



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ  
ÚSTAV MANAGEMENTU

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUTE OF MANAGEMENT

# VYUŽITÍ UMĚLÉ INTELIGENCE PRO PODPORU ROZHODOVÁNÍ V PODNIKU

THE USE OF ARTIFICIAL INTELLIGENCE FOR THE DECISION MAKING IN THE FIRM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETR COUFAL

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. PETR DOSTÁL, CSc.

BRNO 2010

# ZADÁNÍ DIPLOMOVÉ PRÁCE

**Coufal Petr, Bc.**

---

Řízení a ekonomika podniku (6208T097)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává diplomovou práci s názvem:

**Využití umělé inteligence pro podporu rozhodování v podniku**

v anglickém jazyce:

**The Use of Artificial Intelligence for the Decision Making in the Firm**

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

- DOSTÁL, P. Pokročilé metody analýz a modelování v podnikatelství a veřejné správě. 1. vyd. Brno: CERM, s.r.o., 2008. 340s. ISBN 978-80-7204-605-8.
- DOSTÁL, P. Advanced Economic Analyses. 1. vyd. Brno: CERM, s.r.o., 2008, 80s. ISBN 978-80-214-3564-3.
- DOSTÁL, P., RAIS, K., SOJKA, Z. Pokročilé metody manažerského rozhodování. 1. vyd. Praha: Grada, 2005. 168s. ISBN 80-247-1338-1.
- MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. Umělá inteligence. 1. vyd. Praha: ACADEMIA, 2003, 1440s., ISBN 80-200-0502-1.
- ALIEV, A., ALIEV, R. Soft Computing and Its Applications. 1. vyd. World Scientific Pub. Ltd, UK 2002, 444s., ISBN 981-02-4700-1.

Vedoucí diplomové práce: doc. Ing. Petr Dostál, CSc.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2009/2010.

L.S.

---

PhDr. Martina Rašticová, Ph.D.  
Ředitel ústavu

---

doc. RNDr. Anna Putnová, Ph.D., MBA

V Brně, dne 09.05.2010

## **Anotace**

Diplomová práce se zabývá možností využití umělé inteligence pro podporu manažerského rozhodování v podniku. Práce obsahuje návrh aplikace fuzzy logiky jako nástroje pro hodnocení materiálových dodavatelů firmy a poskytovatele informací pro efektivnější řízení vztahů s dodavateli.

## **Annotation**

The Master's thesis deals with the topic of the use of artificial intelligence for managerial decision making in the firm. This thesis contains an application of fuzzy logic system for firm's supplier evaluation that provides informations for more efficient collaboration with suppliers.

## **Klíčová slova**

adaptivní neurofuzzy učení	adaptive neurofuzzy learning techniques
analýza	analysis
databáze	database
fuzzy logika	fuzzy logic
hodnocení	evaluation
rozhodování	decision making
umělá inteligence	artificial intelligence

## Bibliografická citace

COUFAL, P. *Využití umělé inteligence pro podporu rozhodování v podniku*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2010. 79 s. Vedoucí diplomové práce doc. Ing. Petr Dostál, CSc.

## **Čestné prohlášení**

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil autorská práva (ve smyslu zákona č. 121/200 Sb. O právu autorském a o právech souvisejících s právem autorským).

V Brně dne

.....

podpis

# Obsah

1	Úvod.....	9
2	Vymezení problému a cíle práce .....	10
3	Teoretická východiska práce .....	11
3.1	Definice umělé inteligence .....	11
3.2	Aplikační oblasti umělé inteligence.....	12
3.3	Rozhodování v podmínkách neurčitosti v datech.....	16
3.4	Fuzzy množiny a fuzzy logika.....	17
3.4.1	Aplikace fuzzy logiky.....	18
3.5	Využití fuzzy logiky pro rozhodovací úlohy .....	20
3.6	Matlab .....	21
3.6.1	Fuzzy Logic Toolbox.....	22
3.6.2	Database toolbox.....	27
4	Analýza současného stavu .....	29
4.1	Název, právní forma a sídlo společnosti.....	29
4.2	Předmět podnikání, výrobní sortiment.....	29
4.3	Organizační struktura společnosti.....	30
4.4	Současný vývoj hospodaření .....	31
4.5	Informační systém.....	31
4.5.1	SAP R/3 .....	31
4.5.2	Správa dat pro informační systém .....	32
4.6	Řízení nákupu .....	33
4.7	Řízení jakosti .....	33
4.8	Současný systém hodnocení dodavatelů.....	34
4.9	Závěry z analýzy .....	34
5	Vlastní řešení .....	35
5.1	Hodnocení dodavatelů .....	36
5.1.1	Cíle hodnocení .....	36
5.1.2	Stanovení kritérií a jejich vah .....	37
5.2	Příprava dat .....	38
5.3	Výpočet ukazatelů.....	41

5.3.1	Načtení dat z databáze .....	42
5.3.2	Přepočet ukazatelů Certifikát a Jakost .....	44
5.3.3	Výpočet ukazatele Cenová úroveň .....	45
5.3.4	Výpočet ukazatelů Cenové chování, Dodržování Termínů a Dodržování množství .....	47
5.3.5	Uložení výsledků analýzy do databáze .....	50
5.4	Aplikace fuzzy logiky .....	51
5.5	Aplikace v prostředí Visual Basic .NET .....	52
5.5.1	Vytvoření transformační matice .....	52
5.5.2	Výpočet hodnocení .....	53
5.6	Aplikace v Matlabu .....	55
5.6.1	Trénování modelu v ANFIS editoru .....	55
5.6.2	Model Hodnocení ceny .....	59
5.6.3	Model Hodnocení jakosti .....	62
5.6.4	Model Hodnocení dodání .....	65
5.6.5	Model Celkové hodnocení .....	67
5.7	Propojení fuzzy modelů .....	70
5.7.1	Načtení dat .....	70
5.7.2	Vyhodnocování dodavatelů .....	70
5.7.3	Grafické znázornění hodnocení .....	72
5.7.4	Uložení výstupů hodnocení .....	73
5.8	Závěry z hodnocení .....	73
6	Závěr .....	75
7	Seznam použité literatury .....	76
8	Seznam použitých zkratk a symbolů .....	77
9	Seznam tabulek .....	77
10	Seznam obrázků .....	78
11	Seznam programů .....	79
12	Seznam příloh .....	79



# 1 Úvod

S vývojem a rozmachem moderních informačních technologií se v oblasti podpory manažerského rozhodování začínají objevovat nové postupy a možnosti optimalizace a řízení. Oblast umělé inteligence je velice rozsáhlá a možnosti využití rychle rostou.

Manažerské rozhodování často probíhá za podmínek nejistoty a neurčitosti, kdy dané problémy nelze plně přesně definovat, vychází se z neúplných či nepřesných informací a vágních pojmů.

Umělá inteligence je označením pro moderní technologie jako jsou umělé neuronové sítě, fuzzy logika, genetické algoritmy či jejich kombinace. Možnosti uplatnění a podmínky použití se liší podle použitého prostředku, procesu, složitosti řešeného systému a definovaného problému.

Teorie fuzzy logiky má v oblasti manažerského rozhodování značné uplatnění. Hlavním zdrojem úspěšnosti je to, že fuzzy logika umožňuje zahrnout nepřesnost a poměrně jednoduchým způsobem pracovat s významy slov přirozeného jazyka.

Diplomová práce se zabývá využitím aplikovaných metod fuzzy logiky a adaptivního učení pro vytvoření systému hodnocení dodavatelů společnosti Minerva Boskovice, a.s.

Teoretická východiska práce tématicky rozvíjejí danou problematiku od obecných definic umělé inteligence přes možnosti aplikace některých metod po podrobnější rozbor teorie fuzzy množin a její aplikaci při tvorbě fuzzy inferenčních systémů. Součástí východisek je i rozbor možnosti praktického využití fuzzy logiky v prostředí programu Matlab.

Následující kapitola se zabývá charakteristikou společnosti, její současnou situací a identifikuje podmínky, které představují výchozí předpoklady pro vlastní řešení.

Vlastní řešení sestává v první fázi z navržení ukazatelů jako kritérií pro hodnocení, zdrojů relevantních dat a algoritmu výpočtu jednotlivých ukazatelů. Pro vytvoření fuzzy systému je využita metoda neurofuzzy inference pro učení modelu podle souboru vytvořených dat popisujících chování modelu.

## 2 Vymezení problému a cíle práce

Hlavním cílem diplomové práce je navržení aplikace využívající fuzzy logiku pro hodnocení výkonnosti materiálových dodavatelů firmy. Výstupem aplikace by měly být informace užitečné pro podporu rozhodování v oblasti řízení vztahů s dodavateli, a které by přesně vypovídaly o vhodnosti dodavatelů pro kvalitní a dlouhodobou spolupráci, případně signalizovaly pokles výkonnosti a vypověděly důvody, které k tomuto poklesu vedly.

Díličními cíly práce jsou:

- navržení ukazatelů a kritérií jako vstupů pro hodnocení
- výběr relevantních dat pro výpočet ukazatelů
- maximální automatizace výpočtu ukazatelů i samotného systému hodnocení
- využití adaptivních technik učení pro tvorbu fuzzy systému

Současný model pro hodnocení dodavatelů ve firmě sestává z poměrně jednoduché metody známkování vedené v Excelu, kde se dodavatelé rozdělují do skupin podle udělené celkové známky, která je tvořena známkou za jakost, známkou za cenu a známkou za dodávku. Jednotlivé známky se zadávají ručně dle podkladů nákupních referentů.

Navrhovaná aplikace by měla přinést výhody v podobě zpřesnění hodnocení a hlavně automatizace většiny činností související s výpočtem hodnocení.

## 3 Teoretická východiska práce

### 3.1 Definice umělé inteligence

Umělá inteligence jako vědní disciplína se formuje již od 50. let minulého století a je jedním z nejrychleji se vyvíjejících vědeckých a technických oborů. Zastřešuje několik poměrně různorodých oblastí jako kognitivní psychologie, neurologie, kybernetika, matematická logika, teorie rozhodování, informatika, teorie her, lingvistika atd. [7]

Dosud neexistuje definice, která by byla obecně přijímána. Mezi známější definice patří dle [7]:

*„Umělá inteligence je věda o vytváření strojů nebo systémů, které budou při řešení určitého úkolu užívat takového postupu, který – kdyby ho dělal člověk – bychom považovali za projev jeho inteligence.“*

(M. Minsky, 1967)

*„Umělá inteligence se zabývá tím, jak počítačově řešit úlohy, které dnes zatím zvládají lidé lépe.“*

(E. Rich, 1991)

*„Umělá inteligence je vlastnost člověkem uměle vytvořených systémů vyznačujících se schopností rozpoznávat předměty, jevy a situace, analyzovat vztahy mezi nimi a tak přijímat účelná rozhodnutí, za pomoci schopností předvídat důsledky těchto rozhodnutí a objevovat nové zákonitosti mezi různými modely nebo jejich skupinami.“*

(Z. Kotek, 1983)

Oblast umělé inteligence sdružuje teoretické postupy, metody a techniky, které lze využít k počítačovému řešení některých složitých úloh rozhodování, plánování, diagnostiky.

Ve výzkumu umělé inteligence existují dva směry. Jeden je inspirován novými biologickými a fyziologickými poznatky o fungování nervové soustavy (ten vedl ke konstrukci a rozvoji umělých neuronových sítí), druhý je založen na analýze způsobu lidského uvažování a chápe inteligenci jako schopnost reprezentace světa abstraktními pojmy a schopnost s nimi správně manipulovat [8].

### **3.2 Aplikační oblasti umělé inteligence**

Široké uplatnění v praxi našly především metody označující se souhrnně Soft Computing [8]. Patří sem např. umělé neuronové sítě, genetické algoritmy, systémy na zpracování vágnosti a nejistoty. Níže uvádím metody a techniky, které se úspěšně uplatňují v ekonomické praxi.

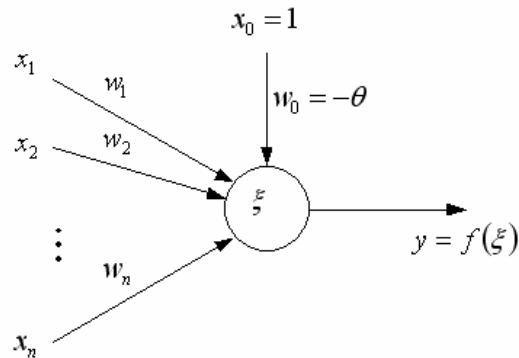
#### **Neuronové sítě**

Neuronové sítě jsou inspirovány poznatky o neuronech a nervových sítích živých organismů. Jejich hlavním účelem je klasifikace, jsou vhodné pro regresi a predikci časových řad. Mají schopnost extrahovat a reprezentovat závislosti v datech, které nejsou zřejmé, mohou řešit silně nelineární úlohy. Mají schopnost učit se a zevšeobecňovat [8].

Protože nelze znát detailně vnitřní strukturu neuronové sítě, jsou často označovány termínem „černá skříňka“. Na vnitřní strukturu sítě je kladeno pouze několik předpokladů, které umožní popsat chování funkcí provádějící transformaci vstup-výstup. Užití neuronových sítí je vhodné, když v modelovaném procesu hraje velkou roli náhoda a závislosti v modelu jsou značně složitě identifikovatelné [2].

Neuronové sítě pracují ve dvou fázích. Nejprve se učí nastavit parametry nejlépe vyhovující dané typologii sítě, poté produkují výstupy na základě získaných znalostí. Typologie sítě představuje graf vazeb mezi jednotlivými neurony. Síť má definovány své vstupní, skryté a výstupní vrstvy.

Obrázek 1 znázorňuje schéma modelu neuronu. Jednotlivé vstupní informace  $x$  mají váhy  $w$ . Odečítá se od nich prahová hodnota a pomocí aktivační funkce  $\zeta$  jsou transformovány na výstupní hodnotu  $y$ , která je předána dál.



**Obrázek 1 - model neuronu**

*Pramen: [2]*

Při učení neuronové sítě obvykle začínají s náhodnými váhami pro své neurony. Sítě poté musejí být školeny na řešení zamýšleného konkrétního problému. Při neřízeném učení je neuronová síť vystavena velkému množství dat a směřuje k odhalení souvislostí v těchto datech.

Při řízeném učení (tréninku) jsou vyhodnocovány výstupy neuronové sítě, cílem je minimalizace výstupní chyby  $E$ , tj. odchylky hodnoty vstupní závisle proměnné a vypočítané hodnoty této proměnné. Nejprve se provede výpočet výstupů na základě vstupů a vah, provede se výpočet chyby. Výpočty se provádí pro každý vstup při každém opakování. Zpětně se přitom provádí úprava vah vstupů, podstatou je snaha o minimalizaci chyby  $E$ .

Mezi nejpoužívanější algoritmy řízeného učení patří metoda back propagation. Metoda provádí postupné úpravy synaptických vah, které přibližují chybu lokálnímu (globálnímu) minimu. Globální minimum představuje teoreticky nejmenší možnou chybu [2].

V ekonomické praxi jsou neuronové sítě využívány často velkými institucemi ke zlepšení výkonnosti např. při hodnocení obligací, úvěrovém hodnocení, cíleném

marketingu. Velké užití mají pro analýzy k odhalování podvodů např. při transakcích s kreditními kartami.

## Genetické algoritmy

Genetické algoritmy umožňují řešit složité problémy, u kterých by systematická analýza trvala téměř nekonečně dlouho. Jsou to heuristické postupy, které aplikují principy poznatků evoluční biologie.

Princip vychází z teorie evoluce. Evoluce je velmi úspěšná metoda adaptace v biologických systémech vycházejících z předpokladů že potomci dědí vlastnosti rodičů a lepší jedinci lépe přežívají a mají proto více potomků.

Pro vlastnost jedince se používá označení *fenotyp* a pro jeho reprezentaci se používá termín *genotyp*, *genom* nebo *chromozom*. Chromozom se dělí na jednotlivé sekvenčně uspořádané geny, každý  $i$ -tý gen chromozomů stejného typu reprezentuje stejnou charakteristiku. Kódování chromozomu bývá obvykle popsána binárním řetězcem dané délky [8].

Manipulace s chromozomy probíhá na základě navržených genetických operátorů. Nejčastěji používané operace jsou selekce, křížení a mutace.

Genetický algoritmus sestává dle [2] z následujících kroků:

1. Inicializace – náhodné vytvoření nulté populace  $n$  jedinců
2. ocenění kvality chromozomů populace pomocí fitness funkce
3. tvorba nových chromozomů s použitím selekce, křížení a mutace
4. odstranění starých chromozomů z populace, přidání nových, ocenění nových chromozomů
5. pokud není splněna podmínka pro konec cyklu, tak se pokračuje znovu od bodu 3
6. při splnění ukončující podmínky se označí nejlepší chromozom jako řešení problému a algoritmus se ukončí

Genetické algoritmy se dají použít pro řešení problémů jinak těžko řešitelných, vždy poskytnou nějaké řešení a dají se snadno implementovat, avšak neexistuje záruka,

že nalezené řešení je optimální. Proces výpočtu může být někdy velmi pomalý, například při nevhodném návržení reprezentace jedinců a operátorů křížení a mutace.

V praxi lze genetické algoritmy aplikovat například na problém predikce akciových trhů, různé plánovací problémy, jejichž úlohy jsou vzájemně závislé, optimalizace infrastruktury, uspořádání výrobních hal atd.

### **Zpracování vágnosti a nejistoty**

Metody jsou založeny na analýze způsobu lidského uvažování a snaží se napodobit lidské rozhodování při výskytu neúplných, nepřesných informací a vágních pojmů. Jsou využívány v problematice automatické dedukce. Na základě vágních interpretací jsou schopny zjednodušit popis řešené problematiky a značně urychlit výpočet.

Využívají oblasti umělé inteligence jako Fuzzy logika, Dempster-Shafferova teorie a bayesovské sítě [9]. K dané problematice se budou blíže vyjadřovat následující kapitoly.

### **Expertní systémy**

Expertní systémy jsou počítačové programy, které simulují činnosti specialistů při řešení složitých rozhodovacích úloh a využívají báze znalostí převzatých od expertů s cílem dosažení v dané problematice kvality rozhodování na úrovni experta. Expertní znalosti bývají většinou vágního, neurčitého charakteru. Důležitou roli zde hraje schopnost systému zpracovat neurčitost [10].

Expertní systémy představují v praxi dobře využitelnou aplikaci výzkumu umělé inteligence. Jejich efektivita závisí na kvalitě báze znalostí. Tvorba báze znalostí zahrnuje získávání znalostí od experta, jejich kódování do tvaru, použitelném pro daný řešící mechanismus.

Rozhodovací procesy mohou být realizovány modelem fuzzy logiky, neuronové sítě, genetickými algoritmy, jejich kombinací.

Podle úrovně využívání může expertní systém dle [10] plnit různé funkce:

- kontrolní funkci, kdy systém pracuje jako poradce uživatele na potvrzení či zpochybnění jeho názorů
- návrhovou funkci, kdy systém navrhuje řešení a konečné rozhodnutí závisí na uživateli
- autonomní funkce, kdy systém řeší úkoly, které uživatel není schopen sám vyřešit. Systém zde má konečné rozhodnutí a uživatel obvykle není schopen kontrolovat správnost rozhodnutí

### **3.3 Rozhodování v podmínkách neurčitosti v datech**

Techniky zpracování neurčitosti ve znalostech i v datech jsou velmi důležitou oblastí teorie umělé inteligence.

Neurčitost může být tzv. vynucená, která je způsobena chybami a odchylkami v měřených datech nebo tzv. volitelná, kdy záměrně není potřeba přesnějších informací. Důvodem pro redukci nadbytečných informací bývá snaha o nižší složitost a vyšší přehlednost modelovaného systému a získání dostatečné úrovně informací pro řešení dané úlohy. Schopnost rozhodovat se v situacích, kdy nejsou všechny informace přesné a dostupné, je běžnou vlastností lidského rozumu.

V praxi se neurčité a vágní informace používají při popisu chování velmi složitých systémů, jejichž přesný analytický popis by byl příliš komplikovaný, nákladný a často nemožný.

*„Roste-li složitost nějakého systému, klesá naše schopnost tento systém popsat přesně. Po překročení určité hranice složitosti systému se přesnost a relevantnost popisu stávají vzájemně se vylučujícími charakteristikami.“*

(L. A. Zadeh, Princip inkompability)



Na odstranění problému definovaného L. A. Zadehem se dají uplatnit dle [9] dva přístupy:

- Klasický přístup – zjednodušení předpokladů popisu, vyšší míra zjednodušení však vede k nespolehlivosti výsledků rozhodování, snižuje přesnost modelu.
- Fuzzy přístup – snížení úrovně přesnosti popisu zavedením práce s vágními pojmy. Přístup sice také snižuje přesnost, ale model lze zpracovat v celé jeho složitosti.

Nástroje pro zpracování neurčitých, vágních a nepřesných informací poskytuje fuzzy logika, která vychází z teorie fuzzy množin. Její aplikace umožňuje rychlé nalézání řešení vágně a neúplně popsanych problémů.

### **3.4 Fuzzy množiny a fuzzy logika**

*„Lidé nepotřebují přesnou číslem vyjádřenou informaci a přesto jsou schopni rozhodovat na vysoké úrovni, odpovídající adaptivnímu řízení. Pokud by systémy řízení byly programovány tak, aby uměly pracovat s nepřesnými informacemi na vstupu, mohly by být mnohem účinnější a snad i jednodušeji programovatelné.“*

(L. A. Zadeh)

V klasické teorii množin lze určit, zda daný prvek patří do množiny či nepatří a míru jeho příslušnosti lze vyjádřit charakteristickou funkcí nabývající hodnot 0 nebo 1.

Zadehem vytvořená teorie fuzzy množin určuje míru příslušnosti prvku do množiny, umožňuje, aby prvek do množiny patřil jen částečně. Je zde zavedena funkce příslušnosti, která je definována v rozmezí intervalu  $\langle 0;1 \rangle$ , kdy 0 znamená úplné nečlenství a 1 úplné členství. Míra příslušnosti se dá popsat jako míra subjektivního přesvědčení o tom, zda daný prvek patří do množiny.

Fuzzy množina dle [4] je uspořádaná dvojice

$$A = \{x, \mu_A(x)\}, \forall x \in A$$

kde  $x$  je prvkem množiny  $A$ , patřící do referenční množiny  $X$  a reálné číslo  $\mu_A$  mírou příslušnosti prvku  $x$  k fuzzy množině  $A$ .

Funkce příslušnosti lze interpretovat jako:

$$\mu_A : x \rightarrow \langle 0,1 \rangle$$

Funkce příslušnosti mohou mít různý tvar. Skutečnost nejlépe vystihují funkce ve tvaru křivek, jako Gaussova křivka, sigmoidní křivka, zvonová křivka apod. Pro usnadnění výpočtů se tvary nahrazují funkcemi příslušnosti sestavenými z lomených přímk. Nejčastěji používané tvary po částech lineárních funkcí příslušnosti jsou L-funkce,  $\Gamma$ -funkce,  $\Lambda$ -funkce a  $\Pi$ -funkce, tyto tvary lze definovat dvěma, třemi nebo čtyřmi body.

S fuzzy množinami lze podobně jako s klasickými množinami provádět základní operace průnik, sjednocení, doplněk. Při vyhodnocování se používá různých postupů, které vycházejí z definovaných pravidel.

Pomocí fuzzy množin se dá pracovat se znalostmi, které jsou uváděny v jazykové podobě. Základní jednotou je lingvistická proměnná, jejíž hodnota se vyznačuje neurčitostí, matematická funkce fuzzy množin umožňuje tuto neurčitost zpracovat.

Lingvistická proměnná je popsána jejím jménem, množinou slovních hodnot, číselným rozsahem hodnot a funkcí vyjadřující význam slovních hodnot pomocí fuzzy množin [2].

### 3.4.1 Aplikace fuzzy logiky

Nejčastější aplikací fuzzy logiky jsou fuzzy inference systémy. Používají se ve velké míře při rozhodování, řízení a modelování procesů, ve kterých vystupují veličiny, které lze velice obtížně popsat pomocí konvenčních matematických prostředků.

Tvorba fuzzy inference systému sestává dle [2] ze tří základních kroků:

1. *fuzzifikace*
2. *fuzzy inference*
3. *defuzzifikace*

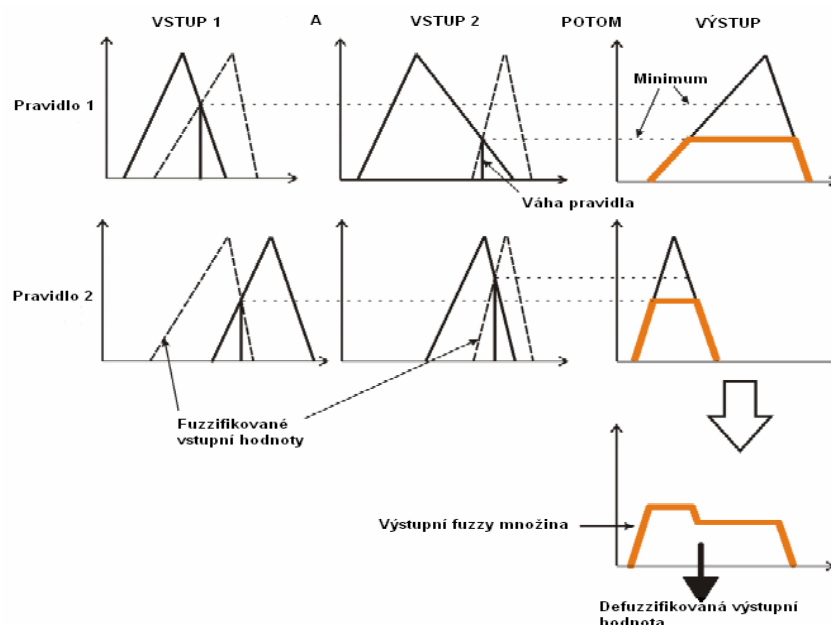
Vstupem fuzzifikace jsou ostré hodnoty vstupních proměnných, které jsou dány jejich referenčními množinami. Výstupem fáze je funkce příslušnosti fuzzy množiny. Dochází k převedení reálných proměnných na proměnné jazykové [4].

Fuzzifikované vstupní hodnoty vstupují do procesu inference. Ve fázi inference se na základě báze dat a pravidel provede výpočet řešení. Báze pravidel sestává z podmíněných pravidel typu *if - then* na jazykové úrovni. Podmíněná pravidla vyhodnocují stav příslušné proměnné.

Soubor pravidel je dle [4] dán:

$$R^{(k)} : \text{if}(x_1 = A_1^{(k)}) \text{and}(x_2 = A_2^{(k)}) \text{and} \dots \text{and}(x_n = A_n^{(k)}) \text{then}(y = B^{(k)})$$

kde  $k=1,2,\dots$  je označení pravidla,  $A^{(k)}$  fuzzy množina na vstupu a  $B^{(k)}$  fuzzy množinu na výstupu. Každé pravidlo představuje fuzzy relaci mezi fuzzy množinami  $A^{(k)}$  a  $B^{(k)}$ . Provede se vyhodnocení fuzzifikovaných vstupních proměnných, určuje se výška průniku fuzzifikovaných vstupních hodnot s příslušnými částmi pravidel před příkazem *then*. Hledá se minimální hodnota určené výšky v rámci každého pravidla, tzn. váha pravidla, která určuje míru podobnosti předpokládaných hodnot a vstupních hodnot. Celková výstupní fuzzy množina je sjednocením závěrů všech pravidel. Viz Obrázek 2. Tento postup fuzzy inference je tzv. *Mamdani* metoda.



Obrázek 2 - Fuzzy inference

Pramen: [4]

Další využívanou metodou fuzzy inference je tzv. *Takagi – Sugeno (Sugeno)* metoda. Ta se od *Mamdani* metody liší tím, že výstupní hodnota je reprezentována přímo funkcí vstupních proměnných. Obvykle se volí konstantní nebo lineární funkce.

$$\text{if}(x_1 = A_1^{(k)}) \text{and}(x_2 = A_2^{(k)}) \text{and} \dots \text{and}(x_n = A_n^{(k)}) \text{then}(y = f(x_1, x_2, \dots, x_n)) \quad [4]$$

Velkou výhodou *Sugeno* modelu je, že se může využít souboru naměřených dat k extrahování údajů o vstupních a výstupních proměnných a o pravidlech, tj. báze dat a báze pravidel. Vhodnou metodou pro vytvoření *Sugeno* modelu je použití Adaptivního neurofuzzy inferenčního systému (ANFIS), který využívá adaptivní neuronovou síť.

Výsledkem fuzzy inference je jazyková proměnná. Ta se ve fázi defuzzifikace převádí na ostré reálné hodnoty. Pro defuzzifikace v *Sugeno* modelu se používá vážený průměr výstupních funkčních hodnot pro jednotlivá pravidla [2].

Řešení úloh s použitím fuzzy logiky je možné například prostřednictvím nástroje *Fuzzy Logic Toolbox* v prostředí programu Matlab. Toolbox umožňuje vytvořit znalostní i datovou bázi fuzzy systému a obsahuje prostředky pro vytvoření, editaci a ladění fuzzy modelu.

### **3.5 Využití fuzzy logiky pro rozhodovací úlohy**

Velkou výhodou použití fuzzy systému pro rozhodování je možnost bezprostředního použití empirických znalostí člověka o řešeném problému, které tvoří bázi znalostí. Pravidla fuzzy logiky slouží jako expertní systém, lze do nich poměrně snadnou formou implementovat znalosti odborníka.

Teorie fuzzy logiky umožňuje používat vágní pojmy, jako je málo, středně, moc atd. Tyto vágní pojmy značně zjednodušují popis řešeného systému. Výhodné je použití při úlohách řízení rizik, v případech, kdy jednotlivá kritéria rizika lze obtížně vyjádřit kvantifikovatelnými hodnotami.

Při navrhování modelu lze do báze znalostí zakomponovat i subjektivní názory odborníka. Například pojem „nízký plat“ může pro různé lidi znamenat odlišný pojem.

Tato subjektivnost může představovat nevýhodu při řešení problémů, vyžadujících vyšší přesnost prezentace dat. Pro tyto úlohy může být užití fuzzy logiky nevhodné.

Důležité je poznamenat, že manažer, který rozhoduje, může při rozhodování používat nejružnější nástroje, které mu rozhodování usnadní, ale konečnou zodpovědnost za dopad rozhodnutí nese sám manažer [2].

### **3.6 Matlab**

Matlab (MATrix LABoratory) je výkonné, interaktivní prostředí pro vědecké a inženýrské výpočty a vizualizaci dat. Matlab integruje numerickou analýzu, maticové výpočty, zpracování signálů a grafiku do prostředí, kde lze problémy a řešení zapisovat jako v matematice. Základním datovým prvkem Matlabu je matice, u které se nezadáva rozměr. Umožňuje to řešení numerických problémů rychleji a výkonněji než použití běžných programovacích jazyků [5].

Matlab je snadno rozšiřitelný, lze jej doplňovat o uživatelsky definované funkce (*M-soubory*) i celé aplikace. K dispozici je celá řada specificky zaměřených nadstaveb, tzv. toolboxů.

#### **Výhody použití Matlabu dle [5]:**

- Rozšířenost – Matlab je celosvětově rozšířený produkt, existuje celá řada rozšiřujících toolboxů pro různé aplikace
- Jednoduchost – snadné ovládání, lze se s ním rychle naučit pracovat
- Import a export dat v mnoha běžných formátech
- Interaktivní nástroj na tvorbu uživatelských rozhraní
- Obecné funkce – umožňují obsáhnout poměrně široký záběr aplikací
- Zabudované numerické metody – např. pro řešení diferenciálních rovnic, maticových operací atd.
- Kvalitní a podrobná nápověda, množství tutorialů

### **Nevýhody použití Matlabu dle [5]:**

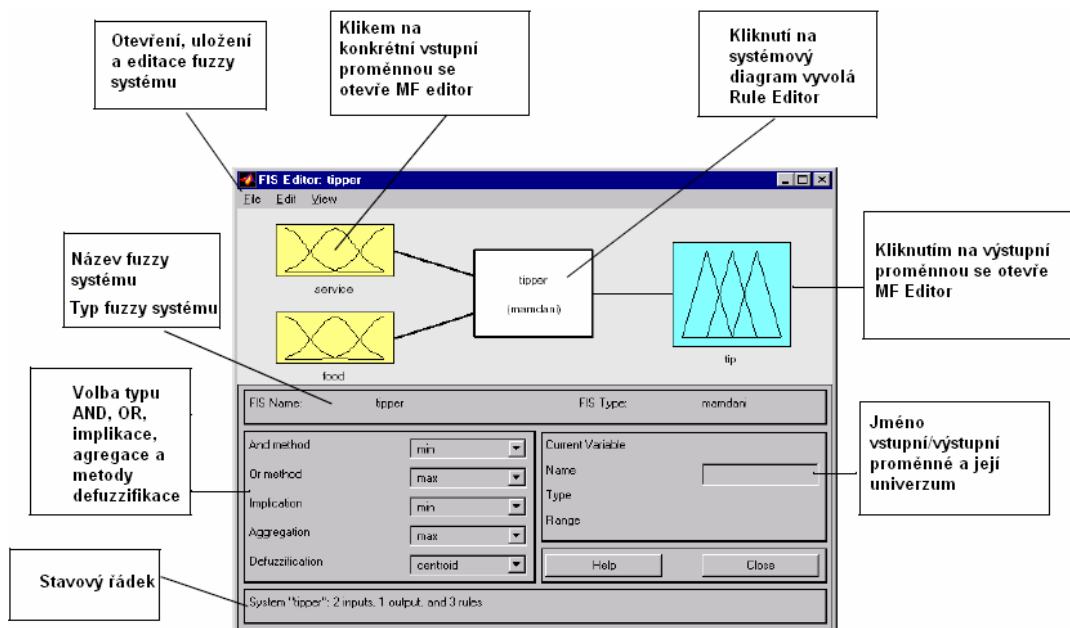
- Jediný pracovní mód (využívání toolboxů) – může být pro řešení některých úloh velmi svazující. Toolboxy jsou tvořeny vždy jen pro určitou oblast. Není příliš vhodný pro složité úlohy, které využívají více toolboxů zároveň, vyžaduje to větší znalosti u řešitele.
- Komplikované vytváření toolboxů – toolboxy musí zapadat do interaktivního prostředí Matlabu
- Nedostupnost zdrojových kódů – jedná se o komerční produkt, proto narozdíl od open-source programů nemůže uživatel upravovat zdrojové kódy samotného jádra Matlabu (zdrojové kódy jednotlivých toolboxů, které jsou řešeny ve formě m-souborů, však upravovat lze).
- Všechny vlastnosti ovlivnitelné a nastavitelné nejen v aplikaci, ale i v prostředí

### **3.6.1 Fuzzy Logic Toolbox**

Fuzzy Logic Toolbox je jedním z mnoha nástrojů programu Matlab. Umožňuje pomocí grafického rozhraní naplnit znalostní i datovou bázi fuzzy systému. Jako v celém prostředí Matlabu lze pracovat s fuzzy systémem z příkazového řádku, ale pro lepší názornost je vhodnější využití grafického rozhraní, tzv. Graphical User Interface – GUI. Příkaz *fuzzy* v příkazovém řádku provede spuštění FL toolboxu.

FL toolbox se skládá ze tří nástrojů pro tvorbu a editaci fuzzy systému – FIS Editoru, MF Editor a Rule Editoru. Obsahuje i dva prohlížeče – Rule Viewer a Surface Viewer [12].

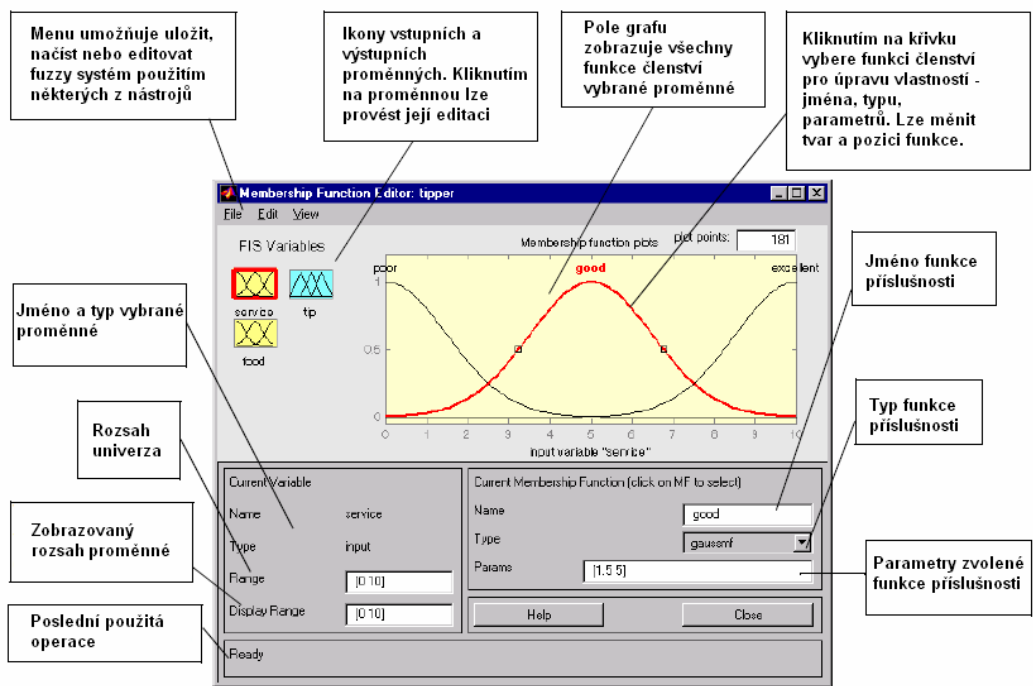
Hlavní částí toolboxu je **FIS editor** (Fuzzy Inference System Editor), ve kterém se definují základní parametry systému, jako počet vstupních a výstupních proměnných, jejich jména, metody defuzzifikace (Obrázek 3).



Obrázek 3 - FIS Editor

*Pramen: [12]*

Dalším nástrojem je **MF Editor** (Membership Function Editor - Obrázek 4), pomocí kterého lze definovat počet a tvar jednotlivých funkcí příslušnosti vstupních a výstupních proměnných, jejich jména a rozsahy. V menu „Edit – Add MFs...“ se dá volit počet a tvar funkcí příslušnosti. Jsou předdefinovány tvary: trimf – trojúhelníková křivka, trapmf – trapezoidální křivka, gbell – obecná zvonová křivka, gaussmf – Gaussova křivka, gauss2mf – dvoustranná Gaussova křivka, pimf – splajnová křivka tvaru  $\Pi$ , dsigmf – rozdíl dvou sigmoidálních křivek, který dává jako výsledek křivku tvaru  $\Pi$ , psigmf – součin dvou sigmoidálních křivek, který dává křivku tvaru  $\Pi$  [12].

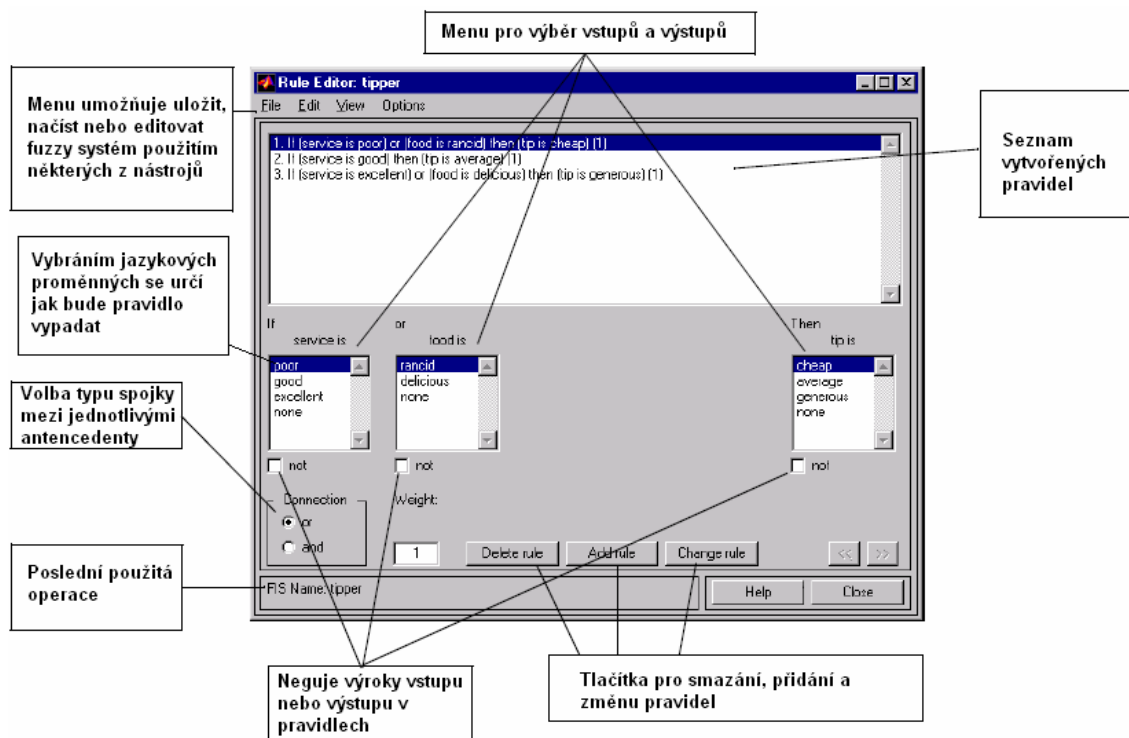


Obrázek 4 - MF Editor

*Pramen: [12]*

K zadávání báze znalostí fuzzy systému slouží grafický **Rule Editor** (Obrázek 5). Zobrazuje vstupní a výstupní proměnné nadefinované pomocí MF Editoru a umožňuje jednoduché vytvoření pravidel typu „IF – THEN“. Způsob tvorby pravidel se liší podle zvoleného typu modelu. Typ Mamdani má v konsekventu (za příkazem *then*) pravidel výrok – výstupní univerzum je pokryto souborem fuzzy množin. Typ Sugeno má v konsekventu pravidel funkci vstupních proměnných.





Obrázek 5 - Rule editor

*Pramen: [12]*

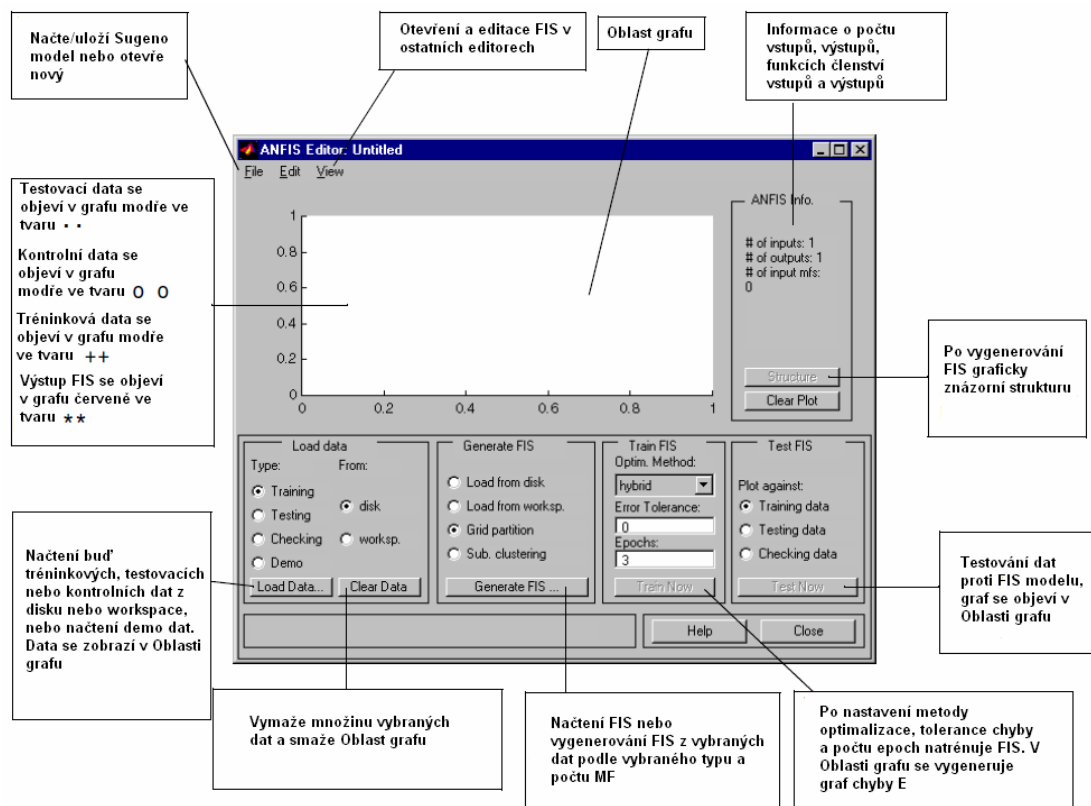
Nástroj **Rule Viewer** umožňuje sledovat chování systému při konkrétních hodnotách vstupních proměnných. V řádcích jsou zobrazena pravidla, ve sloupcích pak vstupní a výstupní proměnné. Zobrazuje se, jakou měrou se na výstupních hodnotách podílejí jednotlivé vstupy.

Nástroj **Surface Viewer** zobrazuje řídicí plochu navrhovaného systému, která je funkcí  $n$ -proměnných. Řídicí plocha je dána průchodem ostrých hodnot přes svá univerza, pro každou  $n$ -tici ostrých hodnot je provedena posloupnosti činností fuzzifikace – fuzzy inference – defuzzifikace, což má za následek jednoznačné určení funkce  $n$ -proměnných [12].

Výstupem systému je FIS matice nebo *\*.fis* soubor, které obsahují všechny informace o navrhovaném fuzzy systému.

Při zvolení *Sugeno* modelu je možné využít souboru naměřených dat k extrahování báze dat a báze pravidel. Pro vytvoření je možné využít metody ANFIS

(Adaptivní Neuro Fuzzy Inferenční Systém), ve Fuzzy Logic Toolboxu je pro tuto metodu vytvořen ANFIS editor GUI [12].



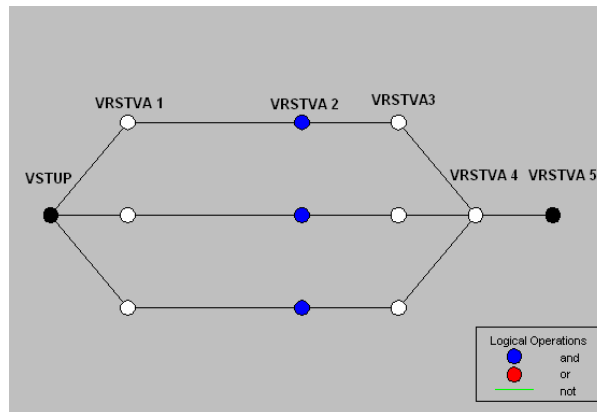
Obrázek 6 - ANFIS Editor

*Pramen: [12]*

ANFIS editor (Obrázek 6) umožňuje načtení souboru dat se vstupními a výstupními proměnnými, volbu tvaru a počtu funkcí příslušnosti a zvolení metody pro učení adaptivní sítě.

Pro proces učení využívá ANFIS metody řízeného učení neuronové sítě (metodou zpětného šíření chyby) nebo hybridní metody, která kombinuje zpětné šíření chyby s metodou nejmenších čtverců. Lze navolit minimum hodnoty chyby  $E$  a počet epoch (iterací) výpočtu.

Součástí editoru je i zobrazení struktury vrstev adaptivní neurofuzzy sítě.



Obrázek 7 - Příklad struktury ANFIS

*Pramen: [12]*

Popis struktury (Obrázek 7):

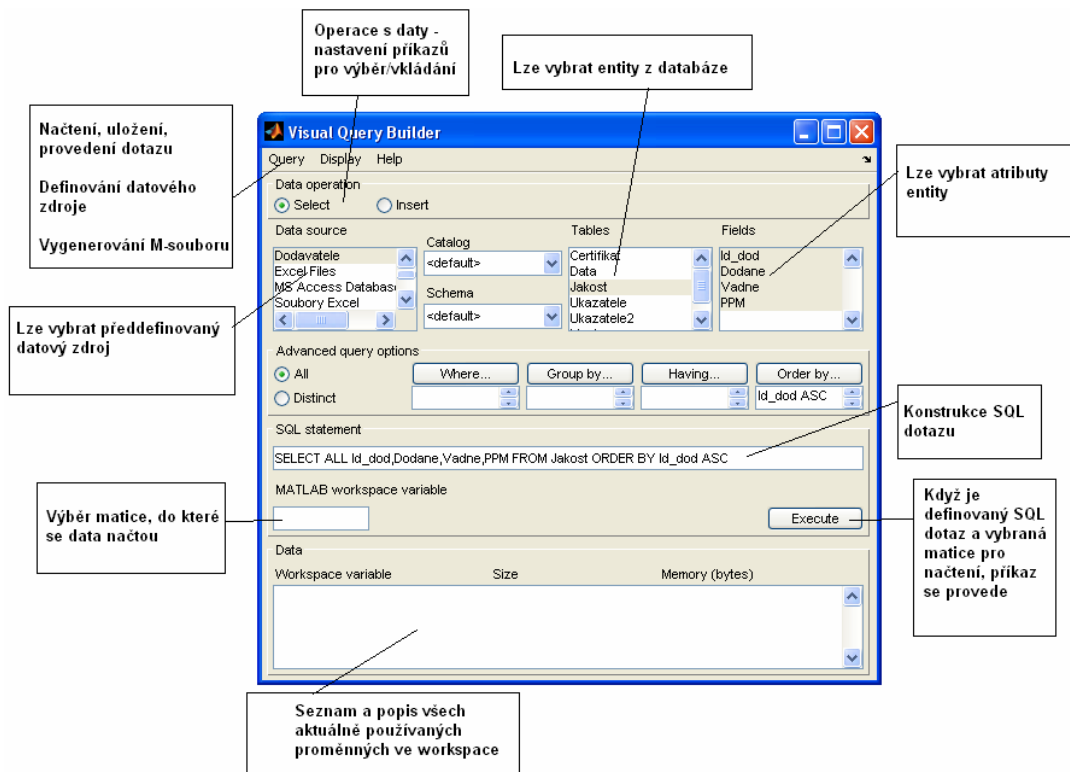
- VRSTVA 1 – Každý uzel  $i$  vrstvy je adaptivní. Výstupem je hodnota příslušnosti vstupu  $x$  k úrovni  $A_i$  lingvistické proměnné
- VRSTVA 2 – Každý uzel vrstvy je neadaptivní. Násobí vstupní signály a vysílá jejich součin. Představuje sílu pravidla
- VRSTVA 3 – Každý uzel vrstvy je neadaptivní, počítá poměr síly  $i$ -tého pravidla k součtu všech sil pravidel
- VRSTVA 4 – Uzel je adaptivní
- VRSTVA 5 – Neadaptivní uzel plní funkci sčítání vstupních signálů. Při počtu vstupů  $x_n$  a počtu funkcí příslušnosti  $MF_i$  je počet všech možných pravidel  $Rule_n$ :

$$Rule_n = \prod_{i=1}^{x_n} MF_i$$

### 3.6.2 Database toolbox

Database toolbox umožňuje výměnu dat mezi workspace Matlabu a relačními databázemi. Lze například načíst data z databáze do Matlabu, využít nástrojů Matlabu na analýzu nebo zpracování dat a poté zapsat výsledky zpátky do databáze, případně i do jiné databáze. Toolbox obsahuje Visual Query Builder, který je grafickým

prostředím pro práci s databázemi a umožňuje vytvářet dotazy nad databázi. Toolbox sám o sobě není databázovým nástrojem, je potřeba mít samostatný systém řízení báze dat (MS Access, MS SQL Server, MySQL, Oracle...), toolbox pak zprostředkovává spojení [13].



Obrázek 8 - Database toolbox

Před připojením je třeba nastavit zdroj dat – adresář, server, ovladač. Zdroj dat komunikuje s ODBC rozhraním MS Windows nebo JDBC pro aplikace založené na Javě. Zdroj dat se definuje pro práci s ODBC zdroji dat v menu *Query/ Define ODBC data Source*. Otevře se Správce zdrojů dat ODBC z Ovládacích panelů Windows. Volbou *Přidat* lze zvolit příslušný driver z knihovny ovladačů Microsoft Data Access Components (MDAC). Z nejpoužívanějších datových zdrojů obsahuje např. *Driver do Microsoft Access (\*.mdb)*, *Microsoft Excell Driver (\*.xls)*, *SQL server*, *Microsoft ODBC for Oracle* a jiné [2].

Po stisknutí volby *Dokončit* je třeba definovat připojení k databázi.

Data načtená z databáze se ukládají do matice ve workspace Matlabu. To umožní snadné a rychlé zpracování dat [13].

## 4 Analýza současného stavu

### 4.1 Název, právní forma a sídlo společnosti

MINERVA BOSKOVICE, a.s.

Sokolská 60, Boskovice, PSČ 680 17

Hodnota základního kapitálu **MINERVY BOSKOVICE, a.s.** činí 312 899 tis. Kč a je zcela splacena. Majoritním akcionářem je společnost Dürkopp Adler AG, Potsdamer Straße 190, D-33719 Bilerfeld, Spolková republika Německo, která vlastní (ke dni 31.12.2008) 87,99% celkové výše základního kapitálu s hlasovacím právem a je právnickou osobou ovládající **MINERVA BOSKOVICE, a.s.** Mezi oběma subjekty není uzavřena ovládací smlouva. Ovládající osoba nesmí využít svého vlivu k tomu, aby prosadila přijetí opatření, z nichž může ovládané osobě vzniknout majetková újma. Představenstvo ovládané osoby má povinnost ve lhůtě 3 měsíců od skončení účetního období zpracovat písemnou zprávu o vztazích mezi ovládající a ovládanou osobou.

Firma Minerva Boskovice má sesterské společnosti v USA, Francii, Polsku, Itálii, Rumunsku atd. Společnost **MINERVA BOSKOVICE, a.s.** nemá v současné době majetkovou účast v žádné právnické osobě.

Areál podniku Minerva leží v Boskovicích nedaleko autobusového nádraží a rozkládá se na 84 000 m<sup>2</sup>, přičemž zastavěná plocha je 38 000 m<sup>2</sup>, z toho výrobní plocha činí 32 000 m<sup>2</sup>. V areálu se nachází několik výrobních budov, budovy skladů (materiálový a odbytový sklad) a správní budovy (obchodní úsek, ekonomický úsek a IT středisko).

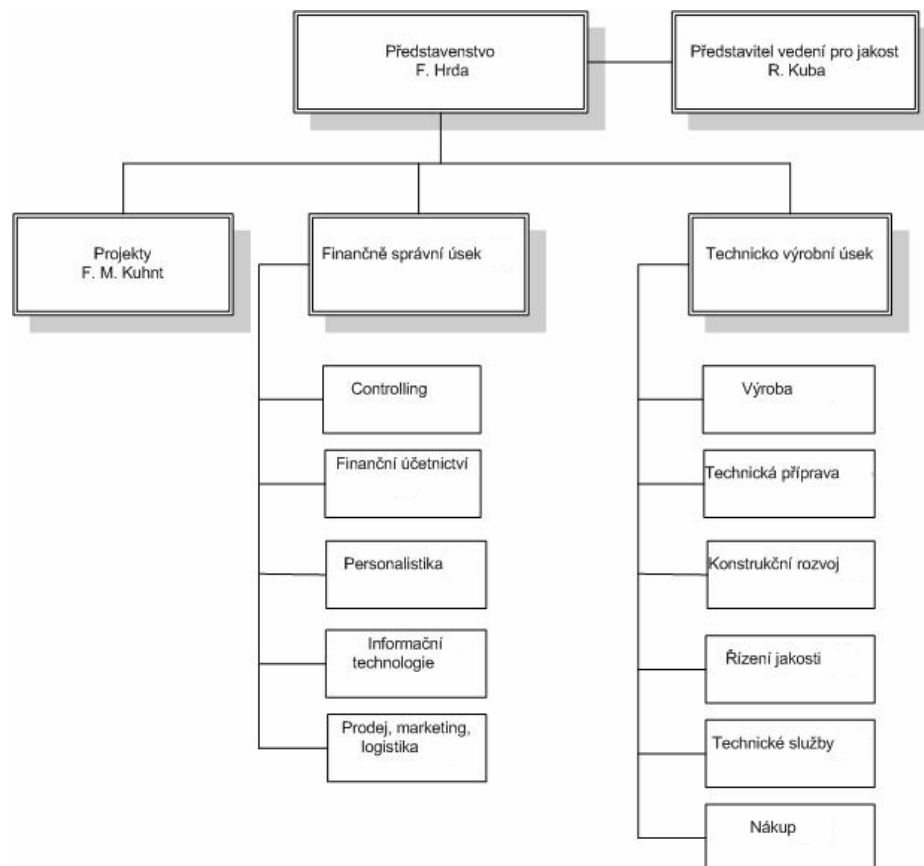
### 4.2 Předmět podnikání, výrobní sortiment

Hlavní činností společnosti je výroba, instalace a opravy elektrických průmyslových šicích strojů a přístrojů pro šití obuvi, kůže, textilu a galanterie, která představuje 93% výnosů společnosti. Jelikož se jedná o společnost pohybující se v oblasti strojírenství, do předmětu podnikání dále patří:

- Stavba strojů s mechanickým pohonem
- Výroba a opravy elektrických šicích strojů a elektrických přístrojů pro domácnost
- Obchodní činnost, včetně zahraniční, s výjimkou činností taxativně stanovených živnostenským zákonem
- Kovoobrábění
- Zámečnictví
- Výzkum a vývoj v oblasti přírodních a technických věd nebo společenských věd

### 4.3 Organizační struktura společnosti

V čele společnosti **MINERVA BOSKOVICE, a.s.** je představenstvo, pod které spadá jak úsek pro kontrolu jakosti, tak i tři další úseky: úsek služeb, finančně-správní úsek a technicko výrobní úsek.



Obrázek 9 - Organizační struktura společnosti

## **4.4 Současný vývoj hospodaření**

Události v průběhu roku 2008 až poloviny roku 2009 byly pro společnost Minerva Boskovice, a.s. zlomové. Celosvětová ekonomická krize výrazně zasáhla také výrobu a prodej průmyslových šicích strojů. Po předchozích letech ekonomického růstu došlo nejen k poklesu obrátu a výroby, ale také produktivity práce a propadu hospodářského výsledku. K prudkému poklesu zakázek došlo především v oblasti středně těžké řady strojů pro šití automobilů. Společnost na tuto situaci reagovala snížením výroby a redukcí nákladů, snížením stavu zaměstnanců a omezováním výdajů v oblasti investic. Pokles celkového obrátu činil 49,3%. Za rok 2009 hospodařila společnost se ztrátou ve výši 124 381 tis. Kč po zdanění. Došlo k vyšší zadluženosti vůči mateřské společnosti, díky této pomoci se podařilo zvládnout krizové období a zajistit stabilitu společnosti.

Hlavním cílem roku 2009 bylo minimalizovat ztráty, podržet si klíčové pracovníky a nastavit vlastní procesy tak, aby společnost mohla rychle reagovat na požadavky zákazníků v následujících obdobích. V roce 2009 obdržela firma dotace od MPO na podporu výzkumu a prodeje. Tyto prostředky použila na financování projektu „Nová řada průmyslových šicích strojů M-Type“.

Po předchozím útlumu prodeje dochází koncem roku 2009 opět k nárůstu poptávky, přičemž velké množství zakázek je požadováno v poměrně krátkých termínech. Z důvodů snížených výrobních kapacit si situace od firmy vyžaduje především zvýšené úsilí v optimalizaci logistického řetězce, aby mohla plnit požadované zakázky.

## **4.5 Informační systém**

### **4.5.1 SAP R/3**

Firma používá ERP systém SAP R/3, ovšem jen jeho základní moduly :

- AM - Účetnictví investičního majetku
- FI - Finanční účetnictví
- SD - Odbyt a prodej

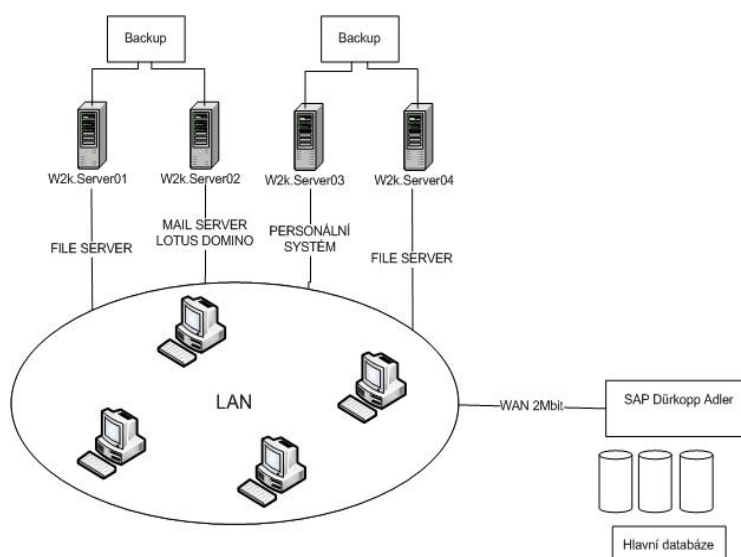
- MM - Materiálové hospodářství a logistika
- PP - Plánování a řízení výroby
- HR - Personalistika a mzdy

Báze dat je realizována nad SŘBD v Oracle, systém umožňuje pohodlný a transparentní přístup k datům mezi jednotlivými moduly. Centralizace dat vede ke snížení chyb a zefektivnění a zrychlení ekonomických procesů.

Informační systém umožňuje podniku udržovat kvalitní datovou základnu, chybí ovšem pokročilé nástroje pro získávání znalostí z dat. Samotné získávání znalostí spočívá ve zpracování dat příslušným administrativním pracovníkem a přináší nevýhody jako nepružnost v získávání, nízká transparentnost nebo vysoká pracnost.

#### 4.5.2 Správa dat pro informační systém

Pro účely systému se využívá čtyř serverů – dva File Servery, Mail Server a Server pro personální systém. Informační systém Minervy je propojen přes pronajatou linku se systémem SAP mateřské společnosti Dürkopp Adler v Německu. Na straně mateřské společnosti jsou i hlavní databázové servery. Všechna důležitá data jsou denně zálohována na backup pásky a na hlavním serveru mateřské společnosti v Německu.



Obrázek 10 - Systém správy dat



## **4.6 Řízení nákupu**

Nákup je realizován v modulu *MM - Materiálové hospodářství a logistika* systému SAP. Touto fází logistického řetězce se zabývá oddělení Nákupu. Konkrétní materiálové požadavky přiřazuje systém jednotlivým nákupním referentům, kteří mají zodpovědnost za bezproblémové zajištění materiálu pro potřeby jednotlivých zakázek. Relevantní data jsou obsažena v databázi informačního systému.

Z hlediska současné situace, kdy výrazně narůstá počet zakázek, které jsou požadovány v poměrně krátkých termínech, značně roste význam správného řízení dodávek materiálu. Tato skutečnost se týká především materiálových druhů, u nichž není z hlediska celkové nákladovosti výhodné držet zásoby. Jedná se o materiálovou skupinu, která má ve finančním vyjádření největší podíl na celkové spotřebě firmy. Pro udržení nízkého stavu zásob se tento materiál dodává Just-in-Time přímo pro potřeby výroby, přičemž každá odchylka v termínech, množství, či kvalitě komplikuje plnění zakázek.

Důležitým úkolem se pro oddělení Nákupu stává hodnocení výkonnosti materiálových dodavatelů. Rozhodování o nákupu závisí na tomto hodnocení, požaduje se proto navržení lepšího systému hodnocení.

## **4.7 Řízení jakosti**

Kvalita produkce společnosti se opírá o systém jakosti certifikovaný firmou BVQI podle mezinárodní normy EN ISO 9001:2008. Politika jakosti je každý rok rozpracována do jednotlivých cílů a to se zaměřením na výrobek, na proces a na systém. Tyto cíle jsou v průběhu roku vyhodnocovány Radou jakosti. Reklamace od zákazníků jsou uplatňovány přes odbor prodeje a logistiky. Tyto reklamace jsou průběžně evidovány a vyřizovány v zákonné lhůtě. Z celkového počtu prodaných obuvnických strojů za rok 2008 se reklamace pohybují ve výši 1,08 %.

System pro řízení jakosti je realizován pomocí kontingenčních tabulek v excelu a umožňuje sledovat jakostní ukazatele pro procesy na vstupu, výstupu i v rámci

vnitropodnikové části logistického řetězce. Systém je poměrně propracovaný, velkou nevýhodou však zůstává jeho izolace od nástrojů firemního informačního systému SAP.

#### **4.8 *Současný systém hodnocení dodavatelů***

Současný model pro hodnocení dodavatelů ve firmě sestává z poměrně jednoduché metody známkování vedené v Excelu, kde se dodavatelé rozdělují do skupin podle udělené celkové známky, která je tvořena známkou za jakost, známkou za cenu a známkou za dodávku. Jednotlivé známky se zadávají ručně dle podkladů nákupních referentů.

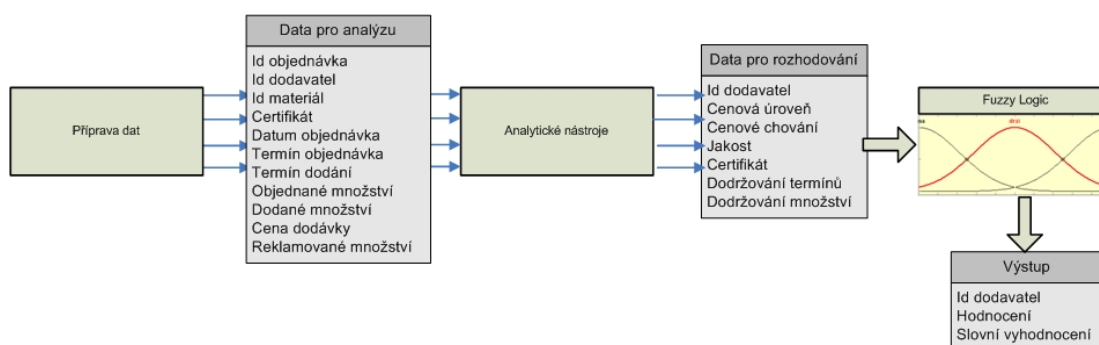
#### **4.9 *Závěry z analýzy***

Z analýzy současného stavu lze vyvozovat následující závěry:

- roste význam hodnocení dodavatelů jako informačního nástroje pro rozhodování o nákupu
- současný systém hodnocení je nevyhovující z hlediska vypovídací schopnosti a pracnosti vyhotovení
- jako zdroje potřebných dat pro hodnocení lze využít databází IS namísto podkladů nákupních referentů
- při načítání dat z databáze lze značně automatizovat celý proces hodnocení

## 5 Vlastní řešení

Vlastní návrh řešení se zabývá využitím získávání dat z podnikové databáze a aplikace fuzzy logiky pro systém hodnocení stávajících materiálových dodavatelů pro firmu. Obrázek 11 představuje schematické znázornění fází řešení.



Obrázek 11 - Model řešení

Řešení sestává z následujících fází:

- Příprava dat* – se skládá z určení relevantních dat, způsobů jejich získání a extrakce do použitelné podoby. Většinu požadovaných nákupních dat lze získat z databáze systému SAP, další data lze načíst z evidence nákupu a jakosti v tabulkách Excel (evidence vychází z logiky používané v SAPu, lze proto data pro analýzu využít). Výstupem fáze přípravy dat je databáze v MS Access, která obsahuje tabulky požadovaných dat.
- Analýza dat* – v této fázi probíhá výpočet ukazatelů pro hodnocení dodavatelů na základě připravených dat. Výpočet je realizován vytvořeným algoritmem v programu Matlab. Výstupem fáze analýzy dat je tabulka ukazatelů pro jednotlivé dodavatele, která slouží jako množina dat pro zpracování aplikací využívající fuzzy logiku.
- Tvorba fuzzy modelu* – v této fázi proběhne vytvoření modelu pro hodnocení dodavatelů. Aby model co nejvíce odpovídal skutečným požadavkům hodnocení, je vhodné jej vygenerovat v ANFIS editoru v Matlabu podle závislosti vstupy/výstup v souboru tréninkových dat.

- d) *Hodnocení dodavatelů fuzzy modelem* – konečná fáze využívá vytvořený model pro zpracování ukazatelů. Zpracování je realizováno algoritmem v Matlabu a konečným výstupem je tabulka dat v databázi obsahujících hodnocení jednotlivých dodavatelů a jejich rozdělení do skupin dle tohoto hodnocení.

## **5.1 Hodnocení dodavatelů**

Hlavním účelem hodnocení dodavatelů ve společnosti Minerva Boskovice a.s. je měřit výkonnost stávajících dodavatelů (z nichž velkou část Minerva převzala od mateřské firmy) a určit vhodnost těchto dodavatelů pro efektivní spolupráci. Jedná se o vybrané dodavatele poskytující materiál spadající do skupiny, která má ve finančním vyjádření nejvyšší podíl hodnot na výši celoroční spotřeby – dle řízení zásob metodou ABC se jedná o materiál ze skupiny A.

### **5.1.1 Cíle hodnocení**

System hodnocení musí poskytovat měřitelné informace o výkonnosti jednotlivých dodavatelů, na jejichž základě lze učinit rozhodnutí o povaze vztahu s dodavatelem. Rozdělením do skupin dle toho, do jaké míry vyhovují požadavkům nákupu, lze identifikovat dodavatele, od kterých může firma nakupovat za určitých kritických podmínek, dodavatele, se kterými by firma měla upravit svoje vztahy (např. vyžádat interní audit) nebo dodavatele, se kterými by bylo nejlepší ukončit spolupráci.

Dle výsledku hodnocení může oddělení nákupu získat lepší podporu rozhodování pro činnosti, kdy potřebuje:

- zajistit kvalitu dodávek tím, že se trvale zlepšuje kvalita dodavatelů
- včas podchytit pokles kvality dodavatele, umožnit vyvození závěrů (jakým způsobem na pokles reagovat – audit dodavatele, rozvázání vztahu)
- získat objektivní informace o dodavateli tím, že se vyhodnotí reálná zkušenost spolupráce s využitím objektivně měřitelných dat
- optimalizovat portfolio dodavatelů

## 5.1.2 Stanovení kritérií a jejich vah

Při stanovování kritérií pro hodnocení je třeba brát v potaz, že neexistuje žádná univerzální metoda, která by byla použitelná pro všechny organizace. To znamená, že se musí vycházet z praktické situace ve firmě.

Musí se určit, jaké požadavky firma na své dodavatele klade. Je zohledňována cenová výhodnost, a to jak samotná cena materiálu, tak i její vývoj. Dále je kladen důraz na vysokou jakost materiálu, vedení firmy požaduje od dodavatelů certifikaci dle norem ISO 9000. Mezi další požadavky patří pružnost dodavatele – ať už se jedná o včasnost dodávek, zajištění potřebného množství nebo reakce na změny. Minerva usiluje o minimalizaci skladových nákladů, preferuje dodávky Just-in-Time přímo pro potřeby výroby; včasnost dodávek je tedy velmi významným kritériem.

Lze volit i další, pro firmu méně významná kritéria, jako vzdálenost dodavatele, ekologická politika nebo finanční stabilita.

Na základě posouzení výše uvedených požadavků lze již navrhnout a stanovit nejhodnější kritéria hodnocení. Níže uvádím 6 zvolených, včetně váhy jejich významu v rámci celého modelu.

Váhy jednotlivých kritérií jsou stanoveny dle jejich priority pro hodnocení a při jejich stanovení vycházím ze subjektivního hodnocení stanoveného dle posouzení vedoucího Oddělení nákupu. Součet všech vah je roven 1.

### *I. Cenová kritéria*

#### 1. Cenová úroveň (váha 0,24)

Sestaveno dle porovnání ceny daného materiálu s cenou ostatních možných dodavatelů

#### 2. Cenové chování (váha 0,16)

Sestaveno dle vývoje ceny u jednotlivého dodavatele

## *II. Jakostní kritéria*

### 3. Jakost (váha 0,24)

Určena jako poměr reklamací k celkovému objemu dodávky u jednotlivých dodavatelů

### 4. Certifikát (váha 0,06)

Jestliže dodavatel má/nemá certifikát

## *III. Kritéria dodání*

### 5. Dodržování termínu (váha 0,18)

Stanoveno jako odchylka data dodávky v objednávce od skutečného data příjmu materiálu.

### 6. Dodržování množství (váha 0,12)

Představuje odchylku mezi objednaným a dodaným množstvím.

## **5.2 Příprava dat**

Pro potřeby analýz je nejprve nutné určit relevantní data a jejich zdroje.

Většinu potřebných dat lze získat přímo ze záznamů podnikového systému. Zajímají nás záznamy týkající se objednávek od dodavatelů materiálu, u nichž byl roční obrat minimálně 100 tis. Kč, a za období 3 měsíců (tj. od 1.1.2010 do 31.3.2010, kdy firma po období útlumu začala opět navyšovat produkci).

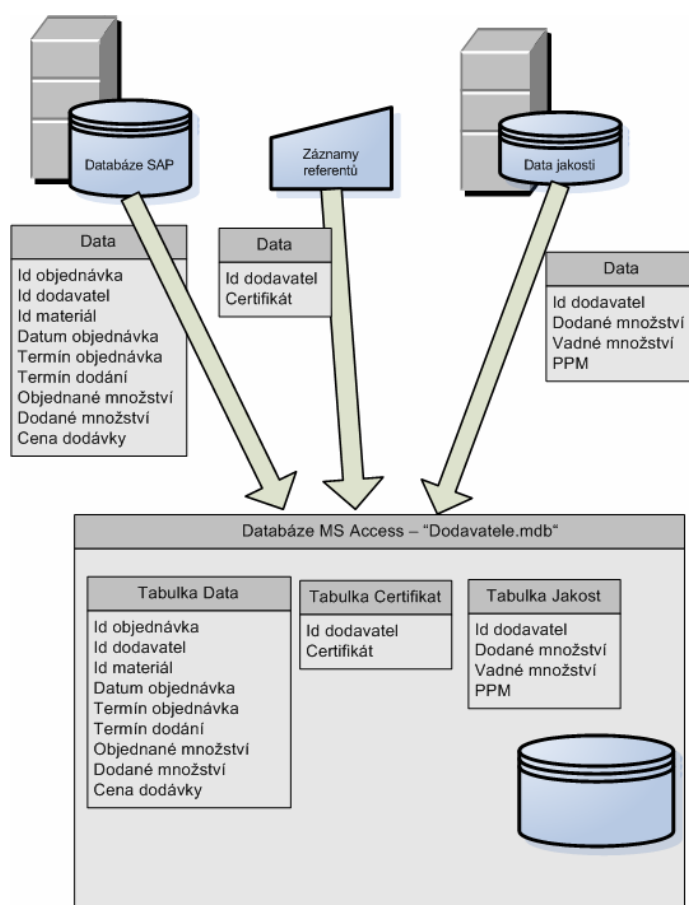
Výběr dat lze provést zadáním SQL dotazu nad databází systému v Oracle a výsledek dotazu poté importovat do databáze v MS Access.

Při přípravě dat se objevují 2 problémy:

1. Informace o platném certifikátu u dodavatele není vedena v systému SAP, ale pouze v podkladech nákupních referentů. Data se zadávají ručně ve tvaru {'A'; 'N'} do tabulky v Excelu. Pro identifikaci dodavatele je zde použito jeho číslo

ze systému SAP, je proto možné data přímo importovat do databáze v MS Access, aniž by došlo k problémům při vytváření vazeb mezi daty.

2. Oddělení jakosti používá pro evidenci reklamací materiálu vlastní systém z kontingenčních tabulek v Excelu, nikoliv SAP. Ale čísla dodavatelů zde odpovídají záznamům v SAPu, a tabulky obsahují požadovaná data dodané množství, vadné množství a ukazatel PPM (udávající pravděpodobný počet výskytů vadného materiálu na milion dodaných). Požadovaná data lze proto přímo importovat do databáze v MS Access.



Obrázek 12 - Fáze přípravy dat

Obrázek 12 znázorňuje výběr datových zdrojů a extrakci dat do požadované podoby. Výsledkem extrakce dat je databáze Dodavatele.mdb v MS Access, která obsahuje tabulku *Data* (obsahující data exportovaná ze SAPu), *Certifikát* a *Jakost*.

Tabulka 1 zobrazuje strukturu atributů, datové typy a velikosti atributů v entitě *Data*. Tabulka 2 zobrazuje strukturu atributů, datové typy a velikosti atributů v entitě *Certifikat* a Tabulka 3 zobrazuje strukturu atributů, datové typy a velikosti atributů v entitě *Jakost*.

Tabulka Data			
	Název pole	Datový typ	Délka
PK	Id_obj	text	8
	Id_dod	text	8
	Id_mat	text	8
	Datum_obj	Datum a čas	
	Termin_obj	Datum a čas	
	Termin_dodani	Datum a čas	
	Obj_mn	číslo	dlouhé celé číslo
	Dod_mn	číslo	dlouhé celé číslo
	Cena	číslo	dvojitá přesnost

Tabulka 1 - Struktura entity Data

Tabulka Certifikat			
	Název pole	Datový typ	Délka
PK	Id_dod	text	8
	Certifikat	text	1

Tabulka 2 - Struktura entity Certifikat

Tabulka Jakost			
	Název pole	Datový typ	Délka
PK	Id_dod	text	8
	Dodane	číslo	dlouhé celé číslo
	Vadne	číslo	dlouhé celé číslo
	PPM	číslo	dvojitá přesnost

Tabulka 3 - Struktura entity Jakost

Dále je třeba připravit tabulky pro uložení výsledků analýz, tabulku *Ukazatele* pro vypočítané ukazatele pro hodnocení a tabulku *Vystup* pro výsledky hodnocení dodavatelů.



Tabulka 4 zobrazuje strukturu atributů, datové typy a velikosti atributů v entitě *Ukazatele*. Tabulka 5 zobrazuje strukturu atributů, datové typy a velikosti atributů v entitě *Vystup*.

Tabulka Ukazatele			
	Název pole	Datový typ	Délka
PK	Dodavatel	text	8
	Uroven	číslo	dvojitá přesnost
	Chovani	číslo	dvojitá přesnost
	Certifikat	číslo	celé číslo
	Jakost	číslo	dvojitá přesnost
	Terminy	číslo	dvojitá přesnost
	Mnozstvi	číslo	dvojitá přesnost

Tabulka 4 - Struktura entity Ukazatele

Tabulka Vystup			
	Název pole	Datový typ	Délka
PK	Dodavatel	text	8
	hodnoceniCena	číslo	dvojitá přesnost
	hodnoceniJakost	číslo	dvojitá přesnost
	hodnoceniDodani	číslo	dvojitá přesnost
	hodnoceniCelk	číslo	dvojitá přesnost
	skupina	číslo	celé číslo

Tabulka 5 - Struktura entity Vystup

### 5.3 Výpočet ukazatelů

Pro každého dodavatele je určeno 6 ukazatelů, které slouží jako vstupní proměnné pro hodnocení ve Fuzzy modelu.

Za účelem výpočtu ukazatelů pro rozhodování je v Matlabu vytvořen M-soubor *ukazatele.m*.

Úloha sestává z načtení dat z tabulek *Data*, *Certifikat* a *Jakost* databáze *Dodavatele.mdb*. Připojení k databázi je umožněno prostřednictvím ODBC driveru nadefinovaném v toolboxu Visual Query Builder. Značnou výhodou pro práci s daty je maticové orientování Matlabu, kde lze požadovaná data načítat do matic a zpracování těchto dat provádět pomocí různých úprav těchto matic (výběr řádků a sloupců, sumarizace a průměrování vektorů, vektorové součiny a podíly atd.).

Úloha je vytvořena za účelem automatického zpracování většího objemu dat na zvolené ukazatele s požadovanou vypovídací hodnotou. Po ukončení výpočtů se ukazatele uloží do databáze a slouží jako podklad pro rozhodování pomocí Fuzzy logiky.

### 5.3.1 Načtení dat z databáze

První část M-souboru definuje připojení k databázi a načítá prostřednictvím SQL dotazů požadovaná data do matic ve Workspace programu.

Dotaz *'SELECT Data.Id\_dod, Jakost.PPM, Certifikat.Certifikat FROM ((Data LEFT OUTER JOIN Jakost ON Data.Id\_dod = Jakost.Id\_dod) LEFT OUTER JOIN Certifikat ON Data.Id\_dod = Certifikat.Id\_dod) GROUP BY Data.Id\_dod, Jakost.PPM, Certifikat.Certifikat'* slouží k propojení tří tabulek z databáze vnějším spojením, které k *Id\_dod* přiřadí odpovídající záznamy z dalších tabulek a v případě neexistence odpovídajícího záznamu doplní hodnotu Null. Umožní seskupením vybrat z dat o objednávkách čísla jednotlivých dodavatelů a přiřadit jim míru zmetkovitosti (PPM) z tabulky *Jakost* a certifikát z tabulky *Certifikat*.

Dotaz *'SELECT Id\_dod, Id\_mat, Datum\_obj, Termin\_obj, Termin\_dodani, Obj\_mn, Dod\_mn, Cena FROM Data ORDER BY Id\_dod'* vybere z tabulky *Data* o jednotlivých objednávkách a seřadí je podle čísla dodavatele.

Dotaz *'SELECT Id\_dod FROM Data GROUP BY ID\_dod'* vybere z dat o objednávkách jednotlivá čísla dodavatelů.

Dotaz *'SELECT Id\_mat, SUM(Cena) / SUM(Dod\_mn) AS PrumCena FROM Data GROUP BY Id\_mat'* vybere z dat o objednávkách jednotlivé typy materiálu a přiřadí jim průměrnou cenu tohoto materiálu.

```

s.DataReturnFormat = 'cellarray'; s.ErrorHandling = 'store';
s.NullNumberRead = '0';
s.NullNumberWrite = '0'; s.NullStringRead = 'null'; s.NullStringWrite
= 'null';
s.JDBCDataSourceFile = ''; s.UseRegistryForSources = 'yes';

setdbprefs(s)

conn = database('Dodavatele', '', 'password');

e = exec(conn, 'SELECT Data.Id_dod, Jakost.PPM, Certifikat.Certifikat
FROM ((Data LEFT OUTER JOIN Jakost ON Data.Id_dod = Jakost.Id_dod)
LEFT OUTER JOIN Certifikat ON Data.Id_dod = Certifikat.Id_dod) GROUP
BY Data.Id_dod, Jakost.PPM, Certifikat.Certifikat');
e = fetch(e);
close(e)

f = exec(conn, 'SELECT Id_dod, Id_mat, Datum_obj, Termin_obj,
Termin_dodani, Obj_mn, Dod_mn, Cena FROM Data ORDER BY Id_dod');
f = fetch(f);
close(f)

dod = exec(conn, 'SELECT Id_dod FROM Data GROUP BY ID_dod');
dod = fetch(dod);
close(dod)

mat = exec(conn, 'SELECT Id_mat, SUM(Cena) / SUM(Dod_mn) AS PrumCena
FROM Data GROUP BY Id_mat');
mat = fetch(mat);
close(mat)

close(conn)

```

#### Program 1 – Načtení dat z databáze

Zdrojový kód umožňuje načtení dat z databáze v souboru *Dodavatele.mdb*. Příkaz *setdbprefs(s)* definuje databázi podle množiny preferencí, nastavené v proměnné *s*. *DataReturnFormat* nastaví výstup dat v matici buněk (cellarray), do níž lze vkládat různé datové typy. Důležitá je vlastnost *NullNumberRead*, která zajistí, že v případě načtení hodnoty Null bude tato hodnota převedena do číselného formátu s hodnotou 0 (např. při vnějším spojení s tabulkou jakost, kdy se předpokládá, že nenalezenou relaci v tabulce lze prezentovat jako nulovou úroveň zmetkovitosti).

Příkaz *conn* připojuje k databázi Dodavatele.

Příkaz *exec* funguje jako kurzor, vykonává SQL dotaz nad databází a výsledky dotazu předává do jednotlivých proměnných *e*, *f*, *dod* a *mat*. Příkaz *fetch* převádí data do podoby cellarray.

Do proměnné *e* jsou načtena data z dotazu, který přiřazuje jednotlivým Id dodavatele z tabulky Data hodnoty ukazatele PPM z tabulky jakost a údaj Certifikát z tabulky Certifikát .

Do proměnné *f* se načítají údaje týkající se jednotlivých objednávek z tabulky Data.

Do proměnné *dod* se vytvoří seznam analyzovaných dodavatelů.

Do proměnné *mat* se načítají průměrné ceny daného typu materiálu.

Příkaz *close(conn)* ukončí spojení s databází.

### 5.3.2 Přepočítání ukazatelů Certifikát a Jakost

Tato část M-souboru zpracuje ukazatele Certifikát a Jakost. Certifikát je v databázi uveden ve formátu {A;N}, ale pro účely další analýzy je požadován formát {0;1}. Jakost je uvedena ve formě počet vadných kusů na milion, je třeba převést na procentní vyjádření.

```
vstup = e.data;
Velikost=size(vstup);
PocRadku=Velikost(1);

for i=1:PocRadku
    radek=vstup(i,:);

    if radek{3}=='A'
        certifikat=1;
    elseif radek{3}=='N'
        certifikat=0;
    else
        certifikat=0;
    end
    radek{3}=certifikat;

    if isnumeric(radek{2})
        jakost=radek{2}/10^4;
    else
        jakost=0;
    end
    radek{2}=jakost;
    vstup(i,:)=radek;
end
```

Program 2 - Přepočítání ukazatelů Certifikát a Jakost

Vstupem jsou hodnoty z proměnné *e*. Příkaz *size* zjistí rozměr vstupní proměnné, uloží do vektoru *Velikost*, z něž lze zjistit počet řádků v proměnné.

Cyklus *for* prochází *vstup* postupně od prvního do posledního řádku. Aktuální řádek se načte do proměnné *radek*. Podmínka *if radek{3}=='A'* zajistí převedení řetězcového formátu logické hodnoty Certifikát na hodnoty {0;1}. Podmínka *if isnumeric(radek{2})* ošetřuje možnost vstupu hodnoty Null pro PPM a převádí na číselnou hodnotu 0; v případě číselného vstupu se hodnota PPM přepočte na procentní vyjádření.

### 5.3.3 Výpočet ukazatele Cenová úroveň

Ukazatel Cenová úroveň představuje procentní vyjádření odchylky průměrné ceny daného dodavatele od průměrné ceny všech dodavatelů daného materiálu.

$$\text{Cenová úroveň} = [(\text{prům. cena dodavatele} / \text{prům. cena všech dodavatelů daného materiálu}) - 1] * 100 [\%]$$

```
VelikostDod=size (dod.data) ;
Velikostf=size (f.data) ;
VelikostMat=size (mat.data) ;
PocRadkuDod=VelikostDod(1) ;
PocRadkuf=Velikostf(1) ;
PocRadkuMat=VelikostMat(1) ;

l=1;
CenUroven=[];
CenovaUrovenM=[];
for i=1:PocRadkuDod
    dodd=dod.data(i) ;

while not (strcmp (f.data{l,1},dod.data{i}))
    l=l+1;
end

while (strcmp (f.data{l,1},dod.data{i}))
    dodmn=f.data{l,7};
    cena=f.data{l,8};
    prumcena=cena/dodmn;
    idmat=f.data{l,2};

    for ff=1:PocRadkuMat
        if (strcmp (mat.data{ff,1},idmat))

            break
        end
    end
end
```

```

end
    pomercen= ((prumcena/mat.data{ff,2})-1)*100;
    pom=CenUroven;
    CenUroven=[pom;pomercen];

    if l==PocRadkuf
    break
    end
    l=l+1;
end

Ukcenuroven=mean(CenUroven);
pom=CenovaUrovenM;
CenovaUrovenM=[pom; dodd, Ukcenuroven];
end

```

### Program 3 - Výpočet ukazatele Cenová úroveň

Program nejprve zjistí velikosti jednotlivých proměnných, určí se tímto počet iterací v jednotlivých cyklech. Proměnná *l* funguje jako čítač, udávající pozici aktuálního řádku ze záznamu objednávek. Cyklus *for i=1:PocRadkuDod* prochází postupně seznam dodavatelů a do proměnné *dodd* se načítá aktuální id dodavatele. Cyklus s podmínkou *while not(strcmp(f.data{l,1},dod.data{i}))* se opakuje, dokud nenarazí na první výskyt daného id dodavatele v záznamech objednávek. Čítač *l* si pamatuje tuto pozici a díky tomu, že jsou záznamy objednávek seřazeny podle id dodavatele, a že čítač *l* se v další iteraci přičítá od hodnoty pozice posledního výskytu, lze značně snížit celkový počet iterací.

Cyklus *while (strcmp(f.data{l,1},dod.data{i}))* vybírá z objednávek pouze ty záznamy, které odpovídají danému dodavateli. Z těchto záznamů se vybere do proměnných dodané množství, cena, průměrná cena jako podíl ceny a dodaného množství, a id materiálu. Cyklus *for ff=1:PocRadkuMat* prochází záznamy materiálů a umožňuje zjistit průměrnou cenu daného materiálu.

Proměnná *pomercen* obsahuje procentní vyjádření odchylky ceny daného dodavatele od průměrné ceny všech dodavatelů daného materiálu. Do vektoru *CenUroven* se přidávají hodnoty proměnné *pomercen* pro každý záznam z objednávek.

Podmínka *if l==PocRadkuf* ukončuje cyklus v případě, že se dojde na poslední řádek záznamu.

Do proměnné *Ukcenuroven* se vypočítá aritmetický průměr hodnot vektoru *CenUroven* a představuje ukazatel Cenové úrovně pro daného dodavatele. Ukazatele se poté přidávají do výstupní matice *CenovaUrovenM*.

### 5.3.4 Výpočet ukazatelů Cenové chování, Dodržování Termínů a Dodržování množství

Pro výpočet následujících ukazatelů se využívá společné množiny dat (záznamy o objednávkách), lze je proto vypočítat v rámci jedné části algoritmu.

Ukazatel Cenové chování představuje procentní vyjádření vývoje ceny u konkrétního dodavatele. Pro zpřesnění výpočtu se použije postup, kdy se ceny dodavatele rozdělí podle data do dvou období a následně se vypočítá průměr daného období.

$$\text{Cenové chování} = (\text{prům. koeficient růstu ceny}) * 100[\%]$$

Ukazatel Dodržování termínu představuje průměrnou odchylku termínu skutečného dodání od termínu dodání v objednávce vyjádřenou v procentech. Kladná hodnota značí zpoždění dodávek, záporná hodnota předčasné dodávky způsobující nárůst skladových zásob.

$$\text{Dodržování termínu} = \left( \frac{\text{Suma}(\text{skutečná doba dodání mat. od dod.} / \text{plánovaná doba dodání mat. od dod.})}{\text{počet objednávek od dod.}} - 1 \right) * 100$$

Ukazatel Dodržování množství představuje průměrnou odchylku skutečně dodaného množství od množství v objednávce.

$$\text{Dodržování množství} = \left( \frac{\text{Suma}(\text{skutečná velikost dodávky mat. od dod.} / \text{plánované množství materiálu oddod.})}{\text{počet objednávek od dod.}} - 1 \right) * 100$$

```
UkDodavka=[];
k=1;

for i=1:PocRadkuDod
    dodd=dod.data(i);
mn=[];
datum=[];
objcena=[];

while not(strcmp(f.data{k,1},dod.data{i}))
    k=k+1;
end

j=1;

while (strcmp(f.data{k,1},dod.data{i}))
    objmn=f.data{k,6}; dodmn=f.data{k,7}; datumobj=f.data{k,3};
```

```

    terminobj=f.data{k,4}; datumdod=f.data{k,5}; cena=f.data{k,8};
    prumcena=cena/dodmn;

    pom=mn; mn=[pom; [objmn dodmn]]; pom=datum;
    datum=[pom; [datenum(datumobj)
    datenum(terminobj) datenum(datumdod)]];
    pom=objcena;
    objcena=[pom; [datenum(datumobj) prumcena]];

    j=j+1;

if k==PocRadkuf
    break
end
    k=k+1;

    end

SkutDoba=datum(:,3)-datum(:,1); PlanDoba=datum(:,2)-datum(:,1);
pomer=SkutDoba./PlanDoba;

polovina=(max(objcena(:,1))+min(objcena(:,1)))/2;

if (j-1)>1
    cena1=[];
    cena2=[];

for ii=1:(j-1)
if objcena(ii,1)<=polovina
pom=cena1;
cena1=[pom; objcena(ii,2)];

else
pom=cena2;
cena2=[pom; objcena(ii,2)];

end

end

prumcena1=sum(cena1)/length(cena1);
prumcena2=sum(cena2)/length(cena2);

UkCenChovani=((prumcena2/prumcena1)-1)*100;

else UkCenChovani=0;

end

UkDodrzTermin=(sum(pomer-1)/(j-1))*100;
UkDodrzMn=abs(sum(mn(:,2)./mn(:,1)-1)/(j-1))*100;
Pom=UkDodavka;
UkDodavka=[Pom; dodd, UkCenChovani, UkDodrzTermin, UkDodrzMn ];

end

```

**Program 4 - Výpočet ukazatelů Cenové chování, Dodržování Termínů a Dodržování množství**



Struktura kódu [Program 4] je obdobná jako v kódu [Program 3], čítač  $k$  určuje pozici aktuálního řádku v záznamu, cyklus  $for\ i=1:PocRadkuDod$  prochází seznam dodavatelů,  $while\ not(strcmp(f.data\{k,1\},dod.data\{i\}))$  vyhledává záznam v objednávkách podle čísla dodavatele,  $while\ (strcmp(f.data\{k,1\},dod.data\{i\}))$  vybírá objednávky daného dodavatele.

Do proměnných se přiřazuje Objednané množství, Dodané množství, Datum objednání, Plánované datum dodání, Skutečné datum dodání, Cena dodávky a Cena položky v dodávce.

Do matice ***mn*** se třídí Objednané množství a Dodané množství; do matice ***datum*** se ukládají Datum objednání, Plánované datum dodání, Skutečné datum dodání v numerickém vyjádření. Matice ***objcena*** obsahuje numericky Datum objednání a Cenu položky v dodávce.

***SkutDoba*** obsahuje rozdíl skutečného data dodání od data objednání, ***PlanDoba*** obsahuje rozdíl plánovaného data dodání od data objednání, proměnná ***pomer*** vyjadřuje odchylky skutečné doby dodání od plánované.

Proměnná ***polovina*** určuje polovinu sledovaného časového intervalu a umožňuje jej rozdělit na dvě období. Podmínka  $if\ (j-1)>1$  zjišťuje, jestli do sledovaného období připadá více než jeden záznam (Pokud je v daném období nalezen pouze jeden záznam, nelze určit koeficient růstu pro Cenové chování. Lze tedy předpokládat, že koeficient je nulový.) V cyklu  $for\ ii=1:(j-1)$  se postupně třídí ceny objednávek podle data buď do prvního nebo druhého období. Proměnné ***prumcena1*** a ***prumcena2*** obsahují průměrnou cenu položek dodávky v příslušném období. Proměnná ***UkCenChovani*** potom obsahuje koeficient růstu ceny u daného dodavatele.

Proměnná ***UkDodrzTermin*** vyjadřuje v procentech odchylku skutečné doby dodání od plánované u daného dodavatele. Proměnná ***UkDodrzMn*** vyjadřuje procentní odchylku skutečně dodaného a objednaného množství u daného dodavatele.

Matice ***UkDodavka*** obsahuje ukazatele ***UkCenChovani***, ***UkDodrzTermin*** a ***UkDodrzMn*** pro všechny analyzované dodavatele.

### 5.3.5 Uložení výsledků analýzy do databáze

Poslední část M-souboru složí jednotlivé matice ukazatelů do jedné matice výstupní. Hodnoty této výstupní matice uloží do databáze.

```
Vystup=[CenovaUrovenMat,UkDodavka(:,2), vstup(:,[2 3]),
UkDodavka(:,[3 4])];

s.DataReturnFormat = 'cellarray'; s.ErrorHandling = 'store';
s.NullNumberRead = 'NaN'; s.NullNumberWrite = 'NaN';
s.NullStringRead = 'null'; s.NullStringWrite = 'null';
s.JDBCDataSourceFile = ''; s.UseRegistryForSources = 'yes';
setdbprefs(s)

conn = database('Dodavatele','','password');

insert(conn,'Ukazatele',{'Dodavatel','Uroven','Chovani','Jakost','Cer
tifikat','Terminy','Mnozstvi'},Vystup)

close(conn)

clear
```

**Program 5 - Uložení výstupu do databáze**

Do matice *Vystup* jsou přiřazeny hodnoty ukazatelů vypočítaných v [Program 2], [Program 3] a [Program 4], ve všech dílčích výstupních maticích jsou hodnoty seřazeny podle id dodavatele, lze je tedy sloučit bez rizika vzniku chyb.

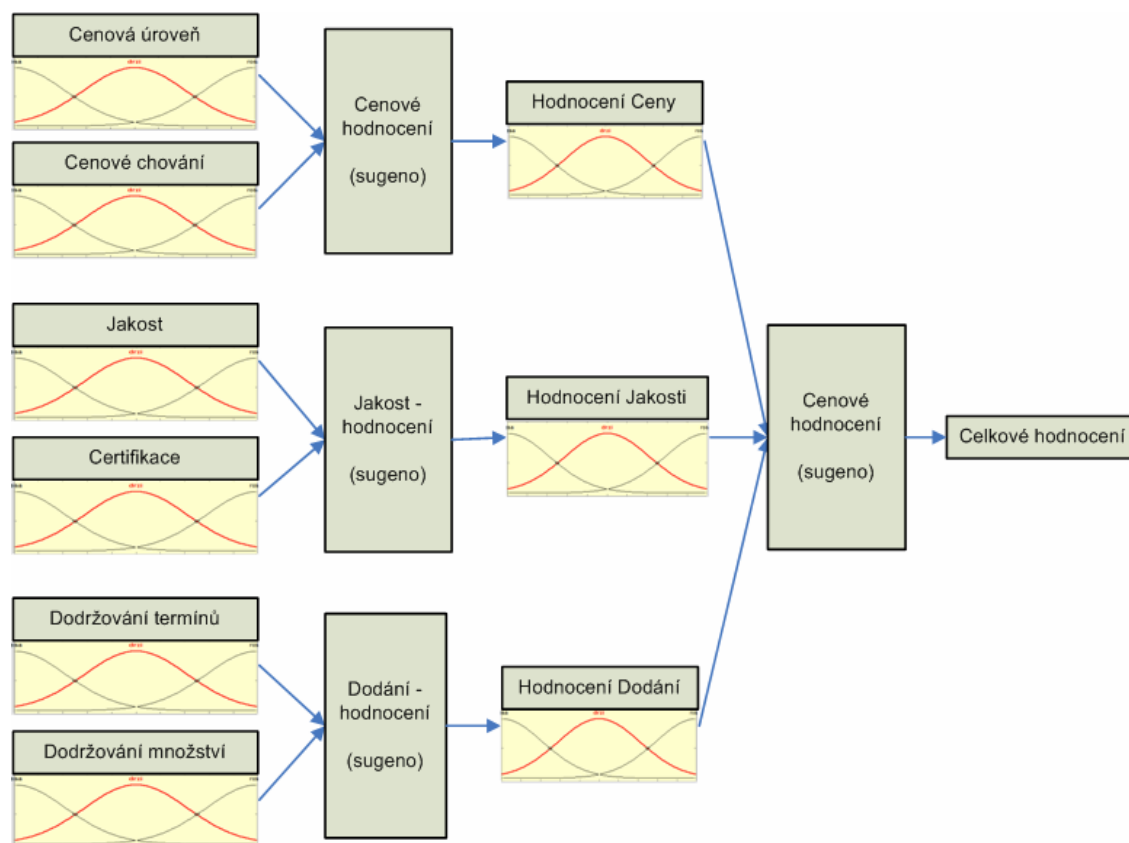
Příkaz *setdbprefs(s)* definuje databázi podle množiny preferencí, nastavené v proměnné *s*. Příkaz *conn* připojuje k databázi Dodavatele. Příkaz *insert* vloží do tabulky Ukazatele záznamy z matice *Vystup*. Příkaz *close(conn)* ukončí spojení s databází. Příkaz *clear* vymaže všechny proměnné z Workspace Matlabu a uvolní paměť.

## 5.4 Aplikace fuzzy logiky

Aplikace je řešena šesti vstupními proměnnými a na výstupu má jednu proměnnou pro celkové hodnocení v rozsahu  $\langle 0;100 \rangle$  bodů.

Aplikaci je vhodné dekomponovat na dílčí fuzzy modely. Dekompozice značně zjednoduší učení systému a usnadní i kontrolu funkčnosti systému pro jeho ladění.

Z hlediska společných vlastností lze aplikaci rozdělit na model pro hodnocení ceny, model pro hodnocení jakosti, model pro hodnocení dodání a model pro celkové hodnocení, jenž bude brát pro vstupní proměnné hodnoty předchozích tří modelů. Vazby mezi fuzzy modely jsou znázorněny na schématu Obrázek 13.



Obrázek 13 - Schéma fuzzy aplikace

Pro vytvoření fuzzy modelů je výhodné zvolit možnost jejich natrénování a vygenerování prostřednictvím ANFIS Editoru v programu Matlab.

Pro učení modelu musí mít editor připravená tréninková data. Při trénování modelu jsou pro tato data hledány funkční závislosti mezi vstupy a výstupem.

Pro vytvoření těchto dat poslouží program vytvořený v prostředí Visual Basic .NET, který pro zvolené vstupní proměnné vygeneruje dle stanoveného bodového hodnocení výstupní hodnoty.

## 5.5 Aplikace v prostředí Visual Basic .NET

Aplikace slouží pro testování vstupů a vytvoření výstupů k naučení ANFIS v prostředí Matlab. Výpočet probíhá na základě bodového hodnocení.

### 5.5.1 Vytvoření transformační matice

Hodnoty pro vytvoření transformační matice se načítají z databáze aplikace do kolekce hodnot ve sloupcích komponent *Listview* a jsou zobrazeny na formuláři. Jsou zde nadefinovány vlastnosti jako interval (počáteční a koncová hodnota, tj. popis transformační matice) a jejich bodové hodnocení.

Transformační matice (Obrázek 14 - Transformační matice) zobrazuje bodové hodnocení pro různé varianty jednotlivých kritérií. Bodové hodnocení je stanoveno dle subjektivního odhadu vedoucího Oddělení nákupu.

Transformační matice																	
Certifikát		Cenové chování			Cenová úroveň			Jakost			Dodržování množství			Dodržování termínů			
Hodnoty	Body	<	:	>	Body	<	:	>	Body	<	:	>	Body	<	:	>	Body
True	100	-100	0	100	-100	0	100	0	2	100	0	20	100	-100	-50	1	
False	50	0	2	95	0	2	95	2	5	95	20	30	90	-50	-40	30	
		2	5	90	2	5	90	5	10	90	30	40	80	-40	-30	60	
		5	10	80	5	10	80	10	20	80	40	50	70	-30	-20	70	
		10	20	70	10	20	70	20	30	70	50	60	50	-20	-10	80	
		20	30	60	20	30	60	30	40	60	60	70	30	-10	-5	90	
		30	40	50	30	40	50	40	50	50	70	100	1	-5	0	100	
		40	50	40	40	50	40	50	70	30				0	5	100	
		50	60	30	50	60	30	70	90	10				5	10	95	
		60	100	1	60	100	1	90	100	0				10	20	90	
														20	30	80	
														30	40	60	
														40	50	50	
														50	100	1	

Obrázek 14 - Transformační matice

V případě logického kritéria Certifikát je varianta True (dodavatel má platný certifikát) ohodnocena 100 body, varianta False 50 body. U číselných kritérií – např. Cenové chování - představují varianty jednotlivé intervaly, např. odchylka ležící v intervalu <-100; 0) je hodnocena 100 body. Ačkoliv se nepředpokládají odchylky mimo interval <-100; 100>, v programu jsou tyto výjimky ošetřeny.

## 5.5.2 Výpočet hodnocení

Program umožňuje načtení hodnot ze souboru (.txt, .dat), který obsahuje cvičná data ve formátu odpovídajícím jednotlivým kritériím. Tato cvičná data jsou vytvořena jako náhodně vygenerovaná čísla s důrazem na správné a rovnoměrné pokrytí intervalu přípustných hodnot (dle transformační matice Obrázek 14)

Po načtení dat se provede výpočet pro jednotlivé řádky vstupu. Testuje se, zda proměnná daného kritéria spadá do intervalu hodnot stanoveného popisnou maticí.

```
Dim radek As String = CStr(soubor.ReadLine())
Dim part() As String = radek.Split(" ")
Dim uk_supplyquantity As Integer = CInt(part(5))

For g As Integer = 0 To ListView5.Items.Count - 1
    If Math.Abs(uk_supplyquantity) >=
CInt(ListView5.Items(g).SubItems(1).Text) And uk_supplyquantity <
CInt(ListView5.Items(g).SubItems(2).Text) Then
        hodnoceni = hodnoceni +
CInt(ListView5.Items(g).SubItems(3).Text) * 0.12
        dodanihodn = dodanihodn +
CInt(ListView5.Items(g).SubItems(3).Text) * 0.4
        dok = 1
    Exit For
End If
Next g
```

**Program 6 - Hodnocení pro dodržování množství**

Část kódu v [Program 6] slouží pro výpočet hodnocení na základě ukazatele Dodržování množství. Stanovuje se celkové hodnocení, do kterého se přičítá suma součinů bodů a váhy všech ukazatelů, a hodnocení dodání, do kterého se přičítá vážené hodnocení ukazatelů pro dodávku.

Do proměnné *radek* se načítá aktuální řádek ze souboru a rozdělí se do pole *part()* na jednotlivé hodnoty. Proměnná *uk\_supplyquantity* obsahuje hodnotu ukazatele Dodržování množství.

Cyklus *For* prochází jednotlivé řádky v kolekci *ListView5*, které obsahují hodnoty transformační matice. Podmínka *If* testuje, jestli proměnná *uk\_supplyquantity* spadá do intervalu definovaném hodnotami v položkách *SubItems(1)* a *SubItems(2)* v kolekci *ListView5.Items*, jestliže do tohoto intervalu spadá, je k celkovému hodnocení připočten počet bodů z *SubItems(3)* vynásobený váhou 12% a k hodnocení dodání se připočte počet bodů s váhou 40%. Cyklus je poté ukončen a pokračuje se hodnocením dalších pěti ukazatelů pro aktuální řádek.

Výsledky výpočtu se vloží do Výstupní matice (Obrázek 15 - Výstupní matice). Na základě podmínek výstupní matice se přiřadí k vypočítané hodnotě i lingvistické hodnocení.

Výstupní matice											
	Váha: 0,24	Váha: 0,16	Váha: 0,24	Váha: 0,06	Váha: 0,18	Váha: 0,12					
Cenová úroveň	Cenové chov...	Jakost	Certifikát	Dodrž. termínů	Dodrž. množství	Známka za cenu	Známka za ja...	Známka za dodání	Hodnocení	Slovní hodnocení	
-66	-1	90	1	-25	16	100	20	82	70,6	nevyhovující	
29	8	60	0	85	29	68	34	36,6	48,38	nevyhovující	
-41	14	33	1	36	95	88	68	36,4	66,52	nevyhovující	
88	17	8	0	15	17	28,6	82	94	64,24	nevyhovující	
-1	0	0	1	-1	5	98	100	100	99,2	vyhovující	
-2	5	5	1	-3	4	92	92	100	94,4	vyhovující	
-1	7	8	1	5	8	92	92	97	93,5	vyhovující	
0	1	9	0	2	7	95	82	100	92,6	vyhovující	
-2	3	1	1	-3	2	96	100	100	98,4	vyhovující	
-1	5	7	1	2	2	92	92	100	94,4	vyhovující	
1	2	3	0	1	3	93	86	100	93	vyhovující	
2	9	5	1	0	9	86	92	100	92	vyhovující	
3	7	9	1	-1	12	86	92	100	92	vyhovující	

Načíst      Načtený soubor: D:\diplomka\fuzzy.txt      Uložit Hodnocení Ceny      Uložit Hodnocení jakosti      Uložit Hodnocení dodání      Uložit Celkové hodnocení

Obrázek 15 - Výstupní matice

Program umožňuje uložit hodnoty z výstupní matice v požadovaném formátu do datových souborů *cenatraindata.dat*, *jakosttraindata.dat*, *dodavkatraindata.dat* a *celktraindata.dat*.

Například soubor *cenatraindata.dat* obsahuje matici

$$\begin{pmatrix} cen.uroven & cen.chovani & hodnoceni.ceny \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{pmatrix}.$$

Tyto soubory obsahují data pro učení jednotlivých fuzzy modelů v ANFIS editoru v Matlabu.

## 5.6 Aplikace v Matlabu

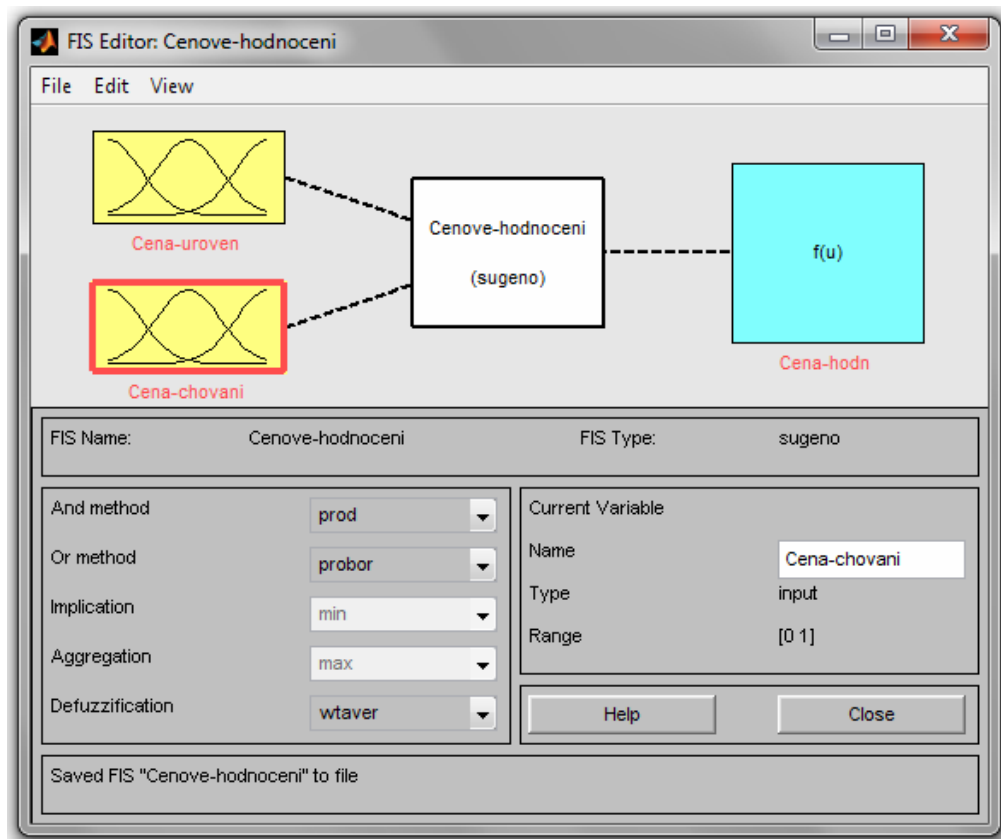
Pro vytvoření Fuzzy modelu pro hodnocení dodavatelů volím využití Fuzzy Logic Toolboxu v programu MATLAB. Toolbox lze spustit v příkazovém okně příkazem *fuzzy*. Jak bylo uvedeno výše, aplikace sestává z celkem čtyř dílčích modelů. Tyto modely se vytvoří naučením v prostředí ANFIS Editoru.

### 5.6.1 Trénování modelu v ANFIS editoru

Základním předpokladem pro využití trénování v ANFIS editoru pro modelování fuzzy systému je existence souboru vstupních/výstupních dat vhodných pro modelování. Potřebné soubory dat byly vytvořeny prostřednictvím aplikace naprogramované ve VB.NET.

ANFIS Editor pracuje s fuzzy modelem typu Sugeno. V této kapitole se budu zabývat trénováním modelu pro *Cenové hodnocení*. Vytváření ostatních modelů probíhá obdobným způsobem, pouze s použitím odlišných souborů tréninkových dat.

Po spuštění okna FIS editoru je třeba navolit v menu *File/New FIS/Sugeno* pro model typu Sugeno. Vytvoří se jednoduchý model s jednou vstupní a jednou výstupní proměnnou. Příkazem v menu *Edit/Add variable.../Input* se do modelu přidá další vstupní proměnná. Názvy vstupních proměnných je vhodné přejmenovat na *Cena-uroven* a *Cena-chovani*, název výstupní proměnné na *Cena-hodn*.



Obrázek 16 - FIS editor - Hodnocení ceny

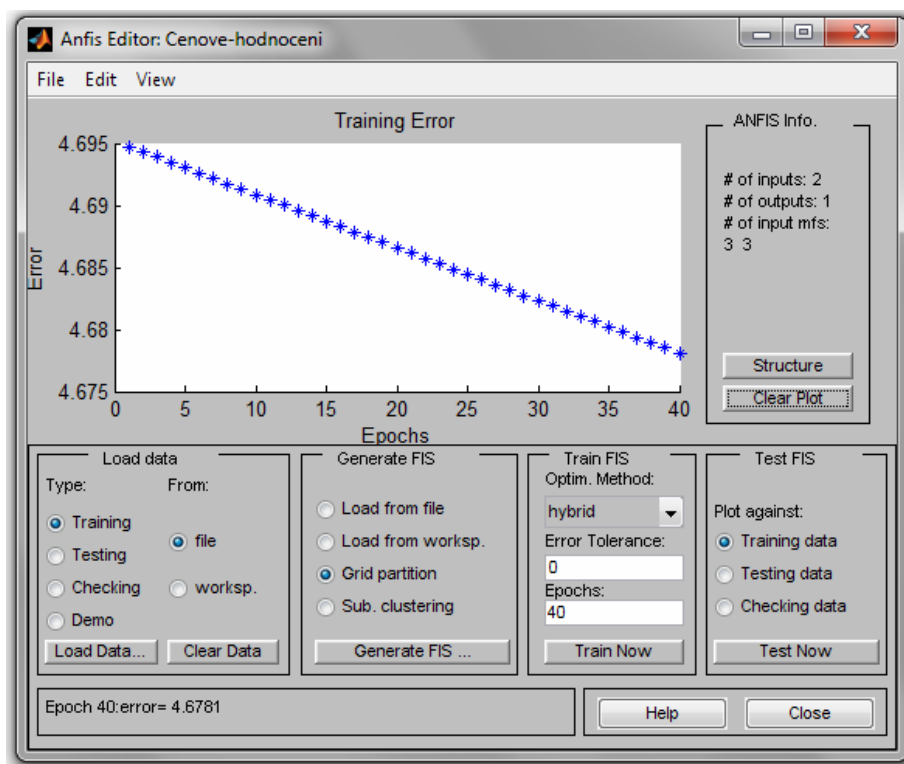
Příkazem v menu *Edit/Anfis...* se spustí ANFIS editor. Pro načtení dat je nutné zvolit v *Load data* typ dat *Type: Training* a načítání ze souboru *From: File*. Pro Cenové hodnocení je vytvořen soubor dat *cenatrainedata.dat*.

Pro generování FIS struktury volím metodu *Grid Partition*, typ funkce příslušnosti *gaussmf* tvaru Gaussovy křivky, počet funkcí [3 3] a lineární typ výstupní funkce.

Pro naučení FIS vybírám hybridní optimalizační metodu (kombinující metodu zpětného šíření chyby s metodou nejmenších čtverců pro větší přesnost učení) s nulovou tolerancí chyby a počet epoch 40 iterací (výchozí počet epoch je 3).

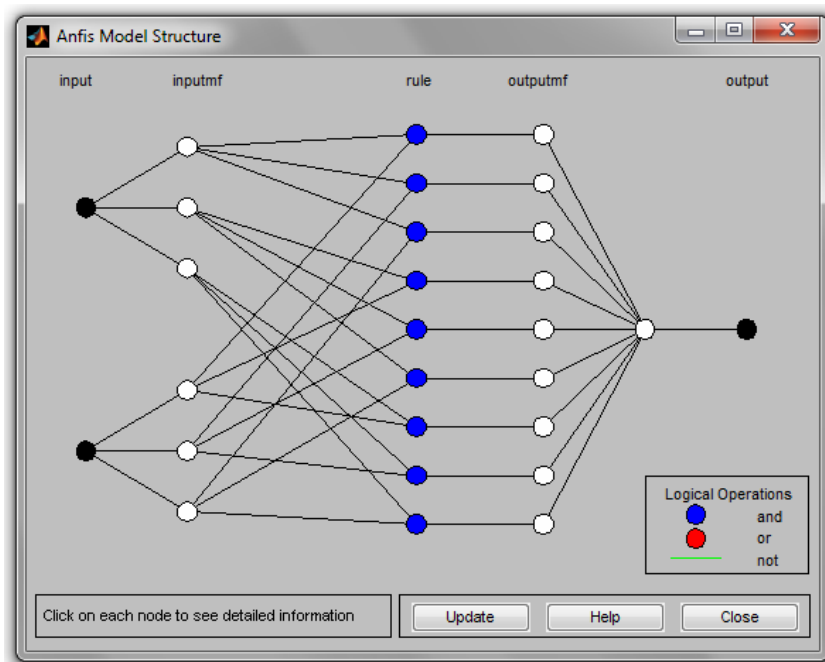
Učení probíhá dokud není dosaženo buď úrovně tolerance chyby nebo konečného počtu epoch. V grafu na obrázku je znázorněn průběh minimalizace chyby při trénování.





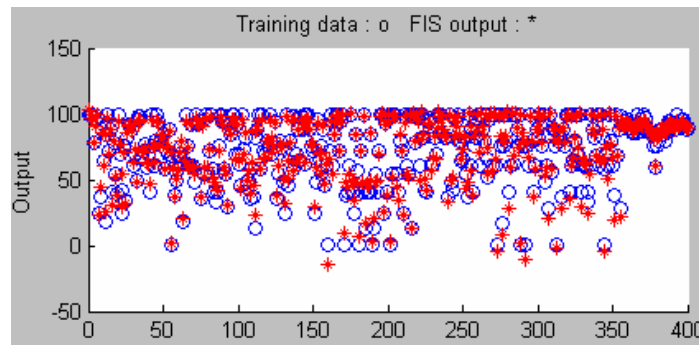
**Obrázek 17 - ANFIS editor - trénování modelu**

Natrénováním se vytvoří struktura modelu, funkce členství, pravidla a výstupní funkce. Struktura modelu lze zobrazit tlačítkem *Structure* v *ANFIS Info*.



**Obrázek 18 - Struktura modelu**

Pro otestování modelu lze v grafu zobrazit výstupní hodnoty FIS oproti výstupům ze souboru tréninkových dat. Vzhledem k tomu, že data pro trénink se vytvářela bodováním, dají se předpokládat určité odchylky v zobrazených hodnotách, tyto odchylky však nejsou nijak velké.



Obrázek 19 - Graf výstupních hodnot modelu a tréninkových dat

Obdobným způsobem probíhá učení dalších modelů.

Model *Hodnocení jakosti* má vstupní proměnné *Jakost* a *Certifikat*, výstupní proměnnou *Jakost-hodn*, pro trénink využívá soubor *jakosttraindata.dat*. Vygenerovaný typ funkce příslušnosti je *trimf* tvaru  $\Lambda$ , počet funkcí [3 2] a konstantní výstupní funkce.

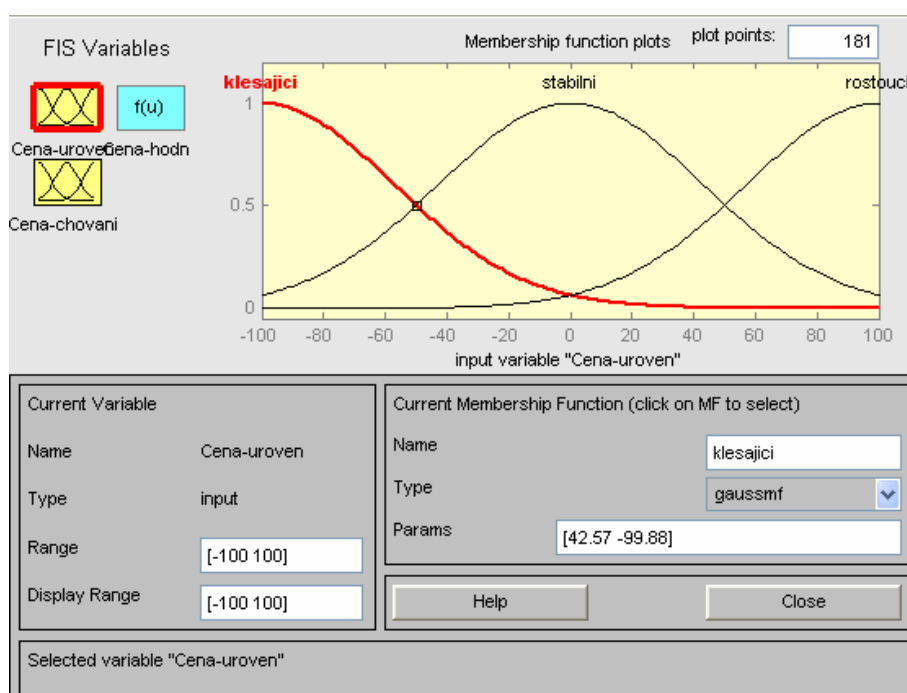
Model *Hodnocení dodávky* má vstupní proměnné *Terminy* a *Mnozstvi*, výstupní proměnnou *Dodani-hodn*, pro trénink využívá soubor *dodavkatraindata.dat*. Vygenerovaný typ funkce příslušnosti je *gaussmf*, počet funkcí [3 3] a výstupní funkce je lineární.

Model *Celkové hodnocení* má tři vstupní proměnné *Cena-hodn*, *Jakost-hodn* a *Dodani-hodn*, výstupní proměnnou *Dodani-hodn*, pro trénink využívá soubor *celktraindata.dat*. Vygenerovaný typ funkce příslušnosti je *gaussmf*, počet funkcí [3 3] a výstupní funkce je lineární.

## 5.6.2 Model Hodnocení ceny

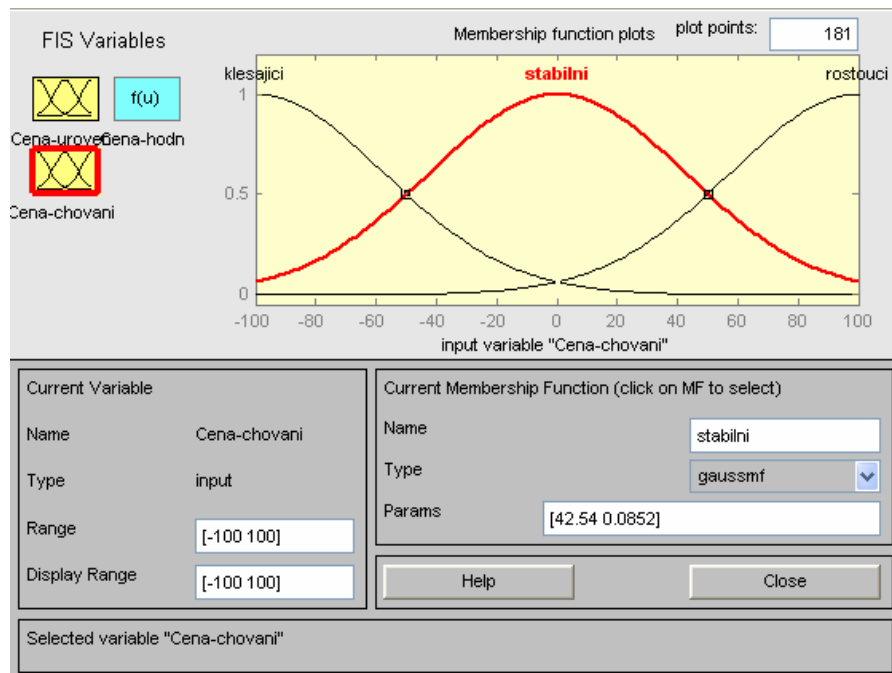
Parametry modelu byly vygenerovány ANFIS editorem. Ve FIS editoru lze tyto parametry prohlížet, případně měnit. Příkaz *mfedit('Cenove-hodnoceni.fis')* umožňuje zobrazení a nastavení funkce členství vstupních proměnných a parametrů výstupu.

Proměnná *Cena-uroven* má rozsah vstupů v intervalu  $<-100; 100>$ , tři funkce členství tvaru gaussovy křivky popisující úroveň jako klesající, stabilní nebo rostoucí.



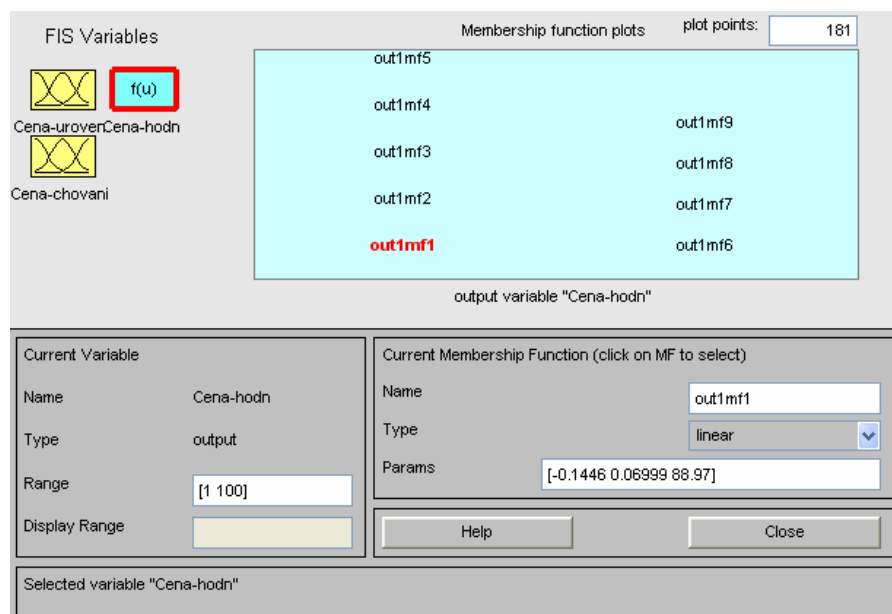
Obrázek 20 - Membership function editor - Cenová úroveň

Proměnná *Cena-chovani* má obdobné parametry jako *Cena-uroven*. Funkce členství pro stabilní cenové chování má maximální hodnotu přibližně v bodě 0.



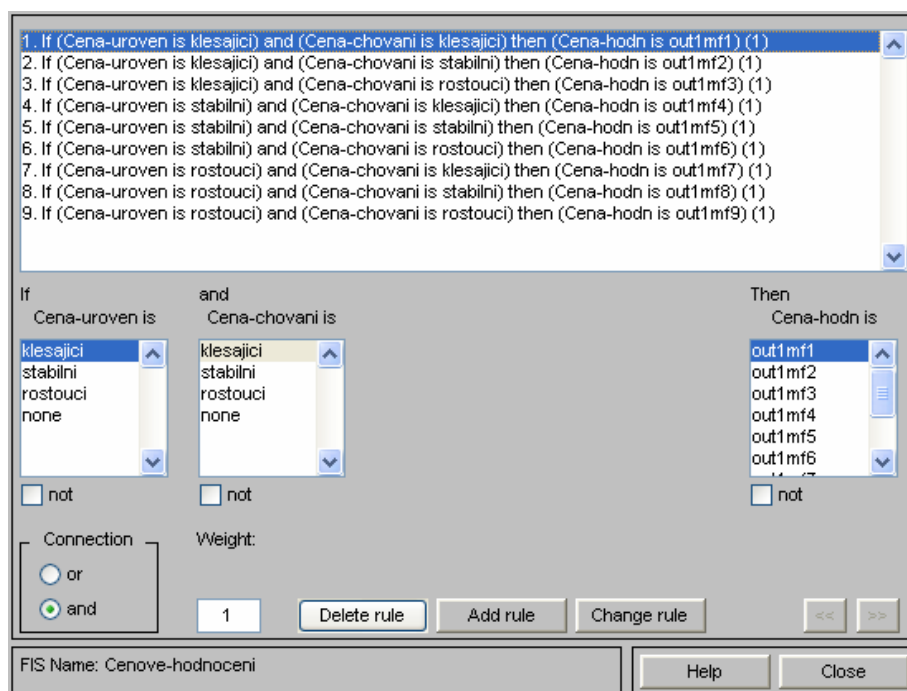
Obrázek 21 - Membership function editor - Cenové chování

Výstupní proměnná *Cena-hodn* má rozsah hodnot v intervalu  $<0; 100$ ), a devět výstupních lineární funkcí.



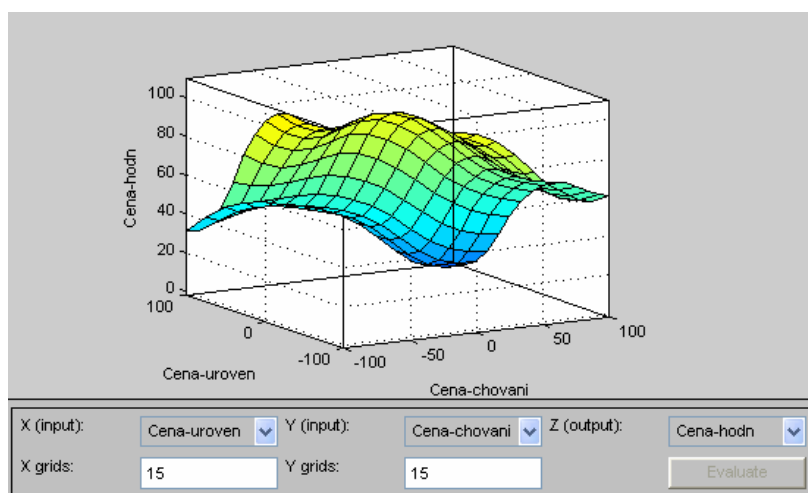
Obrázek 22 - Membership function editor – Výstup

Příkaz *ruleedit('Cenove-hodnoceni.fis')* umožňuje zobrazení a práci s fuzzy pravidly. Vygenerovalo se 9 pravidel pro model *Hodnocení ceny*. Na základě příslušnosti vstupních hodnot do funkcí členství je pro výstup přiřazena patřičná výstupní lineární funkce. V případě klesající cenové úrovně a stabilního cenového chování je to funkce *out1mf2*.



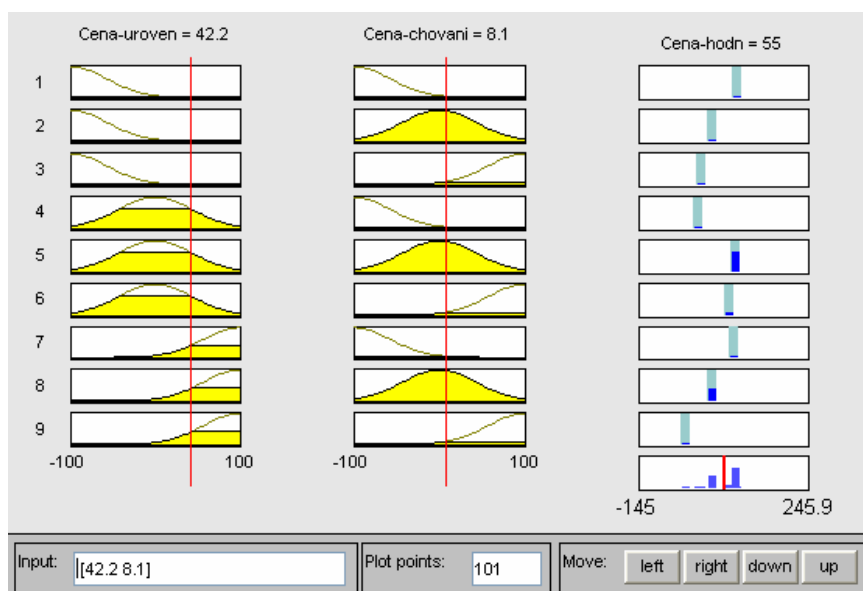
Obrázek 23 - Rule Editor - Hodnocení ceny

Příkaz *surfview('Cenove-hodnoceni.fis')* umožňuje grafické znázornění závislosti vstupních proměnných a výstupní proměnné.



Obrázek 24 - Surface viewer - Hodnocení ceny

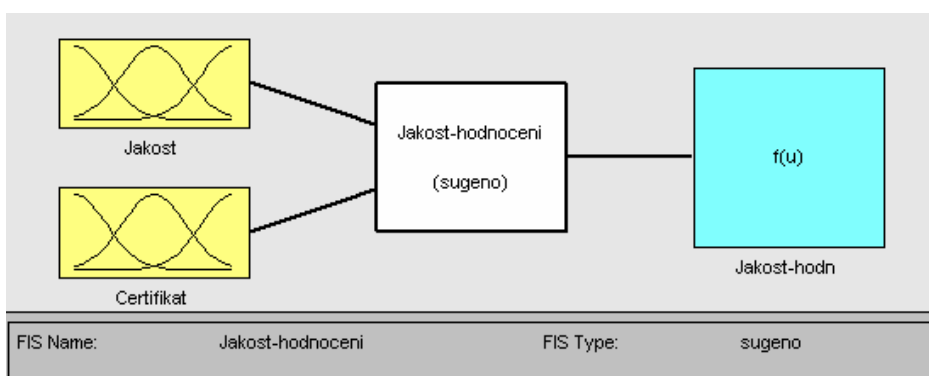
Příkaz *ruleview('Cenove-hodnoceni.fis')* umožňuje prohlížení výstupu *Cena-hodn* na vstupech *Cena-uroven* a *Cena-chovani*. Při hodnotách vstupů [42.2 8.1] (tj. situace, kdy je průměrná cena dodavatele o 42% vyšší než u ostatních a ve sledovaném období cena vzrostla o 8%) je hodnota výstupu 55, což představuje vážený průměr funkčních hodnot výstupních funkcí.



Obrázek 25 - Rule Viewer - Hodnocení Ceny

### 5.6.3 Model Hodnocení jakosti

Příkaz *fuzzy('Jakost-hodnoceni.fis')* otevře model pro úpravy ve FIS editoru. Model je typu Sugeno a má dvě vstupní *Jakost* a *Certifikat* a jednu výstupní proměnnou *Jakost-hodn*.

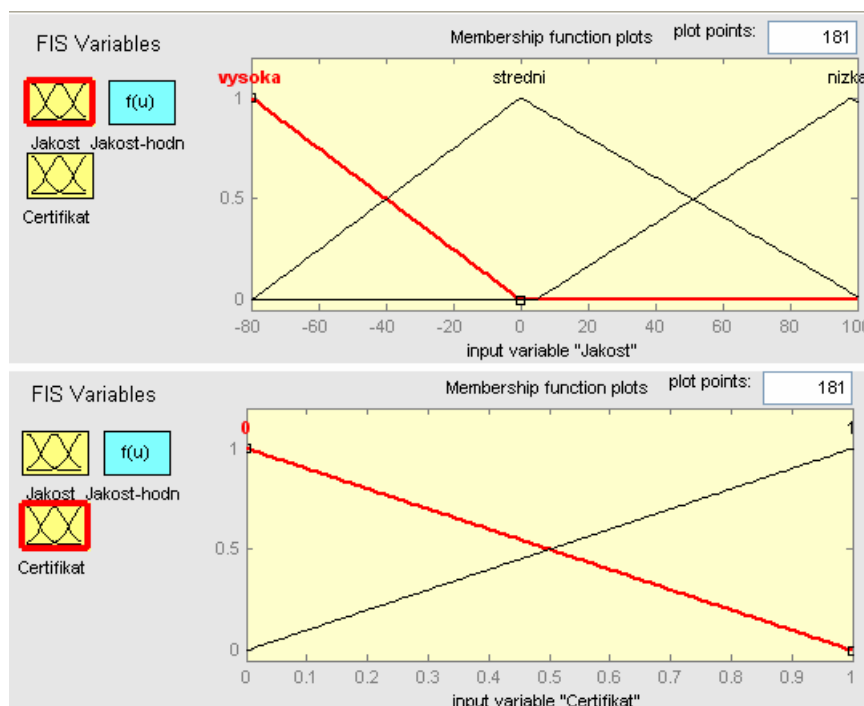


Obrázek 26 - FIS editor - Hodnocení jakosti

Příkaz *mfedit('Jakost -hodnoceni.fis')* umožňuje zobrazení a nastavení funkce členství vstupních proměnných a parametrů výstupu pro model Hodnocení jakosti.

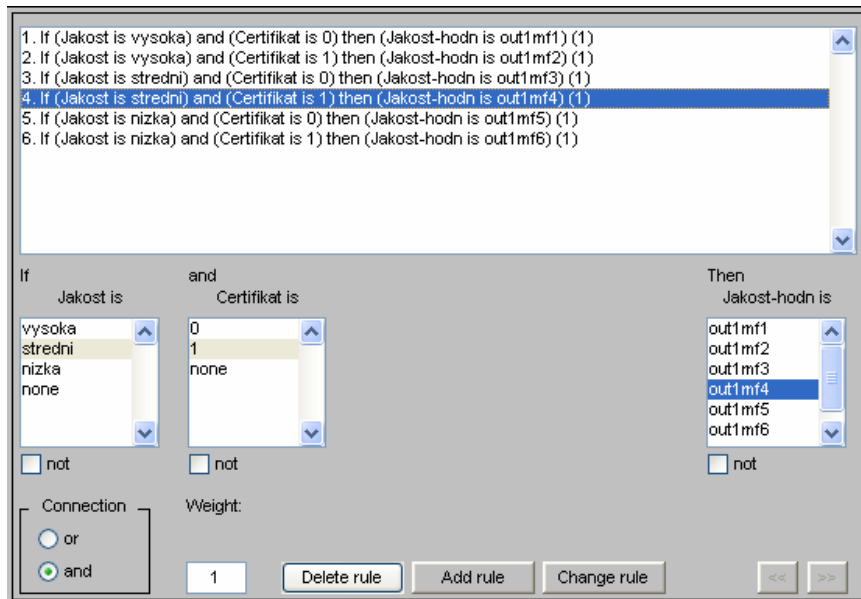
Proměnná *Jakost* má tři funkce členství definované pro vysokou, střední a nízkou jakost. Rozsah pro vstupní hodnoty je vygenerován v intervalu  $\langle -80; 100 \rangle$ , reálně však mohou nabývat pouze hodnot  $\langle 0; 100 \rangle$ . Funkce jsou standardního typu  $\Lambda$ .

Proměnná *Certifikát* má dvě funkce členství pro vstupy v intervalu  $\langle 0; 1 \rangle$ . Reálně mohou být na vstupu pouze hodnoty  $\{0; 1\}$ .



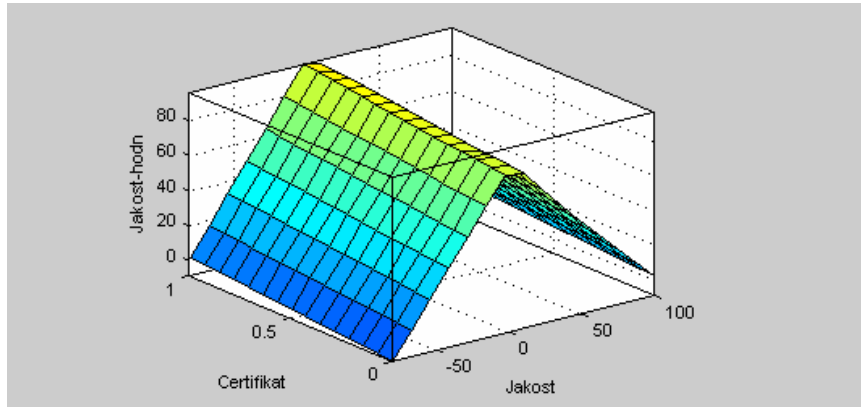
Obrázek 27 - Membership function editor - Hodnocení jakosti

Příkaz *ruleedit('Jakost -hodnoceni.fis')* umožňuje definování a úpravu fuzzy pravidel. Pro model je definováno šest pravidel. Například v případě střední jakosti a platného certifikátu je výstup hodnocen hodnotou funkce *out1mf4*.



Obrázek 28 - Rule Editor - Hodnocení jakosti

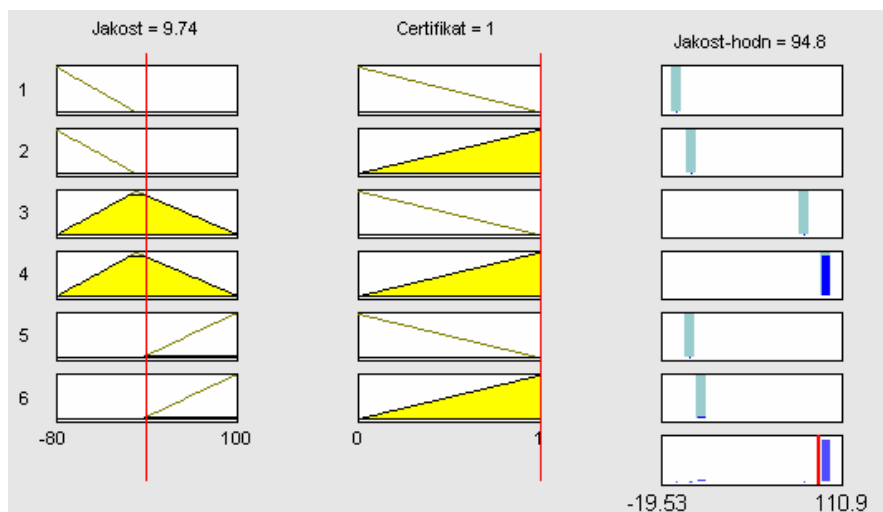
Příkaz *surfview('Jakost-hodnoceni.fis')* zobrazí plochu závislosti výstupu *Jakost-hodn* na hodnotách vstupních proměnných *Jakost* a *Certifikat*. Tvar povrchu závislosti je ovlivněn hlavně hodnotami proměnné *Certifikat*.



Obrázek 29 - Surface Viewer - Hodnocení jakosti

Příkaz *ruleview('Jakost-hodnoceni.fis')* umožní prohlížení hodnoty výstupu při zadaných vstupních hodnotách. Při hodnotách vstupů [9.74 1], tj. při odchylce jakosti 9,7% a platném certifikátu, je hodnota výstupu (hodnocení jakosti) 94,8.

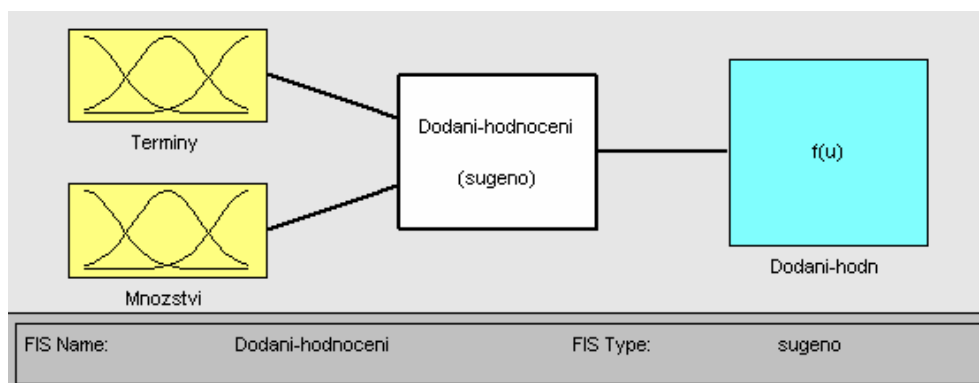




Obrázek 30 - Rule Viewer - Hodnocení jakosti

#### 5.6.4 Model Hodnocení dodání

Příkaz *fuzzy* ('*Dodani-hodnoceni.fis*') otevře model pro úpravy ve FIS editoru. Model je typu Sugeno a má dvě vstupní proměnné *Terminy* a *Mnozstvi* a jednu výstupní proměnnou *Dodani-hodn*.

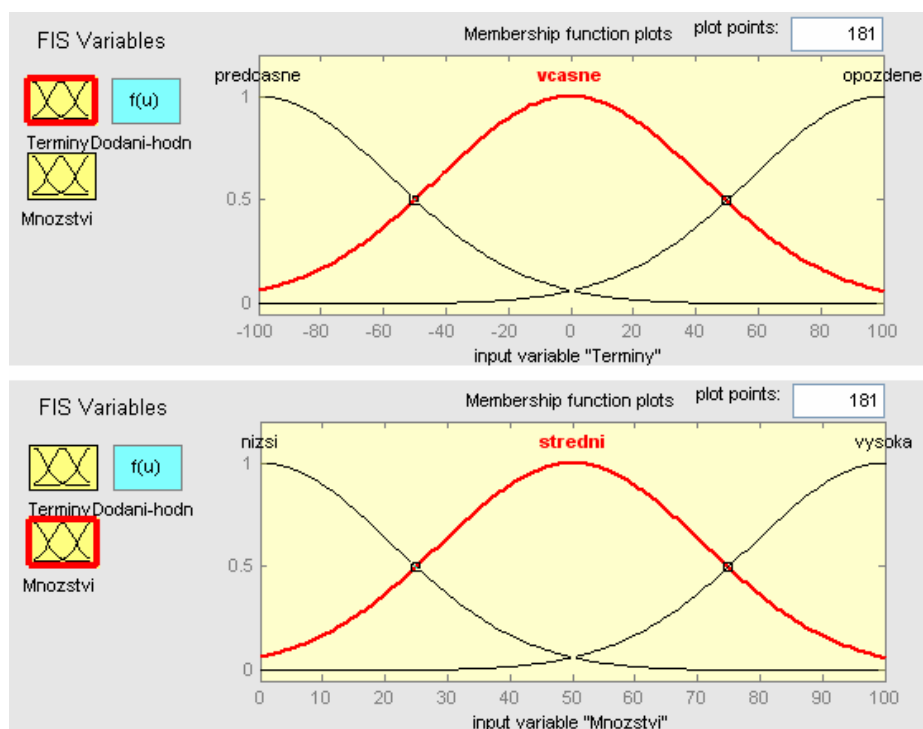


Obrázek 31 - FIS editor - Hodnocení dodání

Příkaz *mfedit* ('*Dodani-hodnoceni.fis*') umožňuje zobrazení a nastavení funkcí členství vstupních proměnných a parametrů výstupu pro model Hodnocení dodání.

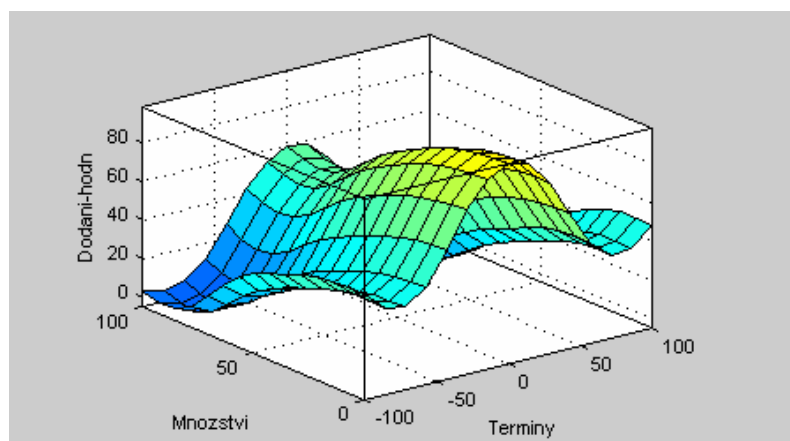
Proměnná *Terminy* má tři funkce členství definované pro předčasné, včasné a opožděné dodání. Rozsah pro vstupní hodnoty je vygenerován v intervalu  $\langle -100; 100 \rangle$ . Funkce jsou typu Gaussovy křivky.

Proměnná *Mnozstvi* má tři funkce členství typu Gaussovy křivky pro vstupy v intervalu  $\langle 0; 100 \rangle$ .



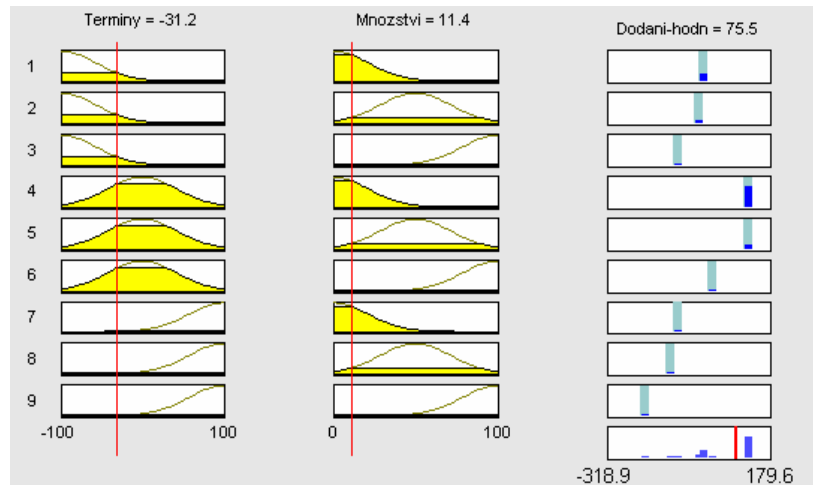
Obrázek 32 - Membership function editor - Hodnocení dodání

Příkaz `surfview('Dodani-hodnoceni.fis')` zobrazí plochu závislosti výstupu *Dodani-hodn* na hodnotách vstupních proměnných *Terminy* a *Mnozstvi*.



Obrázek 33 - Surface Viewer - Hodnocení dodání

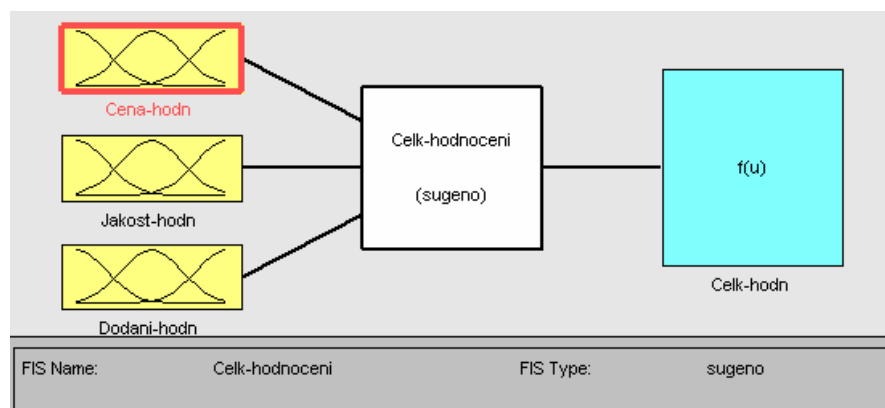
Příkaz *ruleview('Dodani-hodnoceni.fis')* umožní prohlížení hodnoty výstupu při zadaných vstupních hodnotách. Při hodnotách vstupů [-31.2 11.8], tj. při dodání o 31.2% času dříve a odchylce v dodaném množství  $\pm 11.4\%$  je hodnota výstupu (hodnocení dodání) 75.5.



Obrázek 34 - Rule Viewer - Hodnocení dodání

### 5.6.5 Model Celkové hodnocení

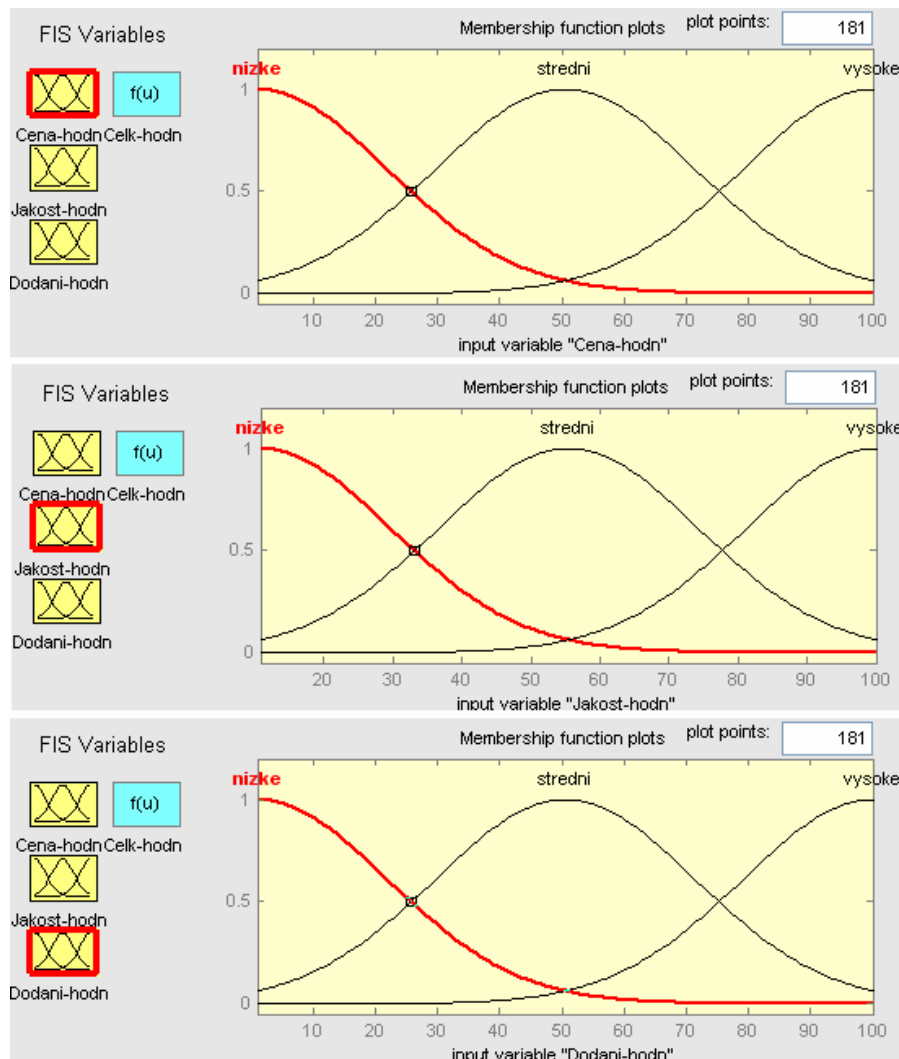
Příkaz *fuzzy('Celk-hodnoceni.fis')* otevře model pro úpravy ve FIS editoru. Model je typu Sugeno a má tři vstupní proměnné *Cena-hodn*, *Jakost-hodn* a *Dodani-hodn* a jednu výstupní proměnnou *Celk-hodn*.



Obrázek 35 - FIS editor - Celkové hodnocení

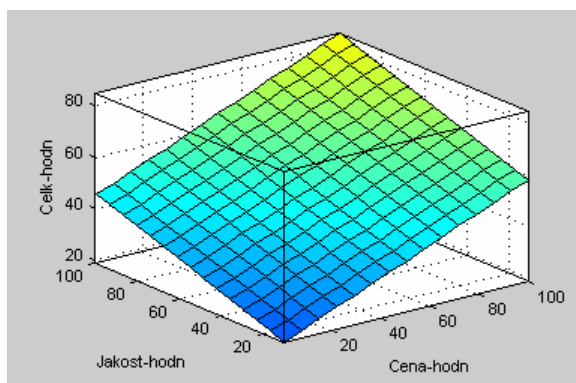
Příkaz `mfedit('Celk-hodnoceni.fis')` umožňuje zobrazení a nastavení funkcí členství vstupních proměnných a parametrů výstupu pro model Celkové hodnocení

Každá vstupní proměnná má tři funkce členství definované pro nízké, střední a vysoké hodnocení. Rozsah pro vstupní hodnoty je vygenerován v intervalu  $\langle 0; 100 \rangle$ . Funkce jsou typu Gaussovy křivky.



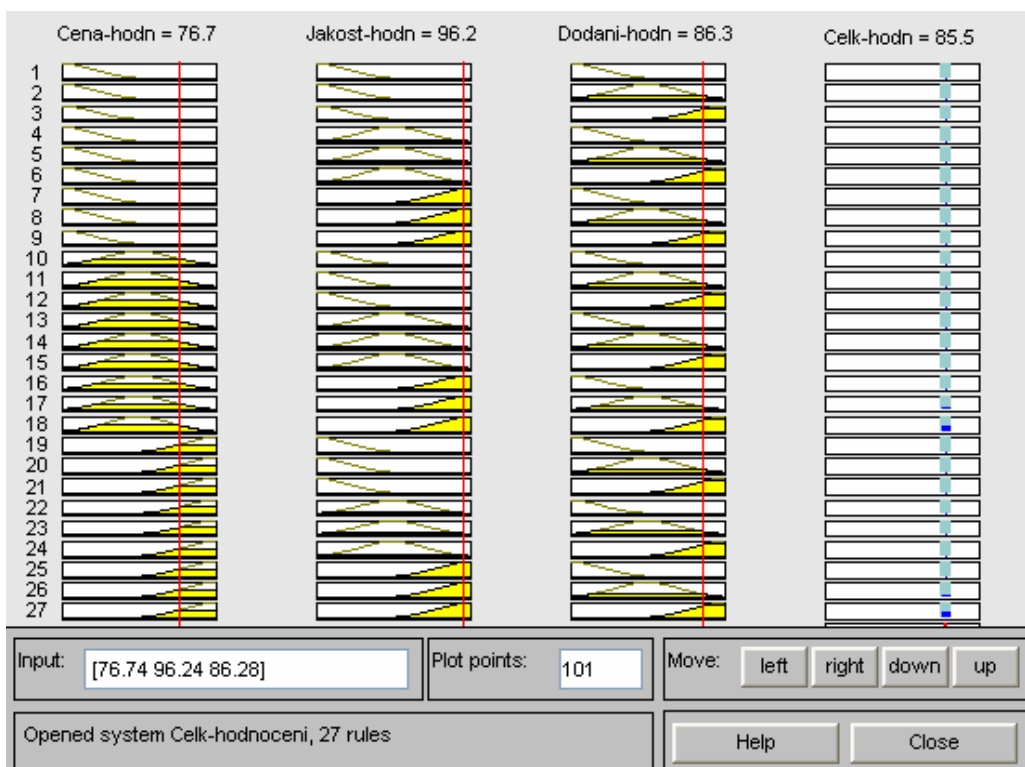
**Obrázek 36 - Membership function editor - Celkové hodnocení**

Příkaz `surfview('Celk-hodnoceni.fis')` zobrazí plochu závislosti výstupu *Celk-hodn* na hodnotách vstupních proměnných *Jakost-hodn* a *Cena-hodn*. Velikost výstupní hodnoty je dána součtem vah jednotlivých proměnných, to se odráží ve tvaru plochy.



Obrázek 37 - Surface Viewer - Celkové hodnocení

Příkaz *ruleview('Celk-hodnoceni.fis')* umožní prohlížení hodnoty výstupu při zadaných vstupních hodnotách. Při hodnotách vstupů [76.7 96.2 86.28], tj. při hodnocení ceny 76,7 bodů, hodnocení jakosti 96,2 bodů a hodnocení dodání 86,3 bodů, je hodnota výstupu (celkové hodnocení) 85.5.



Obrázek 38 - Rule Viewer - Celkové hodnocení

## 5.7 Propojení fuzzy modelů

Za účelem hodnocení dodavatelů je v Matlabu vytvořen M-soubor *fuzzyhodn.m*. Úloha sestává z načtení záznamu hodnot ukazatelů z tabulky *Ukazatele* v databázi *Dodavatele.mdb*, načtení parametrů fuzzy modelů, postupného automatického dosazování ukazatelů v podobě vstupních proměnných do modelů, uložení jednotlivých hodnocení do výstupní matice a konečně uložení výstupní matice do tabulky *Vystup* v databázi.

### 5.7.1 Načtení dat

První část M-souboru (Program 7) definuje připojení k databázi a načítá prostřednictvím SQL dotazů požadovaná data do matic ve Workspace programu. Dotaz *'SELECT ALL Dodavatel,Uroven,Chovani,Jakost,Certifikat,Terminy,Mnozstvi FROM Ukazatele'* vybere z tabulky hodnoty ukazatelů pro jednotlivé dodavatele.

```
e = exec(conn, 'SELECT ALL
Dodavatel,Uroven,Chovani,Jakost,Certifikat,Terminy,Mnozstvi FROM
Ukazatele');
e = fetch(e);
close(e)
```

Program 7 - Načtení dat

Kód načítá všechny záznamy z tabulky *Ukazatele* v databázi a ukládá je do matice *e*.

### 5.7.2 Vyhodnocování dodavatelů

Tato část kódu (Program 8) načítá parametry fuzzy modelů, postupně dosazuje hodnoty vstupních proměnných a vypočítává hodnocení. Dle výše celkového hodnocení je dodavatel zařazen do skupiny 1, 2 nebo 3. Hodnocení dodavatele je uloženo do výstupní matice.

```

VystupMat=[];
DataHodn=[];

Cen = readfis('Cenove-hodnoceni.fis');
Jak = readfis('Jakost-hodnoceni.fis');
Dod = readfis('Dodani-hodnoceni.fis');
Celk = readfis('Celk-hodnoceni.fis');

vstup = e.data(:, [2 3 4 5 6 7]);
id= e.data(:,1);
Velikost=size(vstup);
PocRadku=Velikost(1);

for i=1:PocRadku
    radek=vstup(i,:);

    vstupCena=[(radek{1}) (radek{2})];
    vstupJakost=[(radek{3}) (radek{4})];
    vstupDodani=[(radek{5}) (radek{6})];

    hodnoceniCena=evalfis(vstupCena, Cen);
    hodnoceniJakost=evalfis(vstupJakost, Jak);
    hodnoceniDodani=evalfis(vstupDodani, Dod);

    vstupCelk=[hodnoceniCena hodnoceniJakost hodnoceniDodani];
    hodnoceniCelk=evalfis(vstupCelk, Celk);

    if hodnoceniCelk <75
        skupina=3;
    elseif hodnoceniCelk<90
        skupina=2;
    else
        skupina=1;
    end

    Vystup=[num2cell(hodnoceniCena) hodnoceniJakost hodnoceniDodani
    hodnoceniCelk skupina];
    Predch=VystupMat;

    VystupMat=[Predch; Vystup];
    Pom=DataHodn;
    DataHodn = [Pom; hodnoceniCelk];
end

Uloz=[id VystupMat];

```

#### Program 8 - Vyhodnocení dodavatelů

Příkaz *readfis* načítá do proměnných *Cen*, *Jak*, *Dod* a *Celk* parametry jednotlivých vytvořených fuzzy modelů: *Cenove-hodnoceni.fis*, *Jakost-hodnoceni.fis*, *Dodani-hodnoceni.fis* a *Celk-hodnoceni.fis*.

Do matice *vstup* se načítají z matice *e* sloupce Uroven, Chovani, Jakost, Certifikat, Termíny, Mnozství. Do vektoru *id* se načítá sloupec Dodavatel. Zjistí se

velikost matice *vstup* a počet řádků v matici. Cyklus *for i=1:PocRadku* prochází postupně vstupní matici a do vektoru *radek* načítá aktuální řádek vstupu. Do vektoru *vstupCena* se přiřadí ukazatele, které se použijí jako vstupní proměnné pro Cenové hodnocení; do vektoru *vstupJakost* vstupní proměnné pro Hodnocení jakosti a do vektoru *vstupDodani* vstupní proměnné pro Hodnocení dodávky.

Příkaz *evalfis* dosadí vstupní proměnné do dané FIS a výstupní hodnotu přiřadí do proměnných *hodnoceniCena*, *hodnoceniJakost* a *hodnoceniDodani*. Tyto tři proměnné se posléze dosazují jako vstupní proměnné do FIS pro Celkové hodnocení.

Podmínky v těle *if...end* rozdělují dodavatele podle celkového hodnocení do tří skupin. Dodavatel hodnocený více než 90 body je označen jako 1- Vyhovující, méně než 90 a více nebo rovno 75 bodům jako 2 - Podmínečně vyhovující a méně než 75 body jako 3 - Nevyhovující.

Do vektoru *Vystup* se ukládají proměnné *hodnoceniCena*, *hodnoceniJakost*, *hodnoceniDodani*, *hodnoceniCelk* a skupina. Vektor *Vystup* se v cyklu vkládá jako řádky matice *VystupMat*, která slouží jako výstupní matice hodnocení.

Vektor *DataHodn* obsahuje data pro grafické znázornění hodnocení. Matice *Uloz* obsahuje hodnoty pro uložení do tabulky databáze.

### 5.7.3 Grafické znázornění hodnocení

Tato část kódu umožňuje zobrazit celkové hodnocení dodavatelů graficky v podobě sloupcového grafu.

```
prubeh=15;
ee=linspace(0,prubeh+1,1000);

v=DataHodn(1:prubeh);
rozsah= [min(v)-10 100];

bar(v), title('Hodnocení dodavatelů'), xlabel('Číslo dodavatele (SAP)'),
ylabel('Celkové hodnocení [%]'), set(gca, 'YLimMode', 'auto', 'YLim', rozsah, 'XLimMode', 'auto', 'XTickLabel', Uloz(:,1)); grid off;
hold on;
plot(ee, 90, 'g-');
hold on;
plot(ee, 75, 'r-');
hold off;
```

Program 9 - Grafická reprezentace hodnocení



Proměnná *prubeh* zjišťuje počet vykreslovaných hodnot. Pro přehlednost by bylo vhodné vykreslovat 15 hodnot. Proměnná *ee* slouží jako atribut pro vykreslení hraničních přímk pro oddělení vyhovujících, podmíněně vyhovujících a nevyhovujících dodavatelů. Vektor *v* obsahuje hodnoty celkového hodnocení pro vynesení do grafu. Proměnná *rozsah* určuje minimální a maximální hodnotu mezi na y-ové ose grafu, slouží pro přehlednější vykreslení.

Příkaz *bar* vytvoří sloupcový graf hodnot vektoru *v*. Příkazy *title*, *xlabel* a *ylabel* pojmenují graf a osy. Příkaz *set* s parametrem '*XTickLabel*' pojmenuje hodnoty na ose *x* jednotlivými čísly dodavatelů. Parametr '*XLimMode*', '*auto*' nastaví automatické popisování hodnot na ose *x* podle dat pro '*XTickLabel*'. Parametr '*YLim*' nastaví meze osy *y*. Příkaz *hold on* umožní do okna grafu přikreslit další funkce. Příkazy *plot* vykreslí přímky pro hranice a příkaz *hold off* zruší přikreslování dalších funkcí.

#### 5.7.4 Uložení výstupů hodnocení

Hodnocení ve výstupní matici se nakonec uloží do tabulky *Vystup* v databázi.

```
setdbprefs(s)

conn = database('Dodavatele', '', 'password');

insert(conn, 'Vystup', {'Dodavatel', 'hodnoceniCena', 'hodnoceniJakost',
'hodnoceniDodani', 'hodnoceniCelk', 'skupina'}, Uloz)

close(conn)
```

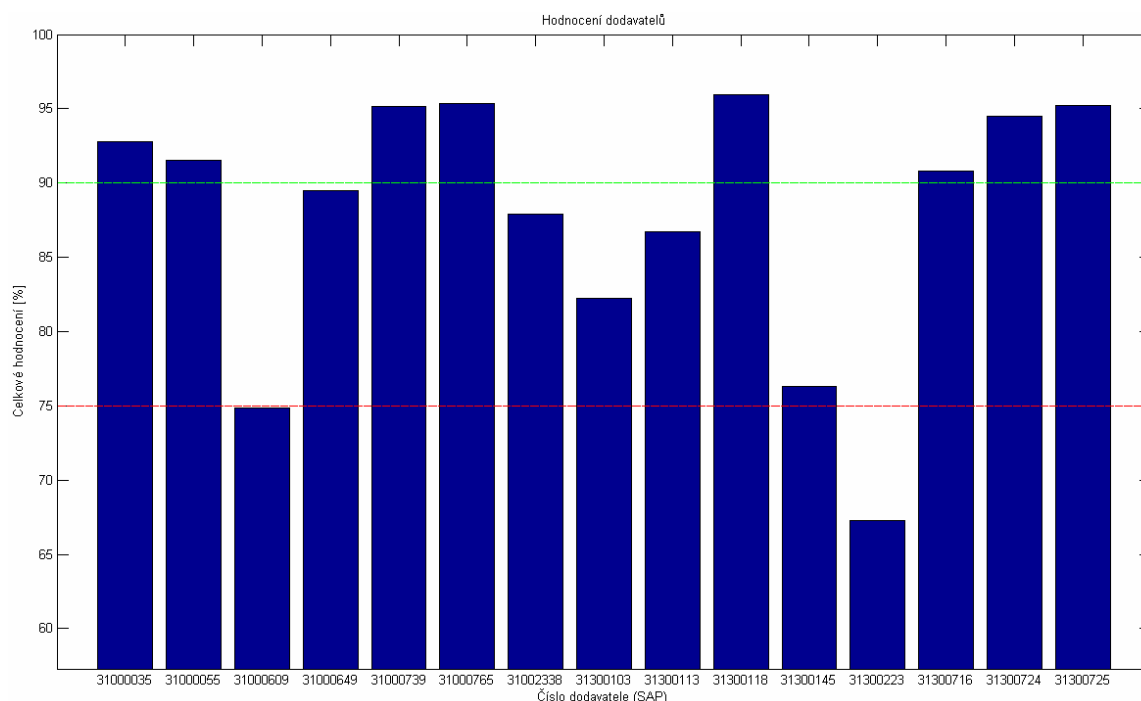
#### Program 10 - Uložení výstupů hodnocení

Příkaz *setdbprefs* definuje databázi, příkaz *conn* připojuje k databázi *Dodavatele.mdb*. Příkaz *insert* uloží hodnoty z matice *Uloz* do tabulky *Vystup* v databázi. Příkaz *close(conn)* ukončí spojení s databází.

### 5.8 Závěry z hodnocení

Příkazem *bar(v)*, je možné graficky zobrazit výsledky hodnocení. Na obrázku jsou ve sloupcích grafu znázorněny výše celkového hodnocení 15 dodavatelů. Protože vedoucí Oddělení nákupu požaduje, aby nebyly zveřejněny názvy dodavatelů, jsou zde uvedeny pouze jejich identifikační čísla z podnikového IS.

Dodavatelé hodnocení nad 90 bodů jsou vyhovující, firma je může považovat za velmi spolehlivé a pro nákup daného typu materiálu by měli být voleni především. Hodnocení pod 75 bodů ze zobrazované skupiny obdrželi dodavatelé ‘31000609’ a ‘31300223’, z celého analyzovaného souboru dodavatelů se jich pod touto hranicí umístilo asi 12%, pro firmu jsou nevyhovující a oddělení nákupu by problém s nimi mělo řešit. Ostatní dodavatelé (v grafu sloupce ‘31000649’, ‘31002338’, ‘31300103’, ‘31300113’ a ‘31300145’) jsou pro firmu podmíněčně vyhovující, to znamená, že ve vztahu s nimi musí být opatrnější a při nákupu materiálu spíše preferovat lépe hodnocené dodavatele.



**Obrázek 39 - Grafické znázornění hodnocení dodavatelů**

Hodnocení dodavatelů lze provádět pravidelně např. pro jednotlivé kvartály, což umožní zpětně porovnávat vývoj výkonnosti dodavatelů v čase. Výstupní data jsou uložena v databázi Dodavatele.mdb. V případě velké změny v hodnocení u konkrétního dodavatele mezi dvěma obdobími lze v tabulce *Vystup* vyhledat dílčí hodnocení a v tabulce *Ukazatele* výši jednotlivých ukazatelů, to umožní zjistit přesnou příčinu této změny.

## 6 Závěr

Cílem diplomové práce bylo navržení systému pro hodnocení výkonnosti materiálových dodavatelů firmy s využitím metod fuzzy logiky. Aplikace podává informace pro podporu rozhodování o vhodnosti dodavatelů pro kvalitní a dlouhodobou spolupráci a při opakovaném používání může podávat informace signalizující významné změny ve vývoji výkonnosti daného dodavatele.

Využití metody učení modelu podle souboru zadaných dat umožnilo namodelovat jeho parametry tak, aby co nejvíce odpovídaly požadavkům na výstupní hodnoty.

Při následném porovnání výstupů fuzzy modelu s hodnocením podle současné metody používané ve firmě lze tvrdit, že došlo k prokazatelnému zpřesnění hodnocení. To je dáno jak výpočtem ukazatelů hodnocení přímo podle přesných dat ze systému, tak i tím, že navrhovaný systém hodnotí vybrané ukazatele na základě míry jejich příslušnosti k určité množině hodnot (viz. kapitola 3.4 - Fuzzy množiny a fuzzy logika) a ne jen podle jejich realizování v daném intervalu.

Kromě vyšší přesnosti je největším přínosem navrhovaného řešení i rychlost hodnocení. To je dáno automatickým zpracováním dat extrahovaných přímo z databáze. Nevýhodou však zůstává absence dat o jakosti dodávek v databázi informačního systému, tato data se musejí do systému exportovat z tabulky v excelu, což brání možnosti úplné automatizace systému.

Využití fuzzy logiky pro úlohu hodnocení dodavatelů má svoje opodstatnění, neboť zde lze využít popisu problému pomocí vágních slovních pojmů.

Přínosem řešení pro firmu je tedy vyšší přesnost a vypovídací schopnost získaných informací, na jejichž základě lze činit rozhodnutí o nákupu. Tyto informace lze získat rychleji než při užití stávající metody a nesporným přínosem je i ušetření práce nákupních referentů, kteří nemusejí data zadávat ručně.

Osobně vidím velký přínos v možnosti ve firmě použít nové technologie a postupy na konkrétní praktické úloze.

## Seznam použité literatury

- [1] BRŮHA, L. *Visual Basic .net*. 2002. Computer Press. ISBN 80-7226-785-X.
- [2] DOSTÁL P. *Pokročilé metody analýz a modelování v podnikatelství a veřejné správě*. CERM. ISBN 978-80-7204-605-8.
- [3] DOSTÁL, P., RAIS, K., SOJKA, Z. *Pokročilé metody manažerského rozhodování*. 1. vyd. Praha: Grada, 2005.168s. ISBN 80-247-1338-1.
- [4] JURA, P. *Základy fuzzy logiky pro řízení a modelování*. Brno: Nakladatelství VUTIUM. 2003. ISBN 80-214-2261-0.
- [5] KOVAŘÍK, M. *Matlab*. Zlín: Nakladatelství Martin Stříž. 2008. ISBN 978-80-87106-09-9.
- [6] KROPÁČ, J. *Statistika B*. Tiskárna Blansko, 2007. 155 s. ISBN 80-214-3295-0.
- [7] MAŘÍK, V., ŠTĚPÁNKOVÁ, O. *Umělá inteligence I*. Praha: ACADEMIA, 1993. ISBN 80-200-0496-3.
- [8] MAŘÍK, V., ŠTĚPÁNKOVÁ, O. *Umělá inteligence II*. Praha: ACADEMIA, 1997. ISBN 80-200-0504-8.
- [9] MAŘÍK, V., ŠTĚPÁNKOVÁ, O. *Umělá inteligence IV*. Praha: ACADEMIA, 2003. ISBN 80-200-1044-0.
- [10] MIKULECKÝ, P. *Umělá inteligence a expertní systémy*. [online]. 2007 [cit. 25.4.2010]. Dostupné z <[lide.uhk.cz/fim/ucitel/mikulpe1/ES/ES1.htm](http://lide.uhk.cz/fim/ucitel/mikulpe1/ES/ES1.htm)>.
- [11] *Poznámky z konzultace s firemními zaměstnanci*.
- [12] THE MATHWORKS. *MATLAB – Fuzzy Logic Toolbox - User's Guide*. The MathWorks, Inc. 2008.
- [13] THE MATHWORKS. *MATLAB – Database Toolbox - User's Guide*. The MathWorks, Inc. 2008.

## 7 Seznam použitých zkratek a symbolů

ANFIS	Adaptive neuro-fuzzy inference system, kombinace fuzzy inferenčního systému a neuronové sítě
FIS	Fuzzy Inference System, aplikace fuzzy logiky umožňující vyhodnocování neurčitých proměnných
MF	Membership function, funkce členství proměnné v teorii fuzzy množin
SQL	Structured Query Language, je standardizovaný dotazovací jazyk používaný při práci s daty v relačních databázích
SŘBD	System řízení báze dat, softwarové vybavení, které zajišťuje práci s databází

## 8 Seznam tabulek

Tabulka 1 - Struktura entity Data .....	40
Tabulka 2 - Struktura entity Certifikat.....	40
Tabulka 3 - Struktura entity Jakost .....	40
Tabulka 4 - Struktura entity Ukazatele .....	41
Tabulka 5 - Struktura entity Vystup .....	41

## 9 Seznam obrázků

Obrázek 1 - model neuronu .....	13
Obrázek 2 - Fuzzy inference .....	19
Obrázek 3 - FIS Editor .....	23
Obrázek 4 - MF Editor .....	24
Obrázek 5 - Rule editor .....	25
Obrázek 6 - ANFIS Editor .....	26
Obrázek 7 - Příklad struktury ANFIS .....	27
Obrázek 8 - Database toolbox .....	28
Obrázek 9 - Organizační struktura společnosti .....	30
Obrázek 10 - Systém správy dat .....	32
Obrázek 11 - Model řešení .....	35
Obrázek 12 - Fáze přípravy dat .....	39
Obrázek 13 - Schéma fuzzy aplikace .....	51
Obrázek 14 - Transformační matice .....	52
Obrázek 15 - Výstupní matice .....	54
Obrázek 16 - FIS editor - Hodnocení ceny .....	56
Obrázek 17 - ANFIS editor - trénování modelu .....	57
Obrázek 18 - Struktura modelu .....	57
Obrázek 19 - Graf výstupních hodnot modelu a tréninkových dat .....	58
Obrázek 20 - Membership function editor - Cenová úroveň .....	59
Obrázek 21 - Membership function editor - Cenové chování .....	60
Obrázek 22 - Membership function editor – Výstup .....	60
Obrázek 23 - Rule Editor - Hodnocení ceny .....	61
Obrázek 24 - Surface viewer - Hodnocení ceny .....	61
Obrázek 25 - Rule Viewer - Hodnocení Ceny .....	62
Obrázek 26 - FIS editor - Hodnocení jakosti .....	62
Obrázek 27 - Membership function editor - Hodnocení jakosti .....	63
Obrázek 28 - Rule Editor - Hodnocení jakosti .....	64
Obrázek 29 - Surface Viewer - Hodnocení jakosti .....	64
Obrázek 30 - Rule Viewer - Hodnocení jakosti .....	65

Obrázek 31 - FIS editor - Hodnocení dodání.....	65
Obrázek 32 - Membership function editor - Hodnocení dodání.....	66
Obrázek 33 - Surface Viewer - Hodnocení dodání.....	66
Obrázek 34 - Rule Viewer - Hodnocení dodání .....	67
Obrázek 35 - FIS editor - Celkové hodnocení .....	67
Obrázek 36 - Membership function editor - Celkové hodnocení .....	68
Obrázek 37 - Surface Viewer - Celkové hodnocení .....	69
Obrázek 38 - Rule Viewer - Celkové hodnocení .....	69
Obrázek 39 - Grafické znázornění hodnocení dodavatelů.....	74

## 10 Seznam programů

Program 1 – Načtení dat z databáze.....	43
Program 2 - Přepočítání ukazatelů Certifikát a Jakost.....	44
Program 3 - Výpočet ukazatele Cenová úroveň .....	46
Program 4 - Výpočet ukazatelů Cenové chování, Dodržování Termínů a Dodržování množství.....	48
Program 5 - Uložení výstupu do databáze .....	50
Program 6 - Hodnocení pro dodržování množství.....	53
Program 7 - Načtení dat .....	70
Program 8 - Vyhodnocení dodavatelů .....	71
Program 9 - Grafická reprezentace hodnocení .....	72
Program 10 - Uložení výstupů hodnocení .....	73

## 11 Seznam příloh

Příloha 1 – program *ukazatele.m*

Příloha 2 – program *Hodnocení VB.NET*

Příloha 3 – program *fuzzyhodn.m*

## Příloha 1 – program *ukazatele.m*

```
%Nacitani z databaze
%-----
% Nastaveni pro pripojeni
s.DataReturnFormat = 'cellarray'; s.ErrorHandling = 'store'; s.NullNumberRead
= '0';
s.NullNumberWrite = '0'; s.NullStringRead = 'null'; s.NullStringWrite =
'null';
s.JDBCDataSourceFile = ''; s.UseRegistryForSources = 'yes';
setdbprefs(s)

% Pripojeni k databazi
% Uziti ODBC driver.
conn = database('Dodavatele','','password');%nutno nastavit pres Visual Query
Builder ODBC DataSource na databazi v MS Access

% Nacteni dat z databaze
e = exec(conn,'SELECT Data.Id_dod, Jakost.PPM, Certifikat.Certifikat FROM
((Data LEFT OUTER JOIN Jakost ON Data.Id_dod = Jakost.Id_dod) LEFT OUTER JOIN
Certifikat ON Data.Id_dod = Certifikat.Id_dod) GROUP BY Data.Id_dod,
Jakost.PPM, Certifikat.Certifikat');
e = fetch(e);
close(e)

f = exec(conn,'SELECT Id_dod, Id_mat, Datum_obj, Termin_obj, Termin_dodani,
Obj_mn, Dod_mn, Cena FROM Data ORDER BY Id_dod');
f = fetch(f);
close(f)

dod = exec(conn,'SELECT Id_dod FROM Data GROUP BY ID_dod');
dod = fetch(dod);
close(dod)

mat = exec(conn,'SELECT Id_mat, SUM(Cena) / SUM(Dod_mn) AS PrumCena FROM
Data GROUP BY Id_mat');
mat = fetch(mat);
close(mat)
% Ukonci spojeni s databazi
close(conn)
%//Nacitani z databaze
%-----

%Vypocet ukazatelu
%-----
%zjisteni velikosti jednotlivych matic hodnot
VelikostDod=size(dod.data);
Velikostf=size(f.data);
VelikostMat=size(mat.data);
PocRadkuDod=VelikostDod(1);%pocet dodavatele
PocRadkuf=Velikostf(1);%pocet zaznamu vybranych z tabulky Data, nacteno do f
PocRadkuMat=VelikostMat(1);%pocet typu materialu

%vypocet ukazatele Cenova uroven
%-----
l=1;
CenUroven=[];
CenovaUrovenMat=[];
for i=1:PocRadkuDod%prochazi vsechny vybrane id dodavatele
dodd=dod.data(i);

while not(strcmp(f.data{l,1},dodd.data{i}))%dokud nenarazi na prvni vyskyt,
zaznamy jsou serazeny podle id_dod = rychlejsi a jednodussi prohledavani
```



```

        l=l+1;
end

j=1;
    while (strcmp(f.data{l,1},dod.data{i}))%plati pro vsechny radky s danym id
dodavatele
        dodmn=f.data{l,7};%dodane mnozstvi
        cena=f.data{l,8};%skutecne datum dodani
        prumcena=cena/dodmn;
        idmat=f.data{l,2};%objednane mnozstvi

        for ff=1:PocRadkuMat
            if (strcmp(mat.data{ff,1},idmat))%porovnani retezcu, hleda se
odpovidajici id materialu

                    break
            end
        end
        pomercen=((prumcena/mat.data{ff,2})-1)*100;%procentni
vyjadreni odchylky ceny daneho dodavatele od prumerne ceny vsech dodavatelu
daneho mat

        pom=CenUroven;
        CenUroven=[pom;pomercen];

        if l==PocRadkuf
            break
        end
        l=l+1;
end

Ukcenuroven=mean(CenUroven);%prumerna cenova uroven dodavatele

pom=CenovaUrovenMat;
CenovaUrovenMat=[pom; dodd, Ukcenuroven];

end

%//vypocet ukazatele Cenova uroven
%-----

%vypocet ukazatelu Cenove chovani, Dodrzovani terminu a Dodrzovani mnozstvi
%-----
UkDodavka=[];
k=1;

for i=1:PocRadkuDod%prochazi vsechny vybrane id dodavatele
    dodd=dod.data(i);
    mn=[];
    datum=[];
    objcena=[];

    while not(strcmp(f.data{k,1},dod.data{i}))%dokud nenarazi na prvni vyskyt,
zaznamy jsou serazeny podle id_dod = rychlejsi a jednodussi prohledavani
        k=k+1;
    end

    j=1;

    while (strcmp(f.data{k,1},dod.data{i}))%plati pro vsechny radky s danym id
dodavatele

        objmn=f.data{k,6};%objednane mnozstvi
        dodmn=f.data{k,7};%dodane mnozstvi

```

```

datumobj=f.data{k,3};%datum objednani
terminobj=f.data{k,4};%planovane datum dodani
datumdod=f.data{k,5};%skutecne datum dodani

cena=f.data{k,8};%skutecne datum dodani
prumcena=cena/dodmn;

pom=mn;
mn=[pom; [objmn dodmn]];%vlozeni do matice - objednane mnozstvi,dodane
mnozstvi
pom=datum;
datum=[pom; [datenum(datumobj) datenum(terminobj)
datenum(datumdod)]];%vlozi do matice datum v numerickem vyjadreni
pom=objcena;
objcena=[pom;[datenum(datumobj) prumcena]];

j=j+1;

if k==PocRadkuf
    break
end
k=k+1;
end

SkutDoba=datum(:,3)-datum(:,1);%rozdil skutecneho data dodani a data
objednani
PlanDoba=datum(:,2)-datum(:,1);%rozdil planovaneho data dodani a data
objednani
pomer=SkutDoba./PlanDoba;%odchylka

polovina=(max(objcena(:,1))+min(objcena(:,1)))/2; %umozni rozdelit zaznamy
podle data na dve poloviny

if (j-1)>1%ve sledovanem obdobi je vice nez jeden zaznam

    cena1=[];
    cena2=[];

    for ii=1:(j-1)
        if objcena(ii,1)<=polovina
            pom=cena1;
            cena1=[pom; objcena(ii,2)];
        else
            pom=cena2;
            cena2=[pom; objcena(ii,2)];
        end
    end

    prumcena1=sum(cena1)/length(cena1);
    prumcena2=sum(cena2)/length(cena2);

    UkCenChovani=((prumcena2/prumcena1)-1)*100;

else UkCenChovani=0;%pokud je v danem obdobi nalezen pouze jeden zaznam,
nelze urcit koef. UkCenChovani, predpokladame tedy, ze koeficient je nulovy

end

UkDodrzTermin=(sum(pomer-1)/(j-1))*100;%v procentech vyjadrena prumerna
odchylka skutecne doby dodani od planovane doby dodani

```

```

    UkDodrzMn=abs(sum(mn(:,2)./mn(:,1)-1)/(j-1))*100;% v absolutni hodnote
    procentualni prumerna odchylna dodaneho mnozstvi od objednaneho
    Pom=UkDodavka;
    UkDodavka=[Pom; dodd, UkCenChovani, UkDodrzTermin, UkDodrzMn];%vlozeni do
    matice - id dodavatele, ukazatel Cenove chovani, ukazatel Dodrzovani terminu,
    ukazatel Dodrzovani mnozstvi
end
%//vypocet ukazatelu Cenove chovani, Dodrzovani terminu a Dodrzovani mnozstvi
%-----

%prepcet ukazatelu Certifikat a Jakost
%-----
vstup = e.data;
Velikost=size(vstup);%zjistí velikost matice
PocSloupcu=Velikost(2);%zjistí pocet sloupcu v matici
PocRadku=Velikost(1);%zjistí pocet radku v matici

Mcertifikat=[];
for i=1:PocRadku %smyčka postupne nacita jednotlivé radky matice
    radek=vstup(i,:);%nacteni i-teho radku

    if radek{3)=='A'
        certifikat=1;
    elseif radek{3)=='N'
        certifikat=0;
    else
        certifikat=0;
    end
    radek{3}=certifikat;

    if isnumeric(radek{2})
        jakost=radek{2}/10^4;%prevede na procenta
    else
        jakost=0;
    end
    radek{2}=jakost;
    vstup(i,:)=radek;
end
%//prepcet ukazatelu Certifikat a Jakost
%-----

%vlozeni hodnot ukazatelů do vystupni matice
Vystup=[CenovaUrovenMat,UkDodavka(:,2), vstup(:, [2 3]), UkDodavka(:, [3 4])];

%-----
%ulozeni vystupu do databaze
%-----
% Nastaveni pro pripojeni
s.DataReturnFormat = 'cellarray'; s.ErrorHandling = 'store'; s.NullNumberRead
= 'NaN'; s.NullNumberWrite = 'NaN';
s.NullStringRead = 'null'; s.NullStringWrite = 'null'; s.JDBCDataSourceFile =
''; s.UseRegistryForSources = 'yes';
setdbprefs(s)

% Pripojeni k databazi
% Uziti ODBC driver.
conn = database('Dodavatele','','password');

% Ulozeni dat do databaze
insert(conn,'Ukazatele',{'Dodavatel','Uroven','Chovani','Jakost','Certifikat',
'Terminy','Mnozstvi'},Vystup)

% Ukonci spojeni s databazi
close(conn)
%clear%vymaze promenne z Workspace, uvolneni pameti

```

## Příloha 2 – program *Hodnocení VB.NET*

```
Imports Microsoft.Office.Interop
Public Class Form1

    Public hodnoceni As Double = 0

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        OpenFileDialog1.ShowDialog()
        Button2.Enabled = True
        Button3.Enabled = True
        Button4.Enabled = True
        Button5.Enabled = True
    End Sub

    Private Sub OpenFileDialog1_FileOk(ByVal sender As System.Object, ByVal e
As System.ComponentModel.CancelEventArgs) Handles OpenFileDialog1.FileOk

        Label11.Text = OpenFileDialog1.FileName
        Label32.Text = "Načtený soubor: "

        'PROMENNE UKAZATELU
        Dim uk_certificate As Integer
        Dim uk_pricebehavior As Integer
        Dim uk_pricelevel As Integer
        Dim uk_quality As Integer
        Dim uk_supplyquantity As Integer
        Dim uk_supplyterms As Integer

        'VAHY UKAZATELU
        Dim w_certificate As Double = 0.06
        Dim w_pricebehavior As Double = 0.16
        Dim w_pricelevel As Double = 0.24
        Dim w_quality As Double = 0.24
        Dim w_supplyquantity As Double = 0.12
        Dim w_supplyterms As Double = 0.18

        Dim soubor As New IO.StreamReader(OpenFileDialog1.FileName)
        'otevřít soubor
        While Not soubor.EndOfStream           'číst, dokud nejsme na konci
souboru
            Dim radek As String = CStr(soubor.ReadLine())

            Dim part() As String = radek.Split(" ")

            uk_pricelevel = CInt(part(0))
            uk_pricebehavior = CInt(part(1))
            uk_quality = CInt(part(2))
            uk_certificate = CInt(part(3))
            uk_supplyterms = CInt(part(4))
            uk_supplyquantity = CInt(part(5))

            Dim hodnoceni As Double = 0
            Dim cenahodn As Double = 0
            Dim jakosthodn As Double = 0
            Dim dodanihodn As Double = 0
            Dim dok As Integer = 0

            'certifikat
            For g As Integer = 0 To 1
```

```

        If CBool(uk_certificate) =
CBool(ListView1.Items(g).SubItems(1).Text) Then
            hodnoceni = hodnoceni +
CInt(ListView1.Items(g).SubItems(2).Text) * w_certificate
            jakosthodn = jakosthodn + 0.2 *
CInt(ListView1.Items(g).SubItems(2).Text)
        End If
    Next g

'cenove chovani
For g As Integer = 0 To ListView2.Items.Count - 1
    If uk_pricebehavior < -100 Then
        hodnoceni = hodnoceni + 100 * w_pricebehavior
        cenahodn = cenahodn + 100 * 0.4
        dok = 1
        Exit For
    End If
    If uk_pricebehavior >=
CInt(ListView2.Items(g).SubItems(1).Text) And uk_pricebehavior <
CInt(ListView2.Items(g).SubItems(2).Text) Then
        hodnoceni = hodnoceni +
CInt(ListView2.Items(g).SubItems(3).Text) * w_pricebehavior
        cenahodn = cenahodn +
CInt(ListView2.Items(g).SubItems(3).Text) * 0.4
        dok = 1
        Exit For
    End If
Next g

If dok = 1 Then
    dok = 0
Else
    hodnoceni = hodnoceni + w_pricebehavior
    cenahodn = cenahodn + 0.4
End If

'cenova uroven
For g As Integer = 0 To ListView3.Items.Count - 1
    If uk_pricelevel < -100 Then
        hodnoceni = hodnoceni + 100 * w_pricelevel
        cenahodn = cenahodn + 100 * 0.6
        dok = 1
        Exit For
    End If

    If uk_pricelevel >= CInt(ListView3.Items(g).SubItems(1).Text)
And uk_pricelevel < CInt(ListView3.Items(g).SubItems(2).Text) Then
        hodnoceni = hodnoceni +
CInt(ListView3.Items(g).SubItems(3).Text) * w_pricelevel
        cenahodn = cenahodn +
CInt(ListView3.Items(g).SubItems(3).Text) * 0.6
        dok = 1
        Exit For
    End If
Next g

If dok = 1 Then
    dok = 0
Else
    hodnoceni = hodnoceni + w_pricelevel
    cenahodn = cenahodn + 0.6
End If

'jakost
For g As Integer = 0 To ListView4.Items.Count - 1

```

```

        If uk_quality >= CInt(ListView4.Items(g).SubItems(1).Text) And
uk_quality < CInt(ListView4.Items(g).SubItems(2).Text) Then
            hodnoceni = hodnoceni +
CInt(ListView4.Items(g).SubItems(3).Text) * w_quality
            jakosthodn = jakosthodn + 0.8 *
CInt(ListView4.Items(g).SubItems(3).Text)
            dok = 1
            Exit For
        End If
    Next g

    If dok = 1 Then
        dok = 0
    Else
        hodnoceni = hodnoceni + w_quality
        jakosthodn = jakosthodn + 0.8
    End If

    'dodrzovani mnozstvi
    For g As Integer = 0 To ListView5.Items.Count - 1
        If Math.Abs(uk_supplyquantity) >=
CInt(ListView5.Items(g).SubItems(1).Text) And uk_supplyquantity <
CInt(ListView5.Items(g).SubItems(2).Text) Then
            hodnoceni = hodnoceni +
CInt(ListView5.Items(g).SubItems(3).Text) * w_supplyquantity
            dodanihodn = dodanihodn +
CInt(ListView5.Items(g).SubItems(3).Text) * 0.4
            dok = 1
            Exit For
        End If
    Next g

    If dok = 1 Then
        dok = 0
    Else
        hodnoceni = hodnoceni + w_supplyquantity
        dodanihodn = dodanihodn + 0.4
    End If

    'dodrzovani terminu
    For g As Integer = 0 To ListView6.Items.Count - 1
        If uk_supplyterms < -100 Then
            hodnoceni = hodnoceni + w_pricebehavior
            dodanihodn = dodanihodn + 0.6
            dok = 1
            Exit For
        End If

        If uk_supplyterms >= CInt(ListView6.Items(g).SubItems(1).Text)
And uk_supplyterms < CInt(ListView6.Items(g).SubItems(2).Text) Then
            hodnoceni = hodnoceni +
CInt(ListView6.Items(g).SubItems(3).Text) * w_supplyterms
            dodanihodn = dodanihodn +
CInt(ListView6.Items(g).SubItems(3).Text) * 0.6
            dok = 1
            Exit For
        End If
    Next g

    If dok = 1 Then
        dok = 0
    Else
        hodnoceni = hodnoceni + w_supplyterms
        dodanihodn = dodanihodn + 0.6
    End If

```

```

End If

Dim slovni_vystup As String = ""

If hodnoceni < 75 Then
    slovni_vystup = "nevyhovující"
ElseIf hodnoceni < 90 Then
    slovni_vystup = "podmínečně vyhovující"
Else : slovni_vystup = "vyhovující"
End If

Dim kontrola As Double = dodanihodn + jakosthodn + cenahodn

Dim polozka As New ListViewItem() 'definovani objektu polozka pro
vypis radku do seznamu
polozka.Text = uk_pricelevel
polozka.SubItems.Add(uk_pricebehavior)
polozka.SubItems.Add(uk_quality)
polozka.SubItems.Add(uk_certificate)
polozka.SubItems.Add(uk_supplyterms)
polozka.SubItems.Add(uk_supplyquantity)
polozka.SubItems.Add(cenahodn)
polozka.SubItems.Add(jakosthodn)
polozka.SubItems.Add(dodanihodn)
polozka.SubItems.Add(hodnoceni)
polozka.SubItems.Add(slovni_vystup)

ListView7.Items.Add(polozka) 'pridani radku

End While
soubor.Close() 'zavřít soubor

End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    Call vypln()

End Sub

Private Sub vypln()
    'VYPLNENI POPISU TRANSFORMACNI MATICE
    Dim conString As String = "Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\Database1.mdf;Integrated
Security=True;User Instance=True"

    '-----CERTIFIKAT-----
    Dim con As New SqlConnection(conString) 'pripojeni k
databazi
    Dim cmd As New SqlCommand( _
"SELECT * FROM Certificate") 'formulovani dotazu
    cmd.Connection = con

    Dim da As New SqlDataAdapter(cmd)
    Dim dt As New DataTable()

    da.Fill(dt)
    Dim row As DataRow
    ListView1.Items.Clear() 'vymazani prvku ze seznamu
    For Each row In dt.Rows 've smyčce vypise radky z db do seznamu
        Dim polozka As New ListViewItem() 'definovani objektu polozka pro
vypis radku do seznamu

```

```

Dim id As String = row.Item(0)
Dim value As String = row.Item(1)
Dim weight As String = row.Item(2)

polozka.Text = id
polozka.SubItems.Add(value)
polozka.SubItems.Add(weight)

ListView1.Items.Add(polozka) 'pridani radku

Next row

'-----CENOVE CHOVANI-----
Dim con2 As New SqlConnection(conString) 'pripojeni k
databazi
Dim cmd2 As New SqlCommand( _
    "SELECT * FROM PriceBehavior") 'formulovani dotazu
cmd2.Connection = con2

Dim da2 As New SqlDataAdapter(cmd2)
Dim dt2 As New DataTable()

da2.Fill(dt2)
Dim row2 As DataRow
ListView2.Items.Clear() 'vymazani prvku ze seznamu
For Each row2 In dt2.Rows 've smyccce vypise radky z db do seznamu

    Dim polozka2 As New ListViewItem() 'definovani objektu polozka pro
vypis radku do seznamu

    Dim id As String = row2.Item(0)
    Dim low As String = row2.Item(1)
    Dim high As String = row2.Item(2)
    Dim weight As String = row2.Item(3)

    polozka2.Text = id
    polozka2.SubItems.Add(low)
    polozka2.SubItems.Add(high)
    polozka2.SubItems.Add(CInt(weight))

    ListView2.Items.Add(polozka2) 'pridani radku

Next row2

'-----CENOVA UROVEN-----
Dim con3 As New SqlConnection(conString) 'pripojeni k
databazi
Dim cmd3 As New SqlCommand( _
    "SELECT * FROM PriceLevel") 'formulovani dotazu
cmd3.Connection = con3

Dim da3 As New SqlDataAdapter(cmd3)
Dim dt3 As New DataTable()

da3.Fill(dt3)
Dim row3 As DataRow
ListView3.Items.Clear() 'vymazani prvku ze seznamu
For Each row3 In dt3.Rows 've smyccce vypise radky z db do seznamu

    Dim polozka3 As New ListViewItem() 'definovani objektu polozka pro
vypis radku do seznamu
    Dim id As String = row3.Item(0)
    Dim low As String = row3.Item(1)
    Dim high As String = row3.Item(2)
    Dim weight As String = row3.Item(3)

```



```

        polozka3.Text = id
        polozka3.SubItems.Add(low)
        polozka3.SubItems.Add(high)
        polozka3.SubItems.Add(CInt(weight))

        ListView3.Items.Add(polozka3) 'pridani radku
    Next row3

    '-----JAKOST-----
    Dim con4 As New SqlClient.SqlConnection(conString) 'pripojeni k
databazi
    Dim cmd4 As New SqlClient.SqlCommand( _
        "SELECT * FROM Quality") 'formulovani dotazu
    cmd4.Connection = con4
    Dim da4 As New SqlClient.SqlDataAdapter(cmd4)
    Dim dt4 As New DataTable()

    da4.Fill(dt4)
    Dim row4 As DataRow
    ListView4.Items.Clear() 'vymazani prvku ze seznamu
    For Each row4 In dt4.Rows 've smyce vypise radky z db do seznamu

        Dim polozka4 As New ListViewItem() 'definovani objektu polozka pro
vypis radku do seznamu
        Dim id As String = row4.Item(0)
        Dim low As String = row4.Item(1)
        Dim high As String = row4.Item(2)
        Dim weight As String = row4.Item(3)

        polozka4.Text = id
        polozka4.SubItems.Add(low)
        polozka4.SubItems.Add(high)
        polozka4.SubItems.Add(CInt(weight))

        ListView4.Items.Add(polozka4) 'pridani radku
    Next row4

    '-----DODRZOVANI MNOZSTVI-----
    Dim con5 As New SqlClient.SqlConnection(conString) 'pripojeni k
databazi
    Dim cmd5 As New SqlClient.SqlCommand( _
        "SELECT * FROM SupplyQuantity") 'formulovani dotazu
    cmd5.Connection = con5

    Dim da5 As New SqlClient.SqlDataAdapter(cmd5)
    Dim dt5 As New DataTable()
    da5.Fill(dt5)
    Dim row5 As DataRow
    ListView5.Items.Clear() 'vymazani prvku ze seznamu
    For Each row5 In dt5.Rows 've smyce vypise radky z db do seznamu

        Dim polozka5 As New ListViewItem() 'definovani objektu polozka pro
vypis radku do seznamu
        Dim id As String = row5.Item(0)
        Dim low As String = row5.Item(1)
        Dim high As String = row5.Item(2)
        Dim weight As String = row5.Item(3)

        polozka5.Text = id
        polozka5.SubItems.Add(low)
        polozka5.SubItems.Add(high)
        polozka5.SubItems.Add(CInt(weight))

        ListView5.Items.Add(polozka5) 'pridani radku

```

```

Next row5

'-----DODRZOVANI TERMINU-----
Dim con6 As New SqlConnection(conString) 'pripojeni k
databazi
Dim cmd6 As New SqlCommand(
    "SELECT * FROM SupplyTerms") 'formulovani dotazu
cmd6.Connection = con6
Dim da6 As New SqlDataAdapter(cmd6)
Dim dt6 As New DataTable()

da6.Fill(dt6)
Dim row6 As DataRow
ListView6.Items.Clear() 'vymazani prvku ze seznamu
For Each row6 In dt6.Rows 've smyccce vypise radky z db do seznamu

    Dim polozka6 As New ListViewItem() 'definovani objektu polozka pro
vypis radku do seznamu
    Dim id As String = row6.Item(0)
    Dim low As String = row6.Item(1)
    Dim high As String = row6.Item(2)
    Dim weight As String = row6.Item(3)

    polozka6.Text = id
    polozka6.SubItems.Add(low)
    polozka6.SubItems.Add(high)
    polozka6.SubItems.Add(CInt(weight))

    ListView6.Items.Add(polozka6) 'pridani radku

Next row6
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    SaveFileDialog2.ShowDialog()
End Sub

Private Sub SaveFileDialog2_FileOk(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles SaveFileDialog2.FileOk
    Dim part(2) As String

    Dim soubor As New IO.StreamWriter(SaveFileDialog2.FileName)
'otevřít soubor

    For f = 0 To ListView7.Items.Count - 1
        part(0) = (ListView7.Items(f).SubItems(0).Text).Replace(",", ".")
        part(1) = (ListView7.Items(f).SubItems(1).Text).Replace(",", ".")
        part(2) = (ListView7.Items(f).SubItems(6).Text).Replace(",", ".")

        Dim radek As String = String.Join(" ", part)
        soubor.WriteLine(radek)
    Next f

    soubor.Close()
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    SaveFileDialog3.ShowDialog()
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click

```

```

        SaveFileDialog4.ShowDialog()
    End Sub

    Private Sub SaveFileDialog3_FileOk(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles SaveFileDialog3.FileOk
        Dim part(2) As String

        Dim soubor As New IO.StreamWriter(SaveFileDialog3.FileName)
'otevřít soubor

        For f = 0 To ListView7.Items.Count - 1
            part(0) = (ListView7.Items(f).SubItems(2).Text).Replace(",", ".")
            part(1) = (ListView7.Items(f).SubItems(3).Text).Replace(",", ".")
            part(2) = (ListView7.Items(f).SubItems(7).Text).Replace(",", ".")
            Dim radek As String = String.Join(" ", part)
            soubor.WriteLine(radek)
        Next f

        soubor.Close()
    End Sub

    Private Sub SaveFileDialog4_FileOk(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles SaveFileDialog4.FileOk
        Dim part(2) As String

        Dim soubor As New IO.StreamWriter(SaveFileDialog4.FileName)
'otevřít soubor

        For f = 0 To ListView7.Items.Count - 1
            part(0) = (ListView7.Items(f).SubItems(4).Text).Replace(",", ".")
            part(1) = (ListView7.Items(f).SubItems(5).Text).Replace(",", ".")
            part(2) = (ListView7.Items(f).SubItems(8).Text).Replace(",", ".")
            Dim radek As String = String.Join(" ", part)
            soubor.WriteLine(radek)
        Next f

        soubor.Close()
    End Sub

    Private Sub Button2_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button2.Click
        SaveFileDialog1.ShowDialog()
    End Sub

    Private Sub SaveFileDialog1_FileOk(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles SaveFileDialog1.FileOk
        Dim part(3) As String
        Dim soubor As New IO.StreamWriter(SaveFileDialog1.FileName)
'otevřít soubor

        For f = 0 To ListView7.Items.Count - 1
            For g = 0 To 3
                part(g) = (ListView7.Items(f).SubItems(g +
6).Text).Replace(",", ".")
            Next g
            Dim radek As String = String.Join(" ", part)

            soubor.WriteLine(radek)

        Next f

        soubor.Close()
    End Sub
End Class

```

## Příloha 3 – program *fuzzyhodn.m*

```
% Set preferences with setdbprefs.
s.DataReturnFormat = 'cellarray'; s.ErrorHandling = 'store'; s.NullNumberRead
= 'NaN';
s.NullNumberWrite = 'NaN'; s.NullStringRead = 'null'; s.NullStringWrite =
'null';
s.JDBCDataSourceFile = ''; s.UseRegistryForSources = 'yes';

setdbprefs(s)

% Make connection to database. Note that the password has been omitted.
% Using ODBC driver.
conn = database('Dodavatele','','password');%nutno nastavit pres Visual Query
Builder ODBC DataSource na databazi v MS Access

% Read data from database.

e = exec(conn,'SELECT ALL
Dodavatel,Uroven,Chovani,Jakost,Certifikat,Terminy,Mnozstvi FROM Ukazatele');
e = fetch(e);
close(e)

% Close database connection.
close(conn)

%vyhodnoceni dat ve Fuzzy modelu
%-----
VystupMat=[];%prazdna vystupni matice
DataHodn=[];

%nacteni jednotlivych fuzzy inferencnich systemu
Cen = readfis('Cenove-hodnoceni.fis');
Jak = readfis('Jakost-hodnoceni.fis');
Dod = readfis('Dodani-hodnoceni.fis');
Celk = readfis('Celk-hodnoceni.fis');

vstup = e.data(:, [2 3 4 5 6 7]);
id= e.data(:,1);
Velikost=size(vstup);%zjistí velikost matice
PocSloupcu=Velikost(2);%zjistí pocet sloupcu v matici
PocRadku=Velikost(1);%zjistí pocet radku v matici

for i=1:PocRadku %smyčka postupne nacita jednotlivé radky matice
    radek=vstup(i,:);%nacteni i-teho radku

    vstupCena=[(radek{1}) (radek{2})];%nacte hodnoty cenova-uroven a cenove-
chovani
    vstupJakost=[(radek{3}) (radek{4})];%nacte hodnoty jakost a certifikat
    vstupDodani=[(radek{5}) (radek{6})];%nacte hodnoty termíny a množství

    hodnoceniCena=evalfis(vstupCena, Cen);%vyhodnoceni ceny ve Fuzzy systemu
    hodnoceniJakost=evalfis(vstupJakost, Jak);%vyhodnoceni jakosti ve Fuzzy
systemu
    hodnoceniDodani=evalfis(vstupDodani, Dod);%vyhodnoceni dodavky ve Fuzzy
systemu

    vstupCelk=[hodnoceniCena hodnoceniJakost hodnoceniDodani];%pouziti vystupu z
predchozich FIS pro vstup na celkove hodnoceni
    hodnoceniCelk=evalfis(vstupCelk, Celk);%celkove vyhodnoceni dodavatele ve
Fuzzy systemu
%//vyhodnoceni dat ve Fuzzy modelu
%-----
```

```

%slovní vyhodnocení
%-----
if hodnoceniCelk <75
    skupina=3;%Nevyhovující
elseif hodnoceniCelk<90
    skupina=2;%Podmínečně vyhovující
else
    skupina=1;%Vyhovující
end
%//slovní vyhodnocení
%-----

Vystup=[num2cell(hodnoceniCena) hodnoceniJakost hodnoceniDodani hodnoceniCelk
skupina];%vstupní vektor, hodnoty převedeny do bunek
Predch=VystupMat;
VystupMat=[Predch; Vystup];%sloučení matice předchozích výsledků s vektorem
aktuálních výsledků

%tvoreni matice dat pro vykreslení grafu
Pom=DataHodn;
DataHodn = [Pom; hodnoceniCelk];
end

Uloz=[id VystupMat];

%vykreslení grafu pro prvních 15 dodavatelů
%-----
prubeh=PocRadku;
ee=linspace(0,prubeh+1,1000);
v=DataHodn(1:prubeh);
rozsah= [min(v)-10 100];

bar(v), title('Hodnocení dodavatelů'), xlabel('Číslo dodavatele
(SAP)'),
ylabel('Celkové hodnocení [%]'),set(gca,'YLimMode', 'auto', 'YLim',
rozsah, 'XLimMode', 'auto', 'XTickLabel',Uloz(:,1)); grid off;
hold on;
plot(ee,90, 'g-');
hold on;
plot(ee,75, 'r-');
hold off;
%//vykreslení grafu
%-----

%-----
%uložení výstupu do databáze
%-----
% Set preferences with setdbprefs.
s.DataReturnFormat = 'cellarray'; s.ErrorHandling = 'store'; s.NullNumberRead
= 'NaN'; s.NullNumberWrite = 'NaN';
s.NullStringRead = 'null'; s.NullStringWrite = 'null'; s.JDBCDataSourceFile =
''; s.UseRegistryForSources = 'yes';
setdbprefs(s)

% Make connection to database.Note that the password has been omitted.
% Using ODBC driver.
conn = database('Dodavatele','','password');
% Write data to database.
insert(conn,'Vystup',{'Dodavatel','hodnoceniCena','hodnoceniJakost','h
odnoceniDodani','hodnoceniCelk','skupina'},Uloz)
% Close database connection.
close(conn)

```