), lze provést následujícím dotazem:

```
MATCH (n:MOVIE)-[r:IS_OF_GENRE]->(m:GENRE {name: "Drama"})
RETURN n
```

RETURN Klauzule *RETURN* definuje, které části vzoru mají být dotazem vráceny. U příkladu u klauzule *MATCH* může být požadavek na vrácení jen názvů filmů, jejichž žánr je drama. Tedy chceme znát jen hodnoty vlastností *name* u vybraných uzlů s označením „MOVIE“. Tento požadavek lze provést následujícím dotazem:

```
MATCH (n:MOVIE)-[r:IS_OF_GENRE]->(m:GENRE {name: "Drama"})
RETURN n.name
```

WHERE Klauzule *WHERE* přidává další omezení na hledaná data. Délku filmu s názvem Scarface lze zjistit následujícím dotazem:

```
MATCH (n:MOVIE)
WHERE n.name = "Scarface"
RETURN n.duration
```

DELETE Klauzule *DELETE* slouží k vymazání uzlů a vztahů (k vymazání označeních a vlastností slouží klauzule *REMOVE*). Vymazání vztahu mezi filmem Scarface a žánrem thriller (tedy vztahu, s označením „IS_OF_GENRE“, mezi uzlem, s označením „MOVIE“ a vlastností *name* s hodnotou „Scarface“, a uzlem, s označením „GENRE“ a vlastností *name* s hodnotou „Thriller“) lze provést následujícím dotazem:

```
MATCH (n:MOVIE {name: "Scarface"})-[r:IS_OF_GENRE]->(m:GENRE {
    name: "Thriller"})
DELETE r
```

Kapitola 2

Analýza IDM ČVUT a rolí

Tato kapitola obsahuje popis systému IDM ČVUT a rolí. V první sekci je stručný úvod do systému IDM a jeho vlastností. Následující sekce už obsahují především popis rolí, konkrétně popis dat obsažených v CSV souboru rolí, atributů role, doménového modelu role a definice chyb a anomálií v rolích.

2.1 IDM

ČVUT používá IDM s názvem Oracle Waveset, verze 8.1.1, který je pod správou VIC (Výpočetní informační centrum) ČVUT. Dříve se tento systém jmenoval Sun Identity Manager a v některých dokumentech k systému Oracle Waveset lze tento název ještě najít.

2.1.1 Uživatelské rozhraní

IDM má dvě hlavní uživatelská rozhraní. Jedno je pro administrátory a druhé pro běžné uživatele. Obě uživatelská rozhraní jsou webová. Administrátoři používají administrátorské rozhraní pro:

- správu uživatelů
- nastavení a přiřazení zdrojů (IT systém spravovaný IDM)
- definici přístupových oprávnění
- stanovení auditních pravidel
- další administrátorské funkce

Běžní uživatelé používají rozhraní pro běžné uživatele pro:

- změnu hesla
- nastavení odpovědí k autentizačním otázkám

- požádání přístupu ke zdroji
- správu jim přidělených úkolů (např. schválení přiřazení role k uživateli)

2.1.2 RBAC

IDM používá při řízení přístupu RBAC (viz sekce 1.1.6). Pro větší přehlednost a lepší kontrolu jsou role rozdělené do skupin na základě typu role. Typy rolí jsou následující:

Byznys role Typicky reprezentuje pracovní funkci v organizaci. Uživatel, kterému je přiřazena role tohoto typu, získá přístupová práva potřebná k vykonávání svých pracovních povinností.

Technická role Tento typ role je přiřazován k některé byznys roli, aby uživatel, který má přiřazenou příslušnou byznys roli, získal přístupová práva ke zdroji, např. studijní informační systém KOS, který potřebuje pro svou práci.

2.1.3 Úprava rolí

IDM má všechny role uložené v databázi Oracle Database. Provádět operace s rolemi je možné buď přímo přes uživatelské rozhraní IDM nebo exportováním rolí do CSV (Comma-Separated Values) souboru, upravením CSV souboru a importováním rolí do Oracle Waveset (naimportováním dojde k aktualizaci rolí v databázi podle rolí z CSV souboru).

Metoda úpravy přes uživatelské rozhraní se používá hlavně při potřebě upravit jen jednu roli. Výhodou této metody je, že uživatelské rozhraní pomáhá uživateli se správným vyplněním, tudíž při této metodě vzniká malé množství chyb. Nevýhodou je, že je možné najednou upravit jen jednu roli.

Metoda úpravy přes export a import rolí se používá hlavně při potřebě upravit více rolí. Výhodou této metody je, že je možné upravit více rolí najednou. Nevýhodou je, že při této metodě vzniká poměrně velké množství chyb.

2.2 CSV soubor rolí

V následujících odstavcích je uveden základní popis obsažených dat v exportovaném CSV souboru rolí z IDM.

Každý neprázdný řádek, kromě hlavičky, CSV souboru rolí je popis jedné role. Jednotlivé sloupce těchto řádků jsou hodnoty atributů příslušných rolí. Jednotlivé sloupce v hlavičce označují názvy atributů.

Uveďme si příklad zjednodušeného CSV souboru se 4 rolemi (tyto 4 role se všemi 36 atributy lze najít v souboru `sample_roles-version_2015_03_18.csv`, viz příloha C):

```
name,type,containedRoles,containedRolesAssociationType,
    approverRoles,cvutType
B-00000-TEST-CLEN,BusinessRole,T-SFIS-12361-TEST-CONTROLLING,
    required,B-00000-TEST-PREDSEDA,160
B-00000-TEST-PREDSEDA,BusinessRole,B-00000-TEST-CLEN,required
    ,,170
T-SFIS-12360-TEST-VEDOUCI,ITRole,,,"B-00000-TEST-PREDSEDA,B
    -00000-TEST-CLEN",1350
T-SFIS-12361-TEST-CONTROLLING,ITRole,,,"B-00000-TEST-PREDSEDA,B
    -00000-TEST-CLEN",1380
```

Jak můžeme vidět první role má hodnotu atributu *name* „B-00000-TEST-CLEN“, atributu *type* „BusinessRole“ a atributu *containedRoles* „T-SFIS-12361-TEST-CONTROLLING“. Z toho vyplývá, že první role má název „B-00000-TEST-CLEN“, je typu *Business* a jsou jí přiřazena určitá práva do systému SFIS pomocí role „T-SFIS-12361-TEST-CONTROLLING“.

2.3 Atributy role

Tato sekce je výsledkem analýzy atributů role. Jsou zde popsány všechny atributy role, které se vyskytují v CSV souboru rolí z IDM k datu 18. 3. 2015.

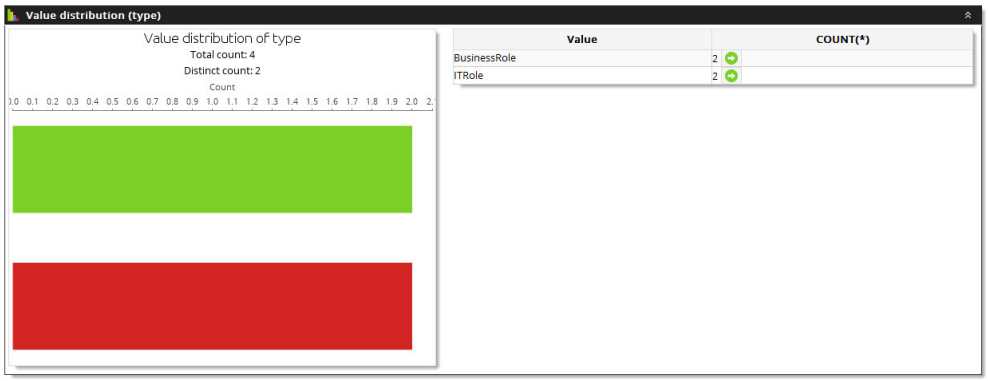
Nejdříve v několika následujících odstavcích budou uvedeny zdroje informací k popisu atributů a struktura popisu atributu. Poté bude v každé další podsekci popsán jeden atribut.

Zdroje informací k popisu atributů Zdroje informací k popisu atributů jsou následující:

- dokumentace IDM, [11] a [12]
- profilování dat CSV souboru `roles-2015_03_18.csv`, který obsahuje 40 332 rolí
  - K profilování dat byla použita aplikace DataCleaner (verze 4.0, Professional edition). Typ analýzy má název „Value distribution“ (rozdělení hodnot).
  - CSV soubor `roles-2015_03_18.csv` obsahuje všechny role z IDM ČVUT, tedy i hodně citlivých informací. Hlavně z bezpečnostních důvodů nemůže být tento soubor veřejně dostupný, a proto není ani na přiloženém CD. Na přiloženém CD není ani výsledek profilování dat z aplikace DataCleaner, protože ze zmíněného výsledku lze získat všechna data, která obsahuje CSV soubor rolí. Aby měl čtenář alespoň představu jak vypadá výsledek profilování dat CSV souboru rolí z aplikace DataCleaner, tak v souboru `datacleaner-sample_roles-version_2015_03_18-value_distribution.html` lze

najít výsledek profilování dat souboru `sample_roles-version_2015_03_18.csv`, který obsahuje několik testovacích rolí velmi podobných rolím z CSV souboru `roles-2015_03_18.csv`, viz příloha C. Výsledek profilování dat pro atribut *type* je vidět na obrázku 2.1.

- Ing. Jan Scherks, správce IDM a zaměstnanec VIC ČVUT



Obrázek 2.1: Výsledek profilování dat pro atribut *type*, analýza typu „Value distribution“ (rozdělení hodnot)

Struktura popisu atributu

- název atributu, který nedodržuje velikost prvního počátečního písmene („přesný název atributu“)
  - popis atributu
  - povolená hodnota atributu definována Java regulárním výrazem (java.util.regex, Java SE 7, viz [13]), řetězec „// ???“ znamená, že definici povolené hodnoty atributu se nepodařilo zjistit

Obecné informace o atributech

- Atributy začínající řetězcem „cvut“ jsou atributy specifické pro ČVUT a nejsou o nich informace v oficiální dokumentaci IDM.
- Řetězec „:MERGE_CLEAR“ v hodnotě atributu znamená, že aktuální hodnota v systému bude smazána a místo ní bude vložena nově definována hodnota (viz [12]).

2.3.1 ApproverDefine

(„approverDefine“)

2.3.1.1 Popis atributu

Hodnotu může mít jen byznys role a to pouze jednu z těchto dvou:

1. „true“ – role může být schvalovatelem přiřazení role k uživateli
2. „false“ – role nemůže být schvalovatelem přiřazení role k uživateli

2.3.1.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^true$|^false$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.2 ApproverRoles

(„approverRoles“)

2.3.2.1 Popis atributu

Byznys role, které mají oprávnění schválit přiřazení této role k uživateli. Hodnotou je posloupnost hodnot atributu *name* oddělených čárkou.

2.3.2.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
(([a-zA-Z0-9_\-()/]+)(,([a-zA-Z0-9_\-()/]+))*){0,1}
```

2.3.3 Approvers

(„approvers“)

2.3.3.1 Popis atributu

Schvalovatelé přiřazení role k uživateli. Pokud je aspoň jeden definován, pak každé přiřazení této role k některému uživateli musí být schváleno některým schvalovatelem. Pokud není definován žádný schvalovatel, pak se nemusí schvalovat žádné přiřazení této role (tedy roli může přiřadit jen ten, kdo k tomu má oprávnění, ale už nemusí dojít ke schválení schvalovatelem). Hodnotou je posloupnost číselných identifikátorů uživatelů z IDM oddělených čárkou (např. „80482“, „18752,80482“).

2.3.3.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^(([0-9]+)(,[0-9]+)*){0,1}$
```

2.3.4 ContainedRoles

(„containedRoles“)

2.3.4.1 Popis atributu

Role, které mohou být přiřazeny k uživateli, když je mu přiřazena tato role. (Jestli role budou uživateli opravdu přiřazeny záleží na hodnotě atributu *containedRolesAssociationType*.) Hodnotou je posloupnost hodnot atributu *name* oddělených čárkou. Např. „B-00000-KOLEGIUM-CLEN,B-00000-GREMIUM-CLEN,B-00000-VEDENI-CLEN“, „B-11000-GREMIUM-CLEN,T-SGS-11123-VEDOUCI,T-KOS-11123-REFERENT-KATEDRY“.

2.3.4.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
(([a-zA-Z0-9_\-()/]+)(,([a-zA-Z0-9_\-()/]+))*){0,1}
```

2.3.5 ContainedRolesAssociationType

(„containedRolesAssociationType“)

2.3.5.1 Popis atributu

Posloupnost řetězců (oddělených čárkou), kde n-tý řetězec označuje druh přiřazení n-té role v atributu *containedRoles*. Řetězce mohou nabývat následujících 3 hodnot:

1. „required“ – role je vždy přiřazena k uživateli
2. „conditional“ – role je přiřazena pouze v případě, že jsou splněny dané podmínky
3. „optional“ – role není přiřazena uživateli, ale uživatel si o tuto roli může zažádat (pokud příslušný schvalovatel schválí žádost, pak teprve bude uživateli role přiřazena)

Např. „required,required,required“, „required“.

2.3.5.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
((required|conditional|optional)(,(required|conditional|optional
    ))*){0,1}
```

2.3.6 CvutAcademicFncNo

(„cvutAcademicFncNo“)

2.3.6.1 Popis atributu

Označuje, zda je role akademická nebo ne. Atribut může nabývat dvou hodnot:

1. „0“ – neakademická role
2. „1“ – akademická role

2.3.6.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^0$|^1$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.7 CvutAutorole

(„cvutAutorole“)

2.3.7.1 Popis atributu

Určuje, zda budou dané roli přidělené automatické role nebo ne. Atribut může nabývat následujících hodnot:

1. „true“ – dané roli budou přidělené automatické role
2. „false“ – dané roli nebudou přidělené automatické role
3. prázdný řetězec – použije se výchozí hodnota

2.3.7.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^true$|^false$|^$
```

2.3.8 CvutBeginningDate

(„cvutBeginningDate“)

2.3.8.1 Popis atributu

Označuje období, ve kterém se daná studentská role nachází. Atribut může nabývat následujících hodnot:

1. „0-1ROK“ – student je v 1. ročníku
2. „1-2ROK“ – student je ve 2. ročníku
3. „2-9ROK“ – student je ve 3. nebo vyšším ročníku

2.3.8.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^0-1ROK$|^1-2ROK$|^2-9ROK$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.9 CvutFinishingDate

(„cvutFinishingDate“)

2.3.9.1 Popis atributu

Označuje ukončení studentské role. Atribut může nabývat následujících hodnot:

1. „>“ – datum ukončení studentské role je větší než dnešní datum a není přesně určeno

2.3.9.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^>$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.10 CvutFormOfStudy

(„cvutFormOfStudy“)

2.3.10.1 Popis atributu

Označuje formu studia dané studentské role. Atribut může nabývat tří hodnot:

1. „K“ – kombinovaná forma studia
2. „P“ – prezenční forma studia
3. „Z“ – zaměstnanec

2.3.10.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^K$|^P$|^Z$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.11 CvutLocation

(„cvutLocation“)

2.3.11.1 Popis atributu

Význam atributu se nepodařilo zjistit, hlavně z důvodu, že ani jedna role nemá vyplněnou hodnotu tohoto atributu.

2.3.11.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
// ???
```

2.3.12 CvutName

(„cvutName“)

2.3.12.1 Popis atributu

Název role.

Název byznys role Např. „předseda AS - České vysoké učení technické v Praze“.

Název technické role Např. „KOS - 14118 - Rozvrhář katedry“.

2.3.12.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
// ???
```

2.3.13 CvutNameEn

(„cvutNameEn“)

2.3.13.1 Popis atributu

Název role v anglickém jazyce.

Název byznys role Např. „Chairman - AS - Czech Technical University in Prague“.

Název technické role Není vyplněn.

2.3.13.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
// ???
```

2.3.14 CvutOrder

(„cvutOrder“)

2.3.14.1 Popis atributu

Priorita role. Hodnotou je čtyřmístné celé číslo.

Tento atribut se využívá hlavně při zobrazení rolí v hierarchickém pořadí (např. v systému Usermap). Hodnota atributu pro vyšší roli (funkci) je nižší než hodnota atributu pro nižší roli (funkci). Např. role „B-00000-REKTOR“ je vyšší než role „B-00000-GREMIUM-CLEN“, tudíž role „B-00000-REKTOR“ má hodnotu *cvutOrder* = 1001, která je nižší než hodnota *cvutOrder* role „B-00000-GREMIUM-CLEN“, která je 1202 („B-00000-REKTOR“ > „B-00000-GREMIUM-CLEN“ $\Rightarrow$ $cvutOrder_{B-00000-REKTOR} < cvutOrder_{B-00000-GREMIUM-CLEN}$).

2.3.14.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^[1-9]{1}[0-9]{0,4}$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.15 CvutOrgUnit

(„cvutOrgUnit“)

2.3.15.1 Popis atributu

Označuje číslo organizační jednotky dané role v rámci ČVUT (např. „18925“ (oddělení pro vnější vztahy Fakulty informačních technologií)).

Nejvyšší organizační jednotka

„0“ – České vysoké učení technické v Praze

Druhé nejvyšší organizační jednotky

1. „11000“ – Fakulta stavební
2. „12000“ – Fakulta strojní
3. „13000“ – Fakulta elektrotechnická
4. „14000“ – Fakulta jaderná a fyzikálně inženýrská
5. „15000“ – Fakulta architektury
6. „16000“ – Fakulta dopravní
7. „17000“ – Fakulta biomedicínského inženýrství
8. „18000“ – Fakulta informačních technologií
9. „31000“ – Kloknerův ústav
10. „32000“ – Masarykův ústav vyšších studií
11. „34000“ – Ústav tělesné výchovy a sportu
12. „35000“ – Ústav technické a experimentální fyziky
13. „36000“ – Univerzitní centrum energeticky efektivních budov

14. „37000“ – Český institut informatiky, robotiky a kybernetiky
15. „51000“ – Rektorát ČVUT
16. „81000“ – Výpočetní a informační centrum
17. „82000“ – Česká technika - nakladatelství ČVUT
18. „83000“ – Ústřední knihovna
19. „84000“ – Inovacentrum
20. „91000“ – Správa účelových zařízení

2.3.15.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^0$|^11[0-9]{3}$|^12[0-9]{3}$|^13[0-9]{3}$|^14[0-9]{3}$
   |^15[0-9]{3}$|^16[0-9]{3}$|^17[0-9]{3}$|^18[0-9]{3}$
   |^31[0-9]{3}$|^32[0-9]{3}$|^34[0-9]{3}$|^35[0-9]{3}$
   |^36[0-9]{3}$|^37[0-9]{3}|^51[0-9]{3}$|^81[0-9]{3}$
   |^82[0-9]{3}$|^83[0-9]{3}$|^84[0-9]{3}$|^91[0-9]{3}$|^$
```

2.3.16 CvutProgramType

(„cvutProgramType“)

2.3.16.1 Popis atributu

Označuje typ studia dané studentské role. Atribut může nabývat tří hodnot:

1. „B“ – bakalářské studium
2. „N“ – navazující magisterské studium
3. „D“ – doktorské studium

2.3.16.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^B$|^D$|^N$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.17 CvutQuarantineLength

(„cvutQuarantineLength“)

2.3.17.1 Popis atributu

Doba (ve dnech), po kterou uživateli zůstává tato role i po ztrátě statutu studenta nebo zaměstnance ČVUT. Např. „635“, „485“. Pokud atribut nemá žádnou hodnotu, pak role přestává platit ihned poté, co uživatel ztratí statut studenta nebo zaměstnance ČVUT.

2.3.17.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^[1-9]{1}[0-9]{0,3}$|^$
```

2.3.18 CvutRelationType

(„cvutRelationType“)

2.3.18.1 Popis atributu

Hodnotou je číslo, které označuje druh pracovního poměru role k ČVUT, např.:

1. „1“ – HPP (hlavní pracovní poměr)
2. „6“ – DPČ (dohoda o pracovní činnosti)
3. „7“ – DPP (dohoda o provedení práce)

2.3.18.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^[1-9]{1}[0-9]{0,4}$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.19 CvutResource

(„cvutResource“)

2.3.19.1 Popis atributu

Označuje ČVUT zdrojový systém, např.:

1. „VYM_STUD“ – informační systém s názvem Výměník, typ role je student
2. „VYM_ZAME“ – informační systém s názvem Výměník, typ role je zaměstnanec
3. „VYM_HOST2“ – informační systém s názvem Výměník, typ role je host

2.3.19.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
// ???
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.20 CvutStateOfStudy

(„cvutStateOfStudy“)

2.3.20.1 Popis atributu

Označuje stav studia dané studentské role. Atribut může nabývat dvou hodnot:

1. „P“ – přerušení studia
2. „S“ – studuje

2.3.20.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^P$|^S$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.21 CvutStudyProgram

(„cvutStudyProgram“)

2.3.21.1 Popis atributu

ČVUT studijní program.

2.3.21.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^kurzista$|^P\*$|^B\*$|^N\*$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.22 CvutType

(„cvutType“)

2.3.22.1 Popis atributu

Hodnotou atributu je celé číslo. Role, které mají stejnou hodnotu atributu *cvutType* mají významné společné vlastnosti. Většinou se stejnými hodnotami označují stejné funkce z rozdílných organizačních jednotek (např. role „B-11000-DEKAN“ (děkan Fakulty stavební) má hodnotu $cvutType = 2$ a role „B-18000-DEKAN“ (děkan Fakulty informačních technologií) má stejnou hodnotu $cvutType = 2$).

2.3.22.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^[1-9]{1}[0-9]{0,4}$|^$
```

2.3.23 CvutUnit

(„cvutUnit“)

2.3.23.1 Popis atributu

Označuje číslo organizační jednotky dané role. Hodnoty, kterých může atribut *cvutUnit* nabývat, závisí na typu role.

Byznys role Hodnota atributu *cvutUnit* byznys role je organizační jednotka v rámci ČVUT.

Technická role Hodnota atributu *cvutUnit* technické role je organizační jednotka v rámci ČVUT nebo fiktivní organizační jednotka, která je nutná pro potřeby příslušného informačního systému.

2.3.23.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^0$|^11[0-9]{3}$|^12[0-9]{3}$|^13[0-9]{3}$|^14[0-9]{3}$
    |^15[0-9]{3}$|^16[0-9]{3}$|^17[0-9]{3}$|^18[0-9]{3}$
    |^31[0-9]{3}$|^32[0-9]{3}$|^34[0-9]{3}$|^35[0-9]{3}$
    |^36[0-9]{3}$|^37[0-9]{3}|^51[0-9]{3}$|^81[0-9]{3}$
    |^82[0-9]{3}$|^83[0-9]{3}$|^84[0-9]{3}$|^91[0-9]{3}$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^0$|^11[0-9]{3}$|^12[0-9]{3}$|^13[0-9]{3}$|^14[0-9]{3}$
    |^15[0-9]{3}$|^16[0-9]{3}$|^17[0-9]{3}$|^18[0-9]{3}$
    |^24[0-9]{3}$|^26[0-9]{3}$|^31[0-9]{3}$|^32[0-9]{3}$
    |^34[0-9]{3}$|^35[0-9]{3}$|^36[0-9]{3}$
    |^37[0-9]{3}|^38[0-9]{3}|^39[0-9]{3}|^51[0-9]{3}$
    |^52[0-9]{3}|^81[0-9]{3}$|^82[0-9]{3}$|^83[0-9]{3}$
    |^84[0-9]{3}$|^91[0-9]{3}$|^99[0-9]{3}$|^$
```

2.3.24 CvutWorkplace

(„cvutWorkplace“)

2.3.24.1 Popis atributu

Označuje pracoviště byznys role pomocí čísla organizační jednotky v rámci ČVUT (např. „18925“ (oddělení pro vnější vztahy Fakulty informačních technologií)).

2.3.24.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^0$|^11[0-9]{3}$|^12[0-9]{3}$|^13[0-9]{3}$|^14[0-9]{3}$
    |^15[0-9]{3}$|^16[0-9]{3}$|^17[0-9]{3}$|^18[0-9]{3}$
    |^31[0-9]{3}$|^32[0-9]{3}$|^34[0-9]{3}$|^35[0-9]{3}$
    |^36[0-9]{3}$|^37[0-9]{3}|^51[0-9]{3}$|^81[0-9]{3}$
    |^82[0-9]{3}$|^83[0-9]{3}$|^84[0-9]{3}$|^91[0-9]{3}$|^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
^$
```

2.3.25 CvutZasobnik

(„cvutZasobnik“)

2.3.25.1 Popis atributu

Hodnotou je počet dní.

2.3.25.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^[1-9]{1}[0-9]{0,3}$|^$
```

2.3.26 Description

(„description“)

2.3.26.1 Popis atributu

Popis role, často také popis odlišností od jiné velmi podobné role.

Popis byznys role Např. „00000 - ČVUT - České vysoké učení technické v Praze - AS - komise pro SÚZ - předseda/předsedkyně“, „12118 - FS - ústav procesní a zpracovatelské techniky - akademický zaměstnanec DPP“.

Popis technické role Např. „Správa rozvrhů a předmětů za svou katedru, vypisuje předměty do semestrů, stanovuje jejich kapacitu, přiřazuje vyučující k předmětům na své katedře. Zapisuje rozvrh svých předmětů, upravuje textové informace o předmětech. Disponuje místnostmi své katedry (např. vyčleňuje místnosti pro konání zkoušek).“, „T-AD-18000-ZAMESTNANEC-NEAKADEMICKY“.

2.3.26.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
// ???
```

2.3.27 Disabled

(„disabled“)

2.3.27.1 Popis atributu

Tento atribut může mít pouze jednu z následujících dvou hodnot:

1. „true“ – role je zablokovaná (disabled, nelze jí používat)
2. „false“ – role není zablokovaná (může se normálně používat)

2.3.27.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^true$|^false$
```

2.3.28 ExcludedRoles

(„excludedRoles“)

2.3.28.1 Popis atributu

Role, které nemůžou být přiřazeny k uživateli. Hodnotou je posloupnost hodnot atributu *name* oddělených čárkou.

2.3.28.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
(([a-zA-Z0-9_\-()/]+)(,([a-zA-Z0-9_\-()/]+))*){0,1}
```

2.3.29 Name

(„name“)

2.3.29.1 Popis atributu

Jedinečný identifikátor role v IDM.

Identifikátor byznys role Má v IDM následující strukturu:

1. B (= byznys role)
2. -
3. XXXXX (= pětice čísel označující organizační jednotku ČVUT)
4. -
5. libovolný podřetězec takový, že celkový řetězec (hodnota atributu *name*) vyhovuje regulárnímu výrazu v sekci 2.3.29.2

Např. „B-00000-AS-PREDSEDA“.

Identifikátor role typu *IT* Má v IDM následující strukturu:

1. T (= technická role)
2. -
3. jednoznačný identifikátor informačního systému ČVUT
4. -
5. XXXXX (= pětice čísel označující organizační jednotku ČVUT)
6. -
7. libovolný podřetězec takový, že celkový řetězec (hodnota atributu *name*) vyhovuje regulárnímu výrazu v sekci 2.3.29.2

Např. „T-KOS-14117-ROZVRHAR-KATEDRY“.

2.3.29.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^[a-zA-Z0-9_\-()/]+$
```

2.3.30 Notifications

(„notifications“)

2.3.30.1 Popis atributu

Příjemci upozornění o přiřazení této role k uživateli. Hodnotou je posloupnost číselných identifikátorů uživatelů z IDM oddělených čárkou (např. „80482“, „18752,80482“).

2.3.30.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^(([0-9]+)(,[0-9]+)*){0,1}$
```

2.3.31 Owners

(„owners“)

2.3.31.1 Popis atributu

Vlastníci role. Pokud je aspoň jeden definován, pak každá změna v této roli musí být schválena některým vlastníkem. Pokud není definován žádný vlastník, pak se nemusí schvalovat žádné změny v této roli. Hodnotou je posloupnost číselných identifikátorů uživatelů z IDM oddělených čárkou (např. „80482“, „18752,80482“).

2.3.31.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^(([0-9]+)(,[0-9]+)*){0,1}$
```

2.3.32 RequestedForRole

(„requestedForRole“)

2.3.32.1 Popis atributu

Tento atribut může mít pouze jednu z následujících dvou hodnot:

1. „true“ – pokud vlastnictví této role je potřebné pro získání jiné role
2. „false“ – v ostatních případech

2.3.32.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^true$|^false$
```

2.3.33 Resources

(„resources“)

2.3.33.1 Popis atributu

Hodnotou je řetězec, který představuje ID (identifikátor) systému, pro který je definována sada zdrojových atributů v atributu *roleAttributes*. Např. „KOS“, „AD“.

2.3.33.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
// ???
```

2.3.34 RoleAttributes

(„roleAttributes“)

2.3.34.1 Popis atributu

Hodnotou je řetězec, který představuje sadu zdrojových atributů (např. uživatelské jméno, skupinu, systémová práva) pro jeden zdroj (systém) ČVUT. Např. „VICWiki:role:ADMIN-CVUT|11000:MERGE_CLEAR“.

2.3.34.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:
Pro roli typu *Business*:

```
^$
```

Pro roli typu *IT*, *Application* nebo *Asset*:

```
// ???
```

2.3.35 Type

(„type“)

2.3.35.1 Popis atributu

Označuje typ role. Typy rolí jsou následující:

Business Typicky reprezentuje pracovní funkci v organizaci. Uživatel, kterému je přiřazena role tohoto typu, získá přístupová práva potřebná k vykonávání svých pracovních povinností. Tento typ role je označován jako **byznys**.

IT, Application, Asset Tyto typy rolí jsou přiřazovány k některé byznys roli, aby uživatel, který má přiřazenou příslušnou byznys roli, získal přístupová práva ke zdroji (resource), např. studijní informační systém KOS, který potřebuje pro svou práci. Tyto typy rolí jsou označovány jako **technické**.

K byznys roli lze přiřadit množinu technických rolí, avšak byznys role může být přiřazena pouze k byznys roli. K technické roli lze přiřadit množinu technických rolí.

2.3.35.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
^ApplicationRole$|^AssetRole$|^BusinessRole$|^ITRole$
```

2.3.36 ValidFor

(„validFor“)

2.3.36.1 Popis atributu

Hodnotou je posloupnost hodnot atributu *name* oddělených čárkou. Např. „Zam,Stud,Host“.

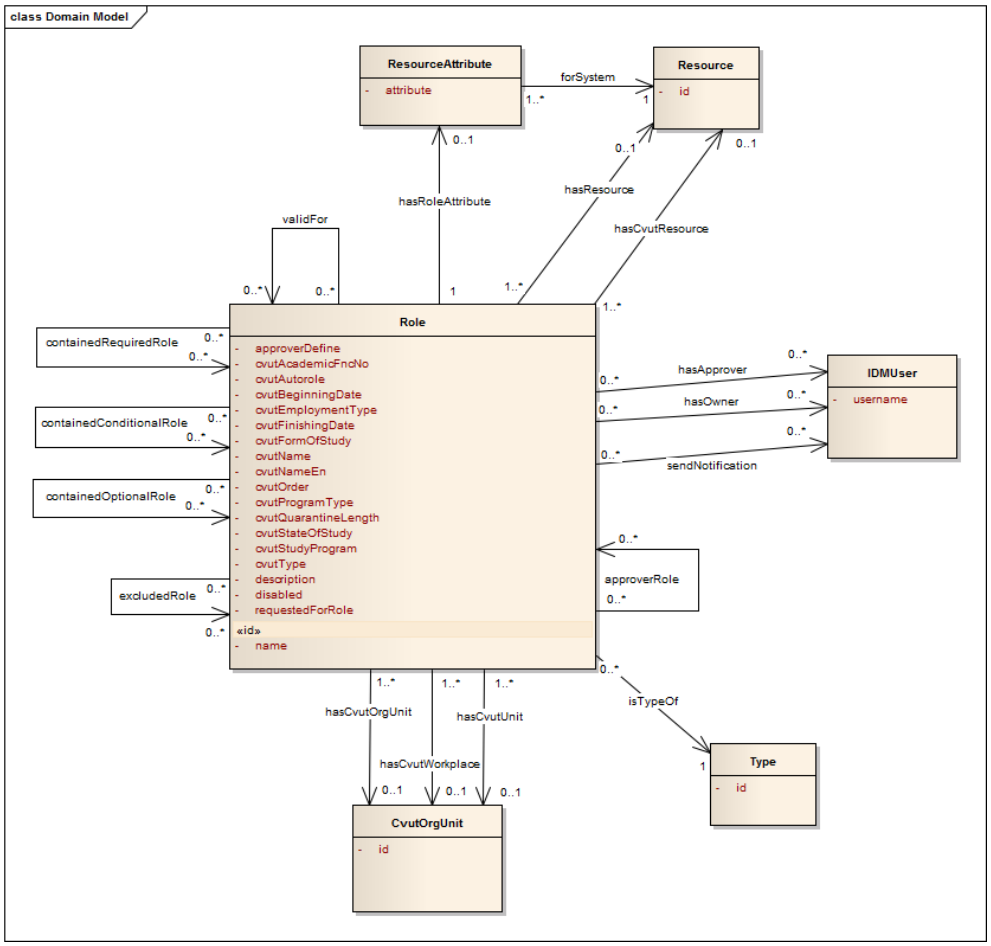
2.3.36.2 Povolená hodnota atributu

Hodnota atributu musí vyhovovat následujícímu regulárnímu výrazu:

```
(([a-zA-Z0-9_\-()/]+)(,([a-zA-Z0-9_\-()/]+))*){0,1}
```

2.4 Doménový model role

Na obrázku 2.2) je vidět doménový model role z IDM. Hlavní entitou je *Role* a ostatní entity vznikly z atributů role. Většinou to byly atributy reprezentující subjekt, který může existovat nezávisle na roli. Například organizační jednotka ČVUT, reprezentována atributem *cvutOrgUnit*, má význam nezávisle na roli, takže byla vytvořena entita *CvutOrgUnit* a příslušná asociace k entitě *Role*. (Jak je vidět na obrázku entita *CvutOrgUnit* má tři asociace s entitou Role, protože ještě atributy *cvutUnit* a *cvutWorkplace* reprezentují organizační jednotku ČVUT, jen v jiném kontextu.) Atributy, které nebyly transformovány do entity, byly transformovány do asociací. Například atribut *approverRoles* reprezentuje množinu rolí, které jsou schvalovateli dané role, tudíž byla vytvořena asociace s názvem *approverRole* znázorňující tento vztah mezi jednotlivými rolemi.



Obrázek 2.2: Doménový model

2.5 Definice chyb a anomálií v rolích

Tato sekce obsahuje definici chyb a anomálií v atributech a závislostech rolí z IDM. První podsekce zavádí pojmy pro potřeby této sekce. Druhá podsekce představuje definici chyb. A nakonec třetí podsekce obsahuje definici anomálií, včetně popisu významu slova anomálie.

2.5.1 Pojmy

Vztah mezi rolemi Vztahem mezi rolemi je závislost mezi rolemi definovaná pomocí atributů *containedRoles* nebo *approverRoles*.

2.5.2 Definice chyb

2.5.2.1 Definice chyb v atributech

Neunikátní ID Pokud hodnota atributu *name* není unikátní v celé množině rolí, pak je to chyba.

Hodnoty neodpovídající regulárním výrazům Každá hodnota atributu A, která neodpovídá příslušnému regulární výrazu, je označena jako chybná. Příslušný regulární výraz je regulární výraz ze sekce 2.3, který popisuje povolenou hodnotu atributu A. Pokud není regulární výraz pro atribut A uveden nebo atribut A v sekci 2.3 chybí (protože byl tento atribut přidán do IDM až po vypracování této sekce), pak není kontrola tohoto atributu provedena.

2.5.2.2 Definice chyb v závislostech mezi atributy

Závislost mezi atributy type, resources a roleAttributes

BusinessRole Pokud má atribut *type* hodnotu „BusinessRole“, pak hodnota atributů *resources* a *roleAttributes* musí být prázdný řetězec, jinak je taková hodnota označena jako chybná.

ITRole, ApplicationRole Pokud má atribut *type* hodnotu „ITRole“ nebo „ApplicationRole“, pak hodnota obou atributů *resources* a *roleAttributes* musí být prázdný řetězec nebo hodnota obou atributů musí být neprázdný řetězec odpovídající příslušnému regulárnímu výrazu pro daný atribut, jinak je taková hodnota označena jako chybná. Dále jestliže hodnota atributu *resources* je neprázdný řetězec R, pak hodnota atributu *roleAttributes* musí začínat řetězcem „R:“, jinak jsou hodnoty obou atributů označeny jako chybné.

Závislost mezi atributy containedRoles a containedRolesAssociationType Počet rolí (resp. příslušných hodnot atributu *name* oddělených čárkou) v hodnotě atributu *containedRoles* musí odpovídat počtu řetězců R, oddělených čárkou, v hodnotě atributu *containedRolesAssociationType*, jinak jsou hodnoty obou atributů označeny jako chybné. (Hodnoty, kterých může nabývat řetězec R, jsou uvedeny v sekci 2.3.)

Příklad bez chyby

- Atribut *containedRoles* má hodnotu „B-00000-KOLEGIUM-CLEN,B-00000-GREMIUM-CLEN,B-00000-VEDENI-CLEN,T-AD-00000-AS-CLEN“.
- Atribut *containedRolesAssociationType* má hodnotu „required,required ,required,required“.

Příklad s chybou

- Atribut *containedRoles* má hodnotu „B-00000-KOLEGIUM-CLEN,B-00000-GREMIUM-CLEN,B-00000-VEDENI-CLEN,T-AD-00000-AS-CLEN“.
- Atribut *containedRolesAssociationType* má hodnotu „required,required,required“.

V hodnotě atributu *containedRoles* jsou 4 role, ale v hodnotě atributu *containedRolesAssociationType* jsou jen 3 řetězce *R*.

2.5.2.3 Definice chyb ve vztazích mezi rolemi

Vztahy typu containedRole Vztahy mezi rolemi typu *containedRole* definované podle atributu *containedRoles*.

Popis chyb

- **Neexistující role** Pokud hodnota atributu *containedRoles* obsahuje roli, která neexistuje, pak je tato hodnota označena jako chybná.
- **Duplicitní role** Pokud hodnota atributu *containedRoles* obsahuje nějakou roli dvakrát, pak je tato hodnota označena jako chybná.
- **Špatný typ role** Pokud role není typu *Business* a zároveň její hodnota atributu *containedRoles* obsahuje roli typu *Business*, pak je to chyba.

Vztahy typu approverRole Vztahy mezi rolemi typu „approverRole“ definované podle atributu *approverRoles*.

Popis chyb

- **Neexistující role** Pokud hodnota atributu *approverRoles* obsahuje roli, která neexistuje, pak je tato hodnota označena jako chybná.
- **Duplicitní role** Pokud hodnota atributu *approverRoles* obsahuje nějakou roli dvakrát, pak je tato hodnota označena jako chybná.
- **Špatný typ role** Pokud hodnota atributu *approverRoles* obsahuje roli, která není typu *Business*, pak je to chyba.

Vztahy obou typů (containedRole a approverRole)

Popis chyb

- **Cyklus** Nechť orientovaný graf $G = \langle H, U, \sigma \rangle$ – kde množina uzlů U reprezentuje role, množina orientovaných hran H reprezentuje vztahy typu „containedRole“ a „approverRole“ a incidence σ je zobrazení $H \longrightarrow U \times U$ – představuje role a jejich vztahy. Pokud v grafu G existuje cyklus, pak je to chyba.

2.5.3 Definice anomálií

Tato sekce obsahuje definici anomálií. Anomálie znamená odchylka od normálu, resp. normálních dat. Normální data byla stanovena jako data, která se vyskytují aspoň v $X\%$ případů.

X Proměnná X, která se vyskytuje i v následujících podsekcích, má hodnotu 60.

2.5.3.1 Pojmy

Ekvivalentní role Role r_e je ekvivalentní role k roli r právě tehdy, když hodnota atributu *cvutType* je u rolí r a r_e stejná. (Např. role „B-11000-DEKAN“ (děkan Fakulty stavební) má hodnotu $cvutType = 2$ a role „B-18000-DEKAN“ (děkan Fakulty informačních technologií) má stejnou hodnotu $cvutType = 2$, tudíž jsou tyto dvě role ekvivalentní.)

Skupina role Množina všech rolí ekvivalentních k roli r se nazývá skupina role r.

2.5.3.2 Definice anomálií v atributech

U všech atributů je popis anomálií tento:

- Pokud se hodnota h atributu liší od $X\%$ stejných hodnot atributu ostatních rolí ve skupině, pak hodnota h je anomálie.

2.5.3.3 Definice anomálií ve vztazích mezi rolemi

Vztahy typu containedRole Vztahy mezi rolemi typu „containedRole“ definované podle atributu *containedRoles*.

Popis anomálií Hodnota atributu *containedRoles* role r má tvar $n_1, \ldots, n_n$, kde n_i ($i \in \{1, \ldots, n\}$) je hodnota atributu *name* příslušné role, s kterou má role r vztah. Hodnota atributu *containedRoles* se převede na *typový-seřazený* tvar $c_1, \ldots, c_n$, kde c_i ($i \in \{1, \ldots, n\}$) je hodnota atributu *cvutType* příslušné role nebo hodnota -1, pokud hodnota atributu *cvutType* není číslo, a zároveň platí, že pro $s < t$ ($s, t \in \{1, \ldots, n\}$) je $c_s \leq c_t$. (Pokud by hodnota atributu *containedRoles* byla prázdný řetězec, pak *typový-seřazený* tvar je také prázdný řetězec.)

Pokud ve skupině role r má aspoň $X\%$ rolí *typový-seřazený* tvar $c_1, \ldots, c_m$, který není shodný s *typovým-seřazeným* tvarem role r, pak hodnota atributu *containedRoles* role r je anomálie. (Tato anomálie značí, že role r má odlišný počet příslušných vztahů a nebo existuje typ role, který chybí nebo naopak je navíc v roli r.)

Vztahy typu approverRole Vztahy mezi rolemi typu „approverRole“ definované podle atributu *approverRoles*.

Popis anomálií Hodnota atributu *approverRoles* role r má tvar $n_1, \ldots, n_n$, kde n_i ($i \in \{1, \ldots, n\}$) je hodnota atributu *name* příslušné role, s kterou má role r vztah. Hodnota atributu *approverRoles* se převede na *typový-seřazený* tvar $c_1, \ldots, c_n$, kde c_i ($i \in \{1, \ldots, n\}$) je hodnota atributu *cvutType* příslušné role nebo hodnota -1, pokud hodnota atributu *cvutType* není číslo, a zároveň platí, že pro $s < t$ ($s, t \in \{1, \ldots, n\}$) je $c_s \leq c_t$. (Pokud by hodnota atributu *approverRoles* byla prázdný řetězec, pak *typový-seřazený* tvar je také prázdný řetězec.)

Pokud ve skupině role r má aspoň $X\%$ rolí *typový-seřazený* tvar $c_1, \ldots, c_m$, který není shodný s *typovým-seřazeným* tvarem role r, pak hodnota atributu *approverRoles* role r je anomálie. (Tato anomálie značí, že role r má odlišný počet příslušných vztahů a nebo existuje typ role, který chybí nebo naopak je navíc v roli r.)

Kapitola 3

Specifikace požadavků

Tato kapitola obsahuje specifikaci požadavků pro výslednou aplikaci pro analýzu a vizualizaci uživatelských rolí v IDM ČVUT, která bude mít název „Role Checker“.

3.1 Model požadavků

3.1.1 Funkční požadavky

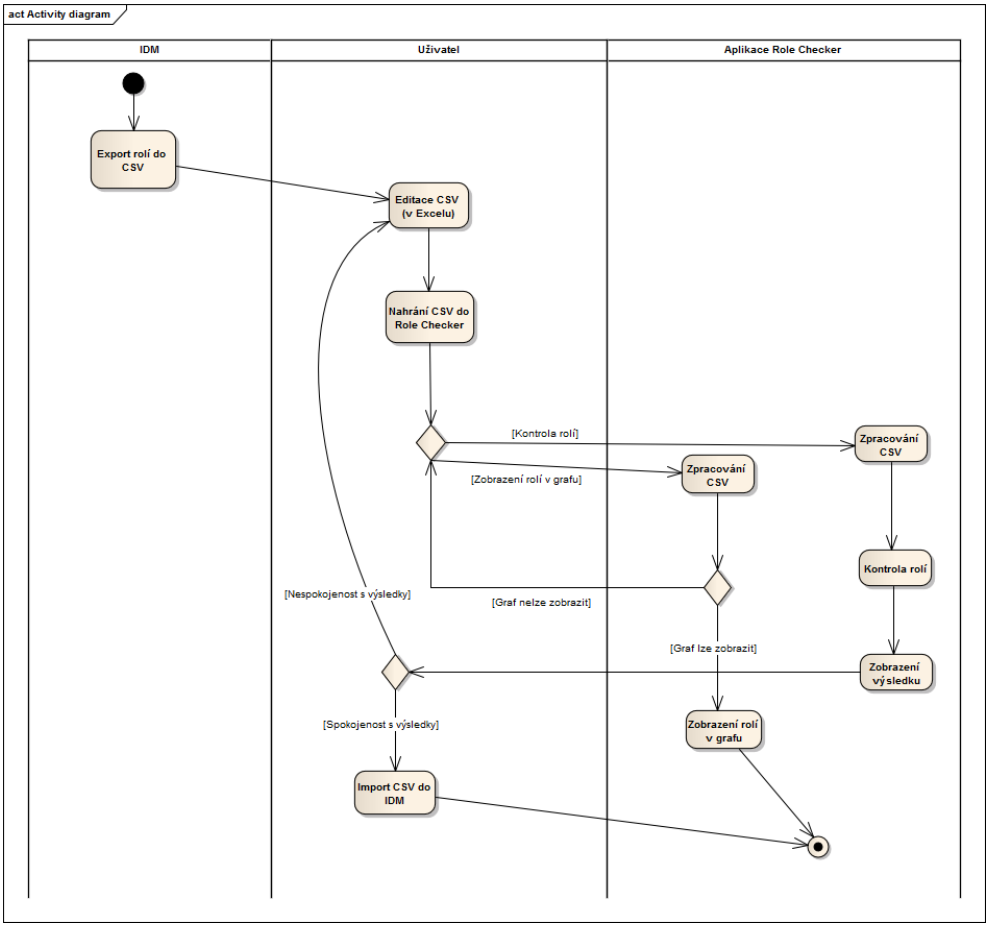
- Aplikace Role Checker bude hledat chyby a anomálie v závislostech mezi rolemi.
- Aplikace Role Checker bude hledat chyby a anomálie v atributech role.
- Aplikace Role Checker zobrazí nalezené chyby a anomálie nebo umožní stažení CSV souboru, který bude obsahovat všechny nalezené chyby a anomálie.
- Aplikace Role Checker zobrazí graf rolí a bude v něm vyhledávat a provádět nad ním filtry na základě vstupu od uživatele.
- Aplikace Role Checker bude podporovat autentizaci a autorizaci.

3.1.2 Nefunkční požadavky

- Aplikace Role Checker bude mít webové uživatelské rozhraní.
- Aplikace Role Checker bude podporovat webové prohlížeče Mozilla Firefox (verze 36 a vyšší) a Internet Explorer (verze 11 a vyšší).
- Aplikace Role Checker bude schopná obsluhovat minimálně dva paralelně pracující uživatele.
- Aplikace Role Checker bude načítat role ze souboru ve formátu CSV.

3.2 Diagram aktivit

Na obrázku 3.1 je pomocí diagramu aktivit znázorněn proces, který se týká zpracování rolí. Tento proces začíná vyexportováním rolí ze systému IDM do CSV souboru. Pokračuje úpravou CSV souboru uživatelem, což většinou zahrnuje přidání rolí a změnu atributů některých stávajících rolí, a dalšími akcemi, hlavně zobrazením rolí v grafu a kontrolou rolí v aplikaci Role Checker. Proces většinou končí naimportováním upraveného CSV zpátky do systému IDM.



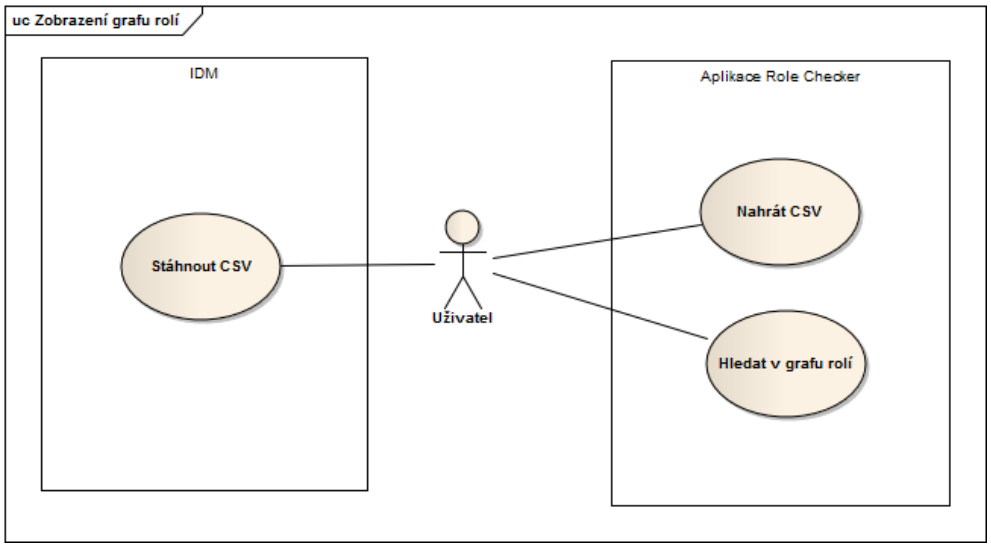
Obrázek 3.1: Diagram aktivit

3.3 Model případů užití

Model případů užití zachycuje používané funkce konkrétních systémů jednotlivými účastníky. V této sekci jsou znázorněny případy užití v rámci dvou hlavních procesů, zobrazení grafu rolí a kontroly rolí.

3.3.1 Zobrazení grafu rolí

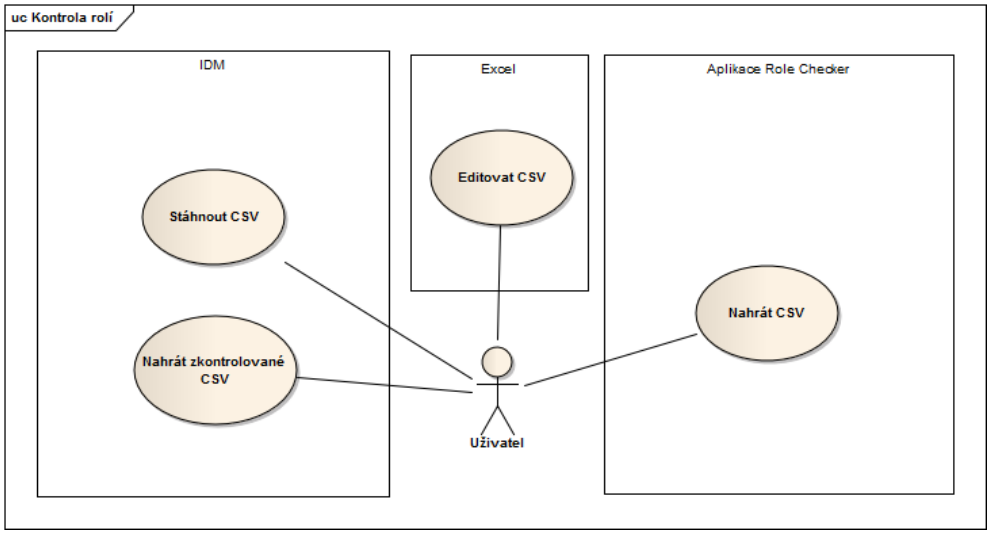
Na obrázku 3.2 jsou zachyceny případy užití v rámci procesu zobrazení grafu rolí.



Obrázek 3.2: Zobrazení grafu rolí

3.3.2 Kontrola rolí

Na obrázku 3.3 jsou zachyceny případy užití v rámci procesu kontroly rolí.



Obrázek 3.3: Kontrola rolí

Kapitola 4

Návrh

Tato kapitola popisuje návrh aplikace Role Checker. Role Checker bude webová aplikace typu klient-server, která bude umožňovat zobrazení a kontrolu IDM rolí a jejich závislostí. V následujících sekcích jsou popsány vlastnosti aplikace.

4.1 Model nasazení

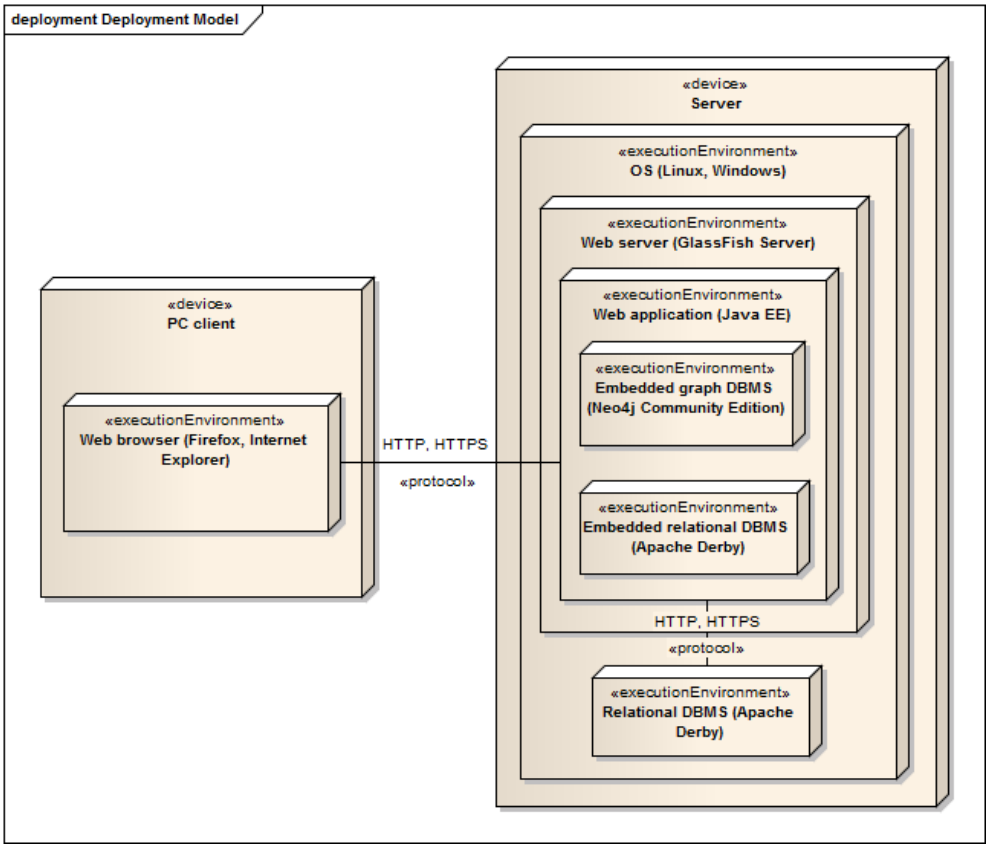
Na modelu nasazení (obrázek 4.1) je vidět, že webová aplikace Role Checker bude typu klient-server. Aplikace bude provozována na webovém serveru GlassFish Server Open Source Edition. Pro trvalé uložení dat bude použita grafová databáze Neo4j Community Edition v módu embedded (výhoda embedded módu oproti server módu viz sekce 1.3.3.2) a relační databáze Apache Derby v módu embedded i network server.

Aplikace bude postavena na platformě Java Platform, Enterprise Edition (Java EE) a programovacím jazyku Java. Hlavní důvody pro výběr jazyka Java jsou následující:

1. Je možné pracovat s databází Neo4j v embedded módu přes Neo4j API, protože Neo4j databáze je implementována v programovacích jazycích Java a Scala.

2. Jazyk Java má rozsáhlou podporu knihoven.

Klientský počítač bude komunikovat s aplikací přes protokol Hypertext Transfer Protocol (HTTP) a Hypertext Transfer Protocol Secure (HTTPS). Uživatel bude k aplikaci přistupovat přes webový prohlížeč Mozilla Firefox (verze 36 nebo vyšší) nebo Internet Explorer (verze 11 nebo vyšší).

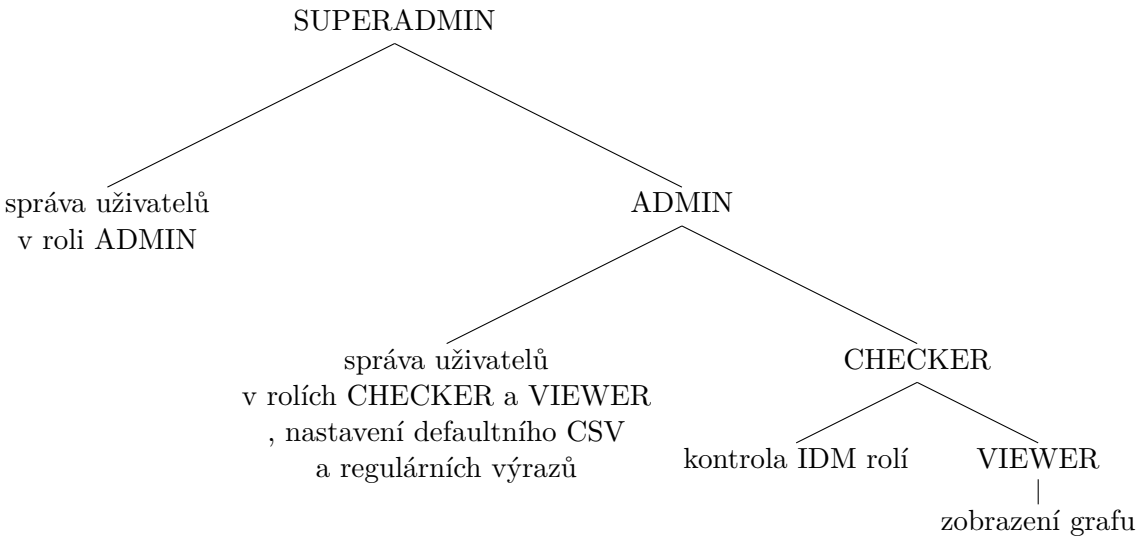


Obrázek 4.1: Model nasazení (deployment model)

4.2 Autentizace a autorizace

Autentizace uživatele se bude provádět pomocí uživatelského jména a hesla. Autorizace uživatele se bude provádět na základě uživatelské role.

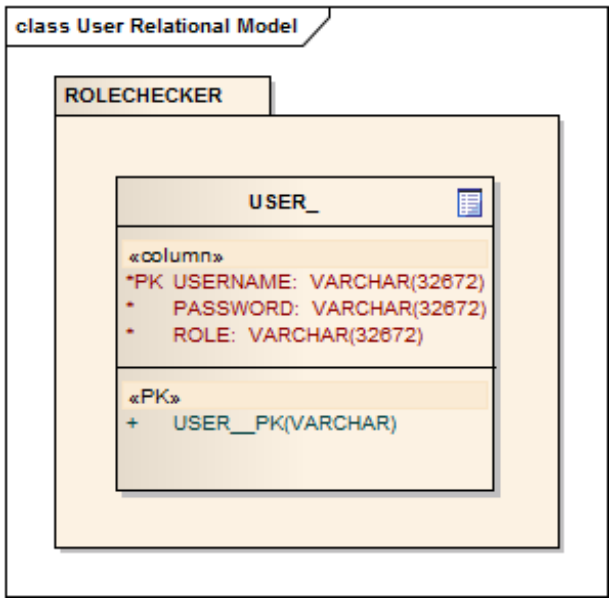
Na obrázku 4.2 je znázorněna hierarchie uživatelských rolí a práva uživatelských rolí. Nejméně práv má role, která je nejníže, což je role *VIEWER*. Každá role výše obsahuje zároveň práva role níže, tudíž nejvíce práv má role, která je nejvýše, což je role *SUPERADMIN*. Např. role *VIEWER* má jen právo zobrazit graf IDM rolí, ale role *CHECKER*, která je o jednu úroveň výše než role *VIEWER*, má právo zobrazit graf IDM rolí i právo provedení kontroly IDM rolí.



Obrázek 4.2: Hierarchie uživatelských rolí v aplikaci Role Checker

4.3 Relační model uživatelů

Relační model na obrázku 4.3 ukazuje uložení uživatelů aplikace Role Checker v databázi Derby. V následující podsekci je popsána tabulka *USER_*.



Obrázek 4.3: Relační model uživatelů

4.3.1 Tabulka USER_

Tabulka *USER_* reprezentuje uživatele aplikace Role Checker. Tabulka obsahuje následující sloupce:

USERNAME uživatelské jméno

PASSWORD heš uživatelského hesla

ROLE název uživatelské role, která byla přidělena uživateli v aplikaci Role Checker

4.4 Kódování a formát CSV

Následující dvě sekce obsahují definici kódování a formátu CSV souboru rolí, který slouží jako vstupní data pro aplikaci Role Checker.

4.4.1 Kódování CSV

Soubor musí být v kódování UTF-8 (UCS Transformation Format – 8-bit).

4.4.2 Formát CSV

Formát CSV souboru rolí musí splňovat normu RFC 4180 (viz [14]), kromě rozdílů uvedených v sekci 4.4.2.1, a zároveň doplňující pravidla uvedená v sekci 4.4.2.2.

4.4.2.1 Rozdíly od normy RFC 4180

- Oddělovačem řádků může být:
  1. line feed (LF)

     `\n`

  2. carriage return (CR)

     `\r`

  3. CR LF

     `\r\n`

4.4.2.2 Doplňující pravidla

- První řádek musí být definicí hlavičky rolí. Hlavička musí obsahovat minimálně následující hodnoty:

  `name,type,containedRoles,approverRoles,cvutType`

- Druhý až předposlední řádek je definicí role, např.:

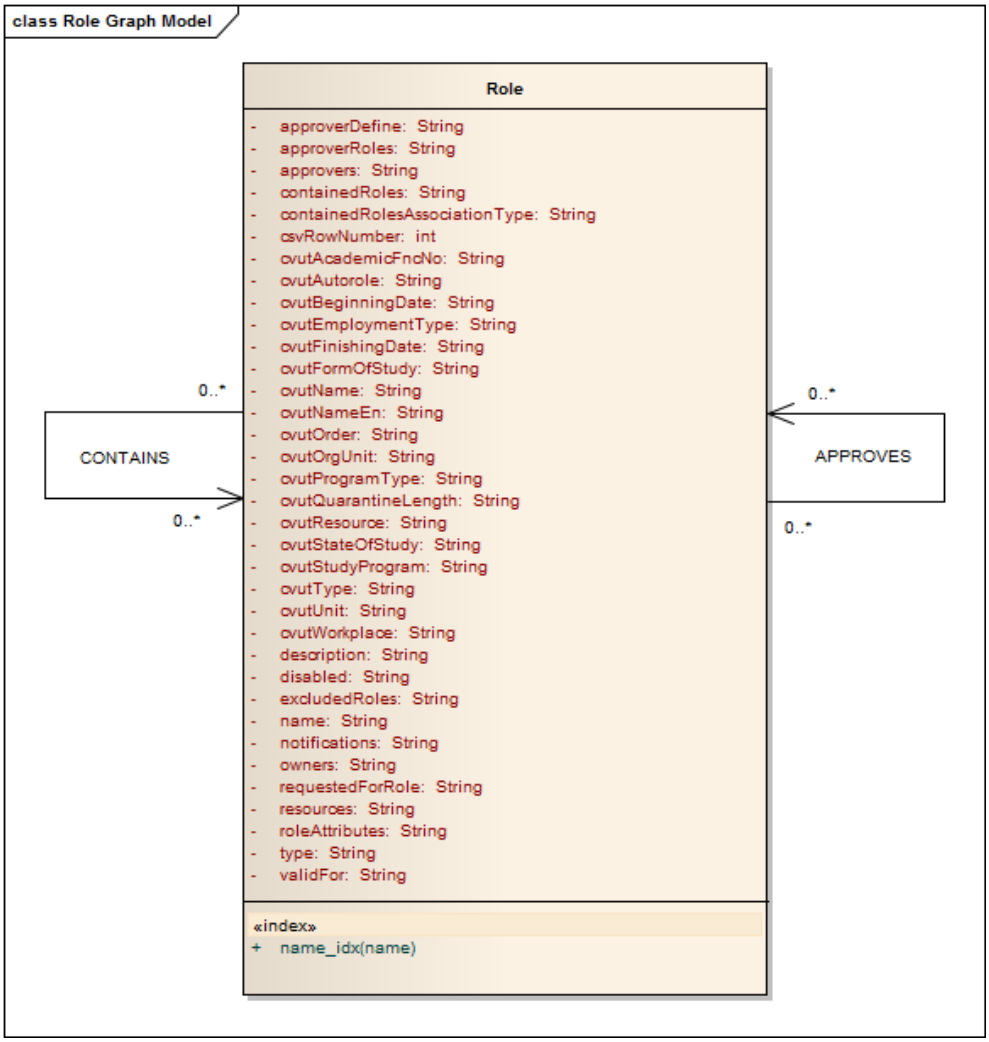
  B-00000-TEST-PREDSEDA,testovací předseda - České vysoké učení technické v Praze,BusinessRole,00000 - ČVUT - České vysoké učení technické v Praze - testovací předseda/předsedkyně,false,true,false,B-00000-TEST-CLEN,required,,,,"Zam,Stud",,,,,,,,,,,,,,,,,,,,,0,170,Test Chairman - AS - Czech Technical University in Prague,1310,

- Poslední řádek může být:

  1. definicí role
  2. pouze ukončovací řádek, pokud neobsahuje žádný znak (tento řádek je při parsování ignorován)

4.5 Grafový model rolí

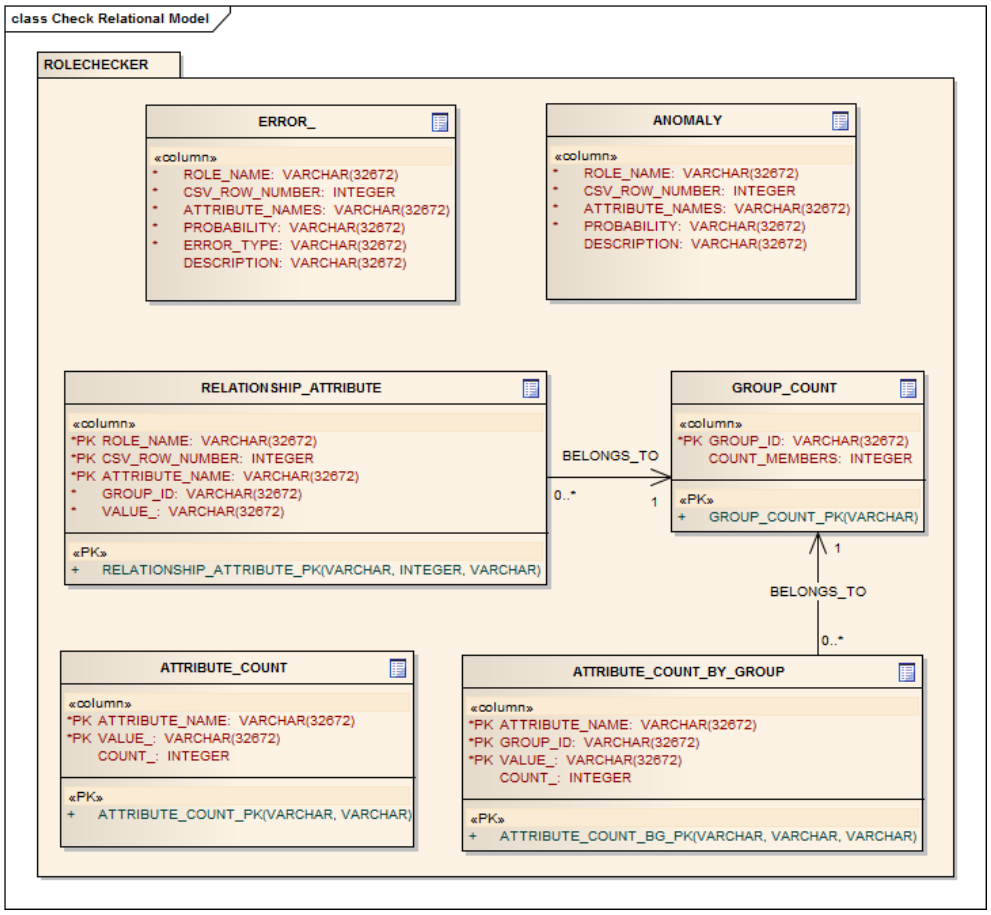
Grafový model rolí na obrázku 4.4 ukazuje reprezentaci IDM rolí v grafové databázi Neo4j. Role jsou uzly s označením „Role“. Všechny atributy reprezentují vlastnosti uzlu. Všechny tyto vlastnosti jsou datového typu *String*. Navíc je pouze vlastnost *csvRowNumber*, datového typu *int*, která označuje číslo CSV řádku příslušné role v CSV souboru rolí. Pro vlastnost *name* bude vytvořen index. Vztah s označením „CONTAINS“, resp. „APPROVES“, je definován podle atributu *containedRoles*, resp. *approverRoles*.



Obrázek 4.4: Grafový model rolí

4.6 Relační model kontroly rolí

Relační model na obrázku 4.5 ukazuje reprezentaci uložených dat, která vznikla při kontrole rolí, v databázi Derby. V následujících podsekcích jsou popsány jednotlivé tabulky.



Obrázek 4.5: Relační model kontroly rolí

4.6.1 Tabulka ERROR_

Tabulka *ERROR_* reprezentuje chybu v atributu role nebo v závislosti mezi rolemi. Tabulka obsahuje následující sloupce:

ROLE_NAME hodnota atributu *name* role, ke které se chyba vztahuje

CSV_ROW_NUMBER číslo CSV řádku role, ke které se chyba vztahuje

ATTRIBUTE_NAMES názvy atributů, ke kterým se chyba vztahuje

PROBABILITY pravděpodobnost, že nalezená chyba je opravdu chybou

ERROR_TYPE typ chyby

DESCRIPTION popis chyby

4.6.2 Tabulka ANOMALY

Tabulka *ANOMALY* reprezentuje anomálii v atributu role nebo v závislosti mezi rolemi. Tabulka obsahuje následující sloupce:

ROLE_NAME hodnota atributu *name* role, ke které se anomálie vztahuje

CSV_ROW_NUMBER číslo CSV řádku role, ke které se anomálie vztahuje

ATTRIBUTE_NAMES názvy atributů, ke kterým se anomálie vztahuje

PROBABILITY pravděpodobnost, že nalezená anomálie je chybou

DESCRIPTION popis anomálie

4.6.3 Tabulka ATTRIBUTE_COUNT

Tabulka *ATTRIBUTE_COUNT* reprezentuje určitou hodnotu určitého atributu a počet výskytů této hodnoty v různých rolí. Tabulka obsahuje následující sloupce:

ATTRIBUTE_NAME název atributu, ke kterému se daná hodnota vztahuje

VALUE_ daná hodnota

COUNT_ počet výskytů dané hodnoty

4.6.4 Tabulka GROUP_COUNT

Tabulka *GROUP_COUNT* reprezentuje skupinu role (skupina se určuje na základě hodnoty atributu *cvutType*) a počet členů v této skupině. Tabulka obsahuje následující sloupce:

GROUP_ID ID skupiny (hodnota atributu *cvutType*)

COUNT_MEMBERS počet členů ve skupině

4.6.5 Tabulka ATTRIBUTE_COUNT_BY_GROUP

Tabulka *ATTRIBUTE_COUNT_BY_GROUP* reprezentuje určitou hodnotu určitého atributu a počet výskytů této hodnoty v různých rolí určité skupiny. Tabulka obsahuje následující sloupce:

ATTRIBUTE_NAME název atributu, ke kterému se daná hodnota vztahuje

GROUP_ID ID skupiny

VALUE_ daná hodnota

COUNT_ počet výskytů dané hodnoty

4.6.6 Tabulka RELATIONSHIP_ATTRIBUTE

Tabulka *RELATIONSHIP_ATTRIBUTE* reprezentuje transformovanou hodnotu určitého vztahového atributu (*approverRoles* nebo *containedRoles*). Tabulka obsahuje následující sloupce:

ROLE_NAME hodnota atributu *name* role, ke které se transformovaná hodnota vztahuje

CSV_ROW_NUMBER číslo CSV řádku role, ke které se transformovaná hodnota vztahuje

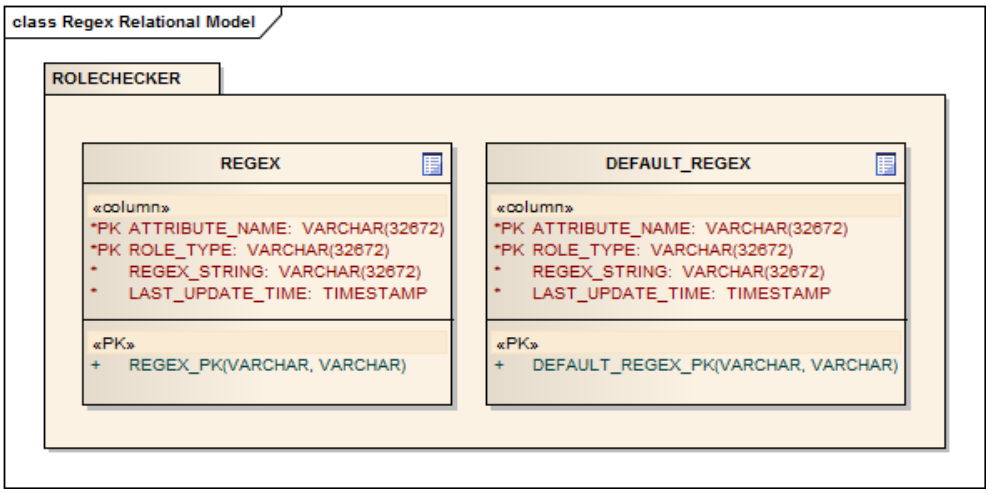
ATTRIBUTE_NAME název atributu, ke kterému se transformovaná hodnota vztahuje

GROUP_ID ID skupiny

VALUE_ transformovaná hodnota

4.7 Relační model regulárních výrazů pro kontrolu rolí

Relační model na obrázku 4.6 ukazuje uložení regulárních výrazů pro kontrolu rolí v databázi Derby. V následujících podsekcích jsou popsány jednotlivé tabulky.



Obrázek 4.6: Relační model regulárních výrazů pro kontrolu rolí

4.7.1 Tabulka REGEX

Tabulka *REGEX* reprezentuje regulární výraz pro určitý typ role. Tabulka obsahuje následující sloupce:

ATTRIBUTE_NAME název atributu, ke kterému se regulární výraz vztahuje

ROLE_TYPE typ role, pro který je regulární výraz určen

REGEX_STRING regulární výraz

LAST_UPDATE_TIME čas poslední aktualizace regulárního výrazu

4.7.2 Tabulka DEFAULT_REGEX

Tabulka *DEFAULT_REGEX* reprezentuje výchozí regulární výraz pro určitý typ role, tedy ten regulární výraz, který byl nastaven při instalaci aplikace Role Checker. Tabulka obsahuje následující sloupce:

ATTRIBUTE_NAME název atributu, ke kterému se regulární výraz vztahuje

ROLE_TYPE typ role, pro který je regulární výraz určen

REGEX_STRING regulární výraz

LAST_UPDATE_TIME čas poslední aktualizace regulárního výrazu

Kapitola 5

Implementace

Tato kapitola popisuje implementaci aplikace Role Checker. Implementace byla provedena přesně podle návrhu uvedeného v kapitole 4. V následující sekci je uvedena většina použitých technologiích, včetně podrobného popisu nejdůležitějších technologií, které nebyly popsány v kapitole 1. V dalších sekcích jsou popsány architektura aplikace, implementace zobrazení grafu rolí a implementace kontroly rolí.

5.1 Technologie

Následující seznam obsahuje většinu použitých technologií v aplikaci Role Checker:

Alchemy.js, verze 0.4.2 technologie je popsána v sekci 5.1.1

Apache Commons Codec, verze 1.10 knihovna, napsaná v programovacím jazyku Java, která implementuje kódovací a dekódovací funkce (Base64, MD5, SHA-256 a další) [15]

Apache Commons Collections, verze 4.0 knihovna, napsaná v programovacím jazyku Java, implementující kolekce, které jsou nadstavbou kolekcí obsažených v JDK (Java SE Development Kit)

Apache Commons IO, verze 2.4 knihovna, napsaná v programovacím jazyku Java, implementující vstupní a výstupní nástroje, které jsou nadstavbou vstupních a výstupních nástrojů obsažených v JDK

Apache Commons Lang, verze 3.3.2 knihovna, napsaná v programovacím jazyku Java, která implementuje pomocné nástroje pro třídy z knihovny *java.lang* obsažené v JDK

Apache Derby, verze 10.10.2.0 relační databáze, napsaná v programovacím jazyku Java, která může být provozována v módu network server i embedded

Apache FOP, verze 2.0 knihovna, napsaná v programovacím jazyku Java, která implementuje transformování XSL (Extensible Stylesheet Language) formátovacích objektů do různých výstupních formátů

Java EE, verze 7 technologie je popsána v sekci 5.1.2

Neo4j Community Edition, verze 2.1.8 technologie je popsána v sekci 1.3

Primefaces, verze 5.1 knihovna, napsaná v programovacím jazyku Java, která implementuje nové komponenty uživatelského rozhraní pro technologii JavaServer Faces (JSF), více o JSF viz sekce 5.1.2

Super CSV, verze 2.3.1 knihovna, napsaná v programovacím jazyku Java, která implementuje importování CSV souboru do Java objektů a exportování Java objektů do CSV souboru

V dalších podsekcích jsou popsány nejdůležitější technologie, které nebyly popsány v kapitole 1.

5.1.1 Alchemy.js

Alchemy.js je knihovna, napsána v programovacím jazyku JavaScript, sloužící k vizualizaci grafu ve webovém prohlížeči [1]. Je postavena na knihovně D3.js, také napsané v programovacím jazyku JavaScript, která umožňuje manipulaci s dokumenty založenými na datech a následné vizualizaci ve webovém prohlížeči. Výhodou knihovny Alchemy.js oproti samostatné knihovně D3.js je, že knihovna Alchemy.js je přímo určena k vizualizaci grafu, takže stačí nastavit grafová data a několik parametrů a ve webovém prohlížeči můžou být zobrazeny graf, který už má nastaven barevné styly a neobsahuje překrývající se uzly, a panel umožňující provádět základní operace nad grafem, viz obrázek 5.1. Konfigurace Alchemy.js pro zmíněný graf vypadá následovně:

```
1  var config = {
2    dataSource: 'data/team.json',
3
4    cluster: true,
5
6    nodeTypes: {"node_type":
7                [
8                 "family",
9                 "coworker",
10                "classmate",
11                "friend",
12                "other"
13               ]
```

```
14              },
15    nodeCaption: "firstName",
16    rootNodeRadius: 30,
17
18    showControlDash: true,
19
20    showStats: true,
21    nodeStats: true,
22
23    showFilters: true,
24    nodeFilters: true,
25
26    captionToggle: true,
27    edgesToggle: true,
28    nodesToggle: true,
29
30    zoomControls: true
31  };
```

Význam jednotlivých parametrů je následující:

dataSource zdroj dat grafu ve formátu GraphJSON (JSON (JavaScript Object Notation) objekt nebo cesta k souboru, který obsahuje JSON objekt), viz sekce 5.1.1.1

cluster zapnutí rozdělení uzlů do klastrů (shluků) podle hodnoty atributu „cluster“ definovaného v datech uzlu (uzly ve stejném klastru mají stejnou barvu a každý klastr má přidělenou jinou barvu)

nodeTypes typy uzlů

nodeCaption titulek uzlu

rootNodeRadius výchozí rádius kořenových uzlů

showControlDash zobrazení panelu, který bude dále v textu nazýván „Control Dash“, umožňujícího provádět operace nad grafem a zobrazovat statistiky grafu

showStats zobrazení statistik v panelu *Control Dash*

nodeStats zobrazení statistik pro uzly

showFilters zobrazení filtrů v panelu *Control Dash*

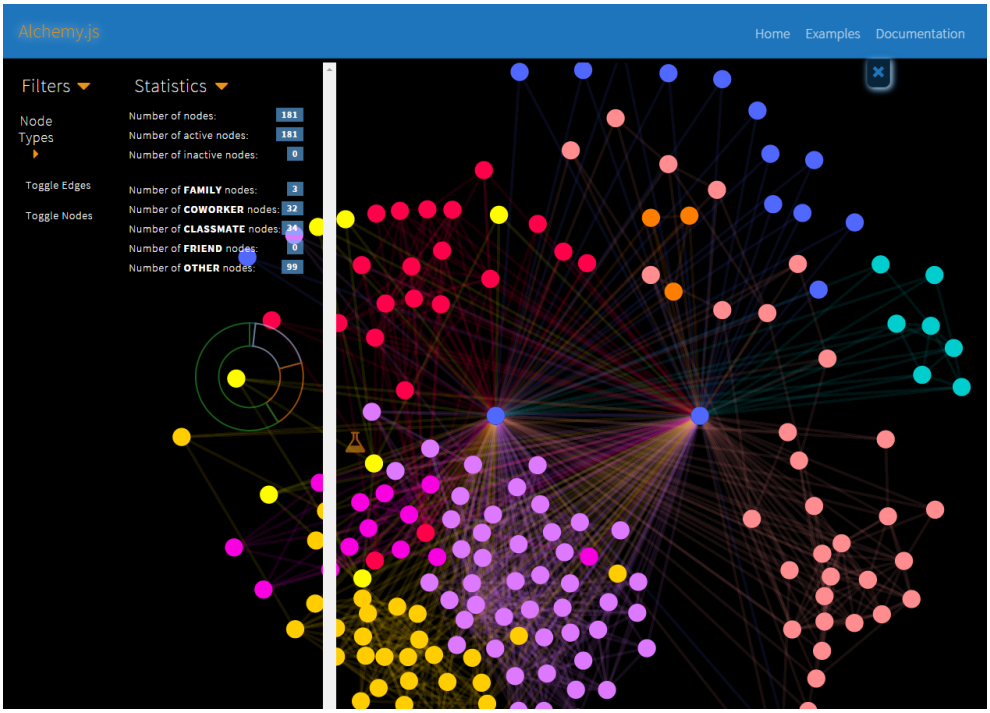
nodeFilters zobrazení filtrů pro uzly

captionToggle zobrazení, v panelu *Control Dash*, elementu, který umožňuje skrytí, resp. zobrazení, titulků uzlů (tento element není na obrázku 5.1 zobrazen z důvodu chyby v knihovně Alchemy.js, podrobnosti viz sekce 5.1.1.2)

edgesToggle zobrazení, v panelu *Control Dash*, elementu, který umožňuje skrytí, resp. zobrazení, hran

nodesToggle zobrazení, v panelu *Control Dash*, elementu, který umožňuje skrytí, resp. zobrazení, uzlů

zoomControls zobrazení tlačítek pro přiblížení a oddálení grafu (tato tlačítka nejsou na obrázku 5.1 zobrazeny z důvodu chyby v knihovně Alchemy.js, podrobnosti viz sekce 5.1.1.2)



Obrázek 5.1: Graf zobrazený pomocí knihovny Alchemy.js (vlevo lze vidět panel umožňující provádět operace nad grafem) [1]

5.1.1.1 GraphJSON

Knihovna Alchemy.js přijímá vstupní data grafu ve formátu GraphJSON. GraphJSON je formát navržený pro reprezentaci grafu a je tvořen JSON objektem,

který obsahuje dva objekty: *nodes* a *edges*. Objekt *nodes* je datové pole obsahující jednotlivé uzly grafu, včetně jejich atributů. Objekt *edges* je datové pole obsahující jednotlivé hrany grafu, včetně jejich atributů. Následující kód je příkladem grafu definovaného ve formátu GraphJSON (graf zobrazený na základě tohoto kódu je vidět na obrázku 5.2):

```
{
   "nodes": [
      {
         "id": "B-00000-TEST-CLEN",
         "type": "Role",
         "csvRowNumber": 2,
         "root": true
      },
      {
         "id": "B-00000-TEST-PREDSEDA",
         "type": "Role",
         "csvRowNumber": 3,
         "root": false
      },
      {
         "id": "T-SFIS-12360-TEST-VEDOUCI",
         "type": "Role",
         "csvRowNumber": 4,
         "root": false
      },
      {
         "id": "T-SFIS-12361-TEST-CONTROLLING",
         "type": "Role",
         "csvRowNumber": 5,
         "root": false
      }
   ],
   "edges": [
      {
         "source": "B-00000-TEST-CLEN",
         "target": "T-SFIS-12360-TEST-VEDOUCI",
         "type": "APPROVES"
      },
      {
         "source": "B-00000-TEST-CLEN",
         "target": "T-SFIS-12361-TEST-CONTROLLING",
         "type": "CONTAINS"
      },
```

```
39      {
40         "source": "B-00000-TEST-PREDSEDA",
41         "target": "B-00000-TEST-CLEN",
42         "type": "APPROVES"
43      }
44   ]
45 }
```

Nejdůležitější atributy uzlů a hran, definovaných ve formátu GraphJSON, jsou:

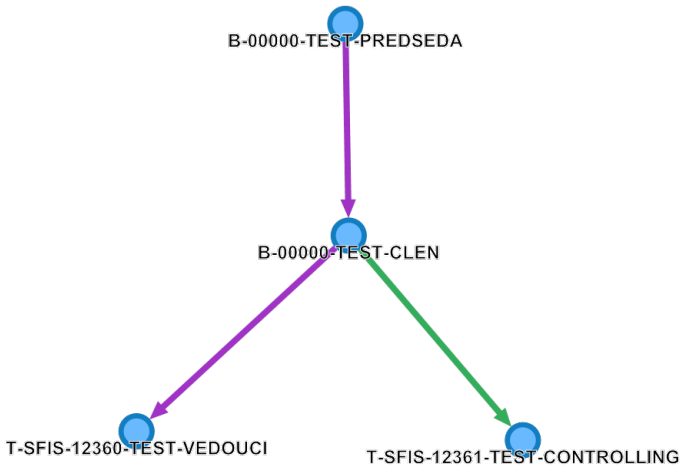
id unikátní identifikátor uzlu, resp. hrany, v grafu

type typ uzlu, resp. hrany

root označuje, zda uzel je nebo není kořenem

source *id* uzlu dané hrany

target *id* uzlu dané hrany



Obrázek 5.2: Graf zobrazený na základě dat ve formátu GraphJSON

5.1.1.2 Problémy

Při implementaci zobrazení grafu v aplikaci Role Checker se objevila řada problémů spojených s knihovnou Alchemy.js, které výrazně ztížily dokončení aplikace Role Checker. V následujících odstavcích jsou uvedeny některé z těchto problémů.

Editor a tlačítka přiblížení V dokumentaci Alchemy.js [16] je uvedeno, že po dosazení hodnoty *true* do parametru *showEditor*, resp. *zoomControls*, se v panelu *Control Dash* zobrazí menu editoru, resp. se zobrazí tlačítka přiblížení a oddálení grafu, avšak to se nestane. Během načítání grafu je ohlášena chyba „TypeError: a.editor.Editor is not a constructor“.

Export SVG Přestože v informacích o vydaných verzích Alchemy.js [17] je, že Alchemy.js od verze 0.4.0 obsahuje funkci pro export SVG, tak nikde není uvedeno (ani v dokumentaci Alchemy.js) jak tuto funkci zpřístupnit a použít.

Studiem zdrojového kódu Alchemy.js bylo zjištěno, že se daná funkce zpřístupňuje pomocí parametru *exportSVG*, který se nastaví na hodnotu *true*. Tím se zobrazí element, v panelu *Control Dash*, který umožňuje spustit funkci pro export SVG. Avšak výstupem této funkce je nespustitelný HTML kód. Při parsování tohoto HTML kódu je ohlášena chyba „XML Parsing Error: unclosed token“.

Zdroj dat Pokud se jako hodnota parametru *dataSource* použije přímo JSON objekt, a ne cesta k souboru, který obsahuje JSON objekt, tak přestanou fungovat některé vlastnosti Alchemy.js, ačkoli jsou oba způsoby uvedeny v dokumentaci Alchemy.js. Například přestanou fungovat následující vlastnosti:

- Při kliknutí na část panelu *Control Dash*, by se tento panel měl zobrazit celý, ale to se nestane. Tudíž není možné vykonávat žádné funkce, které se spouští přes panel *Control Dash*.
- Nelze vybrat žádný uzel, resp. převést uzel do stavu „selected“ pomocí kliknutí myši. Během provádění této operace je ohlášena chyba „TypeError: alchemy.get is undefined“.

5.1.2 Java EE

Java EE je platforma firmy Oracle určená pro vývoj a provoz podnikových webových aplikací [18]. Podstatné technologie, které obsahuje tato platforma, v kontextu této práce jsou následující:

- Contexts and Dependency Injection (CDI) 1.1
- Enterprise JavaBeans (EJB) 3.2

- Expression Language (EL) 3.0
- JavaServer Faces (JSF) 2.2

Než bude možné popsat výše uvedené technologie, je třeba vysvětlit pojem Java Bean.

5.1.2.1 Java Bean

Java Bean je znovupoužitelná softwarová komponenta, se kterou může být manipulováno pomocí vizuálních vývojových prostředí [19]. (V rámci aplikace Role Checker je tímto prostředím nástroj pro tvorbu webových stránek.) Java Bean komponenta je implementována jako třída jazyka Java a její typické rysy jsou následující:

Podpora introspekce umožňující vývojovým nástrojům analyzovat fungování komponenty

Podpora upravitelnosti umožňující nastavit chování komponenty

Podpora událostí umožňující komunikaci s komponentou

Podpora vlastností umožňující nastavit hodnoty určené pro programovou obsluhu komponenty

Podpora perzistence umožňující uložit stav komponenty a tento stav později načíst

5.1.2.2 CDI

CDI je technologie, která umožňuje vnést (inject) Java Bean komponentu do jiné Java Bean komponenty. Následující kód je příkladem vnesení, pomocí anotace *@Inject*, Java Bean komponenty s názvem *SecondBean* do Java Bean komponenty s názvem *FirstBean*, včetně zavolání veřejné metody s názvem *getValue*:

```
1  @Named(value="firstBean")
2  @SessionScoped
3  public class FirstBean implements Serializable {
4
5      @Inject
6      SecondBean secondBean;
7
8      public String getSecondBeanValue() {
9          return secondBean.getValue();
10     }
11
12 }
```

CDI Managed Bean CDI Managed Bean je komponenta Java Bean, která splňuje následující podmínky:

- Není to nestatická vnitřní třída.
- Je to konkrétní třída nebo je anotována anotací *@Decorator*.
- Není anotovaná komponentu-definující EJB anotací a není deklarována jako EJB Java Bean v souboru *ejb-jar.xml*.
- Má konstruktor bez parametrů nebo konstruktor anotovaný anotací *@Inject*.

Typická komponenta CDI Managed Bean je anotována anotací *@Named*, která umožňuje přiřadit komponentě jméno, přes které se komunikuje s touto komponentou z webových stránek používajících technologii JSF. Například následující kód by přiřadil komponentě CDI Managed Bean jméno *userBean*:

```
1 @Named(value="userBean")
```

Jak dlouho bude komponenta CDI Managed Bean uchovávat svůj stav závisí na jejím rozsahu (scope). Typy rozsahů, které jsou definované ve specifikaci CDI [20], jsou následující:

Dependent Výchozí rozsah, pokud není žádný jiný definován. Komponenta bude držet svůj stav stejně dlouho jako její klient (Java Bean komponenta, do které byla daná komponenta vnesena). Rozsah se také může přiřadit ke komponentě pomocí anotace *@Dependent*.

Request Drží stav po dobu jednoho HTTP požadavku. Přiřazuje se pomocí anotace *@RequestScoped*.

Session Drží stav po celou dobu platnosti uživatelova sezení (session). Přiřazuje se pomocí anotace *@SessionScoped*.

Application Drží stav po celou dobu běhu aplikace. Přiřazuje se pomocí anotace *@ApplicationScoped*.

Conversation Drží stav na základě omezení definovaných programátorem. Přiřazuje se pomocí anotace *@ConversationScoped*.

Následující kód je příkladem komponenty CDI Managed Bean, které je přiřazen rozsah *SessionScoped*:

```
1 @Named(value="userBean")
2 @SessionScoped
3 public class UserBean implements Serializable {
4 }
```

5.1.2.3 EJB

EJB je Java Bean komponenta určená pro byznys logiku aplikace [21]. EJB se vnáší do jiné Java Bean komponenty pomocí anotace @EJB, viz následující příklad:

```
@Named(value="userBean")
@SessionScoped
public class UserBean implements Serializable {

    @EJB
    UserManager userManager;

    public User getUser() {
        return userManager.findUser("smith");
    }

}
```

V aplikaci Role Checker je použita pouze EJB typu Session. Session EJB se dělí na následující tři typy:

Stateless neukládá svůj stav pro znovupoužití

Stateful ukládá svůj stav pro znovupoužití, pro každého klienta se vytvoří jiná instance

Singleton ukládá svůj stav pro znovupoužití, v aplikaci existuje pouze jedna instance

Následující kód je příkladem Stateful Session EJB:

```
@Stateful(name = "UserManager")
public class UserManager {

    UserDAO userDAO;

    public UserManager() {
        userDAO = new UserDAO();
    }

    public User findUser(String username) {
        return userDAO.find(username);
    }

}
```

5.1.2.4 EL

EL je technologie, která umožňuje prezentační vrstvě (webovým stránkám) komunikovat s komponentou Managed Bean. Následující kód je příkladem EL výrazu, který volá metodu *getUsername* komponenty Managed Bean s přiděleným jménem *userBean*:

```
1 #{userBean.getUsername()}
```

5.1.2.5 JSF

JSF je framework sloužící k vývoji webových aplikací, především jejich uživatelského rozhraní (webových stránek). JSF se skládá z následujících technologií:

- API pro reprezentaci komponent a spravování jejich stavu (správa událostí, validace na straně serveru, navigace stránek atd.)
- knihovny tagů pro přidávání komponent do webových stránek a pro navázání komponent na objekty na straně serveru

V následujícím kódu je JSF komponenta s názvem *h:commandButton*, která způsobí vykreslení tlačítka, na straně klienta, umožňujícího zavolat metodu s názvem *logIn* na straně serveru:

```
1 <h:commandButton id="logInButton"
2                  value="Log in"
3                  action="#{loginBean.logIn()}"/>
```

5.2 Architektura

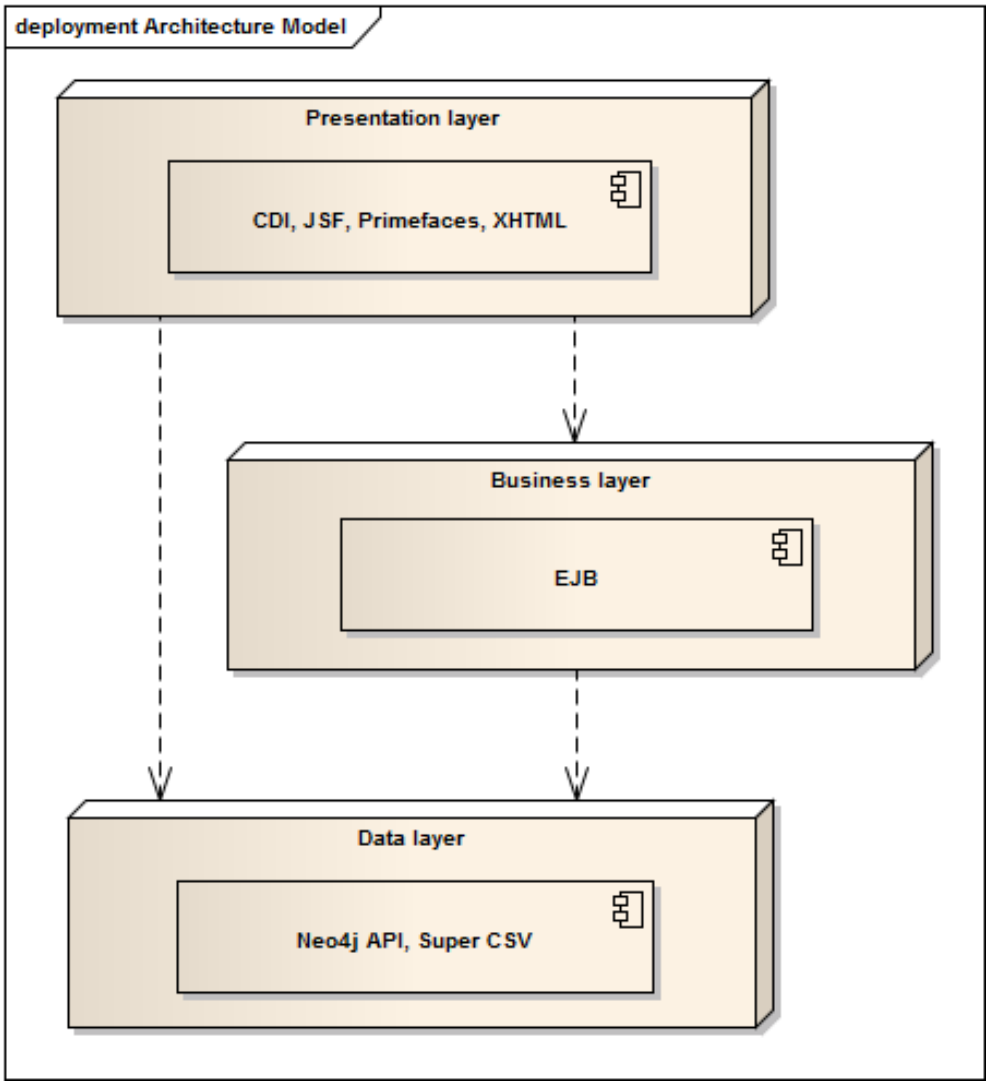
Na modelu architektury, viz obrázek 5.3, je vidět, že aplikace Role Checker má třívrstvou architekturu. U každé vrstvy jsou uvedeny některé technologie, které byly v této vrstvě použity.

Nejvýše je prezentační vrstva. Prezentační vrstva zajišťuje prezentování informací uživateli (ve webovém uživatelském rozhraní) a přijímání jeho požadavků. K těmto operacím využívá především XHTML stránky, které obsahují komponenty technologií JSF a Primefaces, a komponenty typu CDI Managed Bean, které implementují logiku zpracování a odesílání požadavků. Prezentační vrstva je přímo závislá na nižší byznys vrstvě, ale také na nejnižší datové vrstvě, ve které využívá komponenty pro práci se soubory.

Pod prezentační vrstvou je byznys vrstva. Byznys vrstva vykonává byznys logiku aplikace Role Checker (např. kontrolu rolí) a je závislá pouze na datové vrstvě.

Nejnižší vrstvou je datová vrstva. Datová vrstva zajišťuje správu datových zdrojů, tedy databází a souborů. Správa datových zdrojů zahrnuje hlavně:

komunikaci s databází, poskytování DAO (Data Access Object) objektů, provádění CRUD operací, parsování souborů a zapisování do souborů. Ke správě datových zdrojů v datové vrstvě jsou využívány především knihovny Neo4j API, pro práci s databází Neo4j, a Super CSV, pro práci s CSV soubory.



Obrázek 5.3: Model architektury

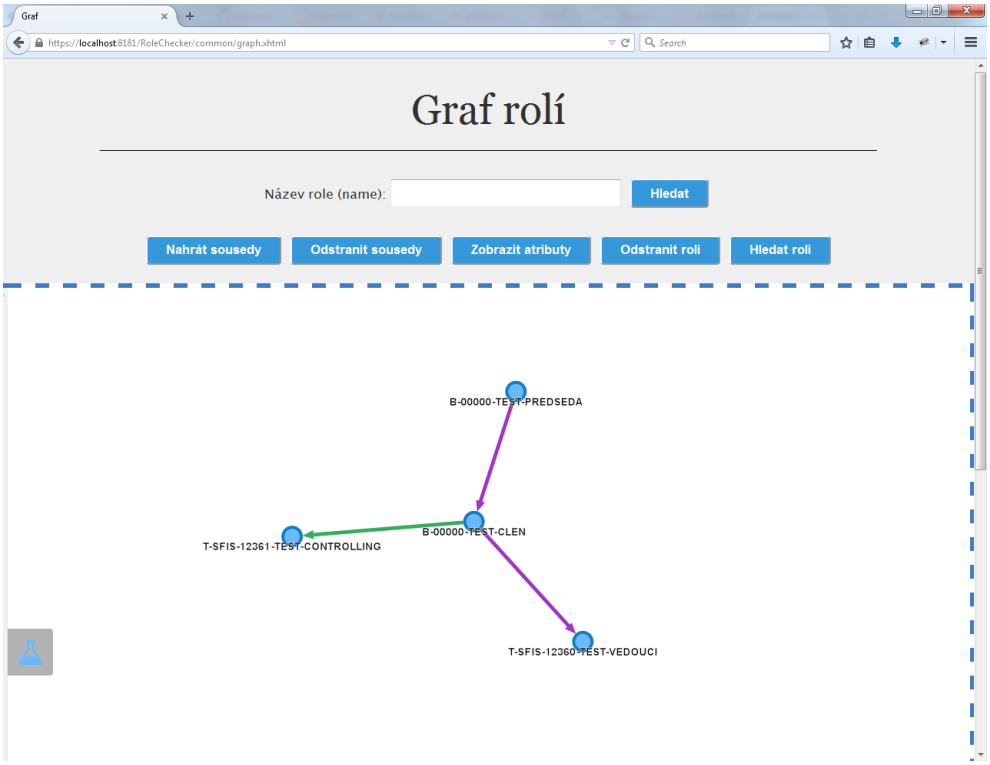
5.3 Zobrazení grafu rolí

Následující body zjednodušeně popisují hlavní části procesu zobrazení grafu rolí, podle vlastního CSV souboru rolí, v aplikaci Role Checker a jejich im-

plementaci (zdrojové kódy níže zmíněných objektů aplikace Role Checker lze najít v adresáři `src/impl` na přiloženém CD, viz příloha C):

- Uživatel vloží CSV soubor rolí pomocí webového rozhraní (viz *viewGraphComponent.xhtml*).
- Pomocí technologií JSF a CDI Managed Bean se provede nahrání CSV souboru na server (viz třída *ViewGraphBean*).
- EJB komponenta (*CsvConverter*) provede načtení CSV souboru, pomocí třídy *RoleParser* a knihovny Super CSV, transformování Java objektů rolí do objektů databáze Neo4j a trvalé uložení těchto objektů do databáze Neo4j, pomocí třídy *RoleDAO*.
- Uživateli se odešle zpráva o úspěšném dokončení nahrání a zpracování CSV souboru rolí.
- Uživatel pomocí webového rozhraní odešle požadavek ohledně načtení stránky pro zobrazení grafu (*graph.xhtml*).
- Server zpracuje požadavek a při následujících akcí je z databáze Neo4j načten podgraf obsahující jednu náhodnou roli. Tento podgraf, který je Neo4j objektem, se převede na Java objekt grafu, pomocí tříd *GraphRole*, *GraphData* a *GraphManager*. Java objekt grafu se převede na objekt typu GraphJSON, pomocí třídy *GraphjsonBuilder*, a tento objekt se zapíše do souboru.
- Server odešle uživateli stránku, která obsahuje URL (Uniform Resource Locator) souboru, kde je zapsán podgraf ve formátu GraphJSON, a také URL souborů knihovny Alchemy.js zajišťující zobrazení grafu na straně klienta.
- Uživatelovi se zobrazí stránka s vykresleným podgrafem, nad kterým může provádět různé operace.

Příklad zobrazení grafu, kde byli navíc načteni sousedi náhodně vybrané role, podle CSV souboru `sample_roles-version_2015_03_18.csv` lze vidět na obrázku 5.4. Popis funkcí, provádějících operace nad zobrazeným grafem, lze najít v uživatelském manuálu, viz příloha B.



Obrázek 5.4: Příklad zobrazení grafu rolí

5.4 Kontrola rolí

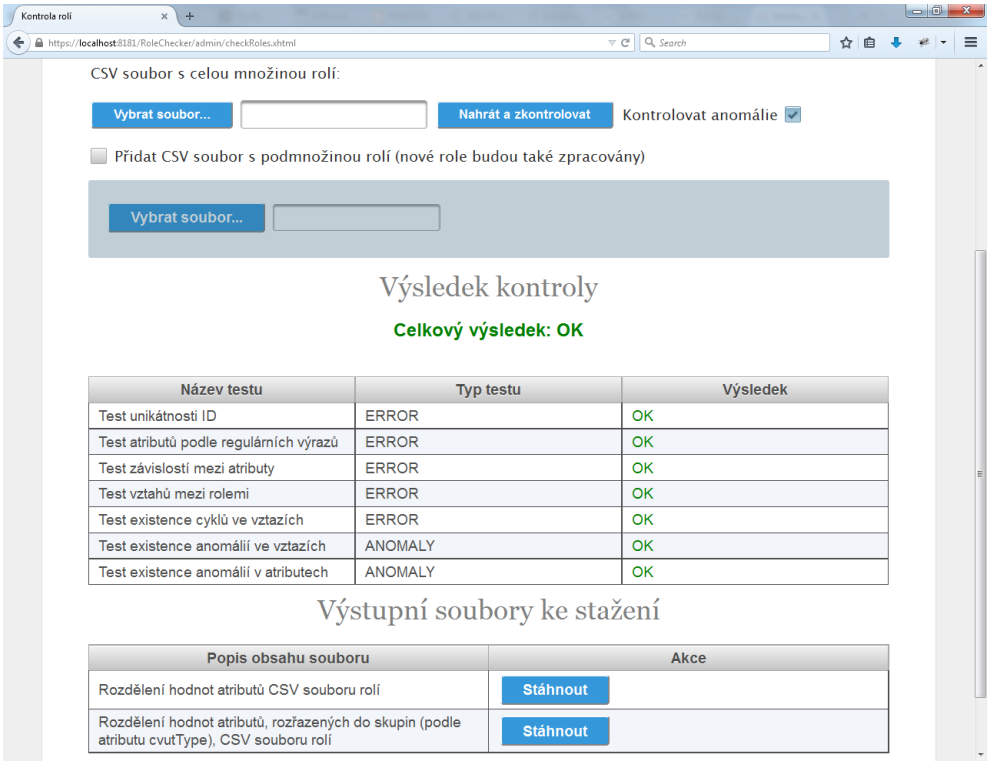
Následující body zjednodušeně popisují hlavní části procesu kontroly chyb a anomálií v rolích, na základě CSV souboru s celou množinou rolí, v aplikaci Role Checker a jejich implementaci (zdrojové kódy níže zmíněných objektů aplikace Role Checker lze najít v adresáři `src/impl` na přiloženém CD, viz příloha C):

- Uživatel vloží CSV soubor rolí pomocí webového rozhraní (viz *checkRolesComponent.xhtml*).
- Pomocí technologií JSF a CDI Managed Bean se provede nahrání CSV souboru na server (viz třída *CheckRolesBean*).
- Objekt třídy *Checker* provede načtení CSV souboru, pomocí třídy *RoleParser* a knihovny Super CSV.
- Pro každou roli se provedou první dva testy kontroly chyb, tedy test unikátnosti ID (atribut *name*) a test atributů podle regulárních výrazů, pomocí tříd *CheckRoleDAO* a *RegexChecker*. Pokud je během testu nalezena chyba, pak je uložena do databáze Derby, pomocí třídy *ErrorDAO*.

V případě, že během testu nebyla nalezena fatální chyba (definice fatální chyby viz soubor *ErrorType.java*), tak je role uložena do databáze Neo4j, jinak je role ignorována (pro další testování jakoby v CSV souboru nikdy nebyla). Po provedení předchozí akce je proveden třetí test kontroly chyb, tedy test závislostí mezi atributy, pomocí třídy *AttributeDependencyChecker*. Dále pak zpracování hodnot atributů role pro pozdější kontrolu anomálií, pomocí třídy *AnomalyManager*.

- Provedou se poslední dva testy kontroly chyb, tedy test vztahů mezi rolemi a test existence cyklů ve vztazích, pomocí třídy *CheckRoleDAO*.
- Provedou se dva testy kontroly anomálií, tedy test existence anomálií ve vztazích a test existence anomálií v atributech, pomocí třídy *AnomalyManager*.
- Nalezené chyby, resp. anomálie, které jsou uloženy v databázi Derby, jsou zapsány do CSV souboru chyb, resp. CSV souboru anomálií.
- Vytvoří se CSV soubor obsahující rozdělení hodnot (value distribution) atributů rolí a CSV soubor obsahující rozdělení hodnot atributů rolí na základě skupin (hodnot atributu *cvutType*).
- Server odešle uživateli stránku, ve které jsou uvedeny výsledky jednotlivých testů a komponenty umožňující stažení výstupních souborů. (Komponenty pro stažení CSV souborů obsahujících rozdělení hodnot jsou, při zahrnutí kontroly anomálií, k dispozici vždy. Komponenta pro stažení CSV souboru chyb, resp. komponenta pro stažení CSV souboru anomálií, je k dispozici pouze v případě, že byla nalezena aspoň jedna chyba, resp. v případě, že byla nalezena aspoň jedna anomálie.)

Výsledek kontroly rolí CSV souboru `sample_roles-version_2015_03_18.csv` lze vidět na obrázku 5.5.



Obrázek 5.5: Příklad výsledku kontroly rolí

Kapitola 6

Nasazení a testování

Tato kapitola popisuje nasazení a testování aplikace Role Checker v produkčním prostředí, ve VIC ČVUT. Aplikace byla nasazena na serveru a pomocí klienta, odlišného počítače od fyzického serveru, bylo provedeno testování přes webové rozhraní pomocí protokolů HTTP a HTTPS.

První sekce popisuje nasazení, včetně konfigurace serveru a klienta. Druhá sekce popisuje testování a výsledek testování.

6.1 Nasazení

Tato sekce obsahuje popis nasazení aplikace Role Checker. V první sekci je uvedena konfigurace serveru. Ve druhé sekci je uvedena konfigurace klienta. A nakonec ve třetí sekci je uvedena instalace aplikace Role Checker na serveru.

6.1.1 Konfigurace serveru

Mezi fyzickým serverem a aplikací Role Checker ještě běžel virtuální server. Pro virtualizaci byl na fyzickém serveru použit virtualizační nástroj VMware ESXi, verze 5.5.0. Virtuálnímu serveru byl přidělen hardware, který je uveden v tabulce 6.1. V tabulce 6.2 je vidět, že operačním systémem na virtuálním serveru byl Red Hat Enterprise Linux. V tomto operačním systému byla nainstalována platforma Oracle JDK, na které byl provozován aplikační server GlassFish Server Open Source Edition. Aplikace Role Checker běžela v tomto aplikačním serveru.

Tabulka 6.1: Konfigurace hardwaru přiděleného virtuálnímu serveru

Název komponenty	**Popis komponenty**
CPU (Central Processing Unit)	Intel(R) Xeon(R) CPU E7-2870 @ 2.4 GHz
RAM (Random-Access Memory)	8192 MB

Tabulka 6.2: Konfigurace softwaru na virtuálním serveru

Název komponenty	**Popis komponenty**
Operační systém	Red Hat Enterprise Linux, verze 6, 64bit
JDK	Oracle JDK, verze 1.8.0_45
Aplikační server	GlassFish Server Open Source Edition, verze 4.1

6.1.2 Konfigurace klienta

V tabulce 6.3 je vidět, že na klientovi běžel operační systém Microsoft Windows 8.1. V tomto operačním systému běžel webový prohlížeč Mozilla Firefox. V prohlížeči Mozilla Firefox, obsahujícím zásuvný modul Selenium IDE, byly spouštěny testy.

Tabulka 6.3: Konfigurace softwaru na klientovi

Název komponenty	**Popis komponenty**
Operační systém	Microsoft Windows 8.1
Webový prohlížeč	Mozilla Firefox, verze 38.0.5
Zásuvný modul	Selenium IDE, verze 2.9.0

6.1.3 Instalace

Postup instalace aplikace Role Checker na server byl následující:

1. Obsah adresářů `application` a `glassfish_server`, které lze najít na přiloženém CD, viz příloha C, byl zkopírován do adresáře `/opt`.

2. Byl spuštěn databázový server zadáním následujících příkazů do příkazové řádky:

   ```
   cd /opt/rolechecker/rolechecker/derby

   java -classpath '/opt/rolechecker/lib/apache-derby' \
   -jar '/opt/rolechecker/lib/apache-derby/derbyrun.jar' \
   server start -h localhost -p 1527 &
   ```

3. Byl spuštěn aplikační server GlassFish Server zadáním následujících příkazů do příkazové řádky:

   ```
   cd /opt/glassfish/bin

   ./asadmin --user admin --passwordfile password.file \
   start-domain rolechecker-domain
   ```

4. Aplikace Role Checker byla nasazena (deploy) na server GlassFish Server zadáním následujících příkazů do příkazové řádky:

```
cd /opt/glassfish/bin

./asadmin --user admin --passwordfile password.file \
deploy '/opt/rolechecker/RoleChecker.war'
```

Po vykonání výše uvedených akcí byla aplikace Role Checker plně funkční a připravena přijímat požadavky na adrese `https://role.is.cvut.cz/`, v privátní síti VIC ČVUT.

6.2 Testování

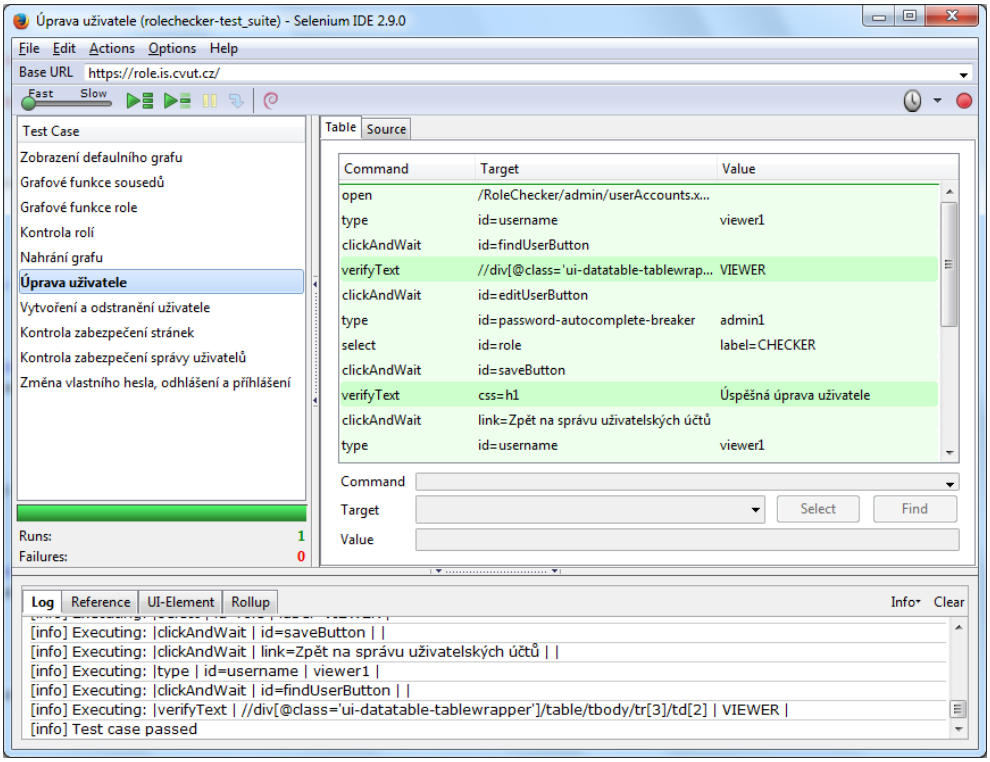
Tato sekce obsahuje formu testování a výsledek testování aplikace Role Checker. Před testováním byl do aplikace nahrán CSV souboru všech rolí z IDM, k datu 18. 3. 2015, pomocí funkce pro nahrání defaultního souboru. Testy byly spouštěny na klientovi pomocí webového prohlížeče Mozilla Firefox a zásuvného modulu Selenium IDE (verze obou aplikací jsou uvedeny v tabulce 6.3).

Nejdříve bude popsán nástroj Selenium IDE, který byl použit k testování. Poté budou uvedeny jednotlivé testy. A nakonec bude popsán výsledek testování.

6.2.1 Selenium IDE

Selenium IDE patří mezi nástroje projektu Selenium. Nástroje projektu Selenium slouží k automatickému vykonávání webových operací a testování webových elementů. Selenium IDE je integrované vývojové prostředí pro vytváření a spouštění skriptů obsahujících příkazy, které provádí výše zmíněné webové akce. Instaluje se ve formě zásuvného modulu do prohlížeče Mozilla Firefox. [22]

Z nástrojů projektu Selenium bylo Selenium IDE vybráno k testování aplikace Role Checker hlavně díky rychlému a jednoduchému vytváření automatických testů. Na obrázku 6.1 je ukázáno okno programu Selenium IDE po vykonání testu s názvem „Úprava uživatele“.



Obrázek 6.1: Okno programu Selenium IDE

6.2.1.1 Příkazy

V této sekci jsou popsány některé základní příkazy nástroje Selenium IDE. Hlavním jazykem skriptů pro Selenium IDE je HTML, a proto i následující příkazy budou v jazyku HTML.

- Níže uvedený příkaz *open* vykoná přechod, v aktuálním okně prohlížeče, na webovou stránku podle URL `http://www.seznam.cz/`, která mu byla zadána jako parametr.

```
<tr>
    <td>open</td>
    <td>http://www.seznam.cz/</td>
    <td></td>
</tr>
```

- Níže uvedený příkaz *veirfyText* ověří, že na aktuální stránce v některém elementu *h1* je obsažen text „Graf“.

```
<tr>
    <td>verifyText</td>
```

```
3     <td>css=h1</td>
4     <td>Graf</td>
5 </tr>
```

- Níže uvedený příkaz *clickAndWait* provede akci kliknutí pro element s hodnotou atributu *id* „loginButton“. Pokud akce kliknutí způsobí nahrání nové stránky, pak se počká až se nahraje stránka celá.

```
1 <tr>
2     <td>clickAndWait</td>
3     <td>id=loginButton</td>
4     <td></td>
5 </tr>
```

6.2.2 Testy

Každý test je skript pro program Selenium IDE, který má ověřit správnou funkčnost určité vlastnosti aplikace Role Checker. V rámci testování byly provedeny následující testy:

Zobrazení defaultního grafu ověřuje správné zobrazení grafu podle předem nahraného defaultního CSV souboru rolí

Grafové funkce sousedů ověřuje správné fungování funkcí pro nahrání a odstranění sousedních rolí (uzlů)

Grafové funkce role ověřuje správné fungování funkcí pro zobrazení všech atributů role a vyhledání role v aktuálně zobrazeném grafu

Kontrola rolí ověřuje správnou funkčnost kontroly rolí, včetně zpracování CSV souborů a zobrazení výstupních parametrů

Nahrání grafu ověřuje správné zpracování vlastního CSV souboru rolí a jeho transformaci do grafové databáze

Úprava uživatele ověřuje správné fungování funkcí pro úpravu uživatele

Vytvoření a odstranění uživatele ověřuje správnost funkcí pro vytvoření a odstranění uživatele

Kontrola zabezpečení stránek ověřuje správné zabezpečení webových stránek, hlavně nepovolené přístupy uživatelů na určité stránky

Kontrola zabezpečení správy uživatelů ověřuje správné zabezpečení operací nad uživateli

Změna vlastního hesla, odhlášení a přihlášení ověřuje správné fungování funkcí pro změnu vlastního hesla, odhlášení uživatele a přihlášení uživatele

Na přiloženém CD lze najít zdrojové kódy všech testů v adresáři `selenium_ide_tests`, viz příloha C.

6.2.3 Výsledek testování

Všechny testy proběhly v pořádku, žádná chyba nebyla nalezena.

Závěr

V úvodu byl popsán cíl této práce, který se týká analýzy IDM rolí a vytvoření aplikace pro vizualizaci a kontrolu rolí. Tento cíl byl splněn. V prvních dvou kapitolách je uvedena analýza IDM a rolí. Následující tři kapitoly popisují specifikaci požadavků, návrh a implementaci aplikace. Poslední kapitola obsahuje popis úspěšného nasazení a testování této aplikace v produkčním prostředí, ve VIC ČVUT.

Přínosem této práce je možnost lepšího pochopení rolí a jejich závislostí díky podrobnému popisu rolí a zobrazení grafu rolí v aplikaci. Dále pak možnost odstranění chyb pomocí kontroly rolí v aplikaci.

Nejnáročnější byla implementace zobrazení grafu rolí kvůli problémům spojeným s knihovnou Alchemy.js (viz sekce 5.1.1.2). I přes tyto problémy se zdá, že použití této knihovny bylo nejlepším řešením, protože jediné dvě srovnatelné knihovny, které byly nalezeny, měly oproti knihovně Alchemy.js velké nevýhody. První knihovna s názvem D3.js nebyla určena přímo pro grafy, takže by bylo třeba implementovat i většinu základních funkcí zobrazení grafu. Druhá knihovna KeyLines Starter Edition měla cenu licence 10 000 £, přitom knihovna Alchemy.js je zcela zdarma.

Tématem případné navazující práce by mohla být implementace knihovny pro vizualizaci grafu IDM rolí ve webovém uživatelském rozhraní, která by obsahovala pokročilé funkce. Pojmem pokročilé funkce je myšleno hlavně: nahrání a odstranění většiny rolí bez komunikace se serverem (tedy možnost zpracování velkého grafu na straně klienta a jeho postupné zobrazování), filtrování podle různých atributů role, uživatelsky přívětivá funkce hledání cest v zobrazeném grafu a možnost nastavení mnoha různých stylů grafu.

Literatura

[1] GraphAlchemist: *Alchemy.js*. [cit. 2015-06-21]. Dostupné z WWW: `http://graphalchemist.github.io/Alchemy/#/`.

[2] Skip Slone & The Open Group Identity Management Work Area: *Identity Management*. Document No.: W041. The Open Group, 2004.

[3] Computerworld: *Identity management – centrální správa uživatelských účtů [online]*. [cit. 2014-10-21]. Dostupné z WWW: `http://computerworld.cz/securityworld/identity-management-centralni-sprava-uzivatelskych-uctu-47568`.

[4] Kolář, J.: *Teoretická informatika*. Praha, Česká informatická společnost, 2000.

[5] Bachman, M.: *GraphAware: Towards Online Analytical Processing in Graph Databases*. Londýn, Imperial College London, 2013.

[6] Emison, J. M.: *2014 State Of Database Tech*. InformationWeek, 2014.

[7] Neo Technology, Inc.: *What is a Graph Database? - Neo4j Graph Database [online]*. [cit. 2015-04-17]. Dostupné z WWW: `http://neo4j.com/developer/graph-database/`.

[8] Neo Technology, Inc.: *Neo4j 2.1.8 API [online]*. [cit. 2015-04-17]. Dostupné z WWW: `http://neo4j.com/docs/2.1.8/javadocs/`.

[9] Wrox Press Ltd.: *ACID properties [online]*. [cit. 2015-04-17]. Dostupné z WWW: `https://msdn.microsoft.com/en-us/library/aa480360.aspx`.

[10] The Neo4j Team: *The Neo4j Manual v2.1.8*. Neo Technology, Inc., 2015.

[11] Oracle: *Oracle Waveset 8.1.1 Overview*. Part No: 821–0762. 2010.

[12] Oracle: *Oracle Waveset 8.1.1 Business Administrator's Guide*. Part No: 821–0094. 2010.

[13] Oracle: *java.util.regex (Java Platform SE 7) [online]*. [cit. 2015-03-19]. Dostupné z WWW: `http://docs.oracle.com/javase/7/docs/api/java/util/regex/package-summary.html`.

[14] Shafranovich Y.: *RFC 4180 - Common Format and MIME Type for Comma-Separated Values (CSV) Files*. 2005. Dostupné z WWW: `http://tools.ietf.org/html/rfc4180`.

[15] The Apache Software Foundation: *Apache Commons – Apache Commons*. [cit. 2015-06-21]. Dostupné z WWW: `https://commons.apache.org/`.

[16] GraphAlchemist: *Alchemy.js*. [cit. 2015-06-21]. Dostupné z WWW: `http://graphalchemist.github.io/Alchemy/#/docs`.

[17] GraphAlchemist: *Releases - GraphAlchemist/Alchemy - GitHub*. [cit. 2015-06-21]. Dostupné z WWW: `https://github.com/GraphAlchemist/Alchemy/releases`.

[18] Oracle: *Introduction to Java Platform, Enterprise Edition 7*. Oracle White Paper. 2013.

[19] Sun Microsystems: *JavaBeans*. JavaBeans API Specification 1.01.

[20] Oracle: *Contexts and Dependency Injection for the Java EE platform*. Contexts and Dependency Injection 1.1 (JSR-346).

[21] Oracle: *JSR 345: Enterprise JavaBeansTM, Version 3.2*. EJB Core Contracts and Requirements. 2013.

[22] Selenium Project: *Introduction – Selenium Documentation*. [cit. 2015-06-20]. Dostupné z WWW: `http://www.seleniumhq.org/docs/01_introducing_selenium.jsp#introducing-selenium`.

Příloha **A**

Seznam použitých zkratek

API Application Programming Interface

CDI Contexts and Dependency Injection

CPU Central Processing Unit

CRUD Create Read Update Delete

CSV Comma-Separated Values

CTU Czech Technical University in Prague

ČVUT České vysoké učení technické v Praze

DAO Data Access Object

EJB Enterprise JavaBeans

EL Expression Language

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

ID Identifikátor

IDM Identity Manager

IT Informační technologie

Java EE Java Platform, Enterprise Edition

Java SE Java Platform, Standard Edition

JDK Java SE Development Kit

JSF JavaServer Faces

JSON JavaScript Object Notation

JVM Java Virtual Machine

NoSQL Not only SQL

OS Operační systém

RAM Random-Access Memory

RBAC Role-Based Access Control

REST Representational State Transfer

SQL Standard Query Language

UTF-8 UCS Transformation Format – 8-bit

URL Uniform Resource Locator

VIC Výpočetní informační centrum

XML Extensible Markup Language

XSL Extensible Stylesheet Language

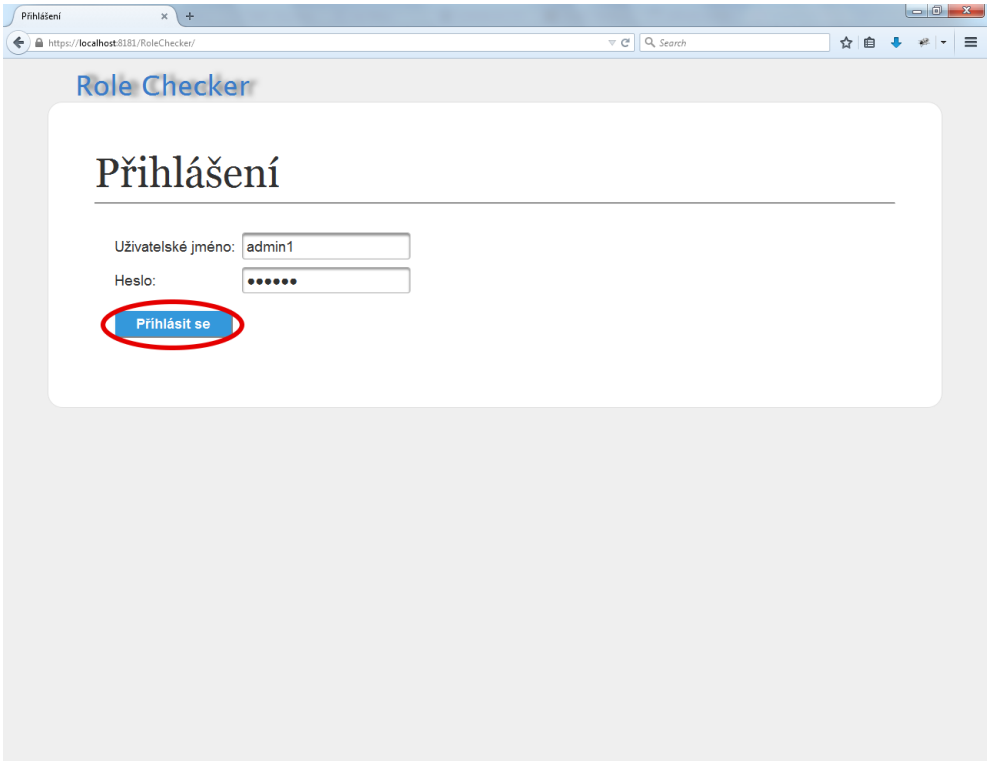
Příloha B

Uživatelský manuál

Toto je uživatelský manuál aplikace Role Checker (dále jen „aplikace“), verze 1.0. V každé následující sekci je popis jedné akce aplikace. První sekce popisuje přihlášení k aplikaci. Druhá sekce popisuje zobrazení grafu rolí. Třetí sekce popisuje kontrolu rolí. Čtvrtá sekce popisuje nastavení aplikace. Pátá sekce popisuje akci nápovědy. (Některé akce nemusí být dostupné nebo můžou být zakázané v závislosti na uživatelské roli, která vám byla přidělena.)

B.1 Přihlášení

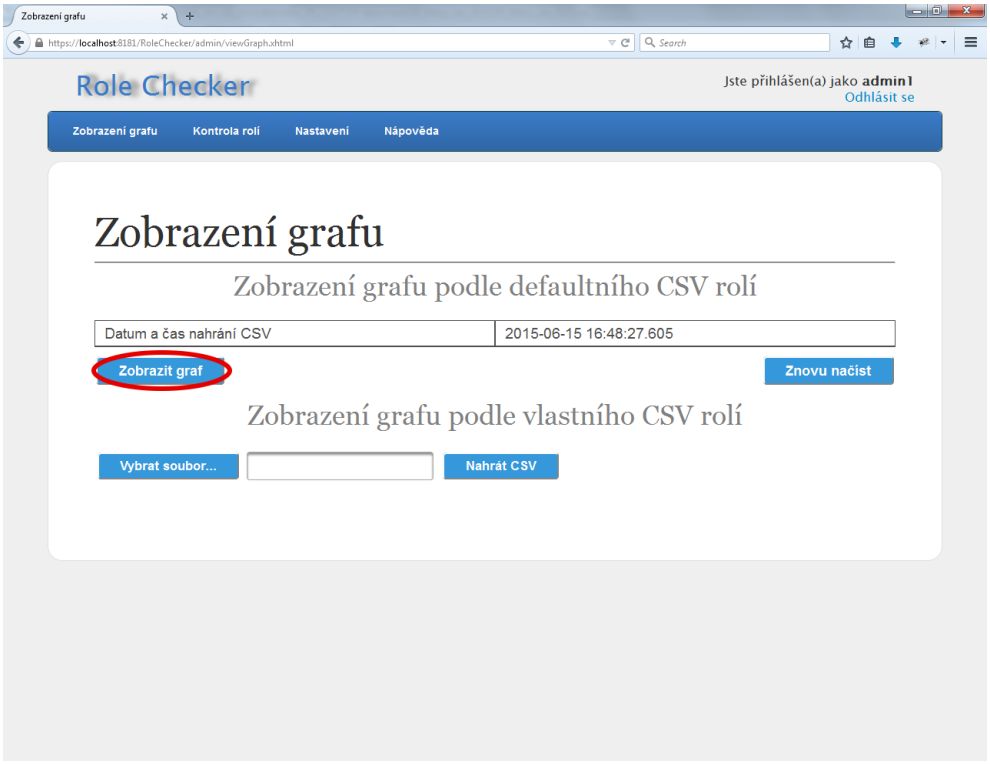
Zadejte vaše uživatelské jméno a heslo, a poté klikněte na tlačítko s nápisem „Přihlásit se“, viz obrázek B.1. Dále pokračujte jednou ze čtyř níže uvedených sekcí v závislosti na vámi vybrané akci.



Obrázek B.1: Přihlášení

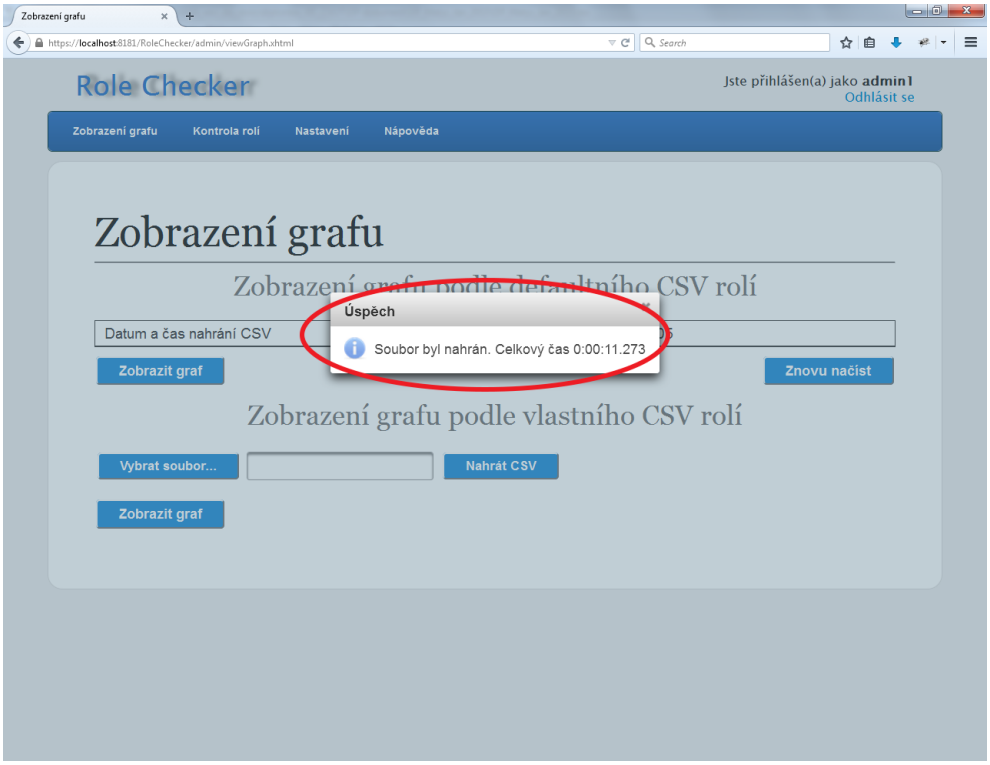
B.2 Zobrazení grafu rolí

Po přihlášení se zobrazí stránka s nadpisem „Zobrazení grafu“. Pokud se v části s podnadpisem „Zobrazení grafu podle defaultního CSV rolí“ zobrazí tlačítko s názvem „Zobrazit graf“, viz obrázek B.2, pak můžete zobrazit graf rolí, z už nahraného defaultního CSV souboru rolí, kliknutím na toto tlačítko.

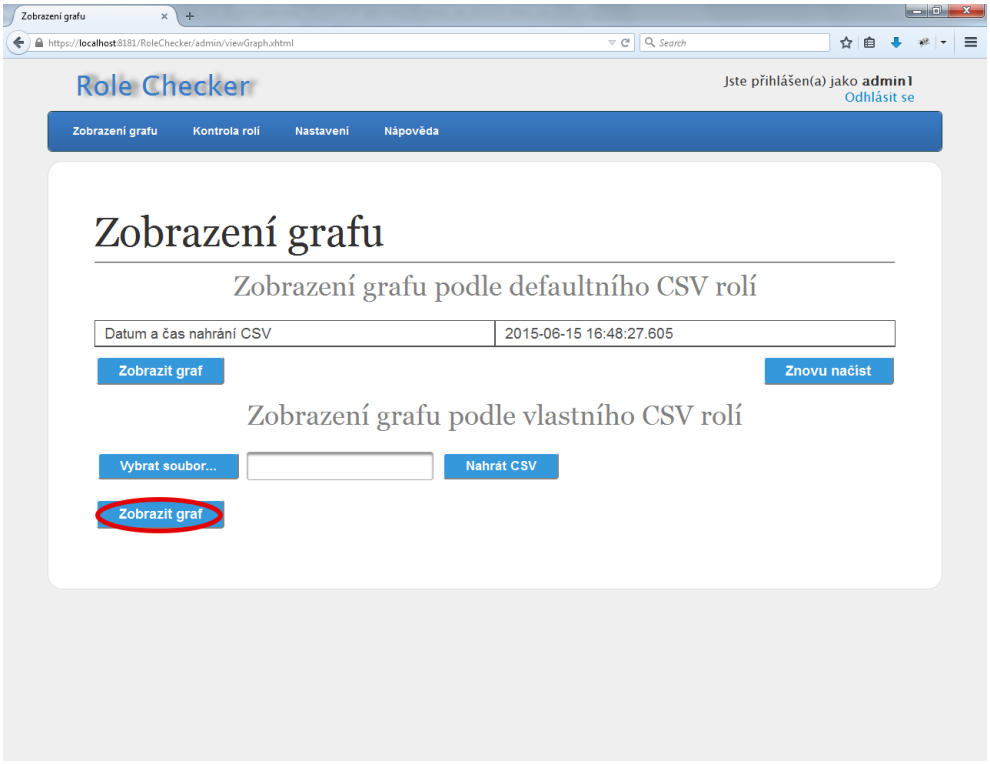


Obrázek B.2: Zobrazení grafu podle defaultního CSV rolí

Pokud se toto tlačítko nezobrazí, pak defaultní CSV soubor není k dispozici a je třeba nahrát vlastní CSV soubor rolí. Klikněte na tlačítko s názvem „Vybrat soubor...“ a vyberte CSV soubor s rolemi z IDM. Poté klikněte na tlačítko s názvem „Nahrát CSV“ a počkejte až proběhne nahrání a zpracování souboru. Po úspěšném nahrání a zpracování souboru se objeví dialog oznamující tuto informaci a celkový čas zpracování CSV souboru, viz obrázek B.3. Tento dialog zavřete a klikněte na tlačítko s názvem „Zobrazit graf“, které se objevilo pod tlačítkem s názvem „Vybrat soubor...“, v části s podnadpisem „Zobrazení grafu podle vlastního CSV rolí“, viz obrázek B.4.



Obrázek B.3: Dialog oznamující úspěšné nahrání a zpracování souboru



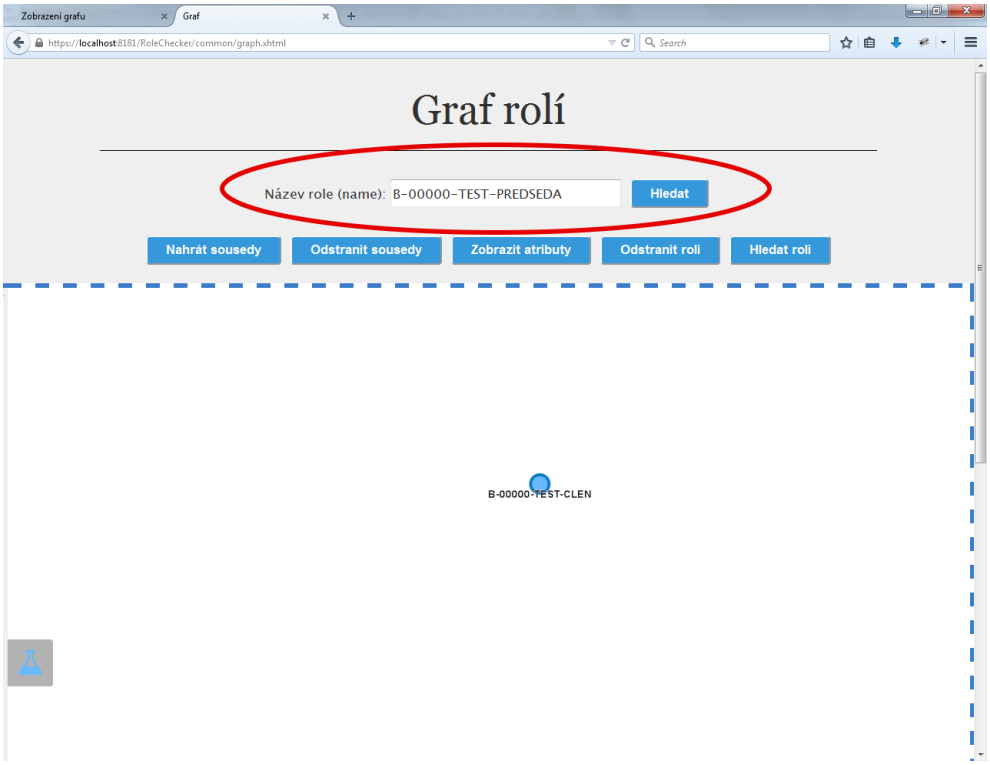
Obrázek B.4: Zobrazení grafu podle vlastního CSV rolí

B.2.1 Graf rolí

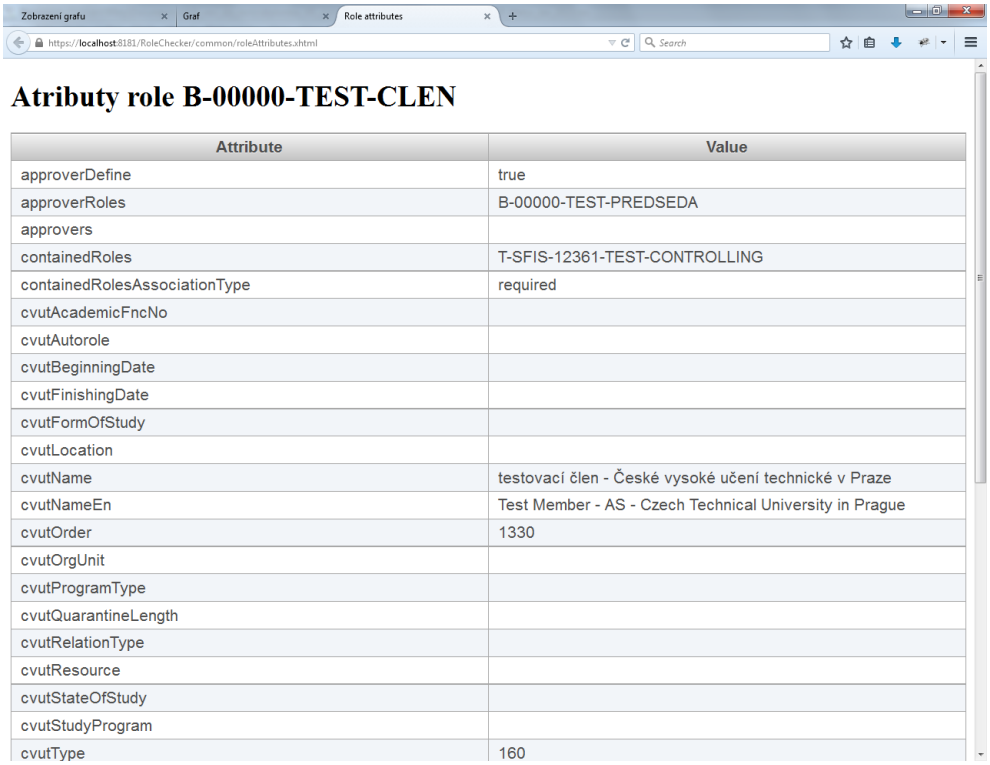
Kliknutím na jedno z výše zmíněných tlačítek s názvem „Zobrazit graf“ se otevře nové okno, ve kterém se zobrazí stránka s nadpisem „Graf rolí“. V horní části stránky je tlačítko s názvem „Hledat“, které umožňuje vyhledat v databázi roli podle jejího názvu (hodnota musí být zadána přesně, záleží na velikosti písmen), resp. hodnoty atributu *name*, viz obrázek B.5. Pod tímto tlačítkem jsou tlačítka, která umožňují následující akce:

1. nahrání sousedních rolí právě vybrané role do zobrazeného grafu
2. odstranění sousedních rolí právě vybrané role zobrazeného grafu
3. zobrazení atributů (v novém okně) právě vybrané role, viz obrázek B.6
4. odstranění aktuálně vybrané role, včetně vztahů této role
5. vyhledání role, podle hodnoty atributu *name*, v aktuálně zobrazeném grafu (stačí zadat část hodnoty atributu *name* a nezáleží na velikosti písmen) – nalezené role budou v grafu označeny oranžovou barvou, viz obrázek B.7

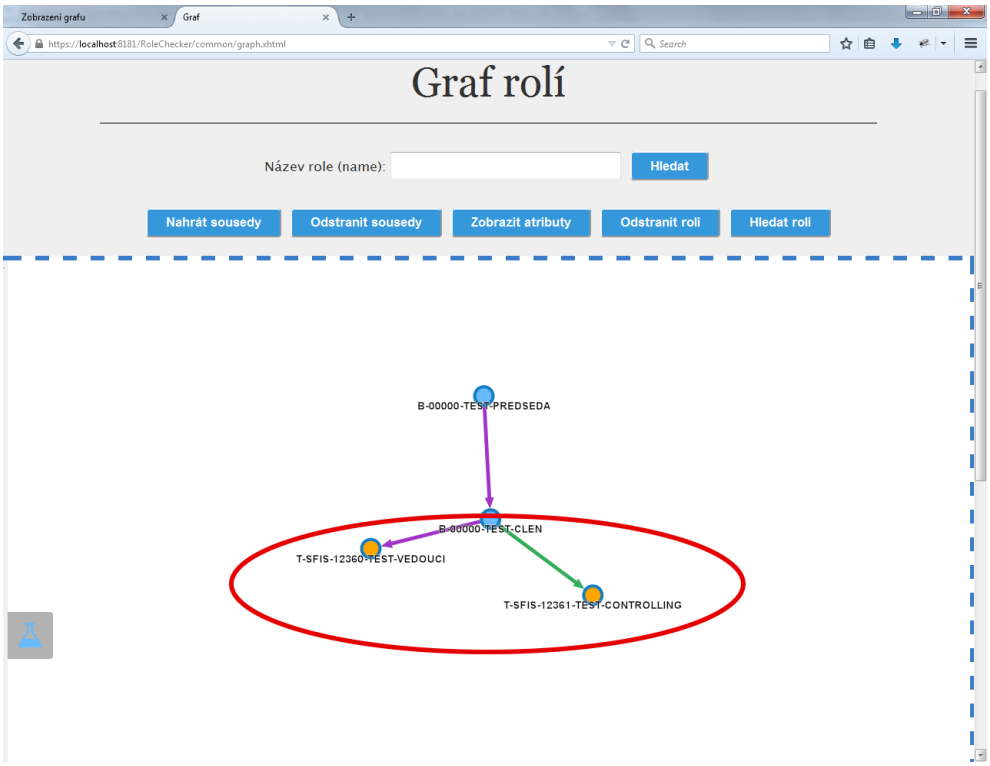
V dolní části stránky jsou tlačítka, která umožňují stažení aktuálně zobrazeného grafu ve formátech SVG a PDF, a legenda popisující význam vztahů v grafu, viz obrázek B.8.



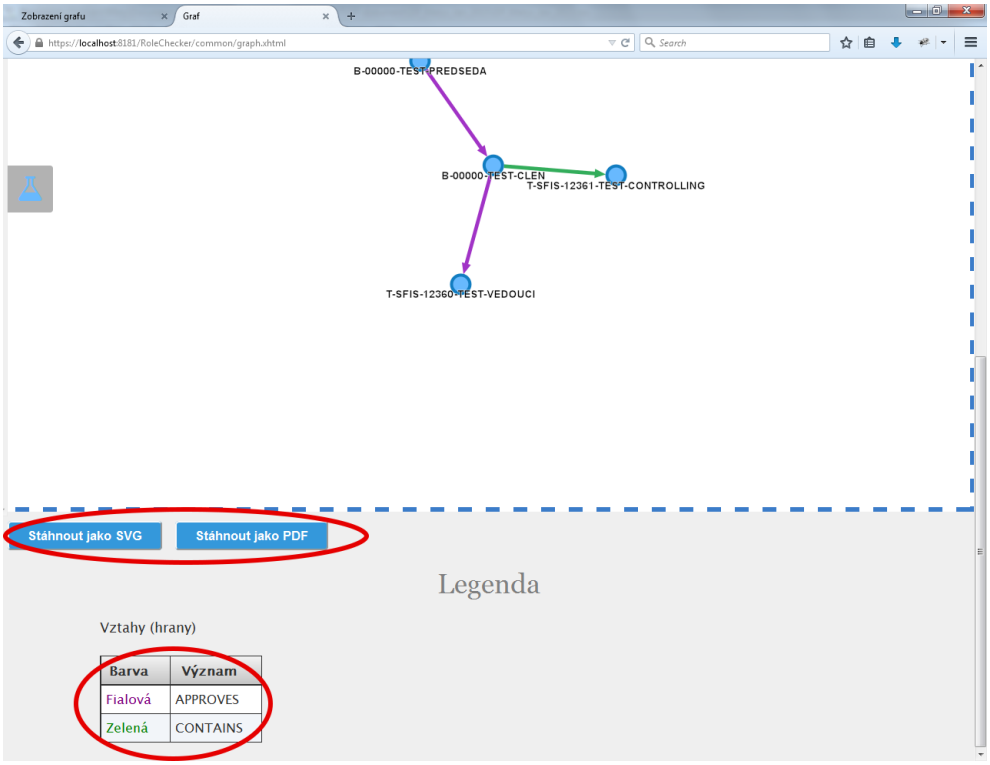
Obrázek B.5: Vyhledání role v databázi



Obrázek B.6: Atributy role



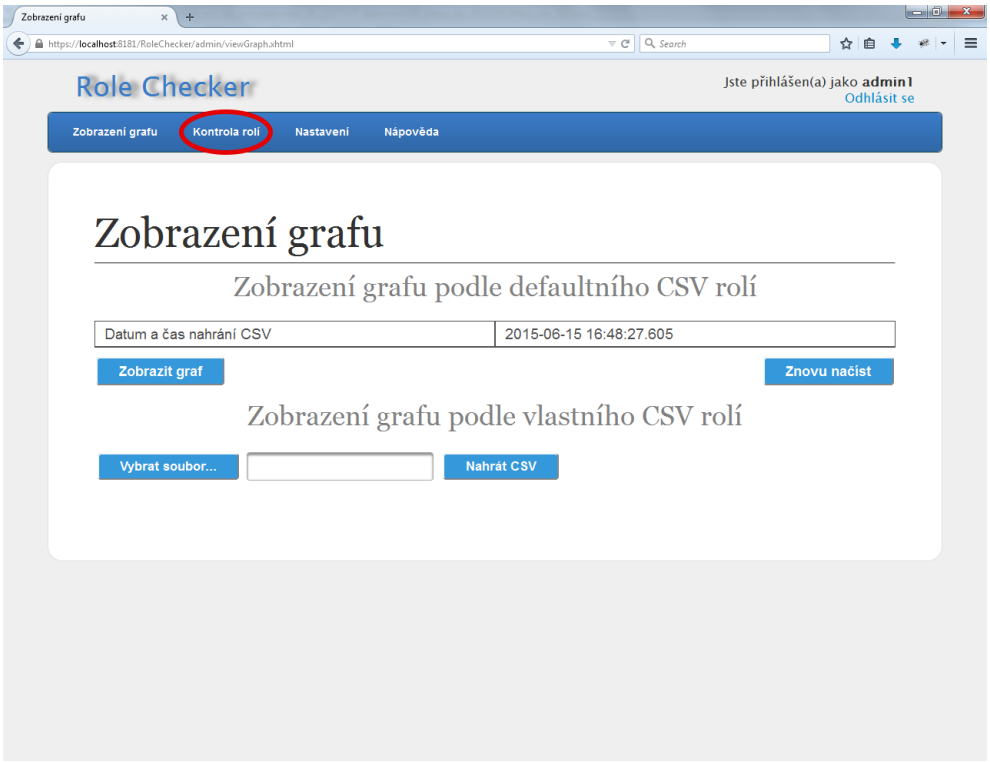
Obrázek B.7: Nalezené role v aktuálně zobrazeném grafu pomocí tlačítka s názvem „Hledat roli“ (byla zadána hodnota „sfis“)



Obrázek B.8: Tlačítka pro stažení grafu ve formátech SVG a PDF a legenda

B.3 Kontrola rolí

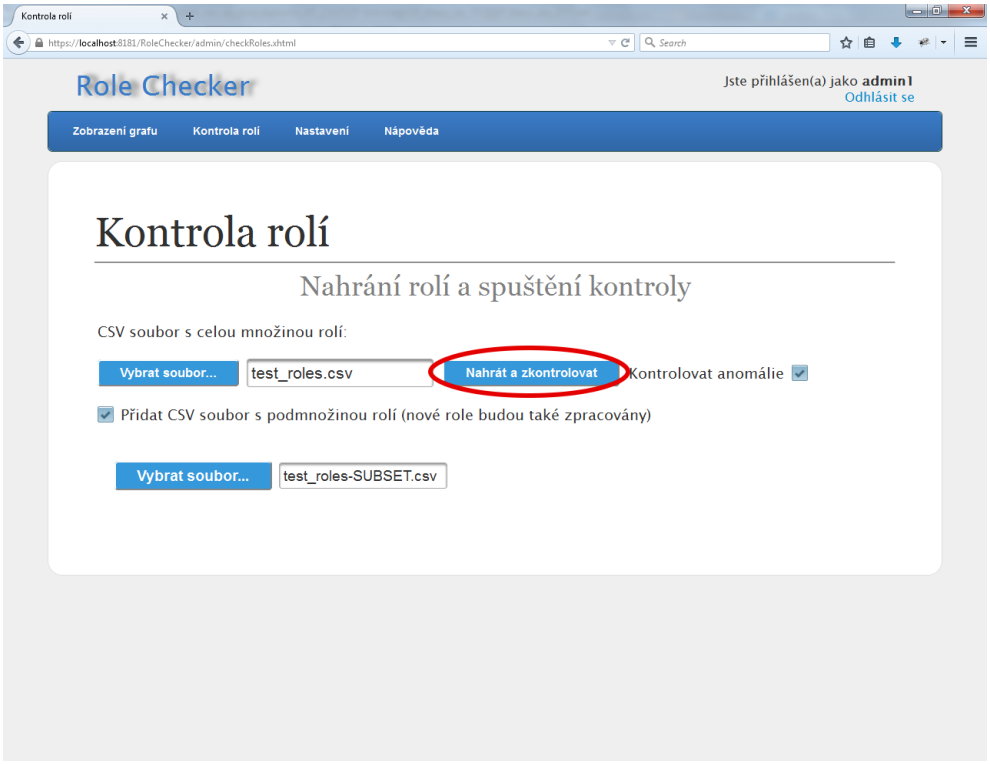
Po přihlášení klikněte v horním menu na položku s názvem „Kontrola rolí“, viz obrázek B.9.



Obrázek B.9: Položka „Kontrola rolí“ v horním menu

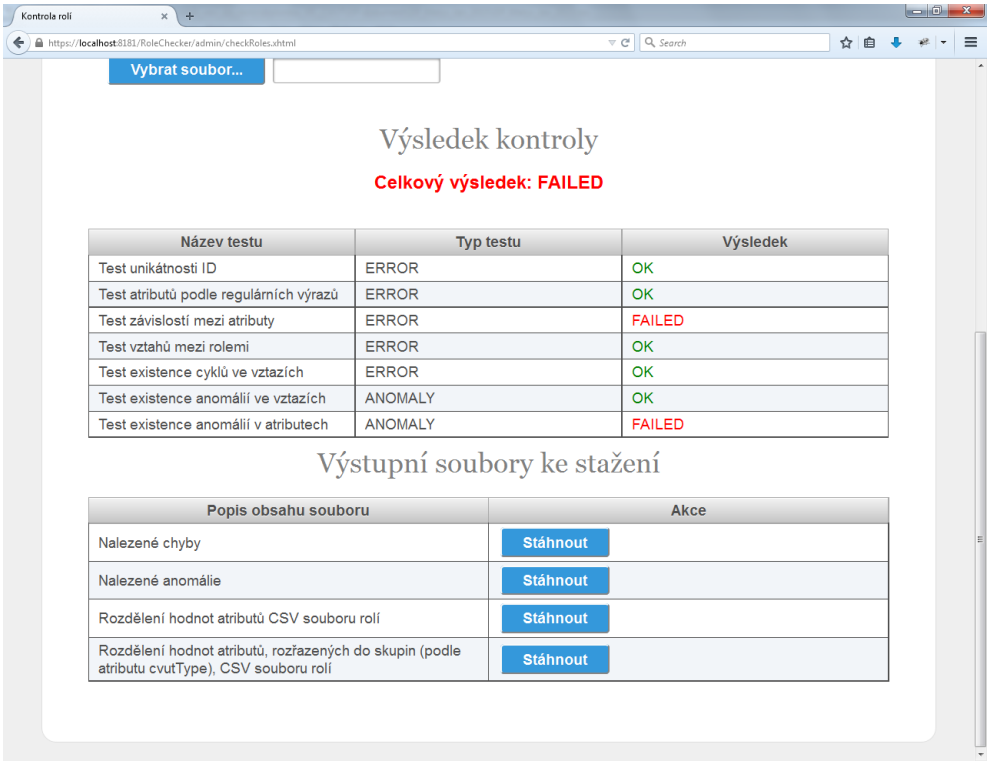
Nyní se vám zobrazí stránka s nadpisem „Kontrola rolí“. Pomocí parametrů nastavení můžete volit mezi různými typy kontrol. V tomto návodu bude uvedena kontrola, zahrnující chyby i anomálie, CSV souboru s celou množinou rolí a CSV souboru s podmnožinou rolí (kontrola s podmnožinou rolí je užitečná v případě, kdy jste upravovali jen podmnožinu rolí a chcete vědět, zda jsou tyto role korektní v rámci celé množiny rolí předtím, než provedete import podmnožiny rolí do IDM).

Klikněte na tlačítko s názvem „Vybrat soubor...“ těsně pod nápisem „CSV soubor s celou množinou rolí:“ a vyberte soubor s celou množinou rolí. Zaškrtněte kontrolu anomálií. Dále zaškrtněte přidání CSV souboru s podmnožinou rolí, a poté klikněte na dolní tlačítko s názvem „Vybrat soubor...“ a vyberte CSV soubor s podmnožinou rolí. Nakonec klikněte na tlačítko s názvem „Nahrát a zkontrolovat“, viz obrázek B.10, a počkejte až se dokončí kontrola. (V případě, že jeden z CSV souborů měl 40 000 rolí a více, může kontrola trvat více než 45 minut.)



Obrázek B.10: Tlačítko „Nahrát a zkontrolovat“

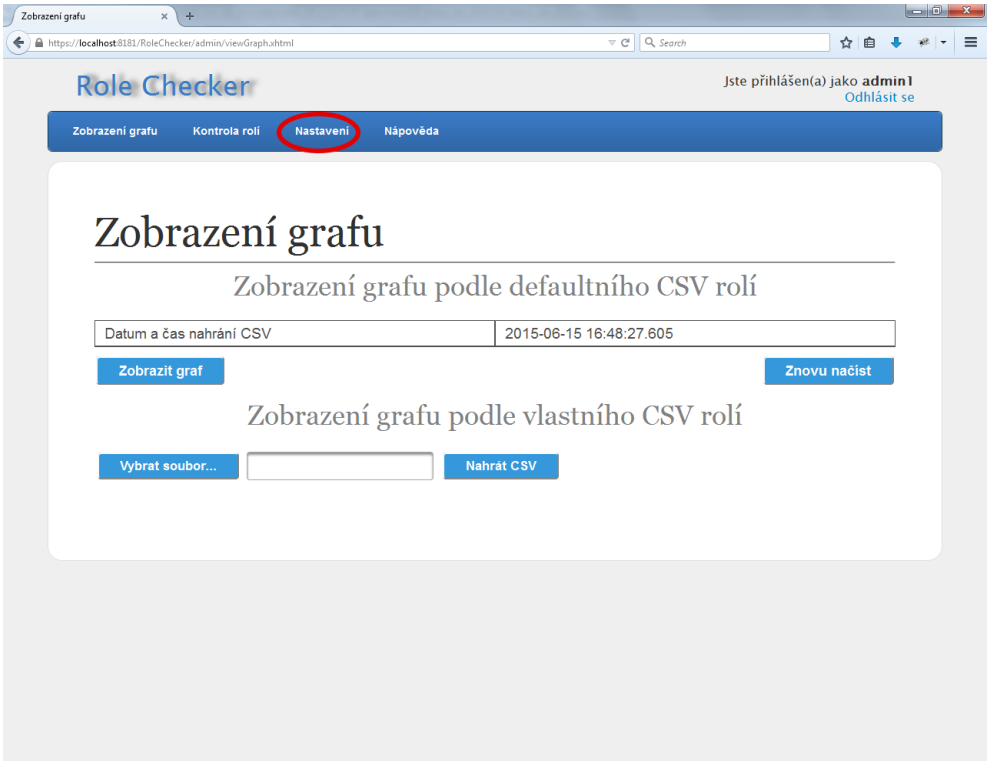
Po dokončení kontroly a zavření dialogu, oznamujícího úspěšné ukončení kontroly, se v dolní části stránky zobrazí výsledky kontroly a výstupní soubory ke stažení, viz obrázek B.11. Nabídka výstupních souborů ke stažení závisí na nastavených parametrech kontroly rolí a také na tom, zda byla při kontrole nalezena aspoň jedna chyba, resp. aspoň jedna anomálie (soubor obsahující chyby, resp. anomálie, je nabídnut ke stažení jen v případě, že byla nalezena aspoň jedna chyba, resp. aspoň jedna anomálie). Kliknutím na tlačítko s názvem „Stáhnout“ v řádce příslušného souboru se daný soubor stáhne do vašeho počítače.



Obrázek B.11: Výsledek kontroly rolí a výstupní soubory ke stažení

B.4 Nastavení

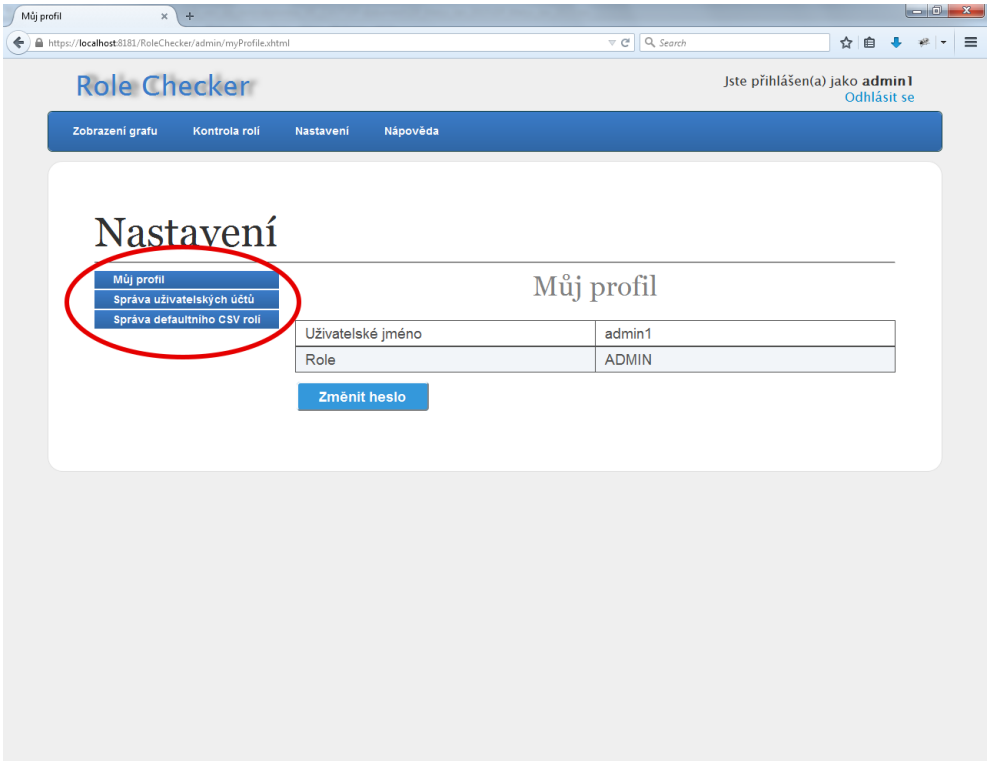
Po přihlášení klikněte v horním menu na položku s názvem „Nastavení“, viz obrázek B.12.



Obrázek B.12: Položka „Nastavení“ v horním menu

Nyní se zobrazí stránka s nadpisem „Nastavení“. V levém postranním menu, viz obrázek B.13, si vyberte, kliknutím na danou položku, jednu z následujících akcí:

1. můj profil
2. správa uživatelských účtů
3. správa defaultního CSV rolí

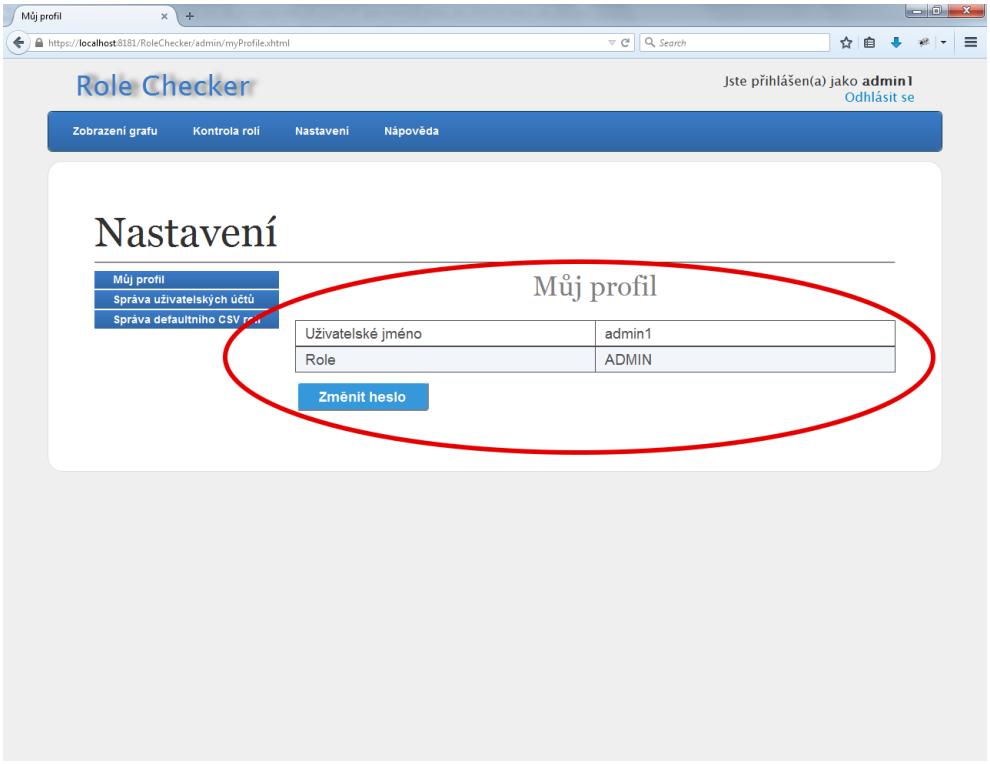


Obrázek B.13: Levé postranní menu v nastavení

Dále pokračujte jednou z podsekcí na základě vámi vybrané akce.

B.4.1 Můj profil

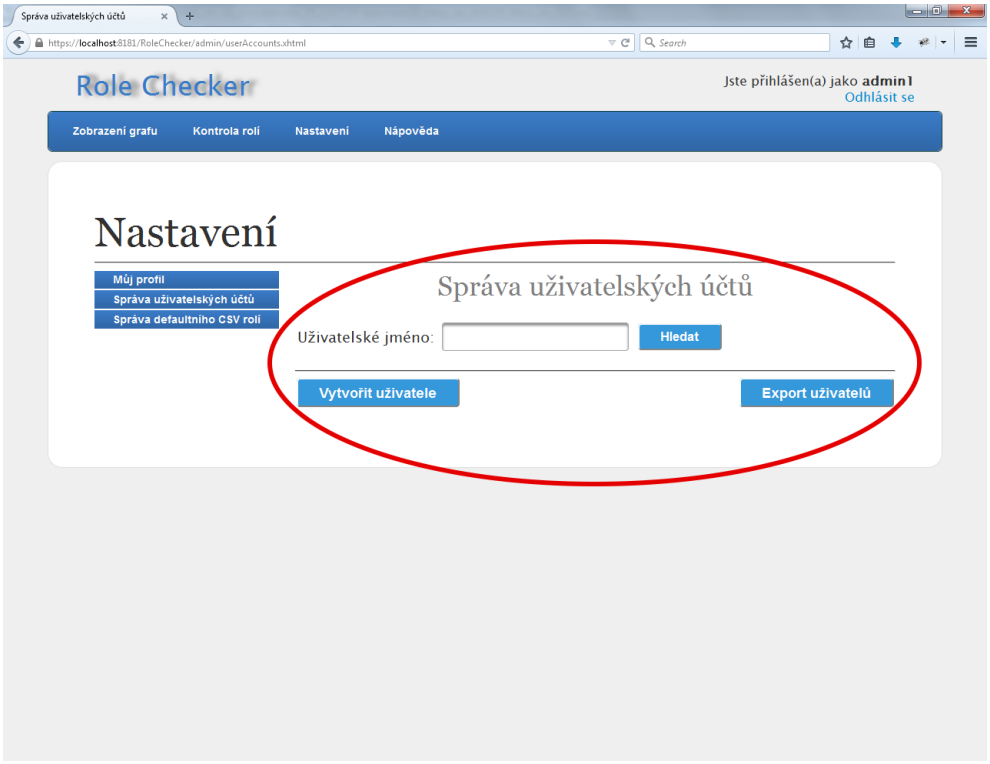
Nyní se zobrazí stránka s podnadpisem „Můj profil“, viz obrázek B.14. Na této stránce jsou základní informace o přihlášeném uživateli a tlačítko s názvem „Změnit heslo“, které umožňuje změnit heslo právě přihlášenému uživateli.



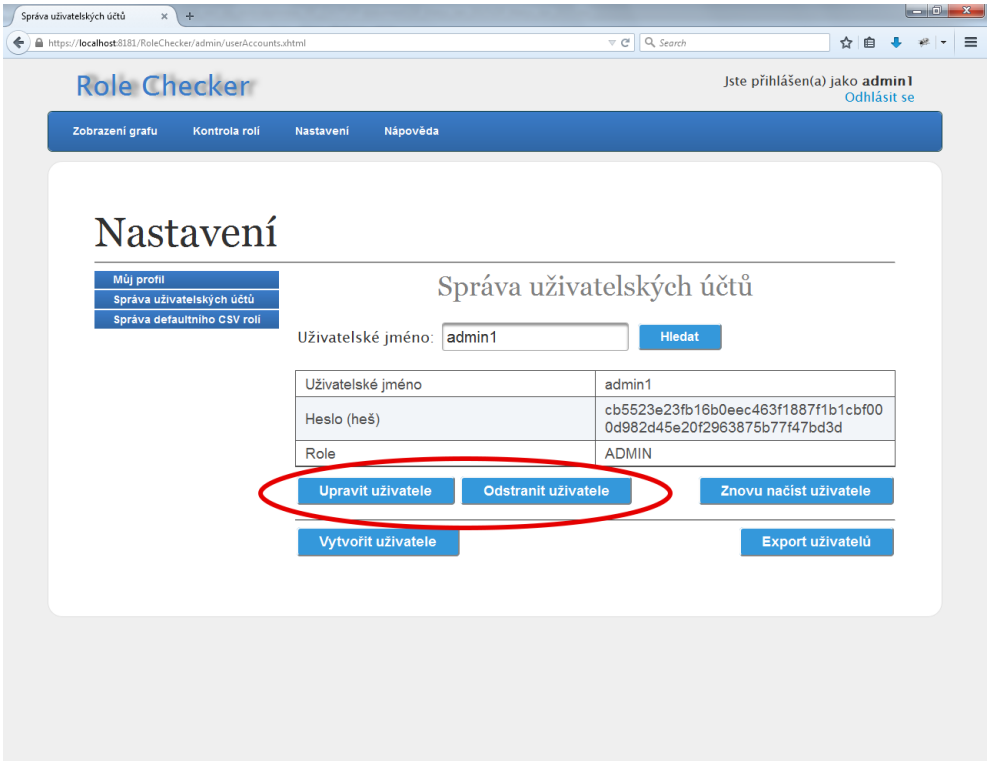
Obrázek B.14: Můj profil

B.4.2 Správa uživatelských účtů

Nyní se zobrazí stránka s podnadpisem „Správa uživatelských účtů“, viz obrázek B.15. Na této stránce najdete tlačítka pro vytvoření uživatele a export uživatelů do CSV souboru. Dále také tlačítko s názvem „Hledat“, které umožňuje vyhledat uživatele podle jeho uživatelského jména. Pokud úspěšně vyhledáte nějakého uživatele, pak se vám zobrazí i tlačítka pro úpravu a odstranění tohoto uživatele, viz obrázek B.16.



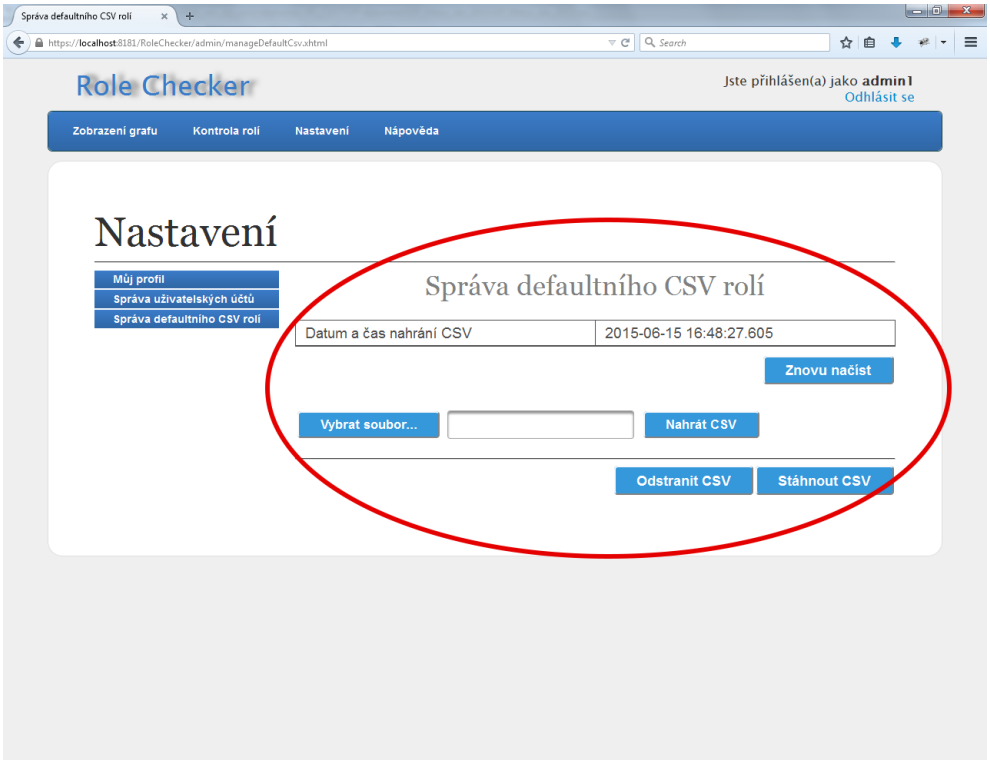
Obrázek B.15: Správa uživatelských účtů



Obrázek B.16: Tlačítka pro úpravu a odstranění uživatele

B.4.3 Správa defaultního CSV rolí

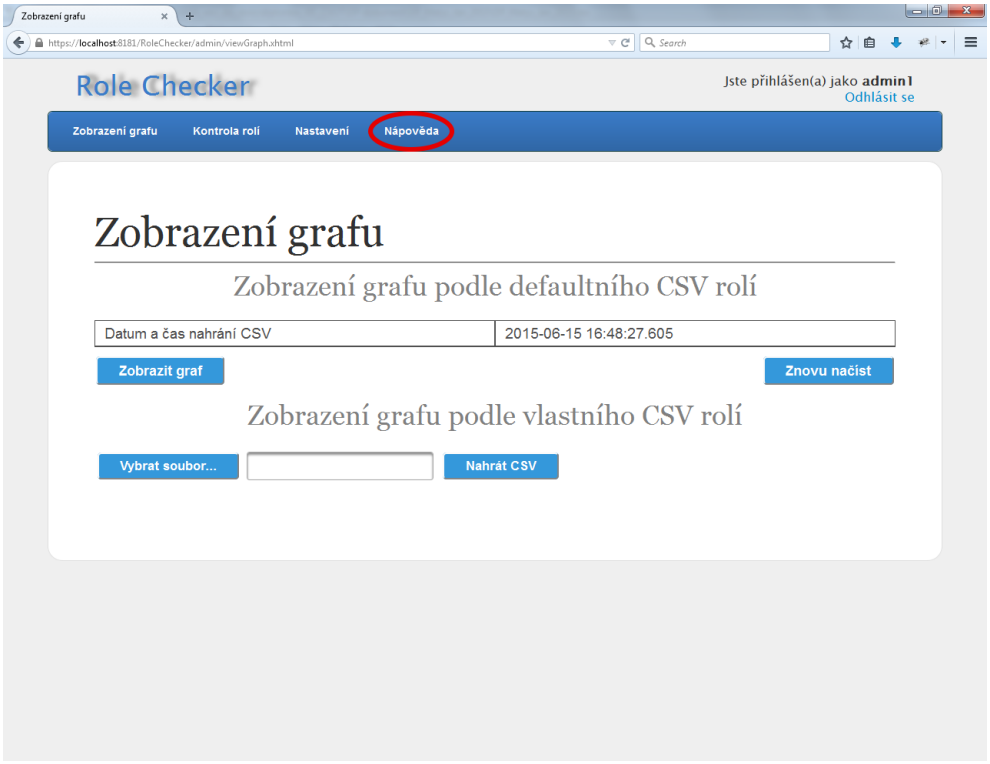
Nyní se zobrazí stránka s podnadpisem „Správa defaultního CSV rolí“, viz obrázek B.17. Na této stránce najdete tlačítka, která umožňují spravovat defaultní CSV soubor rolí.



Obrázek B.17: Správa defaultního CSV rolí

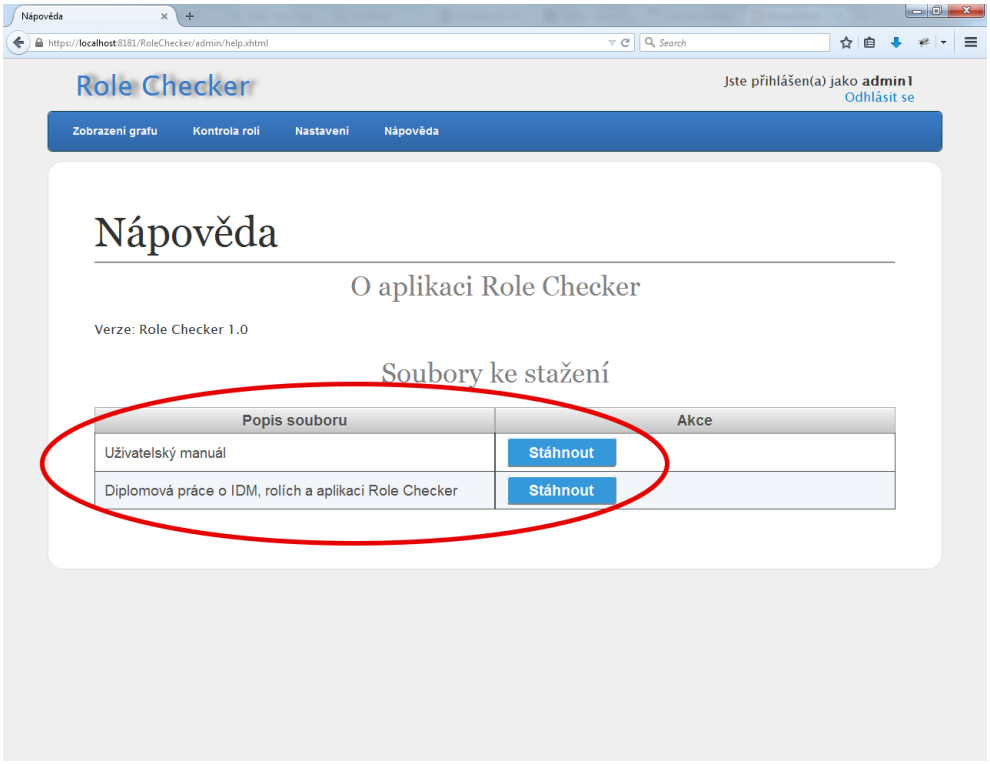
B.5 Nápověda

Po přihlášení klikněte v horním menu na položku s názvem „Nápověda“, viz obrázek B.18.



Obrázek B.18: Položka „Nápověda“ v horním menu

Nyní se zobrazí stránka s nadpisem „Nápověda“. Na této stránce se dozvíte informace o aplikaci a také zde můžete stáhnout dokumenty vztahující se k aplikaci a IDM rolím, viz obrázek B.19.



Obrázek B.19: Dokumenty vztahující se k aplikaci

Příloha C

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `application` ..... adresář se spustitelnou formou implementace
- `csv` ..... adresář s ukázkovým CSV souborem rolí
- `datacleaner` ..... adresář obsahující výsledky profilování dat
- `glassfish_server` ..... adresář obsahující aplikační server GlassFish
- `selenium_ide_tests` ..... adresář obsahující testy Selenium IDE
- `src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text`
  - `DP_Matys_Jan_2015.pdf` ..... text práce ve formátu PDF