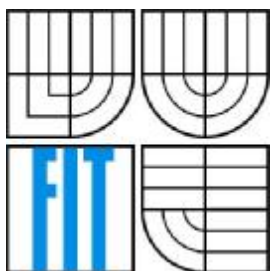




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
CENTRUM VÝPOČETNÍCH A INFORMAČNÍCH
SLUŽEB
FACULTY OF INFORMATION TECHNOLOGY
COMPUTER AND INFORMATION SERVICES CENTRE

HODNOCENÍ RIZIKOVÝCH PROCESŮ RATING OF RISK PROCESSES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. ROBERT JAKAB

VEDOUCÍ PRÁCE
SUPERVISOR

ING., PH.D. JAROMÍR MARUŠINEC

BRNO 2007

Abstrakt

První část práce se zabývá technologií Oracle. Jsou zde zmíněny jednotlivé jazyky, kterými se s daty pracuje a jednotlivé datové typy. Je zde také vysvětlen způsob, jakým jsou data v databázi uložena. Dále jsem popsal technologii Apollo, na které je postaven informační systém Apollo. Další částí práce je analýza, ve které je popsán současný stav hodnocení rizikových procesů a požadavky, které jsou na tento modul kladeny. Následující kapitolou je implementace. V ní jsou jednotlivě popsány jak databázové schéma, tak jednotlivé části modulu ve formě záložek. Je zde důkladně vysvětlen způsob hodnocení rizika a rizikových procesů. Tato kapitola také obsahuje popis jednotlivých tiskových sestav pro tento modul. Další kapitola je věnována uvedení modulu do provozu. Kromě samotného nasazení tohoto modulu v informačním systému Apollo je zde i kapitola o školení zaměstnanců. V závěru jsem uvedl zhodnocení dosažených výsledků, naznačil další vývoj a uvedl i můj osobní přínos.

Klíčová slova

Oracle, Apollo, Asta, Query, databáze, oblast, součást VUT, rizikový proces, riziko, hodnocení, tisková sestava, faktor, váha, pravděpodobnost, dopad, opatření

Abstract

The first part of my work is focused on technology of Oracle. Here are mentioned individual languages, which works with data and different data types. Here is made clear manner, how are data saved in the database. Next I described technology of Apollo, on which information system Apollo is build. Next part of my work is analyze, in which is described present state of evaluation risk process and requests, which are layed on this module. Following chapter is implementation. There are described both database schema and by detail parts of module in a form of strips. Here is at large interpreted the way how to value risk and risk process. This chapter contains print reports for this module as well. The next chapter is dedicated to how module was brought to functioning. Except making this module running in information system Apollo is here a chapter about education of employees. In the closure I present evaluation of reached results, show next development and present my own benefit.

Keywords

Oracle, Apollo, Asta, Query, database, area, part of VUT, risk process, risk, evaluation, print report, factor, weight, probability, impact, prevention

Citace

Robert Jakab: Hodnocení rizikových procesů, diplomová práce, Brno, 2007, FIT VUT v Brně,

Hodnocení rizikových procesů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing., Ph.D. Jaromíra Marušince.

Další informace mi poskytli Ing. Rudolf Musil a Ing. Marta Válková

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Bc. Robert Jakab
19.5.2007

Poděkování

Rád bych poděkoval mému vedoucímu Ing., Ph.D. Jaromíru Marušinci a Ing. Rudolfu Musilovi za odbornou pomoc při řešení diplomové práce a Ing. Martě Válkové za poskytnutí všech informací důležitých při řešení této diplomové práce.

© Robert Jakab, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod.....	3
2	Oracle.....	4
2.1	Historie Oracle.....	4
2.2	Oracle 9i.....	5
2.3	Jazyk SQL.....	6
2.4	Data Definition Language (DDL).....	6
2.5	Data Manipulation Language(DML).....	7
2.6	Data Control Language(DCL).....	7
2.7	Transaction Control Commands (TCC).....	7
2.8	Znakové datové typy.....	7
2.9	Číselné datové typy.....	8
2.10	Datové typy pro kalendářní a časové hodnoty.....	9
2.11	Datové typy pro objemná data.....	10
2.12	Databáze Oracle.....	10
2.13	Databázové struktury.....	12
3	Technologie IS Apollo.....	19
3.1	Akira.....	20
3.2	Apollo.....	20
4	Analýza.....	22
4.1	Současný stav hodnocení rizik.....	22
4.2	Požadavky na systém.....	23
4.3	Diagramy.....	26
5	Implementace.....	28
5.1	Borland Delphi.....	28
5.2	Návrh databázového schéma.....	30
5.3	Vytvoření modulu.....	33
5.4	Tiskové sestavy.....	44
6	Uvedení modulu do provozu.....	47
6.1	Nasazení modulu.....	47
6.2	Školení uživatelů.....	47
7	Závěr.....	49
7.1	Zhodnocení dosažených výsledků.....	49
7.2	Další vývoj.....	49
7.3	Vlastní přínos.....	49

8	Literatura	50
9	Seznam příloh	51
9.1	Uživatelská příručka	51
9.2	Hodnocení rizik v sešitu aplikace Excell	60
9.3	Schéma databáze pro rizikové procesy	61
9.5	Rizikové faktory a jejich hodnoty.....	62
9.6	Tiskové sestavy	63

1 Úvod

Mým úkolem bylo analyzovat současný stav modulu pro hodnocení rizik a rizikových procesů na VUT. Dále bylo nutné zjistit všechny požadavky kladené na nový modul, který budu vytvářet. Všechny požadavky jsem získával a konzultoval s Ing. Martou Válkovou, která se hodnocením rizikových procesů zabývala doposud. Cílem mé diplomové práce bylo implementovat modul rizikových procesů jako součást celoškolského informačního systému Apollo. Mnou vytvářený modul bude jen jedním z mnoha, které již tento informační systém obsahuje. V první řadě mám za úkol seznámit se s fungováním tohoto systému a dále pak s postupem, jakým jsou do tohoto systému nové moduly začleňovány. Dalším úkolem bylo naučit se používání jednotlivých nástrojů pro vývoj tohoto modulu, mezi které patří jak samotné vývojové prostředí tak i jiné podpůrné nástroje, jak pro vytváření a editaci SQL dotazů, tak pro tvorbu tiskových sestav.

2 Oracle

2.1 Historie Oracle

- R. 1977** - Larry Ellison, Bob Miner a Ed Oates založili firmu SDL (Software Development Laboratory), která se zabývala vývojem RDBS (Relational Database Management System) podle teorie doktora E.F.Codda pod názvem Oracle.
- R. 1978** – společnost se přejmenovala na Relational Software Inc. Byla vyvinuta verze Oracle V1 pro počítač PDP-11 a pracující pod operačním systémem RSX. Tato verze byla napsána ve strojovém jazyce a nebyla nikdy oficiálně uvedena na trh.
- R. 1980** - společnost RSI představila komerční verzi Oracle - **Oracle V2**. Tato verze byla také napsána ve strojovém jazyce počítače PDP-11, ale pracuje pod operačním systémem Vax/VMS.
- R. 1982** - představena verze **Oracle V3**. Oracle se stal prvním DBMS pro sálové počítače, minipočítače i počítače PC. V této verzi je zabudována podpora transakčního zpracování. Program. kód pro stranu serveru byl napsán v jazyce C.
- R. 1983** - společnost RSI se přejmenovala na **Oracle Corporation**
- R. 1984** - představena verze **Oracle V4**. Oracle je dostupný pro více platform, umožněna komunikace mezi PC a serverem
- R. 1985** - **Oracle V5** - postavena na architektuře klient - server, podpora VAX clusterů a distribuovaného zpracování
- R. 1988** - **Oracle V6**
- R. 1989** - **Oracle 6.2** - používá Oracle Parallel Server
- R. 1991** - na paralelně pracujících strojích dosaženo výkonnosti 1000 TPS (Transactions Per Second). Database benchmark results.
- R. 1992 - Oracle7 - pro platformu Unix
- R. 1994** - Oracle7 - pro platformu PC
- R. 1997** - **Oracle8** - podpora více uživatelů, větší množství dat, vyšší dostupnost
- R. 1999** - **Oracle8i (Oracle 8.1.5)** - integrace Javy
- R. 2000** - **Oracle8i Release 2** - Oracle se řadí mezi ERP (Enterprise Resource Planning) aplikace..

2.2 Oracle 9i

Při výběru databázové platformy kladou zákazníci důraz na to, aby tato platforma splňovala jejich požadavky. Každá platforma by měla splňovat tyto základní:

1. spolehlivost
2. bezpečnost
3. cena
4. výkon
5. rychlost
6. jednoduchá administrativa
7. nízké nároky na kvalifikaci zaměstnanců (jednoduchost a přehlednost)
8. náklady na provoz
9. možnost provádět většinu administrátorských činností za provozu
10. 100% dostupnost

Tato platforma již obsahuje architekturu umožňující administrátorům reorganizaci databáze a změny databázových struktur za provozu, to např. umožňuje přesunout tabulku na jiné místo, konvertovat na jinou formu nebo rozdělit do více oddílů. Administrátoři mají také možnost uvést databázi do tzv. „klidového“ režimu, který je určený pro vykonání operací nevyžadující aktivní transakční zpracování. Mezi další přírůstky nové verze patří architektura Real Applications Clusters a Cache Vision, ty umožňují zákazníkům horizontální škálování jejich databázové vrstvy v závislosti na zvýšeném počtu přístupů k datům. Technologie Cache Fusion využívá společných vyrovnávacích pamětí všech počítačů, které vzájemně spolupracují nad jednou databází, aby uspokojily všechny požadavky. Tím se snižuje počet vstupně-výstupních operací nad diskem, které jsou časově náročné. Potřebná data jsou hledána v kolektivních vyrovnávacích pamětech všech počítačů a pouze pokud nejsou data nalezena zde, tak teprve potom jsou prováděny vstupně výstupní operace.

Tato platforma umožňuje také ukládat, spravovat a seskupovat všechny typy multimediálního obsahu webové aplikace do jiné databáze. V této verzi je také Internet File System, který dokáže uchovávat obsah jakéhokoli typu. Uživatelé mohou přistupovat k těmto datům prostřednictvím služeb operačního systému, přes webové rozhraní nebo prostřednictvím poštovních či souborových přenosových protokolů. Tato platforma také podporuje zpracování obrázků (také typy PNG a Exif), zvuků a videa prostřednictvím InterMedia. Je zde také možnost přidávat další multimediální formáty a schránka na bázi prohlížeče, která podporuje vkládání, obnovování a anotaci multimediálních objektů. Aplikační server obsahuje služby pro řízení profilů návštěvníků, vytváření, modifikaci, odstraňování uživatelských účtů a správu přihlášených zákazníků.

2.3 Jazyk SQL

SQL (Structured Query Language) byl poprvé představen v projektu společnosti IBM. Dále se na vývoji tohoto jazyka pro práci s daty v databázích začaly zabývat i ostatní firmy. Komerčně tento jazyk uvedla do praxe až společnost Oracle. Poslední používaná verze tohoto jazyka je SQL 92, kterému se zkráceně říká SQL 2 a nyní se pracuje na verzi SQL 3. Tento jazyk je využíván k tomu, že uživatel zformuluje dotaz a databázový server mu na to vygeneruje data jako výsledek. Důležité je při formování dotazů dodržovat syntaktická a lexikální pravidla tohoto jazyka. Na serveru se příkaz dekóduje, optimalizuje a provede.

2.4 Data Definition Language (DDL)

Pomocí tohoto jazyka můžeme definovat, vytvářet, měnit a rušit databázové struktury v databázi. Mezi tyto struktury patří tabulky (v nich jsou uložena data), indexy (pro rychlejší vyhledávání mezi daty), trigger (funkce které se např. vykonají při současném zápisu více dat do databáze) a uložené procedury. Do tohoto jazyka patří následující:

- CREATE DATABASE
- CREATE TABLE
- ALTER TABLE (změnění tabulky)
- DROP TABLE (zrušení tabulky)
- CREATE INDEX
- DROP INDEX
- CREATE VIEW (vytvoří pohled (SQL dotaz který se provede a tváří se jako tabulka ale výsledek tohoto dotazu není uložen v databázi))
- ALTER VIEW
- DROP VIEW
- CREATE SEQUENCE (při vkládání nového záznamu do tabulky se nemusí vkládat index daného záznamu(jeho id) a díky této sekvenci se vloží další číslo v pořadí a toto pořadí si databáze pamatuje)
- ALTER SEQUENCE
- DROP SEQUENCE
- CREATE PROCEDURE
- DROP PROCEDURE
- CREATE TRIGGER
- DROP TRIGGER

2.5 Data Manipulation Language(DML)

Jde o jazyk používaný pro manipulaci s daty. Patří sem tyto příkazy:

- SELECT (dotazovací příkaz)
- INSERT (pro vkládání dat)
- UPDATE (pro modifikaci dat)
- DELETE (pro mazání dat)

2.6 Data Control Language(DCL)

Jde o jazyk určený pro správu dat a patří do něj příkazy pro řízení provozu a údržby databáze. Patří sem např.:

- GRANT
- REVOKE
- REVOKE CREATE USER
- ALTER USER
- DROP USER

2.7 Transaction Control Commands (TCC)

Tento jazyk obsahuje příkazy pro řízení transakcí. Patří sem:

- SET TRANSACTION
- COMMIT
- ROLLBACK
- SAVEPOINT

2.8 Znakové datové typy

Jsou určena pro ukládání především textových dat. Mohou obsahovat data, která jsou tvořena z alfanumerických znaků, číslic a jiných znaků.

CHAR(délka) [BYTE | CHAR]

- slouží pro uložení řetězce o pevné délce která je určena parametrem délka. Tento parametr je povinný a může nabývat hodnot <1,2000> a standardně je nastaven na 1. Pomocí volitelného přepínače BYTE | CHAR lze upřesnit, zda číslo v parametru délka udává počet bajtů nebo počet znaků.

NCHAR(délka)

- používá se pro vládání řetězců pevné délky ve vybrané národní znakové sadě, která se definuje při vytváření databáze. Povinný parametr délka zde udává velikost v bajtech a

stejně jako datový typ CHAR nabývá hodnot z intervalu <,2000>. Rozdíl od datového typu CHAR je zde jen v uložení specifických znaků národní abecedy.

VARCHAR2(délka) [BYTE | CHAR]

- slouží pro uložení řetězce proměnné délky. Parametr délka je zde povinný a určuje maximální možnou délku řetězce. Může nabýt hodnot z intervalu <1,4000> a volitelný přepínač BYTE | CHAR udává zda parametr délka znamená počet znaků nebo počet bajtů.

NVARCHAR2(délka)

- slouží pro uložení řetězce v národní znakové sadě. Povinný parametr délka zde udává maximální délku řetězce v bajtech a nabývá hodnot z intervalu <1,4000>.

LONG

- i když se tento datový typ většinou používá pro uložení číselné proměnné tak v platformě Oracle 9i slouží pro uložení řetězce proměnné délky (maximálně 2GB). Doporučuje se místo tohoto datového typu používat datový typ LOG, protože při práci s tímto typem pomocí jazyka SQL zde existuje celá řada omezení.

2.9 Číselné datové typy

Slouží pro uložení číselných dat a to jak celých čísel, tak desetinných a to buďto s pevnou řadovou čárkou nebo s plovoucí řadovou čárkou.

NUMBER(n_číslic, n_číslic_za_des_čárkou)

- je určen pro uložení kladných i záporných čísel, celých i desetinných čísel a to v rozsahu 1.0×10^{-130} až $9.9..9 \times 10^{125}$. platnost tohoto datového typu je na 38 platných číslic. Parametr n_číslic udává celkový počet platných číslic a parametr n_číslic_za_des_čárkou udává počet platných číslic za desetinnou čárkou. U celého čísla se udává je počet platných číslic v parametru n_číslic a nebo se udávají oba parametry s tím, že parametr n_číslic_za_des_čárkou je roven nule. Desetinné číslo s pohyblivou desetinnou čárkou se deklaruje jako NUMBER. Parametr n_číslic_za_des_čárkou může být také záporný a to znamená že se bude zaokrouhlovat číslo před desetinnou čárkou. Pokud bude parametr n_číslic menší než je počet platných číslic před desetinnou čárkou daného čísla nepodaří se toto číslo uložit.

2.10 Datové typy pro kalendářní a časové hodnoty

Patří sem dva druhy datový typů:

1. pro vyjádření hodnoty data a času
2. pro vyjádření datového a časového intervalu

Rozsahy datových a časových typů.

Název pole	Platné pro datum a čas	Platné pro interval
YEAR		kladné nebo záporné číslo
MONTH	01 až 12	0 až 11
DAY	01 až 31 (28,29,30)	kladné nebo záporné číslo
HOURL	00 až 23	0 až 23
MINUTE	0 až 59	0 až 59
SECOND	00 až 59.9(9..9)	00 až 59.9(9..9)
TIMEZONE_HOUR	(-12) až 13	
TIMEZONE_MINUTE	0 až 59	

2.10.1 Datové typy pro vyjádření konkrétního data a času

DATE

- tento datový typ slouží pro uložení hodnoty data a času z intervalu od 1.ledna 4712 před naším letopočtem do 31.prosince 9999 našeho letopočtu. Pro převody data z různých formátů je zde funkce TO_DATE, která umožňuje zadat hodnotu data a času v libovolném formátu. Příklad použití: TO_DATE('98-DEC-25:17:30','YY-MON-DD:HH24:MI'). Pokud potřebujeme uložit aktuální čas a datum nabízí Oracle funkci SYSDATE, která se napíše mezi vkládané hodnoty na místo kde chceme uložit aktuální datum a čas. Pokud chceme pracovat pouze s datovou hodnotou a ne s časovou nabízí platforma Oracle funkci TRUNC pro zaokrouhlování. Pokud tedy chceme uložit jen aktuální datum , zadáme mezi vkládané položky TRUNC(SYSDATE).

TIMESTAMP [přesnost_sekund]

- slouží stejně jako datový typ DATE pro uložení data a času. Je ale přesnější. Parametr přesnost_sekund zde udává na kolik desetinných míst se má počet sekund ukládat. Standardně je nastaven na 6, takže je přesnost na miliontiny sekundy. Pro usnadnění zde lze využít převodní funkce TO_TIMESTAMP, která dané datum a čas převede na tento datový typ.

2.10.2 Datové typy pro vyjádření datového a časového intervalu

INTERVAL YEAR [(rozsah_roků)] TO MONTH

- slouží k vyjádření delšího časového intervalu. Parametr rozsah_roků slouží k nastavení počtu číslic čísla udávajícího počet roků, standardně má hodnotu 2.

INTERVAL DAY [(rozsah_dní)] TO SECOND [přesnost_sekund]

- slouží k uložení kratšího a přesnějšího časového intervalu. Parametr rozsah_dní zde udává počet číslic udávajícího počet dní. Standardně má hodnotu 2. Parametr přesnost-sekund určuje počet platných číslic sekund, standardně má hodnotu 6.

2.11 Datové typy pro objemná data

Pro ukládání velkého objemu se používají datové typy LOB(Large Object Block). Mezi ně patří následující:

BLOB

- používá se pro uložení rozsáhlých binárních objektů přímo do databáze. Ukládají se sem např. rozsáhlá multimediální data (zvuk, video). Maximální velikost je u tohoto datového typu 4GB.

CLOB

- má obdobné vlastnosti jako datový typ BLOB.

NCLOB

- slouží také k uložení rozsáhlých binárních objektů a obsahuje znaky UNICODE. Maximální velikost je u tohoto datového typu 4GB.

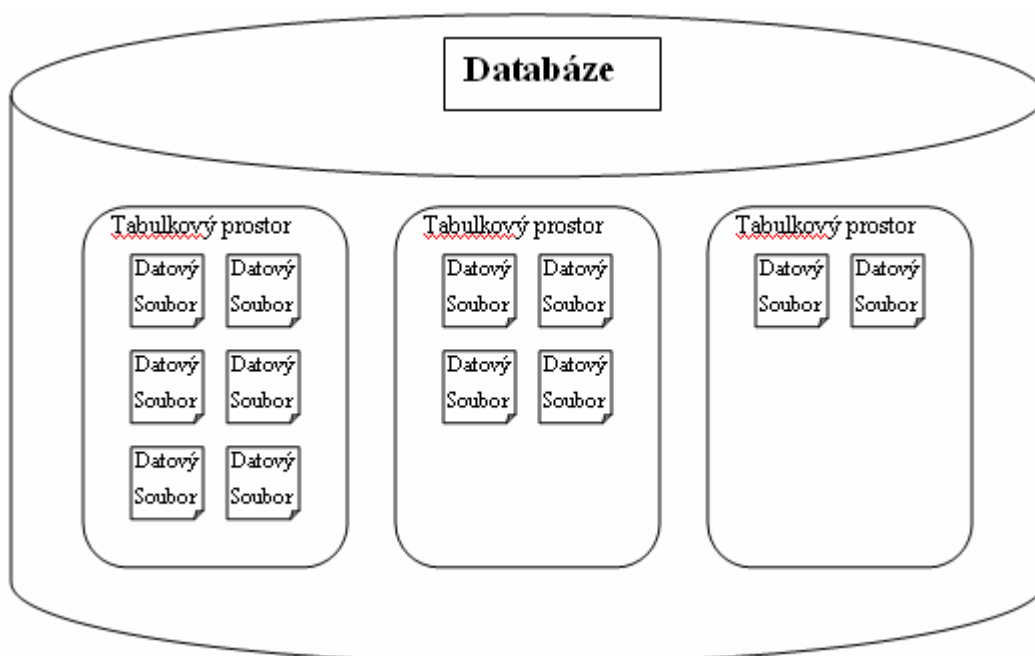
BFILE

- slouží také k uložení rozsáhlých binárních objektů. Data se ukládají do externích souborů mimo databázi.). Maximální velikost je u tohoto datového typu 4GB.

2.12 Databáze Oracle

Oracle je relační systém řízená báze dat. Data jsou zde ukládány do tabulek, které jsou zde definovány sloupci, které mají různé názvy a u každého sloupce je definován typ dat jaké v něm budou obsažena. Data jsou uložena v jednotlivých řádcích dané tabulky. Slovo relační u těchto databázích znamená, že vazby mezi tabulkami se zde vytvářejí tak, že v tabulkách které chceme propojit je v každé jeden sloupec, který nabývá stejných informačních hodnot jako jeden ze sloupců

druhé tabulky. Oracle ale nepodporuje přístup k datům jen podle relačního modelu, ale podporuje i objektově orientované struktury (např. abstraktní datové typy a metody). Objekty lze mezi sebou provázat nebo mohou obsahovat další objekty. Veškerá data jsou interními strukturami databáze logicky mapována na soubory. Tato logická rozdělení jsou označována jako tabulkové prostory. Každý tabulkový prostor se skládá z jednoho nebo více souborů na disku (tzv. datových souborů). Datový soubor může patřit pouze k jednomu tabulkovému prostoru. Po přidání datového souboru k tabulkovému prostoru ho není možné z tabulkového prostoru odstranit ani přiřadit k jinému tabulkovému prostoru. Díky tomu, že je možné uložit datové soubory tabulkových prostorů na samostatné disky, je možné optimalizovat zpracování vstupních a výstupních požadavků na databázi. Na Obr.1 je vidět vztah mezi databází, tabulkovými prostory a datovými soubory.



Obr. 1 Uložení dat v databázi

Samotná databáze se samozřejmě neskládá jen z datových souborů, které jsou požitý pro uložení dat. Mezi ostatní soubory, které také náleží k databázi jsou tyto:

- ✓ Transakční protokoly
- ✓ Řídící soubory
- ✓ Trasovací soubory
- ✓ Protokol upozornění

2.12.1 Transakční protokol

Oracle uchovává protokoly všech transakcí prováděných nad databází. Transakcí se zde rozumí jakékoli vytváření, modifikace či mazání dat. Tyto transakce jsou zaznamenávány do online souborů transakčního protokolu. Tyto soubory se při selhání databáze používají pro obnovení databáze do původního stavu před výpadkem pomocí právě těchto informací o provedených transakcích. Tyto

soubory jsou uloženy mimo datové soubory databáze. Všechny databáze Oracle obsahují tři nebo více souborů transakčního protokolu. Zapisuje se do nich cyklickým způsobem. Jakmile se zaplní první soubor, začne se zapisovat do druhého souboru atd. jakmile je zaplněn poslední soubor začne se opět přepisovat první soubor. Pokud je databáze spuštěná v režimu ArchiveLog, vytvářejí se kopie aktivních souborů transakčního protokolu ještě předtím, než se přepíše.

2.12.2 Řídicí soubory

V řídicích souborech se uchovává celková fyzická architektura databáze. Obsahují informace o řízení všech souborů v databázi. Udržují také interní konzistenci databáze a řídí operace vedoucí k jejímu zotavení v případě poruchy či výpadku. Z důvodu důležitosti těchto souborů se ukládá více jejich kopií a to často na samostatné disky z bezpečnostních důvodů kdyby některý disk selhal.

2.12.3 Trasovací soubory

Ke každému procesu na pozadí spuštěnému v instanci je navázán jeden trasovací soubor. Ten obsahuje informace o důležitých událostech probíhajících na pozadí. Jsou využívány hlavně při zjišťování příčin kritického selhání databáze.

2.12.4 Protokol upozornění

Do tohoto souboru se ukládají příkazy a výsledky hlavních událostí týkajících se fungování databáze. Používají se pro správu databáze. Obsahuje položky, které nás budou informovat o všech problémech, které souvisejí s během celé databáze, včetně všech interních chyb.

2.13 Databázové struktury

2.13.1 Tabulky

Tabulky se v databázi používají pro uložení dat. Tabulky obsahují sloupce, které mají jasně definovaný název a datový typ, který určuje jaké hodnoty se ve sloupci budou vyskytovat. Sloupce popisují jednotlivé vlastnosti dané entity nebo slouží jako vazba na jiné tabulky. Tabulky vlastněně uživatelem SYS se nazývají tabulky systémového katalogu. Poskytují systémový katalog, který databáze používá ke své správě.

2.13.2 Dočasné tabulky

Má stejné vlastnosti jako běžná tabulka až na to, že data vložená do dočasné tabulky jsou k dispozici pouze po dobu trvání relace nebo transakce. Definice dočasné tabulky (jednotlivé sloupce a jejich datové typy) zůstává zachována. Pokud potřebujeme aby mohly s tabulkou pracovat (vkládat do ní

data a pracovat s nimi) všechny relace připojující se k databázi vytvoříme globální dočasnou tabulku. Do tabulky se ukládají data z různých relací, ale daná relace může pracovat jen s daty, kterými tabulku naplnila.

2.13.3 Integritní omezení

Každý sloupec bulky může mít při vytvoření tabulky nějaké integritní omezení, které definuje jaké mohou být dané hodnoty sloupce. Takže jedním omezením se dal nazvat datový typ sloupce jako další omezení je právě toto integritní omezení. Mezi integritní omezení, které se dají aplikovat na sloupce tabulky patří následující.

1. PRIMARY KEY
Jeden či více sloupců mohou mít integritní omezení PRIMARY KEY uvedené za definicí tohoto sloupce. Znamená to že tento (více) sloupec musí být unikátní a nesmí být prázdný.
2. NOT NULL
Toto integritní omezení se používá u sloupců, které nejsou primárními klíči a potřebujeme zaručit, že hodnoty v nich nebudou prázdné
3. DEFAULT (hodnota)
Znamená, že pokud se při vkládání dat u tohoto sloupce neuvede hodnota, bude zde uložena defaultní hodnota z parametru hodnota.
4. CHECK (podmínka)
toto integritní omezení nepovolí uložit do tohoto sloupce hodnotu, která nespĺňuje definovanou podmínku. Podmínky lze vytvářet jen nad sloupci této tabulky
5. UNIQUE
Toto omezení se používá u sloupců, které nejsou primárními klíči a potřebujeme zaručit že hodnota v nich bude v rámci všech hodnot v tomto sloupci unikátní
6. FOREIGN KEY (název sloupce tabulky) REFERENCES název_tabulky(název sloupce)
Definuje se za všemi sloupci tabulky a uvádí se zde nejdříve název sloupce této tabulky který bude plnit funkci cizího klíče do druhé tabulky a bude navazovat na daný sloupec druhé tabulky. Díky tomu bude zaručeno, že v tomto sloupci budou jen hodnoty, které obsahuje sloupec druhé tabulky a tím nelze narušit konzistenci databáze (vztahu mezi těmito tabulkami), protože databáze nedovolí nikomu smazat daný záznam z druhé tabulky, pokud se na něj odkazuje některý záznam z této tabulky pomocí tohoto cizího klíče.

2.13.4 Abstraktní datové typy

Pokud je v databázi nainstalována možnost Objekt, může uživatel definovat své vlastní datové typy. Tento abstraktní datový typ se vytváří pomocí příkazu create type (jméno abstraktního typu) as object (jeho jednotlivé složky). Jeho jednotlivé složky se definují stejně jako sloupce tabulky avšak bez integritních omezení. Každá položka má své jméno a datový typ. Takto nadefinovaný abstraktní datový typ lze poté používat při vytváření tabulek a může být přiřazen jakémukoli sloupci tabulky. Po zavedení abstraktního datového typu se automaticky vytvoří konstruktor pro sloupec používající daný

datový typ se stejným jménem jako je název datového typu. Atributy tohoto datového typu jsou poté také parametry příslušné Metody.

2.13.5 Oddíly a pododdíly

Příčinou vzniku oddílů a pododdílů je složitá správa velkých tabulek. Čím větší je tabulka tím déle také trvají dotazy na ní kladené. Z toho důvodu je lepší tuto tabulku rozdělit do samostatných menších tabulek podle času, oddělení nebo hodnot. Lze tedy velkou tabulku rozdělit do menších podle nějakého rozsahu hodnot. Tyto menší tabulky se nazývají oddíly a jejich správě je značně jednodušší. Lze tedy pracovat s tímto oddílem a nad ním provádět dané operace aniž by se tyto operace dotkly jiného oddílu. Přesto Oracle nahlíží na všechny oddíly tabulky jako na jednu tabulku, ale umožňuje správu nad jednotlivými oddíly tabulky. Tím se tedy zvyšuje i výkon aplikace a optimalizace dotazů, protože při zpracování dotazů se zjistí o jaký oddíl se jedná a pracuje se jen z daty z tohoto oddílů, kterých je méně a o to je toto zpracování rychlejší. Mohou se takto rozdělovat i indexy. U takovéto indexové tabulky se jedná o tzv. místní index. Tabulky lze tedy rozdělit na oddíly podle rozsahu určitých hodnot a tyto oddíly lze dále dělit na pododdíly podle rozsahu jiných hodnot.

2.13.6 Uživatelé

Uživatelé jsou ve skutečnosti vlastníci databázových objektů. Uživatel SYS vlastní tabulky systémového katalogu, které uchovávají informace o ostatních strukturách databáze. Živitel SYSTÉM vlastní pohledy s přístupem k tabulkám systémového katalogu, které využívají ostatní uživatelé. Objekty databáze jsou přiděleny jednotlivým uživatelským účtům. Pro uživatele lze také vytvořit výchozí tabulkový prostor který mohou využívat. Lze také propojit účty databáze a účty operačního systému tak aby uživatelé po připojení do operačního systému již nemuseli zadávat heslo pro připojení k databázi.

2.13.7 Schémata

Uživatelské schéma je sada objektů vlastněná uživatelským účtem

2.13.8 Indexy

V systému Oracle jsou data vyhledávány pomocí identifikátoru RowID, který je přiřazen ke každému řádku ve všech tabulkách. Pomocí tohoto identifikátoru databáze přesně zjistí ve kterém souboru, bloku a na kterém řádku se záznam nachází. Tento index je používán serverem k rychlému vyhledání řádku. Existují tři druhy indexů:

1. indexy clusterů
 - uchovávají klíčové hodnoty clusterů
2. indexy tabulek

- zde jsou uloženy hodnoty řádků tabulky a umístění jednotlivých řádků tabulky (RowID)

3. bitmapové indexy

- je určen k podpoře dotazů velkých tabulek se sloupci obsahujícími několik rozdílných hodnot. Používají se pokud jsou data v tabulce velmi podobná (rotili mezi sloupci jsou velmi malé). V těchto případech urychlují vyhledávání. Jsou vhodnější pro statická data.

Položky indexu jsou v systému Oracle ukládány pomocí mechanismu B*tree, který zajišťuje rychlý přístup ke klíčové hodnotě. Dotaz při přístupu k indexu vyhledá položky indexu, které odpovídají kritériím dotazu. Tímto indexy zvyšují výkon databáze a zajišťují jedinečnost sloupce. Oracle automaticky vytvoří index u sloupců tabulky, které obsahují omezení UNIQUE nebo PRIMARY KEY. Vlastní indexy se vytvářejí pomocí příkazu create index. Lze také vytvářet tabulky uspořádané podle indexů. Vytváří se pomocí klauzule organization index v příkazu create table. Její data jsou rozříděna podle primárního klíče. Tato tabulka poté neobsahuje identifikátory RowID a místo nich používá hodnotu primárního klíče.

2.13.9 Clustery

Jsou využívány pokud se dotazujeme často současně do některých tabulek. Poté je vhodné tyto tabulky fyzicky uchovávat společně. V tomto případě je vhodné vytvořit cluster, který bude tabulky udržovat pohromadě. Tím můžeme zvýšit výkon databáze, protože data v tabulkách budou ukládána do clusteru společně a tím se minimalizuje počet V/V operací. Sloupce sdružených tabulek se nazývají klíč clusteru. Tento klíč je indexován pomocí indexu clusteru, jehož hodnota je ukládána pro více tabulek v clusteru pouze jednou. Index clusteru se musí vytvořit před vložením jakýchkoli dat do tabulek daného clusteru. Tím že jsou data tabulek v clusteru ukládány do stejných bloků, poté také dotazy týkající se těchto tabulek mohou provádět méně V/V operací a jsou tudíž rychlejší než kdyby byly tabulky uloženy odděleně. Avšak rychlost vkládání, aktualizace a odstraňování může být menší než u tabulek uložených odděleně což je nutné před sdružováním tabulek do clusteru zvážit.

2.13.10 Clustery hash

Cluster hash pracuje s clusterovým klíčem řádku pomocí funkcí hash, které používá pro určení fyzického umístění řádku. Tyto clustery jsou z hlediska výkonu výhodné pro dotazy založené na ekvivalenci.

2.13.11 Pohledy

Pohled se tváří jako tabulka obsahující sloupce a je dotazován stejným způsobem jako tabulka. Při dotazu na pohled se server dotáže tabulek na kterých je pohled založen a vrátí data. Definice pohledu zahrnuje dotaz na kterém je pohled založena vrátí sloupce v daném pořadí a udělená oprávnění. Dalo by se říct že se jedná o SQL dotaz který je v databázi uložen pod nějakým jménem a lze s ním pracovat jako s tabulkou akorát v něm nelze data modifikovat, protože data z tohoto pohledu jsou jen výsledek daného SQL dotazu nad danými tabulkami. Jsou často používány pro zabezpečení dat na úrovni řádků a sloupců. Lze udělit uživatelský přístup pouze k těm řádkům tabulky, které uživatel vlastní a to samé se sloupci tabulky.

2.13.12 Sekvence

Definice sekvencí jsou uloženy v systémovém katalogu. Tyto sekvence vytvářejí seznam jedinečných čísel. Při prvním volání sekvence je vrácena předdefinovaná hodnota. Následná dotaz vrátí vždy hodnotu zvětšenou o určený přírůstek, který je nadefinován v sekvenci. Používá se například při vkládání jednotlivých řádků do tabulky a u nich k generaci primárního klíče (id záznamu v tabulce). Tyto sekvence se mohou stále opakovat nebo se zvětšovat až do zadané maximální hodnoty.

2.13.13 Procedury

Uložená procedura představuje blok příkazů PL/SQL. Jsou uloženy v systémovém katalogu a jsou volány aplikacemi. Spuštěním procedury se spustí její příkazy jako celek. Nevracejí volajícímu programu žádné hodnoty. Často se používají k zabezpečení dat. Než by se někomu přidělili práva na celou tabulku, přidělí se mu právo na spuštění procedury která má k těmto tabulkám přístup a vykoná dané operace. Spuštěním procedury se aktivují i oprávnění vlastníka procedury.

2.13.14 Funkce

Uložené funkce jsou bloky kódu uložené v databázi, které na rozdíl od procedur mohou volajícímu programu vracet hodnoty. Plno funkcí poskytuje platforma Oracle, ale uživatel si může nadefinovat své vlastní funkce. Když budete vlastníkem nějaké funkce musíte mít uděleno oprávnění EXECUTE k této funkci abyste jí mohl používat. Uživatelsky definované funkce můžete používat pouze pokud nebudou měnit žádné řádky tabulky.

2.13.15 Balíky

Používají se k seskupení procedur a funkcí do logických seskupení (tedy balíků). Specifikace, obsahy a zdrojové kódy balíků, funkcí a procedur jsou uloženy v systémovém katalogu. Různé prvky balíku lze definovat jako „veřejné“ nebo „soukromé“. Veřejné jsou přístupné uživateli balíku a soukromé jsou tomuto uživateli skryta. Soukromé prvky mohou například obsahovat procedury které jsou volány jinými procedurami.

2.13.16 Triggery

Jsou to procedury, které se spustí když dojde k specifické události databáze. Existují dva typy triggerů:

1. Triggery příkazů
Spustí se jednou při každém odpovídajícím příkazu
2. Triggery řádků
Spustí se jednou pro každý řádek ovlivněný daným příkazem

Je možné vytvořit triggery BEFORE, AFTER, INSTEAD OF které se vztahují k událostem insert, update, delete, create a alter. Trigger BEFORE se provede před vykonáním dané události, trigger AFTER se vykoná po provedení dané události a trigger INSTEAD OF se vykoná místo dané události.

2.13.17 Synonyma

Synonyma jsou ukazatele na tabulky, pohledy, procedury, funkce, balíky a sekvence. Rovněž mohou ukazovat na objekty v rámci lokální databáze nebo ve vzdálených databázích. U případu vzdálených databází je pro identifikaci databázového objektu nutné určit název hostitelského počítače, název instance, vlastníka objektu a název objektu. Podle toho kde je objekt umístěn bude nutné pro uživatele určit jeden či více těchto parametrů. Proto se pro zjednodušení přístupu k těmto databázovým objektům vytvářejí synonyma.

2.13.18 Oprávnění a role

Pro přístup k objektu vlastněnému jiným účtem musíte nejdříve udělit oprávnění k přístupu k tomuto objektu. Lze udělovat oprávnění k vložení, výběru, aktualizaci nebo odstranění řádků tabulky nebo zobrazení. Dále lze také uživatelům přidělit oprávnění k výběru hodnot ze sekvencí a k spuštění procedur, funkcí balíku a abstraktních datových typů. Dále lze udělit oprávnění k čtení adresářů (pro datové typy BFILE a externí tabulky) a oprávnění ke spouštění týkající se knihoven. Oprávnění lze udělovat jen některým uživatelům nebo PUBLIC, které poskytuje oprávnění všem uživatelům v databázi. Celý proces přidělování oprávnění lze zjednodušit vytvořením rolí. Role představují

skupiny oprávnění. Dané roli se tedy nejdříve přidělí daná oprávnění a tato role se pak přidělí uživatelům.

2.13.19 Databázová propojení

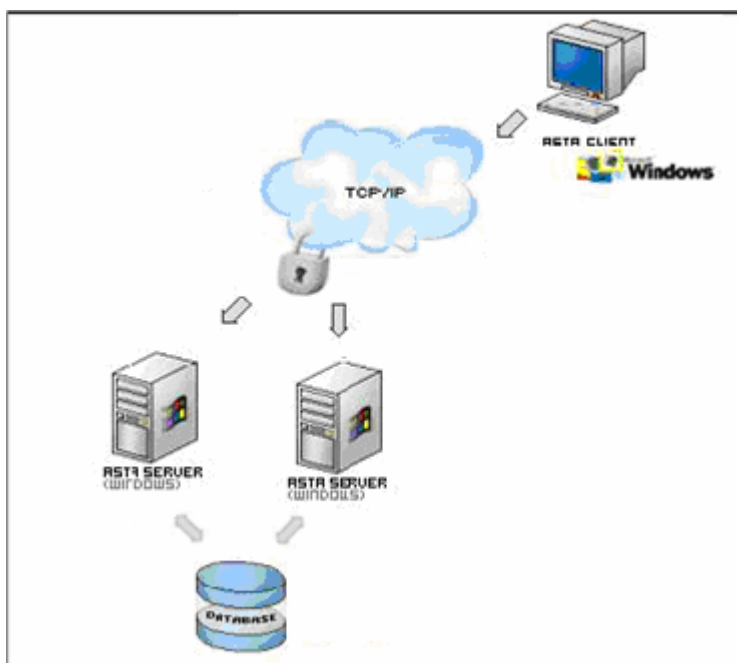
Pokud se potřebujeme dostat k databázovému objektu ve vzdálené databázi musíme vytvořit databázové propojení. Tyto databázové propojení mohou být veřejná (přístupná všem účtům dané databáze) nebo soukromá (vytvořená uživatelem pouze pro potřeby jeho účtu). Při vytváření každého databázového propojení je nutné zadat název účtu, ke kterému se chceme připojit, heslo k tomuto čtu a název služby související se vzdálenou databází.

2.13.20 Materializované pohledy

Používají se pro poskytování kopie vzdálených dat nebo dat v rámci stejné databáze. Tento pohled je založen na dotazu ale výsledek tohoto dotazu je fyzicky uložen v databázi. Tyto pohledy mohou být jen ke čtení a mohou být pravidelně aktualizovány podle určitého plánu. Materializované pohledy mohou být např. použity. Pokud je dotaz na kterém jsou založeny časově náročný a data nemusejí být až tak moc aktuální. Aktuálnost těchto dat záleží na četnosti jeho aktualizace. Jsou významným nástrojem pro optimalizaci dotazů a vývojář si pomocí nich může hodně ulehčit práci, zrychlit dotaz na potřebná data a aktualizace těchto pohledů může probíhat v čase kdy je databáze nejméně zatěžovaná z důvodu složitosti dotazu.

3 Technologie IS Apollo

Základem systému je třívrstvá technologie firmy AstaTech. Schéma jejího fungování je znázorněno na Obr.2. Tato technologie dovoluje vytvořit bezpečné připojení aplikačního klienta k serveru. Pro zabezpečené připojení se používá protokol SSL. Klient se připojuje přímo k aplikačnímu serveru. Výběr nejméně zatíženého serveru provádí klient tak, že klient odešle všem dostupným aplikačním serverům dotaz na počet připojených uživatelů - od serverů dojdou informace, kolik je k nim připojeno uživatelů. Tento uživatel je poté připojen k serveru, ke kterému je připojen nejmenší počet uživatelů. U méně náročných databázových aplikací stačí pouze jeden Asta server. Počet Asta serveru se tedy může měnit tak, aby dokázal v budoucnu pokrýt vyšší nárůst uživatelů. Asta server spravuje práva uživatele pro připojení k databázím Oracle. Při připojení pošle klientovi jeho ovládací menu s položkami na které má právo. Klient pak žádá a posílá data pomocí zástupných kódu SQL. Před vykonáním jakéhokoliv kódu vždy dojde k ověření, zda má uživatel právo na spuštění tohoto kódu. To znamená, že aplikační klient nemůže být úspěšně napaden hrubou silou. Server přistupuje k databázi pomocí komponent DOA – Direct Oracle Access. Jak vyplývá z již výše uvedeného, je jako databázový server použit Oracle 9i běžící v několika instancích na velmi výkonných víceprocesorových strojích. Je zde více instancí serveru. Jedna databáze je určená pro vývoj. Je určena pro vývojáře, kteří si zde tvoří potřebné tabulky naplňují je nezbytnými daty a ladí své moduly. Dále je testovací databáze, která se často aktualizuje. Zde se testuje funkčnost jednotlivých modulů před jejich nasazením na ostrou databázi a ladí se poslední chyby. Jako třetí je ostrá databáze. Zde již jsou aktuální a platná data.



Obr. 2 Schéma fungování serveru Asta

Aplikace psaná s použitím CLX knihoven zabírala velké množství systémových prostředků a byla pomalá. Proto se přestoupilo na použití standardních knihoven Delphi VCL, určených k provozu jen pod systémy Windows. Výhodou je značné zvýšení rychlosti a podstatně nižší nároky na systémové prostředky. Ochuzení ovšem nejsme ani o provoz pod Linuxem, neboť můžeme využít služeb projektu WINE, určeného k provozování aplikací psaných pro Win32 API. A nemusíme se bát ani zvýšených nákladů, protože projekt WINE je open source a freeware. Dále je možné připojit se i pod Linuxem na terminálový server Apollo.vutbr.cz.

3.1 Akira

Akira je server střední vrstvy postavený na technologii Asta. Tento server se po přihlášení uživatele stará o zpracování databázových dotazů a předává odpovídající výsledky zpět klientovi. Na tomto serveru leží hlavní míra zodpovědnosti za bezpečnost celého systému. Komunikace s ním probíhá přes zabezpečené připojení pomocí SSL, díky čemuž je zabráněno odposlechu dat. Dotazy jsou vykonávány jen ty, které jsou uloženy na serveru a tyto jsou vybírány za pomoci zástupných kódů. Díky tomuto řešení je zabráněno neoprávněnému vytažení nebo modifikaci dat. Každý dotaz má navíc svá oprávnění a není možné spustit dotaz na který nemá uživatel právo. Samozřejmostí je zaznamenávání operací do logovacího souboru pro dohledávání problému a také omezení počtu možných špatných přihlášení za den, aby se zabránilo útoku silou na heslo. Mezi další vlastnosti serveru patří možnost fetchování dat (data odpovědi nejsou poslána všechna, ale jsou posílána na požádání v blocích o zadaném počtu řádků) a také jejich komprese (díky čemuž může být dosaženo zrychlení komunikace na pomalejších linkách). Důležitá je také možnost parametrizovat dotazy za pomoci proměnných. Zde se Akira stará například o to, aby všechny řetězce měly zrušeny význam speciálních znaku, pokud nějaké obsahují. A jsou zde i další typy proměnných, které musí být a jsou speciálně ošetřeny tak, aby se vložením vhodné hodnoty nedaly zneužít. Vzhledem k tomu, že na tomto serveru leží zodpovědnost za bezpečnost systému, je mu věnována i odpovídající pozornost. Je zde snaha o malý a přehledný kód. Důvodem je minimalizování možnosti výskytu chyby a maximalizování výkonu.

3.2 Apollo

Apollo je tenký klient, určený hlavně pro zaměstnance VUT, kteří požadují rychlé, uživatelsky příjemné rozhraní a budou často pracovat s IS VUT. Celý systém je samozřejmě modulární, takže k uživateli se stahují jen části, které používá. Tento klient zajišťuje automatickou aktualizaci svou i

svých modulů (načítání aktuální poslední verze), připojení k Asta serveru, autentizaci, zamykání po delší době nečinnosti, ukládání nastavení na serveru, tisk, přístup k nápovědě a spouštění modulu. Moduly jsou tvořeny DLL knihovnamí s určeným rozhraním. Rozhraní je tvořeno dvěma funkcemi. Funkce GetVer vrací verzi modulu, čehož se využívá při automatické aktualizaci a funkce CreateFrame zajišťuje vytvoření Apollo framu. Frame je následně zadokován v Apollu a jsou zavolány odpovídající události tohoto framu. Veškerá další komunikace mezi Apollem a framem probíhá právě pomocí událostí Frame

4 Analýza

4.1 Současný stav hodnocení rizik

Proč je potřeba vůbec hodnotit jednotlivá rizika a to na všech součástech VUT? Je to z toho důvodu, že jednou za čas (momentálně je toto období stanoveno na půl roku) se vyhodnotí rizika hodnocená v tomto období a vytvoří se statistika. Ta obsahuje všechna hodnocená rizika popořadě od těch nejzávažnějších po ty nejméně závažná. Na závažnost takového rizika má vliv jak stupeň závažnosti, kterým ohodnotila riziko daná součást, tak i to, kolik součástí toto riziko ohodnotilo. U těch nejzávažnějších poté VUT řeší možná opatření, která by mohla vést ke snížení jejich stupně závažnosti. V budoucnu bude například tento seznam ohodnocených rizik potřeba k novým projektům, aby se daly čerpat dotace od Evropské unie na tyto projekty. Z toho důvodu je hodnocení těchto rizik důležité. Rizikové procesy se nehodnotí. Jejich hodnocení se odvozuje od hodnocení rizik, které se k těmto rizikovým procesům váží.

Pro hodnocení rizik se doposud používal Microsoft Excel. Pověřeným lidem na jednotlivých součástech VUT se posílal soubor této aplikace a v něm bylo několik záložek, které symbolizovaly jednotlivé oblasti rizik, která se hodnotila. Na každé záložce byl seznam rizik a rizikových procesů. Byl zde sloupec „kód procesu“, který identifikuje příslušný rizikový proces. Skládá se z jednoho znaku, na který začíná název oblasti do které patří, a tří číslic, představující jeho číslo. Následuje sloupec obsahující název daného rizikového procesu. Po něm následuje číslo rizika, které se k tomuto rizikovému procesu vztahuje. Po čísle rizika je sloupec s názvem tohoto rizika. Následují dva sloupce, které zástupci součástí vyplňují pokud dané riziko hodnotí. V jednom se zadává hodnota pro dopad daného rizika. Zde se zadává hodnota od 1 (minimální dopad) do 5 (maximální dopad). Vyjadřuje jak velký může mít dané riziko dopad. V druhém sloupci se vyplňuje pravděpodobnost daného rizika a opět se zde zadávají hodnoty od 1 (minimální pravděpodobnost) do 5 (maximální pravděpodobnost). Poslední sloupec se pomocí funkce doplní sám. Vyjadřuje stupeň významnosti rizika a počítá se jako násobek pravděpodobnosti rizika a dopadu rizika. U řádku který obsahuje informace o rizikovém procesu se ve sloupci „Dopad“ vypočítává průměrná hodnota dopadu u hodnocených rizik pro tento proces. Ve sloupci „pravděpodobnost“ se opět vypočítává průměrná hodnota pravděpodobnosti u hodnocených rizik pro tento proces. A ve poslední sloupci u procesu vyjadřuje stupeň významnosti procesu a obsahuje průměrnou hodnotu stupňů významnosti rizik, které náležejí k tomuto rizikovému procesu.

4.1.1 Statistiky ohodnocených rizik

Statistika takto ohodnocených rizik se dělala pro celé VUT a byla označována jako plán rizik. Zaměstnanec, který byl za tyto statistiky zodpovědný se staral jak o rozesílání seznamu rizik příslušným lidem na jednotlivé součásti VUT, tak o zpracování ohodnocených rizik, které obdržel od jednotlivých součástí. Rizika (rizikové procesy) mohly mít maximální stupeň významnosti rizika (procesu) 25. Byly to rizika (procesy) které měly ohodnoceny dopad i pravděpodobnost maximálními hodnotami (5). Tato statistika se vyhodnocovala jednou za rok označovala jako plán rizik pro VUT.

4.1.2 Nevýhody současného stavu hodnocení rizik

Hlavní nevýhodou současného stavu hodnocení rizikových procesů bylo malé rozpětí hodnot stupně významnosti jednotlivých rizik. Tyto stupně významnosti rizik nabýval jen hodnot od 0 do 25. Tudiž pokud bylo potřeba identifikovat v statistice největší rizika bylo jich hned několik (celkem hodně rizik mělo stupeň významnosti právě 25). Bylo potřeba tuto škálu možných hodnot rozšířit. Druhou nevýhodou bylo to, že výslednou statistiku musel ručně vypracovávat zaměstnanec pověřený vytvořením plánu rizik. Při požadavku na přidání nového rizika nebo rizikového procesu musel tento zaměstnanec riziko nebo rizikový proces do seznamu přidat dříve, než byly seznamy rizik rozeslány k ohodnocení. Chybělo centrální uložení informací o těchto rizikách, o jejich minulých i současných ohodnoceních a možnosti tyto rizika rovnou ohodnocovat.

4.2 Požadavky na systém

Pro veškeré potřebné úkony nad daty vztahující se k hodnocení rizik byl zvolen informační systém VUT Apollo, ve kterém budu vytvářet modul pro veškeré operace potřebné pro editaci rizik a rizikových procesů, jejich hodnocení jednotlivými součástmi a zobrazování statistik. Data budou uložena v centrální databázi Oracle 9i.

4.2.1 Editace oblastí

Modul bude umožňovat editovat oblasti rizikových procesů. Každý rizikový proces bude muset patřit k nějaké oblasti rizikových procesů. Mezi atributy oblasti patří jen její název. Protože se přepokládá, že rizikových procesů bude přibývat, bude potřeba tyto oblasti přidávat, editovat (měnit jejich název) a případně i mazat. Zde se musí ošetřit, aby nedošlo ke smazání oblasti ve které jsou již ohodnocené rizikové procesy.

4.2.2 Editace rizikových procesů

V modulu bude možné rizikové procesy přidávat, editovat i mazat. Mezi atributy rizikového procesu patří oblast, ke které se rizikový proces váže, kód procesu a název procesu. Kód procesu bude číslo, které bude rizikovému procesu přidělované automaticky. V každé oblasti bude toto číslování začínat od čísla 1 a pokračovat vždy o jedna vyšším číslem. Při mazání procesu se bude testovat, zda některé riziko patřící k tomuto procesu již nebylo hodnoceno. Pokud ne, proces bude smazán i se všemi riziky, které obsahuje.

4.2.3 Editace rizik

Opět se budou moci rizika přidávat, editovat i mazat. Atributy rizika bude rizikový proces, ke kterému se riziko vztahuje, kód rizika, který se bude pro každý proces číslovat od 1 a postupně o jedna větší, název rizika a jeho platnost. U jeho platnosti se bude zadávat od kdy je riziko platné a do jaké doby bude riziko platné. Pokud nebude riziko platné v dané době, nebude ho možné hodnotit. U mazání rizika se bude testovat zda již nebylo riziko hodnoceno.

4.2.4 Hodnocení rizik

Zde bylo nutné se zamyslet nad tím, jak zvětšit škálu ohodnocení jednotlivých rizik. Z toho důvodu se vytvořilo 7 faktorů, které budou ovlivňovat celkové ohodnocení daného rizika. Každý faktor bude mít 5 stupňů, které budou vyjadřovat míru významnosti daného faktoru v hodnocení daného rizika. Původně se uvažovalo, že by mohly být faktory vázané na určitou oblast a tak by se u každé oblasti rizikových procesů mohly vytvářet a následně pomocí nich i hodnotit různé faktory. Prozatím se ale zvolilo jiné řešení z toho důvodu, že by se v tomto případě stejně musel dodržovat stejný počet faktorů pro každou oblast, ale i díky tomu, že jednotlivé faktory se dají aplikovat na většinu rizik. Zvolilo se tedy nakonec řešení, ve kterém bude existovat jen jedna množina faktorů pro všechny oblasti rizikových procesů. Bylo ale nutné respektovat fakt, že u některých rizik se nedají hodnotit všechny faktory, protože na riziko nemají žádný vliv, a kdyby se u nich zvolila nejmenší míra vlivu daného rizika, zmírnilo by to neprávem celkový stupeň závažnosti daného rizika. Z toho důvodu se k daným 5 stupňům u každého faktoru přidal stupeň, který označuje nulový vliv faktoru a na výsledný stupeň významnosti rizika nebude mít tento faktor vliv. Při samotném hodnocení rizik se bude muset zadat dopad a pravděpodobnost daného rizika (číslo od 0 do 5, 0 – nejmenší, 5 - největší). Dále se bude muset ohodnotit nejméně jeden faktor (všechny faktory mohou mít nulový vliv, ale alespoň u jednoho faktoru musí být zvolen jiný než nulový vliv). Při každém hodnocení musí být popsány opatření, které daná součást provedla a měly vliv na dané riziko. Hodnocení jednotlivých rizikových procesů a jejich rizik se bude vyhodnocovat jednou za půl roku. Během tohoto pololetí budou moci

zástupci jednotlivých součástí svá hodnocení rizik k danému pololetí měnit. Hodnocení rizikových procesů a rizik si budou moci tito zástupci prohlížet i z minulých let či minulého pololetí, ale nebudou je už moci měnit.

4.2.5 Faktory

Faktory jsou u hodnocení rizik vytvořeny z toho důvodu, aby více zpřesnily hodnocení jednotlivých rizik. Samozřejmě je toto zpřesnění závislé na objektivním přístupu k výběru hodnot u jednotlivých faktorů při hodnocení rizik. Konečná podoba jednotlivých faktorů a jejich hodnot ukazuje tabulka v příloze.

4.2.6 Váha faktorů

Při konečném výpočtu stupně významnosti rizika se násobí hodnota dopadu, pravděpodobnosti a váha faktorů, která tím pádem hraje důležitou roli. Váha faktorů se počítá jen z faktorů, u kterých je při hodnocení rizika vybrán nenulový vliv. Váha je pak počítána tak, že se nejprve sečtou hodnoty faktorů ohodnocených nenulovým vlivem (hodnoty mezi 1 a 5). Tento součet se pak vydělí hodnotou, která je rovna součinu počtu faktorů ohodnocených nenulovým vlivem a maximální možnou hodnotou faktoru (5). Výsledek je ještě vynásoben stem a vyjadřuje váhu faktorů. Pro přehlednost je zde uveden vzorec:

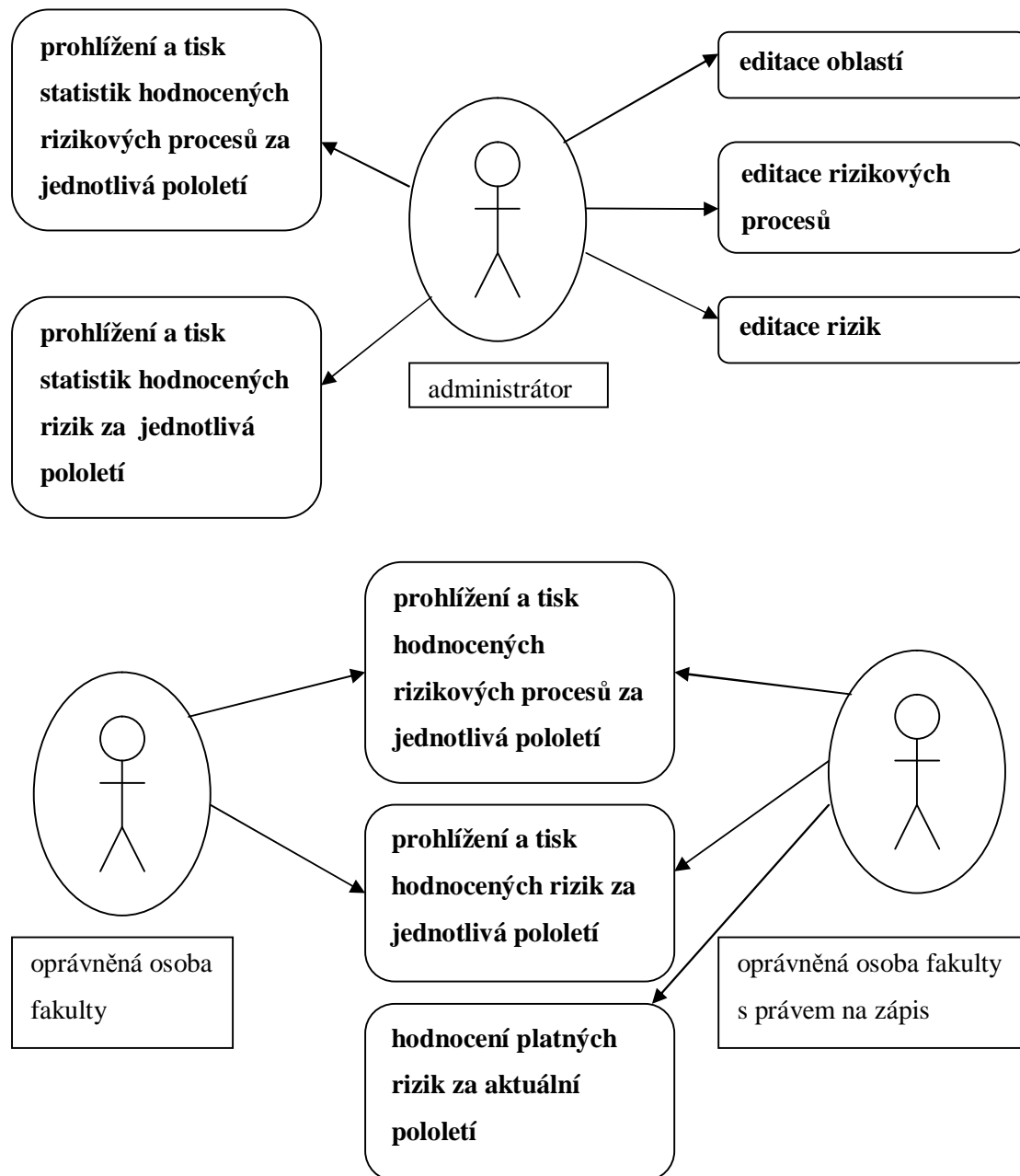
$$\frac{\sum h}{p * 5} * 100 = \text{váha}$$

$\sum h$ - součet hodnot všech faktorů hodnocených nenulovým vlivem (hodnoty 1 - 5)
 p - počet faktorů hodnocených nenulovým vlivem

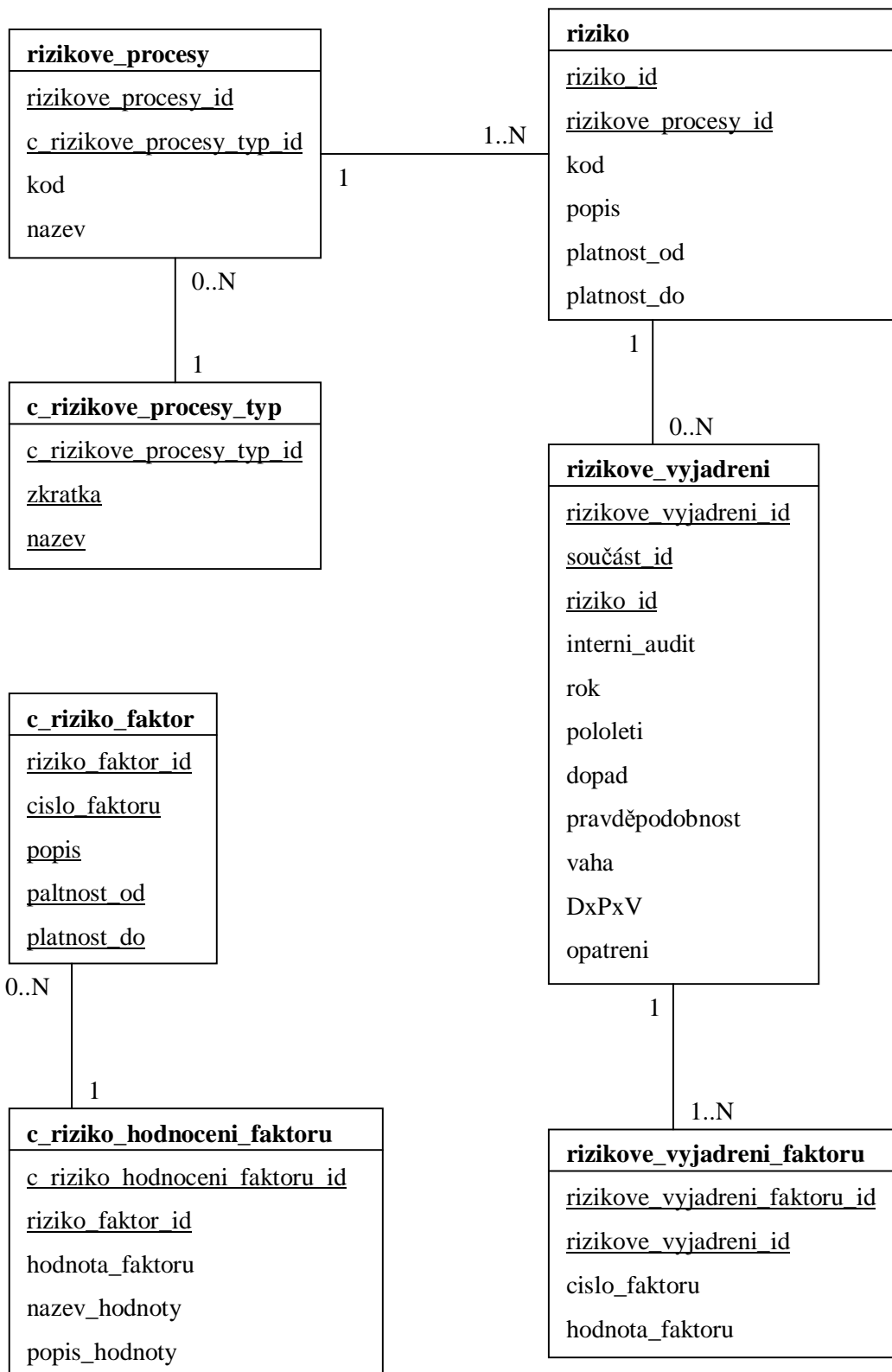
Díky takto zvolenému výpočtu je například při zadání maximálních hodnot u dvou faktorů rovna 100. Stejnému počtu bude rovna, pokud touto maximální hodnotou bude ohodnoceno 5 faktorů. Co by se mohlo zdát jako nevýhoda je pak to, že pokud uživatel ohodnotí jeden faktor maximální hodnotou a druhý minimální, váha bude rovna 60. Zde stojí do budoucna za úvahu, jestli mají všechny faktory stejnou váhu na vytváření celkové váhy faktorů. Jinými slovy jestli by některé faktory neměly mít na výpočet váhy větší vliv než jiné faktory.

4.3 Diagramy

4.3.1 Diagram případu použití



4.3.2 ER-diagram



5 Implementace

Protože mnou vytvořený modul bude součástí aplikace Apollo, který je napojen na centrální databázi, tak jsem veškeré tabulky vytvářel právě v této databázi.

5.1 Borland Delphi

Vývojový nástroj Borland Delphi byl vyvinut firmou Borland Software Corporation a jeho první verze vyšla již v roce 1995. Toto vývojové prostředí je určeno pro systémy Windows a jeho překladač je plně 32bitový. Každá verze vychází ve třech edicích. Edice Standard je určena začátečníkům v oblasti programování a neobsahuje zdrojové texty knihoven komponent a chybí zde i podpora pro práci s databázemi. Další edice Professional již obsahuje zdrojové kódy knihoven komponent, její omezení však spočívá v absenci komponent pro vývoj aplikací klient/server. Dalším omezením je nemožnost přistupovat k dotazům pomocí technologie ADO. Poslední edicí je Enterprise a ta již představuje kompletní vývojový nástroj.

Já jsem pro vývoj mé aplikace používal Borland Delphi Professional verze 6. Vývojové prostředí Borland Delphi je založené na jazyce Object Pascal. Samotné Delphi je založeno na objektově orientované architektuře. Vývojové prostředí se potom skládá z horního menu, ve kterém můžete vybrat požadovanou operaci, kterou chcete provést (vytvoření nového projektu, kompilace projektu atd.). Pod tímto menu jsou na levé straně dvě lišty obsahující ikony s nejpoužívanějšími operacemi. Po pravé straně od těchto lišt je několik záložek. Každá záložka přitom obsahuje komponenty, které jsou seskupeny v této záložce podle určité podobnosti jejich významu (použití). Největší část prostoru pak při tvorbě projektu zabírá formulář, na který se pak vkládají komponenty umístěné v záložkách výše a pomocí nich je takto tvořena vizuální podoba aplikace. Tyto komponenty tedy ve finále tvoří kostru aplikace. Některé komponenty za běhu programu vidět jsou, jiné nikoli. Komponenty jsou reprezentovány takzvanými objekty. Všechny tyto objekty tvoří mezi sebou hierarchii a jsou sdruženy v knihovně vizuálních komponent. To jakou moc představuje Delphi při návrhu aplikací potvrzuje i ten fakt, že samotné Delphi je vytvořeno právě z této knihovny vizuálních komponent. Další nedílnou součástí tohoto vývojového prostředí je Object Inspector umístěný v levé části tohoto vývojového prostředí. Jeho obsah se dynamicky mění v závislosti na tom, jakou komponentu jste právě vybrali. U každé komponenty se nastavují dvojice základní atributy. Její vlastnosti a události. Proto jsou i v tomto Object Inspektoru dvě záložky. Záložce Vlastnosti (Properties) se nastavují jednotlivé vlastnosti vybrané komponenty. To se provádí tak, že jen vyberete v tomto seznamu název vlastnosti v levé sloupci a v pravém sloupci nastavíte hodnotu této vlastnosti. U některých vlastností (Písmo) se nastavuje více dílčích vlastností a proto je u názvu této vlastnosti

malé plus, na které pokud kliknete rozbalí se seznam všech dílčích vlastností, které pak můžete nastavovat. Záložka Události obsahuje seznam událostí, na které může daná komponenta reagovat provedením nějaké operace. V levém sloupci je název události. Odkud chceme nějakou událost ošetřit provedením nějakého kódu, poklepeme dvakrát levým tlačítkem do pole v pravém sloupci u názvu této události a Delphi za nás do zdrojového kódu aplikace samo deklaruje tuto obslužnou proceduru a otevře se okno Code Editoru s nadefinovanou prázdnou procedurou ošetřující tuto událost. Programátor zde pak jen doplní svůj kód. V tomto Code Editoru tedy programátor píše svůj kód. Hodně věcí však za něj udělá samo Delphi. Například při vkládání komponent na formuláře provede i nadefinování těchto komponent ve zdrojovém kódu. To stejné platí pro nastavování vlastností u jednotlivých komponent v Object Inspektoru. Tyto změny vlastností se také automaticky promítnou do kódu aplikace. Pro přepínání mezi náhledem jak budou jednotlivé formuláře vypadat, který se používá pro vkládání a umisťování jednotlivých komponent na formulář, a Code Editorem, ve které se nachází zdrojový kód aplikace, se používá tlačítko F12. Další součástí Delphi, která programátorovi usnadní jeho práci, je Code Completion. Napíše-li programátor jméno třídy nebo proměnné, u kurzoru se mu rozbalí seznam všech jeho vlastností, událostí a metod. Pokud začne psát další písmena, tento seznam se zužuje, aby vyhovoval již zapsaným znakům. Tento seznam může být programátoru nabídnut i pokud nenapíše žádný název třídy nebo proměnné. Poté se objeví seznam všech proměnných a tříd, které může programátor vybrat. Tento seznam se vyvolá stiskem kláves Ctrl + Spacebar. Co se týče překladu a spuštění samotného projektu, Delphi obsahuje svůj vlastní kompilátor. Pro spuštění projektu nakonec celý projekt programátor zkompiluje a kompilátor vytvoří spustitelný soubor. Kompilace se provádí buďto stisknutím tlačítek Ctrl + F9 nebo volbou v menu Project -> Compile Project. Podobný efekt má i zadání project -> build project spuštěný z hlavního menu, který na rozdíl od příkazu Compile Project zkompiluje všechny soubory v programu bez ohledu na to, byly-li změněny. Spuštění projektu pak programátor provede kliknutím na soubor s příponou exe nebo ho spustí přímo z Delphi v jeho integrovaném Debuggeru. To se provádí zvolením Run -> Run v hlavním menu nebo stiskem kláves Ctrl + F9. Nápověda je volána pomocí stisku tlačítka F1. Jedná se o kontextovou nápovědu což znamená, že se otevře téma bezprostředně související s aktuálně vybraným prvkem. Velmi užitečný je i rejstřík, který se nachází v samostatné záložce v této nápovědě a ve kterém zpravidla najdete všechny položky, jejichž popis byste mohli potřebovat. V této nápovědě se nacházejí jak informace vztahující se k ovládání samotného Delphi, tak k popisu položek hlavního menu, podrobnému popisu všech komponent, vlastností, událostí, datových typů a příkazů jazyka Object Pascal. Dalším typem nápovědy je tzv. bublinková nápověda, která je také někdy označována jako plovoucí. Jedná se o žlutý obdélníček s vysvětlujícím textem, který se zobrazí když podržíte kurzor myši nad prvkem, jehož popis vás zajímá.

5.2 Návrh databázového schéma

Centrální databáze VUT má několik instancí. Instance označená jako CDBX obsahuje platná data. Proto je také označována jako „ostrá“. Vedle ní je zde také instance CISD, která je pravidelně aktualizována platnými daty z instance CDBX. Data z této instance využívá testovací verze informačního systému Apollo, kterou většinou využívají uživatelé k vyzkoušení práce s jednotlivými moduly informačního systému Apolla aniž by mohly svojí prací napáchat škody v platných datech. Třetí instance CISB je označována jako vývojová databáze. Využívají ji většinou vývojáři při vyvíjení jednotlivých částí informačního systému VUT. Tuto instanci databáze jsem také využíval při tvorbě mé aplikace.

Pro návrh databázového schématu pro hodnocení rizikových procesů jsem využil program Case Studio. Zde jsem si vytvořil jednotlivé tabulky i s vlastnostmi a vazbami mezi nimi. Barevně jsem si zde i odlišil číselníky a stěžejní tabulky.

Pro ladění SQL dotazů jsem využíval program PL/SQL Developer.

Při navrhování tabulek jsem se držel zavedených konvencí. Názvy tabulek jsou jednoduché a výstižné. Název sloupce obsahující identifikátor tabulky začíná stejně jako název samotné tabulky a je ukončen „_id“. Je to vždy primární klíč a pro jeho generování se používá sekvence uložená v DB. Pro každou tabulku je jiný, jeho název se skládá z „sq_“ a názvu dané tabulky. Při vkládání nového záznamu je tento sekvence požádán o hodnotu a vrací vždy o jedno větší číslo. Jedná se tedy o posloupnost. U každého sekvenceru se dá nastavit, na jaké hodnotě bude začínat. Součástí zavedených konvencí obsahuje také každá tabulka 3 sloupce:

upd_uid – zde je uloženo osobní číslo uživatele, který záznam naposledy změnil

upd_ts – zde je uložen přesný čas poslední změny tohoto záznamu

status – ten nabývá čtyř hodnot, 0 nebo 9 značí že záznam je platný (0 – vytvořen synchronizací ze starého studijního systému, 9 – nově vytvořený záznam), 1 a -1 značí že záznam je neplatný (1 pro smazané záznamy, které by z nějakých důvodů měly ještě zůstat v databázi, -1 pro záznamy určené k vymazání)

5.2.1 Tabulka c_rizikove_procesy_typ

Název této tabulky začíná na c z důvodu že se jedná o číselník. Tato tabulka obsahuje jednotlivé oblasti rizikových procesů. Každá zde vložená oblast má svojí zkratku a svůj název.

název sloupce	popis
c_rizikove_procesy_typ_id	id oblasti
zkratka	zkratka oblasti
název	celý název oblasti

5.2.2 Tabulka rizikove_procesy

V této tabulce jsou uloženy všechny rizikové procesy. Každý rizikový proces bude patřit do nějaké oblasti. Dále má každý rizikový proces svůj kód, který je mu přidělován při jeho vytvoření. Jedná se o číslo z posloupnosti čísel. Tato posloupnost začíná u každé oblasti rizikových procesů číslem jedna. Vedle kódu patří mezi jeho atributy i jeho název.

název sloupce	popis
rizikove_procesy_id	id rizikového procesu
c_rizikove_procesy_typ_id	id oblasti rizikových procesů, ke které patří
kod	kód rizikového procesu
nazev	název rizikového procesu

5.2.3 Tabulka riziko

Zde jsou uložena všechna rizika. Každé riziko patří k nějakému rizikovému procesu. Mezi další atributy rizika patří jeho kód. Ten tvoří číslo z posloupnosti čísel, která je u daného rizikového procesu. Prvnímu riziku v rámci daného rizikového procesu je přidělen jako kód číslo 1. Dalšímu riziku k tomuto rizikovému procesu je přiděleno jako kód číslo o jedno větší atd. Při tvorbě rizika se také zadává jeho popis a datum od kterého platí a datum po které dané riziko platí. Rizika, která v dané pololetí nejsou platná nemohou být ani hodnoceny.

název sloupce	popis
riziko_id	id rizika
rizikove_procesy_id	id rizikového procesu, ke kterému toto riziko patří
kod	kód rizika
popis	popis rizika
platnost_od	datum od kterého riziko platí
platnost_do	datum kterým platnost rizika končí

5.2.4 Tabulka rizikove_vyjadreni

Tato tabulka slouží k uchování základních informací o hodnocení jednotlivých rizik. Pokud oprávněná osoba na dané součásti VUT ohodnotí nějaké riziko, do této tabulky se uloží jednotlivé atributy hodnocení. Mezi ně patří identifikátor součásti VUT, ke které se toto hodnocení vztahuje, rok a pololetí, ke kterému se hodnocení vztahuje, zvolenou míru dopadu a pravděpodobnosti, váhu, která je vypočítávána z ohodnocených faktorů, násobek pravděpodobnosti, dopadu a váhy, který se využívá při statistikách, a nakonec opatření, která provedla fakulta a která změnila ohodnocení tohoto rizika.

název sloupce	popis
rizikove_vyjadreni_id	id rizikového vyjádření
soucast_id	identifikátor součásti VUT, ke které se hodnocení vztahuje
riziko_id	kód rizika, které je hodnoceno
rok	rok, ve kterém je riziko hodnoceno
pololeti	pololetí, ve kterém je riziko hodnoceno
dopad	míra dopadu rizika (1 - 5)
pravdepodobnost	míra pravděpodobnosti vzniku rizika (1 - 5)
vaha	hodnota vypočtená z hodnocených faktorů
DxPxV	násobek dopadu, pravděpodobnosti a váhy
opatreni	opatření, která provedla daná součást VUT k tomuto riziku

5.2.5 Tabulka rizikove_vyjadreni_faktoru

Aby mohla pověřená osoba na dané součásti VUT hodnocení nějakého rizika vůbec uložit, musí zadat alespoň jeden z faktorů s nenulovým vlivem. Do této tabulky se ukládají všechny hodnoty faktorů, které daný uživatel ohodnotí nenulovým vlivem. Ukládá se zde identifikátor rizikového vyjádření, u kterého byly tyto faktory zadány, číslo faktoru (každý faktor má své jedinečné číslo k jeho identifikaci) a hodnota faktoru, která u faktorů může nabývat hodnot 1 až 5.

název sloupce	popis
rizikove_vyjadreni_faktoru_id	id rizikového vyjádření faktoru
rizikove_vyjadreni_id	id rizikového vyjádření, ke kterému se ohodnocení tohoto faktoru vztahuje
cislo_faktoru	identifikační číslo faktoru
hodnota_faktoru	zvolené hodnota u daného faktoru

5.2.6 Tabulka c_riziko_faktor

Tato tabulka obsahuje seznam všech faktorů. Pro každý faktor se zde ukládá číslo faktoru, popis faktoru, datum od kdy má faktor platnost a datum kdy tato platnost končí.

název sloupce	popis
riziko_faktor_id	id faktoru
cislo_faktoru	číslo daného faktoru
popis	slovní popis faktoru
platnost_od	datum, kterým začíná jeho platnost
platnost_do	datum kdy jeho platnost končí

5.2.7 Tabulka c_riziko_hodnoceni_faktoru

V této tabulce jsou uloženy jednotlivé hodnoty faktorů. Pro každou hodnotu se zde ukládá identifikátor faktoru, ke kterému se tato hodnota vztahuje, hodnota (číslo od 0 do 5) název této hodnoty (př. nízký vliv) a popis této hodnoty.

název sloupce	popis
riziko_hodnoceni_faktoru	id hodnoty faktoru
riziko_faktor_id	id faktoru, ke kterému se hodnota vztahuje
hodnota_faktoru	číslo od 0 do 5
nazev_hodnoty	zkrácený název
popis_hodnoty	sovní popis dané hodnoty

5.3 Vytvoření modulu

Pro vytvoření modulu jsem použil vývojový nástroj Delphi Professional od firmy Borland. I když toto vývojové prostředí nabízí nepřeberné množství komponent, jen málokdy jsem nějakou použil. Vývojové oddělení CVIS, které se tvorbou těchto modulů pro informační systém Apollo zabývá si totiž většinu komponent navrhl samo. Z toho důvodu jsem musel před samotným programováním modulu nainstalovat nezbytné balíčky komponent Apolla a nastavit jednotlivé výstupy a vstupy tohoto vývojového prostředí Borland Delphi. Potom jsem vytvořil hlavní ApFrame, na který jsem postupně vkládal jednotlivé komponenty a sestavoval tak celý modul. Ten jsem kompiloval do knihovny exRizikoveProcesy.dll. Tuto knihovnu jsem uložil do databáze k ostatním knihovnám a navázal k ní i jednotlivá práva, která se k modulu váží. O tom jsem také musel vytvořit odkaz, ze kterého se bude tento modul v informačním systému Apollo spouštět. V horní liště informačního systému Apollo je menu s jednotlivými záložkami. Odkaz na tento modul jsem umístil pod položku VUT a záložku jsem nazval „Rizikové procesy“. Dále jsem k této záložce i nastavil práva, díky kterým se uživateli vůbec tento odkaz v záložce VUT zobrazí. Poté jsem do databáze ukládal i jednotlivé SQL dotazy. Pod jedním názvem Query se zde dá uložit hned několik různých dotazů. Jsou zde záložky pro různé typy dotazů. Mezi ně patří vytvářecí dotazy (záložka Create), dotazy na modifikace uložených dat (záložka Modify), dotazy určené pro mazání dat (záložka Delete) a klasické dotazy, které vracejí data, (záložka View). U takto uloženého dotazu lze ještě nastavit, s jakým právem bude uživatel moci daný SQL dotaz vůbec spustit. Práva zde lze přidělit na jednotlivé dotazy (záložky). To vše přispívá k bezpečnosti operací prováděnými nad daty a tím i k ochraně před neoprávněnou manipulací s daty.

5.3.1 Práva a role

Ne každý uživatel informačního systému Apollo má mít přístup do tohoto modulu a proto jsem pro něj musel vytvořit práva k přístupu do něj. Vytvořená práva jsou následující:

- `apolloRizikoveProcesyR` – díky tomuto právu může uživatel modul spustit a může v něm číst hodnocení rizik ze všech období a ze součástí VUT, které má přiděleny v roli `AP_RIZIKPROCESY` a která se k tomuto právu váže. Pokud tedy nebude mít v této roli přidělenou žádnou součást neuvidí v modlu ani žádná hodnocená rizika
- `apolloRizikoveProcesyW` – toto právo je jen rozšiřující právo k právu `apolloRizikoveProcesyR`. Toto právo ke spuštění modulu nestačí ale spolu s právem `apolloRizikoveProcesyR` umožňuje ukládat hodnocení jednotlivých rizik za součást VUT, kterou má přidělenou opět v roli `AP_RIZIKPROCESY`.
- `apolloRizikoveProcesyA` – toto právo je určeno pro administrátora a uživatel s tímto právem může modul spustit. Není navázáno na žádnou roli a uživateli umožňuje editování oblastí, rizikových procesů a rizik, umožňuje mu také zobrazení a tisk statistik za celé VUT

Co se týče toho, jak jsem pracoval s právy v samotné aplikaci, tak díky tomu, že jsem na hlavním `ApFramu` vytvořil záložky, tak jsem viditelnost těchto záložek navázal na práva. Vytvořil jsem si proměnnou s názvem `admin` a typem `Boolean` a pokud měl uživatel právo `apolloRizikoveProcesyA`, nastavil jsem proměnnou `admin` na `True` a zviditelnil mu příslušné záložky s editací oblastí, rizik a rizikových procesů a záložky se statistikami. Pokud uživatel neměl toto právo zobrazoval jsem mu zbytek záložek pro prohlížení hodnocení rizik za jednotlivá období. Vytvořil jsem si nový datový typ `TPrava`, který obsahoval 2 hodnoty („cteni“ a „zapis“). Poté jsem si vytvořil proměnnou „prava“ tohoto typu a pokud měl uživatel jen právo na čtení (`apolloRizikoveProcesyR`) naplnil jsem proměnnou „prava“ hodnotou „cteni“ a jinak pokud měl i právo na zápis (`apolloRizikoveProcesyW`) naplnil jsem proměnnou „prava“ hodnotou „zapis“. Podle této proměnné jsem uživateli, pokud měl uživatel i právo na zápis, umožnil v aktuálním období i hodnocení vytvářet a editovat pro jednotlivá rizika.

5.3.2 Hlavní Frame

Tento `ApFrame` se zobrazí každému uživateli, který spustí tento modul. Jsou na něm umístěny jednotlivé záložky, při jejichž zvolení se zobrazí příslušný `ApFrame` spojený s danou záložkou. Viditelnost jednotlivých záložek je nastavena podle práv uživatele (viz. kap. 3.2.1). Pokud se přihlásí

uživatel s administrátorským právem zobrazí se mu na výběr tři záložky: Statistika rizik, Statistika procesů a Editace. Pokud se přihlásí uživatel s právem pro čtení (případně i pro zápis) jsou mu zobrazeny dvě záložky: Seznam rizik a Seznam procesů. Dále jsou zde v pravé části horní lišty za jednotlivými záložkami celkem čtyři rozbalovací seznamy. Jsou to komponenty typu *TApComboBox* a první z nich obsahuje názvy jednotlivých rizikových oblastí, druhá zobrazuje jednotlivé součásti VUT, které má uživatel v roli AP_RIZIKPROCESY přiděleny, třetí zobrazuje kalendářní roky a poslední zobrazuje jednotlivá pololetí. Dále jsou na tomto framu také uloženy komponenty typu *TRvCustomConnection*, které se využívají při plnění tiskových sestav daty. Je zde také objekt typu *TApIkony*, díky němuž lze tlačítkům, záložkám a jiným komponentám přidělovat různé ikony.

5.3.3 Obarvování jednotlivých řádků v tabulkách

V jednotlivých záložkách popsaných níže jsem použil pro zobrazování jednotlivých rizik nebo rizikových procesů Gridy Apolla. Jedná se o *MemoryDBGrid* a *SMDGrid*. Po vložení dat do nich jsem procházel všechny záznamy (řádky) a volal proceduru, která je obarvovala podle stupně závažnosti rizika nebo rizikového procesu. Na levé straně tohoto Gridu byl pak panel s vysvětlením, jaká barva odpovídá jakým hodnotám. V záložce „Statistika rizik za VUT“ odpovídala bílá barva vyřešeným rizikům (ta, která měla stupeň závažnosti roven 0), dále tmavě zelená odpovídala hodnotám stupně závažnosti 1 až 80, světle zelená hodnotám 81 až 160, žlutá hodnotám 161 až 240, světle červená hodnotám 241 až 320 a červená hodnotám 321 až 400. U ostatních záložek zůstaly použité barvy stejné akorát rozsah hodnot se změnil. Pro jednotlivé barvy vypadal následovně: 0, 1 – 500, 501 – 1000, 1001 – 1500, 1501 – 2000, 2001 – 2500.

5.3.4 Záložka Seznam rizik

Po vybrání této záložky se zobrazí *ApFrame*, který bude mít uprostřed tabulku se seznamem rizikových procesů a jejich rizik. Tato tabulka je tvořena komponentou třídy *TMemoryDBGrid*. Její nezbytnou součástí je komponenta třídy *TMemoryDS*. Jedná se o dataset, který se při volání příslušného SQL dotazu naplní daty. Komponenta třídy *TMemoryDBGrid* (tabulka) má jako zdroj dat nastaven právě tuto komponentu třídy *TMemoryDS* a tudíž může následně zobrazovat data z této komponenty. Tento *MemoryDBGrid* umožňuje zobrazovat všechny možné informace týkající se daného rizika. Informace jsou v jednotlivých sloupcích a některé jsou implicitně viditelné a jiné nikoli. Uživatel si samozřejmě může jakékoli sloupce zviditelnit či skrýt. Implicitně jsem nechal viditelné sloupce zobrazující kód rizikového procesu, název rizikového procesu, kód rizika, název rizika a stupeň významnosti rizika. Dále také tento *MemoryDBGrid* umožňuje řadit záznamy v něm

obsažené i podle více jak jednoho sloupce, záznamy můžete filtrovat podle zvolené hodnoty v jakémkoli sloupci. Rizika jsem rozdělil do tří základních skupin:

- Rizika ohodnocená v tomto pololetí – spadají sem všechna rizika, která byla hodnocena v kalendářní rok, který je zvolen již v dříve uváděném rozbalovacím seznamu, a pololetí, které je zvoleno v rozbalovacím seznamu který se nachází hned za rozbalovacím seznamem s kalendářními roky
- Rizika do tohoto pololetí včetně ještě nikdy nehodnocená – sem patří rizika, která nebyla hodnocena v minulosti od výše zvoleného kalendářního roku a pololetí včetně zvoleného pololetí
- Rizika v minulosti od tohoto pololetí vyřešená – sem spadají rizika, které byly v minulosti od výše uvedeného kalendářního roku a pololetí vyřešena (stupeň závažnosti rizika byl nakonec 0)

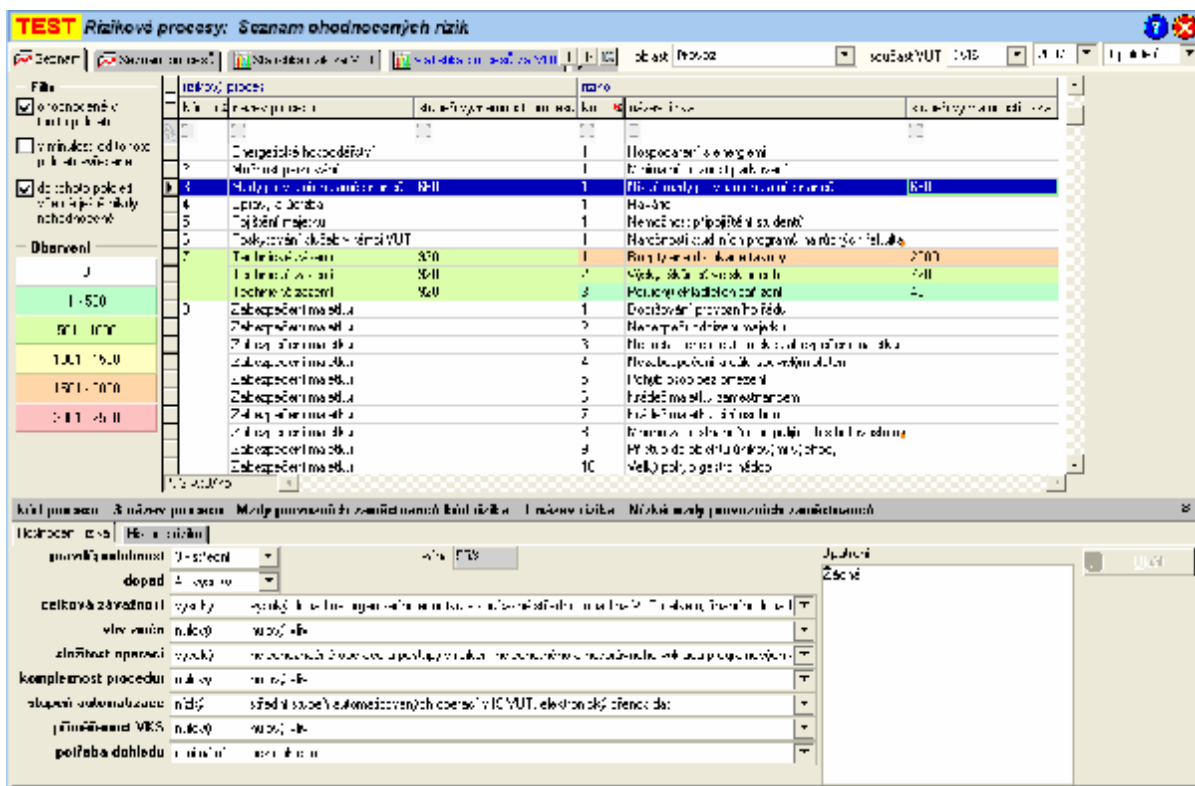
Nalevo od Gridu s riziky je filtr se zaškrťovacími políčky. Každé políčko představuje jednu z těchto skupin rizik a podle toho, která z nich jsou zaškrtnuta, tak takové rizika jsou v Gridu zobrazena. Kromě tohoto filtru má na zobrazení rizikových procesů a rizik také samozřejmě vliv rozbalovací seznam s oblastí, kalendářním rokem a pololetím umístěný výše vedle záložek.

Ve spodní části je umístěna komponenta třídy *TCoolControlDetail*. Jedná se o okno které se dá minimalizovat do tenké lišty (lze ho skrýt). Ve své hlavičce zobrazuje informace o právě zvoleném záznamu v Gridu umístěném výše. Zobrazuje kód procesu, název procesu, kód rizika a název rizika. Pod touto hlavičkou jsou dvě záložky. Jedna s názvem „Hodnocení rizika“ obsahuje informace o hodnocení daného rizika. Druhá je nazvána „Historie rizika“ a v ní se ve formě grafu a tabulky zobrazuje hodnocení rizika v minulosti.

Záložka „Hodnocení rizika“ obsahuje několik komponent. A levé straně jsou to samé komponenty třídy *TApComboBox*. Jedná se o rozbalovací seznamy. První z nich je dopad (obsahuje hodnoty 0 až 5). Pod ním se nachází pravděpodobnost se stejnými hodnotami. Za nimi následuje sedm různých faktorů a každý má jiné hodnoty, které se načítají z databáze. Na této straně je ještě jedna komponenta třídy *TApEdit*, která je jen ke čtení a zobrazuje váhu podle právě zvolených hodnot jednotlivých faktorů. Střední část vyplňuje komponenta třídy *TApMemo*, která slouží k zobrazování (zadávání) opatření u daného rizika. V pravé části je jediné tlačítko s názvem „Uložit“. Pokud má uživatel právo jen na čtení, tak se v této záložce zobrazují detailní informace o hodnocení rizika zvoleného výše v tabulce. Pokud zvolené riziko nebylo hodnoceno budou komponenty v této záložce

prázdné. Tento uživatel ale nebude moci žádné hodnocení měnit ani hodnotit žádné riziko. Pokud bude mít uživatel i právo na zápis a v horní části bude vybrán aktuální kalendářní rok a pololetí, uživatel bude moci jak měnit hodnocení rizika (bude moci změnit dopad, pravděpodobnost, vliv jakéhokoli faktoru s tím že alespoň jeden bude muset mít nenulový vliv, a opatření), tak hodnotit nově ještě nehodnocená rizika.

Záložka „Historie rizika“ zobrazuje historii hodnocení rizika zvoleného v tabulce výše. V levé části se nachází graf. Jedná se o komponentu *TChart*. Nastavil jsem u ní typ grafu sloupcový. Co se týče hodnocení rizika může stupeň závažnosti rizika nabýt maximální hodnoty 2500. To se stane po zvolení maximální hodnoty pro dopad (5), pravděpodobnost (5) a váhu (100 – u všech hodnocených faktorů je vybrána vždy největší míra daného faktoru). Vynásobením těchto tří hodnot dostávám stupeň významnosti rizika. Z toho důvodu výška sloupce v grafu závisí na stupni závažnosti rizika (0 - 2500) . Co se týče horizontální lišty, tak zde je u každého sloupce zobrazen rok a pololetí, kdy bylo toto riziko takto ohodnoceno. V pravé části je komponenta třídy *TSMDGrid*. Jedná se opět o tabulku, která zobrazuje detailní informace o hodnocení daného rizika v jednotlivých obdobích v minulosti. Výslednou podobu této záložky můžete vidět na Obr.3.



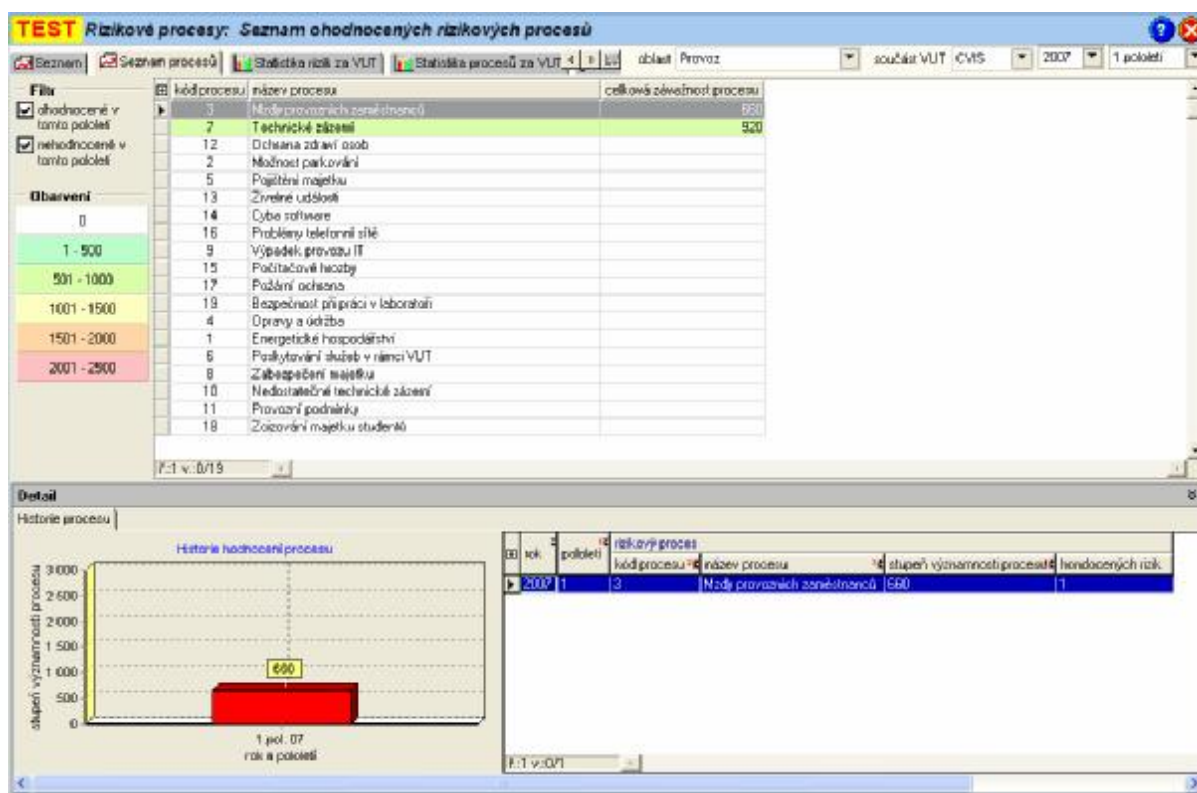
Obr. 3 Záložka se seznamem rizik

5.3.5 Záložka Seznam procesů

Po vybrání této záložky se zobrazí seznam rizikových procesů. Zde jsem použil komponentu třídy *TSMDGrid*. A je naplněna všemi rizikovými procesy z výše zvolené oblasti, kalendářního roku a pololetí. U každého rizikového procesu se zobrazuje kód procesu, název procesu a stupeň významnosti procesu. Ten se spočítá jako průměr všech stupňů významností rizik, která byla u tohoto rizikového procesu v daném roce a pololetí hodnocena. Záznamy v tomto Gridu lze opět řadit i podle více sloupců a filtrovat. V dolní části je opět umístěna komponenta třídy *TCoolControlDetail*. V její hlavičce se opět zobrazují informace o právě zvoleném rizikovém procesu. Je zde zobrazen kód a název rizikového procesu. Má opět dvě záložky.

První záložka se jmenuje „Detail rizikového procesu“. Obsahuje komponentu třídy *TSMDGrid*, která obsahuje jednotlivé hodnocené rizika, ze kterých se počítal stupeň významnosti výše zvoleného rizikového procesu. U každého rizika se zde zobrazují detaily o jeho hodnocení.

Druhá záložka se jmenuje „Historie rizikového procesu“. Ta obsahuje na své levé straně graf, jedná se opět o komponentu třídy *TChart*. Graf je sloupcový. Zobrazuje stupeň významnosti rizikového procesu v minulosti. Maximální stupeň významnosti rizikového procesu je 2500 a proto je tato hodnota i maximální výškou sloupce, který jí charakterizuje. U každého sloupce je pod ním uveden kalendářní rok a pololetí, ke kterému se tento stupeň významnosti rizikového procesu vztahuje. Na pravé straně je komponenta třídy *TSMDGrid*, která obsahuje jednotlivé rizikové procesy a jejich stupně významnosti. Výslednou podobu této záložky můžete vidět na Obr.4.



Obr. 4 Záložka se seznamem rizikových procesů

5.3.6 Záložka Editace

Tato záložka se zobrazuje jen uživatelům s administrátorským právem na tento modul. Po jejím vybrání se uprostřed zobrazí komponenta třídy *TMemoryDBGrid*. Jako zdroj dat má komponentu třídy *TMemoryDS* která je také na tomto framu. Tato komponenta se naplní jednotlivými riziky v závislosti na tom jaká je vybrána oblast rizikových procesů výše. U jednotlivých rizik se zde zobrazuje kód rizika, název rizika, kód a název rizikového procesu ke kterému se vztahuje. Data jsou v tomto Gridu seskupována podle rizikových procesů. V pravé části jsou dvě tlačítka.

První tlačítková název „Nový proces“. Při definici nového procesu je nutné rovnou vytvořit i první riziko v tomto procesu protože jinak ztrácí tvorba rizikového procesu význam. Po stisku tohoto tlačítka se vyprázdní všechny komponenty v záložce „editace“, která se nachází v komponentě třídy *TCoolControlDetail* umístěné ve spodní části. Nyní musí uživatel pro vytvoření nového rizikového procesu zadat název tohoto procesu, kód rizika se nastaví automaticky na hodnotu 1, musí ale zadat název tohoto rizika, dále musí uživatel zvolit oblast pod kterou bude rizikový proces i riziko patřit. Po zvolení oblasti se automaticky vypočte kód rizikového procesu, který je o jedno větší než maximální hodnota kódu procesu ze zvolené oblasti. Poslední ale nepovinné údaje které může uživatel zadat je datum odkdy bude riziko platit a datum po které bude riziko platit. Toto datum spíše značí pololetí od kterého nebo po které bude riziko platit a tudíž se nabízet k hodnocení. Pokud se zde datum nezadá nebude platnost rizika nijak omezena.

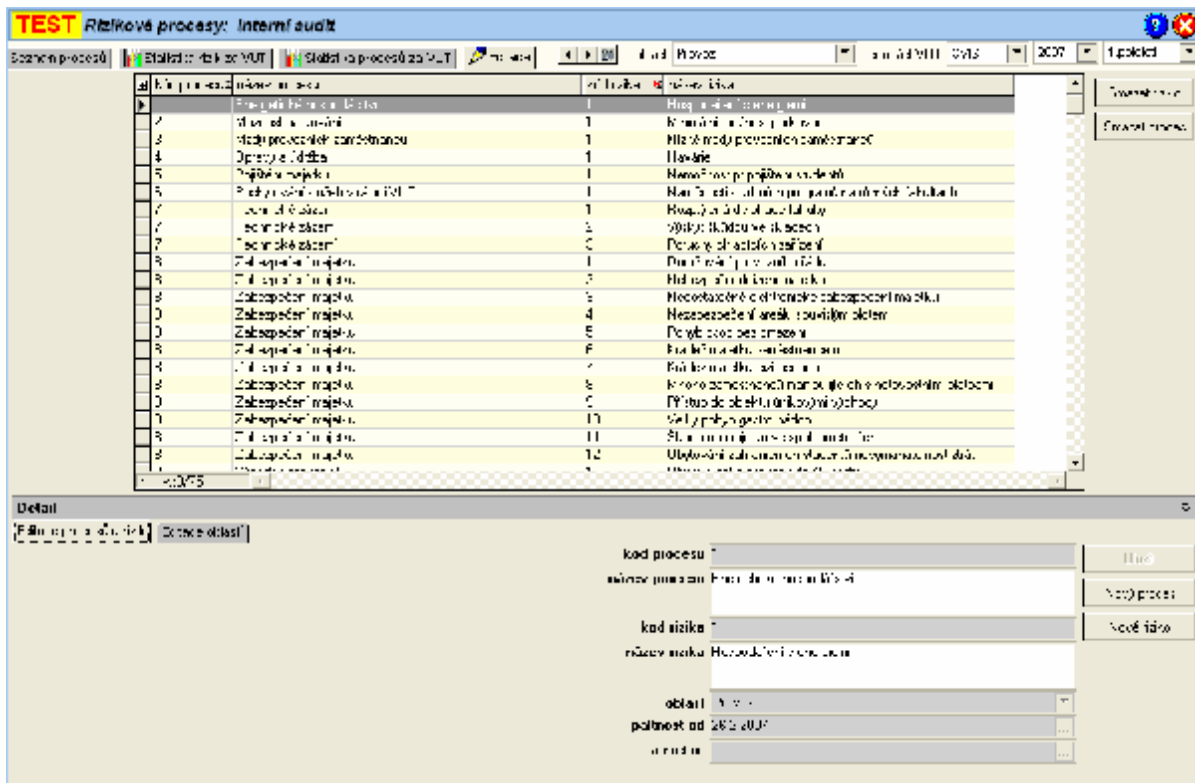
Další tlačítko má název „Smazat rizikový proces“. Po jeho stisknutí se ověří, zda žádné riziko spadající pod tento rizikový proces nebylo ještě hodnoceno a pokud ne upozorní ještě uživatele kolik rizik vztahujících se k tomuto rizikovému procesu bude s tímto procesem smazáno a pokud bude uživatel souhlasit rizikový proces i všechna rizika, která se k němu váží budou smazána. Pokud již některé z rizik vztahujících se k tomuto rizikovému procesu bylo ohodnoceno, zobrazí se uživateli že daný rizikový proces nelze smazat protože obsahuje riziko, které již bylo hodnoceno.

A poslední tlačítko v pravé části se nazývá „Nové riziko“. Po jeho stisku se vyprázdní všechny komponenty týkající se rizika. Riziko se automaticky vytváří k rizikovému procesu, který je právě zvolen v Gridu výše. Podle toho se také automaticky vypočítá kód rizika (je to maximální kód rizika u tohoto rizikového procesu zvětšený o jedna). Uživateli tedy jen zbývá zadat název rizika a volitelně může zadat jeho platnost.

Ve spodní části je komponenta třídy *TCoolControlDetail*. V její hlavičce se zobrazuje kód a název rizikového procesu a kód a název rizika vybraného v Gridu výše. Obsahuje dvě záložky.

První záložka se nazývá „Editace“. V její střední části jsou umístěny komponenty třídy *TApEdit* ve kterých se zobrazuje kód, název rizikového procesu a kód a název rizika vybraného v Gridu výše. Je zde také komponenta třídy *TApComboBox*, ve které je seznam oblastí rizikových procesů a je naplněna oblastí, do které právě zvolené riziko a rizikový proces patří. Poslední dvě komponenty jsou třídy *TApDateTimePicker* a jedna obsahuje datum odkdy je riziko platné a druhá obsahuje datum po které je riziko platné. Uživatel zde může měnit název procesu, název rizika, platnosti rizika a oblast pod kterou daný rizikový proces spadá. V pravé části je tlačítko „Uložit“ pro uložení změn.

Druhá záložka se nazývá „Editace oblastí“. V její levé části se nachází komponenta třídy *TSMGrid*, který je naplněn jednotlivými oblastmi rizikových procesů. U každé oblasti je zde uvedena její zkratka a celý název. Ve střední části jsou komponenty třídy *TApEdit*, ve kterých se zobrazuje zkratka a název oblasti, která je vybrána v Gridu nalevo. U každé oblasti rizikových procesů lze změnit její zkratku nebo název. Na pravé straně je tlačítko s názvem „Uložit“ díky kterému se dají změny uložit. Pod ním je tlačítko s názvem „Nová oblast“. Po jeho zmáčknutí se vyprázdní komponenty obsahující zkratku a název oblasti a uživatel sem může zadat zkratku a název nové oblasti. Výslednou podobu této záložky můžete vidět na Obr.5.

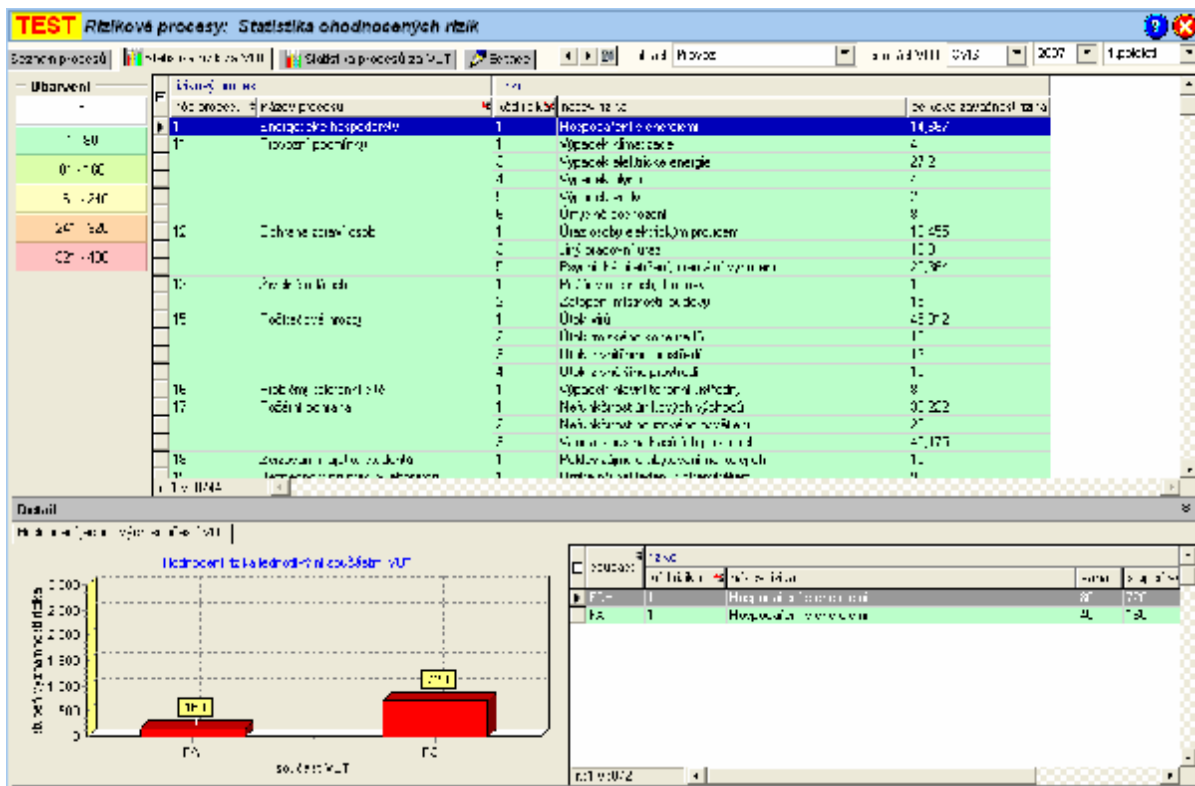


Obr. 5 Záložka pro editaci

5.3.7 Záložka Statistika rizik

Tato záložka se zobrazuje jen uživatelům s administrátorským právem na tento modul. Nahoře uprostřed je umístěna komponenta třídy *TMemoryDBGrid*. Jako svůj zdroj má komponentu třídy *TMemoryDS*, která je naplněna daty. Jedná se o statistiku rizik za celé VUT. Jde o seznam hodnocených rizik. U každého rizika je zde zobrazen kód a název rizikového procesu ke kterému se riziko vztahuje, kód a název samotného rizika, počet součástí VUT, které riziko hodnotily a celkový stupeň závažnosti rizika. Ten se vypočítá tak, že se sečtou stupně významnosti rizika na jednotlivých součástech VUT, které riziko v tomto pololetí hodnotily a tento součet je pak vydělen součtem vah, které se vztahují k jednotlivým hodnocením rizika na jednotlivých součástech VUT. To je pořád jen mezivýsledek, který je ale v Gridu také viditelný. Tento mezivýsledek se ještě vynásobí počtem součástí VUT, které riziko hodnotily a vyjde nám celkový stupeň významnosti rizika za celé VUT. Tento výpočet celkového stupně závažnosti rizika samozřejmě nemusí být ten poslední. Dále se uvažuje o objektivnějších postupech pro jeho výpočet ale momentálně se zdá zatím tento jako nejlepší. Ve spodní části se ještě nachází komponenta třídy *TCoolControlDetail*. V její hlavičce se zobrazují informace o riziku zvoleném v Gridu výše. Obsahuje jen jednu záložku s názvem „Detail rizika“. Ta obsahuje ve své pravé části komponentu třídy *TSMDGrid* ve které jsou zobrazeny detaily ohledně hodnocení rizika vybraného výše na jednotlivých součástech VUT v daném pololetí. V levé části je graf tvořený komponentou třídy *TChart*. Jedná se o sloupcový graf. Výška sloupce záleží na stupni

závažnosti rizika na dané součásti VUT. Pod každým sloupcem v grafu je uvedena součást, ke které se sloupec značící stupeň významnosti rizika na dané součásti vztahuje. Výslednou podobu této záložky můžete vidět na Obr.6.

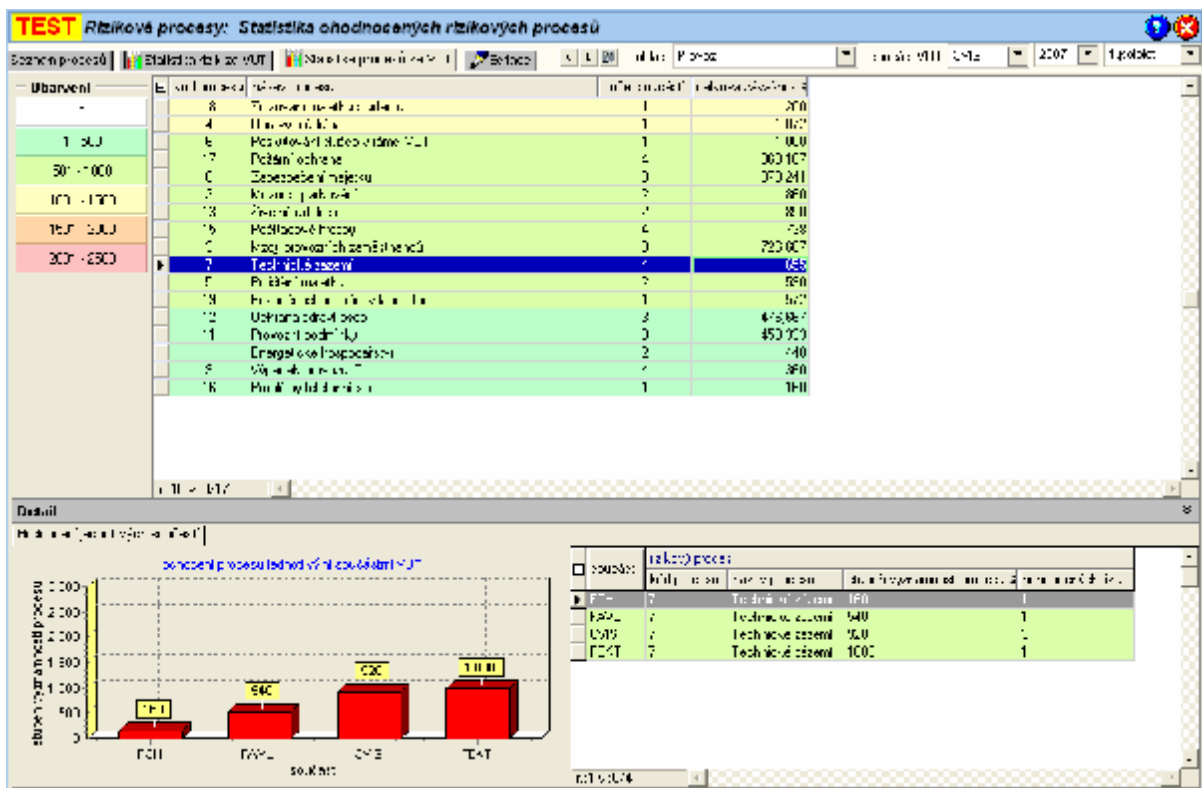


Obr. 6 Záložka se statistikou rizik za celé VUT

5.3.8 Záložka Statistika procesů

K této záložce mají přístup jen uživatelé s administrátorským oprávnění. Nahoře uprostřed je umístěna komponenta třídy *TMemoryDBGrid*, která má jako zdroj dat zvolenou komponentu třídy *TMemoryDS*. Zde je zobrazena statistika rizikových procesů za celé VUT. U každého rizikového procesu je v tomto Gridu uveden kód rizikového procesu, název rizikového procesu, počet součástí VUT, které nějaké riziko z tohoto rizikového procesu hodnotily a celkový stupeň závažnosti rizikového procesu. Celkový stupeň významnosti každého rizikového procesu se vypočítá tak, že se nejdříve vypočítá stupeň významnosti rizikového procesu pro jednotlivé součásti VUT, které alespoň jedno z rizik, které se k tomuto rizikovému procesu vztahovaly, hodnotily. Poté se vypočítá průměr z těchto jednotlivých stupňů významnosti rizikových procesů a nakonec se tato hodnota ještě vynásobí počtem součástí VUT, které v tomto pololetí alespoň jedno z rizik spadajících pod tento rizikový proces hodnotily. Ve spodní části je umístěna komponenta třídy *TCoolControlDetail*. V její hlavičce se zobrazují informace o rizikovém procesu vybraném v Gridu výše. Je zde umístěna jedna záložka s názvem „Detail rizikového procesu“. V této záložce je umístěna na levé straně komponenta

třídě *TSMDGrid*. A obsahuje informace o hodnocených rizikových procesech za jednotlivé součásti v rámci zvoleného pololetí. U každého rizikového procesu je zde uveden jeho kód, název, součást VUT, ke které se toto konkrétní hodnocení vztahuje, a stupeň významnosti tohoto rizikového procesu vztahující se k dané součásti VUT. V levé části se pak nachází graf, který je tvořen komponentou třídy *TChart*. Jde o sloupcový graf a výška sloupce je závislá na stupni významnosti rizikového procesu v rámci dané součásti VUT. Pro každou součást které bylo v zvoleném pololetí hodnoceno alespoň jedno riziko spadající pod tento rizikový proces je zde jeden sloupec. Od každým sloupcem je pak název součásti VUT, ke které se tato hodnota stupně významnosti rizikového procesu vztahuje. Výslednou podobu této záložky můžete vidět na Obr.7.



Obr. 7 Záložka se statistikou rizikových procesů za celé VUT

5.4 Tiskové sestavy

Všechny tiskové sestavy jsem sestavoval v programu Rave Designer. Po jeho instalaci jej lze spouštět přímo z Delphi. Umožňuje jednoduše navrhnout jak bude výsledná tisková sestava vypadat.

5.4.1 Tisková sestava pro seznam rizik

K této tiskové sestavě budou mít přístup všichni uživatelé IS Apollo, kteří budou mít právo čtení k tomuto modulu. Programu Rave Designer jsem nejdříve vytvořil Report. U tohoto reportu jsem nastavil velikost stránky na formát A4. nazval jsem tento report (tiskovou sestavu) „Seznam rizikových procesů a rizik“. Pak jsem na tuto stránku umístil hlavičku typu *DataBand*. Nastavil jsem zdroj dat na *RVCustomConenctionHlavicka*. Tato komponenta je nadefinována v mém modulu a při požadavku na tisk této sestavy je naplněna daty podle jednotlivých událostí, které jsou u této komponenty nadefinovány. Ve vrchní části hlavičky jsem vložil textové pole s tučným textem „Seznam rizikových procesů a rizik“. Pod ním jsem na levé straně vložil komponentu typu *DataMemo*, u které jsem jako zdroj dat zadal text „VUT v Brně,“ + fakulta. Fakulta je jedna položka dat která je předávána právě přes komponentu *RVCustomConnectionHlavicka* a obsahuje název zvolené součásti VUT v modulu. Na pravou stranu jsem umístil komponenty typu text obsahující popisy: rok, pololetí a oblast. Za každou z těchto komponent jsem umístil komponenty typu *DataText* a jako zdroj jejich dat jsem opět postupně zvolil: rok, pololetí, součást a oblast, což jsou opět data předávaná z komponenty *RVCustomConnectionHlavicka* obsahující informace o tom, k jakému roku, pololetí a oblasti se rizikové procesy jejich rizika vztahují. Pod tento *DataBand* s informacemi o hlavičce jsem vložil Band, u kterého jsem nastavil jako jeho styl ba Body Leader. To znamená že bude vytvořen na každé straně tiskové sestavy a bude tvořit hlavičku pro data v ní obsažená. Zde jsem pomocí komponent typu text popsal obsah jednotlivých sloupců tiskové sestavy. Tisková sestava bude obsahovat popis rizikového procesu a pod ním popisy všech rizik k dané oblasti. Proto jsem zde vytvořil dva řádky a v jednom jsem do jednoho sloupce napsal „Kód p.“ značící kód procesu a do druhého „Název rizikového procesu“. Do druhého řádku jsem vložil do prvního sloupce „Kód r.“ značící kód rizika a do druhého „Název rizika“. Pod tento Band jsem vložil další *DataBand*, který bude vypisovat do sestavy jednotlivé rizikové procesy. Do něj jsem umístil dvě komponenty typu *DataText* a jako jejich zdroj jsem zadal „kod_procesu“ a „navez_procesu“. Ty představují data předávaná z komponenty *RVCustomConnectionProcesy*. Pod tento *DataBand* jsem umístil další *DataBand*, který bude vypisovat jednotlivé rizika. Ten obsahuje komponentu *DataText*, která má jak zdroj „kod_rizika“, a komponentu typu *DataMemo*, která má jako svůj zdroj „navez_rizika“. Data jsou předávány z komponenty *RvCustomConnectionRizika*. Aby mohla být vždy pod daným rizikovým procesem vypsána jednotlivá rizika, která se k němu vztahují, nastavil jsem u obou

DataBandů jako *DetailKey* a *MasterKey* *proces_id*. Díky této hodnotě, která se předává jak z *RVCustomConnnectionProcesy* tak z *RvCustomConnectionRizika* jsou záznamy seskupovány.

5.4.2 Tisková sestava pro hodnocená rizika

K této tiskové sestavě budou mít přístup všichni uživatelé IS Apollo, kteří budou mít právo čtení k tomuto modulu. Vytvořil jsem nový Report (tiskovou sestavu) programem Rave Designer a opět nastavil velikost papíru na A4. Nastavil jsem také orientaci stránky na šířku. Vytvořil jsem zde *DataBand* reprezentující hlavičku, která je stejná jako v sestavě se seznamem rizik až na to, že název této sestavy je „Sestava hodnocených rizik“. Pod tuto hlavičku jsem umístil Band a nastavil jeho styl na Body Leader. To značí že se bude generovat na každé stránce. Do něj jsem pomocí 9 komponent typu Text popsal obsah jednotlivých sloupců. V prvním sloupci je „Kód p.“ značící kód procesu, v druhém je „Název procesu“, ve třetím je „Kód r.“ značící kód rizika, ve čtvrtém je „Název rizika“ a v posledním je „Stupeň závažnosti“ značící stupeň významnosti rizika. Pod tímto *DataBandem* je umístěn další. Ten obsahuje tak jak bylo nadefinováno výše jednotlivé sloupce. Všechny komponenty se plní daty prostřednictvím komponenty *RvCustomConnectionPodleRizika*. Pro první je použita komponenta typu *DataText* a jako zdroj má „kod_rizika“. Druhá komponenta je typu *DataMemo* a jako zdroj má „navez_rizika“. Tu následuje v dalším sloupci komponenta typu *DataMemo*, která má jako zdroj „navez_rizika“. Poslední komponenta je typu *DataText* a jako zdroj má „dpxv“, což je stupeň závažnosti rizika. Rizika jsou zde seřazena od nejzávažnějších po ty nejméně závažná a vztahují se k danému roku, pololetí, součásti a oblasti

5.4.3 Tisková sestava pro hodnocené procesy

K této tiskové sestavě budou mít přístup všichni uživatelé IS Apollo, kteří budou mít právo čtení k tomuto modulu. Vytvořil jsem nový Report (tiskovou sestavu) programem Rave Designer a opět nastavil velikost papíru na A4. Nastavil jsem také orientaci stránky na výšku. Vytvořil jsem zde *DataBand* reprezentující hlavičku, která je stejná jako v sestavě pro hodnocená rizika až na to, že název této sestavy je „Sestava hodnocených procesů“. Pod něj jsem umístil další Band u kterého jsem nastavil jeho styl na Body Leader. Díky tomu se bude generovat na každé stránce výsledné sestavy. Do něj jsem umístil komponenty typu text pro popis jednotlivých sloupců. První s názvem „Kód p.“ značící kód procesu, druhou s názvem „Název procesu“ a poslední s názvem „Stupeň závažnosti“. Pod tento Band jsem vložil *DataBand*, který bude generovat jednotlivé řádky sestavy. Všechny komponenty se plní daty prostřednictvím komponenty *RvCustomConnectionPodleProcesu*. Do prvního sloupce jsem umístil komponentu typu *DataText*, která má jako zdroj „kod_procesu“. Druhá komponenta je typu *DataMemo* a jako zdroj má „navez_procesu“. Poslední komponenta je typu

DataText a jako zdroj má stupeň, který je plněn stupněm závažnosti daného rizikového procesu. . Rizikové procesy jsou zde seřazeny od nejzávažnějších po ty nejméně závažná a vztahují se k danému roku, pololetí, součásti a oblasti.

5.4.4 Tisková sestava pro statistiku hodnocených rizik za VUT

K této tiskové sestavě budou mít přístup všichni uživatelé IS Apollo, kteří budou mít administrátorské právo k tomuto modulu. Vytvořil jsem nový Report (tiskovou sestavu) programem Rave Designer a opět nastavil velikost papíru na A4. Nastavil jsem také orientaci stránky na šířku. Vytvořil jsem zde *DataBand* reprezentující hlavičku, která je stejná jako v sestavě pro hodnocená rizika až na to, že zde místo celého názvu fakulty je jen „VUT v Brně“ a název této sestavy je „Seznam hodnocených rizik za celé VUT“. Tato sestava je stejná jako sestava pro hodnocená rizika až na to, že jako zdroj dat je zde u komponent *RvCustomConnectionStatistikaRizik*. Posledním rozdílem je, že se zde u jednotlivých rizik v komponentě se zdrojem stupeň zobrazuje stupeň závažnosti rizika vypočtený za celé VUT. . Rizika jsou zde seřazeny od nejzávažnějších po ty nejméně závažná a vztahují se k danému roku, pololetí a oblasti.

5.4.5 Tisková sestava pro statistiku hodnocených rizikových procesů za VUT

K této tiskové sestavě budou mít přístup všichni uživatelé IS Apollo, kteří budou mít administrátorské právo k tomuto modulu. Vytvořil jsem nový Report (tiskovou sestavu) programem Rave Designer a opět nastavil velikost papíru na A4. Nastavil jsem také orientaci stránky na šířku. Vytvořil jsem zde *DataBand* reprezentující hlavičku, která je stejná jako v sestavě pro statistiku hodnocených rizik za VUT až na to, že název této sestavy je „Seznam hodnocených rizikových procesů za celé VUT“. Tato sestava je stejná jako sestava pro hodnocené rizikové procesy až na to, že jako zdroj dat je zde u komponent *RvCustomConnectionStatistikaProcesu*. Posledním rozdílem je, že se zde u jednotlivých rizikových procesů v komponentě se zdrojem stupeň zobrazuje stupeň závažnosti rizikového procesu vypočtený za celé VUT. . Rizikové procesy jsou zde seřazeny od nejzávažnějších po ty nejméně závažná a vztahují se k danému roku, pololetí a oblasti.

6 Uvedení modulu do provozu

6.1 Nasazení modulu

Během vývoje tohoto modulu jsem jednotlivé tabulky, které tento modul využívá nenechal vytvářet jen na instanci databáze s označením CISB, která slouží pro vývoj, ale i na všech ostatních instancích. Teď mi tedy již jen zbývalo naplnit tabulky sloužící jako číselníky daty z instance vývojové databáze. Co se týče jednotlivých práv ty jsem také vytvářel při vývoji modulu jak na vývojové instanci databáze tak na všech ostatních. Na instanci, kterou využívá IS Apollo (CDBX) a na instanci kterou využívá testovací IS Apollo (CISD) jsem typ všech práv i role nastavil na A (to znamená, že toto právo může přidělovat a odebírat jen malý okruh zaměstnanců vývojového oddělení CVIS). To jsem učinil z toho důvodu, že tento modul bude obsahovat choulostivá data a bude k němu mít přístup jen několik uživatelů a je nutné zaručit aby žádná součást VUT tyto práva a roli nemohla nikomu přidělit. Dalším krokem při nasazování modulu bylo nasazení SQL dotazů. Všechny SQL dotazy vztahující s k tomuto modulu jsem si vyexportoval z instance CISB do jediného souboru. Ten jsem pak otevřel v programu PL/SQL, který jsem připojoval k jednotlivým instancím databáze a spouštěl. Spuštěním se nasadili nejen SQL dotazy ale i nastavení práv, které je mohou spouštět. Pro nasazení modulu do IS Apollo bylo nutné zkopírovat zdrojové soubory modulu z mého počítače na server, na kterém se IS Apollo i jeho jednotlivé moduly kompilují. Na tomto serveru je umístěn i program Delphi s aktuálními verzemi balíčků komponent. V tomto Delphi jsem můj modul zkompileval a vytvořenou knihovnu exRizikoveProces.dll jsem zkomprimoval spolu se soubory rav obsahující tiskové sestavy do exRizikoveProces.dll.zip, které jsou umístěny na aktualizacím serveru.

6.2 Školení uživatelů

Před samotným uvedením modulu do provozu jsem provedl školení uživatelů, kteří budou s tímto modulem pracovat. Spolu s ing. Martou Válkovou, která mi poskytla základní požadavky na tento modul a která bude s tímto modulem pracovat zatím jako jediná s právem administrátora, jsme rezervovali učebnu a rozeslaly pozvánky na školení všem, kteří budou s tímto modulem pracovat. Před tímto školením jsem již samotný modul nasadil, ale práva jsem patřičným uživatelům nastavil jen na testovacím Informačním systému Apollo. Ing. Marta Válková se ujala první části prezentace ve které všem zúčastněným vysvětlila proč byl vlastně tento modul vytvořen a proč je důležité ho používat. Já jsem se v druhé části ujal školení samotné práce s tímto modulem. Každý zúčastněný již měl přiděleny práva na spuštění modulu v testovacím Apollu a tak jsem nechal každého uživatele do Apolla přihlásit. Já jsem měl můj počítač napojen na projektor, podle kterého mohli všichni následovat moje kroky. Všem jsem tedy předvedl spuštění modulu, v jednotlivých záložkách jsem jim

popsal všechny komponenty které na nich byly zobrazeny, k čemu slouží a jak je používat, ukázal jsem jim jak je možno ohodnotit jednotlivá rizika, jak si lze vyfiltrovat rizika podle jednotlivých vlastností a jaké hodnoty mají vliv na celkový stupeň významnosti rizika. Na konci této prezentace jsem odpovídal na jednotlivé dotazy a řešil různé výtky vůči tomuto programu a ukázal všem zúčastněným, kde mohou nalézt nápovědu k tomuto modulu.

7 Závěr

7.1 Zhodnocení dosažených výsledků

Díky podrobné analýze, díky které jsem zjistil všechny požadavky na hodnocení rizik, jsem se mohl pustit do návrhu. V analýze jsem narazil na několik problémů, ale díky konzultaci a několika návrhům jak je vyřešit, jsme vždy došli k řešení akceptovatelnému na obou stranách. Některé věci nebyly v době analýzy jasné, ale to díky tomu, že je těžké, jak se bude do budoucna tento modul dále vyvíjet. Z toho důvodu jsem při návrhu dbal na to, aby databázové schéma bylo co nejvíce otevřené možným změnám, co se týče hodnocení jednotlivých rizik. Modul je již nasazen v ostrém provozu. Z hodnocení jednotlivých součástí VUT již byly použity jednotlivé statistiky pro zobrazení nejzávažnějších rizik a rizikových procesů.

7.2 Další vývoj

Ohledně dalšího vývoje tohoto modulu je zde několik věcí, které by mohlo být v budoucnu potřeba změnit. Například hodnocení všech faktorů má na váhu a tudíž i na stupeň závažnosti rizika stejný vliv. V budoucnu by mohla být potřeba zavést různé podíly vlivu jednotlivých faktorů na výpočet celkové váhy faktorů, která má vliv na celkový stupeň závažnosti rizika. V současné době jsou faktory dostatečně společné pro všechny oblasti a navíc má uživatel možnost jakýkoli faktor nehodnotit pokud se k danému riziku nehodí. S rozrůstajícím počtem rizik a rizikových oblastí by mohla v budoucnu být potřeba definovat faktory pro jednotlivé oblasti. Poslední otázkou vývoje tohoto modulu by mohl být způsob výpočtu celkové závažnosti rizika a rizikových procesů za celé VUT. Například když je hodnoceno riziko jen na jedné fakultě, má v této statistice stejnou hodnotu stupně závažnosti jako je jeho stupeň závažnosti pro danou fakultu. Pokud ale je např. riziko ohodnoceno na třech fakultách stejnou hodnotou, pak celkový stupeň závažnosti rizika je v statistice za celé VUT trojnásobek této hodnoty. Zde by teoreticky možná mohl být zvolen jiný vzorec výpočtu této hodnoty.

7.3 Vlastní přínos

Zjistil jsem jak je důležitá důkladná analýza před samotným návrhem a získal jsem zkušenosti při návrhu aplikace, za které vděčím Ing. Rudolfu Musilovi. Také jsem získal mnoho zkušeností při tvorbě SQL dotazů a výhody použití různých funkcí v SQL.



8 Literatura

- [1] Lacko, L. *Oracle - Správa, programování a použití databázového systému*, Computer Press, Praha, 2002, ISBN 80-7226-699-3
- [2] Loney, K., Theriault, M. *Mistrovství v Oracle - Kompletní průvodce tvorbou, správou a údržbou databází*, Computer Press, Praha, 2002, ISBN 80-7226-635-7
- [3] Kadlec, V. *Učíme se programovat v Delphi a jazyce Object Pascal*, Computer Press, Praha, 2001, ISBN 80-7226-245-9

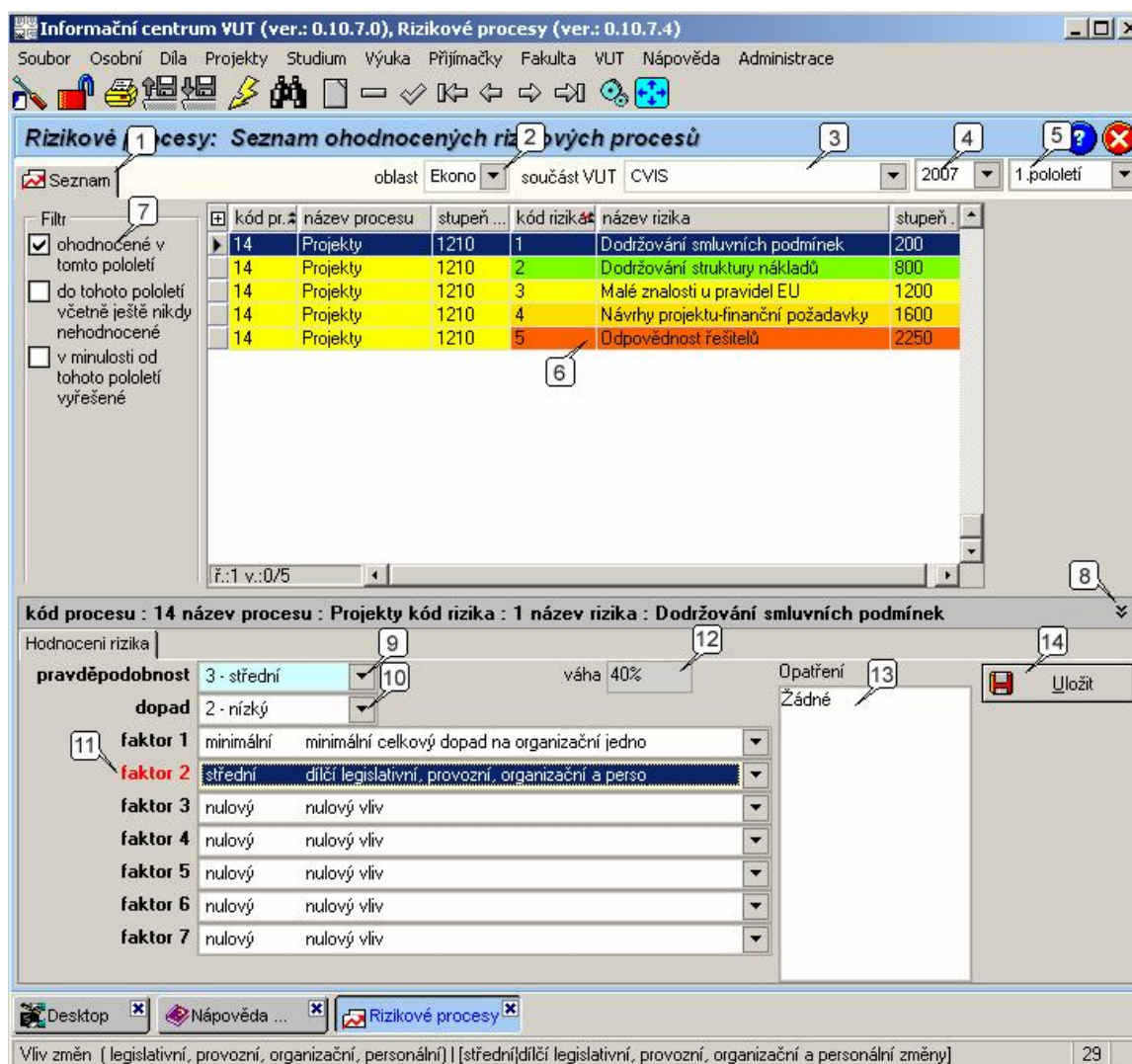
9 Seznam příloh

9.1 Uživatelská příručka

9.1.1 Spuštění modulu Rizikové

Modul **Rizikové procesy** se spouští z **Hlavního menu**  **VUT**  **Rizikové procesy**. Následně se vám zobrazí přehled rizikových procesů (viz. obr. 8). Procesy se zobrazují za oblast, která je právě zvolena.

9.1.2 Popis prostředí a ovládacích prvků modulu Rizikové procesy



Obr. 8 Přehled rizikových procesů

9.1.2.1 Rozbalovací seznamy (2,3,4,5)

V horní části modulu se nachází rozbalovací seznamy, které slouží k zobrazení požadované skupiny "rizikových procesů". Po klepnutí myší na šipku na pravé straně rozbalovacího seznamu se vám zobrazí nabídka, ve které lze provést výběr. Skupinu "rizikových procesů" volíte podle:

- **oblasti (2)** rizikových procesů - **D** - Dotace, rozpočet; **E** - Ekonomika, účetnictví; **P** - provoz; **R** - řízení; **N** - nedotační činnosti
- **součástí VUT (2)** - nabídka součástí VUT, ke kterým máte přiděleno oprávnění pro zobrazování a editaci "rizikových procesů"

- **kalendářního roku (4) / pololetí (5)**, ke kterému se vztahuje hodnocení "**rizikových procesů**" (1. pololetí - 1.1. do 30.6.; 2. pololetí - 1.7. do 31.12.)

9.1.2.2 Rychlé filtry (7)

Umožňují jednoduše zobrazovat "**rizikové procesy**" v seznamu "**Rizikové procesy**"(6) podle zvolených kritérií. (Filtry se vztahují ke kalendářnímu roku a pololetí, které je vybráno v "**Rozbalovacích seznamech**" -4,5). Modul obsahuje následující rychlé filtry:

- **ohodnocené v tomto pololetí** - v seznamu se zobrazí "**rizikové procesy**" ohodnocené ve zvoleném období
- **do tohoto pololetí včetně ještě nikdy neohodnocené** - budou zobrazeny "**rizikové procesy**" ohodnocené před zvoleným obdobím a procesy nikdy neohodnocené
- **v minulosti od tohoto pololetí vyřešené** - v seznamu budou zobrazeny již vyřešené "**rizikové procesy**", a to i neplatné (tj. procesy jejichž platnost již skončila)

Pro rychlejší práci s filtry můžete využít "**Místní nabídku**", která se vám zobrazí po klepnutí pravým tlačítkem myši na některý z filtrů. Více o místní nabídce filtrů naleznete [zde](#).

9.1.2.3 Seznam Rizikové procesy (6)

Tento seznam obsahuje přehled rizikových procesů dle vybraných kritérií. Zobrazují se zde procesy pouze pro jednu zvolenou oblast. Standardně se v seznamu nachází následující informace k jednotlivým "**rizikovým procesům**":

- **kód procesu** - číselné označení procesu
- **název procesu**
- **stupeň významnosti procesu** - jedná se o zprůměrovaný stupeň významnosti všech rizik daného procesu
- **kód rizika** - číselné označení rizika
- **název rizika**
- **stupeň významnosti rizika** - stupeň významnosti rizika získáme součinem **váhy** hodnocených faktorů, **míry pravděpodobnosti** a **míry dopadu**. Pro větší přehlednost jsou jednotlivé stupně významnosti rizika v seznamu barevně odlišeny. Barvy odpovídají níže uvedenému ohodnocení:

	nehodnoceno
0-500	minimální
500 - 1000	nízký
1000 - 1500	střední
1500 - 2000	vysoký
2000 - 2500	velmi vysoký

V seznamu si můžete nechat zobrazit podle potřeby sloupce poskytující další informace. Pokud klepnete pravým tlačítkem myši na tlačítko v levém horním rohu seznamu ☒, zobrazí se vám místní nabídka. V této nabídce zatržením / odtržením příslušného zatrhávacího pole provádíte volbu sloupců, které potřebujete zobrazit / skrýt.

Pro rychlejší vyhledání požadovaných rizikových procesů lze využít "**Filtrování**".

Podrobnější informace o práci se seznamem naleznete [zde](#).

V případě, že potřebujete do seznamu "**Rizikové procesy**" přidat nový "**rizikový proces**", zašlete svoji žádost Ing. Válkové (valkova@ro.vutbr.cz).

9.1.2.4 Detail rizikového procesu

Ve spodní části modulu se zobrazuje "**Detail hodnocení rizika**" (obr. 9). Jsou zde uvedeny informace, které se vztahují k riziku procesu zvolenému v seznamu "**Rizikové procesy**". Současně tato část modulu slouží k ohodnocení daného rizika. Tuto část modulu si můžete zobrazit / skrýt pomocí tlačítka ☒(8). Mezi položky, které zde hodnotíte patří následující:

- **pravděpodobnost** (9) - míra pravděpodobnosti vzniku rizika
- **dopad** (10) - míra dopadu rizika
- **faktory 1 - 7** (11) - v modulu je možné hodnotit následující váhové faktory:

faktor 1 - Celková závažnost rizika a finanční dopad
faktor 2 - Vliv změn (legislativní, provozní, organizační, personální)
faktor 3 - Složitost operací a postupů
faktor 4 - Komplexnost interních procedur a formalizace postupů (dostupnost písemných pracovních postupů)
faktor 5 - Stupeň automatizace a použití IT, systém informačních toků

faktor 6 - Přiměřenost a transparentnost VKS (finanční, řídicí kontrola, kontrolní mechanismy, úroveň přijatých opatření)

faktor 7 - Potřeba dohledu pro eliminaci rizika (čas, lidské zdroje, změna technologie, zavedených postupů atd.)



Faktory jsou identické pro všechny oblasti a procesy.

Pokud najedete kurzorem na textové pole u některého z faktorů, zobrazí se vám v místní nápovědě (15) a ve stavovém řádku (16) popis jednotlivých faktorů.

- **váha** (12) - položka "**váha**" se vypočítává automaticky z hodnocení jednotlivých faktorů
- **opatření** (14) - jedná se o opatření, které je nutné zavést nebo bylo zavedeno pro minimalizaci vzniku a dopadů rizikových procesů.

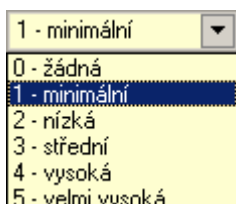
Obr. 9 Detail rizika procesu

9.1.2.5 Hodnocení rizika

1. Otevřete si modul "**Rizikové procesy**" (Obr. 8). Modul se spouští z **Hlavního menu**  **VUT**  **Rizikové procesy**.
2. V rozbalovacích seznamech si vyberete "**Oblast**" a "**Součást**", pro kterou budete rizika hodnotit. Dále zvolíte aktuální "**Rok**" a "**Pololetí**". (Rizika je možné hodnotit pouze pro aktuální rok a pololetí).
3. V "**Rychlých filtrech**" (7) si zaškrtnete políčka podle toho, které rizika chcete mít v seznamu zobrazena
4. V seznamu "**Rizikové procesy**" (6) si vyberete "rizikový proces", který má být hodnocen. Pro rychlejší vyhledání "rizik. procesu" můžete využít "[Filtry seznamu](#)".

5. V "**Detailu rizikového procesu**" provedete ohodnocení rizika v záložce následujícím způsobem:

- u položky "**pravděpodobnost**" vyberete ze seznamu míru pravděpodobnosti vzniku rizika (obr. 10)
- u položky "**dopad**" vyberete ze seznamu míru dopadu rizika (obr. 10)



Obr. 10 Míra pravděpodobnosti a dopadu rizika

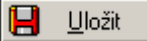
Pokud u položky "**pravděpodobnost**" nebo "**dopad**" zvolíte míru rizika "**0 - žádná**" (tzn. riziko bylo vyřešeno), nebude možné hodnotit jednotlivé "**faktory rizika**" a položka "**váha**" bude vždy "**0%**". V případě, že se součin "**pravděpodobnosti**" a "**dopadu**" nerovná "**0**" je nutné ohodnotit alespoň jeden faktor.

- v položkách "**faktor 1 - 7**" hodnotíte jednotlivé faktory rizika. Hodnotící stupnice je shodná, jako u položek "**pravděpodobnost**" a "**dopad**". U každého faktoru jsou stupně hodnocení blíže charakterizovány např. **faktor 1 - Celková závažnost rizika a finanční dopad** (obr. 11).

nulový	nulový vliv
minimální	minimální celkový dopad na organizační jednotku a finanční dopad 0 Kč
nízký	nízký dopad na organizační jednotku a finanční dopad v tis. Kč
střední	střední dopad na organizační jednotku i na VUT celkem, finanční dopad v desítkách tisíc Kč
vysoký	vysoký dopad na organizační jednotku a současně střední dopad na VUT celkem, finanční dopad ve statisících Kč
velmi vysoký	velmi vysoký dopad na organizační jednotku i VUT celkem, finanční dopad v milionech Kč




Obr. 11 Hodnocení faktorů

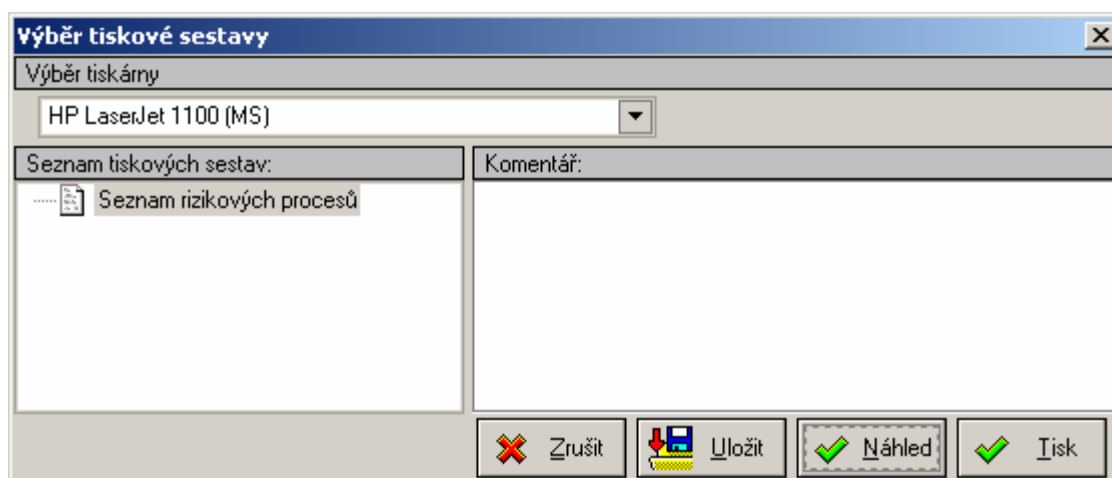
- položka "**váha**" se vypočítává automaticky. (Hodnocení jednotlivých faktorů je sečteno a součet je vydělen počtem hodnocených faktorů). Do výpočtu "**váhy**" nejsou zahrnuty faktory hodnocené "**nulově**".
- poslední zadávanou položkou je "**Opatření**". Tato položka je povinná a musí být vyplněna při každém hodnocení a změně. V případě, že jste zvolili u položky "**pravděpodobnost**" nebo "**dopad**" míru rizika "**0 - žádná**", uvádíte zde informaci o opatření, které bylo použito. (Jestliže nebylo použito žádné opatření, stačí uvést údaj "**žádné**").

6. Hodnocení "**rizikového procesu**" uložíte stisknutím tlačítka 

9.1.3 Tisk

Modul vám umožňuje vytisknout si přehled hodnocených rizikových procesů za dané období. Tisk lze provést následujícím způsobem:

1. Otevřete si modul "**Rizikové procesy**" (Obr. 8). Modul se spouští z **Hlavního menu**  **VUT**  **Rizikové procesy**.
2. V rozbalovacích seznamech si vyberete **oblast**, **součást** a **období**, za které chcete přehled hodnocených "**rizikových procesů**" tisknout.
3. Stisknete tlačítko "**Soubor**" a příkaz "**Tisk**" (můžete také stisknout tlačítko ).
4. Zobrazí se vám dialogové okno "**Výběr tiskové sestavy**"(obr. 12).
5. Z tohoto dialogového okna je možné zobrazit před tiskem náhled tiskové sestavy (obr. 13) pomocí tlačítka "**Náhled**", nebo přímo zaslat sestavu na tiskárnu stisknutím tlačítka "**Tisk**".






Obr. 12 Výběr tiskové sestavy



Seznam hodnocených rizikových procesů		
VUT v Brně, Centrum výpočetních a inform		Datum: 5.3.2007
		Rok: 2007
		Pololetí: 1.pololetí
Kód	Název rizikového procesu	Stupeň závažnosti
14	Projekty	1210
Kód	Název rizika	Stupeň závažnosti
1	Dodržování smluvních podmínek	200
Kód	Název rizika	Stupeň závažnosti
2	Dodržování struktury nákladů	800
Kód	Název rizika	Stupeň závažnosti
3	Malé znalosti u pravidel EU	1200
Kód	Název rizika	Stupeň závažnosti
4	Návrhy projektu-finanční požadavky	1600
Kód	Název rizika	Stupeň závažnosti
5	Odpovědnost řešitelů	2250

Obr. 13 Náhled seznamu hodnocených rizikových procesů


9.1.4 Filtrování

Pro rychlejší vyhledání určitého rizikového procesu v seznamu můžete využít "**Filtry seznamu**".

Nejdříve klepnete levým tlačítkem myši na tlačítko  v levém horním rohu daného seznamu, poté se vám zobrazí pod záhlavím sloupců nový prázdný řádek určený k filtrování - viz. obr 14. (Pokud není zobrazen řádek pro filtrování, obsahuje tlačítko v levém horním rohu "", jinak obsahuje "").


	kód	název procesu	stu...	kód	název rizika	stupeň
						
	1	Doplňková činnost		1	Omezující podmínky pro její vykonávání	
	2	Fondy, tvorba a čer		1	Stanovení pravidel pro tvorbu a čerpání fondy	
	3	Hospodaření s pří		1	nedodržení struktury čerpání projektu	
	3	Hospodaření s pří		2	Nedodržování pravidel čerpání	

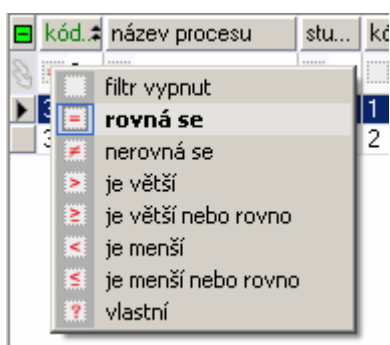
Obr. 14 Otevření lišty pro filtrování

Pokud chcete, aby seznam obsahoval např. jen záznamy s procesy "**Hospodaření s příspěvky a dotacemi**", napíšete do řádku pro filtrování pod záhlaví sloupce „**kód procesu**“ číslo „**3**“ a stisknete klávesu **ENTER**. V seznamu se zobrazí jen záznamy s kódem procesu "**3**" - viz. obr.15. Pokud chcete skrýt řádek pro zadávání filtrů, klepněte levým tlačítkem myši do levého horního rohu tabulky na tlačítko . Všechny zadané filtry budou nadále v seznamu aktivní.

kód	název procesu	stu...	kód !*	název rizika	stupeň
3	Hospodaření s příst		1	nedodržení struktury čerpání projektu	
3	Hospodaření s příst		2	Nedodržování pravidel čerpání	

Obr. 15 Filtrování podle nějaké hodnoty v daném sloupci

Filtry se dají zrušit v místní nabídce filtrů. Klepnete pravým tlačítkem myši na filtr, který chcete zrušit . Zobrazí se vám místní nabídka viz. obr. 16, ve které potvrdíte výběr "**filtr vypnout**". Další možností je označit filtrovaný text v daném sloupci a 2x stisknout tlačítko "**DELETE**".



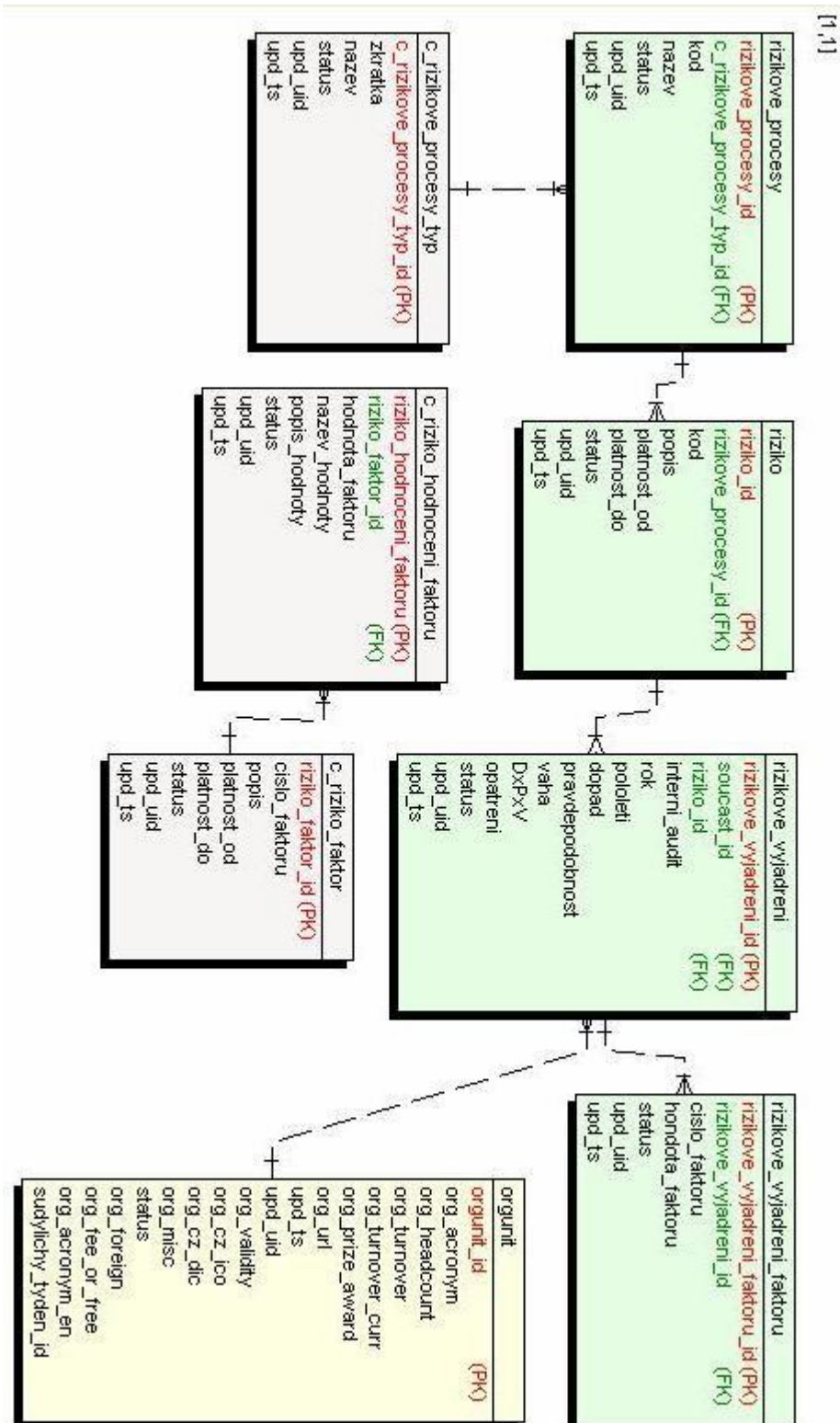
Obr. 16 Výběr pravidla pro filtrování

Více informací o filtrování v seznamech naleznete [zde](#).

9.2 Hodnocení rizik v sešitu aplikace Excel

Řízení						
Kód procesu	Název procesu	Kód rizika	Popis rizika	Dopad	Pravděpod.	Stupeň významnosti procesu významnosti rizika R=D×P
				D	P	
						Stupeň významnosti rizika R=D×P
R1	Určení kompetencí vedoucích zaměstnanců					
		R0101	schvalování operací bez písemného pověření nezprávné rozhodnutí vedení školy s dopadem na CVIS	4,00	2,00	8,00
				3	2	6
		R0102		5	2	10
R2	nepravidelnost operací					
		R0201	četnost nestandardních operací	2,50	3,50	8,75
		R0202	nedůsledná kontrola plnění úkolů	1	5	5
		R0203	nepřímé zatížení zaměstnanců	3	2	6
		R0204	nesystematické plnění úkolů zaměstnanci	3	5	15
				3	2	6
R3	Měnění se ekonomické podmínky organizace					
		R0301	zesilující konkurence v hospodářské činnosti	3,33	2,67	8,89
		R0302	snižování dotačních příjmů	2	3	6
		R0303	snižování příjmů z dotační a hospodářské činnosti	5	2	10
				3	3	9

9.3 Schéma databáze pro rizikové procesy



9.4 Rizikové faktory a jejich hodnoty

Poř. Č.	faktor		váha faktoru					
		nulová (0)	minimální (1)	nizká (2)	střední (3)	vyšoká (4)	velmi vysoká (5)	
1.	Celková závaznost rizika a finanční dopad	nulový vliv	minimální celkový dopad na organizační jednotku a finanční dopad 0 Kč	nizký dopad na organizační jednotku a finanční dopad v tis. Kč	střední dopad na organizační jednotku i na VUT celkem, finanční dopad v desítkách tisíc Kč	vyšoký dopad na organizační jednotku a současně střední dopad na VUT celkem, finanční dopad ve státech Kč	velmi vysoký dopad na organizační jednotku i VUT celkem, finanční dopad v milionech Kč	
2.	Vliv změn (legislativní, provozní, organizační, personální)	nulový vliv	bez legislativních, provozních, organizačních a personálních změn	díleč legislativní změny, provozní, organizační a personální stabilita	díleč legislativní, provozní, organizační a personální změny	nekompetentnost, neodbornost s nestálost personálního vybavení, četnost legislativních změn, organizační komplikovanost	velmi složité operace a postupy s velmi vysokou mírou administrativní náročnosti	
3.	Složitosť operací a postupů	nulový vliv	jednoduché, pravidelně se opakující operace a postupy, programová a projektová pravidla s velmi nízkou mírou administrativní náročnosti	jednoduché operace a postupy, programová a projektová pravidla s minimální mírou administrativní náročnosti	složitéjší operace a postupy, programová a projektová pravidla s střední mírou administrativní náročnosti	nejednoznačné operace a postupy s rizikem nejednoho a nesprávného výkladu programových a projektových pravidel	financování s vysokou mírou administrativní náročnosti	neexistence pracovních postupů
4.	Komplexnosť interních procedur a formalizace postupů (dostupnosť písemných pracovních postupů)	nulový vliv	kompletní pracovní postupy včetně aktualizace změn	opřednělý výskyt absence pracovních postupů nebo jejich aktualizace - minimální požadavky	absence pracovních postupů - méně rizikové oblasti	absence pracovních postupů - rizikové oblasti	neexistence pracovních postupů	
5.	Stupeň automatizace a použití IT, systém informatických toků	nulový vliv	vyšoký stupeň automatizovaných operací v IS VUT, elektronický přenos dat	střední stupeň automatizovaných operací v IS VUT, elektronický přenos dat	díleč stupeň automatizace, tj. v IS organizační jednotky s el. přenosem informací do IS VUT	díleč stupeň automatizace, tj. v IS organizační jednotky bez možnosti přenosu informací do IS VUT	omezená finanční a líniová kontrola a kontrolní mechanismy, slabé pokrytí	absence finanční, líniové kontroly a kontrolních mechanismů
6.	Priměřenosť a transparentnosť VKS (finanční, řídicí kontrola, kontrolní mechanismy, úroveň přijatých opatření)	nulový vliv	vyšoký stupeň automatizovaných kontrol (vč. finančních kontrol) s max. transparentnosť	finanční a líniová kontrola a kontrolní mechanismy pokrývají veškeré činnosť, procesy a programy	adekvátní finanční a líniová kontrola a kontrolní mechanismy pokrývají veškeré činnosť, operace, procesy a programy			
7.	Potreba dohledu pro eliminaci rizika (čas, lidské zdroje, změna technologie, zavedených postupů atd.)	nulový vliv	bez dohledu	dohled s minimální spotřebou zdrojů, minimální změny pouze v organizační jednotce	dohled s minimální spotřebou zdrojů, minimální změny v organizační jednotce i na úrovni VUT	eliminace rizika se silným dohledem, střední spotřeba zdrojů, změny i na úrovni VUT	eliminace rizika se silným dohledem, spotřeba zdrojů přesahuje dopad způsobený rizikem, změny i na úrovni VUT	

9.5 Tiskové sestavy

9.5.1 Tisková sestava pro seznam rizikových procesu a rizik

Seznam rizikových procesů a jejich rizik		
VUT v Brně, Centrum výpočetních a informačních služeb		Datum: 16.5.2007 Rok: 2007 Pololetí: 1.pololetí Oblast: Provoz
Kód p.	Kód r.	Název rizikového procesu Název rizika
1		Energetické hospodářství
	1	Hospodaření s energiemi
2		Možnost parkování
	1	Mínimální možnost parkování
3		Mzdy provozních zaměstnanců
	1	Nízké mzdy provozních zaměstnanců
4		Opravy a údržba
	1	Havárie
5		Pojištění majetku
	1	Nemožnost připojištění studentů
6		Poskytování služeb v rámci VUT
	1	Náročnosti studijních programů na různých fakultách
7		Technické zázemí
	1	Rozptýlená dislokace fakulty
	2	Výskyt škůdců ve skladech
	3	Poruchy chladičích zařízení
8		Zabezpečení majetku
	1	Dodržování provozního řádu
	2	Nebezpečí odcizení majetku
	3	Nedostatečné elektronické zabezpečení majetku
	4	Nezabezpečení areálu souvislým plotem
	5	Pohyb osob bez omezení
	6	Krádež majetku zaměstnancem
	7	Krádež majetku cizí osobou
	8	Mnoho zaměstnanců manipulujících s hotovostními platbami
	9	Přístup do objektu únikovými východy
	10	Velký pohyb gastro nádob
	11	Škoda na majetku ve spol. prostorách
	12	Ubytování zahraničních studentů-nevymahatelnost ztrát
9		Výpadek provozu IT
	1	Obnova dat a provozu do 48 hodin
	2	Síhání procesu zálohování
	3	Data nelze obnovit
	4	Server nelze zprovoznit do 8 hodin
	5	Záložní server není možné spustit
	6	Výpadek všech serverů na víc jak 10 minut
	7	Výpadek všech webových serverů portálu
	8	Výpadek rozřazovacího serveru portálů
	9	Výpadek e-mailové pošty
	10	Výpadek konektivity serveru a uživatelů víc než 30 minut
	10	Přeseknutí optického kabelu
10		Nedostatečné technické zázemí
	1	Nedostatek pracovních stanic
	2	Nedostatek serverů
	3	Nedostatek aktivních prvků sítě
	4	Nedostatek školení
	5	Nedostatek spotřebního materiálu
	6	Nedostatek komunikační techniky

9.5.2 Tisková sestava s hodnocenými riziky

Seznam hodnocených rizik				
VUT v Brně, Centrum výpočetních a informačních služeb				
Datum: 16.5.2007				
Rok: 2007				
Poleť: 1.pololetí				
Oblast: Provoz				
Kód b.	Název procesu	Kód r.	Název rizika	Stupeň závažnosti
7	Technické zázemí	1	Rozptýlená dislokace fakulty	2000
7	Technické zázemí	2	Vyskyt škůdců ve skladech	720
3	Mzdy provozních zaměstnanců	1	Nizké mzdy provozních zaměstnanců	660
7	Technické zázemí	3	Poruchy chladičích zařízení	40

9.5.3 Tisková sestava s hodnocenými rizikovými procesy

Seznam hodnocených rizikových procesů		
VUT v Brně, Centrum výpočetních a informačních služeb		Datum: 16.5.2007
		Rok: 2007
		Pololetí: 1.pololetí
		Oblast: Provoz
Kód p.	Název procesu	Stupeň závažnosti
7	Technické zázemí	920,000
3	Mzdy provozních zaměstnanců	660,000

9.5.4 Tisková sestava pro statistiku hodnocených rizik za VUT

VUT v Brně			Seznam hodnocených rizik za celé VUT		Datum: 16.5.2007
					Rok: 2007
					Poleti: 1.pololetí
					Oblast: Provoz
Kód r.	Název procesu	Kód r.	Název rizika	Stupeň závažnosti	
8	Zabezpečení majetku	2	Nebezpečí ochrizení majetku	83,844	
7	Technické zázení	1	Rozřtylená dislokace fakulty	59,000	
17	Požární ochrana	3	Vandalismus na hasicích přístrojích	45,175	
15	Počítačové hrozby	1	Útok virů	45,012	
8	Zabezpečení majetku	3	Nedostatečné elektronické zabezpečení majetku	41,000	
2	Možnost parkování	1	Minimální možnost parkování	40,000	
3	Mzdy provozních zaměstnanců	1	Nizké mzdy provozních zaměstnanců	39,836	
17	Požární ochrana	1	Netfunkčnost únikových východů	32,222	
11	Provozní podnikny	3	Výpadek elektrické energie	27,200	
8	Zabezpečení majetku	11	Škoda na majetku ve spol. prostorách	25,846	
6	Poskytování služeb v rámci VUT	1	Náročnosti studijních programů na různých fakultách	25,000	
5	Pojštění majetku	1	Nemožnost pripoštění studentů	23,600	
8	Zabezpečení majetku	7	Krádež majetku cizí osobou	22,588	
12	Ochrana zdraví osob	5	Psychické přetižení, mentální vyhoření	20,364	
8	Zabezpečení majetku	1	Dodržování provozního řádu	20,174	
17	Požární ochrana	2	Netfunkčnost nouzového osvětlení	20,000	
8	Zabezpečení majetku	5	Pohyb osob bez omezení	16,909	
4	Opravy a údržba	1	Havárie	16,000	
19	Bezpečnost při práci v laboratorii	5	Rizika dopadu na životní prostředí	16,000	
18	Zcizování majetku studentů	1	Pokles zájmu o ubytování na kolejích	15,000	
13	Živelné události	2	Zatopení místnosti, budovy	15,000	
15	Počítačové hrozby	4	Útok z vnějšího prostředí	15,000	
1	Energetické hospodářství	1	Hospodáření s energiemi	14,867	
12	Ochrana zdraví osob	1	Úraz osoby elektrickým proudem	13,455	
7	Technické zázení	3	Jiný pracovní úraz	12,800	
9	Výpadek provozu IT	2	Výskyt škůdců ve skladech	12,000	
15	Počítačové hrozby	2	Slhání procesu zálohování	12,000	
15	Počítačové hrozby	3	Útok z vnějšího prostředí	12,000	
13	Živelné události	2	Útok trojského koně na IS	12,000	
9	Výpadek provozu IT	1	Požár v místnosti, budovy	10,000	
19	Bezpečnost při práci v laboratorii	3	Data nelze obnovit	9,231	
19	Bezpečnost při práci v laboratorii	3	Rizika skládování nebezpečných chemikálií	9,000	
19	Bezpečnost při práci v laboratorii	4	Rizika likvidace nebezpečných odpadů	9,000	
19	Bezpečnost při práci v laboratorii	1	Rizika při nakládání s chemikáliemi	9,000	
8	Zabezpečení majetku	9	Přístup do objektu únikovými východy	8,000	

9.5.5 Tisková sestava pro statistiku hodnocených rizikových procesů za celé VUT

Seznam hodnocených rizikových procesů za celé VUT		
VUT v Brně		Datum: 16.5.2007
		Rok: 2007
		Pololetí: 1.pololetí
		Oblast: Provoz
Kód p.	Název procesu	Stupeň závažnosti
18	Zcizování majetku studentů	1200,000
4	Opravy a údržba	1072,000
6	Poskytování služeb v rámci VUT	1000,000
17	Požární ochrana	969,167
8	Zabezpečení majetku	873,241
2	Možnost parkování	860,000
13	Živelné události	850,000
15	Počítačové hrozby	738,000
3	Mzdy provozních zaměstnanců	726,667
7	Technické zázemí	655,000
5	Pojištění majetku	590,000
19	Bezpečnost při práci v laboratoři	572,000
12	Ochrana zdraví osob	476,667
11	Provozní podmínky	453,333
1	Energetické hospodářství	440,000
9	Výpadek provozu IT	360,000
16	Problémy telefonní sítě	160,000