

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

System pro hodnocení uchazečů o zaměstnání

**Zpracování požadavku na systém pro podporu rozhodování mezi uchazeči
o zaměstnání podle hodnotících kritérií**

Diplomová práce

Autor: Bc. Václav Slavíček
im2-k

Vedoucí práce: Ing. Tomáš Pábl

Praha

červenec 2005

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Praze dne 30.7. 2005

Václav Slaviček

Poděkování:

Děkuji panu Ing. Tomáši Páblovi za to, že se mého tématu ujal, ponechal mi dostatek prostoru, ale zároveň byl připraven kdykoliv podat pomocnou ruku. Poděkování patří i všem ostatním pedagogům Fakulty informatiky a managementu UHK, kteří nejednou obětavou radou během konzultací nevědomě spoluutvářeli finální podobu projektu.

Anotace práce:

Práce si klade za cíl formulovat požadavky, analyzovat a navrhnout nástroj pro hodnocení uchazečů o zaměstnání. Navržená aplikace by měla pomáhat managementu lidských zdrojů organizace při rozhodování o přijetí či odmítnutí posuzovaných kandidátů.

První část práce popisuje výběr uchazečů jako jednu z manažerských aktivit firmy a snaží se přiblížit jeho metodiku. Tato část pojímá problém jednak z hlediska společenských věd, značná pozornost je věnována použitelným algoritmům vícekriteriálního rozhodování. Autor zkoumá i možnosti využití fuzzy logiky.

Druhá část se pak zabývá návrhem samotného systému pro podporu rozhodování jakožto části informačního systému pro řízení lidských zdrojů organizace. Metodika návrhu přitom vychází z koncepcí RAD³ a UML, což je výhodné pro pozdější případnou implementaci systému v některém objektově orientovaném jazyce.

HR selection system

The purpose of the thesis is to gather requirements, analyse and design a tool for evaluation of applicants within human resource selection process to help management in making effective employment decision. The first part of the thesis explores the selection process as an essential management activity, and it tries to introduce a methodology for such a task. It provides a study of the problem from the perspective of social science. Then, particular attention has been paid to several practical multiple criteria decision making algorithms. The author also considers implementing a fuzzy logic approach.

The second section goes over SW design of the application. The application is conceived as a part of a human resources information system in a company. Promising concepts will be employed throughout all the phases of this task, for instance the RAD³ and UML, for future easy implementation in any object oriented language.

Obsah:

1.	ÚVOD	1
2.	LITERÁRNÍ REŠERŠE	3
2.1	VÝBĚR UCHAZEČŮ JAKO MANAŽERSKÁ AKTIVITA	3
2.2	ANALÝZA A NÁVRH SYSTÉMU	3
3.	VOLBA METODOLOGIE.....	5
3.1	VÝBĚR UCHAZEČŮ	5
3.2	ANALÝZA A NÁVRH SYSTÉMU	5
4.	POŽADAVKY	7
5.	POUŽITÉ TECHNOLOGIE A SOFTWARE PRODUKTY	9
5.1	TECHNOLOGIE.....	9
5.1.1	<i>UML</i>	9
5.1.2	<i>XML</i>	9
5.1.3	<i>Relační databáze</i>	9
5.2	SOFTWARE PRODUKTY	9
5.2.1	<i>Metodologie výběru uchazečů</i>	9
5.2.2	<i>Analýza a návrh systému</i>	10
6.	ANALÝZA PROBLÉMOVÉ OBLASTI.....	11
6.1	PRÁVNÍ RÁMEC	12
6.2	ŘÍZENÍ LIDSKÝCH ZDROJŮ V ORGANIZACI	13
6.3	ANALÝZA PRACOVNÍ POZICE.....	15
6.4	NÁBOR UCHAZEČŮ.....	16
6.5	VÝBĚR UCHAZEČŮ	17
6.5.1	<i>Charakteristiky uchazečů</i>	17
6.5.2	<i>Hodnoty prediktorů a jejich zjišťování</i>	24
6.6	MODEL PRO HODNOCENÍ VARIANT	28
6.6.1	<i>Obecné modely</i>	28
6.6.2	<i>Speciální modely pro hodnocení uchazečů</i>	34
6.6.3	<i>Využití fuzzy teorie v rozhodovacím procesu</i>	38
7.	NÁVRH SYSTÉMU	50
7.1	PŘÍPADY UŽITÍ	50

7.1.1	<i>Import dat o uchazečích a úrovních jejich prediktorů.....</i>	50
7.1.2	<i>Obdržení povelu k získání hodnot dalších prediktorů od uchazečů.....</i>	51
7.1.3	<i>Import znalostí o pracovních pozicích</i>	52
7.1.4	<i>Obdržení hodnocení uchazeče ve vztahu k pozici.....</i>	54
7.1.5	<i>Přijmutí uchazeče.....</i>	55
7.1.6	<i>Import dat o zaměstnancích a jejich hodnocení</i>	55
7.1.7	<i>Obdržení informace o kvalitě znalosti pozice.....</i>	55
7.2	PERZISTENCE OBJEKTŮ	56
7.3	DIAGRAM KOMPONENT	57
7.4	DIAGRAMY TŘÍD	59
8.	VÝSLEDKY	61
9.	ZÁVĚRY A DOPORUČENÍ	62
10.	SEZNAM POUŽITÉ LITERATURY	64
11.	PŘÍLOHY	68

1. Úvod

Motivace k řešení úlohy

Motivaci k systematickému řešení problému spojeného s výběrem uchazečů jsem si uvědomil v době, kdy jsem byl ve svém zaměstnání kromě úkolů řadového pracovníka pověřen navíc nábořem a výběrem svých budoucích kolegů. Tehdy jsem osobně pocítil, o jak složitou a zodpovědnou činnost se jedná. Při tomto procesu jsou v případě falešně pozitivní chyby v sázce nejen zbytečně vynaložené platové prostředky, ale také trpělivost spolupracovníků, čas samotného nevhodného zaměstnance, potenciálně hrozí vznik materiální škody a poškození pověsti organizace. Značná, i když ne tolik hmatatelná škoda vznikne jistě i při falešně negativním výsledku hodnocení uchazeče, který tím pádem obohatí svým přínosem namísto našich řad konkurenci.

Protože jsem ve stejné době navštěvoval přednášky týkající se expertních systémů a systémů na podporu rozhodování, rozhodl jsem se čistě pro školní účely navrhnout systém pro podporu rozhodování při výběru uchazečů v rámci své diplomové práce.

Očekávaný přínos systému

Obecně je prokázáno, že i částečným aplikováním racionálních systematických strategií při rozhodování zvýšíme pravděpodobnost správného rozhodnutí oproti rozhodování intuitivnímu. Je tomu tak zejména protože hodnotitel vybavený toliko intuicí není schopen mít na zřeteli různé váhy jednotlivých kritérií [53]. Otázkou samozřejmě zůstává, do jaké míry proces zautomatizovat. Pro jednoduché porovnání dvou uchazečů by zřejmě stačil papír s tužkou, abychom si vypracovali SWOT analýzy každého z nich. Určitý přínos zkrátka nastane vždy, když k hodnocení použijeme jakoukoliv explicitně určenou metodu. Více či méně tím eliminujeme i možnou diskriminaci uchazečů, která často vychází z předsudků hodnotitelů, aniž by si toho tito byli vědomi.

Pojetí práce

Nejprve stanovíme požadavky na vyvíjený systém (kap. 4.) a definujeme technologické a prostředky, které hodláme využívat při další činnosti (kap. 5).

Podstatnou část práce (kap. 6) věnujeme popisu aktivit, které jsou v rámci řízení lidských zdrojů provozovány v každé organizaci. K nim patří i samotný rozhodovací proces při výběru uchazečů. Na následujících stránkách nepronikneme do detailního popisu náboru uchazečů nebo přípravy a ověřování používaných testů – to jsou úkoly personalistů,

psychologů a dalších odborníků v dílčích oborech. Systém naopak očekává vložení „hotových“ výsledků provedeného testování uchazečů jakožto vstupních dat.

V rámci druhé poloviny diplomové práce (kap. 7) navrhne vlastní systém. Návrh bude sestávat z vytvoření diagramů případů užití, diagramů komponent a diagramů tříd. Na závěr zhodnotíme kvalitu vytvořeného návrhu a nastíníme možnosti jeho případného dalšího rozšíření.

2. Literární rešerše

2.1 Výběr uchazečů jako manažerská aktivita

Knihou *Human Resource Selection* od Roberta D. Gatewooda a Humerta S. Fielda obsahuje detailní vědecký pohled na výběr uchazečů o zaměstnání. Při přípravě analýzy mi byla cennou pomůckou.

Publikace *Základy fuzzy modelování* [41] je počinem Prof. Viléma Nováka, který v současnosti působí jako ředitel Ústavu pro výzkum a aplikaci fuzzy modelování pod Přírodovědeckou fakultou Ostravské univerzity. Autor v ní českého čtenáře přehledně a zároveň fundovaně seznamuje se základy fuzzy logiky a fuzzy regulace.

Portál *HR-guide* (<http://www.hr-guide.com>) [22] obsahuje stovky stránek s informacemi relevantními k problematice lidských zdrojů (Human Resources). Struktura portálu připomíná pojmovou mapu oblasti HR, proto se zde návštěvník snadno orientuje.

Webová stránka <http://www.axiomsoftware.com/products/jobq.asp> představuje již existující nástroj pro hodnocení uchazečů JobQ od britské společnosti Axiom Software, Ltd. Nástroj se snaží zjednodušit všechny aspekty práce spojené s náborem a výběrem uchazečů, včetně automatického rozesílání odmítavých e-mailů nevybraným uchazečům.

Za zmínku stojí také webové stránky společnosti Resource Associates, Inc. (<http://www.resourceassociates.com/>). Tato společnost nabízí komerční testy pro zkvalitnění procesu výběru lidských zdrojů. Testy jsou společností nejen distribuovány, ale rovněž vyvíjeny. Návštěvník zde nalezne přehledy hodnocených kritérií, používaných testů, odkazy na odborné články věnující se dané problematice a další důležité informace.

2.2 Analýza a návrh systému

Vědecký portál *ScienceDirect* (<http://www.sciencedirect.com>) vydavatelství Elsevier představuje rozsáhlou elektronickou knihovnu přehledně rozříděnou podle oborů. Součástí jsou i elektronické žurnály *Decision Support Systems* a *Fuzzy Sets and Systems* [7], ve kterých již několik let vychází měsíčně řádově deset článků zaměřených na teorii i praktickou aplikaci problémových okruhů. Zatímco veřejnosti je k dispozici omezený rozsah článků, z počítačové sítě FIM UHK lze přistupovat i k obsahu určenému pouze subskriptorům. Pro získání prvotních informací o fuzzy přístupu k vícekritériálnímu rozhodování je zvláště vhodný zde publikovaný článek *Fuzzy multiple criteria decision*

making: Recent Developments [11], v jehož závěru je uveden vyčerpávající seznam tématické literatury.

Seriózní návod jak uchopit nástroj UML, a to včetně praktického využití ve firemním prostředí, skýtá publikace *Myslíme v UML* [49]. V souvislosti s návrhy systémů prostřednictvím UML musím také vyzdvihnout *Soubor elektronických knih od RNDr. Ilji Kravala*, které lze zdarma získat na adrese <http://www.objects.cz>. Přestože tato práce není příliš známa, je přínosná svým nadhledem nad problematikou i špetkou typicky českého humoru.

3. Volba metodologie

3.1 Výběr uchazečů

Nejpoužívanější metody metodiky výběru uchazečů jsou podrobně popsány v kapitole 6.6.2 v rámci analýzy systému, kde se také snažíme nalézt metodu nejvhodnější pro konkrétně pro naše účely. V návrhu nakonec pro výpočet vah kritérií použijeme Saatyho metodu (kap. 6.6.1, [23]), vícekriteriální hodnocení variant budeme realizovat vícerozměrnou vícevrstvou fuzzy strategií (kap. 6.6.3, [25]).

3.2 Analýza a návrh systému

K tvorbě aplikací existuje celá řada přístupů. Od požadavků až po finální řešení lze projekt řídit na základě zcela odlišné filozofie.

Jedním z nechvalně známých, často používaných způsobů je např. tzv. „metoda tunel“, kdy je řízení prosto jakékoliv koncepce, pouze je vyvíjen tlak na nalezení řešení, tj. „východu z tunelu“.

Příklady dalších metod jsou:

- *Analýza – design – kód* – projekt postupně prochází těmito třemi fázemi.
- *Iterace a inkrementace (I+I)* – do fází uvedených v předchozím bodu se vstupuje opakovaně.

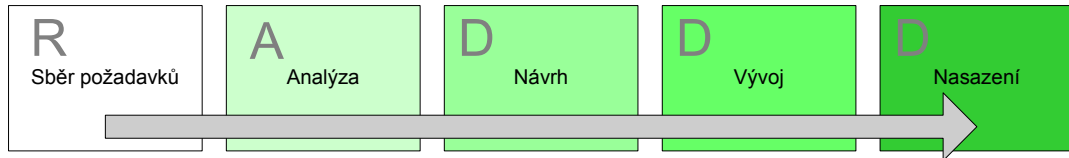
K uchování návrhu se nejčastěji používá jazyk UML (Unified Modeling Language), alternativně DFD (Data Flow Diagramming).

Vývojový proces by měl v případě použití UML být rozhodně řízený diagramem případů užití, zaměřený na architekturu, iterativní a inkrementální. Filozofie předpokládá členění lidských zdrojů, zainteresovaných na tvorbě projektu, do určitých rolí (analytik, designér, programátor). V případě menších projektů může při dostatečném zachování těchto principů zastupovat několik rolí jeden pracovník [35].

Navrhovaná aplikace bude patřit spíše mezi méně rozsáhlé. Přesto jsem se rozhodl použít právě metodu zahrnující oddělenou analýzu a tvorbu kódu. Struktura metody bývá v cizojazyčné literatuře označována také jako RAD³:

- Sběr požadavků (Requirements gathering),
- Analýza (Analysis),

- Návrh (Design),
- Vývoj (Development),
- Nasazení (Deployment) [49]



Obrázek 1– Metoda RAD³ [51]

Metoda samotná se potom někdy označuje jako GRAPPLE (Guidelines for Rapid Application Engineering) [49].

Důvodem výběru právě této metody je příležitost nacvičit si správný postup tak, aby šel využít v budoucí praxi při skutečných zakázkových projektech.

Diplomová práce bude ale obsahovat pouze první tři fáze realizace, to znamená že vývoj a nasazení systému v jejím rámci provádět nebudeme.

4. Požadavky

Výběr uchazečů

Základní funkcí systému bude pochopitelně hodnocení všech uchazečů připadajících v úvahu pro obsazení určité pozice. Pozice bude popsána specifikací pozice, která se v praxi získává z analýzy pozice.

Výběr pozic

Dovedeme si představit, že při nasazení systému ve větší organizaci by byla užitečná nejen funkce výběru vhodných kandidátů pro pozici, ale rovněž tak vhodných pozic pro určitého uchazeče. To by kupříkladu posloužilo uchazečům, kteří projeví spontánní zájem pracovat ve společnosti bez konkrétnějšího povědomí o aktuálně otevřených pozicích.

Průvodce při stanovování váhy prediktorů

Znalosti specifikující otevírané pozice (konkrétně váhy jednotlivých prediktorů, předpovídající ve svém souhrnu budoucí pracovní výkon zaměstnance) by měly být do systému zadávány přímo příslušnými experty. Přesto by bylo vhodné včlenit do systému pomůcku, která by nabízela nastavení vah prediktorů využitím metod odhadu vah vícekriteriálního rozhodování.

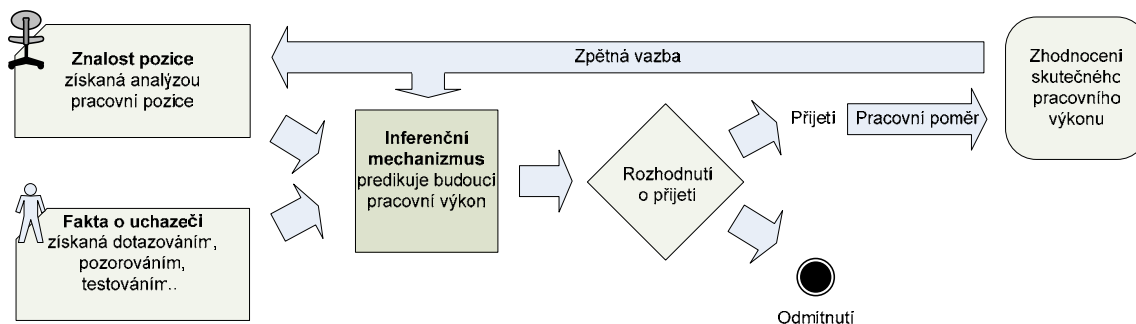
Hodnocení validity rozhodovacího procesu

Systém by měl disponovat zpětnou vazbou. Tato vazba by měla ovlivňovat vlastnosti v budoucnu používaného rozhodovacího procesu na základě hodnocení úspěšnosti dřívějšího výběru pracovníků. Hodnocení zaměstnanců je třeba provádět po časovém odstupu řádově alespoň jeden rok. Zpětná vazba umožňuje prověřit celý logický řetězec od prediktorů (testů, dotazníků atp.) přes spolehlivá a platná kritéria požadovaná pozicí (např. pečlivost, logické myšlení) až po inferenční a rozhodovací mechanismus. Proto bychom se podstatně ochudili, kdybychom nearchivovali:

- podklady vedoucí k rozhodnutím učiněným v historii,
- záznamy o následném pracovním výkonu přijatých zaměstnanců.

Porovnáním těchto dvou poznatků lze odhalit slabá místa použitého modelu a korigovat jej pro budoucí použití. Neméně cennou informací je bezesporu také výkonnost odmítnutého uchazeče/propuštěného zaměstnance, pracujícího nyní u jiného

zaměstnavatele. Zde ovšem narážíme na ustanovení Zákona o ochraně osobních údajů §5 odst. 1 pís. e, f – od odmítnutého uchazeče či dokonce propuštěného zaměstnance lze stěží očekávat souhlas s dalším zpracováváním jeho osobních údajů za účelem sledování výkonnosti v jiné firmě.



Obrázek 2 – Zpětná vazba pro hodnocení validity rozhodovacího procesu.

Škálovatelnost

Systém by měl být použitelný nejen velkými společnostmi. I pro relativně malou firmu čítající cca 10-50 zaměstnanců by jistě bylo ekonomicky přínosné využít alespoň některé funkce systému namísto čistě intuitivního způsobu rozhodování.

Napojení na IS pro řízení lidských zdrojů organizace

Jelikož předpokládáme, že organizace pro řízení lidských zdrojů pravděpodobně používá nějaký stávající IS, bylo by vhodné, kdyby šel navrhovaný systém s tímto IS integrovat. Náš systém by měl obsahovat modul externích zdrojů, který by tato externí data využíval.

Možnost komplexního i diskrétního přístupu

Rozdíl mezi komplexním a diskrétním přístupem k výběru uchazeče popisuje kapitola 6.6.2.

5. Použité technologie a softwarové produkty

5.1 Technologie

5.1.1 UML

Jednotný modelovací jazyk (Unified Modelling Language), nyní ve své třetí generaci, je otevřenou metodou určenou ke specifikaci, vizualizaci, projekci a dokumentaci částí objektově orientovaných SW systémů při jejich vývoji. UML představuje kompilaci „těch nejlepších inženýrských praktik“, což je přínosem při modelování rozsáhlých, komplexních systémů. Dodejme, že UML nemusí popisovat jen softwarový produkt, ale v podstatě jakýkoliv systém [49].

5.1.2 XML

Za zkratkou XML se skrývá standardní textový formát pro výměnu informací. Podobně jako jazyk HTML vychází z množiny jazyků SGML (Standard Generalized Markup Language). Je univerzální, má mezinárodní podporu (podle ISO 10646). Přenáší vysoký informační obsah a snadno se konvertuje do dalších formátů. Strukturu dokumentu vystavěného ve formátu XML lze kontrolovat pomocí schémat [34].

5.1.3 Relační databáze

Jedná se o bázi dat založenou na relačním modelu. Umožňuje definici datových struktur, úschovu dat, výběr dat nebo použití integritních omezení. Data jsou organizována v tabulkách, což jsou vlastně kolekce záznamů [20]. Dotazovací jazyk (SQL) umožňuje požádat o specifikované záznamy z jedné tabulky nebo i z několika tabulek, pokud mezi nimi existuje relace.

5.2 Softwarové produkty

5.2.1 Metodologie výběru uchazečů

- Sanna

Autor: Pavel Urban

Doplňek aplikace Microsoft Excel poskytující nástroj pro vícekritériální analýzu metodami WSA, TOPSIS, ELECTRE, PROMETHEE a MAPPAC.

<http://attofinn.webpark.cz/sanna/czech.html>

- Zákon 4.0.16 - FreeData

Počítačová škola Bárta, spol. s r. o., 2005

Elektronická právnická knihovna obsahující téměř všechny naše platné zákony a vyhlášky.

<http://www.pravnipredpisy.cz/>

5.2.2 Analýza a návrh systému

- **Microsoft® Word 2002, SP-2**

Microsoft Corporation, 1983-2001

Textový editor pro tvorbu textové části analýzy a návrhu.

- **Microsoft® Visio® for Enterprise Architects, ver. 10.0.2705**

Microsoft Corporation, 1991-2001

Tvorba obecných schémat.

- **Microsoft® Office Excel 2003 SP1**

Microsoft Corporation, 1985-2003

Tvorba demonstračních grafů.

- **Geoffrey Spark's Enterprise Architect, ver. 4.10**

Sparkx Systems, 1998-2004

Tvorba schémat v jazyce UML.

- **XML Spy, ver. 4.3.**

Altova GmbH & Altova, inc., 1998-2001

Editace XML souborů, validace podle schémat, generování schémat.

6. Analýza problémové oblasti

Cílem této práce je navrhnout systém podporující výběr uchazečů o zaměstnání. Proces výběru pak bude realizován spoluprací dvou subjektů:

- systém pro hodnocení uchazečů,
- okolí systému,

přičemž okolím systému rozumíme:

- operátora zadávajícího do systému relevantní informace o uchazečích (resp. zaměstnancích),
- obsluhu, zadávající do systému informace o kritériích pracovních pozic a prediktorech měřících tato kritéria – jedná se o experty z řad personalistů, psychologů atp. (ideálně ve spolupráci se znalostním inženýrem, který by řešil technickou stránku věci),
- manažery (nebo obecně rozhodovatele), kteří fakta o uchazečích, popisy pracovních pozic a znalosti způsobu odvození vhodných uchazečů využívají při podpoře rozhodování,
- externí datové zdroje, jimiž je myšlen zejména IS pro řízení lidských zdrojů, který byl v organizaci již dříve nasazen. Zpravidla se jedná o tzv. HRMS (Human Resource Management System). Ten v sobě integruje administrativní funkce jako výpočet výplat, odvodů daní a pojištění, nástroje pro analýzu výkonnosti pracovníků apod.

Do širšího okolí systému spadají také odborníci, kteří vypracovali dotazníky a testy předkládané uchazečům.

Analýza bude pochopitelně sloužit pouze jako příprava návrhu systému, nikoliv popsaného okolí. Na druhou stranu, jelikož je tato práce zaměřena na vývoj SW produktu podporujícího manažery lidských zdrojů při výběru uchazečů, jeví se jako účelné nejprve popsat firemní procesy, které s takovým výběrem souvisí. Bez této úvodní studie by bylo sotva možné systém optimálně navrhnout.

6.1 Právní rámec

V tuzemsku jsme v posledních letech svědky, jak problematika ochrany práv uchazečů nabývá na společenském významu. Proto je důležité mít při ve všech fázích životního cyklu navrhovaného systému tuto otázku neustále na paměti. Níže jsou uvedeny právní normy relevantní k dané problematice.

Zákon o ochraně osobních údajů a o změně některých zákonů č. 101/2000 Sb.

Zákon mimo jiné stanovuje povinnosti správce (tj. zaměstnavatele) při zpracování osobních údajů subjektu údajů (tj. uchazeče nebo zaměstnance). Uvědomme si, že se v našem případě budou zpracovávat také tzv. citlivé osobní údaje o zdravotním stavu uchazeče, což je sice za účelem dodržení povinností a práv správce odpovědného za zpracování v oblasti pracovního práva a zaměstnanosti dovoleno (§9 pís. d), avšak smí se tak činit pouze s výslovným souhlasem subjektu a na dobu nezbytně nutnou.

Zákoník práce

Mnohokrát novelizovaný zákon, původně z roku 1965, ošetřuje pracovněprávní vztah mezi zaměstnancem a zaměstnavatelem. V současnosti se připravuje jeho zcela nová podoba. Pro nás je nicméně důležité, že novelou z roku 2004 došlo v §1 k rozšíření zákazu přímé i nepřímé diskriminace s přesnější definicí, co je tím myšleno. Při hodnocení uchazečů tudíž výslovně nesmíme brát v potaz taková kritéria jakými jsou pohlaví, sexuální orientace, rasový, etnický či sociální původ, zdravotní stav, rodinný stav nebo věk.¹

Důležité je si uvědomit, že diskriminace je nejen nezákonná a nemorální. Jejím, byť skrytým, prosazováním bychom navíc často nedosáhli optimálního výběru. Uchazeč je například diskriminován kvůli věku, jelikož zaměstnavatel uvažuje vysokou korelaci mezi prediktorem věku a sníženou schopností učit se nové věci. Přitom je ale proveditelné otestovat toto kritérium přímo.

¹ Přesnější výklad je takový, že uvedená kritéria je zakázáno uvažovat, pokud nejsou v přímé souvislosti s vykonávanou prací. Při obsazování pozice šatnářky v dámských šatnách pochopitelně pohlaví roli hraje. Zaměstnavatel ovšem musí toto dokázat [6].

Právo v Evropském společenství

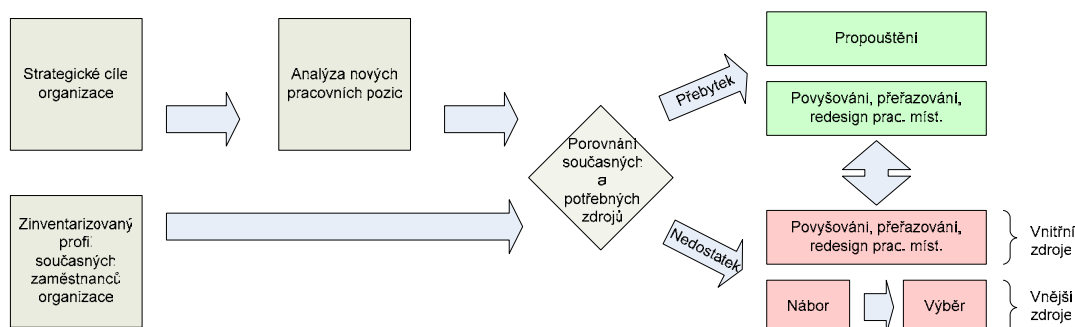
Základem definice práv občana Evropské unie je Evropská konvence o lidských právech a její dodatkové protokoly. Základní definici občanství Unie obsahuje Maastrichtská smlouva, členské státy mají odtud a také díky doplňující Amsterodamské smlouvě povinnost respektovat lidská práva a svobody jednotlivce pod pohrůžkou sankcí. V rámci celé Unie je zaručena jak ochrana dat vyžadovaných od uchazečů při vstupních testech a pohovorech, tak ochrana před diskriminací [55]. Úprava se problematikou zabývá daleko obšírněji než v případě právních norem ČR zmíněných výše.

6.2 Řízení lidských zdrojů v organizaci

Každá organizace průběžně provádí strategické plánování lidských zdrojů. Jedná se o činnost, při které se plánování v této oblasti dává do souladu s obecně vytýčenou strategií organizace. Na jedné straně existují strategické cíle organizace (definované většinou na pět až dvacet let dopředu), na druhé straně profil současných zaměstnanců organizace [6]. Porovnáním těchto dvou aspektů může vedení dospět ke dvěma závěrům:

- Vzhledem k budoucnu existuje **přebytek** určitých pracovních pozic a je třeba je utlumit. Jenom stručně uvedme, že zaměstnance lze kromě krajní možnosti propuštění také přeškolit na jiný druh práce, určitý podíl může vyřešit spontánní fluktuace, odchody do důchodu apod.
- Čelíme-li **nedostatku** specifického typu práce, měli bychom otevřít nové pracovní pozice a zaplnit je. Tento závěr přímo souvisí s navrhovaným systémem. Na základě náboru se totiž většinou přihlásí více zaměstnanců, anebo ne všichni zaměstnanci vyhovují požadavkům pozice. Proto je posléze nutné provést výběr.

Řízení lidských zdrojů, neboli personalistiku, lze tedy stručně definovat jako obsazování pozic v organizační struktuře (získávání, stabilizování, propouštění), a snahu o optimální využití lidských zdrojů v těchto pozicích [32].



Obrázek 3 – Proces řízení lidských zdrojů v organizaci.

Po zjištění nedostatku specifické pracovní síly se provádí důkladná analýza takové pracovní pozice. Z ní se odvodí, koho vlastně organizace hledá a jaké budou zaměstnanci přidělovány pracovní úkoly. O možnostech řešení přebytku jsme si již zmínili. Nedostatek lze řešit buďto z vnitřních anebo vnějších zdrojů.

Vnitřní zdroje

V případě vnitřních zdrojů je dána přednost zájemci z řad vlastních zaměstnanců. Může se jednat o přeřazení na jinou pozici – povýšení či přemístění na jiný odlišný práce. Pracovní úkoly můžeme pracovníkovi také pouze rozšířit, a to horizontálním směrem (přidělení dalších souvisejících pracovních činností – řidič autobusu začne například rovněž prodávat jízdenky) nebo vertikálním směrem (navýšení zodpovědnosti). Kontinuální změny profilu pracovní pozice nazýváme redesignem pracovních míst [29].

Vnější zdroje

Rozhodne-li se organizace využít vnější zdroje (tj. mimopodnikové), nejčastěji zorganizuje **nábor** a posléze **výběr** vhodných zaměstnanců. Existují ale i alternativy – lze si například najmout sezónní brigádníky od zprostředkovatelských firem, řešit nedostatek zaměstnanců jejich leasingem u specializované firmy, či nasmlouvat nezávislé pracovníky (konzultanty atp.) jenom na dílčí projekt. Výhody i nevýhody takového řešení jsou zřejmé – odpadají problémy s nepružností zaměstnaneckého poměru, na druhou stranu ale do takového pracovníka těžko budeme investovat a budovat s ním dlouhodobější vztah.

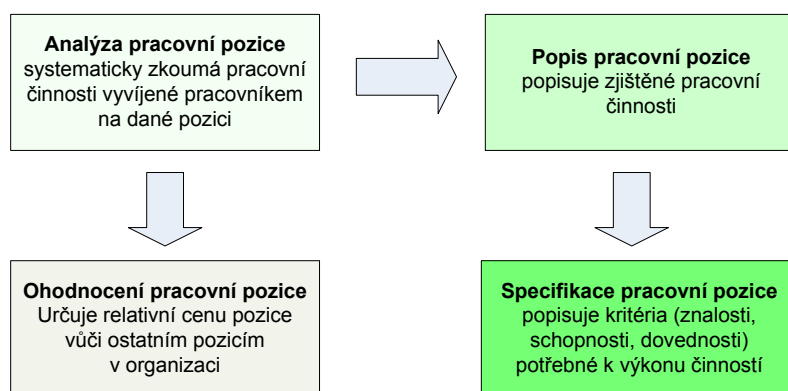
6.3 Analýza pracovní pozice

Analýza pracovní pozice (Job Analysis) je technická procedura, kterou se získávají informace o určitém pracovním místě. Jedná se o zjišťování, zaznamenávání, uchovávání a analyzování informací o úkolech, metodách, odpovědnosti, vazbách na jiné pozice, podmínkách apod. [29].

Tyto informace můžeme získat pozorováním pracovníka na dané pozici při práci, dotázaním pracovníka v interview, uspořádáním skupinového interview s několika takovými pracovníky, obsláním strukturovaného dotazníku, pověřením pracovníka, aby své činnosti zaznamenával do diáře apod. Existují také speciální techniky jako DOL (The Department of Labor's Job Analysis Process) nebo PAQ (Position Analysis Questionnaire), které se snaží pracovní pozici jednotně kategorizovat [6].

Samozřejmě nastává i případ, že pozice, kterou analyzujeme, dosud neexistuje. I tak je analýza, byť s většími problémy, proveditelná. Využívají se při ní poznatky lidí, kteří jsou s povahou vznikající pozice obeznámeni.

Analýza slouží jako podklad pro vypracování dalších přehledů o pracovním místě, a sice popisu, specifikace a ohodnocení pozice (viz Obrázek 4). Její existence je přínosná i pro kandidáta, který díky ní získá všeobecné povědomí o tom, co nabízená pozice obnáší. Hledají se také podobnosti a rozdíly mezi novou pozicí a nějakou příbuznou pozicí, která již existuje [14].



Obrázek 4 – Analýza pracovní pozice slouží jako podklad pro další přehledy.

Popis pracovní pozice

Popis pracovní pozice (Job Description) je písemným přehledem, který kromě názvu pozice obsahuje činnosti, rozsah zodpovědnosti a povinnosti spadající

pod uvažované pracovní místo. Například odborný prodavač prodává zboží zákazníkům (vč. cizinců), vysvětluje jim funkci zboží, je zodpovědný za přijatou tržbu a svým způsobem firmu reprezentuje.

Specifikace pracovní pozice

Z činností uvedených v popisu lze odvodit kritéria určující úspěšného vykonavatele těchto aktivit. Jedná se o znalosti, schopnosti a dovednosti (anglicky KSA – Knowledge, Skills, Abilities). Takový výčet je obsahem specifikace pracovní pozice (Job Specification). Specifikace pracovní pozice „odborný prodavač“ bude tedy klást důraz na kritéria jako jsou: schopnost jednat s lidmi (obchodní talent), jazykové znalosti na základní nebo komunikativní úrovni, technické uvažování, zodpovědnost, reprezentativní vzhled.

Ohodnocení pracovní pozice

Analýza pozice slouží také jako podklad pro ohodnocení pracovní pozice (Job Evaluation). Hodnocením jednoduše řečeno zjišťujeme, jakou pozice (potažmo pracovník na této pozici) představuje pro organizaci hodnotu. Hodnocení má význam při tzv. kompenzaci. Kompenzace je proces, jímž nastolujeme rovnováhu mezi zájmem organizace o vyrovnaný rozpočet a snahou udržet a stimulovat zaměstnance pomocí mezd [22]. Během výběru zaměstnanců je pro nás tato informace také důležitá, neboť si většinou nepřejeme přijmout jinak vyhovujícího uchazeče s neodpovídajícími mzdovými požadavky.

6.4 Nábor uchazečů

Jakmile disponujeme specifikací pracovní pozice, můžeme přistoupit k náboru². Nábořem (Recruitment) označujeme činnost získávání základního souboru uchazečů, nad kterým potom provádíme další výběr (viz kap. 6.4). Lze při tom využít následující zdroje:

- inzerce (novinová, na Internetu, letáková),
- najmutí personální agentury,

² Pomineme organizace, které provádějí nábor pomocí inzerátu, aniž by pracovní pozici blíže specifikovaly.

- spolupráce se středními školami a univerzitami, profesními organizacemi, úřady práce atp.,
- doporučení stávajících zaměstnanců,
- spontánní zájem nevyžádaných uchazečů.

6.5 Výběr uchazečů

Výběrem uchazečů (Selection) se rozumí zúžení základní množiny uchazečů získaných nábořem (kap. 6.5) a jejich propojení s konkrétní pracovní pozicí. Jde tedy o oboustranný proces. Kvalitní výběr je pouze mezicílem. Jak nastíníme v kapitole 6.5.1, konečným cílem je, aby budoucí zaměstnanec pracoval ke spokojenosti vedení organizace. Při výběru nám nezbyvá nic jiného, než se snažit tuto budoucí vlastnost uchazeče predikovat.

6.5.1 Charakteristiky uchazečů

Prediktory a kritéria

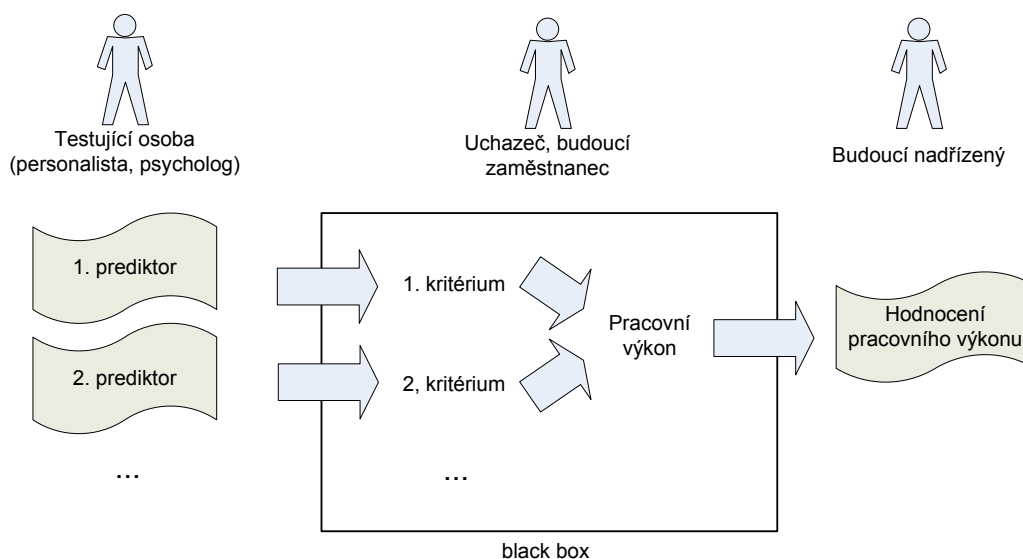
Nejprve definujme pojmy kritérium a prediktor. Kritérium je vlastnost uchazeče, která souvisí s pracovním výkonem a jejíž hodnotu se při výběru tudíž snažíme maximalizovat (minimalizovat). Může se například jednat o poctivost nebo schopnost domluvit se anglicky. Je zřejmé, že kritéria uchazečů nemůžeme měřit přímo. Poctivost uchazeče vyvozujeme z kladné reference od předchozího zaměstnavatele nebo třeba z testu integrity, schopnost domluvit se anglicky předpokládáme na základě předložení certifikátu. Těmto nástrojům nepřímého odhadu kritérií říkáme prediktory.

Základní charakteristiky kandidáta lze při troše důvtipu odvodit i bez použití testů – mnoho napoví úprava zasláného životopisu a případný výskyt gramatických chyb, včasný příchod na smlouvanou schůzku nebo síla stisku podané ruky. Jak takové odvození vlastně probíhá? Personalista podle gramatických chyb v životopise patrně usoudí, že uchazeč buď o nabídku nestojí natolik, aby životopis připravil pečlivě, anebo je kvalitnější příprava limitována jeho omezenými schopnostmi. Tato inference probíhá mnohdy zcela spontánně na základě předchozích zkušeností a přesvědčení personalisty.

Pracovní výkon a efektivnost lidské práce

Pracovní výkon (Job Performance) vyjadřuje míru úspěšného vykonávání pracovní činnosti, neboli jak je ten který pracovník přínosný pro organizaci. U dělníka na výrobní

lince je výkon ztotožňován s počtem vyrobených kusů, u obchodníka s dosaženou tržbou atp. Zobecníme-li tento pojem, jedná se vlastně o míru spokojenosti vedení organizace s tím, jak zaměstnanec svěřené pracovní úkoly zvládá. Nemusí se jednat o výkon v pravém slova smyslu, někdy klademe důraz na kvalitu (vzpomeňme na problém vysoké kvantity úkonů po zavedení bodovacího systému v českém zdravotnictví) nebo na spokojenost zákazníka. Vztah mezi prediktory, kritérii a celkovým pracovním výkonem z těchto kritérií (potažmo i prediktorů) plynoucím ukazuje Obrázek 5.



Obrázek 5 – Závislost pracovního výkonu na kritériích, potažmo změřených prediktorech.

Pracovní výkony měříme především kvůli podkladům pro konstruktivní hodnocení zaměstnanců. Změřený výkon pracovníků však může posloužit také ve zpětné vazbě jako ukazatel toho, zda výběr uchazečů byl proveden kvalitně. To je pro naše účely daleko zajímavější.

Při měření pracovního výkonu se stejně jako při výběru uchazečů zjišťují hodnoty prediktorů, od nichž se snažíme odvodit hodnoty kritérií. Těmito kritérii mohou být:

- objem výkonu (počet vyrobených kusů, tržby atp.), stabilita výkonu,
 - absence, pozdní příchody,
 - počet vyprodukovaných zmetků, nehodovost,
 - rychlost vyřízení zakázky,
 - schopnost učit se (prediktorem je test indikující úspěch prodělaného školení)
- [14],

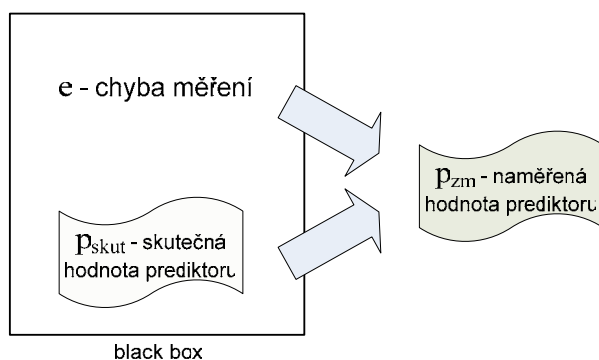
- spokojenost zákazníka (prediktorem jsou výsledky průzkumu spokojenosti zákazníků),
- stabilita zaměstnání, délka trvání pracovního poměru (příp. vertikální mobilita) jako opak fluktuace³.

Celkový pracovní výkon představuje souhrn těchto kritérií. Používané metody jsou obdobné jako při samotném výběru uchazečů.

Zahrneme-li do nastíněného modelu také otázku mzdy pracovníka, získáme pojem efektivnost lidské práce, kterým rozumíme dosažení maximálního zisku (tj. pracovního výkonu) při vynaložení minimálních nákladů (tj. zejména platu pracovníka) [29]. Pokud zaměstnanec za svou práci požaduje nepřiměřenou odměnu, stává se pro organizaci nepřijatelný, i kdyby byly jeho ostatní kvality nadprůměrné.

Na závěr uveďme, že v praxi se často nezkoumá korelace mezi výsledky přijímacích testů a následným pracovním výkonem u všech pracovníků, ale zpracovávají se pouze data u množiny elitních pracovníků (tzv. benchmarking) [46].

Spolehlivost měření prediktorů



Obrázek 6 – Naměřená hodnota prediktoru je kompozicí skutečné hodnoty a chyby měření.

Prediktory určitým způsobem měříme (naměřenou hodnotu představuje např. skóre testu) a při měření obecně vznikají chyby. Obrázek 6 ukazuje, že do výsledku měření

³ Existují situace (např. u prvně zaměstnaných absolventů VŠ), kdy ti nejschopnější, kteří svou práci ovládají nejlépe, paradoxně odcházejí nejdříve, jelikož se jim naskytne výhodná příležitost jinde [6].

prediktoru p_{zm} se promítá nejen skutečná hodnota prediktoru $p_{skut.}$, ale také chyba měření e , podle vztahu

$$p_{zm.} = p_{skut.} + e \quad (1)$$

Chyby měření mohou pocházet z těchto zdrojů [14]:

- Uchazeči odpovídají na dotazy ovlivněni svou individualitou. Na stejný dotaz odpoví každý poněkud jinak.
- Co osoba provádějící měření, to jiný styl pokládání otázek, vč. nevědomého ovlivňování uchazečů. Jedná se o odchylky na cestě *testující* → *testovaný*.
- Co testující, to jiná subjektivní měřítka hodnocení získaných odpovědí. Jedná se o odchylky na cestě *testovaný* → *testující*.
- Proces testování ovlivňují také okolní podmínky (hluk, povětrnostní podmínky atp.).

Spolehlivost měření představuje míru stálosti naměřených hodnot. Je-li měření stálé, měli bychom při jeho opakování dojít k obdobným výsledkům. Spolehlivost měření můžeme vyjádřit koeficientem spolehlivosti, příp. z tohoto koeficientu odvozenou standardní chybou měření.

Koeficient spolehlivosti je důležitým parametrem testu, proto si přibližme způsob jeho výpočtu. Označme *teoretickým rozptylem* s_Y^2 rozptyl hodnot Y_i nějakého rysu (např. inteligence):

$$s_Y^2 = \frac{1}{n} \sum (Y_i - \bar{y})^2, \quad (2)$$

kde Y_i je skutečná hodnota rysu toho kterého uchazeče a \bar{y} je průměrná hodnota. Sestavíme-li za účelem měření rysu nějaký test, pak můžeme *empirickým rozptylem* s_y^2 popsat rozptyl naměřených hodnot y_i uvažovaného kritéria:

$$s_y^2 = \frac{1}{n} \sum (y_i - \bar{y})^2, \quad (3)$$

kde y_i je hodnota rysu uchazeče naměřená naším testem [21]. Rys uchazeče je relativně stálý. Uchazeč disponuje dejme tomu inteligencí na úrovni IQ 110. Testem ale

můžeme naměřit hodnotu 106 a za týden třeba 114. Empirický rozptyl s_y^2 tedy bývá větší než teoretický rozptyl s_y^2 . Kvalitu testu můžeme vyjádřit indexem determinace

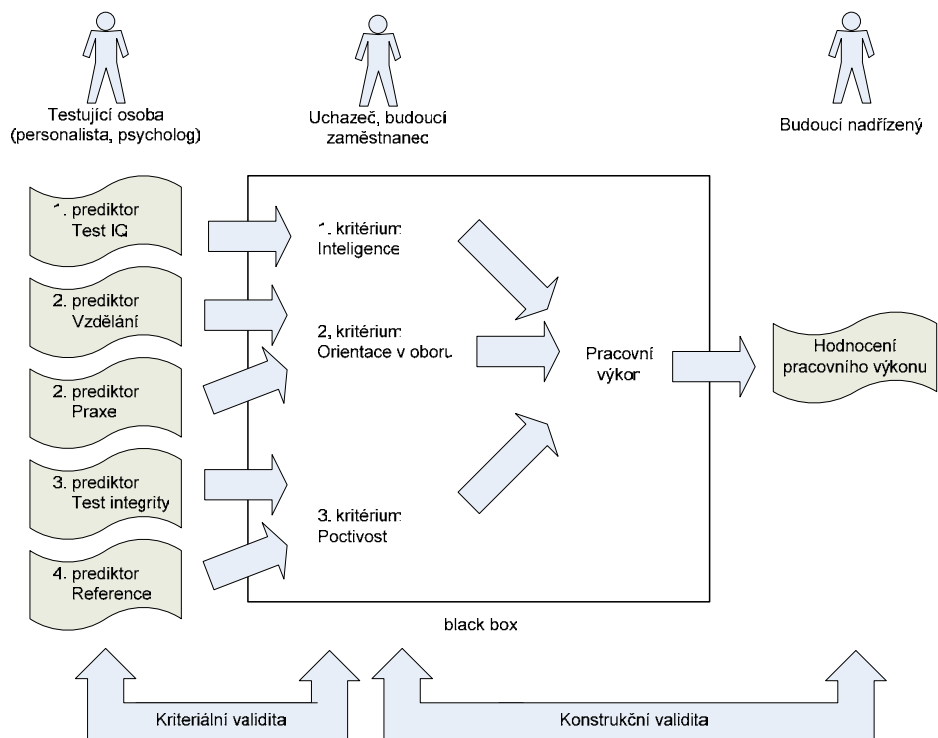
$$I_{yx}^2 = \frac{s_Y^2}{s_y^2}, \quad (4)$$

což je jen jiný název pro koeficient spolehlivosti r_{xx} [14]. Koeficient spolehlivosti je roven jedné, pokud je test naprosto spolehlivý, naopak jestliže jsou skutečné a naměřené hodnoty nekorelované, rovná se nule.

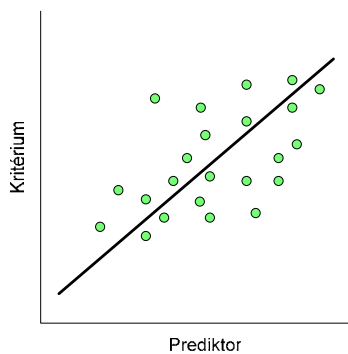
Validita prediktorů

Jak tušíme, vztah mezi pozorovaným jevem (prediktorem) a skutečnou intenzitou vyskytující se vlastnosti (kritériem, anebo přímo budoucím pracovním výkonem jako souhrnem kritérií), nemusí být nijak silný. Příkladem z toho plynoucí chyby při posuzování druhých budiž tzv. "haló efekt" jakožto důležitý, avšak subjektivně zabarvený prvek prvotního vnímání [5]. Intenzitu vztahu mezi hodnotou prediktoru (např. dosaženým skóre v testu z anglického jazyka) a hodnotou relevantního kritéria (např. schopností domluvit se anglicky) označujeme jako kritériální validita (Criterion-Related Validity). Obdobně lze rozlišovat další druhy validit. Kupříkladu obsahová validita (Content Validity) popisuje, nakolik obsah testu (např. problémový okruh, z něž jsou syntetizovány testové otázky) koresponduje s reálnými situacemi, vyskytujícími se v dané pracovní pozici. Konstrukční validita pak představuje vztah mezi abstraktním kritériem (např. inteligencí) a skutečným pracovním výkonem v praxi (viz Obrázek 7) [6]. Vymeźme tedy obecný pojem validita prediktoru jakožto míru kvality prediktoru ve vztahu k výslednému pracovnímu výkonu.

Vztah mezi prediktorem a kritériem (potažmo celkovým pracovním výkonem) lze aproximovat regresní funkcí (viz Obrázek 8). V případě, že je výsledné kritérium ovlivňováno více prediktory, použijeme vícenásobnou regresní funkci. Výsledek regresní funkce (závislá proměnná) pak závisí na několika parametrech (nezávislých proměnných).



Obrázek 7 – Kriteriální a konstrukční validita



Obrázek 8 – Body znázorňující vztah *prediktor* → *kritérium* a křivka regresní fce.

Kriteriální validita (validita vztahu mezi prediktorem a kritériem) odpovídá kvalitě této regresní funkce. Kdybychom provedli měření hodnoty prediktoru a zároveň měření hodnoty kritéria (pomocí nějakého dalšího, absolutně spolehlivého prediktoru) na vzorku kandidátů, validitu bychom mohli kvantifikovat například pomocí indexu korelace [21]:

$$I_{yx} = \sqrt{\frac{\frac{1}{n} \sum (y_i - \bar{y})^2}{\frac{1}{n} \sum (Y_i - \bar{y})^2}}, \quad (5)$$

kde n je rozsah vzorku, y_i je empirická hodnota zjištěná prediktorem a Y_i je teoretická hodnota kritéria.

Funkční závislost pracovního výkonu

Předpokládejme, že známe naměřené hodnoty prediktorů. Abychom mohli jednotlivé uchazeče seřadit podle předpokládaného pracovního výkonu, potřebujeme ještě zjistit funkční závislost mezi hodnotami prediktorů a tímto pracovním výkonem.

Spolehlivost měření uvažujme jako součást validity. Dále necht'

- m je počet prediktorů,
- n je počet zjišťovaných kritérií,
- p_x , $x = 1 \dots m$ je hodnota x -tého prediktoru,
- k_y , $y = 1 \dots n$ je hodnota y -tého kritéria,
- V je předpokládaná hodnota pracovního výkonu.

Hledáme následující dva funkční vztahy:

- $f: K = f(P)$, (6)

kde $P \in \{p_1, p_2, \dots, p_m\}$ je množina hodnot měřených prediktorů a $K \in \{k_1, k_2, \dots, k_n\}$ je množina předpovídaných hodnot kritérií,

- $g: V = g(K)$, (7)

představující závislost pracovního výkonu na hodnotách kritérií.

Protože předpokládáme, že obě zobrazení jsou bijektivní, mohli bychom celý problém zjednodušit tak, že bychom uvažovali pouze složené zobrazení

(8)

$$h = g \circ f$$

s poukazem na to, že hodnoty kritérií v mezikroku nás nezajímají, jelikož nejsou přímo měřitelné. Na druhou stranu, obsluze systému by se zřejmě lépe zadávaly obecné závislosti *prediktor(y) → kritérium* a posléze vztah *kritéria → výkon* pro určitou pracovní pozici, než celý složitější vztah *prediktor(y) → výkon* znovu s každou novou pozicí.

6.5.2 Hodnoty prediktorů a jejich zjišťování

Odpověď na otázku, jaké hodnoty nabývá určité kritérium u určitého uchazeče, získáváme různými cestami. Jestliže výběr probíhá metodou postupného vyřazování (viz kap. 6.6.2), hovoří se v literatuře o tzv. krocích výběru [9]. Jedná se pak zejména o následující kroky výběru [6]:

- **Předběžný pohovor** (Initial Screening) – slouží k odfiltrování naprosto nevhodných uchazečů. Často se jedná o zahájení vzájemného kontaktu mezi uchazečem a organizací.
- **Podnikový dotazník** – organizace se v něm táže jak na životopisná fakta, tak na dovednosti potřebné k výkonu specifické práce. Dotazník bývá vypracováván na základě analýzy pracovního místa.
- **Podrobný pohovor** – může jich být realizováno několik s různými pracovníky organizace klíčovými pro zaplňovanou pozici. Tito účastníci rozhovoru se snaží předpovědět, jak by se uchazeč hodil na zamýšlenou pozici a zda by si rozuměl s příslušným týmem.
- **Testování** – zahrnuje celou řadu testů, viz odborná literatura [14]. V historii se testům střídavě přikládala různá váha. V současnosti převažuje tendence testy nezavrhovat, pokud mají přímou souvislost s vykonávanou prací [32].
Namátkou uvedme:
 - Test **fyzických** nebo **mentálních schopností**.
 - **Psychomotorický test** – zkoumá předpoklady pro práci ve výškách.
 - **Test osobnosti** (např. typologie osobnosti, Myers-Briggs nebo MMPI)[27]. Odhaluje osobní vlastnosti a schopnost sociální interakce.
 - **Test inteligence** – hodnotí schopnost vidět vzájemné souvislosti ve složitějších problémech [32].

- **Situační test** – uchazeč je na krátkou dobu povolán plnit vzorek pracovního úkolu, který by mu měl být v budoucnu přidělen trvale.
- **Hodnotící střediska** (Assessment Centers) – řeší se v nich typické úkoly a problémy, které mohou v praxi nastat, vč. týmové spolupráce s ostatními uchazeči. Kandidáti jsou pozorováni psychology a zkušenými manažery [32].
- **Test na přítomnost drog** v organizmu (Drug Screening) - toxikologické vyšetření vzorku moči nebo vlasů.
- **Detektor lži** – V USA je povolen a využíván pouze při náboru do bezpečnostních složek.
- **Test integrity osobnosti** [14] - provádí se jen pomocí papíru a tužky, zkoumá přítomnou obdobnou oblast jako v případě detektoru lži.
- **Prověření informací** (Background Investigation) – provádí se po dokončení předchozích bodů. Zahrnuje verifikaci referencí⁴, dosaženého vzdělání, negativního záznamu v rejstříku trestů atp. Tento krok každopádně vede ke zvýšení spolehlivosti prověřovaných prediktorů. Hodnotu prediktoru lze vyjádřit datovým typem *boolean* a jeho nesplnění (zjištění nesrovnalostí) obvykle nelze jinými prediktory kompenzovat.

Jako další krok se v některých případech provádí ještě podrobné lékařské vyšetření. Děje se tak však až po výběru uchazeče, proto se tímto faktorem nebudeme zabývat.

Otázka, které faktory zahrnout do hodnocení, je sama o sobě rozhodovací úlohou. Hledáme při ní kompromis mezi hodnotami výsledků, které by nám ten který test mohl přinést a jeho finanční a časovou náročností.

Levným a efektivním nástrojem je spontánní výběr uchazečů, probíhající vlastně již v rámci náboru (kap. 6.4). Například inzerát v novinách v anglickém jazyce automaticky vyřadí uchazeče s nedostatečnou znalostí angličtiny. Mimoto, specifikace pracovního místa v inzerátu by měla být podána objektivně, včetně nepříjemných stránek výkonu povolání.

⁴ Před prověřováním referencí bychom si vždy měli vyžádat souhlas uchazeče [9].

Na takto podanou nabídku pak reagují spíše uchazeči, kteří by takovou práci mohli trvale snášet k oboustranné spokojenosti.

Sofistikovanějším prediktorem výběru jsou komerčně dostupné testy, jejichž vytváření, shromažďování, ověřování validity a prodej jsou předmětem činnosti řady subjektů. Například internetové stránky neziskové organizace *ETS* [10] nebo *Burros institute of Mental Measurements* při *University of Nebraska-Lincoln* [52] nabízejí vyhledání libovolného testu z rozsáhlé databáze a i jejich následné získání.

Jedním z atributů každého komerčního testu je minimální požadované vzdělání nebo praxe testujícího odborníka. Některé testy může uchazečům předkládat kdokoli, tedy i např. nižší manažer při doplňování svého týmu, v případě jiných je vyžadováno univerzitní vzdělání v příslušném oboru.

Někteří autoři zdůrazňují, že kromě exaktně získaných výsledků testů by měla být do rozhodovacího procesu v nezbytné míře zapojena také intuice a „zdravý rozum“ [29]. Nebudeme se tedy bránit ani tomu, aby jako jedno z kritérií figuroval vstup, který se snaží subjektivní míru přijatelnosti uchazeče vyjádřit na nějaké škále pomocí intervalové proměnné. Uchazeče mohou takto hodnotit třeba účastníci podrobného rozhovoru. Mějme však na paměti, že tento subjektivní prvek by mohl být napadnutelný jako diskriminující.

Klasifikace hodnot prediktorů

Třída		Typ	Příklad
Kategoriální	Kvalitativní	Nominální	Klasifikace typu vzdělání (VŠ ekonom. směru, technického směru atp.)
		Ordinální	Percentil uchazeče získaný v testu realizovaném na větším počtu uchazečů.
Spojitě	Kvantitativní	Intervalové	Znalost AJ hodnocená na nějaké škále (1-žádná, .., 5-komunikativní, .., 9-expertní).
		Poměrové	Počet slov napsaných za jednu minutu při testu dovednosti psaní na PC.

Tabulka 1 – Klasifikace proměnných udávajících výsledek testu.

Typy proměnných používaných obecně při statistické analýze se liší svojí výpovědní hodnotou. Proměnné nominálního typu nelze setřídít. Ordinální typ setřídít lze, avšak vzdálenosti mezi hodnotami proměnné nedávají žádný smysl. U intervalového typu

jsou naopak intervaly brány v potaz. Proměnná poměrového typu na rozdíl od typu intervalového specifikuje i o to, kolikrát je jedna hodnota větší než druhá [48].

Způsob, jakým dochází k hodnocení uchazečů, by měl být ve snaze minimalizovat chybu měření co nejpodrobnější. Kupříkladu ordinální údaj o prospěchu uchazeče v profilových předmětech během vysokoškolských studií představuje jednoznačně kvalitnější údaj než výrok typu *boolean* o vykonaném absolutoriu.

Standardní skóre

Pravděpodobnost výsledku testu reprezentovaná percentilem uchazeče v nějakém výběrovém souboru (standardizovaném vzorku populace), bude zřejmě vykazovat znaky normálního rozdělení. Avšak nevýhoda percentilu je v jeho nelinearitě – rozdíl v intenzitě měřeného znaku je mezi 10. a 20. percentilem daleko větší než mezi 40. a 50. percentilem, což plyne z charakteristického “zvonovitého“ tvaru křivky hustoty normálního rozdělení. Výstupy zmíněných komerčních testů jsou proto většinou transformovány do lineární (intervalové) škály, do jednotného, tzv. *standardního skóre* [55]. Transformační vztah je následující:

$$Y = \frac{X - M}{SD} . \quad (9)$$

Hodnota X představuje percentil, kterého uchazeč dosáhl, tj. poměr uchazečů, kteří jsou stejně úspěšní nebo horší než on sám. Tabulka níže udává nejpoužívanější typy standardních skóre a k nim hodnoty parametrů M a SD , které je třeba do vzorce dosadit.

Značka skóre	M	SD	Popis
Z	Střední hodnota souboru μ	Směrodatná odchylka souboru σ	Základní skóre udávající vzdálenost výsledku uchazeče od střední hodnoty v jednotkách rovnajících se σ . Průměrný uchazeč má skóre 0. Výsledkem je reálné číslo, což není vždy vyhovující.
T	$\mu - 500$	$0,1\sigma$	T-skóre má střední hodnotu 50 a směrodatnou odchylku 10. Průměrný uchazeč má pak skóre 50. 99% výsledků leží mezi hodnotami 20 a 80. Zaokrouhuje se na celá čísla a omezuje na interval [1,99]. Výsledek je tak redukován na množinu přirozených čísel.
Stanine	$\mu - 2,5$	$0,5\sigma$	Zkratka slov „standard nine“. Zaokrouhuje se na celá čísla {1, 2, 3, 4, 5, 6, 7, 8, 9}, která (vyjma krajních hodnot) představují intervaly o velikosti poloviny σ . Hodnota 5 tak představuje průměrného uchazeče (do této hodnoty jich spadá

			20%).
--	--	--	-------

Tabulka 2 – Nejpoužívanější typy standardních skóre.

6.6 Modely pro hodnocení variant

6.6.1 Obecné modely

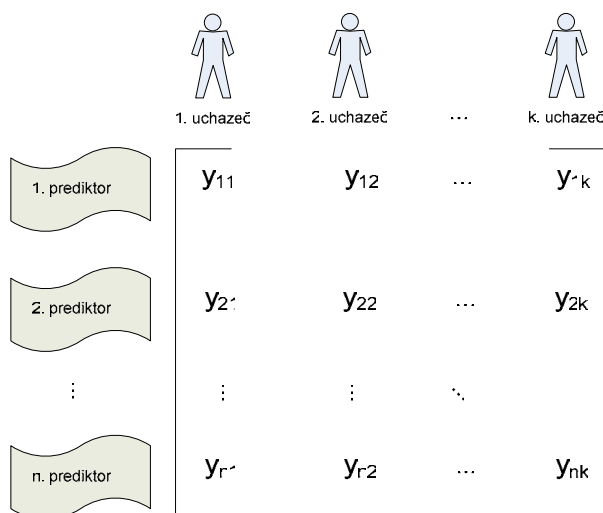
Výběr uchazečů lze pojmout jako praktickou aplikaci modelů pro vícekritériální rozhodování (MCDM). Tuto disciplínu můžeme rozdělit na:

- Rozhodování na základě několika stanovených cílů (MODM), kde nejsou nijak předurčeny konkrétní varianty.
- Hodnocení konkrétních variant na základě několika kritérií (MADM), což je i náš případ [28].

Nenechme se zmýlit odlišnou terminologií – kritérii v obecné rovině rozumíme to, co v managementu lidských zdrojů nazýváme prediktory (viz kap. 6.5.1). Uchazeči vždy představují jednotlivé varianty. Cílem může být:

- uspořádat varianty (uchazeče) podle předpokládaného pracovního výkonu,
- rozdělit varianty (uchazeče) do tříd (typicky do dvou tříd: *přijatelní uchazeči*, *nepřijatelní uchazeči*).

Východiskem budiž tzv. *kritériální matice* (viz Obrázek 9), která obsahuje hodnoty všech kritérií každé varianty [23], tj. všechny hodnoty prediktorů každého uchazeče.



Obrázek 9 – Kriteriální matice [23]

K řešení úloh vícekritériálního hodnocení variant lze s výhodou použít například systém SANNA (viz kap. 5.2.1). Popišme si nyní na jednoduchém příkladu, z jakých kroků řešení úlohy sestává.

Stanovení prediktorů

Na základě analýzy a následné specifikace pracovní pozice stanovíme kritéria a posléze prediktory pro jejich odhad například tak, jak to ukazuje následující tabulka.

Kritéria	Prediktor	Zkratka prediktoru	Škála hodnocení prediktoru
Odborné vzdělání	Vzdělání VŠ, minimálně SŠ technického směru	EDU	1 .. dokončené ÚSO 2 .. nedok. Bc. Stupeň 3 .. dokonč. Bc. stupeň 4 .. nedok. Mgr. stupeň 5 .. dokonč. Mgr. stupeň
Příslušná kvalifikace a pracovní zkušenosti	Praxe v IT oboru (min. 2 roky)	IT	Počet celých let
	Praxe v programování (min. 1 rok)	PRG	-"-
Odborné dovednosti, jazykové znalosti	Test znalosti objektového programování (C#, Java nebo C++)	OOP	1 .. žádná znalost až 9 .. výborná znalost
	Test znalosti DB aplikací (MS-SQL, Oracle atp.)	DB	-"-
	Interview v AJ (požadujeme znalost AJ alespoň na úrovni nutné pro studium technické dokumentace)	EN	1 .. - 3 .. elementary 7 .. good 9 .. excellent
Osobnostní charakteristiky a schopnosti	Situační test na schopnost algoritmického myšlení	ALG	1 .. žádná až 9 .. výborná
	Assessment centrum - týmová spolupráce	TW	-"-
	Assessment centrum - Odolnost proti stresu	SR	-"-

Tabulka 3 – Příklad prediktorů pro hodnocení uchazečů.

Všechny prediktory v tabulce jsou maximalizační, tzn. že čím vyšší hodnoty dosahují, tím lépe. U některých faktorů vidíme stanovené minimální hodnoty (např. minimálně dva roky praxe v IT). Varianty (tj. uchazeče), jejichž libovolný prediktor nabývá nižší hodnoty než minimální povolené (tzv. bazální) hodnoty, proto na začátku vyřadíme.⁵

Odhad vah prediktorů

Tento krok realizujeme v případě, neznáme-li relativní podíl (preferenci) jednotlivých prediktorů na předpovídaném celkovém pracovním výkonu. Popíšeme Bodovací metodu a Saatyho metodu.

Bodovací metoda předpokládá, že je rozhodovatel schopen ohodnotit důležitost prediktorů na nějaké stupnici. Na základě vztahu

$$vp_x = \frac{dp_x}{\sum_{i=1}^n dp_i} \quad (10)$$

kde dp_x je důležitost x -tého prediktoru ve škále $\{1, \dots, 10\}$, získáme váhu prediktoru vp_x v rozsahu $[0, 1]$.

Aktivace	Body	Váhy
EDU	4	0,08000
IT	2	0,04000
PRG	5	0,10000
OOP	7	0,14000
DB	3	0,06000
EN	4	0,08000
ALG	10	0,20000
TW	8	0,16000
SR	6	0,12000
SAL	1	0,02000

Obrázek 10 – Příklad vah prediktorů získaných bodovací metodou v systému SANNA.

Daleko přesnější **Saatyho metoda** [4, 23] předpokládá, že je rozhodovatel schopen provést porovnání i -tého a j -tého prediktoru b_{ij} ($i, j = 1..n, i \neq j$), a ohodnotit je v následující škále:

⁵ V následujících kapitolách 6.6.2 a 6.6.3 ukážeme, že existují i modely, které zdánlivě nepřijatelné uchazeče takto striktně nevyřazují.

Vztah <i>i</i> -tého prediktoru k <i>j</i> -tému prediktoru	Číslo
Rovnocenný vztah (indiference)	1
Slabá důležitost	3
Silná důležitost	5
Zjevná důležitost	7
Absolutní důležitost (preference)	9
Mezilehlé hodnoty	2, 4, 6, 8

Tabulka 4 – Škála pro hodnocení vztahu *i*-tého prediktoru k *j*-tému prediktoru.

Celkové hodnocení by mělo být konzistentní, tzn. že mezi hodnoceními jednotlivých dvojic by měl platit tranzitivní vztah. Ze získaných údajů sestavíme matici $B=(b_{ij})$, pro níž platí, že $b_{ii}=1$ a $b_{ji}=1/b_{ij}$. Jednotlivé složky w_i váhového vektoru W lze spočítat přesně iteračním způsobem, anebo přibližně vzorcem:

$$w_i = \left(\prod_{j=1}^n b_{ij} \right)^{\frac{1}{n}} / \sum_{k=1}^n \left[\left(\prod_{l=1}^n b_{kl} \right)^{\frac{1}{n}} \right]. \quad (11)$$

	EDU	IT	PRG	OOP	DB	EN	ALG	TW	SR	Váhy
EDU	1,00000	3,00000	1,00000	0,33333	1,00000	0,50000	0,14286	0,33333	0,50000	0,05375
IT	0,33333	1,00000	0,33333	0,14286	0,33333	0,16667	0,12500	0,25000	0,33333	0,02424
PRG	1,00000	3,00000	1,00000	1,00000	1,00000	1,00000	0,50000	1,00000	1,00000	0,09199
OOP	3,00000	7,00000	1,00000	1,00000	4,00000	3,00000	0,25000	2,00000	1,00000	0,15051
DB	1,00000	3,00000	1,00000	0,25000	1,00000	0,33333	0,14286	0,33333	0,33333	0,04757
EN	2,00000	6,00000	1,00000	0,33333	3,00000	1,00000	0,25000	0,33333	0,50000	0,08142
ALG	7,00000	8,00000	2,00000	4,00000	7,00000	4,00000	1,00000	2,00000	3,00000	0,30622
TW	3,00000	4,00000	1,00000	0,50000	3,00000	3,00000	0,50000	1,00000	2,00000	0,13699
SR	2,00000	3,00000	1,00000	1,00000	3,00000	2,00000	0,33333	0,50000	1,00000	0,10731

Obrazek 11 – Matice B a váhový vektor W.

Saatyho metodu použijeme i pro odhad vah prediktorů a kritérií v návrhu systému v kapitole 7.

Vícekritériální hodnocení

Závěrečná fáze obsahuje vlastní vícekritériální hodnocení variant (tj. uchazečů). Popíšme nejpoužívanější metody tak, jak jsou implementovány v systému SANNA.

Definujme pět uchazečů s hodnotami prediktorů, jak ukazuje Obrázek 12. V řádku *Váhy* jsou uvedeny váhy prediktorů podle Bodovací metody, viz Obrázek 10.

	EDU	IT	PRG	OOP	DB	EN	ALG	TW	SR
Novák	1	9	5	0	4	3	2	9	8
Černý	5	2	1	4	7	9	7	8	7
Králová	2	8	8	7	6	6	6	2	5
Vaněk	5	4	1	5	2	5	5	1	6
Příbyl	1	3	1	2	1	8	4	8	4
Váhy	0,08163	0,04082	0,10204	0,14286	0,06122	0,08163	0,20408	0,16327	0,12245
Ideální	5	9	8	7	7	9	7	9	8
Bazální	1	2	1	0	1	3	2	1	4

Obrázek 12 – Pět hodnocených uchazečů a příprava pro použití metody WSA.

Metoda WSA (Weighted Sum Approach) spočívá v tom, že nejprve pro každý prediktor zjistíme nejlepší a nejhorší (ideální a bazální) hodnotu v rámci zkoumaných uchazečů, viz poslední dva řádky tabulky. Dále vytvoříme tzv. *normalizovanou matici*. Původní kritériální matici v ní transformujeme vztahem

$$y'_{ij} = \frac{y_{ij} - D_j}{H_j - D_j}, \quad (12)$$

kde y_{ij} je původní hodnota j -tého prediktoru i -tého uchazeče, H_j je ideální a D_j bazální hodnota j -tého prediktoru.

	EDU	IT	PRG	OOP	DB	EN	ALG	TW	SR	u(variant)
Novák	0,00000	1,00000	0,57143	0,00000	0,50000	0,00000	0,00000	1,00000	1,00000	0,41545
Černý	1,00000	0,00000	0,00000	0,57143	1,00000	1,00000	1,00000	0,87500	0,75000	0,74490
Králová	0,25000	0,85714	1,00000	1,00000	0,83333	0,50000	0,80000	0,12500	0,25000	0,60641
Vaněk	1,00000	0,28571	0,00000	0,71429	0,16667	0,33333	0,60000	0,00000	0,50000	0,41642
Příbyl	0,00000	0,14286	0,00000	0,28571	0,00000	0,83333	0,40000	0,87500	0,00000	0,33916
Váhy	0,08163	0,04082	0,10204	0,14286	0,06122	0,08163	0,20408	0,16327	0,12245	

Obrázek 13 – Normalizovaná matice a součty dílčích užiteků v pravém sloupci.

Normalizovaná matice tedy pro každý prediktor každého uchazeče obsahuje skóre v intervalu $[0,1]$. Toto skóre představuje relativní pozici na mezi minimální (0) a maximální (1) hodnotou (viz Obrázek 13). V pravém sloupci je vypočítán vážený součet dílčích užiteků podle vztahu

$$u(X_i) = \sum_{j=1}^k v_j y'_{ij}, \quad (13)$$

kde X_i je i -tý uchazeč a v_j je váha příslušného prediktoru. Nyní již zbývá uchazeče seřadit podle celkových užiteků $u(X_i)$, tj. podle předpokládaného pracovního výkonu [23].

Je zřejmé, že obdobného efektu jako použitím metody WSA bychom dosáhli, kdybychom k predikci užítku použili vícenásobnou lineární regresní funkci, viz rovněž kap. 6.5.1, vztah (8):

$$u(X_i) = h(y_{i1}, y_{i2}, \dots, y_{in}). \quad (14)$$

Tato funkce by jednak zajišťovala transformaci každého prediktoru y_{ij} do intervalu $[0,1]$ a následně by také zohledňovala jeho váhu v_j . Výpočetně složitější, avšak principiálně obdobná s metodou WSA je i metoda **TOPSYS** [23].

Metoda **ELECTRE** vznikla jako odpověď na nedostatky jiných metod a postupně prošla vývojem od až do své 4. verze. Zatímco jiné metody porovnávají, zda je určité kritérium určité varianty jednoznačně (byť nepatrně) lepší nebo horší, ELECTRE je v tomto ohledu citlivější. Prvním krokem je stanovení prahů. Práh preference udává, o kolik musí být hodnota prediktoru prvního uchazeče v porovnání s druhým uchazečem ve dvojici relativně vyšší, abychom jej jednoznačně preferovali. U prahu dispreference se naopak jedná o jednoznačné odmítnutí takové preference [3].

Určitou podobnost s metodou ELECTRE vykazuje například ruská metoda **ZAPROS**, ve které se rozhoduje na základě verbálních otázek pokládaných rozhodovateli. Rozhodovatel má za úkol na ordinální škále subjektivně určit míru rozdílnosti vždy mezi dvěma variantami. Taková binární relace může vyjadřovat buď preference, indifferenci, nebo varianty ve dvojici mohou být neporovnatelné. Výstupem není přesné, ale pouze přibližné pořadí variant. Metoda se snaží přihlížet k vágní povaze vstupu, která je neodmyslitelně spjata s lidským faktorem. Výsledek je tedy méně přesný než při použití exaktních metod, avšak měl by být spolehlivější [38].

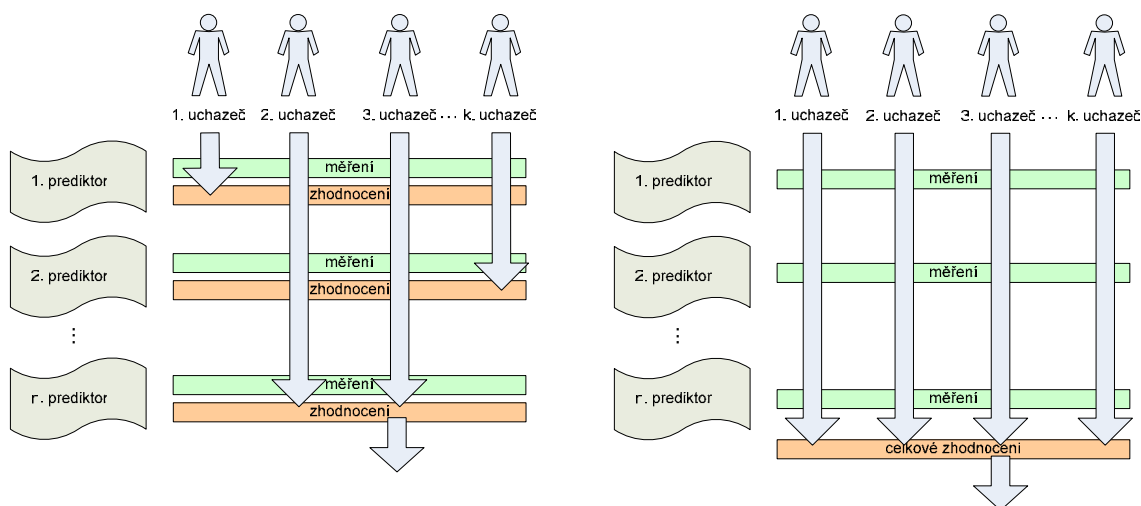
Často užívaná metoda **AHP** (Analytic Hierarchy Process) se rozhodovací problém snaží rozložit do hierarchie od obecnějších prvků ke zvláštnostem. Na podobném principu je založena i metoda **SMART**⁶ (Simple Multi-Attribute Rating Technique) [55].

⁶ On-line demonstrační rozhodovací systém pracující metodou SMART lze vyzkoušet na webové adrese <http://mis.ucd.ie/students/mms9798/g2/main.htm>.

6.6.2 Speciální modely pro hodnocení uchazečů

V disciplíně výběru uchazečů o zaměstnání panují základní dva přístupy:

- Diskrétní přístup (Discrete Approach)
- Komplexní přístup (Comprehensive Approach)



Obrázek 14 – Rozdíl mezi Diskrétním přístupem (vlevo) a Komplexním přístupem (vpravo)

Diskrétní přístup

Praktické posuzování uchazečů jde oproti strategiím popsaným v předchozí kapitole, kde pro každou variantu komplexně zjišťujeme hodnoty všech prediktorů, zjednodušit. Potřeba redukovat počet hledaných hodnot ze součinu *počet uchazečů x počet prediktorů* na nějakou nižší hodnotu vyplývá z faktu, že realizace měření některých prediktorů je velice nákladná. Představme si například, že bychom museli tisíc uchazečů nechat ohodnotit metodou Assessment centra.

Zároveň si uvědomme, že v reálném procesu výběru existuje u většiny kritérií nějaká minimální kritická hodnota, které jestliže uchazeč nedosáhne, nemůže kritérium vykompenzovat ani sebelepšími hodnotami u jiných kritériích. Škola například nemá zájem zaměstnat jako učitele Výpočetní techniky sebezdatnějšího informatika, pokud postrádá základní sociální dovednosti.

Předestřená hlediska bere v úvahu tzv. *diskrétní přístup*. V něm se hodnoty prediktorů u uchazečů zjišťují postupně. Jakmile je některý z uchazečů v probíhající kole vyhodnocen jako nepřijatelný, do dalších kol už nepostupuje. Takto některé uchazeče

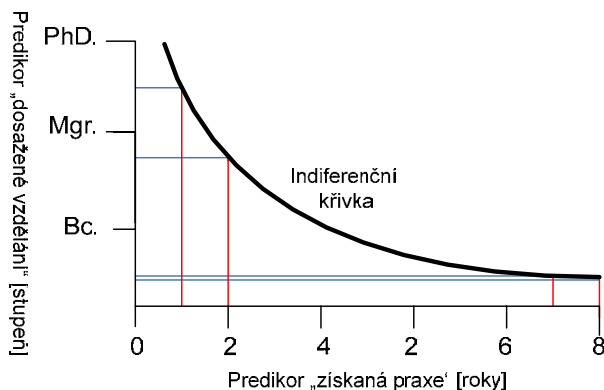
vyřadíme, aniž bychom pro ně musely zjišťovat hodnoty všech prediktorů. Tím se celý proces značně zjednoduší a zlevní. Kupříkladu nedostatečné vzdělání uchazeče lze vyzorovat již z jeho životopisu při prvotní prověře, aniž bychom jej museli zvat k podrobnému interview. Často se také stává, že během výběrového řízení opadne zájem ze strany uchazeče.

Tento přístup využívá **Metoda vícenásobné překážky** (Multiple Hurdle). Obrázek 14 vlevo zobrazuje její průběh. Všichni uchazeči jsou nejprve podrobeni proceduře za účelem změření prvního kritéria. Ihned poté jsou vyřazeni všichni, kdož nesplnili stanovenou minimální hodnotu. Takto jsou uchazeči postupně „prosévání dalším a dalším sítím“. Po posledním kole zůstane množina těch, kteří ve všech ohledech překročili minimální hranici, aniž by mezi sebou byly jakkoliv setřídění.

Komplexní přístup

Komplexní přístup na druhou stranu problém chápe tak, že slabší hodnotu jednoho prediktoru (potažmo kritéria), lze kompenzovat dobrými výsledky ostatních prediktorů. Tato metoda je dražší, neboť vyžaduje zjistit všechny hodnoty prediktorů všech uchazečů, a hodnocení probíhá až nakonec (viz Obrázek 14 vpravo). Nemůže se nám však přihodit, že odmítneme uchazeče, který splňuje místo pěti let požadované praxe pouze roky čtyři, zatímco v ostatních ohledech je vynikající [6].

O **Vícenásobné regresi** jsme se již zmínili v kap. 6.5.1 v souvislosti s predikcí pracovního výkonu. Vychází z ní také některé modely obecného vícekritériálního rozhodování (kap. 6.6.1). Při této metodě se sestaví vícenásobná regresní funkce, jejíž závislé proměnné představují prediktory a nezávislá proměnná figuruje coby předpokládaný pracovní výkon. Funkce je z důvodu jednoduchosti většinou lineární, což ale nemusí vždy vyhovovat. Obrázek 15 ukazuje, že rozdíl mezi tím, zda uchazeč absolvoval jeden nebo dva roky praxe vnímáme velice intenzivně. Roční rozdíl v dosažené praxi tak může značně vykompenzovat handicap ve vzdělání. Naopak roční interval mezi sedmým a osmým rokem praxe se zdá být téměř lhostejný. Vidíme tedy, že kritéria (a tedy i prediktory) vykazují zpravidla klesající mezní užitek. Proto je na místě uvažovat o jiné než přímkové regresi, a to například logaritmické.

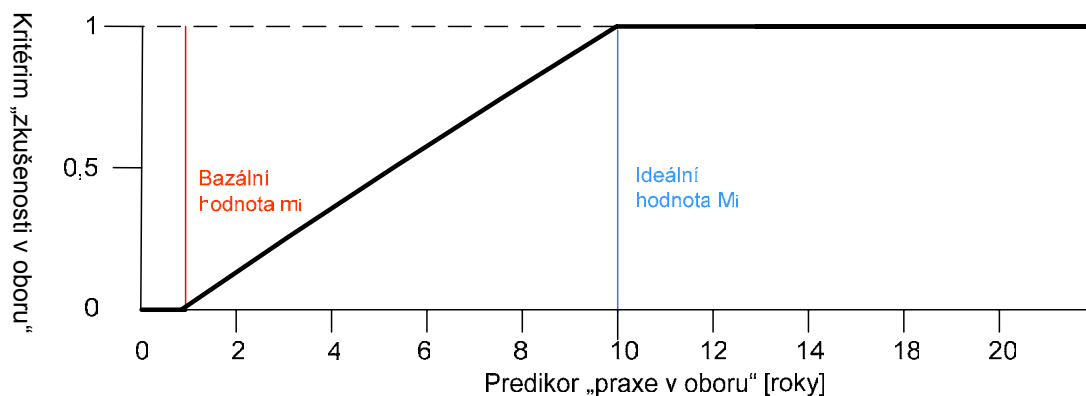


Obrázek 15 – Indiferenční křivka jako soubor kombinací hodnot dvou prediktorů, kde každá kombinace přináší stejný užitek.

Klesající mezní užitek nějakého prediktoru můžeme při značném zjednodušení charakterizovat pouhými dvěma parametry:

- **Bazální** (minimální) hodnotou m_i - jestliže uchazeč nedisponuje znakem alespoň v této míře, bude vždy odmítnut,
- **Ideální** (maximální) hodnotou M_i - nevnímáme rozdíl mezi uchazečem právě s touto mírou znaku a uchazečem vybaveným ještě lépe.

Tyto dva parametry jsme zpravidla schopni bez delšího přemýšlení specifikovat. Zamysleme se kupříkladu nad výší praxe v oboru u přijímaného uchazeče. Bazální hodnotou je například rok praxe. Nedovedeme si představit, kde by kandidát s kratší praxí získal potřebný přehled v problematice. Naopak ideální hodnotu chápeme například jako deset let. Tehdy konstatujeme, že délka praxe od desátého roku je pro nás více méně irelevantní. To vede na velice jednoduchou regresní fci, viz Obrázek 16., která není v celém svém oboru lineární. Prediktor nelineární povahy však transformuje do lineárně vyjádřeného kritéria.



Obrázek 16 – Jednoduchý popis regresní funkce pomocí dvou parametrů

Metoda **Vícenásobného vyřazování** (Multiple Cutoffs) je vlastně speciálním případem Vícenásobné regrese. U každého kandidáta se opět zjistí všechny prediktory a na závěr se provede hodnocení. Uchazeč je vyřazen v případě, kdy alespoň jeden prediktor nedosahuje stanovené minimální hodnoty (např. pro prediktor „praxe“ je stanovena prahová hodnota jeden rok) [14]. Regresní funkce tedy vrací hodnotu *jedna*, pokud hodnoty všech jejích parametrů překračují prahové hodnoty, v opačném případě vrací nulu. Nevýhoda této metody je patrná na první pohled. Použitím obdobné metody *Vícenásobné překážky* v rámci diskrétního přístupu bychom totiž se stejným výsledkem neabsolvovali zbytečná měření.

Při metodě nazvané **Podobnost k ideálnímu profilu** (Profile Matching) nejprve stanovíme takové hodnoty kritérií, které odpovídají profilu ideálního pracovníka. Takový profil lze zjistit změřením vlastností množiny elitních pracovníků organizace, kteří již na obsazované pozici pracují. Následně se hledá takový uchazeč, který se nejvíce blíží ideální představě. K tomu účelu existuje více algoritmů, např. metoda nejmenších čtverců [55]. Dále platí několik pravidel, která jsou podmínkou úspěšného nasazení metody, např. že u každého použitého kritéria by se jeho ideální hodnota měla nacházet uprostřed škály, nikoliv v jeho minimu nebo maximu.

Osobně se domnívám, že nerozvážené užití metody může být dokonce diskriminující. Ze svého okolí znám případ, kdy uchazeč opakovaně poptávající místo prodavače musel nakonec zatajit své vysokoškolské vzdělání. To je jistě nesprávné, protože univerzitní titul není při prodeji zboží na překážku. Manažeři kontaktovaných firem ovšem zřejmě cítili

rozpor mezi charakteristickými rysy tohoto uchazeče a svou představou ideálního prodavače.

Kombinovaný přístup

Komplexní a diskretní přístup lze samozřejmě sloučit. V takovém případě hovoříme o tzv. **Kombinované metodě** (Combination Method). Při ní se nejprve metodou *Vícenásobné překážky* vyřadí nepřijatelní uchazeči, aniž by tito museli absolvovat měření všech prediktorů. Následně se na zbylou množinu vyhovujících uchazečů uplatní metoda *Vícenásobné regrese*. Kombinovaná metoda tedy disponuje výhodami obou dílčích přístupů [14].

6.6.3 Využití fuzzy teorie v rozhodovacím procesu

Základní pojmy z teorie fuzzy množin

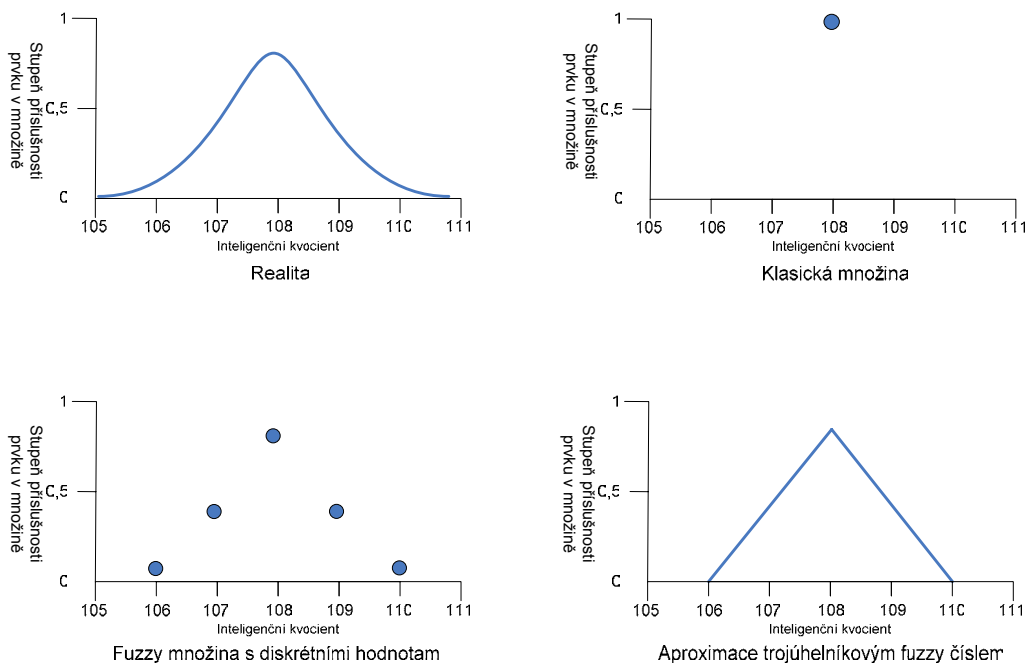
Kompletní popis teorie fuzzy množin a obecněji tzv. „soft computingu“ přesahuje rámec této práce. Tyto informace lze nalézt v literatuře [30, 37, 41]. Připomeňme proto pouze základní pojmy. Fuzzy teorii představil v roce 1965 profesor Lotfi Zadeh. Tvrdí v ní, že prvky mohou patřit do množiny narozdíl od klasického přístupu i pouze částečně. Tento pohled na věc slibuje rozumnější chování systémů (rozhodovacích, regulačních atp.), zvláště pokud jsou vstupní data do systému zadávána člověkem.

Fundamentem teorie je tzv. **fuzzy množina**. Tuto množinu zapisujeme tak, že kromě výčtu prvků patřících do množiny tak jako v klasické logice ke každému prvku určíme **stupeň příslušnosti prvku v množině** v rozsahu [0,1]. Množinu všech prvků, které mohou do nějaké množiny patřit, nazýváme **univerzem** (např. pro množinu členů Odborového svazu železničářů jsou univerzem všichni čeští železničáři).

Fuzzy množina obvykle představuje vlastnost nějakého objektu (např. IQ uchazeče) a její prvky představují výčet hodnot, jichž tato vlastnost nabývá. Jestliže uchazeči předložíme IQ test, který povede k výsledku 108, obsahovala by klasická množina právě tento prvek a nic jiného (viz Obrázek 17 vpravo nahoře). Taková jednoprvková množina existuje i jako speciální případ ve fuzzy teorii, kde ji nazýváme *fuzzy jednotka* a značíme $I_{uchazeč} = \left\{ \frac{a}{x} \right\}$. Konkrétně v tomto případě se bude jednat o fuzzy jednotku $\left\{ \frac{1}{108} \right\}$. Proměnná a nad lomící čarou udává stupeň příslušnosti (zde se jedná o jisté členství prvku v množině), proměnná x identifikátor prvku. Víme však, že v realitě se skutečná inteligence pohybuje někde kolem naměřené hodnoty ve smyslu normálního rozdělení. Podle fuzzy

teorie lze do množiny zařadit několik sousedních diskretních prvků s odpovídajícím členstvím, např.

$$I_{\text{uchazeč}} = \left\{ \frac{0,1}{106}, \frac{0,4}{107}, \frac{0,7}{108}, \frac{0,4}{109}, \frac{0,1}{110} \right\}.$$



Obrázek 17 – Shora zleva: rozdělení veličiny v realitě; zápis klasickou cestou; znázornění diskretními fuzzy prvky; aproximace spojitým trojúhelníkovým fuzzy číslem.

Obrázek 17 vpravo dole znázorňuje, že spojitě normální rozdělení můžeme také aproximovat tzv. *trojúhelníkovým fuzzy číslem*. Tento tvar přináší výhodu v jednoduchém výpočtu stupně příslušnosti kteréhokoliv prvku na spojitě škále.

Tak jako s klasickými množinami, i s fuzzy množinami můžeme provádět různé operace. Operace průniku dvou klasických množin zanechá ve výsledné množině koliko prvků, které se vyskytovaly v obou operandech. Binární operace **fuzzy průniku**, kterou značíme \wedge , ponechá ve výsledné množině prvek v nižším z obou stupňů příslušnosti. Obdobně **fuzzy sjednocení** \vee není nic jiného než aplikace operace maxima na stupně příslušnosti stejného prvku přítomného v obou množinách. Množinové operace ale mohou být implementovány i odlišně, podle potřeby. Různé typy operací, včetně těch uvedených, lze zobecnit jako tzv. *t-normy* nebo *t-konormy*, charakterizované příslušnými parametry [13].

V klasické logice je relace mezi 2..n množinami $M_1, M_2 \dots M_n$ buď pravdivá nebo nepravdivá (např. v případě binární relace „ $=$ “ se množina M_1 množině M_n buď rovná, nebo nikoliv). V případě **fuzzy relace** je však míra její pravdivosti určena jemněji libovolným reálným číslem mezi 0 a 1. Uvažujeme-li opět relaci rovnosti dvou množin, popisuje toto číslo míru jejich podobnosti (0 .. množiny jsou zcela odlišné, 1 .. jedná se o shodné množiny). Obecně je binární fuzzy relace R mezi dvěma prvky (nejen množinami) množinou v kartézském součinu univerz těchto prvků U_1 a U_2 , která vyjadřuje pravdivost relace mezi všemi kombinacemi hodnot, jichž mohou tyto dva prvky nabývat [30].

Defuzzifikací rozumíme odvození reálného čísla, které určitou fuzzy množinu, resp. nějaký její znak, vhodným způsobem popisuje. Mezi nejpoužívanější defuzzifikační metody patří například CA (Centroid Average), MCA (Maximum Center Average), MOM (Mean of Maximum), atp. [26].

Fuzzy vícekriteriální rozhodování

Teorie fuzzy množin obohatila svými přístupy již existující disciplíny vícekriteriálního rozhodování. Koncept fuzzy cílů a fuzzy rozhodování jako první nastínili Belman a Zadeh [28]. Fuzzy teorie ovšem také dala vzniknout řadě nových teorií, např. teorii interdependence [11] nebo vícerozměrné strategii rozhodování [25].

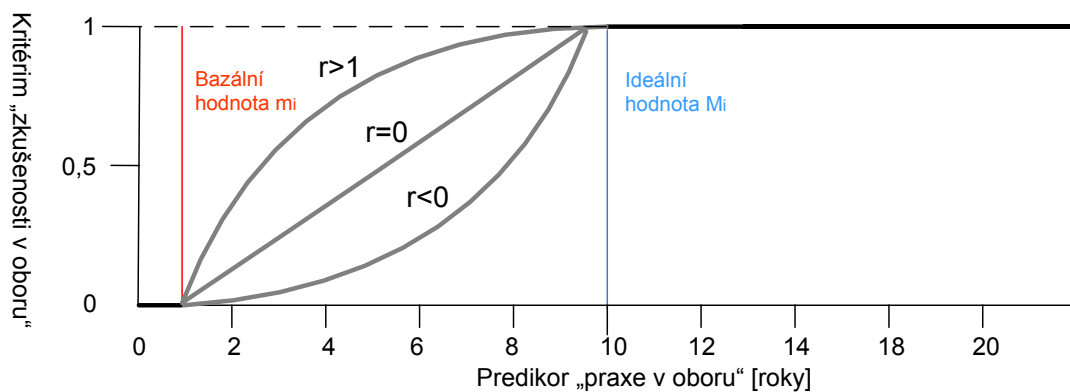
Hovoříme-li o fuzzy vícekriteriálním rozhodování, jedná se ve své podstatě pouze o jiný pohled na výpočty prováděné při vícekriteriálním rozhodování bez přívlastku „fuzzy“.

Především je pro každý prediktor definována tzv. *aplikační funkce*⁷ $h: R \rightarrow [0,1]$, která v závislosti na hodnotě prediktoru (nezávislá proměnná) specifikuje míru členství hodnoty prediktoru ve fuzzy množině „dobrých řešení“ (závislá proměnná). Funkci jsme vlastně již představili v kapitole 6.6.2, viz Obrázek 16. Funkce na obrázku se konkrétně nazývá *lineární aplikační funkcí*, a to podle intervalu mezi bazální a ideální variantou. Klesající mezní užitek ale přesněji popisuje *konkávní aplikační funkce*, definovaná vztahem:

⁷ Termín *aplikační funkce* se týká speciálně závislosti *prediktor* \rightarrow *míra splnění požadavku*. V teorii fuzzy množin se funkce $R \rightarrow [0,1]$ popisující stupeň příslušnosti prvku obecně nazývá *funkce příslušnosti*.

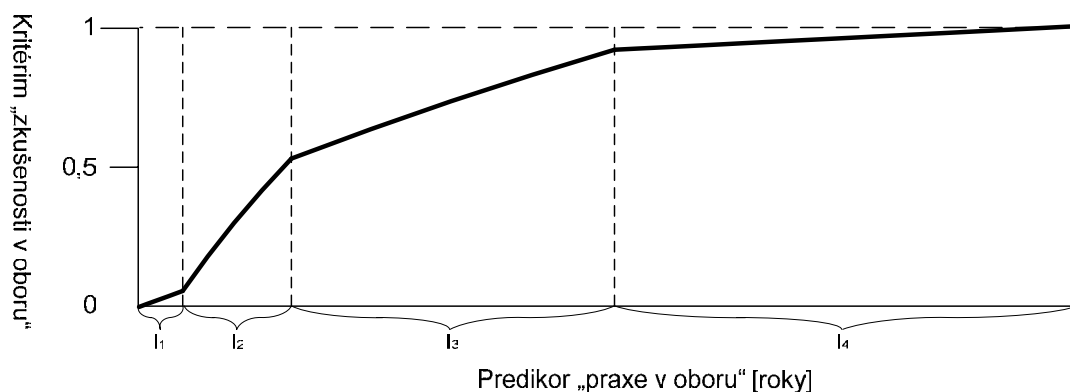
$$h(x) = \begin{cases} 0 & \text{pokud } x \leq m_i \\ \left(1 - \frac{M_i - x}{M_i - m_i}\right)^{\begin{cases} 1 & \text{pokud } r=0 \\ 1/r+1 & \text{pokud } r>0 \\ |r|+1 & \text{pokud } r<0 \end{cases}} & \text{pokud } m_i < x < M_i \\ 1 & \text{pokud } x \geq M_i \end{cases} \quad (15)$$

kde $r > 0$. Pro $r < 0$ bychom dostali fci konvexní, pro $r = 0$ fci lineární [11].



Obrázek 18 – Konkávní , lineární a konvexní aplikační funkce.

Aplikační fci můžeme také definovat množinou lineárních fci, z nichž každá je platná pro určitý interval I_i (viz Obrázek 19).



Obrázek 19 – Aproximace aplikační fce množinou lineárních fci.

Jakmile máme pro každou j -tou variantu ($j=1..k$, kde k je počet uchazečů) zjištěnu příslušnou hodnotu aplikační funkce i -tého prediktoru ($i=1..n$, kde n je počet prediktorů), můžeme sestavit kritériální matici $C_{ij}=(h_i(x_{ij}))$, která je vlastně obdobou normalizované kritériální matice, získané např. metodou WSA při standardním kritériálním rozhodování (viz kap. 6.6.1, Obrázek 13).

Druhý krok spočívá v uplatnění vah w_i na jednotlivé prvky matice C_{ij} . Předpokládáme, že váhový vektor W nám určila některá z metod odhadu vah prediktorů (kap. 6.6.1). Ve většině praktických aplikací (např. [4, 28]) se uplatňuje Yagerův přístup, v němž váženou kritériální matici získáme jednoduchým vztahem

$$C_{ij}^w = h_i(x_{ij})^{w_i}, \quad (16)$$

kde h_i je aplikační funkce i -tého prediktoru, w_i je váha i -tého prediktoru a x_{ij} je hodnota i -tého prediktoru u j -tého uchazeče.

Posledním úkol představuje stanovení celkového skóre jednotlivých uchazečů (tzv. agregace), případně jejich seřazení. K agregaci jednotlivých faktorů varianty se nejčastěji používá konjunkce

$$D_j = C_{1j}^w \wedge C_{2j}^w \wedge \dots \wedge C_{nj}^w, \quad (17)$$

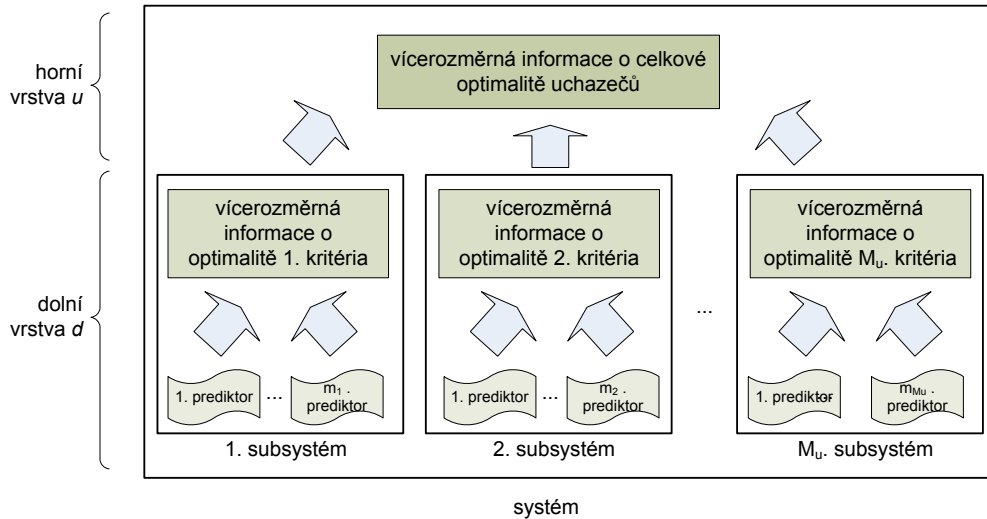
která ovšem každou j -tou variantu hodnotí pouze podle jejího nejhůře zastoupeného faktoru. Požadujeme-li, aby faktory (tj. prediktory) byly do jisté míry kompenzovatelné, lze místo konjunkce (potažmo obecné t-normy) použít operátor **OWA** (operátor seřazeného váženého průměru) [13, 28]. Pak již nezbyvá nic jiného než varianty $D_j \in R$ setřídít. Pokud jsme ale zvolili jiný postup než výše uvedený, existuje samozřejmě množství strategií jak řadit přímo nedefuzziifikované fuzzy množiny (viz [44]).

Vícerozměrná vícevrstvá fuzzy strategie rozhodování

Popišme nyní metodu publikovanou v [25] Jiangem a Chenem, kterou s drobnými modifikacemi použijeme v návrhu systému (kap. 7). Jedním z rysů metody je, že se systém, jehož se rozhodování týká, rozloží na několik vrstev. V našem případě je systémem myšlen vztah mezi prediktory a výkonem uchazečů. Tento systém rozložíme na:

- Horní vrstvu u , představující funkční vztah (7) mezi množinou hodnot kritérií $K=\{k_1, k_2, \dots, k_{M_u}\}$ a předpokládaným výkonem V budoucího zaměstnance.

- Dolní vrstvu d , obsahující skupinu M_u subsystémů. Každý subsystém představuje funkční vztah (6) mezi množinou hodnot prediktorů $P = \{p_1, p_2, \dots, p_{m_k}\}$ a kritériem k_k . Proměnná m_k představuje počet prediktorů vztahujících se ke k -tému kritériu.



Obrázek 20 – Struktura vícezměrné vícevrstvé fuzzy rozhodovací strategie.

V první fázi se zaměříme na dolní vrstvu a postupně zhodnotíme každý subsystém, tzn. že vypočteme míru splnění každého kritéria. Ve druhé fázi potom v horní vrstvě zhodnotíme celkový výkon uchazečů na základě míry splnění každého kritéria.

1. fáze – opakuje se pro 1. až M_u -té kritérium ($k=1.. M_u$):

- Sestavíme matici $C^k = (c_{ij}^k)$, která bude obsahovat hodnoty i -tého prediktoru v rámci k -tého kritéria pro j -tého uchazeče ($i=1..m_k$; $j=1..n$ kde n je počet uchazečů).
- Matici C^k transformujeme do matice $F_{ij}^k = (f_{ij}^k)$ v normalizovaném tvaru, a to pomocí vztahu $f_{ij}^k = h_i^k(c_{ij}^k)$, kde h_i^k je aplikační fce i -tého prediktoru k -tého kritéria. Abychom reflektovali klesající mezní užitek prediktoru, použijeme narozdíl od [25], kde je uvažována lineární aplikační fce, aplikační fci (15) nebo aproximaci aplikační fce množinou lineárních fci (viz Obrázek 19).

- c. Definujeme $L = \{l_1, l_2, \dots, l_l\}$ jakožto množinu l vzorů rozhodnutí, kde l_1 je optimální rozhodnutí, l_l naopak rozhodnutí nejhorší.
- d. Sestavíme fuzzy matici standardních cílů $S^k = (s_{ip}^k)$, kde $i=1, \dots, m_k$; $p=1, \dots, l$. Prvky matice udávají, jakou hodnotu musí mít prvek f_{ij}^k v matici F , abychom rozhodnutí vybrat j -tého uchazeče podle i -tého prediktoru hodnotili jako l -té. První sloupec matice tedy obsahuje vektor ideálního rozhodnutí l_1 , poslední sloupec vektor bazálního rozhodnutí l_l .

$$s_{ip}^k = \begin{cases} s_{ip}^k = 1 & \text{pokud } p = 1 \\ s_{ip}^k = 0 & \text{pokud } p = l \\ s_{ip}^k = s_{i1}^k + \frac{s_{il}^k - s_{i1}^k}{l-1} (l-p) & \text{pokud } 1 < p < l \end{cases}, \quad (18)$$

- e. Pomocí Saatyho metody (kap. 6.6.1) nebo alternativně ručním zadáním určíme váhový vektor prediktorů k -tého kritéria $W^k = (w_1^k, w_2^k, \dots, w_{m_k}^k)^T$, kde $\sum_{i=1}^{m_k} w_i^k = 1$. Navíc předpokládáme kombinovaný přístup k výběru uchazečů (kap. 6.6.2), a proto se hodnocení k -kritéria nebude vždy účastnit všech m_k prediktorů. Například kritérium *zkušenosti v oboru* bude nejprve posuzováno pouze podle „levného“ prediktoru *uvedená délka praxe*, a až po určitém časovém odstupu na menší skupině uchazečů podle dvojice prediktorů *délka praxe*, *test zkušeností*. Je zřejmé, že zatím nepoužité prediktory musí mít dočasně nastavenou nulovou váhu.
- f. Vypočteme vícerozměrnou fuzzy matici $U^k = (u_{pj}^k)$, kde $p=1..l$; $j=1..n$. Matice udává míru členství j -tého uchazeče v p -tém rozhodnutí,

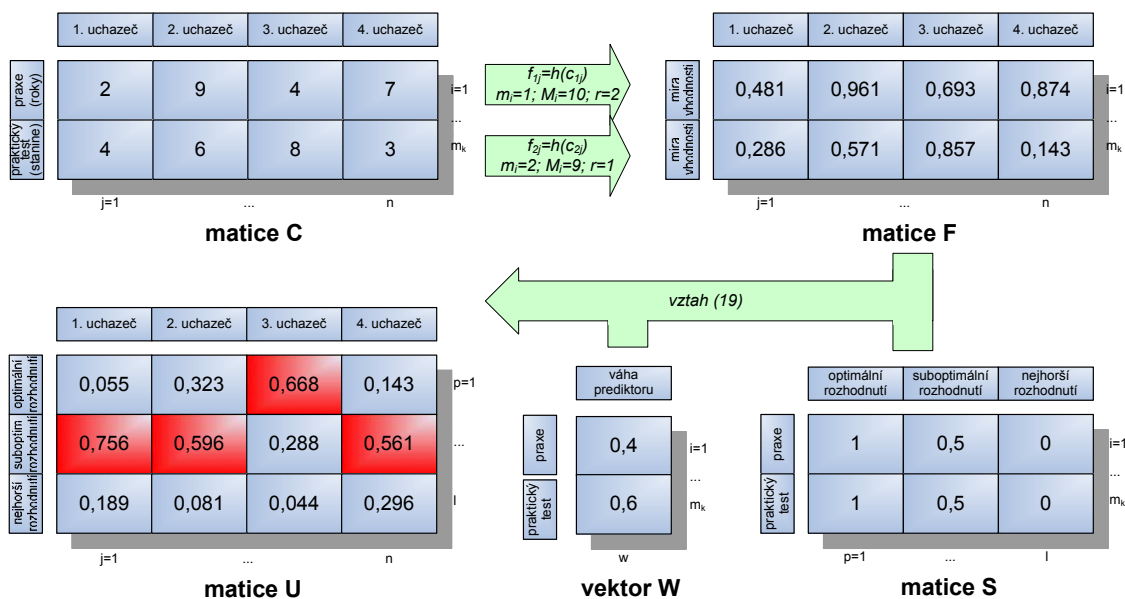
⁸ Původní strategie uvažuje odvození ideální a bazální hodnoty z předkládaných variant. To ale nemůžeme akceptovat – co kdyby se k výběru náhodou dostavili například sami nepřijatelní uchazeči.

agregovaném pro všechny prediktory. Pro j -tého uchazeče ideálně splňujícího k -té kritérium by například měl příslušný sloupec matice obsahovat hodnoty $(1, 0, \dots, 0)^T$. Matici vypočteme podle vztahu⁹:

$$u_{pj}^k = \frac{1}{\sum_{c=1}^l \left[\frac{\sum_{i=1}^{m_k} (w_i^k |f_{ij}^k - s_{ip}^k|)^q}{\sum_{i=1}^{m_k} (w_i^k |f_{ij}^k - s_{ic}^k|)^q} \right]^{2/q}}, \quad (19)$$

kde $i=1..m_k; j=1..n; p,c=1,..l; q$ volíme 2.

g. V souladu s kombinovaným přístupem a metodou vícenásobné překážky nyní vyřadíme ty uchazeče, kteří spadají do bazálního rozhodnutí (tj. u_{ij}^k je v jejich případě vyšší než stanovená kritická hodnota z^k).



Obrázek 21 – Příklad výpočtu v 1. fázi strategie.

⁹ Pokud se ve vztahu (19) v hranaté závorce vyskytne výraz $\frac{0}{0}$, nahradíme jej hodnotou 1.

2. fáze – probíhá analogicky s první fází, pouze místo prediktorů nyní posuzujeme syntetizované hodnoty kritérií:

a. Obdobně jako v kroku 1.e zjistíme váhový vektor kritérií

$$W^u = (w_1^u, w_2^u, \dots, w_{M_u}^u)^T.$$

b. Vícerozměrnou fuzzy matici $U^u = (u_{pj}^u)$ získáme podle vztahu

$$u_{pj}^u = \frac{1}{1 + \left\{ \frac{\sum_{k=1}^{M_u} [w_k^u (1 - u_{pj}^k)]^q}{\sum_{k=1}^{M_u} (w_k^u \cdot u_{pj}^k)^q} \right\}^{2/q}}, \quad (20)$$

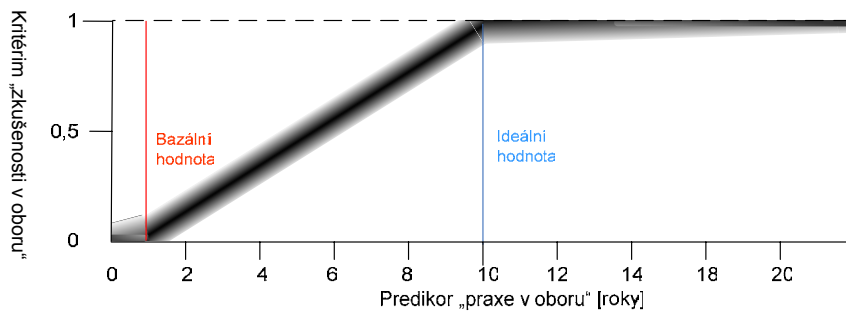
c. Defuzzifikací sloupců matice U^u metodou CA zjistíme kvalitu rozhodnutí při výběru j -tého uchazeče d_j . Podle této hodnoty uchazeče seřadíme:

$$d_j = \sum_{p=1}^l p \cdot u_{pj}^u. \quad (21)$$

Fuzzy rozhodování a přibližná dedukce

Výše popsaná strategie specifikuje kvalitu potenciálních rozhodnutí pomocí výstupní fuzzy množiny, avšak na základě deterministických vstupních údajů. Ukažme si nyní, že možné pracovat také s nepřesnými vstupními údaji a jejich variabilitou.

Skutečnou hodnotu kritéria pouze tušíme někde v blízkosti přesně vypočtené hodnoty. Pravděpodobnost výskytu skutečné hodnoty bude vykazovat znaky normálního rozdělení se střední hodnotou ve vypočtené hodnotě. Rozptyl bude ovlivněn spolehlivostí testu, viz. kap. 6.5.1, vztah (4). Intenzita šedi křivky grafu na obrázku níže vlastně udává, jaká je pravděpodobnost, že uchazeč s praxí x roků má úroveň zkušeností v oboru y .

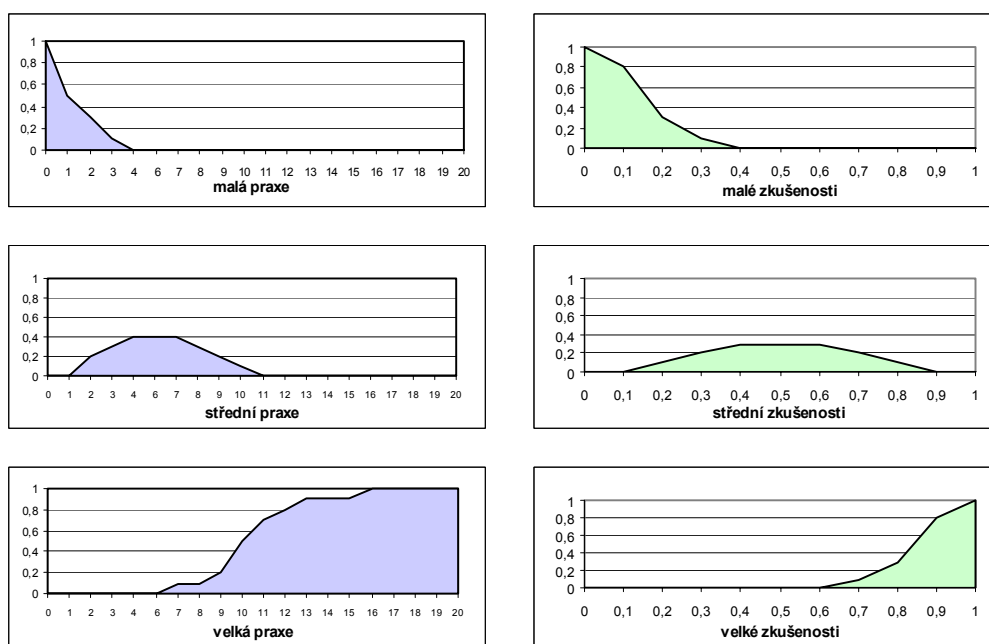


Obrázek 22 – Regresní funkce při uvažování nejistoty.

Doposud (viz Obrázek 16) jsme volili přístup, při němž byl vztah *praxe* → *zkoušenosti* popsán relací rovnosti $R_{pz}(\langle p, z \rangle)$ mezi oběma veličinami, nabývající hodnot *nepravda* nebo *pravda*. Obrázek 22 ale obsahuje fuzzy relaci $R_{pz}(\langle p, z \rangle)$ nabývající hodnot v rozsahu $[0, 1]$. Tu lze alternativně interpretovat jako relaci rovnosti $R_{pzs}(\langle p, z, s \rangle)$, kde s je stupeň příslušnosti veličiny p k veličině z .

Tuto funkční závislost doplněnou o třetí rozměr je již poměrně složité popsat konvenčním matematickým vztahem. Tzv. **metoda fuzzy aproximace** [41] slouží právě k robustnímu popisu takovýchto nejistých, jakoby „rozmazaných“ funkčních závislostí.

Definujme univerzum $U_{praxe} = \{0; 1; \dots; 20\}$ celých roků, nabyté zkušenosti budeme kvantifikovat koeficientem z v universu $U_{zkušenosti} = \{0; 0,1; 0,2; \dots; 0,9; 1\}$. V těchto univerzech definujme fuzzy množiny *malá praxe (MP)*, *střední praxe (SP)*, *velká praxe (VP)* a *malé zkušenosti (MZ)*, *střední zkušenosti (SZ)*, *velké zkušenosti (VZ)*, jejichž funkce příslušnosti jsou vyjádřeny v grafech níže.

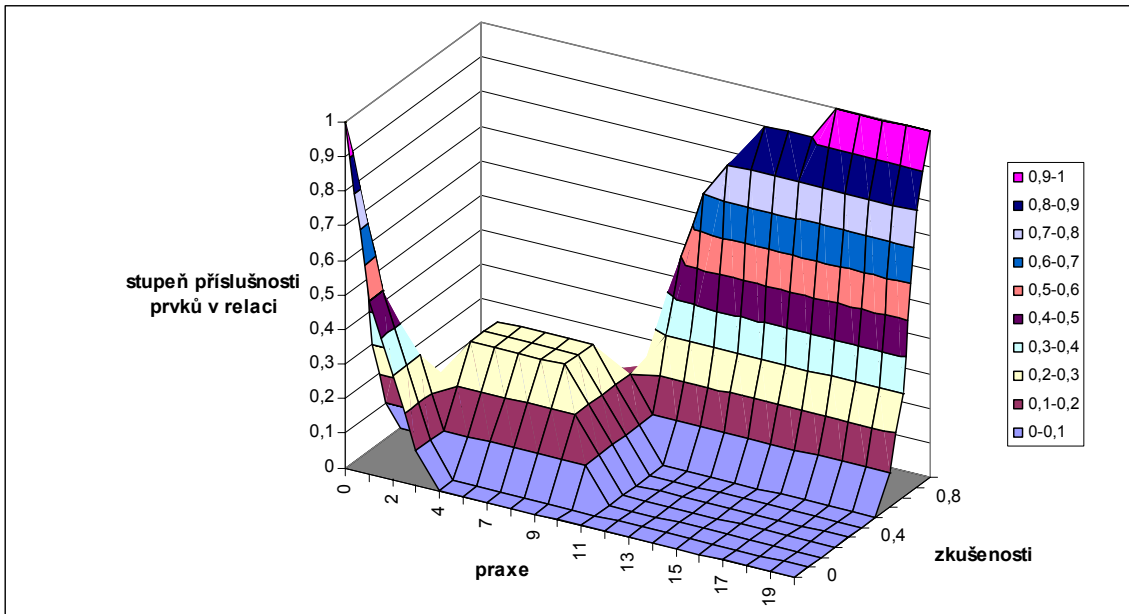


Obrázek 23 – Jazykové výrazy určené pro sestavení fuzzy relace.

Nyní funkční závislost $praxe \rightarrow zkušenosti$ aproximujeme výrokem složeným ze tří pravidel:

$$(MP \wedge MZ) \vee (SP \wedge SZ) \vee (VP \wedge VZ). \quad (22)$$

Odtud lze sestavit relaci odpovídající trojrozměrnému grafu níže. Pokud bychom použili dostatečný počet pravidel (např. deset), složený výrok by již uspokojivě approximoval modelovanou funkční závislost. Místo jazykových výrazů lze samozřejmě používat také fuzzy čísla.



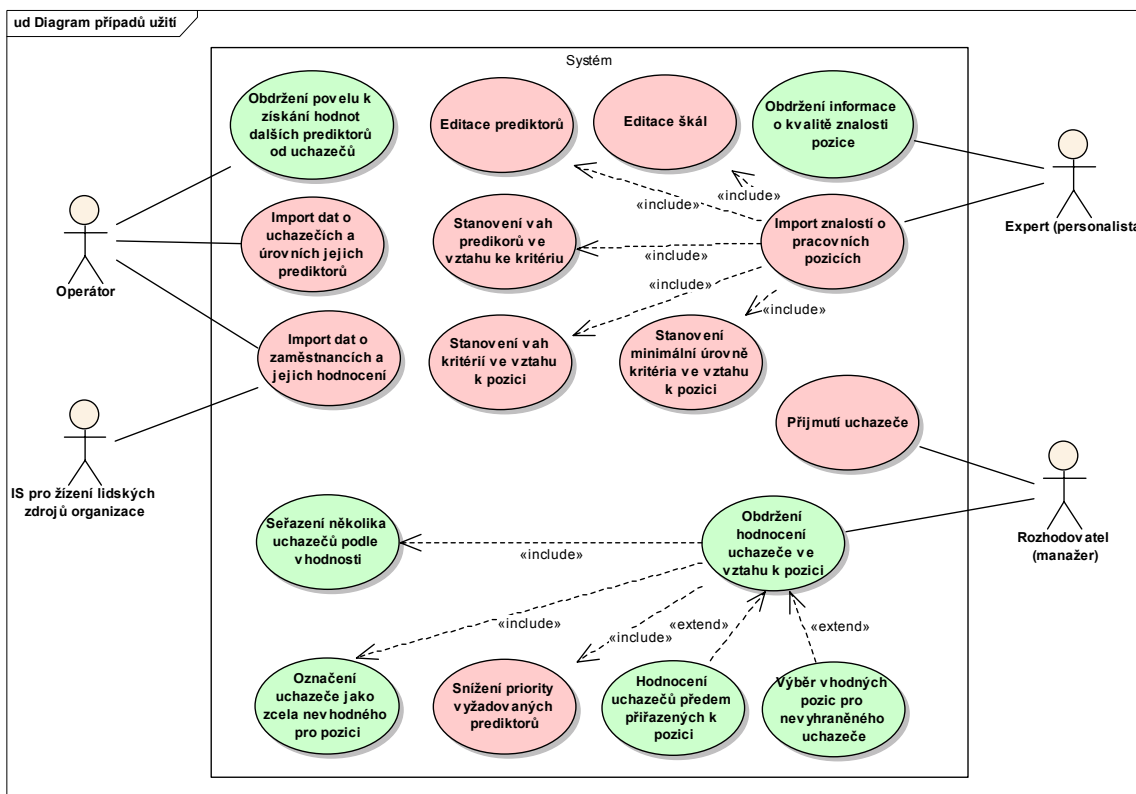
Obrázek 24 – Fuzzy relace mezi praxí a zkušenostmi.

Složený výrok lze pochopitelně rozšířit o další proměnné, abychom mohli vymodelovat například funkční závislost (*praxe, praktický test*) \rightarrow *zkušenosti*. Funkci používáme jednoduše tak, že vypočítáme fuzzy obraz nezávislé hodnoty v sestrojené fuzzy relaci. Jako nezávislou hodnotu funkce můžeme specifikovat nejen přesné číslo (například praxe přesně osm let, tedy $p = \{1/8\}$), ale také fuzzy množinu. Například když zatím neznáme výsledné testové skóre typu *stanine* nějakého testu, použijeme fuzzy množinu

$$t = \left\{ \frac{1}{9}/1, \frac{1}{9}/2, \dots, \frac{1}{9}/8, \frac{1}{9}/9 \right\}.$$

7. Návrh systému

7.1 Případy užití



Obrázek 25 – Diagram případů užití systému pro hodnocení uchazečů o zaměstnání.

Obrázek 25 ukazuje zúčastněné subjekty a případy užití aplikace s nimi spjaté. Červeně jsou zvýrazněny z hlediska zúčastněných subjektů aktivní, zeleně pasivní případy. Rozlišení subjektů pochopitelně nemůžeme chápat striktně. Manažer může například zastupovat i roli personalisty a operátora. V následujících podkapitolách jsou podrobněji vysvětleny jednotlivé případy užití.

7.1.1 Import dat o uchazečích a úrovních jejich prediktorů

Obsluha systému bude vkládat data o uchazečích a již získaných hodnotách prediktorů. Hodnoty prediktorů (viz také kap. 6.5.2) budou zpravidla v jedné z následujících škál:

- ve formátu standardního skóre, např. *stanine*, *z-score* apod.,

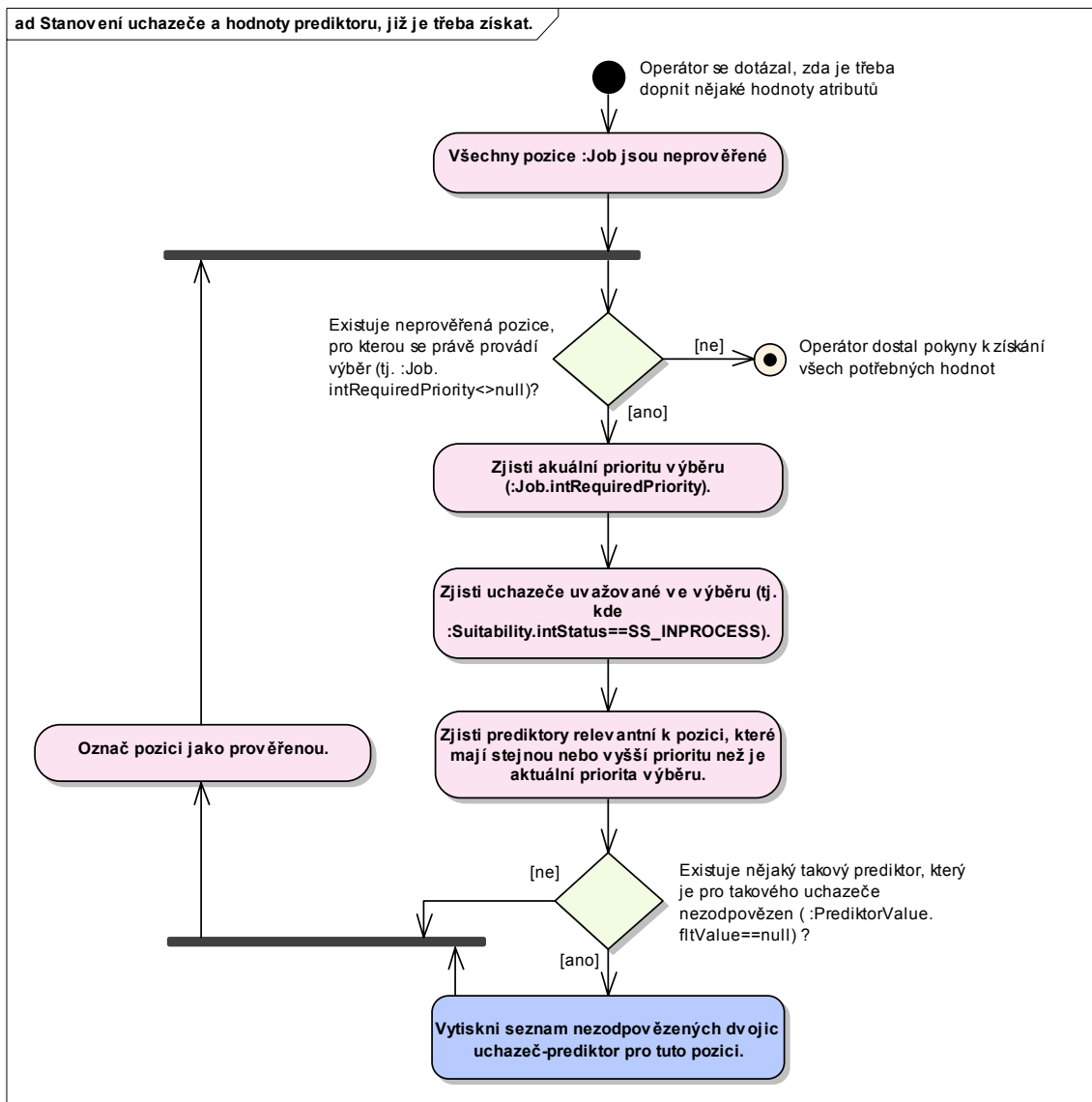
- v nominální podobě ohraničené výčtem přípustných hodnot, např. $\{SOU, SŠ, VŠ\}$,
- jiného typu, např. hodnota udávající počet celých let dosažené praxe.

V počátku výběru kandidátů pro obsazovanou pozici budou zadávána pouze životopisná data, ale pokud uchazeč postoupí do užších kol, pak také testová skóre atp.

7.1.2 Obdržení povelu k získání hodnot dalších prediktorů od uchazečů

Každý prediktor bude disponovat určitou prioritou. Prediktory s nejsnáze zjistitelnými hodnotami (např. životopisná data, předběžný pohovor) budou mít nejvyšší prioritu. Naopak nejobtížněji určitelným prediktorům (např. podrobným pohovorům, výsledkům nákladných testů) bude přiřazena nízká priorita. Zdůrazněme, že priorita se týká pouze času, kdy má být prediktor měřen (ihned/ později) a nemá nic do činění s váhou (tj. relativní důležitostí) prediktoru.

Hodnotitel v případě užití 7.1.4 typicky zhodnotí skupinu uchazečů vzhledem k obsazované pozici při znalosti omezené množiny prediktorů s nejvyšší prioritou. Posléze na základě výsledků hodnocení vyřadí zcela nepřijatelné kandidáty z výběru. Po dokončení tohoto případu užití operátor obdrží povel, aby pro redukovanou skupinu kandidátů zadal hodnoty dalších, doplňujících prediktorů s nižší prioritou (viz Obrázek 26).



Obrázek 26 – Diagram činností při zjišťování aktuálně nezodpovězených hodnot prediktorů.

7.1.3 Import znalostí o pracovních pozicích

Předpokládejme, že po dokončení činnosti uvedené v kap. 7.1.1 systém disponuje informacemi týkajícími se uchazečů. Pro určení vhodnosti kandidáta vzhledem k určité pozici (kap. 7.1.4) ale ještě potřebuje znát charakteristiku této pozice. Definice pozice sestává z následujících úkonů.

Editace škál

Škálou (třída *Jobs::Scale*) rozumíme entitu, která operátorovi usnadňuje vkládání hodnot prediktorů. Editor by samozřejmě mohl vkládat přímo číselnou hodnotu prediktoru (např. pro prediktor pohlaví hodnotu 1 v případě muže a hodnotu 2 v případě ženy), ale bylo by to nepohodlné. Škála proto operátorovi vymezuje minimální a maximální hodnotu, spočetnou či nespočetnou množinu přípustných hodnot (množinu Z , R apod.), prozrazuje jednotku (*roky*, *body* apod.). V případě nominální proměnné navíc definuje slovní názvy hodnot. Systém by měl obsahovat nepoužívanější škály, jako jsou *stanine-score*, *z-score*, *stupeň dosaženého vzdělání*, předdefinované.

Editace prediktorů

Případ užití zahrnuje vkládání, modifikaci a mazání prediktorů. Kromě názvu pro každý prediktor specifikujeme jeho prioritu (viz kap. 7.1.2), a sice celým číslem v rozsahu 1 až 9. Dále prediktoru přiřadíme jednu z předdefinovaných škál a stanovíme tvar aplikační funkce prediktoru, viz kap. 6.6.3. Aplikační funkce může být definována alternativně dvěma způsoby:

- parametry m_i , M_i , r dosazovanými do vztahu (15), viz Obrázek 18,
- lineární interpolací hodnot v intervalech, viz Obrázek 19. Ta bude specifikována množinou vektorů $I_i=(m_i, Y_i)$, kde $i = 0 \dots \text{počet intervalů}$; m_i je počátek $(i+1)$ -tého, resp. konec i -tého intervalu na x -ové ose; Y_i je hodnota aplikační funkce na počátku $(i+1)$ -tého, resp. konci i -tého intervalu na y -ové ose.

Bylo by přínosné, kdyby uživatel mohl kontrolovat grafickou podobu jím sestavené aplikační funkce.

Stanovení vah prediktorů ve vztahu ke kritériu

Expert bude moci stanovit váhy dvěma způsoby:

- Ručně určit váhu prediktoru w_i , $w_i \in [0, 1]$, $\sum_{i=1}^n w_i = 1$. Vhodné je pomocné grafické znázornění v koláčovém grafu.
- Využitím Saatyho metody, viz kap. 6.6.1 nebo [4]. Aby mohl expert v budoucnu model jednoduše korigovat, je třeba uchovávat nejen výslednou

vypočtenou váhu w_i , ale také zadané stupně preference mezi dvojicemi prediktorů (viz Tabulka 4).

Stanovení vah kritérií ve vztahu k pozici

Bude prováděno analogicky jako stanovení vah prediktorů v předcházející kapitole.

Stanovení minimální úrovně kritéria ve vztahu k pozici

Pro každou dvojici entit *kritérium, pozice* je třeba stanovit konstantu $z^k, z^k \in [0, 1]$. Tato konstanta udává maximální stupeň příslušnosti hodnoty kritéria ve vzoru nejhoršího rozhodnutí l_i , aby byl pracovní výkon na dané pozici přesto uspokojivý.

Všimněme si, že operujeme s kritériem a nikoliv s prediktorem. Například místo minimálních třech let praxe požadujeme minimální úroveň praktických zkušeností, specifikovaných jednak délkou praxe, jednak praktickým testem. Tím minimalizujeme možnost vzniku falešně negativní chyby během rozhodovacího procesu.

Konstanta z^k bude použita v kroku 1.g Vícerozměrné vícevrstvé fuzzy strategie, viz případ užití 7.1.4.

7.1.4 Obdržení hodnocení uchazeče ve vztahu k pozici

Tento případ užití lze uplatnit v jednom z následujících dvou kontextů:

- Hodnocení uchazečů hlásících se předem na určitou pozici.
- Výběr vhodných pozic pro dosud nevyhraněného uchazeče, přihlásivšího se spontánně na libovolnou pozici v organizaci.

Pro rozhodování použijeme Vícerozměrnou vícevrstvou fuzzy strategii podle [25] s drobnými modifikacemi, viz kap. 6.6.3.

Rozhodovací proces bude dvouvrstvý. Dolní vrstva hodnotí vztah *prediktor(y) → kritérium*, horní vrstva vztah *kritéria → výkon*.

Označení uchazeče jako zcela nevhodného pro pozici

Provede se v kroku 1.g strategie, pokud uchazeč vykazuje příliš nízkou hodnotu alespoň jednoho k -tého kritéria, kde $k = 1..počet\ kritérií\ ve\ vztahu\ k\ pozici$. Hodnota kritéria je nedostatečná pokud rozhodnutí přijmout uchazeče je podle matice U^k hodnoceno jako nejhorší (l -té) větším stupněm, než udává stupeň příslušnosti z^k .

Seřazení několika uchazečů podle vhodnosti

Predikovaný výkon vhodných uchazečů defuzzifikujeme a setřídíme v souladu s krokem 2.c.

Snížení priority vyžadovaných prediktorů

Když rozhodovatel vyřadí zcela nevhodné uchazeče, sníží prioritu požadovaných prediktorů (atribut *:Job.intRequiredPriority*), což by mělo vyvolat případ užití 7.1.2. Potom operátor do systému vloží hodnoty dalších, doplňujících prediktorů. Vše se opakuje od předběžného pohovoru až po závěrečné prověření informací, dokud do systému nejsou vloženy hodnoty všech prediktorů nezávisle na prioritě.

7.1.5 Přijmutí uchazeče

Uchazeč, potažmo objekt *:Applicant*, je manažerem pomocí metody *employ()* přeřazen do třídy *Employee*.

7.1.6 Import dat o zaměstnancích a jejich hodnocení

V kap. 4 figuruje požadavek na vznik zpětné vazby pro zjišťování kvality hodnocení uchazečů. Za tím účelem je nutné importovat do systému aktuální data o již přijatých uchazečích, tzn. zaměstnancích. Tato data může vkládat ručně operátor, nebo mohou být importovány pravidelně dávkově automaticky z již existujícího IS pro řízení lidských zdrojů organizace (HRMS) komponentou *External* ve formátu XML.

7.1.7 Obdržení informace o kvalitě znalosti pozice

Tento případ užití navazuje na případ užití 7.1.6, kdy jsou nejprve naimportována data o současných zaměstnancích. Tato data sestávají z:

- identifikátoru zaměstnance,
- koeficientu hodnotícího jeho skutečný pracovní výkon,
- případně hodnot prediktorů pro vypočtení předpokládaného pracovního výkonu, pokud se zaměstnanec nepodrobil měření prediktorů již v době jeho výběru.

Možné jsou v zásadě dva scénáře:

1. Zaměstnanec byl přijat na základě výběru naším systémem. Od té doby se v systému archivují naměřené hodnoty prediktorů zjištěné před přijetím. Po určité době (zpravidla jednom roce) se z HRMS do našeho systému

importuje koeficient hodnotící jeho celkový pracovní výkon. Ten lze porovnat s pracovním výkonem predikovaným během výběru.

2. Zaměstnanec byl přijat bez účasti našeho systému. Nicméně jestliže pro organizaci pracuje delší dobu, lze z HRMS opět naimportovat hodnocení jeho výkonnosti. Pak lze provést komparaci tohoto ukazatele s dodatečně vypočteným predikovaným výkonem na základě prediktorů změřených v současnosti.

Porovnání je pochopitelně nutné provádět na větším souboru zaměstnanců, výsledek lze kvantifikovat indexem korelace [21], vhodné by bylo společné grafické znázornění teoretických a empirických výsledků. Jestliže se výsledek jeví jako neuspokojivý, expert může provést korekci modelu (kap. 7.1.3) a pokusit se kvalitu znalostí vyhodnotit znovu.

7.2 Perzistence objektů

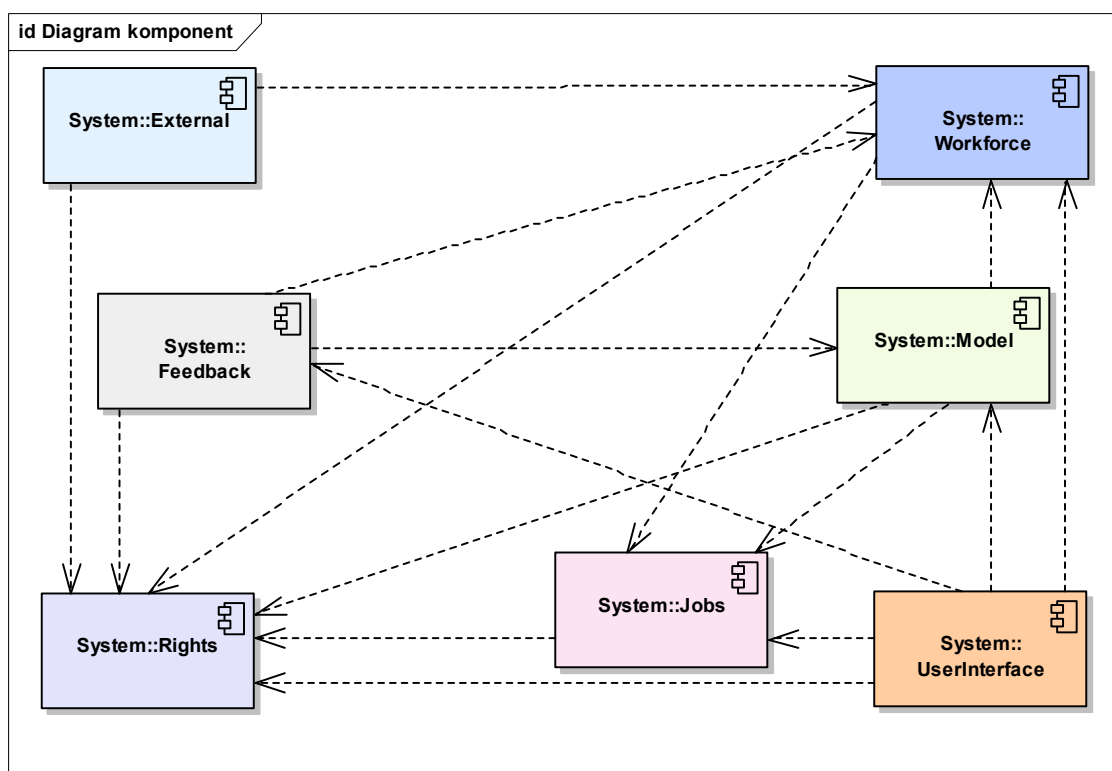
Pro ukládání dat se v zásadě nabízí zapojení jedné z následujících technologií:

- Jazyk SQL spolu s relační databází a systémem pro řízení báze dat. Databázové systémy jsou robustní, data lze snadno indexovat, ale pro zdárný běh systému je nutná přítomnost na platformě závislého databázového serveru.
- Formát XML spolu s XML parserem – Budeme-li například systém implementovat v jazyce Java, lze využít parseru z balíčku *javax.xml.parsers*. Parser je k dispozici ihned po instalaci běhového prostředí Javy a působí robustním dojmem. Později lze použít škálu alternativních parserů [33]. Nicméně formát XML a zejména přístup pomocí architektury DOM není vhodný práci s velkými objemy dat [55].

Doplňme, že existují také doposud méně rozšířené objektově orientované databázové systémy, které eliminují některé nevýhody relační databáze [31] a také že v např. MS SQL server od společnosti Microsoft integruje počínaje verzí 2000 hierarchický formát XML přímo jako zvláštní typ databázového sloupce [55].

V našem systému zajistíme perzistenci objektů ukládáním dat do databáze (např. pomocí serveru MS SQL, Oracle, MySql atp.). Formát XML naopak využijeme pro komunikaci s externími zdroji dat, viz kap. 7.3, komponenta *External*.

7.3 Diagram komponent



Obrázek 27 – Diagram komponent systému pro hodnocení uchazečů o zaměstnání.

Komponenta Jobs

Komponenta sdružuje informace o jednotlivých pracovních pozicích, zejména definici prediktorů a kritérií sloužících k předpovědi pracovního výkonu uchazeče dosazeného na příslušnou pozici. Z pohledu obecného systému pro podporu rozhodování (DSS) spravuje komponenta znalosti systému.

Komponenta zachycuje nižší i vyšší vrstvu znalostí pozice při maximalizaci opakované využitelnosti kódu (tzv. *re-use*). Obsahuje třídu *Factor* a rozhraní *Consequence*, které jsou společné pro nižší i vyšší vrstvu. Pro použití v jednotlivých vrstvách je třída příslušně specializována a rozhraní implementováno. Pro nižší vrstvu je faktorem prediktor a důsledkem kritérium. Pro vyšší vrstvu je faktorem kritérium a důsledkem pracovní pozice (přesněji předpokládaný výkon pracovníka v této pozici).

Komponenta Workforce

Komponenta *Workforce* obsahuje třídy představující uchazeče i současné zaměstnance. Ve filozofii obecného DSS představuje datový subsystém.

Komponenta Model

Komponenta implementuje strategii používanou při hodnocení uchazeče ve vztahu k pozici, viz kap. 6.6.3.

Komponenta UserInterface

Tato komponenta má za cíl zpřehlednit, sjednotit a zjednodušit tvorbu grafického rozhraní, například použitím architektury MVC. Komponenta by měla implementovat případy užití popsané v kapitole 7.1.

Komponenta External

Tak jako komponenta *UserInterface* zajišťuje komunikaci s uživatelem (operátorem, expertem nebo rozhodovatelem), komponenta *External* by měla obstarávat komunikaci mezi našim systémem a externími zdroji, typicky s IS pro řízení lidských zdrojů organizace. Implementuje tedy případ užití 7.1.5.

Komunikace bude probíhat ve formátu XML. Přenášený soubor může vypadat například následovně.

```
<?xml version="1.0" encoding="UTF-8"?>
<workforce>
  <!-- Datum exportu XML souboru z HRMS -->
  <created>2005-08-10</created>
  <!-- Zaměstnanec, jehož hodnoty prediktorů jsou systému známy z doby jeho výběru -->
  <!-- Atribut "externalID" je externí identifikátor zaměstnance (typicky osobní číslo zaměstnance v organizaci). Viz
  atribut :Employee.strExternalID. -->
  <employee externalID="8882">
    <!-- Atribut "hiredAs@externalID" je externí identifikátor pracovní pozice, kterou zaměstnanec zastává. Viz atributy
    :Employee.objHiredAs a :Job.strExternalID. -->
    <hiredAs externalID="PROD331">
      <!-- Datum, od kdy je zaměstnanec zaměstnán v pozici. Viz atribut :Employee.dtmHiredFrom -->
      <from>2004-06-01</from>
      <!-- Hodnocení pracovního výkonu zaměstnance jeho nadřízeným. Viz atribut :Employee.fltRealEvaluation -->
      <realEvaluation>0.74</realEvaluation>
    </hiredAs>
  </employee>
  <!-- Zaměstnanec, jež byl vybrán bez pomoci systému. Jeho hodnoty prediktorů musí být tudíž specifikovány nyní. -->
  <employee externalID="4146">
    <hiredAs externalID="POKL074">
      <from>2003-10-15</from>
      <realEvaluation>0.59</realEvaluation>
    </hiredAs>
    <!-- Hodnoty prediktorů zaměstnance, z nichž může systém predikovat předpokládaný pracovní výkon -->
    <!-- "predictor@externalID" je externí identifikátor prediktoru. Viz atribut :Predictor.strExternalID -->
    <predictor externalID="BranchExperienceLength">
      <!-- Datum zjištění hodnoty prediktoru. Viz atribut :PredictorValue.dtmMeasuring -->
```

```

    <measured>2004-09-02</measured>
    <!-- Hodnota prediktoru u uchazeče. Viz atribut :PredictorValue.fltValue -->
    <value>8</value>
  </predictor>
  <predictor externalID="BranchEducationDegree">
    <measured>2004-09-02</measured>
    <value>2</value>
  </predictor>
  <predictor externalID="PrevEmployerReferenceGrade">
    <measured>2004-09-02</measured>
    <value>6</value>
  </predictor>
  <predictor externalID="BranchSkillTestScore">
    <measured>2004-09-04</measured>
    <value>36</value>
  </predictor>
</employee>
</workforce>

```

Ukázka kódu 1 – Příklad XML souboru přenášeného z HRMS do navrhovaného systému.

V budoucnu může samozřejmě komponenta zajišťovat i libovolnou jinou potřebu výměny informací mezi systémem a externími zdroji.

Komponenta Feedback

Komponenta implementuje zpětnou vazbu pro hodnocení kvality používaných znalostí o pracovních pozicích (viz kap. 7.1.6).

Komponenta Rights

Jelikož databáze systému bude obsahovat citlivá data, komponenta pro řízení přístupových práv má za úkol řídit přístup k jednotlivým částem systému podle definovaných pravidel. Pravidla musí být dostatečně flexibilní, avšak zároveň uživatelsky srozumitelná.

7.4 Diagramy tříd

Jednoduché metody pro přístup k privátním atributům objektů (tzv. *getter* a *setter*) v návrhu tříd většinou neuvádíme. Naopak jsou popsány metody pro zápis a čtení z databáze (typicky metody *load*, *save* a *delete*). Diagramy tříd jsou vytvořeny pouze pro nejpodstatnější komponenty *Jobs*, *Workforce* a *Model*. Diagramy se nacházejí v příloze A. Diagram zpravidla obsahuje nejen třídy příslušné komponenty, ale pro přehlednost i barevně odlišené třídy jiných komponent, pokud mezi komponentami existuje závislost.

Třídy jsem se rozhodl členit do komponent z důvodu logické návaznosti a přehledu. Přestože se třídy různých komponent stále budou nacházet ve stejném prostoru, každá komponenta představuje množinu tříd, které mezi sebou disponují těsnější závislostí, než je

tomu u tříd z rozdílných komponent. Příslušnost tříd ke komponentám je tedy optimální tehdy, pokud mezi komponentami existuje co možná nejméně závislostí. Někdy se tento požadavek formuluje tak, že v diagramu se třídami všech komponent by měl procházet co možná nejpřehlednější „řez“ oddělující jednotlivé komponenty [36]. Obrázek 27 zachycuje vztah závislosti přerušovanou šipkou, která vede od závislé komponenty směrem ke komponentě nezávislé.

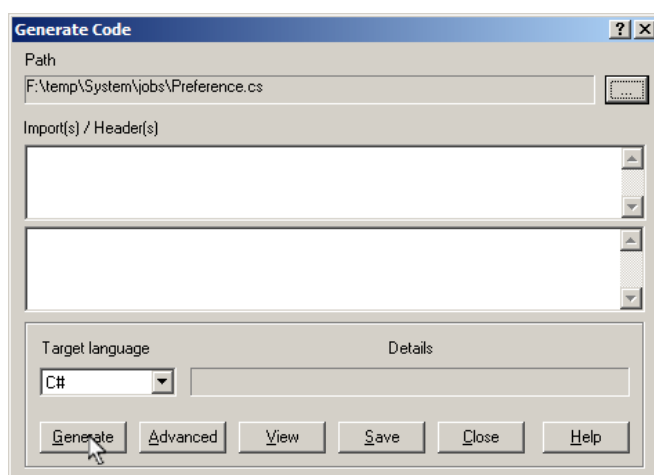
Při implementaci závisí realizace rozdělení tříd do komponent na použitém programovacím jazyce. Například:

- V jazyce C# bude komponentě odpovídat jmenný prostor vymezený blokem *namespace*, který obsahuje příslušné třídy. Vazbu na jiný jmenný prostor realizujeme klíčovým slovem *using*.
- V jazyce Java plní obdobnou funkci balíčky, deklarované klíčovým slovem *package*. Připojení tříd z jiného balíčku umožňuje klíčové slovo *import*.

8. Výsledky

Výsledkem diplomové práce je návrh systému, vypracovaný až do fáze diagramů tříd (viz kap. 7.4 a příloha A) a podrobného popisu tříd (viz příloha B).

V příloze C v elektronické podobě na příloženém CD se navíc nalézá soubor s návrhem systému vytvořený v aplikaci Enterprise Architect (viz kap. 5.2.2). Jestliže návrh v této aplikaci otevřeme a v okně *Project View* nalistujeme libovolnou třídu, volbou *Generate Code (forward engineer)* z kontextové nápovědy jednoduše vygenerujeme šablonu programového kódu této třídy do zvoleného jazyka. K dispozici jsou jazyky C++, Java, VB, VB.NET, C#, Delphi nebo PHP. Pak v podstatě zbývá pouze dopsat vnitřky metod.



Obrázek 28 – Generování šablony programového kódu třídy `Jobs::Preference`.

Dalším krokem úspěšné implementace základní funkčnosti aplikace by bylo vytvoření databázového schématu pro použitou databázi a naprogramování aplikační logiky (tzv. *business layer*), která třídy v jednotlivých komponentách využívá.

Výsledkem diplomové práce samozřejmě není pouze vytvořený návrh sám o sobě, ale také analýza problémové oblasti, na základě které by mělo být možné systém v budoucnu modifikovat a zdokonalovat.

9. Závěry a doporučení

Pokud bychom projekt zamýšleli skutečně realizovat v praxi, analýza problému i návrh systému by musely být daleko objemnější. Dokumentace k oběma fázím vývoje je však vzhledem k požadovanému rozsahu diplomové práce pojata úsporně. Podrobnější popis by si zasloužily jednotlivé případy užití, potažmo celé workflow práce se systémem. Chybí také detailní návrh komponent *External*, *Feedback*, *Rights* a *UserInterface*. Například bez komponenty *UserInterface* by uživatel nemohl se systémem komunikovat. Komponenta *Rights* sice není přímo nezbytná pro funkčnost, ovšem sofistikované řízení práv přístupu v rámci bezpečnostní politiky organizace jistě má své opodstatnění.

Patrně by bylo rovněž přínosné rozšířit komponentu *Model* o perzistenci objektů, aby tato umožňovala archivaci jednotlivých hodnocení uchazečů. Jak hodnoty prediktorů uchazečů, tak hodnotící funkční vztahy se totiž mohou v čase měnit, a bez této funkcionality bychom vůbec nedisponovali přístupem k informaci o hodnoceních realizovaných v minulosti.

Návrh počítá se zpětnou vazbou pro hodnocení kvality modelu, která je založena na dodatečném celkovém hodnocení přijatého zaměstnance nadřizujícím. V praxi se ovšem spíše používá pravidelné hodnocení podřizujících podle několika aspektů, což by šlo do návrhu systému také promítnout.

Zvolený čistě objektový přístup hodnotím velice kladně. Vývoj aplikace by byl podle mých zkušeností daleko méně pracný než při procedurálním programování. Také udržovat, rozšiřovat či měnit v budoucnu architekturu aplikace by mělo být daleko snazší.

Domnívám se, že kdyby došlo k implementaci systému, mohlo by jeho následné nasazení v organizaci znamenat přínos. Ten by plynul ze zvýšení efektivity práce, snížení fluktuace, kvalitnější péče o zákazníky a tím pádem i lepšího image firmy v jejich očích. Nižší fluktuace sekundárně povede k vyšší morálce na pracovišti a terciálně k udržení si stávajících kvalitních pracovníků, což s nasazením systému zdánlivě už vůbec nesouvisí. Nezanedbatelným přínosem by bylo také samotné zavedení propracovanějšího a férovějšího přístupu ke kandidátům během výběrového procesu, což zase vylepší obraz společnosti jak u přijatých, tak u odmítnutých uchazečů.

Na druhou stranu, nasazení systému, potažmo zkvalitnění metodiky náboru a výběru zaměstnanců, nepovede ke zlepšení situace lidských zdrojů organizace, pokud se tato nebude stejně pečlivě věnovat i ostatním aktivitám s řízením lidských zdrojů spojených.

Jinými slovy, „uhodnutí“ toho nejnadějnějšího uchazeče je bezcenné, pokud nezajistíme jeho následnou adaptaci, spravedlivé ohodnocení, možnost seberealizace a růstu.

10. Seznam použité literatury

- [1] ABRAHAM, J., MORRISON, J. *Personality profile matching*, A&M Psychometrics, 2004 (on-line)
Přístup z Internetu:
http://www.ppicentral.com/Pdf/profile_matching.pdf
- [2] AMBER, Scott W. *Modeling Style.info* (on-line), Copyright 2002-2003. Citace kap. 2.5.
Přístup z Internetu:
<http://www.modelingstyle.info/classDiagram.html>
- [3] BUCHAMAN, Z., SHEPPARD, P. *Ranking projects using the ELECTRE method* (on-line). New Zealand: University of Waikato. Přístup z Internetu:
<http://www.geointelligence.org/Athens2001/data/pdf/p58.pdf>
- [4] CHAN, P.M.L., HU, Y.F., SHERIFF, R.E. *Implementation of Fuzzy Multiple Objective Decision Making Algorithm in a Heterogeneous Mobile Environment* (on-line), Orlando (Florida, USA): IEEE Wireless Communications and Networking Conference (WCNC'02), 2002, s. 332-336. Přístup z Internetu:
<http://sci2s.ugr.es/keel/monografia/multiobjective/6-00993517.pdf>
- [5] *ChangingMinds.org, Halo Effect Theory Explanation* (on-line), Syque, 2002-2005.
Přístup z Internetu:
http://changingminds.org/explanations/theories/halo_effect.htm
- [6] DE CENZO, D. A., ROBBINS, S. P. *Human Resource Management*, USA: John Wiley & Sons, Inc., 1996. ISBN: 0-471-12420-6
- [7] *Decision Support Systems* (on-line žurnál), Elsevier, 2005
Přístup z Internetu:
<http://www.sciencedirect.com/science/journal/01679236>
- [8] DOMEOVÁ, L., *Decision Support Systems – WWW stránky k vyučovanému předmětu*, Praha: Česká zemědělská univerzita. Přístup z Internetu:
<http://pef.czu.cz/~DOMEOVA/DSS/>
- [9] DONNELLY, J., GIBSON, J., IVANCEVICH, J. *Management*, přel. V. Dolanský a J. Koubek, Praha: Grada Publishing, 1997. Přel. z *Management*, Richard D. Irwin, Inc., 1995. ISBN: 80-7169-422-3
- [10] *The ETS test collection* (on-line databáze testů), Princeton (NJ): ETS, 2005
Přístup z Internetu:
<http://www.ets.org/testcoll/index.html>
- [11] FULLER, R., CARLSON, C. *Fuzzy multiple criteria decision making: Recent developments* (on-line), *Fuzzy sets and Systems* vol. 78, s. 139-153, Elsevier B.V., 1996. Přístup z Internetu:
<http://www.sciencedirect.com/science/journal/01650114>
- [12] FULLER, R. *Fuzzy Sets and Systems* (e-portál), Åbo (Finland): Åbo Akademi,
Přístup z Internetu:
<http://www.abo.fi/~rfuller/fuzs.html>
- [13] FULLER R. *OWA Operators in Decision Making*, in: Christer Carlsson ed., *Exploring the Limits of Support Systems*, Åbo (Finland): Åbo Akademi tryckeri, TUCS General Publications, No. 3, Åbo, [ISBN 951-650-947-9, ISSN 1239-1905], 1996, s. 85-104
Přístup z Internetu:
<http://www.abo.fi/~rfuller/aeufit97.pdf>

- [14] GATEWOOD, R. D., FEILD, H. S. *Human Resource Selection*, USA: Harcourt College Publishers, Inc., 2001. ISBN: 0-03-031933-1
- [15] GREENGARD, C., RUSZCZYNSKI, A., *Decision Making Under Uncertainty*, New York (USA): Springer-Verlag, Inc., 2002. ISBN: 0-387-95465-1
- [16] GOODWIN, P., WRIGHT, G. *Decision Analysis for Management Judgment, Third Edition*, USA: John Wiley & Sons, 2003. ISBN: 0-470-86108-8
- [17] HAMILTON, C., MAIL, E. *Downshifting in Australia*, Australia Institute 2003 (on-line discussion paper) ISSN: 1322-5421. Přístup z Internetu:
http://www.tai.org.au/WhatsNew_Files/WhatsNew/DP50sum.pdf
- [18] HENDERSON, Kenneth. *Mistrovství v Transact-SQL*, přel. Karel Voráček, Praha: Computer Press, 2000, Přel. z *The Guru's Guide to Transact-SQL*, s. 317. ISBN: 80-7226-393-5
- [19] HORTON, T. *Object Oriented Analyses & Design – UML Class Models*, Virginia (USA): Department of Computer Science, University of Virginia, 2001. Prezentace č. CS 494 (on-line), citace slide G28. Přístup z Internetu:
<http://www.cs.virginia.edu/~horton/cs494/slides/e-class-models-s03.pdf>
- [20] HOWE, Denis. *The Free On-line Dictionary of Computing* (on-line), 1993-2003. Přístup z Internetu:
<http://www.foldoc.org/>
- [21] HINDLS, R. HRONOVÁ, S. SEGER, J. *Statistika pro ekonomy*, Praha: Grada Publishing, 2004. ISBN: 80-86419-59-2
- [22] *HR-Guide* (Human Resources e-portal), © HR-Guide.com, 1999-2005. Přístup z Internetu:
<http://www.hr-guide.com/>
- [23] JABLONSKÝ, J. *Operační výzkum – kvantitativní modely pro ekonomické rozhodování, Kapitola 9 – Vícekriteriální rozhodování*, Praha: Professional Publishing, 2002. ISBN: 80-86419-42-8
- [24] *Java™ 2 Platform, Standard Edition 5.0 API Specification* (on-line)
Status: Final Release. California (USA): Sun Microsystems, Inc, 2004. Přístup z Internetu:
<http://java.sun.com/j2se/1.5.0/docs/api/>
- [25] JIANG, Q., CHEN, C. *A multi-dimensional fuzzy decision support strategy* (on-line), *Decision Support Systems* vol. 38, p. 591-598, Elsevier B.V., 2005.
Přístup z Internetu:
<http://www.sciencedirect.com/science/journal/01679236>
- [26] JOHN, R. I. *Fuzzy Inferencing Systems - Problems and Some Solutions* (working paper n. 62), Leicester: School of Computer Sciences, De Montfort University, 1995
Přístup z Internetu:
<http://www.cse.dmu.ac.uk/~rij/newrep/newrep.html>
- [27] JOHNSON, L. *Human Resource Management* (elektronické přednášky), Texas (USA): University of North Texas, 2004. Přístup k Internetu:
<http://www.coba.unt.edu/MGMT/Johnson/courses/courses.htm>
- [28] JONTELA, T., et al. *Fuzzy Approach for Modeling Multiple Criteria in the Job Grouping Problem* (on-line), Turku (Finland): Turku Centre for Computer Science, TUCS Technical Report No 227, 1998. ISBN 952-12-0346-3. Přístup z Internetu:
<http://staff.cs.utu.fi/staff/jouni.smed/papers/TR227.pdf>

- [29] KADLČÍK, M. *Psychologie a sociologie řízení*, Hradec Králové: Gaudeamus, 2001. ISBN: 80-7041-951-2
- [30] KAEHLER, S. D. *Fuzzy Logic Tutorial*, Encoder (on-line), Seattle (USA): The Seattle Robotics Society, March 1998. Přístup z Internetu:
<http://www.seattlerobotics.org/encoder/mar98/fuz/flindex.html>
- [31] KOM, W., *Introduction to Object-Oriented databases*, Computer Systems. Cambridge (MA): MIT Press, 1990. ISBN 0-262-11124-1
- [32] KOONTZ, H., WEIHRICH, H., *Management*, přel. V. Dolanský, Praha: Victoria Publishing, 1993. Přel. z Management, McGraw – Hill, 1993. ISBN: 80-85605-45-7
- [33] KOSEK, Jiří. *Seriál o XML pro Softwarové noviny* (on-line verze seriálu), Praha: www.kosek.cz, 2000
Přístup z Internetu:
<http://www.kosek.cz/clanky/swn-xml/index.html>
- [34] KOSEK, Jiří. *XML pro každého – podrobný průvodce*, Praha: Grada Publishing, 2000, s. 14-16. ISBN: 80-7169-860-1
- [35] KRAVAL, Ija. *Objektové modelování a UML v praxi 2000*, Valašské Klobouky: Server objektových technologií, 2001 (elektronická kniha), s. 60 – 64. Přístup z Internetu:
<http://www.objects.cz>
- [36] KRAVAL, Ija. *Objektové modelování a UML v praxi 2000*, Valašské Klobouky: Server objektových technologií, 2001 (elektronická kniha), s. 158-159. Přístup z Internetu:
<http://www.objects.cz>
- [37] KRUSE, R. *Foundations of Fuzzy Systems*, Chichester: Wiley, 1994. ISBN: 0-471-94243-x
- [38] LARICHEV, O. I. *Ranking Multicriteria Alternatives: the method ZAPROS III* (vědecká práce), Moskva: Российская ассоциация искусственного интеллекта, 2001. Přístup z Internetu:
http://www.raai.org/library/papers/Larichev/Larichev_2001b.pdf
- [39] LOFTI, V., PEGELS, C. *Decision Support Systems for Management Science/Operations Research*, USA: Irwin, 1989. ISBN: 0-256-07859-9
- [40] MIKULECKÝ, P., PONCE, D. *Znalostní technologie I, II* (PPT prezentace), Hradec Králové: UHK-FIM, 1999-2001. Přístup z Internetu:
<https://oliva.uhk.cz/ZT2KF/presentations/ZT1.zip>
<https://oliva.uhk.cz/ZT2KF/presentations/ZT2.zip>
- [41] NOVÁK, V. *Základy fuzzy modelování*, Praha: BEN, 2000. ISBN: 80-7300-009-1
- [42] OSSENBRUGGEN, P. J. *Fundamental principles of systems analysis and decision-making*, USA: John Wiley & Sons, Inc., 1994. ISBN: 0-471-52156-6
- [43] POND, V. *XML To The Max: Get More Power Out Of Your SQL Server*, (on-line), Technet Magazine, Spring 2005, Microsoft Corporation, 2005. Přístup z Internetu:
<http://www.microsoft.com/technet/technetmag/issues/2005/05/SQLXML/default.aspx>
- [44] PRODANOVIC, P., *Fuzzy Set Ranking Methods and Multiple Expert Decision Making* (on-line), Ontario (Canada): The University of Western Ontario, 2001. Přístup z Internetu:
http://www.eng.uwo.ca/research/iclr/fids/Documents/Fuzzy_set_ranking.pdf
- [45] RENALS, Steve. *Java and UML for programmers*, Department of Computer Science, The University of Sheffield, 2002, prezentace COM 6050 (on-line),

- citace Lecture 10, s. 6/24. Přístup z Internetu:
<http://www.dcs.shef.ac.uk/~sjr/com6050>
- [46] *Resource Associates, Inc.* (www stránky společnosti)
Přístup z Internetu:
<http://www.resourceassociates.com/>
- [47] RICHARDSON, F. *Test administration, scoring and interpretation* (elektronická přednáška), London: South Bank University, 2004. Přístup z Internetu:
www.lsbu.ac.uk/psycho/teaching/ppfiles/pit-2-04-05.ppt
- [48] ŘEZANKOVÁ, H., MAREK, L., VRABEC, M. *IATAT – Interaktivní učebnice statistiky*, 2001. Přístup z Internetu:
http://badame.vse.cz/iastat/typy_promennych.html
- [49] SCHMULLER, Joseph. *Myslíme v jazyku UML*, přel. Jiří Hynek, Praha: Grada Publishing, 2001, s. 172-173. Přel. z Teach yourself UML in 24 hours, Sams, ISBN 80-247-0029-8
- [50] SCHMULLER, Joseph. *Myslíme v jazyku UML*, přel. Jiří Hynek, Praha: Grada Publishing, 2001, Přel. z Teach yourself UML in 24 hours, Sams, ISBN 80-247-0029-8
- [51] SLAVÍČEK, Václav. *Nástroj pro tvorbu křížovek – bakalářská práce*, Hradec Králové: UHK-FIM, 2004. Přístup z Internetu:
<http://www.slavicek.net/uhk/aik/thesis.zip>
- [52] *Test reviews* (on-line databáze testů), Lincoln (NE): Buros Institute of Mental Measurements, 2005. Přístup z Internetu:
<http://buros.unl.edu/buros/jsp/search.jsp>
- [53] VAN RIEL, A. *Antecedents of Effective Decision Making: A Cognitive Approach*, Maastricht (Netherlands): Faculty of Economics and Business Administration, Maastricht University, 2003 (on-line). Přístup z Internetu:
www.seminar.hec.ulg.ac.be/docs/Sem05.04.13_VanRieletal.pdf
- [54] *Velká encyklopedie EU*, Praha: Ministerstvo zahraničních věcí ČR, 2000 (on-line), kap. *Právo ES*. Přístup z Internetu:
<http://library.muni.cz/EU/>
- [55] *Why should I use SAX?*
In Microsoft XML Core Services 4.0 - SAX2 Developer's Guide,
MSDN Library (CD ROM), version July 2003, Microsoft Corporation 1987-2003.

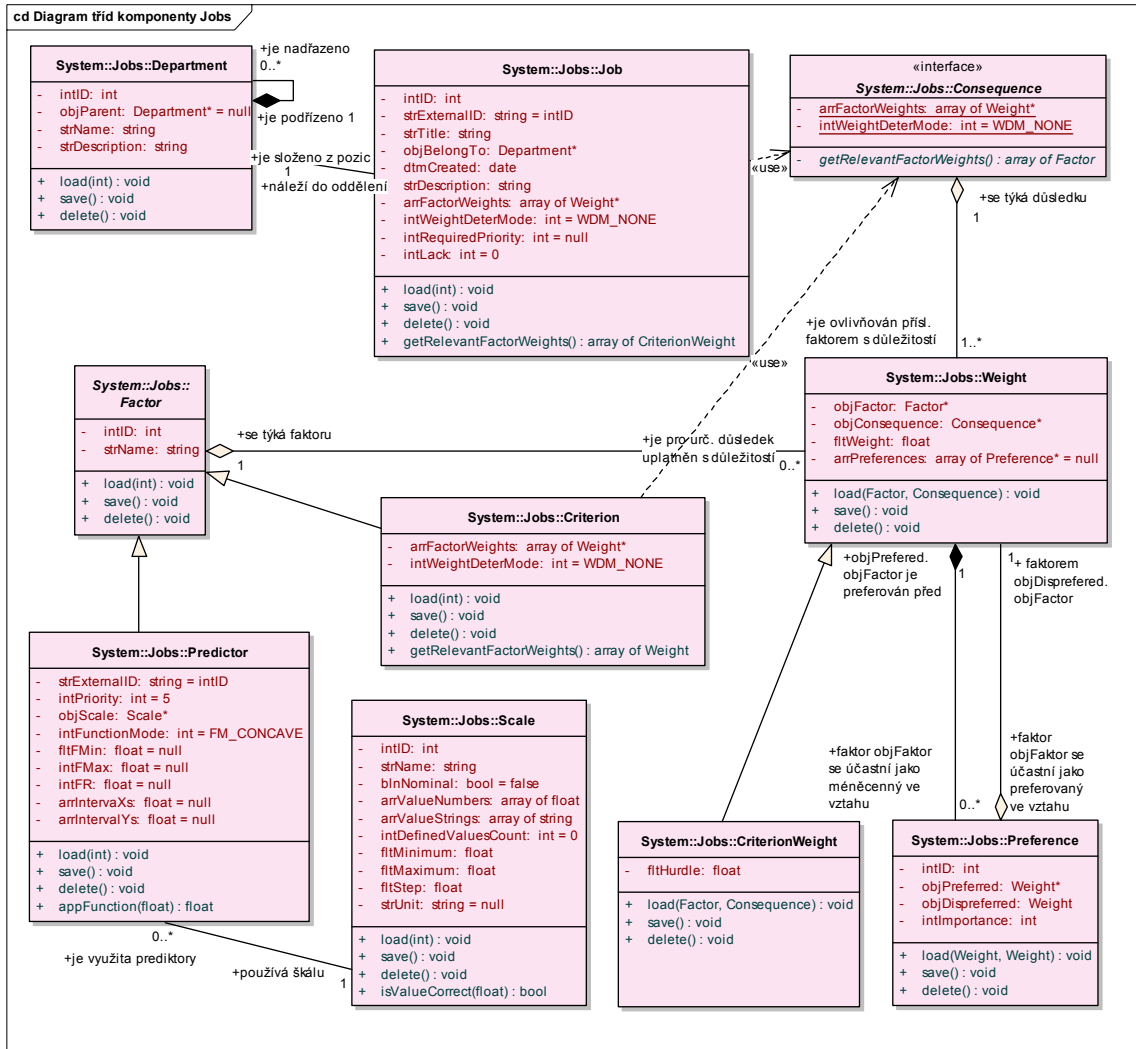
11. Přílohy

- A. UML diagramy tříd
- B. Detailní popis tříd
- C. Soubor s návrhem systému v prostředí Enterprise Architect 4.10.

Příloha C je obsažena na CD spolu s elektronickou podobou diplomové práce.

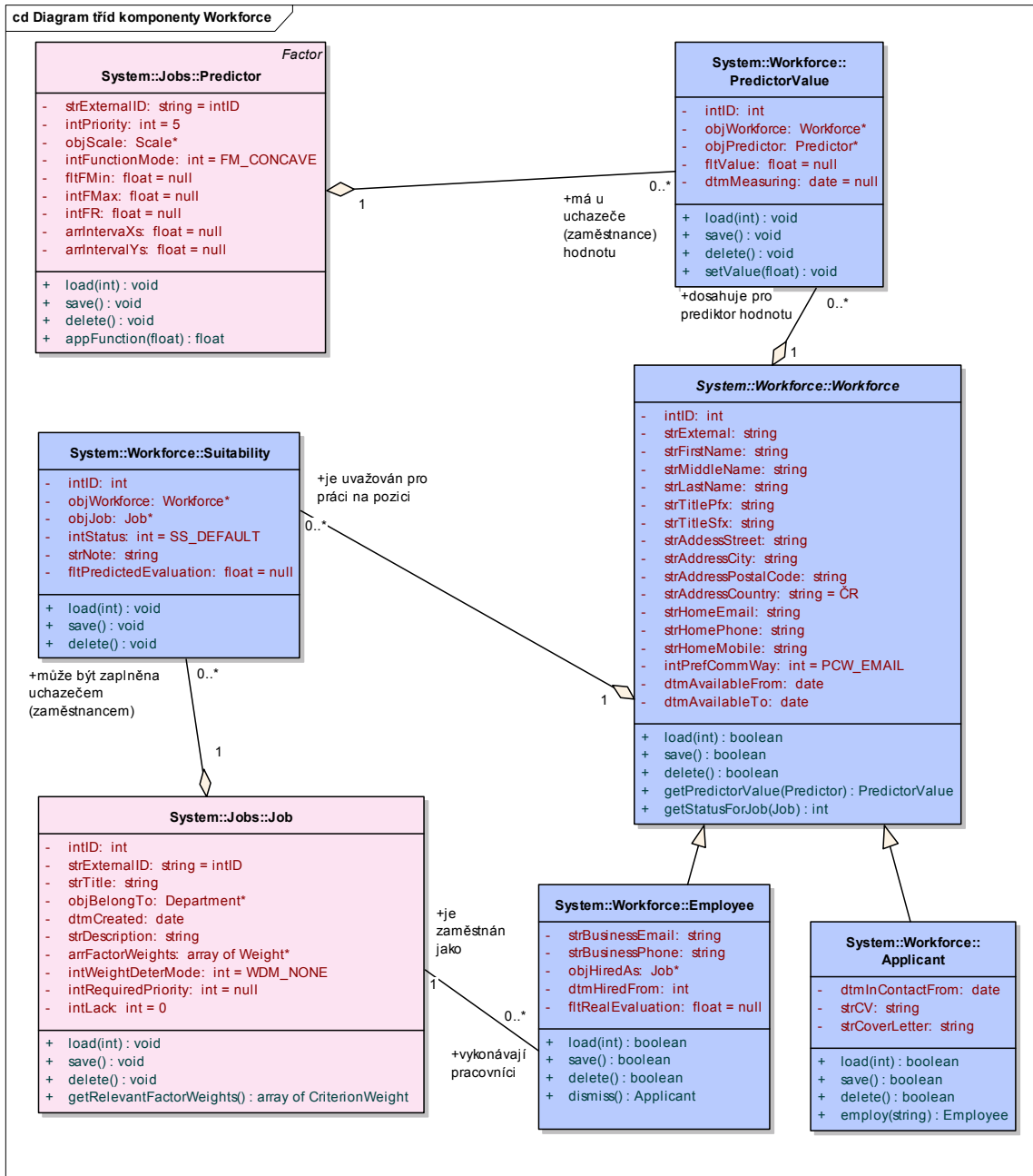
A. UML Diagramy tříd

Komponenta Jobs



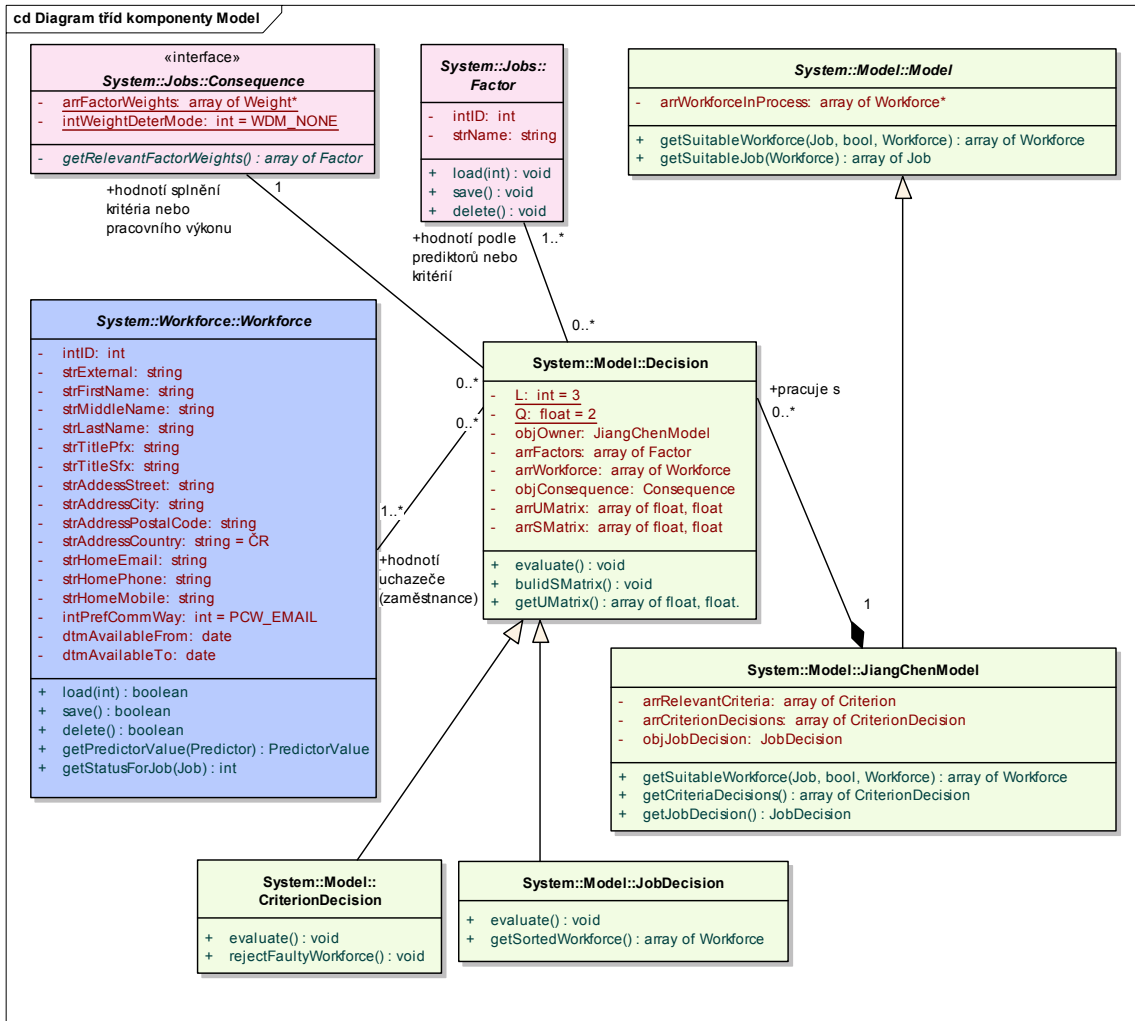
Obrázek 29- Diagram tříd komponenty Jobs.

Komponenta Workforce



Obrázek 30 - Diagram tříd komponenty Workforce.

Komponenta Model



Obrázek 31 – Diagram tříd komponenty Model.

B. Detailní popis tříd

Komponenta Jobs

Jobs::Job

Typ: *public* Class

Třída reprezentuje pracovní pozici. Hodnoty atributů obsažených v této třídě a v souvisejících třídách (zejména ve třídě *CriterionWeight* představující váhy kritérií) by v praxi měly vycházet z analýzy a následného popisu a specifikace pracovní pozice (viz kap. 6.3).

Spojení

- Agregáčn  odkaz ze třídy *Suitability*
- Agregáčn  odkaz na třídu *Department*
- Asociační odkaz ze třídy *Employee*
- Odkaz „Use“ na rozhraní *Consequence*

Jobs::Job Atributy

Atribut	Typ	Poznámky
intID	private : <i>int</i>	Interní identifikátor pracovní pozice v systému.
strExternalID	private : <i>string</i>	Identifikátor pracovní pozice v externích datových zdrojích. Poč. hodnota: <i>intID</i> ;
strTitle	private : <i>string</i>	Název pracovní pozice.
objBelongTo	private : <i>Department</i>	Odkaz na oddělení, do něž pozice spadá.
dtmCreated	private : <i>date</i>	Datum vzniku nebo poslední modifikace pozice.
strDescription	private : <i>string</i>	Textový popis pracovní pozice. Ten obvykle obsahuje výčet povinností a zodpovědnosti držitele pozice.
arrFactorWeights	private : <i>array of Weight</i>	Váhy kritérií relevantních pro tuto pozici.
intWeightDeterMode	private : <i>int</i>	Způsob, jakým jsou zadávány váhy (ručně nebo Saatyho metodou). Poč. hodnota: <i>WDM_NONE</i> ;
intRequiredPriority	private : <i>int</i>	Priorita prediktorů, které mají být zjištěny a zadány operátorem pro příslušné uchazeče do systému. Začíná-li výběr, měl by být atribut nastaven na hodnotu 9. Poté

		<p>rozhodovatel postupně redukuje množinu hodnocených uchazečů a snižuje také hodnotu tohoto atributu. Snížení hodnoty atributu vyvolá případ užití 7.1.2, načež by měl operátor do systému přidat hodnoty prediktorů s novou, nižší prioritou.</p> <p>Neprobíhá-li aktuálně žádný výběr na tuto pracovní pozici, obsahuje atribut hodnotu <i>null</i>.</p> <p>Poč. hodnota: <i>null</i>;</p>
intLack	private : <i>int</i>	<p>Množství neobsazených instancí pracovní pozice, které je třeba zaplnit. Pokud manažer uchazeče přeřadí na zaměstnance (tj. přijme jej, viz kap. 7.1.5), atribut se u příslušné pracovní pozice dekrementuje.</p> <p>Poč. hodnota: <i>0</i>;</p>

Jobs::Job Metody

Metoda	Typ	Poznámky
load (<i>int</i>)	public: <i>void</i>	<p><i>param:</i> aID [<i>int</i> - in] Identifikátor pozice.</p> <p>Načte pracovní pozici s příslušným identifikátorem z databáze.</p>
save ()	public: <i>void</i>	<p>Zkontroluje formát nastavených atributů a uloží informace o pracovní pozici do databáze. Metoda samozřejmě vnořeně volá metodu <i>save()</i> instancí třídy <i>CriterionWeight</i>, na něž odkazují prvky kolekce <i>arrFactorWeights</i>.</p>
delete ()	public: <i>void</i>	<p>Vymaže informace o pracovní pozici z databáze včetně souvisejících objektů (instancí třídy <i>CriterionWeight</i>).</p>
getRelevantFactorWeights ()	public: <i>array of CriterionWeight</i>	<p>Vrátí kolekci kritérií relevantních pro tuto pracovní pozici (podle <i>arrFactorWeights</i>).</p>

Jobs::Department

Typ: *public* **Class**

Oddělení organizace. Tato oddělení lze hierarchicky seskupovat. Jediný význam této třídy tkví ve vytvoření hierarchické struktury jednotlivých oddělení organizace, k

jejímž uzlům lze přiřazovat pracovní pozice (objekty třídy *Job*). Pracovní pozice tak budou přehledně zařazené.

Spojení

- Agregáčn  odkaz ze třídy *Job*
- Agregáčn  odkaz ze třídy *Department*

Jobs::Department Atributy

Atribut	Typ	Poznámky
intID	private : <i>int</i>	Intern� identifikátor oddělení v systému.
objParent	private : <i>Department</i>	Odkaz na nadřazené oddělení, příp. <i>null</i> , pokud se jedná o oddělení nejvyšší úrovně (např. "Ředitelství"). Poč. hodnota: <i>null</i> ;
strName	private : <i>string</i>	Název oddělení.
strDescription	private : <i>string</i>	Popis oddělení, kontaktní údaje atp.

Jobs::Department Metody

Metoda	Typ	Poznámky
load (int)	public: <i>void</i>	<i>param</i> : aID [<i>int</i> - in] Identifikátor oddělení. Načte informace o oddělení z databáze.
save ()	public: <i>void</i>	Uloží informace o oddělení do databáze.
delete ()	public: <i>void</i>	Smaže oddělení z databáze, včetně podřízených oddělení a přiřazených pozic.

Jobs::Factor

Typ: *public abstract* **Class**

Obecný faktor, ovlivňující vhodnost uchazeče pro určitou pozici. Faktor lze dále specializovat na prediktor nebo kritérium

Spojení

- Agregáčn  odkaz ze třídy *Weight*
- Asociační odkaz ze třídy *Decision*

- Generalizační odkaz ze třídy *Predictor*
- Generalizační odkaz ze třídy *Criterion*

Jobs::Factor Atributy

Atribut	Typ	Poznámky
intID	private : <i>int</i>	Interní identifikátor faktoru v systému.
strName	private : <i>string</i>	Název prediktoru, např. "Výsledek testu znalosti práce s PC", nebo kritéria, např. "Uživatelská znalost práce s PC".

Jobs::Factor Metody

Metoda	Typ	Poznámky
load (int)	public: <i>void</i>	<i>param:</i> aID [<i>int</i> - in] Interní identifikátor faktoru. Načte vlastnosti faktoru z databáze.
save ()	public: <i>void</i>	Uloží informace o faktoru do databáze.
delete ()	public: <i>void</i>	Vymaže faktor z databáze.

Jobs::Weight

Typ: *public* **Class**

Třída představuje význam, jaký je přiřazen faktoru vzhledem k určitému důsledku. Odkazy na objekty této třídy se uchovávají v kolekci *:Consequence.arrFactorWeights*. Jestliže je faktor pro důsledek irelevantní, žádný objekt se nevytváří. Význam faktoru je konkrétně determinován atributem váhy faktoru vzhledem k důsledku (atribut *fltWeight*).

Spojení

- Agregační odkaz na třídu *Preference*
- Agregační odkaz ze třídy *Preference*
- Agregační odkaz na rozhraní *Consequence*
- Agregační odkaz na třídu *Factor*
- Generalizační odkaz ze třídy *CriterionWeight*

Jobs::Weight Atributy

Atribut	Typ	Poznámky
objFactor	private : <i>Factor</i>	Faktor, jehož se objekt týká (tj. prediktor nebo kritérium).
objConsequence	private : <i>Consequence</i>	Objekt (kritérium nebo pracovní pozice), který je ovlivněn následkem změny faktoru. Jestliže má odkazovaný objekt : <i>Consequence</i> nastaven atribut <i>intWeightDeterMode</i> na hodnotu <i>WTD_SAATY</i> , bude se v kolekci <i>arrReferences</i> uchovávat síla preference mezi tímto a jinými faktory.
fltWeight	private : <i>float</i>	<p><u>Pro faktor třídy <i>Predictor</i></u> Váha prediktoru vzhledem ke k-tému kritériu v rozmezí [0, 1]. Atribut představuje prvek w_i^k váhového vektoru W^k, viz krok 1.e použité strategie. Musí tedy platit</p> $\sum_{i=1}^{m_k} :Consequence.arrFactorWeights[i].fltWeight = 1.$ <p><u>Pro faktor třídy <i>Criterion</i></u> Váha kritéria vzhledem k pozici v rozmezí [0, 1]. Atribut představuje prvek w_i^u váhového vektoru W^u, viz krok 2.a použité strategie.</p>
arrPreferences	private : <i>array of</i> <i>Preference</i>	<p>Jestliže byla pro stanovení vah faktorů použita Saatyho metoda (tj. <i>objConsequence.intWeightDeterMode==WTD_SAATY</i>), obsahuje tato kolekce 0 až (m-1) prvků, kde m je počet faktorů ovlivňujících důsledek. Prvek je v kolekci přítomen jestliže faktor <i>objFactor</i> je preferován před nějakým jiným faktorem. Pokud je naopak faktor <i>objFactor</i> před jiným faktorem dispreferován, má vztah ve své kolekci <i>arrReferences</i> uveden onen druhý faktor. Jsou-li oba faktory rovnocenné, objekt :<i>Preference</i> není vytvářen ani u jednoho z nich.</p> <p>Poč. hodnota: <i>null</i>;</p>

Jobs::Weight Metody

Metoda	Typ	Poznámky
load (<i>Factor</i> , <i>Consequence</i>)	public: <i>void</i>	<p><i>param</i>: aFactor [<i>Factor</i> - in] Faktor, pro nějž má být načtena váha.</p> <p><i>param</i>: aConsequence [<i>Consequence</i> - in] Důsledek, pro nějž má být načtena váha (konkrétně odkaz</p>

		na objekt, který implementuje rozhraní <i>Consequence</i>). Načte objekt z databáze podle specifikovaného faktoru (<i>aFactor.intID</i>) a důsledku (<i>aConsequence.intID</i>).
save ()	public: <i>void</i>	Uloží modifikovaný objekt do databáze.
delete ()	public: <i>void</i>	Smaže objekt z databáze.

Jobs::Preference

Typ: *public* **Class**

Třída představuje vztah preference faktoru (prediktoru nebo kritéria) udaného ve vlastnosti *objPreferred* před faktorem udaným ve vlastnosti *objDispreferred*. Třída se používá, pokud uživatel pro stanovování vah prediktorů nebo kritérií uplatnil Saatyho metodu. V takovém případě je třeba nastavené preference uchovat, aby mohly být později snadno modifikovány a nemusel se preferenční trojúhelník vytvářet kompletně znovu.

Spojení

- Agregáčn  odkaz ze třídy *Weight*
- Agregáčn  odkaz na třídu *Weight*

Jobs::Preference Atributy

Atribut	Typ	Poznámky
intID	private : <i>int</i>	Intern� identifikátor preferenčního vztahu.
objPreferred	private : <i>Weight</i>	Odkazuje na objekt třídy <i>Weight</i> , jehož faktor <i>objFactor</i> se účastn� vztahu jako preferovaný.
objDispreferred	private : <i>Weight</i>	Odkazuje na objekt třídy <i>Weight</i> , jehož faktor <i>objFactor</i> se účastn� vztahu jako méněcenný.
intImportance	private : <i>int</i>	Síla preferenčního vztahu mezi faktory <i>objPreferred.objFactor</i> a <i>objDispreferred.objFactor</i> . Můž� nabývat hodnot {2,3,4,5,6,7,8,9}, viz Tabulka 4. Jsou-li oba faktory rovnocenné, objekt se nevytváří.

Jobs::Preference Metody

Metoda	Typ	Poznámky
load (<i>Weight</i> , <i>Weight</i>)	public: void	<i>param</i> : aPreferred [<i>Weight</i> - in] Odkaz na objekt třídy <i>Weight</i> s preferovaným faktorem. <i>param</i> : aDispreferred [<i>Weight</i> - in] Odkaz na objekt třídy <i>Weight</i> s méněcenným faktorem. Načte informace o preferenčním vztahu z databáze.
save ()	public: void	Uloží informace o objektu do databáze.
delete ()	public: void	Vymaže objekt z databáze.

Jobs::Consequence

Typ: public abstract «interface» **Interface**

Rozhraní je implementováno do tříd, které představují následek působení jednoho nebo několika faktorů. Konkrétně:

- v kritériu, neboť to je důsledkem několika působících prediktorů,
- v pracovní pozici, jelikož výkon pracovníka je ovlivněn působením několika kritérií.

Spojení

- Agregací odkaz ze třídy *Weight*
- Asociační odkaz ze třídy *Decision*
- Odkaz „Use“ ze třídy *Criterion*
- Odkaz „Use“ ze třídy *Job*

Jobs::Consequence Atributy

Atribut	Typ	Poznámky
arrFactorWeights	private static : <i>array of Weight</i>	Kolekce slouží jako podklad při výběru uchazečů. Obsahuje seznam relevantních faktorů (prediktorů či kritérií), nutných pro úspěšné vykonávání práce na pozici, vč. charakteristiky důležitosti (váhy) a v případě kritérií také minimální hodnoty kritéria.

intWeightDeterMode	private static : <i>int</i>	Mód, v jakém došlo k nastavení váhy faktoru. Může nabývat hodnoty rovné jedné z následujících konstant: <ul style="list-style-type: none"> • <i>WDM_NONE</i> ... váhy zatím nejsou určeny, • <i>WDM_SAAATY</i> ... obsluha se rozhodla určovat váhy Saatyho metodou, • <i>WDM_ADVANCED</i> ... obsluha se rozhodla určovat váhy ručně přímou definicí váhového vektoru, tzn. atributu <i>arrFactorWeights[i].fltWeight</i>. <p>Poč. hodnota: <i>WDM_NONE</i>;</p>
--------------------	--------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Jobs::Consequence Metody

Metoda	Typ	Poznámky
getRelevantFactorWeights ()	private: <i>array of Factor</i>	Pro tento důsledek (kritérium nebo pozici) metoda vrátí kolekci relevantních faktorů (prediktorů nebo kritérií) a jejich vah. Relevantní faktory jsou takové, které mají nenulovou váhu.

Jobs::Predictor

Typ: *public Class*

Rozšiřuje: *Factor*.

Prediktor, jak je popsán v kap. 6.5.1.

Spojení

- *Agregační odkaz ze třídy PredictorValue*
- *Asociační odkaz ze třídy Scale*
- *Generalizační odkaz na třídu Factor*

Jobs::Predictor Atributy

Atribut	Typ	Poznámky
strExternalID	private : <i>string</i>	Identifikátor prediktoru v externích datových zdrojích. Poč. hodnota: <i>intID</i> ;
intPriority	private : <i>int</i>	Priorita prediktoru, viz kapitoly 7.1.2 a 7.1.3. Poč. hodnota: <i>5</i> ;

objScale	private : <i>Scale</i>	Škála přiřazená prediktoru.
intFunctionMode	private : <i>int</i>	Způsob, jakým je definována aplikační funkce prediktoru (viz kap. 7.1.3): <ul style="list-style-type: none"> • <i>FM_CONCAVE</i> ... Vztahem (15) a parametry m_i, M_i a r, tzn. atributy <i>fltFMin</i>, <i>fltFMax</i>, <i>fltFR</i>. • <i>FM_INTERVAL</i> ... lineární interpolací hodnot v intervalech, viz Obrázek 19, tzn. poli <i>arrIntervalXs</i> a <i>arrIntervalYs</i>. <p>Poč. hodnota: <i>FM_CONCAVE</i>;</p>
fltFMin	private : <i>float</i>	Jestliže <i>intFunctionMode</i> == <i>FM_CONCAVE</i> , atribut obsahuje parametr m_i pro vztah (15) aplikační fce. <p>Poč. hodnota: <i>null</i>;</p>
intFMax	private : <i>float</i>	Jestliže <i>intFunctionMode</i> == <i>FM_CONCAVE</i> , atribut obsahuje parametr M_i pro vztah (15) aplikační fce. <p>Poč. hodnota: <i>null</i>;</p>
intFR	private : <i>float</i>	Jestliže <i>intFunctionMode</i> == <i>FM_CONCAVE</i> , atribut obsahuje parametr r pro vztah (15) aplikační fce. <p>Poč. hodnota: <i>null</i>;</p>
arrIntervaXs	private : <i>float</i>	Jestliže <i>intFunctionMode</i> == <i>FM_INTERVAL</i> , i -tý prvek pole obsahuje prvek m_i vektoru I_i (viz kap. 7.1.3) <p>Poč. hodnota: <i>null</i>;</p>
arrIntervalYs	private : <i>float</i>	Jestliže <i>intFunctionMode</i> == <i>FM_INTERVAL</i> , i -tý prvek pole obsahuje prvek Y_i vektoru I_i (viz kap. 7.1.3). Prvky pole nabývají hodnot v intervalu [0, 1]. <p>Poč. hodnota: <i>null</i>;</p>

Jobs::Predictor Metody

Metoda	Typ	Poznámky
load (<i>int</i>)	public: <i>void</i>	<i>param</i> : aID [<i>int</i> - in] Interní identifikátor faktoru. Načte vlastnosti prediktoru z databáze.
save ()	public: <i>void</i>	Uloží informace o prediktoru do databáze.
delete ()	public: <i>void</i>	Vymaže prediktor z databáze.
appFunction (<i>float</i>)	public: <i>float</i>	<i>param</i> : aValue [<i>float</i> - in]

		<p>Hodnota prediktoru v rozsahu definovaném přiřazenou škálou <i>objScale</i>. Jestliže hodnota neodpovídá škále, vyvolá se výjimka.</p> <p>Metoda ve svém těle obsahuje kód implementující aplikační fci. Vrací stupeň příslušnosti specifikované hodnoty v množině "dobrých hodnot" v rozmezí [0, 1].</p>
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Jobs::Scale

Typ: `public Class`

Škála, do které lze vynášet hodnotu prediktoru. Hodnota v této škále je následně pomocí aplikační funkce namapována do intervalu [0, 1], jímž se udává míra členství prediktoru v množině "dobrých hodnot". Více o škálách viz kap. 7.1.3. Základní klasifikaci hodnot prediktorů udává Tabulka 1 v kap. 6.5.2. My budeme škály kategorizovat pomocí několika jednoduchých atributů.

Spojení

- Asociační odkaz na třídu *Predictor*

Jobs::Scale Atributy

Atribut	Typ	Poznámky
intID	private : <i>int</i>	Interní identifikátor škál v systému.
strName	private : <i>string</i>	Název škály, např. "z-score", "stanine-score", "stupeň dosaženého vzdělání" nebo "roky praxe".
blnNominal	private : <i>bool</i>	Příznak, že je škála nominální. Jediný rozdíl nominální a jinou škálou spočívá v tom, že se operátorovi nezobrazují číselné, ale pouze přiřazené jazykové hodnoty (viz atributy <i>arrValueNumbers</i> a <i>arrValueStrings</i>). Poč. hodnota: <i>false</i> ;
arrValueNumbers	private : <i>array of float</i>	Pole s číselným vyjádřením hodnot těch prediktorů k nimž existuje jazykový ekvivalent. Např. pro prediktor "stupeň dosaženého vzdělání" bude pole obsahovat prvky {0,1,2,3,4}. Pojmenovat všechny přípustné hodnoty je povinné u nominálních hodnot (<i>blnNominal==true</i>). Lze však pojmenovat některé hodnoty i u spojitě škály. V takovém případě bude operátorovi seznam těchto hodnot zobrazen

		informativně. Např. u spojité škály "procenta" můžeme hodnotu 50 nazvat "polovina".
arrValueStrings	private : <i>array of string</i>	Pole s textovým vyjádřením hodnot prediktorů, např. pro prediktor "stupeň dosaženého vzdělání" bude obsahovat prvky {"ZŠ", "SŠ", "SŠ s maturitou", "Bc", "Mgr"}.
intDefinedValuesCount	private : <i>int</i>	Počet pojmenovaných hodnot, tj. počet prvků v poli <i>arrValueNumbers</i> , resp. <i>arrValueStrings</i> . Poč. hodnota: 0;
fltMinimum	private : <i>float</i>	Minimální hodnota. Např. u škály "procenta" bude minimální hodnotou 0.
fltMaximum	private : <i>float</i>	Maximální hodnota. Např. u škály "procenta" bude maximální hodnotou 100.
fltStep	private : <i>float</i>	Toto vlastnost omezuje operátora v tom, jaké nejmenší mezilehlé hodnoty diskrétního intervalu může používat. Typické hodnoty jsou: <ul style="list-style-type: none"> • 1 pro celá čísla Z • 0.1 pro hodnoty s přesností na jednu desetinu • 0 pro spojité interval s reálnými čísly (přesnost je zde dána implementací programovacího jazyka a databázovým systémem).
strUnit	private : <i>string</i>	Informativní název jednotky, např. "%", "roky", "body" atp. Veličina může být samozřejmě i bezrozměrná (hodnota <i>null</i>). Poč. hodnota: <i>null</i> ;

Jobs::Scale Metody

Metoda	Typ	Poznámky
load (<i>int</i>)	public: <i>void</i>	<i>param:</i> aID [<i>int</i> - in] Identifikátor škály. Načte hodnotu prediktoru z databáze podle specifikovaného identifikátoru.
save ()	public: <i>void</i>	Uloží informace o škále do databáze.
delete ()	public: <i>void</i>	Vymaže škálu z databáze.
isValueCorrect (<i>float</i>)	public: <i>bool</i>	param: aValue [<i>float</i> - in] Zkoumaná hodnota prediktoru. Zkontroluje, zda specifikovaná hodnota odpovídá omezením škály (minimu, maximu, kroku).

Jobs::Criterion

Typ: `public Class`

Rozšiřuje: *Factor*.

Kritérium, jak je popsáno v kap. 6.5.1.

Spojení

- Odkaz „Use“ na rozhraní *Consequence*
- Generalizační odkaz na třídu *Factor*

Jobs::Criterion Atributy

Atribut	Typ	Poznámky
arrFactorWeights	private : <i>array of Weight</i>	Kolekce obsahuje seznam vah prediktorů relevantních k tomuto kritériu.
intWeightDeterMode	private : <i>int</i>	Poč. hodnota: <i>WDM_NONE</i> ;

Jobs::Criterion Metody

Metoda	Typ	Poznámky
load (int)	public: <i>void</i>	<i>param:</i> aID [<i>int</i> - in] Interní identifikátor kritéria. Načte vlastnosti kritéria z databáze.
save ()	public: <i>void</i>	Uloží informace o kritériu do databáze.
delete ()	public: <i>void</i>	Vymaže kritérium z databáze.
getRelevantFactorWeights ()	public: <i>array of Weight</i>	Viz stejnojmenná metoda předka.

Jobs::CriterionWeight

Typ: `public Class`

Rozšiřuje: *Weight*.

Třída představuje význam, jaký je přiřazen kritériu vzhledem k určité pracovní pozici. Odkazy na objekty této třídy se uchovávají v kolekci `:Job.arrFactorWeights`. Jestliže je kritérium pro pozici irelevantní, žádný objekt se nevytváří.

Význam kritéria je determinován dvěma atributy:

- váhou kritéria (atribut *fltWeight*),
- nejhorší přípustnou hodnotou kritéria (atribut *fltHurdle*).

Spojení

- Generalizační odkaz na třídu *Weight*

Jobs::CriterionWeight Atributy

Atribut	Typ	Poznámky
fltHurdle	private : <i>float</i>	Atribut představuje maximální stupeň příslušnosti kritéria v nejhorším vzoru rozhodnutí. Jedná se o konstantu z^k , použitou v kroku 1.g strategie.

Jobs::CriterionWeight Metody

Metoda	Typ	Poznámky
load (<i>Factor</i> , <i>Consequence</i>)	public: <i>void</i>	<i>param</i> : aFactor [<i>Factor</i> - in] Kritérium, pro něž má být načtena váha. <i>param</i> : aConsequence [<i>Consequence</i> - in] Pracovní pozice, pro niž má být načtena váha. Načte objekt z databáze podle specifikovaného kritéria (<i>aFactor.intID</i>) a pozice (<i>aConsequence.intID</i>).
save ()	public: <i>void</i>	Uloží modifikovaný objekt do databáze.
delete ()	public: <i>void</i>	Smaže objekt z databáze.

Komponenta Workforce

Workforce::Workforce

Typ: *public abstract* **Class**

Pracovní síla, ať už právě zaměstnaná v organizaci či nikoliv. Pracovní sílu tedy můžeme dále specializovat na uchazeče, který v organizaci aktuálně nepracuje, a zaměstnance, který zde zaměstnán je.

Spojení

- Agregáčn  odkaz ze t r dy *Suitability*
- Agregáčn  odkaz ze t r dy *PredictorValue*
- Asocia n  odkaz na t r du *Decision*
- Generaliza n  odkaz ze t r dy *Applicant*
- Generaliza n  odkaz ze t r dy *Employee*

Workforce::Workforce Atributy

Atribut	Typ	Pozn�mky
intID	private : <i>int</i>	Intern� syst�mov� identifik�tor pracovn� s�ly.
strExternal	private : <i>string</i>	Identifik�tor prac. s�ly (zpravidla zam�stnance) v extern�ch datov�ch zdroj�ch.
strFirstName	private : <i>string</i>	Prvn� jm�no pracovn� s�ly.
strMiddleName	private : <i>string</i>	Prost�edn� jm�no pracovn� s�ly.
strLastName	private : <i>string</i>	P�rjmen� pracovn� s�ly.
strTitlePfx	private : <i>string</i>	Titul p�ed jm�nem.
strTitleSfx	private : <i>string</i>	Titul za jm�nem.
strAddressStreet	private : <i>string</i>	Bydlišt� - ulice, �slo popisn� a orienta�n�.
strAddressCity	private : <i>string</i>	Bydlišt� - Obec a �st obce.
strAddressPostalCode	private : <i>string</i>	Bydlišt� - poštovn� sm�rova� �slo.
strAddressCountry	private : <i>string</i>	Bydlišt� - zem� Po�. hodnota: „�R“;
strHomeEmail	private : <i>string</i>	Soukrom� kontaktn� e-mailov� adresa.
strHomePhone	private : <i>string</i>	Telefon dom�.
strHomeMobile	private : <i>string</i>	Soukrom� mobiln� telefon.
intPrefCommWay	private : <i>int</i>	Preferovan� zp�sob komunikace - m�e nab�vat jedn� z n�sleduj�c�ch konstant:

		<ul style="list-style-type: none"> • <i>PCW_EMAIL</i> ... e-mailem, • <i>PCW_PHONEHOME</i> ... telefonem domů, • <i>PCW_PHONEMOBILE</i> ... mobilním telefonem, • <i>PCW_ADDRESS</i> ... poštou <p>Poč. hodnota: <i>PCW_EMAIL</i>;</p>
dtmAvailableFrom	private : <i>date</i>	<p>Datum, odkdy je možné pracovní sílu zařazovat do výběru. V případě uchazeče je význam zřejmý - organizace má souhlas držet jeho osobní údaje a zařazovat jej do výběru po určité časové období. Potom by měl být smazán.</p> <p>Časový interval má však smysl i u zaměstnanců. Například současný zaměstnanec v klíčové pozici odcházející za půl roku do důchodu by mohl být vhodným kandidátem pro méně náročnou pozici otevíranou v té době (obdobně v případě tzv. downshiftingu [28]).</p>
dtmAvailableTo	private : <i>date</i>	<p>Datum, od kdy již pracovní sílu nebudeme zařazovat do žádného výběru. Uchazeč by měl být po dosažení tohoto data vymazán z databáze.</p>

Workforce::Workforce Metody

Metoda	Typ	Poznámky
load (<i>int</i>)	public: <i>boolean</i>	<p><i>param</i>: aID [<i>int</i> - in] interní identifikátor pracovní síly.</p> <p>Načte informace o pracovní síle z databáze podle atributu <i>intID</i>.</p>
save ()	public: <i>boolean</i>	<p>Provede kontrolu formátu nastavených atributů a uloží informace o pracovní síle do databáze.</p>
delete ()	public: <i>boolean</i>	<p>Smaže z databáze položku představující tuto pracovní sílu.</p>
getPredictorValue (<i>Predictor</i>)	public: <i>PredictorValue</i>	<p><i>param</i>: aPredictor [<i>Predictor</i> - in] Prediktor, pro nějž se má zjistit hodnota u tohoto uchazeče (zaměstnance).</p> <p>Metoda vrátí objekt <i>:PredictorValue</i> pro specifikovaný prediktor.</p>
getStatusForJob (<i>Job</i>)	public: <i>int</i>	<p><i>param</i>: aJob [<i>Job</i> - in] Pracovní pozice, k níž zjišťujeme status uchazeče.</p> <p>Vrací status uchazeče (zaměstnance) vzhledem k pracovní pozici. Metoda nejprve vytvoří instanci třídy <i>Suitability</i> pro příslušnou pozici a poté vrátí hodnotu jejího atributu <i>intStatus</i>.</p>

Workforce::Applicant

Typ: *public* **Class**

Rozšiřuje: *Workforce*.

Uchazeč, tj. potencionální budoucí zaměstnanec organizace.

Spojení

- Generalizační odkaz na třídu *Workforce*

Workforce::Applicant Atributy

Atribut	Typ	Poznámky
dtmInContactFrom	private : <i>date</i>	Datum navázání prvního kontaktu mezi uchazečem a organizací.
strCV	private : <i>string</i>	Životopis, popř. odkaz na životopis v souborovém systému (závisí na implementaci).
strCoverLetter	private : <i>string</i>	Text motivačního dopisu uchazeče.

Workforce::Applicant Metody

Metoda	Typ	Poznámky
load (<i>int</i>)	public: <i>boolean</i>	<i>param:</i> aID [<i>int</i> - in] Interní identifikátor pracovní síly. Načte informace o pracovní síle z databáze podle atributu <i>intID</i> .
save ()	public: <i>boolean</i>	Provede kontrolu formátu nastavených atributů a uloží informace o pracovní síle do databáze.
delete ()	public: <i>boolean</i>	Smaže z databáze položku představující tuto pracovní sílu.
employ (<i>string</i>)	public: <i>Employee</i>	<i>param:</i> aExternalID [<i>string</i> - in] Externí ID zaměstnance, je-li známo. Metoda vrátí objekt s obdobnými atributy jako má aktuální objekt, ovšem třídy <i>Employee</i> . Využití viz kap. 7.1.5.

Workforce::Employee

Typ: *public Class*

Rozšiřuje: *Workforce*.

Současný zaměstnanec organizace.

Spojení

- Asociační odkaz na třídu *Job*
- Generalizační odkaz na třídu *Workforce*

Workforce::Employee Atributy

Atribut	Typ	Poznámky
strBusinessEmail	private : <i>string</i>	Pracovní e-mailová adresa.
strBusinessPhone	private : <i>string</i>	Pracovní telefon.
objHiredAs	private : <i>Job</i>	Pozice do níž je zaměstnanec obsazen. Uvažujeme, že každý zaměstnanec zastává právě jednu pozici, i když v praxi se jistě může vyskytnout odlišný případ. V takovém případě bychom místo jednoduchého atributu definovali kolekci <i>arrHiredAs: array of Job</i> .
dtmHiredFrom	private : <i>int</i>	Datum, od kdy je zaměstnanec zaměstnán na pozici <i>objHiredAs</i> .
fltRealEvaluation	private : <i>float</i>	Hodnocení pracovního výkonu zaměstnance jeho nadřízeným. Tento atribut se importuje při případu užití 7.1.5 a slouží ke zhodnocení kvality modelu (kap. 7.1.6). Hodnocení se pohybuje v rozsahu [0, 1]. Poč. hodnota: <i>null</i> ;

Workforce::Employee Metody

Metoda	Typ	Poznámky
load (<i>int</i>)	public: <i>boolean</i>	<i>param: aID [int - in]</i> interní identifikátor pracovní síly. Načte informace o pracovní síle z databáze podle atributu <i>intID</i> .
save ()	public: <i>boolean</i>	Provede kontrolu formátu nastavených atributů a uloží informace o pracovní síle do databáze.

delete ()	public: <i>boolean</i>	Smaže z databáze položku představující tuto pracovní sílu.
dismiss ()	public: <i>Applicant</i>	Metoda má opačný účinek oproti metodě <i>:Applicant.employ()</i> .

Workforce::PredictorValue

Typ: public **Class**

Hodnota prediktoru u určitého uchazeče.

Spojení

- Agregáčn  odkaz na třídu *Predictor*
- Agregáčn  odkaz na třídu *Workforce*

Workforce::PredictorValue Atributy

Atribut	Typ	Poznámky
intID	private : <i>int</i>	Interní identifikátor entity v databázi.
objWorkforce	private : <i>Workforce</i>	Pracovní síla, již se hodnota týká.
objPredictor	private : <i>Predictor</i>	Prediktor, jehož se hodnota týká.
fltValue	private : <i>float</i>	Hodnota prediktoru, která samozřejmě nemusí být specifikována. V takovém případě by ale spíše objekt vůbec neměl existovat. Poč. hodnota: <i>null</i> ;
dtmMeasuring	private : <i>date</i>	Datum zjišťování hodnoty prediktoru (informativní údaj). Poč. hodnota: <i>null</i> ;

Workforce::Workforce::PredictorValue Metody

Metoda	Typ	Poznámky
load (<i>int</i>)	public: <i>void</i>	<i>param:</i> aID [<i>int</i> - in] Identifikátor entity. Načte informace o entitě z databáze.
save ()	public: <i>void</i>	Uloží informace o entitě do databáze.
delete ()	public: <i>void</i>	Vymaže entitu z databáze.

setValue (<i>float</i>)	public: void	<i>param:</i> aValue [<i>float</i> - in] Hodnota prediktoru pro příslušného uchazeče. Nastaví hodnotu prediktoru pro příslušného uchazeče. Jestliže specifikovaná hodnota neodpovídá škále (<i>objPredictor.objScale</i>), vyvolá se výjimka.
---------------------------	--------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Workforce::Suitability

Typ: public Class

Entita vyjadřuje vhodnost uchazeče (příp. zaměstnance) pro práci na určité pozici. Každý uchazeč (zaměstnanec) může být pochopitelně nezávisle posuzován pro přijetí na několik různých pozic.

Spojení

- Agregáčn  odkaz na třídu *Job*
- Agregáčn  odkaz na třídu *Workforce*

Workforce::Suitability Atributy

Atribut	Typ	Poznámky
intID	private : int	Interní identifikátor entity v systému.
objWorkforce	private : Workforce	Uchazeč, příp. zaměstnanec, kterého se vztah týká.
objJob	private : Job	Pracovní pozice, které se vztah týká.
intStatus	private : int	Stav, ve kterém se posuzovaný vztah nachází: <ul style="list-style-type: none"> • <i>SS_DEFAULT</i> - Propojení uchazeče a pozice není nijak akcentováno, ale ani omezováno. • <i>SS_PROHIBITED</i> - Propojení není žádoucí. Typicky si uchazeč nepřeje nabízenou pozici přijmout, nebo propojení brání jiné okolnosti. • <i>SS_INPROCESS</i> - uchazeč se právě nachází v procesu výběru uchazečů pro tuto pozici. • <i>SS_REJECTED</i> - uchazeč se účastnil alespoň jednoho kola výběru kandidátů na pozici a neuspěl. Byl vyhodnocen jako zcela nevhodný (krok 1.g použité strategie). Dalších kol se proto neúčastní. • <i>SS_ACCEPTED</i> ... Uchazeč se účastnil výběru a je předběžně přijat. Pokud se tedy například koná doplňující výběr, nesmí se jej už účastnit.

		<ul style="list-style-type: none"> <i>SS_ACCEPTEDELSE</i> ... Uchazeč byl přijat na jinou pozici v organizaci (Pak existuje objekt <i>:Suitability</i> pro stejného uchazeče ale jinou pozici, který má vlastnost <i>intStatus</i> nastavenou na <i>SS_ACCEPTED</i>). <p>Poč. hodnota: <i>SS_DEFAULT</i>;</p>
strNote	private : <i>string</i>	Textová poznámka ke vztahu <i>uchazeč-pozice</i> .
fltPredictedEvaluation	private : <i>float</i>	Hodnocení pracovního výkonu, jak je predikováno použitou strategií v kroku 2.c. Hodnocení se pohybuje v rozsahu [0, 1].
		Poč. hodnota: <i>null</i> ;

Workforce::Suitability Metody

Metoda	Typ	Poznámky
load (int)	public: <i>void</i>	<i>param: aID [int - in]</i> Interní identifikátor. Načte informace o entitě z databáze podle identifikátoru.
save ()	public: <i>void</i>	Uloží informace o entitě do databáze.
delete ()	public: <i>void</i>	Smaže entitu z databáze.

Komponenta Model

Model::Model

Typ: `public abstract Class`

Obecný model, používaný k hodnocení uchazečů. Metoda `getSuitableWorkforce` (resp. `getSuitableJob`) vrací kolekci s vhodnými uchazeči (resp. vhodnými pozicemi), jejíž položky jsou seřazeny podle vhodnosti.

Spojení

- Generalizační odkaz ze třídy `JiangChenModel`

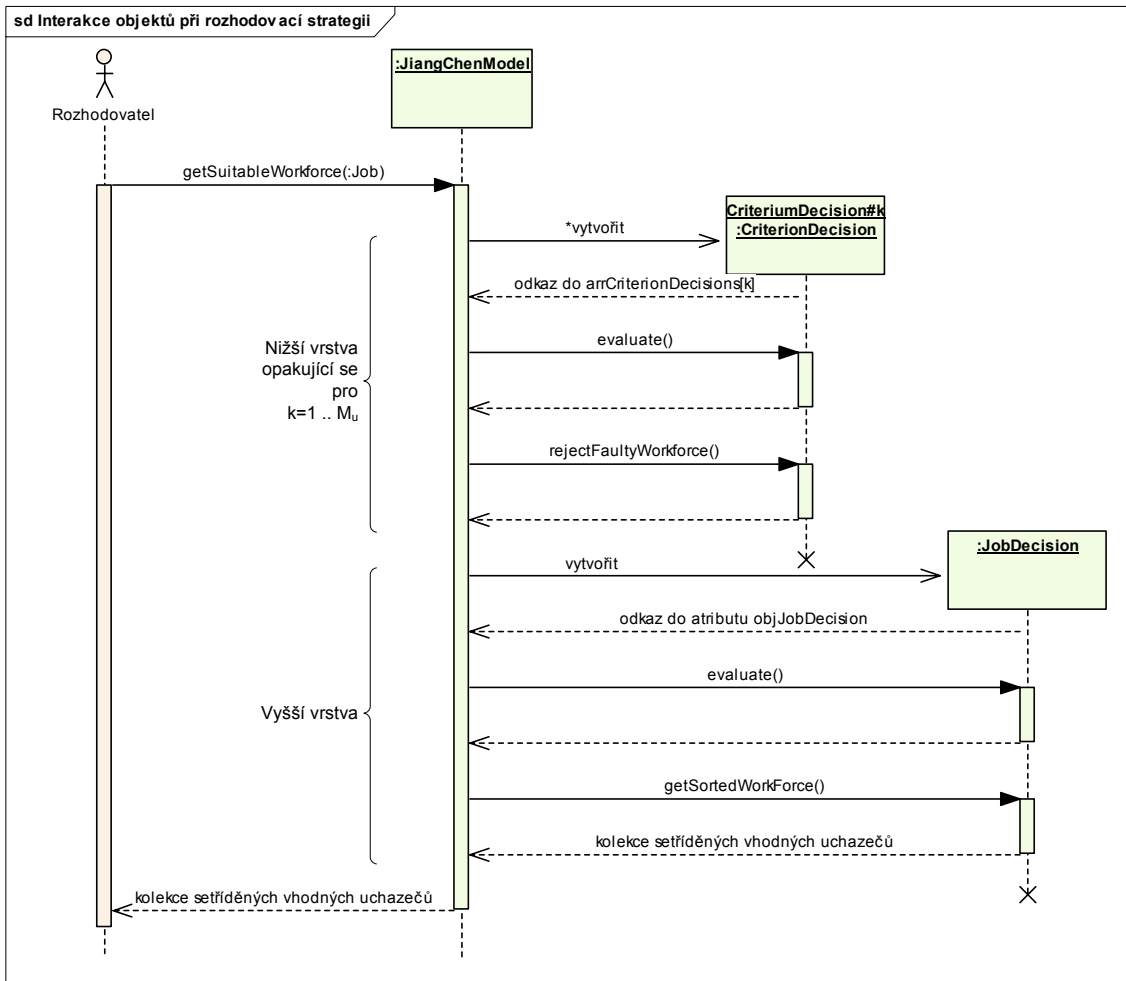
Model::Model Atributy

Atribut	Typ	Poznámky
<code>arrWorkforceInProgress</code>	<code>private : array of Workforce</code>	Kolekce uchazečů, kteří jsou posuzováni v rámci metody <code>getSuitableWorkforce</code> .

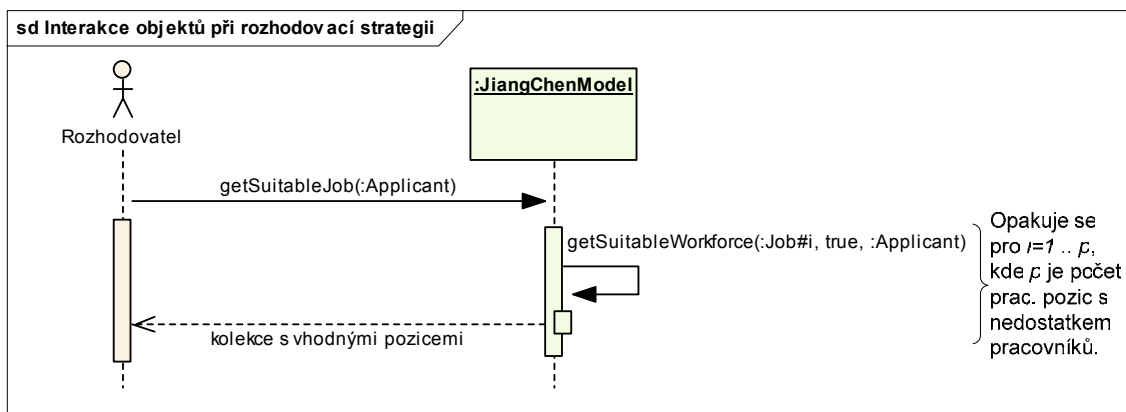
Model::Model Metody

Metoda	Typ	Poznámky
<code>getSuitableWorkforce (Job, bool, Workforce)</code>	<code>public: array of Workforce</code>	<p>param: <code>aJob [Job - in]</code> Pracovní pozice, které se bude výběr týkat.</p> <p>param: <code>aDefaultStatus [bool - in]</code> Defaultně se budou posuzovat pouze ti uchazeči, jejichž vlastnost <code>:Suitability.intStatus</code> je hodnoty <code>SS_INPROCESS</code>, tzn. že se účastní výběru od prvního kola. Pokud ale parametr nastavíme na <code>true</code>, budou se posuzovat všichni potencionální uchazeči, tj. i ti s hodnotou <code>SS_DEFAULT</code>. Nastavit parametr je tedy vhodné v prvním kole výběru všech potencionálních uchazečů.</p> <p>param: <code>aWorkforce [Workforce - in]</code> Je-li parametr specifikován, provede se hodnocení vhodnosti pozice pouze pro tohoto jednoho uchazeče</p> <p>Metoda implementuje případ užití "Hodnocení uchazečů hlásících se předem na určitou pozici", popsany v kapitole 7.1.4. Předem přiřazenými uchazeči se myslí ti, kteří mají atribut <code>:Suitability.intStatus</code> nastaven na hodnotě <code>SS_INPROCESS</code> (resp. <code>SS_DEFAULT</code> i pro nevyhraněné).</p>

		Metoda vrací množinu uchazečů (resp. zaměstnanců) vhodných pro obsazení specifikované pozice. Obrázek 32 zachycuje sekvenční diagram popisující činnost této metody.
getSuitableJob (Workforce)	public: array of Job	<p>param: aWorkforce [Workforce - in] Uchazeč (popř. zaměstnanec), kterého se má výběr příhodných pozic týkat.</p> <p>Metoda implementuje případ užití "Výběr vhodných pozic pro nevyhraněného uchazeče", popsany v kapitole 7.1.4. Uvnitř metody proběhnou tyto činnosti:</p> <ul style="list-style-type: none"> • vypočte se vhodnost všech otevřených pozic (tj. pozic kde <i>:Job.intLack>0</i>) pro uchazeče, • odfiltrují se pozice, které byly v kroku 1.g strategie vyhodnoceny jako zcela nevhodné. • Pozice se seřadí podle vhodnosti a vrátí se v kolekci. <p>Implementace je jednoduchá, stačí volat metodu <i>getSuitableWorkforce</i> pro všechny uvažované pozice s parametrem <i>aWorkforce</i> nastaveným na zkoumaného uchazeče. Potom se musí zjistit, ve kterých případech byl status uchazeče vzhledem k pozici (<i>:Suitability->intStatus</i>) přenastaven na hodnotu <i>SS_REJECTED</i>. Obrázek 33 zachycuje sekvenční diagram popisující činnost této metody.</p>



Obrázek 32 – Realizace případu užití "Hodnocení uchazečů hlásících se na určitou pozici".



Obrázek 33 – Realizace případu užití "Výběr vhodných pozic pro nevyhraněného uchazeče".

Model::JiangChenModel

Typ: *public Class*

Rozšiřuje: *Model*.

Konkrétní model implementující námi zvolenou strategii, viz kap. 6.6.3. Oproti obecnému modelu *:Model* lze po dokončení metod *getSuitableWorkforce* či *getSuitableJob* navíc metodami *getCriteriaDecisions* a *getJobDecision* získat a analyzovat obsah matic U^k a U^u , z nichž vzešlo konečné rozhodnutí v 1. a 2. rozhodovací fázi.

Spojení

- Agregáčn  odkaz ze třidy *Decision*
- Generalizační odkaz na třidu *Model*

Model::JiangChenModel Atributy

Atribut	Typ	Poznámky
arrRelevantCriteria	private : <i>array of Criterion</i>	Kolekce s kritérii relevantními pro pracovní pozici.
arrCriterionDecisions	private : <i>array of CriterionDecision</i>	Kolekce určená pro uložení rozhodovacích objektů <i>:CriterionDecision</i> během metody <i>getSuitableWorkforce</i> . Bude mít tolik prvků, kolik je kritérií relevantních k pozici.
objJobDecision	private : <i>JobDecision</i>	Atribut určený pro odložení rozhodovacího objektu <i>:JobDecision</i> během metody <i>getSuitableWorkforce</i> .

Model::JiangChenModel Metody

Metoda	Typ	Poznámky
<i>getSuitableWorkforce (Job, bool, Workforce)</i>	public: <i>array of Workforce</i>	Konkrétní implementace metody pro Jiang-Chenův model je následující: <ul style="list-style-type: none">• Určení posuzovaných uchazečů a uložení odkazů na ně do kolekce <i>arrWorkforceInProgress</i>. Jedná se vlastně o filtrování veškeré pracovní síly <i>:Workforce</i> podle návratové hodnoty metody <i>getStatusForJob(aJob)</i>.• Určení posuzovaných kritérií pomocí vlastnosti <i>aJob.getRelevantFactorWeights()[i].objFactor</i> a jejich uložení do kolekce <i>arrRelevantCriteria</i> s M_u prvky.• <u>1. fáze</u> Realizuje se výpočet důsledku prediktorů pro každé k-té z M_u kritérií. Při tom se k-krát instanciuje třida

		<p><i>CriterionDecision</i> a odkaz na ni se uloží do kolekce <i>arrCriterionDecisions</i>. Pak se volá se metoda <i>evaluate()</i> pro všechny prvky v kolekci. Nakonec se metodou <i>rejectFaultyWorkforce</i> vyřadí nevhodní uchazeči.</p> <ul style="list-style-type: none"> • <u>2. fáze</u> Instanciuje se třída <i>JobDecision</i> a odkaz na ni se uloží do atributu <i>objJobDecision</i>. Zavolá se metoda <i>evaluate()</i> vytvořeného objektu. Poté lze metodou <i>getSortedWorkforce</i> obdržet finální pořadí uchazečů posouzených podle všech relevantních kritérií.
<i>getCriteriaDecisions ()</i>	public: <i>array of CriterionDecision</i>	Vrací kolekci <i>arrCriterionDecisions</i> . Jednotlivý <i>k</i> -tý prvek kolekce lze jeho metodou <i>getUMatrix</i> požádat o matici <i>U</i> , která přesně specifikuje optimalitu <i>k</i> -tého kritéria jednotlivých posuzovaných uchazečů.
<i>getJobDecision ()</i>	public: <i>JobDecision</i>	Vrací atribut <i>objJobDecision</i> . Odkazovaný objekt lze jeho metodou <i>getUMatrix</i> požádat o matici <i>U</i> , která přesně specifikuje celkovou optimalitu posuzovaných uchazečů.

Model::Decision

Typ: *public Class*

Entita popisující vhodnost množiny faktorů pro dosažení optimálního důsledku (tedy matice *U* v použité strategii). Výpočet metodou *evaluate()* se provádí víceméně obdobně pro nižší i vyšší vrstvu. Před voláním této metody je nutné nastavit atributy *arrFactors*, *arrWorkforce* a *objConsequence*.

Spojení

- Agregační odkaz na třídu *JiangChenModel*
- Asociační odkaz na třídu *Factor*
- Asociační odkaz na rozhraní *Consequence*
- Asociační odkaz ze třídy *Workforce*
- Generalizační odkaz ze třídy *JobDecision*
- Generalizační odkaz ze třídy *CriterionDecision*

Model::Decision Atributy

Atribut	Typ	Poznámky
L	private const static : <i>int</i>	Počet vzorů rozhodnutí <i>I</i> , viz krok 1.c strategie. Poč. hodnota: 3;

Q	private const static: float	Konstanta q , viz kroky 1.f a 2.b použité strategie. Poč. hodnota: 2;
objOwner	private : <i>JiangChenModel</i>	Jiang-Chen model, který pracuje s instancí této třídy.
arrFactors	private : <i>array of Factor</i>	Faktory, tedy řádky matice C , F a váhového vektoru W^T .
arrWorkforce	private : <i>array of</i> <i>Workforce</i>	Kolekce uchazečů, jež se účastní hodnotícího procesu. Ti představují sloupce matice C , F a U .
objConsequence	private : <i>Consequence</i>	Odkaz na důsledek ovlivňovaný faktory (podle vrstvy buď kritérium nebo pracovní pozice).
arrUMatrix	private : <i>array of float,</i> <i>float</i>	Dvourozměrné pole s maticí U . Matice se počítá během metody <i>evaluate()</i> .
arrSMatrix	private : <i>array of float,</i> <i>float</i>	Dvourozměrné pole s maticí S . Matice se vyplňuje uvnitř metody <i>buildSMatrix()</i> .

Model::Decision Metody

Metoda	Typ	Poznámky
evaluate ()	public: <i>void</i>	Metoda provede výpočet vhodnosti faktorů pro dosažení optimálního důsledku. Bližší popis viz specializované třídy.
bulidSMatrix ()	public: <i>void</i>	Metoda sestaví matici S podle vztahu (18), použitého v kroku 1.d strategie, a uloží ji do dvourozměrného pole <i>arrSMatrix</i> .
getUMatrix ()	public: <i>array of</i> <i>float, float</i>	Vrací dvourozměrné pole <i>arrUMatrix</i> .

Model::CriterionDecision

Typ: *public* **Class**

Rozšiřuje: *Decision*.

Entita specializovaná pro použití v 1. fázi (nižší vrstvě) rozhodovací strategie.

Spojení

- Generalizační odkaz na třídu *Decision*

Model::CriterionDecision Metody

Metoda	Typ	Poznámky
evaluate ()	public: void	<p>Metoda provede výpočet vhodnosti prediktorů pro dosažení optimální hodnoty kritéria. Ten zahrnuje následující kroky:</p> <ul style="list-style-type: none"> • Váhový vektor W^k zjistíme voláním metody <i>aConsequence.getRelevantFactorWeights()</i>. • Sloupce matice C^k představují uchazeči <i>arrWorkforce</i>, řádky prediktory <i>arrFactors</i>. Prvky matice zjistíme tak, že pro uchazeče <i>arrWorkforce[j]</i> voláme metodu <i>getPredictorValue(arrFactors[i])</i>. • Každý prvek matice <i>C</i> transformujeme metodou <i>arrFactors[i].appFunction(C_{ij})</i>. Tak získáme matici F^k. • Matici S^k sestaví metoda <i>bulidSMatrix()</i>. • Provedeme výpočet podle vztahu (19), čímž získáme matici U^k. Tu uložíme do dvourozměrného pole <i>arrUMatrix</i>.
rejectFaultyWorkforce ()	public: void	<p>Tuto metodu lze volat až po dokončení metody <i>evaluate()</i>. Metoda podle matice U^k v <i>arrUMatrix</i> krokem 1.g vyřadí zcela nevhodné uchazeče z kolekce <i>arrWorkforce</i>. Vyřazení proběhne tak, že bude pro příslušnou pracovní pozici nastaven jejich status (<i>:Suitability.intStatus</i>) na hodnotu <i>SS_REJECTED</i>.</p>

Model::JobDecision

Typ: *public Class*

Rozšiřuje: *Decision*.

Entita specializovaná pro použití ve 2. fázi (vyšší vrstvě) rozhodovací strategie.

Spojení

- Generalizační odkaz na třídu *Decision*

Model::JobDecision Metody

Metoda	Typ	Poznámky
evaluate ()	public: void	<p>Metoda provede výpočet vhodnosti kritérií pro dosažení optimální hodnoty pracovního výkonu. To zahrnuje následující kroky:</p>

		<ul style="list-style-type: none"> • Váhový vektor W^u zjistíme voláním metody <i>aConsequence.getRelevantFactorWeights()</i>. • Matici S^u sestaví metoda <i>bulidSMatrix()</i>. • Provedeme výpočet podle vztahu (20) a obdržíme tak matici U^u. Jako vstupní data nám budou sloužit matice U^k z nižší vrstvy, získané metodou <i>objOwner.getCriterionDecisions()[i].getUMatrix()</i>. Vypočtenou matici U^u uložíme do dvourozměrného pole <i>arrUMatrix</i>.
getSortedWorkforce ()	public: <i>array of Workforce</i>	Tuto metodu lze volat až po dokončení metody <i>evaluate()</i> , kdy je v poli <i>arrUMatrix</i> přítomna matice U^u . Metoda jednoduše seřadí uchazeče z kolekce <i>arrWorkforce</i> podle defuzzifikované vhodnosti (krok 2.c strategie) a takto seříděnou kolekci vrátí.