

Informační systém pro výrobní firmu

Information system for manufacturing company

Bc. Martin Rode

Diplomová práce
2010



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin RODE**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Informační systém pro výrobní firmu**

Zásady pro vypracování:

1. Proveďte analýzu stávajících open source a komerčních informačních systémů pro výrobní firmy.
2. Na základě této analýzy vyberte systém, který nejlépe vyhovuje firmě DUDR – pilové pásy a doplňte do něj:
3. Nástroj pro plánování a rozvrhování výroby – proveďte analýzu stávajícího systému a navrhnete a realizujete nové řešení dle požadavků firmy.
4. Nástroj pro správu katalogových informací – proveďte analýzu stávajícího systému a navrhnete a realizujete nové řešení dle požadavků firmy.
5. Ve zbývajícím čase proveďte analýzu stávajícího systému pro plánování logistiky (plánování rozvozu a svozu výrobků k/od zákazníkům nejvýhodnějšími trasami, s napojením na systém řízení výroby a CRM tak, aby všechny výrobky pro danou cestu byly včas vyrobeny, s možností editaci mapy rozvozu atd.) a navrhnete jeho případné úpravy.

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. PINEDO, M. L., Scheduling : Theory, Algorithms, and Systems. 3rd edition. New York : Springer, 2002. 678 s. ISBN 978-0-387-78934-7.
2. PINEDO, M. L., Planning and Scheduling in Manufacturing and Services. 2nd edition. New York : Springer, 2009. 537 s. ISBN 978-1-4419-0909-1.
3. BRUCKER, P., Scheduling Algorithms. 5th edition. New York : Springer, 2007. 371 s. ISBN 978-3-540-69515-8.
4. BLAZEWICZ, J., et al. Scheduling Computer and Manufacturing Processes. 2nd edition. New York : Springer, 2001. 485 s. ISBN 978-3-540-41931-0.
5. BLAZEWICZ, J., et al. Handbook on Scheduling : From Theory to Applications. New York : Springer, 2007. 647 s. ISBN 978-3-540-28046-0.
6. ADAMS, J., BALAS, E., ZAWACK, D., The Shifting Bottleneck Procedure for Job Shop Scheduling. In Management Science. 3rd edition. [s.l.] : INFORMS, 1988. Focussed Issue on Heuristics. s. 391-401. Dostupný z WWW: ><http://www.jstor.org/stable/2632051><.
7. MAIER, P., Moderní metody rozvrhování výroby. Brno, 2004. 90 s. Disertační práce na Fakultě strojní Vysokého učení technického v Brně na Ústavu automatizace a informatiky. Vedoucí disertační práce RNDr. Jiří Dvořák, CSc.

Vedoucí diplomové práce:

Ing. Tomáš Dulík

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

19. února 2010

Termín odevzdání diplomové práce:

8. června 2010

Ve Zlíně dne 19. února 2010


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Tato diplomová práce se zabývá návrhem a implementací nástrojů pro správu katalogových produktů a pro plánování a rozvrhování výroby určených pro firmu DUDR - pilové pásy. Cílem je navrhnout řešení, pomocí něhož bude moci firma (prostřednictvím informačního systému) automaticky plánovat výrobu zadaných objednávek, vytvářet výrobní plán, monitorovat výrobu v reálném čase a jednoduše spravovat katalog svých produktů. Po úvodním seznámení se s teoretickým základem plánování se práce věnuje výběru vhodného řešení plánovacího algoritmu, dále provádí analýzu existujících výrobních informačních systémů a popisuje implementaci jednotlivých nástrojů. V poslední části se práce zaměřuje na analýzu nástroje pro plánování logistiky současného informačního systému a navrhuje jeho inovační řešení.

Klíčová slova: plánování, rozvrhování, Grahamova klasifikace, řízení výroby, řídicí pravidla, výrobní informační systém, PHP, MySQL.

ABSTRACT

The master's thesis is focused on design and implementation of tools for managing products as well as planning and scheduling of manufacturing for the company DUDR - pilove pasy. The goal of this work is to design a solution, which may enable the company (via its information system) to plan automatically production of their orders, create a manufacturing plan, monitor manufacturing in real-time and simply manage the database of their products. First of all, theoretical planning background is introduced, then the work aims to develop a suitable solution of planning algorithm. Further, it analyzes existing manufacturing information systems and describes an implementation of single tools. In the last part, the work deals with an analysis of the tool for planning logistics of the contemporary information system and designs its innovative solution.

Keywords: planning, scheduling, Graham's classification, manufacturing management, dispatching rules, manufacturing information system, PHP, MySQL.

Poděkování

V úvodu této práce bych rád poděkoval svému vedoucímu Ing. Tomáši Dulíkovi za vedení diplomové práce, cenné připomínky a čas strávený při konzultacích. Dále bych chtěl poděkovat Ing. Tomáši Dudrovi z firmy DUDR – pilové pásy za nabídku se tomuto tématu věnovat a za poskytnutí podkladů a odborných rad.

Motto

„Ve srovnání se schopností rozumně si rozvrhnout práci na jeden den je všechno ostatní dětskou hrou.“

Johann Wolfgang Goethe

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 7.6.2010

.....
podpis diplomanta

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	12
1 PLÁNOVÁNÍ	13
1.1 HISTORIE	13
1.2 TEORIE PLÁNOVÁNÍ.....	13
1.2.1 Výrobní systémy	14
1.2.2 Statické parametry úlohy.....	14
1.2.3 Dynamické parametry úlohy	15
1.3 GRAHAMOVA KLASIFIKACE	15
1.3.1 Prostředí strojů α	16
1.3.2 Charakteristika úloh β	18
1.3.3 Optimalizační kritéria γ	19
2 VYBRANÁ METODA ŘEŠENÍ	21
2.1 IDENTIFIKACE PROBLÉMU.....	21
2.2 METODY ŘEŠENÍ.....	21
3 ŘÍDÍCÍ PRAVIDLA	23
3.1 ZÁKLADNÍ ŘÍDÍCÍ PRAVIDLA	23
3.2 KOMPOZITNÍ ŘÍDÍCÍ PRAVIDLA	24
3.3 POPIS VYBRANÝCH ŘÍDÍCÍCH PRAVIDEL.....	24
3.3.1 EDD.....	25
3.3.2 LPT.....	25
3.3.3 SPT.....	25
3.3.4 WSPT	26
3.3.5 MSLACK	26
3.3.6 ATC.....	26
3.3.7 EECR.....	27
3.3.8 DUDR	29
II PRAKTICKÁ ČÁST	32
4 ANALÝZA SOUČASNÝCH VÝROBNÍCH INFORMAČNÍCH SYSTÉMŮ	33
4.1 POŽADAVKY	33
4.1.1 Funkční požadavky	33
4.1.2 Nefunkční požadavky.....	37
4.2 ANALÝZA VYBRANÝCH INFORMAČNÍCH SYSTÉMŮ	37
4.2.1 BlueERP	38
4.2.2 Dolibarr	39
4.2.3 FrontAccounting	40
4.2.4 NolaPro	41
4.2.5 PhreeBooks	42
4.2.6 Tine 2.0	43

4.2.7	UzERP	44
4.2.8	Vtiger.....	45
4.2.9	WebERP	46
4.3	VYHODNOCENÍ ANALÝZY	47
5	POUŽITÉ TECHNOLOGIE.....	48
5.1	DIBI.....	48
5.2	PROTOTYPE	49
6	POPIS IMPLEMENTACE.....	51
6.1	NÁSTROJ PRO SPRÁVU KATALOGOVÝCH INFORMACÍ	52
6.1.1	Třídy	52
6.1.2	Databáze	53
6.2	NÁSTROJ PRO PLÁNOVÁNÍ A ROZVRHOVÁNÍ VÝROBY	55
6.2.1	Plánovací algoritmus	56
6.2.2	Třídy aplikace.....	57
6.2.3	Třídy modelu.....	58
6.2.4	Databáze.....	60
6.3	CELKOVÝ POHLED NA DATABÁZI.....	66
7	PROGRAMOVÁ DOKUMENTACE.....	68
7.1	DOCBLOKY	68
7.2	PHPDOCUMENTOR	68
7.3	NÁHLED NA VYGENEROVANOU DOKUMENTACI	69
8	UŽIVATELSKÁ DOKUMENTACE.....	71
8.1	PŘIHLÁŠENÍ DO APLIKACE	71
8.2	HLAVNÍ OKNO APLIKACE	71
8.3	NÁSTROJ PRO SPRÁVU KATALOGOVÝCH INFORMACÍ	73
8.3.1	Správa katalogových výrobků	73
8.3.2	Správa katalogových operací	76
8.4	NÁSTROJ PRO PLÁNOVÁNÍ A ROZVRHOVÁNÍ VÝROBY	79
8.4.1	Plánování výroby	79
8.4.2	Stroje	81
8.4.3	Druhy strojů	83
8.4.4	Pracovní doby strojů	84
8.4.5	Statistiky strojů	85
8.4.6	Plán výroby	87
8.4.7	Nastavení plánování	89
8.5	TERMINÁLOVÁ ČÁST APLIKACE.....	90
9	ANALÝZA STÁVAJÍCÍHO SYSTÉMU PRO PLÁNOVÁNÍ LOGISTIKY.....	92
9.1	POPIS	92
9.2	NAVRŽENÉ ŘEŠENÍ	93
	ZÁVĚR.....	94

CONCLUSION	95
SEZNAM POUŽITÉ LITERATURY	96
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	100
SEZNAM OBRÁZKŮ	102
SEZNAM TABULEK.....	103
SEZNAM PŘÍLOH.....	104

ÚVOD

Rychlost. Veličina, která je v současné době spjata se všemi oblastmi našeho života a stále více jej ovlivňuje. Tento trend se tedy zákonitě objevuje i v oblasti průmyslové výroby. Pouze schopnost pružně a rychle reagovat na přání a potřeby svých zákazníků může firmám pomoci k zajištění konkurenceschopnosti vůči ostatním podnikům. To však vyžaduje nejen finanční, ale i časové úsilí. Přírozeným cílem každé firmy je minimalizace výrobních nákladů. Úkolem plánování výroby je na základě objednávek rozvrhnout materiál a výrobní zakázky. Řízení výroby navazuje na plánování a zajišťuje provedení plánu výroby včetně kontroly jednotlivých zakázek.

Téma této diplomové práce bylo zvoleno po několika jednáních, kdy mne firma DUDR - pilové pásy oslovila s nabídkou spolupráce na vytvoření nástrojů pro správu katalogových produktů a pro plánování a rozvrhování výroby. Tato firma působí na českém trhu od roku 1992 a zabývá se výrobou a servisem pilových pásů a souvisejících produktů. V současné době firma expanduje také na trhy sousedních zemí, a proto bylo nutné řešit stávající nárůst výroby inovacemi firemního informačního systému.

V úvodní části podávám teoretický základ zabývající se plánováním ve výrobě a popisují obecný nástroj pro klasifikaci plánovacích problémů - Grahamovu notaci. Dále pomocí analýzy firemních procesů identifikuji plánovací problém. Vhodným řešením z hlediska velikosti problému a jednoduchosti implementace je použití metody řídicích pravidel.

Ve druhé části práce provádím analýzu existujících informačních systémů s ohledem na funkční a nefunkční požadavky zadané firmou. Po vyhodnocení této analýzy docházím ke konstatování, že žádný ze zkoumaných informačních systémů nevyhovuje požadavkům firmy. Tento závěr mne tedy vede k vytvoření požadovaných nástrojů jako nových komponent do informačního systému, který si firma nechala vytvořit na zakázku a v současnosti jej používá.

Ve třetí části se zabývám použitými technologiemi. Základem je skriptovací programovací jazyk PHP a relační databáze MySQL. Dále v projektu používám javascriptový framework Prototype a databázovou vrstvu Dibi. U posledně dvou jmenovaných nechybí detailní popis a jednoduchý příklad používání.

Ve čtvrté části popisují samotnou implementaci požadovaných nástrojů. U obou nejdříve uvádím třídy a jejich funkci v systému, a poté strukturu databázových tabulek. U nástroje

pro plánování navíc uvádím vývojový diagram plánovacího algoritmu. V závěru této části je k dispozici celkový pohled na strukturu databáze ve formě E-R diagramu.

V následujících dvou kapitolách se zabývám tvorbou programové a uživatelské dokumentace. Programová dokumentace je vytvořena pomocí nástroje Phpdocumentor a je ve formě HTML stránek k nahlédnutí v příloze P I. Uživatelská dokumentace je přímo součástí této práce.

V poslední kapitole analyzuji stávající nástroj pro plánování logistiky a navrhuji inovační řešení propojující (v rámci informačního systému) plánování logistiky s plánováním výroby.

I. TEORETICKÁ ČÁST

1 PLÁNOVÁNÍ

1.1 Historie

Plánování ve výrobě se začalo brát vážně počátkem minulého století s prací Henryho Gantta a jiných průkopníků. Nicméně trvalo mnoho let od první publikace o plánování k tomu, než se objevilo v průmyslu a operačním výzkumu. Některé z prvních děl se objevovaly čtvrtletně v *Naval Research Logistics*¹ na počátku padesátých let.

Velké množství práce se na formulaci plánovacích problémů (dynamické a celočíselné programování) provedlo během šedesátých let. Když vyšel slavný článek Richarda Karpa o teorii složitosti, zaměřil se výzkum v sedmdesátých letech především na hierarchii složitosti plánovacích problémů.

Od osmdesátých let se v akademickém světě a ve světě průmyslu zkoumá několik různých směrů s rostoucí pozorností věnované stochastickým plánovacím problémům. S tím, jak osobní počítače začaly pronikat do výrobních zařízení, se vyvíjí nové generace plánovacích systémů použitelných v praxi. Trendem posledního desetiletí je snaha integrovat plánovací systémy do podnikových informačních systémů. Vývoj v této oblasti byl a je zásluhou vědeckých pracovníků oboru informatiky a průmyslových inženýrů.

1.2 Teorie plánování

Plánování je forma rozhodování, která hraje klíčovou roli ve výrobním a průmyslovém odvětví. V současném konkurenčním prostředí se stalo efektivní plánování nezbytné pro přežití společnosti na trhu. Společnosti musí dodržet termíny dodání, které byly garantovány zákazníkům. Jejich nedodržení může mít za následek značnou ztrátu kredibility. Proto je nutné plánovat činnosti takovým způsobem, aby byly efektivně využity dostupné zdroje. Účelem plánování je tedy optimální přiřazení dostupných zdrojů množině úkolů (které je třeba naplánovat) v čase. Zdroje mohou být stroje v dílně, ranveje

¹ *Naval Research Logistics* byl prvním druhem časopisu zabývající se operačním výzkumem, aplikovanou statistikou a všeobecným kvantitativním modelováním, se zvláštním zájmem o logistické aplikace. Byl založen v roce 1954. [26]

na letišti, skupina pracovníků na stavbě atd. Úlohy mohou být operace v dílně, vzlet a přistání letadla na letišti, nebo podlaží ve stavebním projektu.

Veškeré níže uvedené teoretické základy vychází (pokud není uvedeno jinak) z publikací [3]-[5] a [11]-[16]. Tyto publikace obsahují velice podobný teoretický základ, z něž jsem vycházel. Není proto u některých pojmů dále uvedeno to, ze které publikace přesně pochází (v 90% případů jsou to poznatky z několika publikací shrnuté do jednoho, uvedeného v této práci). Česká terminologie uvedených pojmů byla převzata z [16].

1.2.1 Výrobní systémy

Výrobní systémy se mohou vyznačovat různými faktory, například počet zdrojů, jejich charakteristika a konfigurace, úroveň automatizace, typ manipulace s materiálem atd. Rozdíly mezi těmito faktory daly vzniknout velkému množství plánovacích modelů. Dále se v této práci budeme zabývat pouze modelem výrobním, jelikož řešený problém je tohoto charakteru.

Ve výrobním modelu obvykle označujeme zdroj jako *stroj* a úkol, který musí být na stroji zpracován, nazýváme *úloha*. Úloha může být složena z jedné nebo více operací, které musí být zpracovány na různých strojích. V dále uvedených problémech předpokládáme počet úloh a strojů za konečný. Počet úloh značíme n a počet strojů m . Index j souvisí s úlohami, zatímco index i se stroji. Jestliže se úloha skládá z několika operací, pak pár (i, j) souvisí se zpracováním operace úlohy j na stroji i .

1.2.2 Statické parametry úlohy

Tyto parametry jsou nezávislé na rozvrhu.

Doba trvání (p_{ij})

Doba trvání p_{ij} (processing time) představuje čas zpracování úlohy j na stroji i .

Termín dostupnosti (r_j)

Termín dostupnosti r_j (release date) představuje nejbližší čas, ve kterém může začít stroj zpracovávat úlohu j .

Termín dokončení (d_j)

Termín dokončení d_j (due date) představuje závazné datum dokončení (datum předání hotové úlohy zákazníkovi). Dokončení úlohy po jejím termínu je povoleno, ale je ohodnoceno penalizací. Pokud není povoleno dokončení po termínu, nazývá se termín dokončení konečným termínem (deadline).

Váha (w_j)

Váha w_j (weight) představuje prioritu (relativní důležitost) úlohy j vzhledem k ostatním úlohám.

1.2.3 Dynamické parametry úlohy

Naopak parametry, které jsou závislé na rozvrhu, se nazývají dynamické.

Čas startu (S_{ij})

Čas startu úlohy S_{ij} (starting time) představuje čas počátku zpracování úlohy j na stroji i .

Čas konce (C_{ij})

Čas konce úlohy C_{ij} (completion time) představuje čas dokončení zpracování úlohy j na stroji i .

1.3 Grahamova klasifikace

V teorii plánování a rozvrhování je široce rozšířen pojem Grahamova klasifikace (nebo též Grahamova notace). Tato klasifikace umožňuje reprezentaci velké části plánovacích problémů. Pro popis těchto problémů se používá následující trojice položek

$$\alpha | \beta | \gamma . \quad (1)$$

Tyto položky popisují prostředí strojů, charakterizují úlohy a také požadovaná optimalizační kritéria.

1.3.1 Prostředí strojů α

Jediný stroj (1)

Případ jediného stroje je nejjednodušší ze všech možných prostředí strojů a je speciálním případem všech dalších komplikovanějších prostředí.

Identické paralelní stroje (P_m)

Mějme m identických paralelních strojů. Úloha j je dána jedinou operací a může být zpracována na kterémkoliv z m strojů, či na kterémkoliv stroji, náležícímu dané podmnožině. Jestliže úloha j nemůže být zpracována na kterémkoliv stroji, ale pouze na stroji z podmnožiny M_j , bude v položce β uvedeno M_j (viz 1.3.2).

Paralelní stroje s různou rychlostí (Q_m)

Mějme m paralelních strojů s různými rychlostmi zpracování. Rychlost stroje i se značí v_i . Čas p_{ij} , který úloha j stráví na stroji i , je roven $\frac{p_j}{v_i}$. Toto prostředí se nazývá jednotné stroje. Jestliže mají všechny stroje stejnou rychlost, např. v_i je rovno 1 pro všechny i a p_{ij} je rovno p_j , pak je prostředí identické s předchozím prostředím.

Nezávislé paralelní stroje s různou rychlostí (R_m)

Toto prostředí je zobecněním předchozího. Mějme m různých paralelních strojů. Stroj i zpracuje úlohu j rychlostí v_{ij} . Čas p_{ij} , který úloha j stráví na stroji i , je roven $\frac{p_j}{v_{ij}}$. Jestliže jsou rychlosti strojů nezávislé na úlohách, např. v_{ij} je rovno v_i pro všechny i a j , pak je prostředí identické s předchozím.

Flow shop (F_m)

Mějme m strojů zapojených sériově. Každá úloha musí být zpracována na každém z m strojů. Všechny úlohy musí následovat stejnou trasu, tzn. musí být zpracovány nejprve na stroji A, poté na stroji B atd. Po dokončení zpracování na stroji A se úloha zařadí do fronty ke stroji B.

Flexible flow shop (FF_c)

Flexible flow shop je zobecněním prostředí flow shop a paralelních strojů. Namísto m strojů zapojených sériově je zde c skupin strojů zapojených sériově. Každá skupina obsahuje několik identických paralelních strojů. Každá úloha musí být zpracovaná nejprve ve skupině A, následně ve skupině B atd. Skupina funguje jako řada paralelních strojů; v každé skupině smí být úloha j zpracovaná pouze na jednom ze strojů.

Job shop (J_m)

V prostředí job shop s m stroji má každá úloha svou vlastní předurčenou trasu, kterou musí na strojích absolvovat. Existují dva typy job shop prostředí. V prvním typu každá úloha navštíví každý stroj nejvýše jednou. V druhém z nich může úloha navštívit každý stroj více než jednou. V tomto případě bude položka β obsahovat záznam *rcrc* pro recirkulaci (viz 1.3.2).

Flexible job shop (FJ_c)

Flexible job shop je zobecnění prostředí job shop a paralelních strojů. Namísto m strojů zapojených sériově je zde c pracovních center s několika identickými paralelními stroji. Každá úloha má svou vlastní trasu, kterou musí na strojích absolvovat; v každém pracovním centru smí být úloha j zpracovaná jen na jednom ze strojů. Jestliže úloha může navštívit pracovní centrum více než jednou, bude položka β obsahovat záznam *rcrc* pro recirkulaci (viz 1.3.2).

Open shop (O_m)

Mějme m strojů. Všechny úlohy musí být zpracovány opětovně na každém z nich. Nicméně některé doby trvání mohou být rovny nule. Nejsou zde žádná omezení související s trasou úlohy. Plánovač může určit trasu pro každou úlohu a různé úlohy mohou mít odlišné trasy.

1.3.2 Charakteristika úloh β

Přerušení (prmp)

Umožňuje plánovači kdykoliv přerušit průběh zpracování úlohy a zařadit na její místo jinou (např. úlohu s vyšší prioritou). Taková situace se označuje jako přerušení (preemptions).

Precedenční podmínky (prec)

Precedenční podmínky (precedence constraints) se mohou objevit v prostředí jednoho stroje nebo paralelních strojů. Znamená to, že jedna nebo více úloh smí být dokončena předtím, než se začne zpracovávat úloha jiná. Znázorňuje se pomocí stromů.

Nastavovací doba (s_{jk})

Nastavovací doba (sequence dependent setup times) označuje čas vzniklý mezi zpracováním úloh j a k díky nutnému přenastavení stroje (např. úloha j má jiné parametry výroby než úloha k , a proto se musí upravit nastavení stroje).

Porucha (brkdown)

Porucha stroje (breakdowns) znamená, že stroj není dostupný pro plánování z důvodu jeho odstávky kvůli poruše.

Vhodnost stroje (M_j)

Vhodnost stroje (machine eligibility restrictions) se vztahuje k prostředí paralelních strojů. Znamená to, že ne všechny stroje mohou zpracovat úlohu j . Parametr M_j označuje množinu strojů způsobilou zpracovávat úlohu j .

Blokování (block)

Blokování (blocking) souvisí s prostředím flow shop. Pokud je v tomto prostředí mezi dvěma stroji malý skladovací prostor, může docházet ke blokaci (pokud druhý stroj v pořadí zpracovává operaci a první stroj před ním je volný, ale kapacita skladovacích prostor mezi těmito stroji je obsazená, nemůže první stroj začít zpracovávat další operaci).

Recirkulace (rcrc)

Recirkulace (recirculation) nastává v prostředí job shop nebo flexible job shop, pokud může úloha navštívit stroj nebo pracovní centrum více než jednou.

1.3.3 Optimalizační kritéria γ

Ve výrobních systémech je mnoho důležitých optimalizačních kritérií. V praxi je celkové optimalizační kritérium často složeno z několika základních kritérií popsanych v této podkapitole.

Maximální čas konce úloh (C_{max})

Maximální čas konce úloh (makespan) je důležitý pro konečný počet úloh v systému. Označuje se C_{max} a je definován jako čas, kdy poslední úloha opustí systém.

$$C_{max} = \max(C_1, \dots, C_n), \quad (2)$$

kde C_j je čas konce úlohy j . Minimalizace maximálního času konce úloh vede k rovnoměrnému vytížení strojů a maximalizuje výkon.

Maximální zpoždění (L_{max})

Maximální zpoždění (maximum lateness) je definováno jako

$$L_{max} = \max(L_1, \dots, L_n), \quad (3)$$

kde L_j je zpoždění úlohy j . Zpoždění vypočítáme jako

$$L_j = C_j - d_j. \quad (4)$$

Zpoždění je nezáporné, pokud je úloha zpracována později, a záporné, pokud je úloha zpracována dříve, než její termín dokončení. Maximální zpoždění znamená nedodržení termínů dodání.

Nezáporné maximální zpoždění (T_{max})

Nezáporné maximální zpoždění (maximum tardiness) je definováno jako

$$T_{max} = \max(0, C_j - d_j), \quad (5)$$

nebo také

$$T_{\max} = \max(0, L_j). \quad (6)$$

Na rozdíl od maximálního zpoždění T_{\max} není nikdy záporné.

Jednotka penalizace (U_j)

Jednotka penalizace úlohy j je definována jako

$$U_j = 1 \text{ pro } C_j > d_j, \text{ jinak } U_j = 0. \quad (7)$$

Vážený součet časů konců úloh ($\sum w_j C_j$)

Označuje celkový vážený čas konců úloh (total weighted completion time).

Celkové vážené nezáporné zpoždění ($\sum w_j T_j$)

Označuje celkové vážené nezáporné zpoždění úloh (total weighted tardiness).

Celkový vážený počet nezáporně zpožděných úloh ($\sum w_j U_j$)

Označuje vážený počet nezáporně zpožděných úloh (weighted number of tardy jobs).

2 VYBRANÁ METODA ŘEŠENÍ

2.1 Identifikace problému

Existuje několik metod, jak lze vyřešit plánovací problémy různých typů. Abychom mohli určit metodu řešení, je nejprve nutné identifikovat řešený problém, přičemž vycházíme z analýzy konkrétních výrobních procesů firmy.

Podle Grahamovy klasifikace (viz 1.3) by první parametr řešeného problému – prostředí strojů α (viz 1.3.1) – spadl do skupiny tzv. shop problémů. Konkrétně se jedná o flexible job shop. Druhý parametr Grahamovy klasifikace – charakteristika úloh β (viz 1.3.2) – zahrnuje precedenční podmínky, nastavovací doby, různé termíny dostupnosti a recirkulace. Třetí parametr – optimalizační kritérium γ (viz 1.3.3) – není konkrétně definován, respektive se jedná o více základních kritérií, která jsou následně řešena několika řídicími pravidly (viz 3.3).

Z pohledu výrobního procesu firmy to znamená, že ve firmě existuje několik pracovních center, skládajících se z jednoho nebo více identických paralelních strojů. Úlohy jsou zde reprezentovány jako katalogové výrobky. Termíny dostupnosti jednotlivých katalogových výrobků nejsou předem známy. Katalogové výrobky jsou vždy součástí objednávky, a smí být zpracovány v pořadí, v jakém jsou do objednávky zadány (precedenční podmínky).

Každý katalogový výrobek je sestaven z jedné nebo více katalogových operací. Tyto operace znázorňují trasu (musí být provedena v přesně daném pořadí), kterou musí katalogový výrobek absolvovat skrze pracovní centra. Každý katalogový výrobek může navštívit pracovní centrum vícekrát (recirkulace). Nastavovací doba zde zachycuje nutnost přenastavení strojů při změně parametrů katalogové operace.

Celý problém lze pomocí Grahamovy klasifikace zapsat jako

$$FJ_c \mid prec, rcrc, r_j, s_{jc} \mid \gamma, \quad (8)$$

kde optimalizační kritérium γ závisí na zvoleném řídicím pravidle (viz 3.3).

2.2 Metody řešení

Existuje několik metod, jak vyřešit identifikovaný problém. Hlavním problémem při výběru vhodné metody je určení výpočetní časové akceptovatelnosti vzhledem k optimalitě

rozvrhu. Znamená to rozhodnout, co je pro nás více prioritní. Zda to, jestli bude celý plánovací algoritmus trvat relativně krátkou dobu (vzhledem k velikosti problému²) za cenu jisté odchylky od optimálního rozvrhu, nebo zda bude potřeba řádově více výpočetního času pro běh plánovacího algoritmu, což nám ovšem přinese daleko lepší výsledky pro optimální rozvrh.

V případě firmy, pro kterou nástroj pro plánování výroby vytvářím, je situace následující. V současné době je zde umístěno dvacet pracovních center (většinou pouze s jedním strojem, jedno pracovní centrum je tvořeno dvěma stroji). Celkový počet katalogových výrobků (respektive úloh), které jsou permanentně v systému přítomny se pohybuje zhruba mezi 800 až 1500. Velikost problému je tedy v rozmezí 800-1500x21. Počet nových objednávek, které se během dne vloží se pohybuje kolem padesáti. Je proto nutné spouštět plánovací algoritmus několikrát denně. Požadavkem firmy bylo, aby nově přidané objednávky byly v relativně krátké době zařazeny do plánu výroby, tudíž jsme po dohodě zvolili periodu spouštění plánovací algoritmu na 30 minut. Po prostudování patřičné literatury z těchto uvedených informací nejlépe vychází použití metody řídicích pravidel.

Výhody řídicích pravidel spočívají v dosažení nízkém výpočetního času v poměru k velikosti problému, v jejich relativně snadné implementaci (záleží na složitosti pravidla) a v možnosti jednoduše a pružně měnit parametry pravidel v závislosti na aktuálních potřebách firmy. Na druhou stranu je nutné říci, že existují daleko sofistikovanější a výkonnější algoritmy, avšak ty nejsou použitelné z důvodu vysoké výpočetní časové náročnosti.

² Velikost problému - udává se počtem úloh n a počtem strojů m . Například pro 20 úloh a 10 strojů se velikost problému značí 20x10.

3 ŘÍDÍCÍ PRAVIDLA

Tato kapitola popisuje několik základních a kompozitních řídicích pravidel, která se používají při řešení plánovacích problémů v praxi a která mohou být jednoduše implementována v průmyslových plánovacích systémech.

3.1 Základní řídicí pravidla

Základní řídicí pravidla [14] jsou užitečná tehdy, pokud chceme najít uspokojivý rozvrh vzhledem k jednomu optimalizačnímu kritériu, jako například celkový čas dokončení, či maximální zpoždění. Optimalizační kritéria ve skutečném světě jsou často komplikovanější. Například reálné optimalizační kritérium může být kombinací několika základních kritérií a také funkcí času či funkcí skupiny úloh čekajících na zpracování. Řazení úloh na základě jednoho nebo dvou parametrů nemusí vést k přijatelným rozvrhům. Propracovanější řídicí pravidla, která berou v úvahu několik různých parametrů, mohou vyřešit komplikovanější optimalizační kritéria. Některé z těchto propracovanějších pravidel jsou v podstatě kombinací několika základních řídicích pravidel, a nazývají se kompozitní řídicí pravidla (viz 3.2).

Žádné popisované techniky negarantují optimální řešení (respektive poskytují optimální řešení jen na určité typy problému). Místo toho se zaměřují na nalezení uspokojivého řešení za relativně krátkou dobu. Výzkum zabývající se řídicími pravidly byl aktivní po několik desetiletí.

Tato pravidla mohou být klasifikována různými způsoby. Například je rozdíl mezi statickými a dynamickými pravidly. Statická pravidla nejsou závislá na čase, jsou to jen funkce úlohy a/nebo strojových dat. Dynamická pravidla jsou časově závislá. Příkladem dynamického pravidla je minimální rezerva (viz 3.3.5), řadící úlohy podle $\max(d_j - p_j - t, 0)$. Z toho vyplývá, že v určitém bodě v čase může mít úloha j vyšší prioritu než úloha k a v pozdějším bodě v čase mohou mít úlohy j a k stejnou prioritu.

Druhý způsob klasifikace pravidel je podle informací, se kterými pravidla pracují. Lokální pravidla užívají pouze informace vztahující se buď ke frontě, ve které úloha čeká, nebo ke stroji, kde je úloha ve frontě. Globální pravidla mohou brát v potaz informace týkající se jiných strojů, jako například doba trvání úlohy na dalším stroji v pořadí apod.

3.2 Kompozitní řídicí pravidla

Kompozitní řídicí pravidlo [14] je klasifikováno jako pravidlo, které kombinuje řadu základních řídicích pravidel. Základní pravidlo je funkcí parametrů úloh a/nebo strojů. Parametr může být nějaká vlastnost spojená buď s úlohou, nebo strojem. Může to být také konstanta či časová závislost.

Příkladem parametrů úloh jsou váha (priorita), doba trvání a termín dokončení úlohy; příkladem parametrů stroje jsou rychlost, počet úloh čekajících na zpracování a celkový počet úloh, které čekají ve frontě. Míra, jakou daný parametr ovlivňuje celkovou prioritu úlohy, je určena použitým základním pravidlem a jeho měrným parametrem. Každé základní pravidlo má vlastní měrný parametr, který je brán v určitém poměru jako příspěvek k celkovému ohodnocení úlohy.

Měrné parametry jsou buď pevně určeny návrhářem pravidla, nebo proměnnou a funkcí času, nebo určitou úlohou k naplánování. Jestliže parametry závisí na určité úloze, je nutno vypočítat některé statistické ukazatele úlohy. Tyto ukazatele, nazývané jako faktory, obvykle nezávisí na rozvrhu a lze je vypočítat snadno z daných parametrů úloh a strojů.

Funkce, které mapují statistické ukazatele na měrné parametry, musí být předem určeny návrhářem pravidla. Jsou obvykle stanoveny jen jednou, a to předtím, než je pravidlo zpřístupněno pro použití. Vždy, když je kompozitní řídicí pravidlo použito pro vytvoření rozvrhu, jsou vypočteny nezbytné statistiky. Hodnoty měrných parametrů jsou nastaveny předem určenými funkcemi, založenými na hodnotách statistických ukazatelů. Poté, co jsou určeny měrné parametry, lze pravidlo aplikovat na sadu úloh.

3.3 Popis vybraných řídicích pravidel

V této kapitole popisují osm řídicích pravidel. Ty představují pouhý zlomek všech známých pravidel. Snažil jsem se vybrat nejvíce používané a poukázat na to, jakým způsobem pracují. Zároveň jsem pravidla implementoval v praktické části. Prvních pět z nich se řadí do skupiny základních pravidel, poslední tři jsou pravidla kompozitní. Potřebám firmy, pro kterou je plánovací systém vyvíjen, žádné z dostupných pravidel nevyhovuje, proto je nutné navrhnout pravidlo vlastní (viz 3.3.8).

3.3.1 EDD

Pravidlo EDD [13]-[16], nejdřívější termín dokončení (earliest due date), plánuje úlohy v rostoucím pořadí jejich termínu dokončení d_j . Kdykoliv je stroj volný, vybereme úlohu s nejbližším termínem dokončení jako další ke zpracování.

Toto pravidlo se zaměřuje na minimalizaci maximálního zpoždění mezi úlohami čekajícími na zpracování. V prostředí jednoho stroje s n dostupnými úlohami v čase t poskytuje optimální řešení pro minimalizaci maximálního zpoždění. Nicméně neposkytuje optimální řešení pro další problémy související s termínem dokončení.

Použití EDD pravidla na nesprávný typ problému může mít za následek řešení, které je daleko od řešení optimálního. EDD pravidlo se také používá jako součást kompozitního řídicího pravidla ATC (viz 3.3.6).

3.3.2 LPT

Pravidlo LPT [13]-[16], nejdelší doba trvání (longest processing time), plánuje úlohy v klesajícím pořadí doby trvání p_j . Kdykoliv je stroj volný, vybereme úlohu s nejdelší dobou trvání jako další ke zpracování.

LPT pravidlo se používá v průmyslových plánovacích systémech v prostředích s paralelními stroji k tomu, aby rozdělovalo rovnoměrně pracovní zátěž na stroje. Pravidlo vychází z úvahy, že je výhodné ponechat úlohy s krátkými dobami trvání na později, jelikož tyto úlohy jsou užitečné na konci plánování pro vyrovnávání pracovní zátěže.

Po rozdělení úloh na stroje mohou být ostatní úlohy na dalších strojích dále přeplánovány. Přeplánování úloh nepůsobí na rovnováhu rozdělní zátěže. Může být provedeno například kvůli minimalizaci sekundárního optimalizačního kritéria.

3.3.3 SPT

Pravidlo SPT [13]-[16], nejkratší doba trvání (shortest processing time), plánuje úlohy v rostoucím pořadí doby trvání p_j . Kdykoliv je stroj volný, vybereme úlohu s nejkratší dobou trvání jako další ke zpracování.

V prostředí s jedním strojem s n úlohami, jejichž termín dostupnosti r_j je pro všechny úlohy roven nule, je toto pravidlo optimální pro následující optimalizační kritéria:

- minimalizace času průměrného průchodu úlohy systémem,

- minimalizace průměrného počtu úloh v systému,
- minimalizace průměrné čekací doby úlohy od příchodu úlohy do systému, do doby začátku zpracování,
- minimalizace maximální čekací doby a průměrného zpoždění úlohy.

V prostředí paralelních strojů je SPT pravidlo optimální pro minimalizaci celkového času dokončení úloh na strojích.

3.3.4 WSPT

Pravidlo WSPT [13]-[16], vážená nejkratší doba trvání (weighted shortest processing time), plánuje úlohy v klesajícím pořadí $\frac{w_j}{p_j}$. Kdykoliv je stroj volný, vybereme úlohu s nejvyšším podílem priority w_j a doby trvání p_j jako další ke zpracování.

Pravidlo je zaměřeno na minimalizaci váženého součtu doby trvání $\sum w_j C_j$. V prostředí jednoho stroje s n úlohami dostupnými v čase t poskytuje pravidlo WSPT pro minimalizaci $\sum w_j C_j$ optimální řešení. V případě, kdy mají všechny úlohy své priority w_j rovné, se z pravidla WSPT stává pravidlo SPT (viz 3.3.3).

3.3.5 MSLACK

Pravidlo MSLACK [13]-[16], minimální rezerva (minimal slack), udává míru "naléhavosti" úlohy. Je obměnou pravidla EDD (viz 3.3.1). Kdykoliv je stroj volný, vypočítáme zbývající rezervu každé úlohy v aktuálním čase t , definovanou jako $\max(d_j - p_j - t, 0)$. Úlohu s minimální rezervou zvolíme jako další ke zpracování.

Pravidlo se zaměřuje na minimalizaci termínu dokončení. V prostředí jednoho stroje s n úlohami dostupnými v čase t a termínem dostupnosti r_j rovným nule maximalizuje toto pravidlo minimální zpoždění.

3.3.6 ATC

Apparent Tardiness Cost [13]-[16] je kompozitní řídicí pravidlo kombinující pravidlo WSPT a MS. Pomocí ATC pravidla jsou úlohy naplánovány jedna po druhé; to znamená, že vždy, když je stroj volný, je vypočítán index ohodnocení pro všechny zbývající úlohy. Úloha s nejvyšším indexem ohodnocení je poté naplánována na stroj. Tento index

ohodnocení je funkcí času t , ve kterém se stroj uvolnil a dále pak funkcí parametrů p_j , w_j a d_j zbývajících úloh. Index je definován jako

$$I_j(t) = \frac{w_j}{p_j} \exp\left(-\frac{\max(d_j - p_j - t, 0)}{K\bar{p}}\right), \quad (9)$$

kde K je parametr, který se určí empiricky, \bar{p} je průměrný čas zpracování zbývajících úloh a $\max(d_j - p_j - t, 0)$ je časová rezerva úlohy j . Jestliže je K zvoleno velmi velké, ATC se redukuje na WSPT pravidlo. Naopak, jestliže je K zvoleno blízké nule, pravidlo se redukuje na MS (pokud nedochází ke zpoždění mezi úlohami), nebo na WSPT (pokud existují zpoždění mezi úlohami).

Abychom získali uspokojivé rozvrhy, musí být hodnota parametru K zvolena vzhledem ke konkrétnímu plánovacímu problému. Výpočet parametru K lze provést vytvořením a posouzením detailní analýzy specifické instance plánovacího problému, pro který se bude pravidlo používat. Existuje několik způsobů, jak charakterizovat instanci problému. Jeden z nich spočívá v zavedení faktoru těsnosti termínu dokončení τ , který je definován jako

$$\tau = 1 - \frac{\sum d_j}{n \cdot C_{\max}}, \quad (10)$$

kde $\frac{\sum d_j}{n}$ je průměrný termín dokončení úloh. Hodnoty τ blízké jedné signalizují, že termíny dokončení jsou v těsném rozložení, a hodnoty blízké nule signalizují, že termíny dokončení jsou rozloženy volně. Faktor termínu dokončení R je definován jako

$$R = \frac{d_{\max} - d_{\min}}{C_{\max}}. \quad (11)$$

Velká hodnota R signalizuje široký, malá hodnota úzký rozsah termínu dokončení. Několik výzkumů prokázalo a stanovilo vztahy mezi parametrem K , τ , R a prostředím strojů.

3.3.7 EECR

Před nějakou dobou navrhli Chiang a Fu [6] kompozitní řídicí pravidlo, které nazvali ECR (Enhanced Critical Ratio), jehož principem bylo použití „skupinových informací“ k výběru úloh. Hodnota priority úlohy zde závisí na vlivu všech konkurenční úloh,

v důsledku zpracování úlohy jako první v pořadí. Tímto způsobem pravidlo spojovalo jednoduchá řídicí pravidla SPT (viz 3.3.3), EDD (viz 3.3.1) a LPT (viz 3.3.2). Bylo prokázáno, že pravidlo ECR má dobré výsledky při minimalizaci zpoždění.

Nicméně toto pravidlo lze aplikovat pouze na problémy s úlohami bez zpoždění. Na úlohy se zpožděním vyvinuli Chiang a Fu nové pravidlo EECR (Extended enhanced critical ratio) [7]. To rozšiřuje původní ECR, a činí jej aplikovatelné na všechny typy úloh zavedením rozšířeného termínu dokončení.

Na řízení se podílí dva parametry – e a d_e ,

- parametr e představuje, kolikrát byl úloze prodloužen její termín dokončení,
- parametr d_e je termín dokončení, stanovený jeho e -tým prodloužením.

Počáteční hodnota e_j je pro každou úlohu j nulová a počáteční hodnota d_{j0} je její skutečný zadaný termín dokončení (d_j). Tyto dva parametry jsou použity pouze uvnitř pravidla. Prodloužené termíny dokončení neovlivní skutečný termín dokončení.

Postup

Krok 0. Inicializace

Stroj m musí určit, která úloha ve frontě je ta následující. Označíme všechny úlohy ve frontě jako neohodnocené. Množinu všech úloh ve frontě označíme Q .

Krok 1. Prodloužení termínu dokončení

Pro každou úlohu $j \in Q$.

Pokud

$$t + r_j > d_j^{e_j}, \quad (12)$$

pak

$$e_j = e_j + 1, \quad d_j^{e_j} = d_j^{e_j - 1} + k \cdot r_j. \quad (13)$$

Krok 2. Ohodnocení úlohy

Vezmeme neohodnocenou úlohu i za předpokladu, že je to další úloha ke zpracování. Vypočítáme ohodnocení vybrané úlohy pro všechny úlohy v Q :

$$V_i = \sum_{j \in Q} (v_j)^2 \cdot (e_j + 1)^k, \quad (14)$$

kde pro $j = i$ je

$$v_j = \frac{r_j - p_j}{d_j^{e_j} - t - p_i}, \quad (15)$$

pro $j \neq i$ je

$$v_j = \min\left(\frac{r_j}{d_j^{e_j} - t - p_i}, 1\right). \quad (16)$$

Krok 3.

Označíme úlohu i jako ohodnocenou. Pokud existují další neohodnocené úlohy, opakujeme celý postup od kroku 2.

Krok 4.

Vybereme úlohy ve frontě tak, že nejvyšší prioritou označíme úkol i s nejmenší hodnotou ohodnocení V_i .

3.3.8 DUDR

Pro potřeby této práce bylo nutné vyvinout nové kompozitní řídicí pravidlo, které by plnilo požadavky zadavatele. Pravidlo jsem pojmenoval podle názvu firmy, tj. DUDR. Po konzultacích se zástupci firmy vzniklo následující zadání.

Máme čtyři parametry (tři souvisejí s úlohami a jeden se stroji), dle kterých bude vybrána úloha ke zpracování. Popis těchto parametrů je uveden v následujícím odstavci. Jsou seřazeny sestupně podle míry důležitosti vůči ostatním parametrům. Každý parametr má vlastní váhu, podle které se vypočítá výsledný index ohodnocení úlohy.

Váha úlohy

Váha (viz 1.2.2) představuje relativní důležitost úlohy vzhledem k ostatním úlohám.

Termín dokončení úlohy

Termín dokončení (viz 1.2.2) představuje závazné datum dokončení (datum předání hotové úlohy zákazníkovi). Dokončení úlohy po jejím termínu je povoleno.

Číslo objednávky úlohy

Každá úloha je součástí konkrétní objednávky a každá objednávka smí obsahovat více úloh. Vzhledem k tomu, že je u strojů omezena skladovací kapacita, je nutné zpracovávat úlohy jedné objednávky pokud možno hromadně tak, aby nevznikaly situace, kdy bude najednou rozpracováno několik desítek objednávek a nebude dostupná skladovací kapacita u strojů.

Priorita stroje

Každý stroj má určen prioritní parametr výroby (např. profil, šířka stelitu, typ zubu apod.), jehož změna hodnoty (např. změna šířky stelitu z 20mm na 50mm) má za následek nutné přenastavení stroje, které zabere určitý čas. V terminologii charakteristiky úlohy je to nastavovací doba (viz 1.3.2), avšak v tomto případě jde pouze o přenastavení u vybraného prioritního parametru.

Postup

Krok 1. Nastavení vah parametrů

Nejprve jsou nastaveny váhy jednotlivých parametrů ke každé dostupné úloze.

- Váha úlohy

Je určena přímo parametrem úlohy v rozmezí 1-5 (1 znamená nejmenší důležitost úlohy a 5 největší).

- Termín dokončení úlohy

Je určen přímo parametrem úlohy. Váha je nastavena následovně.

- a) Pokud je zpoždění úlohy větší než aktuální čas plánování o více než 5 dnů je váha rovna 1.
- b) Pokud je zpoždění úlohy větší než aktuální čas plánování o méně než 5 dnů nebo je termín dokončení menší než aktuální čas o 1 den, váha je rovna 5.

- c) Pokud je termín dokončení v intervalu 1-2 dny od aktuálního času plánování, je váha rovna 4.
- d) Pokud je termín dokončení v intervalu 2-3 dny od aktuálního času plánování, je váha rovna 3.
- e) Pokud je termín dokončení v intervalu 3-4 dny od aktuálního času plánování, je váha rovna 2.
- f) Pokud je termín dokončení v intervalu 4-5 dnů od aktuálního času plánování, je váha rovna 1.

- **Číslo objednávky úlohy**

U úlohy, jejíž číslo objednávky je jiné než číslo poslední objednávky zpracované strojem, je váha rovna 5. Jinak je váha rovna 0.

- **Priorita stroje**

U úlohy, jejíž hodnota prioritního parametru výroby (např. šířka stelitu) je jiná než hodnota posledního zpracovaného prioritního parametru strojem, je váha rovna 5. Jinak je váha rovna 0.

Krok 2. Výpočet indexu ohodnocení úlohy

Pomocí nastavených vah je vypočítán výsledný index ohodnocení úlohy s ohledem na relativní důležitost každého parametru vůči ostatním.

Krok 3. Výběr operce ke zpracování

Úloha s nejmenším indexem ohodnocení je vybrána ke zpracování na stroji.

Veškeré uvedené hodnoty vah parametrů jsou výsledkem experimentálního testování několika desítek úloh.

II. PRAKTICKÁ ČÁST

4 ANALÝZA SOUČASNÝCH VÝROBNÍCH INFORMAČNÍCH SYSTÉMŮ

Před samotným započítím výběru vhodného řešení je nejprve nutné analyzovat výrobní procesy firmy a specifikovat funkční a nefunkční požadavky na informační systém. Z této úvodní analýzy zvolíme nejlépe vyhovující řešení.

V podstatě máme na výběr ze dvou možných variant. První z nich je nalezení již hotového softwarového řešení, které bude vyhovovat zadaným požadavkům (popřípadě nalezení řešení, které se bude pokud možno co nejvíce blížit požadavkům s tím, že budeme muset aplikaci patřičně upravit). Druhou variantou je cesta vývoje nových komponent do současného informačního systému firmy.

4.1 Požadavky

V této podkapitole uvádíme seznam funkčních a nefunkčních požadavků firmy na informační systém. Požadavky jsou dále rozděleny do podkategorií. Tyto podkategorie vychází z rozdělení modulů současného informačního systému používaného ve firmě a z nových požadavků na mnou vyvíjené nástroje.

4.1.1 Funkční požadavky

Správa zákazníků

- vedení seznamu zákazníků
- nastavení kontaktních a fakturačních údajů
- nastavení možnosti slevy, způsobu dopravy a platby
- nastavení termínů splatnosti faktur
- lokalizace zákazníka pomocí GPS souřadnic

Správa objednávek

- vedení seznamu objednávek
- nastavení výrobních parametrů u každé objednávky
- automatický výpočet ceny objednávky dle nastavených výrobních parametrů a vybraných katalogových výrobků

- monitoring aktuálního stavu hotovosti jednotlivých položek objednávky
- možnost vyřazení některých katalogových výrobků nebo operací v průběhu zpracování
- nastavení termínů dokončení
- nastavení způsobu dodání a způsobu platby konkrétní objednávky

Správa zaměstnanců

- vedení seznamu zaměstnanců a údajů o nich
- nastavení zaměstnaneckého poměru
- nastavení hodinové mzdy
- nastavení přístupových údajů do terminálové části
- nastavení způsobilosti zaměstnance pracovat jen na určitých typech strojů
- zobrazení provedených činností zaměstnance
- mzdová agenda (počítání odpracovaných hodin, výpočet úkolové mzdy, možnost bonusových příplatků a srážek z platu, započítání dovolených, přehledy za období)
- přehled docházky do zaměstnání

Fakturace

- vedení seznamu faktur na základě zadaných objednávek
- možnost fakturace jen části objednávky (tzn. rozložení objednávky do více faktur) nebo fakturace více objednávek najednou (tzn. jedna faktura může obsahovat několik objednávek, nebo jen některé části různých objednávek)
- statistiky uhrazených a neuhrazených faktur
- nastavení měny (CZ, EUR) pro jednotlivé faktury/zákazníky
- možnost exportování fakturačních dat do aplikace Money S3
- zobrazení/tisk faktur a příjmových dokladů

Skladová evidence

- vedení seznamu skladových karet
- možnost editace skladových karet
- přidávání skladových zásob na sklad
- nastavení minimálního množství skladové zásoby

- dělení skladových karet podle kategorií výrobků
- procentuální zobrazení množství zásob (nadzásobení a podzásobení) vůči nastavenému minimu

Správa úkolů

- vedení seznamu úkolů, které mají na starost jednotliví zaměstnanci (úkoly se vztahují k zákazníkům)
- správa štítků a možnost přiřadit každému úkolu či zákazníkovi jeden nebo více štítků
- filtrování úkolů podle zákazníka, zaměstnance nebo štítků
- zobrazení detailu zákazníka a k němu přiřazených úkolů, možnost zobrazit historii přiřazených úkolů
- zobrazení všech zadaných úkolů na mapě s rozlišením týdenních období

Logistika

- možnost vytvářet logistické plány pro rozvoz objednávek zákazníkům
- výběr zaměstnance a vozidla pro určenou trasu
- automatické vytvoření trasy rozvozu
- automatický výpočet časů návštěvy zákazníka dle pořadí rozvozu
- automatický výpočet začátku cesty podle zadaného času první návštěvy zákazníka, dále pak konec cesty a celkovou délku cesty, pro účely mzdové agendy
- automatický import vytvořeného plánu do GPS stanice pomocí integrovaného rozhraní v systému

Terminálová část

- samostatná část aplikace, napojená na informační systém, běžící na terminálech (průmyslových panelových počítačích) ve výrobní dílně
- uzpůsobená k ovládání pouze pomocí dotykového displeje
- umožňuje přihlášení zaměstnance
- zobrazuje pouze stroje, které může zaměstnanec obsluhovat (dle nastavené způsobilosti zaměstnance)
- u jednotlivých strojů zobrazuje dostupné operace ke zpracování, jejich parametry, termín dodání a ostatní důležité poznámky pro výrobu

- pomocí přihlášení a odhlášení z terminálové části se počítá pracovní doba
- možnost zadat odchod a příchod na/z přestávky
- možnost odepisovat zpracované katalogové operace, v případě špatných kusů možnost vyřadit tyto operace ze zpracovávání
- po odepsání katalogové operace tuto skutečnost promítnout i do skladových zásob
- taktéž se zde zaznamenávají veškeré činnosti zaměstnance, a z takto získaných dat se dále počítá úkolová mzda, počty odpracovaných hodin atd.

Správa katalogových informací

- vedení seznamu katalogových výrobků a katalogových operací (každý katalogový výrobek se skládá z jedné nebo více katalogových operací)
- vedení základních informací o katalogovém výrobku a katalogové operaci
- jednoduchá správa přiřazených katalogových operací u katalogového výrobku (u každé operace možnost přiřazení skladové karty a zadání odepisovaného množství ze skladu podle proměnných výrobních parametrů)
- nastavení tzv. mzdové funkce, tj. mzdy příslušící zaměstnanci při provedení katalogové operace, podle proměnných výrobních parametrů
- nastavení druhu stroje, který je způsobilý zpracovávat katalogovou operaci
- nastavení času zpracování, který je zapotřebí pro zpracování katalogové operace na stroji, podle proměnných výrobních parametrů
- nastavení ceny katalogového výrobků a katalogové operace podle proměnných výrobních parametrů

Plánování a rozvrhování výroby

- rozvrhování nezpracovaných katalogových operací na příslušné stroje tak, aby se pokud možno co nejlépe dodržely požadavky firmy (priorita, čas přenastavení atd., viz 3.3.8)
- zajistit automatické spouštění plánování v určitém intervalu kvůli nově přidávaným objednávkám či jejich změnám, taktéž možnost ručního spuštění plánování
- zobrazení plánu výroby podle rozvržených operací, plán bude obsahovat jednotlivé stroje, jejich aktuální vytížení a časovou osu s úlohami ke každému stroji

- v plánu výroby možnost vybrat a zvýraznit jednotlivé katalogové operace objednávky podle čísla objednávky nebo názvu zákazníka, zobrazení detailu této vybrané objednávky
- v plánu výroby graficky odlišit veškeré katalogové operace, které jsou zpracovány po termínu dokončení objednávky
- v plánu výroby možnost změny měřítka a obnovovací doby plánu
- monitoring výroby - interakce plánu výroby s terminálovou částí systému (viz 8.5)

4.1.2 Nefunkční požadavky

- typ aplikace - webová aplikace
- programovací jazyk PHP, databáze MySQL
- česká lokalizace
- jednoduchost ovládání (na podobné úrovni jako nabízí současný informační systém)
- cena za pořízení informačního systému a školení uživatelů nesmí přesáhnout 20 000 Kč včetně nově vyvíjených nástrojů (správa katalogových informací, plánování a rozvrhování výroby)

4.2 Analýza vybraných informačních systémů

Vzhledem k již vydaným nákladům za vývoj současného informačního systému musí být celá aplikace i s případným školením uživatelů pořízena do celkové sumy 20 000 Kč. Toto je dohodnutá částka, za kterou firmě implementují nástroj pro správu katalogových informací a nástroj pro plánování a rozvrhování výroby do současného informačního systému. Z toho vyplývá fakt, že se v analýze zabývám pouze open-source [28] aplikacemi vyhovujícími tomuto kritériu.

Z několika desítek dostupných open-source informačních systémů byly vyřazeny ty, které nesplňují základní nefunkční požadavek firmy na programovací jazyk, tj. aby byl systém naprogramován v jazyce PHP. Zde je nutno uvést, že tímto požadavkem se vyloučilo převážné množství velice kvalitních informačních systémů, jejichž podstatná část je naprogramována v jazyce Java.

Níže tedy uvádím vybrané informační systémy splňující požadavek na programovací jazyk PHP. U každého z nich uvádím krátký popis, hlavní rysy a zdůvodnění, proč je vhodný, či nevhodný, v souvislosti s dalšími specifikovanými funkčními a nefunkčními požadavky pro řešení zadaného problému.

4.2.1 BlueERP

BlueERP [20] je open-source webová ERP³ aplikace určená pro malé a střední podniky. Je naprogramována v jazyce PHP a používá MySQL databázi. Hlavním cílem tohoto projektu je poskytovat všechny hlavní rysy ERP aplikace, zajistit flexibilní a uživatelsky příjemné rozhraní, ale taktéž umožňovat jednoduchý a otevřený vývoj (k tomu poskytuje adekvátní dokumentaci).

Hlavní rysy aplikace

- Správa kontaktů, dodavatelů a zákazníků.
- Správa výrobků a služeb.
- Správa objednávek.
- Skladová evidence.
- Nastavení uživatelských práv pro různé skupiny uživatelů.
- Fakturace.
- Komplexní reporty o přijatých/zaplacených/nezaplacených pohledávkách vůči dodavatelům/zákazníkům.
- Podvojný zápis účetnictví.

Tento systém je velice přehledný a jednoduchý na ovládání, bohužel však nesplňuje klíčové požadavky. Aplikace nepodporuje české prostředí, dále zde chybí funkční

³ ERP (Enterprise resource planning) – aplikace, které představují softwarová řešení užívaná k řízení podnikových dat a pomáhající plánovat celý logistický řetězec od nákupu přes sklady po výdej materiálu, řízení obchodních zakázek od jejich přijetí až po expedici, včetně plánování vlastní výroby a s tím spojené finanční a nákladové účetnictví i řízení lidských zdrojů. [2]

požadavky (části logistika, správa zaměstnanců a mzdová agenda, terminálová část). Ostatní funkční požadavky aplikace splňuje taktéž pouze jen částečně. To jsou důvody, proč tuto aplikaci nelze použít pro účely firmy.

4.2.2 Dolibarr

Dolibarr [22] je open-source projekt a sestává se různých rysů charakteristických pro ERP a CRM⁴ systémy. Je naprogramován v jazyce PHP a používá MySQL databázi. Tato aplikace je vhodná pro menší až střední podniky a nabízí mnoho užitečných modulů. Výhodou je velice jednoduchá instalace (instalátor obsahuje taktéž Apache, PHP, MySQL).

Hlavní rysy aplikace

- Fakturace.
- Správa kontaktů, dodavatelů a zákazníků.
- Správa výrobků a služeb.
- Správa objednávek.
- Správa bankovních účtů a bankovních transakcí.
- Správa štítků.
- Správa dokumentů.
- Správa projektů a úkolů s přehledným kalendářem.
- Skladová evidence s možností vytvořit více skladových míst.
- Hromadné odesílání mailů uživatelům/skupinám.
- Propracované nastavení uživatelských práv pro různé skupiny uživatelů k různým částem systému.
- Široká škála možností pro export dat.

⁴ CRM (Customer relationship management) - informační systémy zajišťující správu dat souvisejících s péčí o zákazníky. [8]

Tento systém je díky mnoha modulům složitější než předchozí, výhodou však je možnost použít pouze ty moduly, které potřebujeme. Systém je z hlediska ovládání spíše pomalejší a některé akce jsou vcelku nepředvídatelné. Aplikace taktéž nepodporuje české prostředí, dále zde chybí funkční požadavky (části logistika, mzdová agenda, terminálová část). Ostatní funkční požadavky aplikace splňuje v některých případech více, než bychom potřebovali, ovšem v některých případech naopak pouze jen částečně. To jsou důvody, proč tuto aplikaci nelze použít pro účely firmy.

4.2.3 FrontAccounting

FrontAccounting [25] je jednoduchý, ale silný open-source ERP systém vhodný pro malé podniky. Je naprogramován v jazyce PHP a používá MySQL databázi.

Hlavní rysy aplikace

- Fakturace.
- Správa kontaktů, dodavatelů a zákazníků.
- Správa výrobků a služeb.
- Správa objednávek.
- Správa bankovních účtů a bankovních transakcí, možnost použití více měn (uchování historie směnných kurzů).
- Skladová evidence s možností vytvořit více skladových míst.
- Propracované nastavení uživatelských práv pro různé skupiny uživatelů k různým částem systému.
- Široká škála možností pro export dat.
- Správa výroby, vytváření pracovních příkazů, sestavení kusovníků.

Tento systém je z hlediska ovládání a přehlednosti velmi dobře zpracován. Celkové uspořádání uživatelského rozhraní je intuitivní a lze říci, že ze všech tří zatím uvedených systémů je tento z hlediska ovládání nejvíce vyhovující. Aplikace podporuje české prostředí. Ovšem chybí zde funkční požadavky (části logistika, správa úkolů, správa zaměstnanců, mzdová agenda a terminálová část). Ostatní funkční požadavky aplikace

splňuje v některých případech taktéž pouze jen částečně. To jsou důvody, proč tuto aplikaci nelze použít pro účely firmy.

4.2.4 NolaPro

NolaPro [24] je robustní open-source systém určený pro malé a střední podniky. Obsahuje celou škálu užitečných modulů od základů jako jsou fakturace či správa objednávek až po specifické moduly jako terminálová část pro monitoring docházky zaměstnanců se statistikami či webový B2B⁵ klientský systém. Je naprogramován v jazyce PHP a používá MySQL databázi.

Hlavní rysy aplikace

- Fakturace.
- Správa kontaktů, dodavatelů a zákazníků.
- Správa výrobků a služeb.
- Správa objednávek.
- Správa bankovních účtů a bankovních transakcí v různých měnách.
- Správa jednotlivých formulářových polí na všech dostupných formulářích.
- Skladová evidence s možností vytvořit více skladových míst.
- Propracovaná správa zaměstnanců a mzdová agenda.
- Propracované nastavení uživatelských práv pro různé skupiny uživatelů k různým částem systému.
- Terminálová část pro hlášení příchodu a odchodu do práce (taktéž pauzy, přestávky na oběd), výpočet odpracovaných hodin.
- Možnost české lokalizace pomocí jednoduchého nástroje pro přepsání stávajících textů.

⁵ B2B (Business-to-business) - obecně lze B2B definovat jako obchodní vztah mezi dodavatelem na jedné straně a odběratelem, který dodané produkty dále využije ve svém podnikání, na straně druhé. Důležité je, že na straně odběratele nefiguruje koncový spotřebitel. [8]

- B2B klientský systém.
- Hromadné odesílání mailů uživatelům/skupinám.
- Široká škála možností pro tvorbu reportů.
- Široká škála možností pro export dat.
- Podpora propojení s internetovým obchodem OSCommerce.

Ze všech dosud představených aplikací se jeví nejlépe zatím NolaPro a to jak z hlediska funkčnosti, přehlednosti, tak i z hlediska ovládání. Aplikace sice nepodporuje české prostředí, ale pomocí integrovaného nástroje pro překlad systémových textů lze systém upravit pro naše potřeby (za cenu času stráveného překladem, jelikož je systém velice rozsáhlý). Co se týče funkčních požadavků, splňuje systém téměř všechny podstatné základní požadavky. Bohužel zde opět chybí část pro logistiku, která je klíčovou pro chod firmy či terminálová část pro řízení výroby. To jsou důvody proč tuto aplikaci taktéž nelze použít pro účely firmy.

4.2.5 PhreeBooks

PhreeBooks [30] je open-source ERP aplikace určená pro malé podniky. Cílem tohoto projektu je poskytnout vícejazyčný internetový podnikový manažerský nástroj za nízkou pořizovací cenou. Je naprogramován v PHP a používá MySQL databázi. PhreeBooks je platformě nezávislý, taktéž je nezávislý na webovém prohlížeči a díky integrovanému nástroji snadno přeložitelný do mnoha jazyků.

Hlavní rysy aplikace

- Fakturace.
- Správa kontaktů, dodavatelů a zákazníků.
- Správa výrobků a služeb.
- Správa objednávek.
- Správa bankovních účtů a bankovních transakcí.
- Správa jednotlivých formulářových polí na všech dostupných formulářích.
- Skladová evidence.

- Správa zaměstnanců.
- Možnost české lokalizace pomocí jednoduchého nástroje pro přepsání stávajících textů.
- Široká škála možností pro tvorbu reportů.
- Široká škála možností pro export dat.
- Podpora propojení s internetovým obchodem OSCommerce nebo ZenCart.

System je z hlediska ovládání a přehlednosti horší než výše jmenované. Aplikace sice nepodporuje české prostředí, ale pomocí integrovaného nástroje pro překlad systémových textů, stejně jako aplikace NolaPro (viz 4.2.4), lze systém upravit pro naše potřeby (za cenu času stráveného překladem).

Co se týče funkčních požadavků, systém nesplňuje požadavky logistiky, terminálové části a mzdové agendy. Ostatní požadavky splňuje v některých případech jen částečně. Z těchto důvodů aplikaci taktéž nelze použít pro účely firmy.

4.2.6 Tine 2.0

Tine 2.0 [32] je open-source projekt, kombinující výhody CRM a ERP aplikací do jednoho systému. Je proto užitečný pro celou společnost, od personálu až k podpoře kancelářské práce. Je naprogramován v PHP a lze použít buď MySQL nebo Oracle databázi. Z toho vyplývá, že je platformě nezávislý a podporuje celou řadu standardních prohlížečů.

Hlavní rysy aplikace

- Správa kontaktů.
- Správa výrobků.
- Správa objednávek.
- Správa úkolů a kalendář událostí.
- Správa telefonních účtů.
- Správa VOIP serverů Asterix.

Z popisu aplikace vyplývá, že se jedná o systém vhodný pro celou oblast řízení společnosti, kombinující ERP a CRM systém. Po detailní analýze však zjistíme, že tato

aplikace pokulhává ve všech zkoumaných aspektech. Z hlediska funkčních požadavků splňuje tato aplikace pouze část pro správu úkolů. Ve všech ostatních hrubě nesplňuje skoro žádné požadavky. Jedinou výhodou Tine 2.0 je možnost zvolit české prostředí (ovšem i toto je zpracováno zhruba z 80%). Proto není vhodné použít tento systém pro účely firmy.

4.2.7 UzERP

UzERP [33] je open-source systém kombinující výhody ERP a CRM aplikací. Je určen pro malé a střední podniky. Je naprogramován v jazyce PHP a používá MySQL databázi.

Hlavní rysy aplikace

- Fakturace.
- Správa kontaktů, dodavatelů a zákazníků.
- Správa výrobků a služeb.
- Správa objednávek.
- Správa bankovních účtů a bankovních transakcí v různých měnách.
- Správa jednotlivých formulářových polí na všech dostupných formulářích.
- Skladová evidence s možností vytvořit více skladů.
- Správa úkolů a projektů, skupinový kalendář.
- Správa zaměstnanců.
- Správa uživatelských účtů.
- Hromadné odesílání mailů uživatelům/skupinám.
- Široká škála možností pro tvorbu reportů.
- Široká škála možností pro export dat.
- Podpora propojení s internetovým obchodem OSCommerce.
- Podpora pro vytvoření a správu firemních webových stránek.

Ze všech dosud představených aplikací bych jej zařadil na druhé místo hned po systému NolaPro (viz 4.2.4). UzERP je přehledný a jednoduchý na ovládání. Bohužel jako většina

uvedených aplikací nepodporuje české prostředí. Z hlediska funkčních požadavků splňuje systém téměř všechny podstatné základní požadavky. Bohužel zde opět chybí část pro logistiku, která je klíčovou pro chod firmy, dále pak mzdová agenda či terminálová část. To jsou důvody proč tuto aplikaci taktéž nelze použít pro účely firmy.

4.2.8 Vtiger

Vtiger [34] je open-source CRM systém ideální pro malé a střední podniky. Je naprogramován v jazyce PHP a používá MySQL databázi. Instalace tohoto systému je velmi jednoduchá, nezbytný software je integrovaný v jediném spustitelném souboru pro OS Windows nebo Linux. V případech, kdy potřebujeme implementovat novou funkcionalitu, se můžeme obrátit na tým programátorů vtigeru, jenž za určitý finanční obnos poskytuje své služby. Taktéž lze do aplikace dokoupit komerční moduly (zejména komponenty pro práci se soubory typu Office, Outlook, apod.).

Hlavní rysy aplikace

- Fakturace.
- Správa kontaktů, dodavatelů a zákazníků.
- Správa výrobků a služeb.
- Správa objednávek.
- Správa úkolů.
- Správa bankovních účtů a bankovních transakcí v různých měnách.
- Správa jednotlivých formulářových polí na všech dostupných formulářích.
- Skladová evidence s možností vytvořit více skladových míst.
- Správa dokumentů.
- Správa marketingových kampaní.
- Hromadné odesílání mailů uživatelům/skupinám.
- Široká škála možností pro tvorbu reportů.
- Široká škála možností pro export dat.
- Možnost využít klientské aplikace pro mobilní zařízení.

Tento systém je velice povedený jak z hlediska přehlednosti, tak i ovládání. Jistou nevýhodu může být více různých podoken zobrazených na jedné ploše, což v některých případech ztěžuje orientaci. O oblibě tohoto systému svědčí více než jeden milión stažených kopií. Systém však neposkytuje české prostředí a z hlediska funkčních požadavků nesplňuje požadavky na správu zaměstnanců a mzdovou agendu, logistiku a terminálovou část. Proto ani tato aplikace není vhodná pro účely firmy.

4.2.9 WebERP

WebERP [36] je open-source účetní/ERP systém. Svými vlastnostmi je vhodný pro mnoho podniků, zvláště pak pro distribuční podniky či pro účely velkoobchodní distribuce. Je naprogramován v jazyce PHP a používá MySQL databázi.

Hlavní rysy aplikace

- Fakturace.
- Správa kontaktů, dodavatelů a zákazníků.
- Správa výrobků a služeb.
- Správa objednávek.
- Správa bankovních účtů a bankovních transakcí v různých měnách.
- Správa výroby, vytváření pracovních příkazů, sestavení kusovníků.
- Skladová evidence s možností vytvořit více skladových míst.
- Široká škála možností pro tvorbu reportů.
- GPS lokalizace jednotlivých zákazníků/dodavatelů, zobrazení mapy.

Uživatelské rozhraní tohoto systému působí vcelku zastarale, na druhou stranu však přehledně. Ovládací prvky jsou rozmístěny logicky a intuitivně. Systém opět neposkytuje české prostředí a z hlediska funkčních požadavků nesplňuje požadavky na správu zaměstnanců a mzdovou agendu, logistiku a terminálovou část. I ostatní požadavky nejsou v některých případech zcela splněny. Proto tedy ani tato aplikace není vhodná pro účely firmy.

4.3 Vyhodnocení analýzy

Ze všech uvedených informačních systémů, podrobených analýze, ani jeden nevyhovuje zároveň všem funkčním i nefunkčním požadavkům. Nejvíce se zadaným požadavkům blíží systémy NolaPro, následuje UzERP spolu s Dolibarr a PhreeBooks. Všechny čtyři však na druhou stranu nespĺňují nefunkční požadavek na české prostředí.

Z této analýzy jsem usoudil, že nejlepší možná cesta pro splnění všech požadavků je vytvořit komponenty do současného informačního systému používaného ve firmě. K tomuto faktu mne rovněž vedlo prvotní přání ze strany firmy, jelikož zavádět nový informační systém by s sebou neslo, mimo vyšší finanční náročnost a nutnost školení uživatelů, také mnohem větší náročnost časovou.

5 POUŽITÉ TECHNOLOGIE

Jelikož se jedná o webovou aplikaci, používáme značkový jazyk HTML spolu s kaskádovými styly CSS (pro oddělení grafické a obsahové stránky). Jako programovací jazyk je pro celou aplikaci použit jazyk PHP. Data jsou uloženy v databázi MySQL. Tyto technologie jsou dobře známy, tudíž je jejich popis zbytečný. Dále v aplikaci používáme databázovou vrstvu Dibi a javascriptový framework⁶ Prototype.

5.1 Dibi

Dibi [21] je open-source databázová vrstva podporující několik typů databází. Jejím autorem je D. Grudl. Existuje ve dvou verzích. Klasická verze obsahuje několik adresářů, každá třída je umístěná ve samostatném souboru, tedy běžný programovací přístup. Minimalistická verze je umístěná v jednom malém souboru (velikosti 89 kB) a používá se v produkčním prostředí. Tato databázová vrstva obsahuje ovladače pro 8 typů databází (MySQL, MySQLi, PostgreSQL, SQLite, ODBC a experimentální MSSQL, Oracle a PDO). Nespornou výhodou je možnost měnit typ databáze (pouhou změnou ovladače u připojení k databázi), aniž bychom museli měnit zápis databázových dotazů. Při zápisu těchto dotazů ovšem musíme dodržet několik pravidel.

Příklad databázového dotazu vrstvy dibi

```
1 dibi::query('INSERT INTO
2           [stroje_pracovni_doba]
3           SET
4           [id_stroj]=%i', $stroje["id_stroj"],
5           ', [datum]=%d', $datum->format("Y-m-d"),
6           ', [pracuje_od]=%t', $pracuje["od"],
7           ', [pracuje_do]=%t', $pracuje["do"], ''');
```

SQL příkaz zapíšeme jako sérii parametrů. Veškeré parametry musí být uzavřeny do hranatých závorek. Podle typu aktivní databáze jsou pak dotazy převedeny dle konkrétní

⁶ Framework - softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji. [23]

syntaxe. Před vložením proměnné uvedeme modifikátor. Pokud ho neuvedeme, zjistí se datový typ automaticky (kromě datových typů jako je datum apod). Proměnná se naformátuje do výsledného dotazu podle pravidel aktivní databáze. Stejným způsobem se zformátují i časové údaje, řetězce atd.

5.2 Prototype

Prototype [31] je open-source javascriptový framework, který se zaměřuje na jednoduchost vývoje dynamických webových aplikací. Obsahuje snadno použitelný soubor nástrojů pro vývoj tříd a přívětivou knihovnu funkcí AJAX⁷. Usnadňuje práci programátora především tím, že nabízí užitečné funkce, které se nejčastěji používají na webových stránkách a které by jinak programátor musel psát sám. O jeho oblibě svědčí i použití na webových stránkách firem Apple či Nasa.

Prototype poskytuje zjednodušenou formu přístupu k objektům DOM⁸.

Ukázka přístupu k objektům DOM

```
1 <div id="ukazka"></div>
2 <script type="text/javascript">
3 //Klasický přístup pomocí javascriptu:
4 document.getElementById("ukazka");
5 //Přístup pomocí frameworku Prototype:
6 $("ukazka");
7 </script>
```

Podpora AJAXových funkcí je provedena taktéž velice jednoduše a elegantně a funguje navíc pro všechny standardní webové prohlížeče.

⁷ AJAX (Asynchronous JavaScript and XML) - obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání. Na rozdíl od klasických webových aplikací poskytují uživatelsky příjemnější prostředí, ale vyžadují použití moderních webových prohlížečů. [19]

⁸ DOM (Document Object Model) - platformě a jazykově-neutrální rozhraní, které dovoluje programům a skriptům dynamicky zpřístupnit a aktualizovat obsah, strukturu a styl dokumentů. [35]

Příklad AJAXového volání funkce

```
1 <script type="text/javascript">
2 Ajax.Request('/url_php_skriptu.php', {
3   method: 'post',
4   parameters: {promena: 'hodnota'},
5   onSuccess: function(vysledek) {
6     $('ukazka').innerHTML = 'Úspěšně provedeno !';
7   }
8 });
9 </script>
```

Tato funkce pošle požadavek na php skript *url_php_skriptu.php*, paramter *promena* s hodnotou „*hodnota*“ pošle tomuto skriptu metodou *post*. Pokud byla tato akce úspěšná, provede se funkce *vysledek*, která vloží do obsahu html prvku div s identifikátorem *ukazka* text „Úspěšně provedeno !“

6 POPIS IMPLEMENTACE

Ze všeho nejdříve je nutno říci, že stávající informační systém firmy se jako takový stále vyvíjí. Na vývoji pracuje, kromě části kterou zpracovávám já v této práci, i další programátor řešící jiné úkoly. Informační systém se vyvíjí postupně již asi dva roky.

System sám o sobě není naprogramovaný v žádném ze známých frameworků, ale v jednoduchém frameworku, který vyvinul původní programátor. Tento framework se skládá ze základních funkcí nutných pro autentizaci uživatelů a přihlášení do aplikace, dále obsahuje některé užitečné funkce pro práci s databází (např. automatické vykreslení tabulek či prvků formuláře pouze z několika zadaných parametrů). Dále lze pomocí dědičnosti jedné třídy vytvořit kompletní uživatelské rozhraní a zbytek programovat dle svého uvážení, čehož jsem při své práci využil.

Tedy důvodem, proč není aplikace naprogramována v některém ze známých frameworků (původně bylo zamýšleno psát aplikaci v dnes stále více oblíbeném frameworku Nette [27]), je nutnost kompletního přepsání veškerých stávajících modulů, což není cílem této práce.

Před samotným započítím implementace bylo nutné seznámit se stávajícím informačním systémem, jeho databází a procesy, které zde probíhají. Nutno podotknout, že díky chybějící dokumentaci k systému jsem veškeré poznatky získával pomocí metody reverse engineering⁹, což bylo značně časově náročné.

V následujících podkapitolách je uveden popis implementace jednotlivých nástrojů.

⁹ Reverse engineering (zpětné inženýrství) - proces analýzy předmětného systému s cílem identifikovat komponenty systému a jejich vzájemné vazby a/nebo vytvořit reprezentaci systému v jiné formě nebo na vyšší úrovni abstrakce. [17]

6.1 Nástroj pro správu katalogových informací

Nástroj pro správu katalogových informací jsem vyvíjel podle funkčních požadavků uvedených v 4.1.1.

6.1.1 Třídy

Vzhledem k tomu, že byla tato část aplikace vytvořena ze všeho nejdříve, nejsou třídy rozděleny na aplikační a třídy modelu. Toto doporučení bylo vzneseno vedoucím mé diplomové práce až po prvních konzultacích a seznámením se systémem. Veškeré databázové dotazy jsou tedy součástí tříd, které obsahují funkcionality aplikace, což však nemá vliv na funkčnost jako takovou. Veškeré níže uvedené třídy jsou popsány velice zevrubně a pouze z hlediska toho, jakou funkci v systému zastávají. Pro detailní pohled na třídy, jejich metody, atributy či zdrojové kódy je k dispozici vygenerována programová dokumentace ve formě HTML stránek. Je dostupná v příloze č. P I.

KatalogVyrobku

Třída *KatalogVyrobku* obsahuje základní výpis katalogových výrobků s možností jejich řazení podle různých parametrů. Dále lze pomocí této třídy založit nový katalogový výrobek nebo jej editovat či smazat.

KatalogovyVyrobek

Třída *KatalogovyVyrobek* obsahuje funkce pro zpracování dat ze vstupních formulářů (pro vytvoření, editaci či smazání katalogového výrobku) a obsluhu databáze.

KatalogOperaci

Třída *KatalogOperaci* obsahuje základní výpis katalogových operací s možností jejich řazení podle různých parametrů. Dále lze pomocí této třídy založit novou katalogovou operaci nebo ji editovat či smazat.

KatalogovaOperace

Třída *KatalogovaOperace* obsahuje funkce pro zpracování dat ze vstupních formulářů (pro vytvoření, editaci či smazání katalogové operace) a obsluhu databáze.

6.1.2 Databáze

Popis a struktura databázových tabulek zajišťujících datové úložiště pro správu katalogových informací.

Katalogovy_vyrobek

Tabulka reprezentuje katalogové výrobky a jejich parametry.

Tab. 1. Struktura tabulky katalogovy_vyrobek.

PK	Atribut	Datový typ	Popis
✓	c_vyrobku	mediumint(9)	Jednoznačný identifikátor katalogového výrobku.
	popis	varchar(255)	Popis.
	katalogove_cislo	mediumint(9)	Katalogové číslo.
	cena_fce	varchar(255) <i>NULL</i>	Funkce pro výpočet ceny.
	tisknout_stitek	enum('ano') <i>NULL</i>	Možnost tisku štítku.

Katalogova_operace

Tabulka reprezentuje katalogové operace a jejich parametry.

Tab. 2. Struktura tabulky katalogova_operace.

PK	Atribut	Datový typ	Popis
✓	c_katalogove_operace	mediumint(9)	Jednoznačný identifikátor katalogové operace.
	popis	varchar(255)	Popis.
	stroj	smallint(6) <i>NULL</i>	Nepoužívá se.
	mzda_fce	varchar(255)	Funkce pro výpočet ceny.
	cas	varchar(255)	Volba, zda se má u tohoto výrobku tisknout štítek.
	poznámka	text <i>NULL</i>	Poznámka
	nezakazkova	enum('ano') <i>NULL</i>	Volba, zda je možno tuto operaci zpracovat na terminále jako nezakázkovou.
	druh	varchar(40)	Druh stroje, který je způsobilý zpracovat tuto operaci.

Operace_k_vyrobku

Asociativní tabulka reprezentující relaci M:N mezi katalogovými výrobky a katalogovými operacemi.

Tab. 3. Struktura tabulky operace_k_vyrobku.

PK	Atribut	Datový typ	Popis
✓	c_vyrobku	mediumint(9)	Jednoznačný identifikátor katalogového výrobku.
	c_katalogove_operace	mediumint(9)	Jednoznačný identifikátor katalogové operace.
	poradi	tinyint(4)	Pořadí zpracování katalogové operace.
	c_skladove_karty	int(11) <i>NULL</i>	Číslo skladové karty, kterou spotřebovává katalogový výrobek.
	odepisovane_mnozstvi_fce	varchar(255) <i>NULL</i>	Množství odepisované skladové zásoby.

Skladova_karta

Tabulka reprezentuje skladové karty a jejich parametry.

Tab. 4. Struktura tabulky skladova_karta.

PK	Atribut	Datový typ	Popis
✓	c_skladove_karty	int(11)	Jednoznačný identifikátor skladové karty.
	popis	varchar(255)	Popis.
	merna_jednotka	varchar(8)	Měrná jednotka skladové zásoby.
	cena_za_mj	float	Cena za měrnou jednotku.
	minimum	int(255) <i>NULL</i>	Požadované minimální množství zásoby na skladě.
	tisknout_stitek	enum('ano') <i>NULL</i>	Možnost tisku štítku.
	carovy_kod	varchar(80) <i>NULL</i>	Nepoužívá se.

Sklad

Tato tabulka reprezentuje příjem skladových položek na sklad.

Tab. 5. Struktura tabulky sklad.

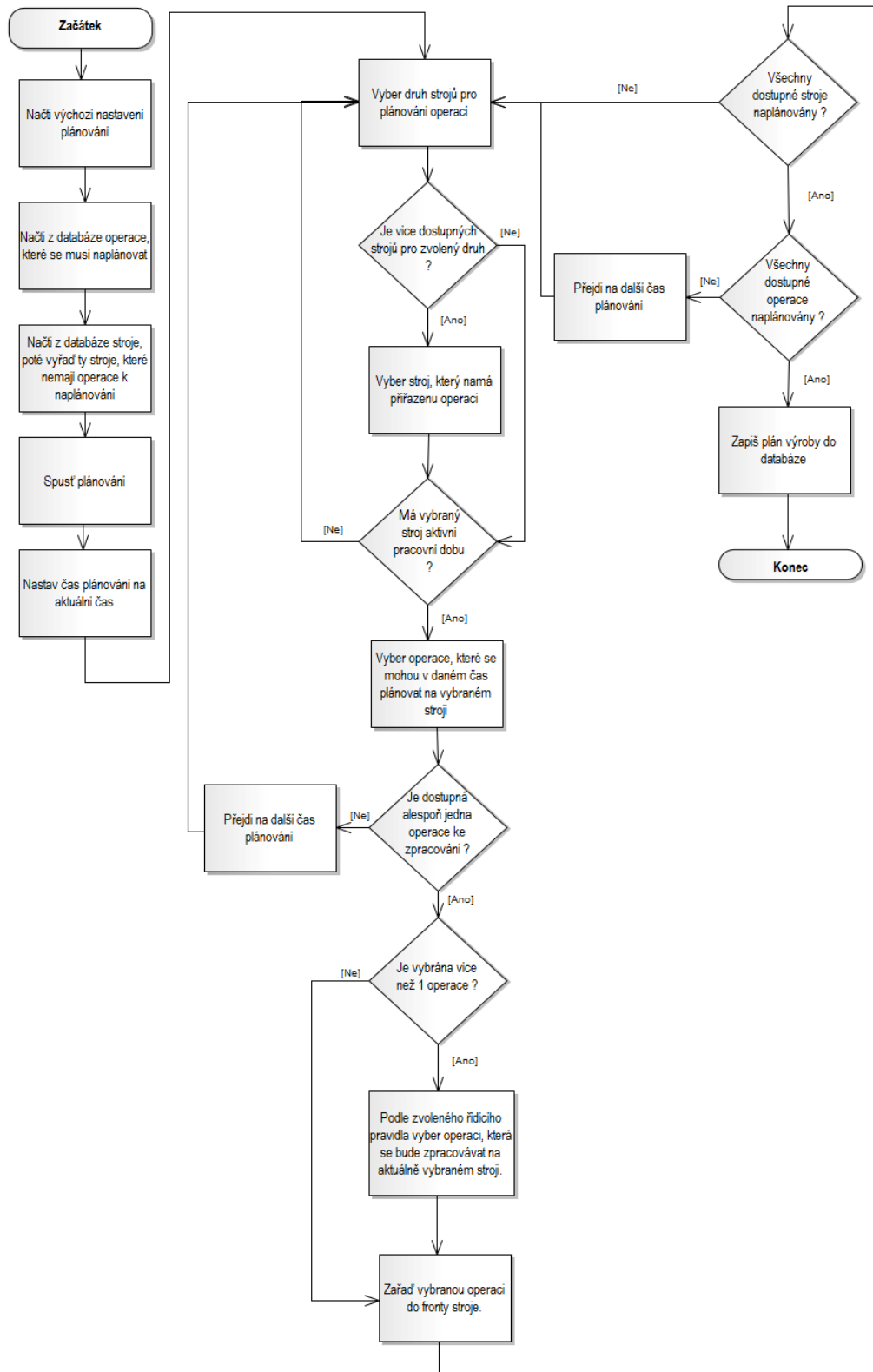
PK	Atribut	Datový typ	Popis
✓	c_skladove_polozky	mediumint(9)	Jednoznačný identifikátor skladové položky.
	evidencni_c	int(9) <i>NULL</i>	Evidenční číslo příjemky.
	c_skladove_karty	int(11)	Číslo skladové karty, pro kterou se provádí příjem.
	mnozstvi	float <i>NULL</i>	Množství přijímané zásoby.
	material_c	varchar(255) <i>NULL</i>	Číslo materiálu.
	datum_objednani	datetime <i>NULL</i>	Datum objednání.
	datum_prijmu	datetime <i>NULL</i>	Datum příjmu na sklad.
	datum_spotreby	datetime <i>NULL</i>	Datum spotřeby ze skladu.
	zustatek	float <i>NULL</i>	Zůstatek na skladě.

6.2 Nástroj pro plánování a rozvrhování výroby

Nástroj pro plánování a rozvrhování výroby jsem vyvíjel podle funkčních požadavků uvedených v 4.1.1.

6.2.1 Plánovací algoritmus

Následující obrázek (Obr. 1) znázorňuje vývojový diagram plánovacího algoritmu.



Obr. 1. Vývojový diagram plánovacího algoritmu.

Před samotným spuštěním plánovacího algoritmu je nejdříve nutné načíst nastavené parametry plánování, dále data o katalogových operacích a strojích a upravit tato data do vhodné podoby. Poté je spuštěn samotný plánovací algoritmus. Po naplánování všech operací na stroje jsou data zapsána do databáze a dále použita k vytvoření grafické podoby plánu výroby. Taktéž jsou operace zpřístupněny v terminálové části ke zpracování (v pořadí určeném plánovacím algoritmem).

6.2.2 Třídy aplikace

Třídy aplikace zajišťují funkcionality aplikace. Veškeré níže uvedené třídy jsou popsány velmi zevrubně a pouze z hlediska toho, jakou funkci v systému zastávají. Pro detailní pohled na třídy, jejich metody, atributy či zdrojové kódy je k dispozici vygenerována programová dokumentace ve formě HTML stránek. Je dostupná v příloze č. P I.

Stroje

Třída *Stroje* zajišťuje správu strojů. Obsahuje metody pro přidávání, editaci a mazání strojů. Součástí je také možnost nastavit týdenní výchozí pracovní dobu stroje.

StrojeDruhy

Třída *StrojeDruhy* zajišťuje správu druhů strojů. Obsahuje metody pro přidávání, editaci a mazání druhů strojů. Každý druh stroje může obsahovat jeden či více strojů.

StrojePracovniDoby

Třída *StrojePracovniDoby* zajišťuje nastavení týdenní pracovní doby strojů. Obsahuje metody pro procházení jednotlivých týdnů, nastavení pracovní doby (nebo volna) pro každý jednotlivý den a stroj nebo pro nastavení volna pro všechny stroje na konkrétní den. Lze taktéž najednou načíst výchozí týdenní pracovní doby pro všechny stroje.

StrojeStatistiky

Třída *StrojeStatistiky* zajišťuje zobrazení statistik. Obsahuje metody pro zobrazení měsíčních statistik vytiženosti jednotlivých strojů, a to jak reálné vytižení (dle zadané pracovní doby), tak i teoretické (při uvažovaném třísměnném provozu). Tato třída využívá uloženou proceduru *PracovniDny* (viz 6.2.4).

Planovac

Třída *Planovac* zajišťuje funkcionalitu plánovacího algoritmu. Obsahuje metody pro inicializaci všech dostupných operací a strojů, přípravu dat pro plánovací algoritmus, implementaci použitých řídicích pravidel (viz 3.3) a další pomocné metody pro běh algoritmu. Taktéž obsahuje metodu pro zápis vypočítaného plánu výroby do databáze.

PlanVyroby

Třída *PlanVyroby* zajišťuje zobrazení vypočítaného plánu výroby. Obsahuje metody pro vykreslení výrobního plánu (ve formě Ganttova diagramu¹⁰), dále pak pro zobrazení detailu (operace či objednávky), vyhledávání podle názvu zákazníka nebo čísla objednávky, aktuální vytížení jednotlivých strojů a neposlední řadě možnost nastavit měřítko a obnovovací dobu plánu.

Plán výroby slouží rovněž pro monitoring výroby. Pokud označíme v terminálové části (viz 8.5) operaci jako hotovou, okamžitě se tato akce promítne i do plánu výroby.

NastaveniPlanovani

Třída *NastaveniPlanovani* zajišťuje zobrazení aktuálního stavu plánovacího procesu a správu parametrů plánování. Obsahuje metody pro zobrazení časových statistik plánovacího algoritmu, dále pro ruční spuštění algoritmu a taktéž pro nastavení parametrů plánovacího algoritmu (zejména řídicí pravidlo). Taktéž slouží jako hlavní menu celého nástroje pro plánování a rozvrhování výroby.

6.2.3 Třídy modelu

Třídy modelu zajišťují datovou vrstvu aplikace. Veškeré databázové dotazy jsou realizovány pomocí databázové vrstvy dibi (viz 5.1). Jednotlivé třídy jsou pojmenovány stejně jako aplikační třídy s tím rozdílem že třídy modelu rovněž obsahují sufix *Model*. Takto je zajištěna orientace ve třídách – každé aplikační třídě náleží třída modelu

¹⁰ Ganttův diagram - Ganttův diagram je grafickou reprezentací rozložení úloh v čase. Je pojmenován po svém vynálezci H.L. Ganttovi. [18]

obsahující metody právě pro tuto třídu. Popis jednotlivých tříd je zbytečný, neboť se jedná převážně o databázové dotazy a v některých případech o kontrolu vstupních dat z formulářů. Pro detailní pohled na třídy, jejich metody, atributy či zdrojové kódy opět odkazují na programovou dokumentaci ve formě HTML stránek. Je dostupná v příloze č. P I.

Pro představu uvádím seznam tříd modelu.

- StrojeModel
- StrojeDruhyModel
- StrojePracovniDobyModel
- StrojeStatistikyModel
- PlanovacModel
- PlanVyrobyModel
- NastaveniPlanovaniModel

6.2.4 Databáze

Popis a struktura databázových tabulek zajišťujících datové úložiště pro plánování a rozvrhování výroby.

Stroje

Tabulka stroje reprezentuje stroje a jejich parametry.

Tab. 6. Struktura tabulky stroje.

PK	Atribut	Datový typ	Popis
✓	id_stroj	mediumint(9)	Jednoznačný identifikátor stroje.
	nazev_stroj	varchar(100)	Název.
	priorita	varchar(100) <i>NULL</i>	Prioritní výrobní parametr.
	cas_prenastaveni	mediumint(9) <i>NULL</i>	Čas, nutný pro přenastavení z jedné hodnoty prioritního výrobního parametru na jinou.
	posl_priorita	varchar(200) <i>NULL</i>	Hodnota posledního prioritního výrobního parametru.
	posl_objednavka	int(11) <i>NULL</i>	Číslo poslední zpracované objednávky.
	id_druh	mediumint(9)	Identifikátor druhu stroje.

Stroje_druh

Tabulky reprezentuje druhy strojů.

Tab. 7. Struktura tabulky stroje_druh.

PK	Atribut	Datový typ	Popis
✓	id_druh	mediumint(9)	Jednoznačný identifikátor druhu stroje.
	nazev_druh	varchar(100)	Název.

Stroje_pracovni_doba

Tabulka reprezentuje denní pracovní doby strojů.

Tab. 8. Struktura tabulky *stroje_pracovni_doba*.

PK	Atribut	Datový typ	Popis
✓	id_doba	bigint(20)	Jednoznačný identifikátor pracovní doby.
	id_stroj	mediumint(9)	Jednoznačný identifikátor stroje.
	datum	date	Datum, pro který je určena pracovní doba.
	pracuje_od	time <i>NULL</i>	Nastavení času, od kdy stroj pracuje.
	pracuje_do	time <i>NULL</i>	Nastavení času, do kdy stroj pracuje.

Stroje_pracovni_doba_default

Tabulka reprezentuje výchozí týdenní pracovní dobu pro jednotlivé stroje.

Tab. 9. Struktura tabulky *stroje_pracovni_doba_default*.

PK	Atribut	Datový typ	Popis
✓	id_stroj	mediumint(9)	Jednoznačný identifikátor stroje.
	PO_od	time NULL	Výchozí pracovní doba, od kdy stroj pracuje v pondělí.
	PO_do	time NULL	Výchozí pracovní doba, do kdy stroj pracuje v pondělí.
	UT_od	time NULL	Stejně tak pro ostatní dny v týdnu.
	UT_do	time NULL	...
	ST_od	time NULL	...
	ST_do	time NULL	...
	CT_od	time NULL	...
	CT_do	time NULL	...
	PA_od	time NULL	...
	PA_do	time NULL	...
	SO_od	time NULL	...
	SO_do	time NULL	...
	NE_od	time NULL	...
	NE_do	time NULL	...

Plan_vyroby

Tabulka reprezentuje plán výroby, tedy data, které zpracoval plánovací algoritmus. Dle těchto dat je plánována výroba.

Tab. 10. Struktura tabulky plan_vyroby.

PK	Atribut	Datový typ	Popis
✓	id	int(11)	Jednoznačný identifikátor operace plánu výroby.
	oznaceni	varchar(20)	Interní označení operace v plánovacím algoritmu.
	cas	int(11)	Doba trvání zpracování operace
	cas_pocatku	int(11)	Čas počátku zpracování operace.
	cas_ukonceni	int(11)	Čas ukončení zpracování operace.
	stroj	mediumint(9)	Stroj, který danou operaci zpracuje.
	c_objednavky	int(11)	Číslo objednávky dané operace.
	c_zakazky	int(11)	Číslo zakázky dané operace.
	c_katalogove_operace	mediumint(9)	Číslo katalogové operace.

Plan_vyroby_nastaveni

Tabulka reprezentuje nastavení parametrů plánovacího algoritmu a plánu výroby.

Tab. 11. Struktura tabulky *plan_vyroby_nastaveni*.

PK	Atribut	Datový typ	Popis
✓	id	int(11)	Jednoznačný identifikátor nastavení.
	planovac_pravidlo	enum('edd','lpt','spt','wspt','mslack','atc','enhanced ecr','dudr')	Zvolené řídicí pravidlo plánovacího algoritmu.
	planovac_stav	tinyint(1) unsigned	Aktuální stav běhu plánovacího algoritmu.
	planovac_posledni_spusteni	datetime	Datum a čas posledního spuštění plánovacího algoritmu.
	planovac_posledni_ukonzeni	datetime	Datum a čas posledního ukončení plánovacího algoritmu.
	plan_meritko	tinyint(4)	Výchozí měřítko plánu výroby.
	plan_obnovovaci_doba	smallint(6)	Výchozí obnovovací doba plánu výroby.

Procedura pro výpočet pracovních dnů

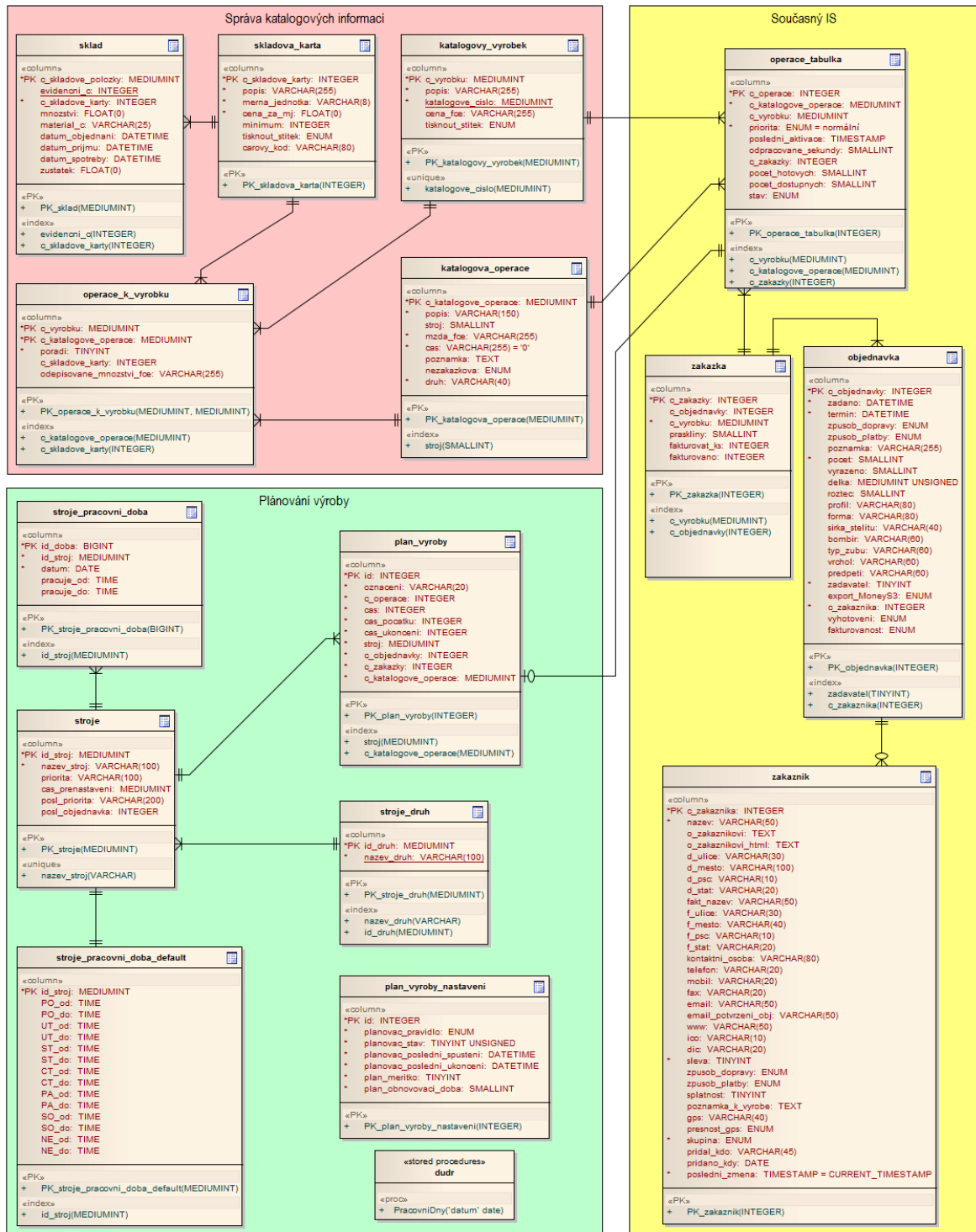
Pro zobrazování statistik strojů je nutné určitým způsobem počítat pracovní dny v zadaném měsíci. V podstatě jsem měl na výběr ze dvou možných řešení. Buď jít cestou programování třídy, která by ze zadaného měsíce vracela počet jeho pracovní dnů, anebo jít cestou vytvoření uložené procedury v MySQL. Jelikož v současné době jazyk PHP neposkytuje tak dobré funkce pro práci s daty jako zmíněné MySQL, padla volba na vytvoření uložené procedury v MySQL.

Zdrojový kód procedury PracovniDny:

```
1 SELECT dd.rozdil AS rozdil, dd.rozdil - dd.vikendy AS pracovniDny,  
   dd.vikendy AS vikendoveDny  
2 FROM (  
3   SELECT  
4     dd.rozdil,  
5     ((dd.tydny * 2) +  
6     IF(dd.sobota >= 0 AND dd.sobota < dd.dny, 1, 0) +  
7     IF (dd.nedele >= 0 AND dd.nedele < dd.dny, 1, 0)) AS vikendy  
8 FROM (  
9   SELECT  
10    dd.rozdil,  
11    FLOOR(dd.rozdil / 7) AS tydny,  
12    dd.rozdil % 7 dny,  
13    5 - dd.zacatek AS sobota,  
14    6 - dd.zacatek AS nedele  
15 FROM (  
16   SELECT  
17     1 + DATEDIFF(LAST_DAY(datum), datum) AS rozdil,  
18     WEEKDAY(datum) AS zacatek  
19   ) AS dd  
20 ) AS dd  
21 ) AS dd
```

Vstupní parametr procedury je proměnná *datum*. Procedura vrací celkový počet dnů, počet pracovních dnů a také počet víkendových dnů (v rozmezí od zadané proměnné *datum* do konce měsíce).

6.3 Celkový pohled na databázi



Obr. 2. E-R diagram části databáze.

Schéma databáze ve formě E-R diagramu (Obr. 2) znázorňuje tři části systému. První část je pojmenovaná jako *Současný IS* a znázorňuje pouze hlavní část stávajícího informačního systému (celkové schéma databáze zde není možné zachytit z důvodu velikosti a počtu

tabulek). Na tuto část jsou v diagramu napojeny další dvě části. *Správa katalogových informací* a *Plánování výroby*. Struktura a popis jednotlivých tabulek jsou uvedeny v kapitolách výše.

7 PROGRAMOVÁ DOKUMENTACE

V této kapitole popisují tvorbu programové dokumentace k systému. Dokumentace byla tvořena během programování aplikace pomocí průběžných poznámek (tzv. docbloků) psaných přímo do zdrojových kódů skriptů. Výhoda tohoto zápisu je jednak v přehlednosti kódů, rovněž spočívá v možnosti vyvolat kontextovou nápovědu u atributů, metod a tříd v editorech, které tuto funkcionalitu nabízí. Třetí výhodou je schopnost vygenerovat pomocí těchto poznámek programovou dokumentaci (např. ve formě HTML stránek). Vygenerovaná dokumentace, vytvořená nástrojem Phpdocumentor, je ve formě HTML stránek dostupná v příloze č. P I.

7.1 Docbloky

Všechny poznámky, které mají být uvedeny v dokumentaci, musí být zapsány před popisovaným elementem v takzvaném docbloku. Za tímto popisem již následuje zápis samotné funkce.

Příklad zápisu docbloku pro libovolnou funkci.

```
1 /**
2  * Vrací pracovní dobu stroje ve zvoleném dni.
3  * @param time $cas čas
4  * @param integer $id_stroj id stroje
5  * @return array
6  */
```

Jako první uvedeme krátký popis funkce, který může být maximálně na tři řádky. Následuje nepovinný popis. Na konci uvedeme seznam značek (tzv. tagů) uvozených znakem „@“. Pomocí tagů přidáváme vlastnosti popisovanému elementu. V ukázkovém příkladě používáme tagy *param* (seznam parametrů popisované funkce) a *return* (popis návratové hodnoty funkce).

7.2 Phpdocumentor

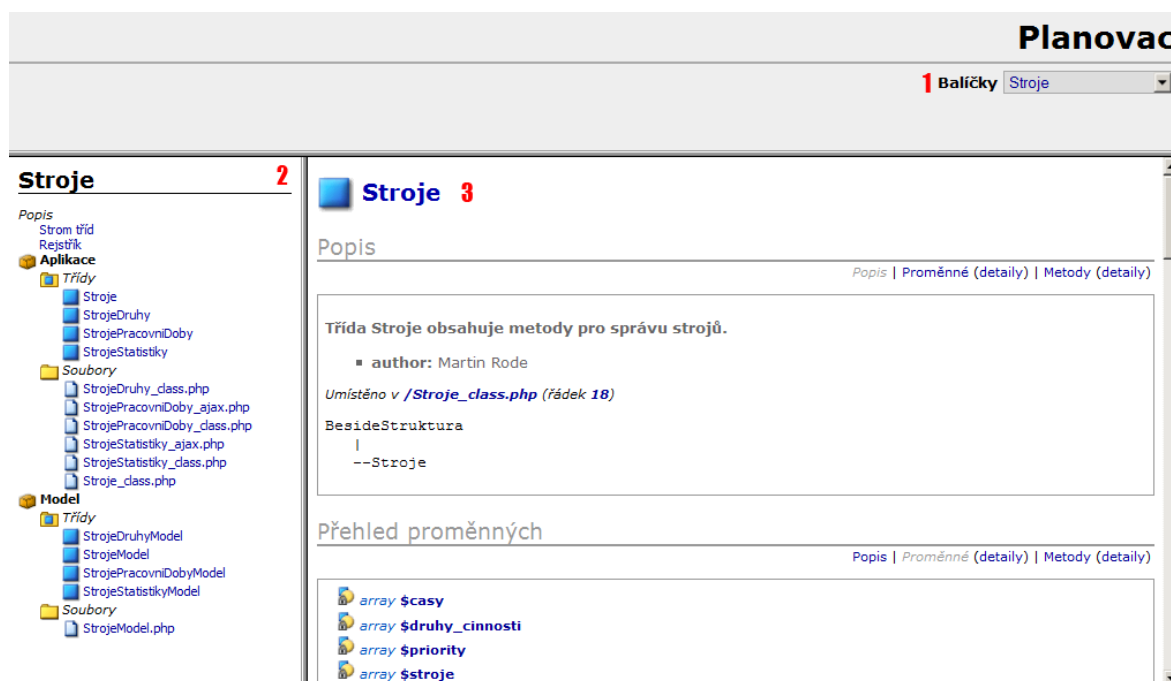
Phpdocumentor [29] je open-source nástroj pro automatickou tvorbu dokumentace programovacího jazyka PHP. V tomto jazyce je celý nástroj rovněž vytvořen. Phpdocumentor lze používat buď přímo z příkazové řádky, nebo skrze uživatelsky

přívětivé webové rozhraní. Pomocí průběžného zápisu poznámek (tzv. docbloků – viz 7.1) můžeme generovat dokumentaci přímo ze zdrojových kódů našeho projektu.

Výhodou takového typu generování dokumentace je velice jednoduché pochopení struktury a významu aplikace. Při změně struktury tak místo úpravy stávající dokumentace jednoduše vygenerujeme dokumentaci novou (čímž ušetříme spoustu času). Pomocí některých značek lze rovněž rozšířit popis a význam struktur (například rozdělit třídy do balíků nebo označit třídy a funkce jako abstraktní apod.). Výsledná dokumentace může být v HTML, CHM, PDF, XML nebo DocBook formátu.

Vzhledem k tomu, že veškeré předdefinované HTML šablony jsou k dispozici pouze v anglické verzi, upravil jsem z jednu z šablon do českého jazyka (viz příloha č. P I), a pomocí ní jsem poté vygeneroval výslednou dokumentaci.

7.3 Náhled na vygenerovanou dokumentaci



Obr. 3. Náhled na vygenerovanou programovou dokumentaci.

Vysvětlivky:

- 1) *Balíčky*. Nabídka s výběrem balíčků. Každý balíček obsahuje soubory a jejich třídy, vztahující se k určité části aplikace.
- 2) Hlavní menu dokumentace. Obsahuje:

- *Strom tříd* – seznam všech tříd, seřazených podle rodičovské třídy.
 - *Rejstřík* – rejstřík všech souborů, jejich tříd, metod a atributů.
 - Seznam tříd a souborů řazených podle podbalíčků:
 - a. *Aplikace* – soubory a třídy aplikace (třídy aplikace, které zajišťují funkcionality).
 - b. *Model* – soubory a třídy modelu (vrstva aplikace, která slouží pro práci s databází).
- 3) Zobrazení vybraného detailu elementu hlavního menu. Na Obr. 3 je vybrána třída *stroje*. K této třídě jsou zobrazeny následující informace. Kliknutím na kterýkoliv prvek se dostaneme do místa deklarace ve zdrojovém kódu.
- *Popis*. Umístění, tj. v jakém souboru a na jakém řádku je třída deklarována, odkaz na zdrojový kód a taktéž je zde znázorněna dědičnost.
 - *Přehled atributů*. Jednoduchý seznam atributů a jejich datový typ.
 - *Přehled metod*. Jednoduchý seznam metod, jejich parametry a návratové hodnoty.
 - *Atributy*. Detailní seznam atributů s popisem.
 - *Metody*. Detailní seznam metod s popisem.

8 UŽIVATELSKÁ DOKUMENTACE

Tato kapitola slouží jako uživatelský manuál k vytvořeným nástrojům. Každá sekce této kapitoly vždy obsahuje náhled aplikace s číslovanými popisky. Tyto jsou poté detailněji popsány ve vysvětlivkách.

8.1 Přihlášení do aplikace

Obr. 4. Formulář pro přihlášení do aplikace.

Vysvětlivky:

- 1) *Jméno.* Pro přihlášení do aplikace je nutno vybrat z nabídky existující uživatelský účet.
- 2) *Heslo.* Zadáním platného hesla a potvrzením pomocí tlačítka *Přihlásit* vstoupíme do aplikace.

8.2 Hlavní okno aplikace

Obr. 5. Hlavní okno aplikace.

Vysvětlivky:

- 1) *Najdi.* Souhrnné pole pro vyhledávání. Zadáním části řetězce se prohledají záznamy v modulech *Zákazníci*, *Výroba* a *Účty*. Výsledek hledání je možno vybrat z vygenerovaného seznamu.
- 2) *Hlavní strana.*

- 3) Modul *Zákazníci*. Slouží pro správu zákazníků.
- 4) Modul *Výroba*. Slouží pro správu objednávek, nastavení priority objednávek. Taktéž lze zobrazit naposledy provedené operace jednotlivých zaměstnanců na strojích.
- 5) Modul *Účty*. Slouží pro správu fakturací a úhrad objednávek. Obsahuje také statistiky faktur dle různých ukazatelů (časové období, uhrazení faktury, typ platby, měna).
- 6) Modul *Sklad*. Slouží pro správu skladových informací. Je možno spravovat skladové zásoby a skladové karty, zobrazit součty skladových zásob, spotřebované zásoby a nedostatky zásob dle nastavených limitů.
- 7) Modul *Katalog*. Slouží pro správu katalogových informací, tj. katalogových výrobků a katalogových operací (viz 8.3).
- 8) Modul *Plánování výroby*. Slouží pro správu druhů strojů, strojů, pracovních dob strojů, nastavení plánování. Obsahuje také statistiky využitelnosti jednotlivých strojů a grafické zobrazení plánu výroby (viz 8.4).
- 9) Odhlášení z aplikace. Kliknutím na odkaz *Odhlásit se z aplikace* odhlásíte.

8.3 Nástroj pro správu katalogových informací

8.3.1 Správa katalogových výrobků

Zobrazení katalogových výrobků a jejich parametrů

The screenshot shows the DUDR web application interface. At the top right, it says 'přihlášen: Dudr Tomáš | odhlásit'. The navigation menu includes: Hlavní strana, Zákazníci, Výroba, Účty, Sklad, Katalog, and Plánování výroby. The main content area is titled 'Katalog výrobků' and contains a search bar (1), a 'Katalog operací' button (2), and a 'Nový katalogový výrobek' button (3). Below is a table titled 'Seznam katalogových výrobků' with columns for 'Číslo výrobku' (4), 'Popis' (5), 'Katalogové číslo' (6), and 'Cenová funkce' (7). The table lists items 161 through 168 with their descriptions, catalog numbers, and price functions. A red '8' is placed at the end of the table header.

Číslo výrobku 4	Popis 5	Katalogové číslo 6	Cenová funkce 7	8
161	Pil. pás UDD 80 x 1,0 válc	5001	0.219 * delka + 129 + 109	
162	Pil. pás UDD 80 x 1,0 nevál.	5002	0.219 * delka + 129	
163	Pil. pás UDD 90 x 1,0 válc	5003	0.249 * delka + 139 + 129	
164	Pil. pás UDD 90 x 1,0 nevál.	5004	0.249 * delka + 139	
165	Pil. pás UDD 90 x 1,1 válc	5005	0.249 * delka + 139 + 129	
166	Pil. pás UDD 90 x 1,1 nevál.	5006	0.249 * delka + 139	
167	Pil. pás UDD 100 x 1,1 válc	5007	0.279 * delka + 179 + 169	
168	Pil. pás UDD 100 x 1,1 nevál.	5008	0.279 * delka + 179	

Obr. 6. Zobrazení katalogových výrobků a jejich parametrů.

Vysvětlivky:

- 1) Pole pro vyhledání katalogového výrobku. Zadáním části řetězce (hledat lze podle čísla nebo podle popisu katalogového výrobku) zobrazíme výsledky hledání. Výsledky se zobrazí pod tímto polem. Je to systém tzv. „našeptávání“, kdy zadáváním jednotlivých znaků zpřesňujeme výsledky hledání, které se zobrazují pod vyhledávacím polem a lze je kdykoliv vybrat.
- 2) *Katalog operací*. Tlačítko pro zobrazení správy katalogových operací.
- 3) *Nový katalogový výrobek*. Tlačítko pro vytvoření nového katalogového výrobku.
- 4) *Číslo katalogového výrobku*.
- 5) *Popis*.
- 6) *Katalogové číslo*.

- 7) *Cenová funkce.*
- 8) Tlačítko pro editaci katalogového výrobku.

Vytvoření nového katalogového výrobku

Postup pro vytvoření nového katalogového výrobku je shodný s postupem pro editaci katalogového výrobku.

Editace stávajícího katalogového výrobku

Najdi: přihlášen: Dudr Tomáš | [odhlásit](#)

Hlavní strana Zákazníci Výroba Účty Sklad Katalog Plánování výroby

Katalog výrobků > Náhled/vložit/úprava katalogového výrobku

Upravit - 256 - Stelitování a broušení

Popis:	<input type="text" value="Stelitování a broušení"/>	1
Katalogové číslo:	<input type="text" value="5095"/>	2
Cenová funkce:	<input type="text" value="12 * round(delka / roztec)"/>	3
Tisknout štítek:	<input type="button" value="▼"/>	4

Katalogové operace

Popis 5	Odepsované množství 6	Skladová karta 7	8
Broušení	<input type="text" value="0.000132 * round(delka /"/>	<input type="text" value="Kotouč k Baracude 99900114"/>	<input type="button" value="⊖"/>
		<input type="text" value="Kotouče na Shark 99900113"/>	
Stelitování	<input type="text" value="1.36 * round(delka / rozte"/>	<input type="text" value="99900093"/>	<input type="button" value="⊖"/>
Ostření stelitu	<input type="text" value="0.000264 * round(delka /"/>	<input type="text" value="99900113"/>	<input type="button" value="⊖"/>
Egalizace	<input type="text" value="0.00000666 * round(delk"/>	<input type="text" value="99900114"/>	<input type="button" value="⊖"/>
Přeoštění pily	<input type="text" value="0.000132 * round(delka /"/>	<input type="text" value="99900113"/>	<input type="button" value="⊖"/>

10

11

12

13

Obr. 7. Formulář pro editaci katalogového výrobku.

Vysvětlivky:

- 1) *Popis.* Pole pro zadání popisu katalogového výrobku.
- 2) *Katalogové číslo.* Pole pro zadání katalogového čísla výrobku. Díky této identifikaci se odepisuje příslušná skladová zásoba.

- 3) *Cenová funkce*. Pole pro zadání cenové funkce. Podle cenové funkce se počítá cena katalogového výrobku pro zákazníky. Funkci zapíšeme matematicky s pomocí zaokrouhlovací funkce *round*. Jako další parametry cenové funkce použijeme parametry výroby, kterými jsou: bombír, délka, forma, profil, předpětí, rozteč, typ zubu, vrchol, šířka. Také zde můžeme zapsat pouze konstantu.
- 4) *Tisknout štítek*. Výběr pro určení, zda se bude ke katalogovému výrobku tisknout štítek s označením a parametry, nebo nebude.
- 5) *Popis*. Popis katalogové operace, která přísluší danému výrobku. Kliknutím a tažením (tzv. metoda Drag & Drop) můžeme měnit pořadí jednotlivých katalogových operací.
- 6) *Odepisované množství*. Pole pro zadání funkce sloužící k odepsání množství skladové zásoby, spotřebované při realizaci této operace. Zápis funkce je totožný se zápisem cenové funkce (bod 3 tohoto seznamu).
- 7) *Skladová karta*. Pole pro zadání skladové karty. Určuje jakou skladovou zásobu katalogová operace spotřebuje. Množství zásoby je určeno funkcí *odepisované množství* zmíněnou výše. Toto pole funguje jako „našeptávač“ (viz 8.3.1).
- 8) Tlačítko pro odebrání katalogové operace.
- 9) Pole pro vyhledání a přidání katalogové operace. Toto pole funguje jako „našeptávač“ (viz 8.3.1). Možnost vyhledávat podle čísla nebo názvu katalogové operace.
- 10) *Ulož*. Tlačítko pro uložení katalogového výrobku.
- 11) *Neukládat, ukaž podrobnosti ke katalogovému výrobku*. Tlačítko pro zobrazení detailu katalogového výrobku. Zadané změny ve formuláři se neprojeví.
- 12) *Vymazat katalogový výrobek z databáze*. Tlačítko pro smazání katalogového výrobku.
- 13) *Zpět na katalog výrobků*. Tlačítko pro zobrazení seznamu stávajících katalogových výrobků. Zadané změny ve formuláři se neprojeví.

Smazání stávajícího katalogového výrobku

Smazání katalogového výrobku provedeme zobrazením formuláře pro editaci katalogového výrobku a stisknutím tlačítka *Vymazat katalogový výrobek z databáze*.

8.3.2 Správa katalogových operací

Zobrazení katalogových operací a jejich parametrů

The screenshot shows the 'Katalog operací' page in the DUDR system. At the top, there is a search bar labeled 'Najdi:' and a user login 'přihlášen: Dudr Tomáš | odhlásit'. Below the search bar is a navigation menu with buttons for 'Hlavní strana', 'Zákazníci', 'Výroba', 'Účty', 'Sklad', 'Katalog', and 'Plánování výroby'. The main content area is titled 'Katalog operací' and contains a search input field (1), a button for 'Katalog výrobků' (2), and a button for 'Nová katalogová operace' (3). Below this is a table titled 'Seznam katalogových operací' with the following columns: 'Číslo katalogové operace' (4), 'Popis' (5), 'Čas' (6), 'Druh' (7), 'Mzdová funkce' (8), 'Poznámka' (9), and 'Nezakázková' (10, 11). The table lists six operations (IDs 15-20) with their respective parameters.

Číslo katalogové operace	Popis	Čas	Druh	Mzdová funkce	Poznámka	Nezakázková
15	Broušení	(delka/roztec)*8	lis 210	(0,15 * round(delka / roztec)) * 0,8	Vybroušení daného profilu	
16	Egalizace	25*(delka/roztec)	Baracuda	(0,45 * round(delka / roztec)) * 0,8	Egalizace na danou šířku	
17	Stříhání UDD 80 x 1.0	delka*0,06	stříhání	(0,001 * delka) * 0,8		
18	Stříhání UDD 90 x 1.0	delka*0,06	stříhání	(0,001 * delka) * 0,8		
19	Stříhání UDD 90 x 1.1	delka*0,06	stříhání	(0,001 * delka) * 0,8		
20	Stříhání UDD 100 x 1.1	delka*0,06	stříhání	(0,001 * delka) * 0,8		

Obr. 8. Zobrazení katalogových operací a jejich parametrů.

Vysvětlivky:

- 1) Pole pro vyhledání katalogové operace. Zadáním části řetězce (hledat lze podle čísla nebo podle popisu katalogové operace) zobrazíme výsledky hledání. Toto pole funguje jako „našeptávač“ (viz 8.3.1).
- 2) *Katalog výrobků*. Tlačítko pro zobrazení správy katalogových výrobků.
- 3) *Nová katalogová operace*. Tlačítko pro vytvoření nové katalogové operace.
- 4) *Číslo katalogové operace*.
- 5) *Popis*.
- 6) *Čas*.

- 7) *Druh.*
- 8) *Mzdová funkce.*
- 9) *Poznámka.*
- 10) *Nezakázková.*
- 11) Tlačítko pro editaci katalogové operace.

Vytvoření nové katalogové operace

Postup pro vytvoření nové katalogové operace je shodný s postupem pro editaci katalogové operace.

Editace stávající katalogové operace

Najdi: přihlášen: Dudr Tomáš | [odhlásit](#)

Hlavní strana | Zákazníci | Výroba | Účty | Sklad | Katalog | Plánování výroby

Katalog operací > Náhled/vložit/úprava katalogové operace

Upravit - 17 - Střihání UDD 80 x 1.0

Popis:	<input type="text" value="Střihání UDD 80 x 1.0"/>	1
Mzda funkce:	<input type="text" value="(0,001 * delka) * 0,8"/>	2
Čas:	<input type="text" value="delka*0,06"/>	3
Druh stroje:	<input type="text" value="střihání"/>	4
Poznámka:	<input type="text"/>	5
Nezakázková:	<input type="text"/>	6

Ulož 7

[Neuklídat, ukaž podrobnosti ke katalogové operaci 8](#)

[Vymazat katalogovou operaci z databáze 9](#)

[Zpět na katalog operací 10](#)

Obr. 9. Formulář pro editaci katalogové operace.

Vysvětlivky:

- 1) *Popis.* Pole pro zadání popisu katalogové operace.
- 2) *Mzda funkce.* Pole pro zadání funkce pro mzdu. Podle mzdové funkce se počítá úkolová mzda zaměstnancům. Zápis funkce je totožný se zápisem cenové funkce (viz 8.3.1).

- 3) *Čas*. Pole pro zadání časové funkce. Podle časové funkce se počítá čas zpracování katalogové operace na stroji. Zápis funkce je totožný se zápisem cenové funkce (viz 8.3.1).
- 4) *Druh stroje*. Pole pro výběr druhu stroje, na kterém se katalogová operace zpracuje. Nabídku druhů strojů lze editovat v sekci druhy strojů (viz 8.4.3).
- 5) *Poznámka*. Pole pro zadání poznámky.
- 6) *Nezakázková*. Pole pro výběr, zda je tato operace dostupná v terminálové části¹¹ jako nezakázková.
- 7) *Ulož*. Tlačítko pro uložení katalogového výrobku.
- 8) *Neukládat, ukaž podrobnosti ke katalogové operaci*. Tlačítko pro zobrazení detailu katalogové operace. Zadané změny ve formuláři se neprojeví.
- 9) *Vymazat katalogovou operaci z databáze*. Tlačítko pro smazání katalogové operace.
- 10) *Zpět na katalog operací*. Tlačítko pro zobrazení seznamu stávajících katalogových operací. Zadané změny ve formuláři se neprojeví.

Smazání stávající katalogové operace

Smazání katalogové operace provedeme zobrazením formuláře pro editaci katalogové operace a stisknutím tlačítka *Vymazat katalogovou operaci z databáze*.

¹¹ Terminálová část – samostatná část aplikace, která běží na panelových počítačích umístěných u strojů.

8.4 Nástroj pro plánování a rozvrhování výroby

8.4.1 Plánování výroby

Najdi: přihlášen: Dudr Tomáš | [odhlásit](#)

Hlavní strana Zákazníci Výroba Účty Sklad Katalog Plánování výroby

Plánování výroby

Aktuální parametry plánování

Poslední čas spuštění	1	11:39:51
Poslední čas ukončení	2	11:39:52
Další plánované spuštění	3	12:00:00
Zvolené rozvrhovací pravidlo	4	dudr
Výchozí měřítko plánu výroby	5	1 : 1
Výchozí obnovovací doba plánu výroby	6	300 s

Další automatické spuštění plánovače za:

00:19:56 7

SPUSTIT RUČNĚ 8

Stroje

- Stroje **9**
- Druhy strojů **10**
- Pracovní doby strojů **11**
- Statistiky strojů **12**

Plánování výroby

- Zobrazit plán výroby **13**
- Nastavení plánování **14**

Obr. 10. Hlavní menu modulu plánování výroby.

Vysvětlivky:

- 1) *Poslední čas spuštění.* Zobrazuje poslední čas spuštění plánovacího algoritmu.
- 2) *Poslední čas ukončení.* Zobrazuje poslední čas ukončení plánovacího algoritmu. Rozdíl mezi tímto časem a časem posledního spuštění udává délku běhu plánovacího algoritmu.
- 3) *Další plánované spuštění.* Zobrazuje čas, kdy bude automaticky spuštěn plánovací algoritmus. Toto se děje každých 30 minut.
- 4) *Zvolené rozvrhovací pravidlo.* Zobrazuje pravidlo, které se použije v plánovacím algoritmu pro systém plánování operací.
- 5) *Výchozí měřítko plánu výroby.* Zobrazuje měřítko plánu výroby, které se použije jako výchozí.
- 6) *Výchozí obnovovací doba plánu výroby.* Zobrazuje obnovovací dobu plánu výroby, která se použije jako výchozí.

- 7) Zobrazuje odpočet času, po kterém se provede automatické spuštění plánovacího algoritmu.
- 8) Tlačítko pro ruční spuštění plánovacího algoritmu. Slouží k ručnímu spuštění v případě, že potřebujeme plánovací algoritmus spustit okamžitě, bez čekání na automatické spuštění. Tlačítko je aktivní pouze v době od konce úspěšného provedení plánovacího algoritmu do doby 4 minut před automatickým spuštěním (z důvodu možné kolize ručního a automatického spuštění). V ostatních případech je zde zobrazena informace o právě probíhajícím přepočtu plánu výroby nebo informace o zákazu ručního spuštění.
- 9) *Stroje*. Tlačítko pro zobrazení správy strojů (viz 8.4.2).
- 10) *Druhy strojů*. Tlačítko pro zobrazení správy druhů strojů (viz 8.4.3).
- 11) *Pracovní doby strojů*. Tlačítko pro zobrazení správy pracovních dob strojů (viz 8.4.4).
- 12) *Statistiky strojů*. Tlačítko pro zobrazení měsíčních statistik využitelnosti strojů (viz 8.4.5).
- 13) *Zobrazit plán výroby*. Tlačítko pro zobrazení aktuálního plánu výroby (viz 8.4.6).
- 14) *Nastavení plánování*. Tlačítko pro zobrazení úpravy nastavení plánování (viz 8.4.7).

8.4.2 Stroje

Zobrazení strojů a jejich parametrů

Najdi: přihlášen: Dudr Tomáš | [odhlásit](#)

Hlavní strana Zákazníci Výroba Účty Sklad Katalog Plánování výroby

Plánování výroby > Stroje

Stroje

[Vytvořit nový stroj](#) 1

Název stroje 2	Druh stroje 3	Priorita 4	Čas přenastavení 5	Editovat 6	Smazat 7
balení nožů	balení nožů				
Baracuda	Baracuda				
BAS 100	BAS 100				
BAS 210	BAS 210				
BAS 60	BAS 60				
broušení čelistí	broušení čelistí				
EKV 2000	EKV 2000	forma	1200		
jiné	jiné				

Obr. 11. Zobrazení strojů a jejich parametrů.

Vysvětlivky:

- 1) *Vytvořit nový stroj.* Tlačítko pro vytvoření nového stroje.
- 2) *Název stroje.*
- 3) *Druh stroje.*
- 4) *Priorita.*
- 5) *Čas přenastavení.*
- 6) Tlačítka pro editaci stroje.
- 7) Tlačítka pro smazání stroje.

Vytvoření nového stroje

Najdi: přihlášen: Dudr Tomáš | [odhlásit](#)

DUDR

Hlavní strana Zákazníci Výroba Účty Sklad Katalog Plánování výroby

Plánování výroby > Stroje > Nový stroj

Nový stroj

Nový stroj

Název stroje: **1**

Priorita: **2**

Čas přenastavení: **3**

Druh stroje: BAS 100 **4**

Výchozí pracovní doba 5

	OD	DO
Pondělí	<input type="text" value="06:00:00"/>	<input type="text" value="14:00:00"/>
Úterý	<input type="text" value="06:00:00"/>	<input type="text" value="14:00:00"/>
Středa	<input type="text" value="06:00:00"/>	<input type="text" value="14:00:00"/>
Čtvrtek	<input type="text" value="06:00:00"/>	<input type="text" value="14:00:00"/>
Pátek	<input type="text" value="06:00:00"/>	<input type="text" value="14:00:00"/>
Sobota	<input type="text" value="volno"/>	<input type="text" value="volno"/>
Neděle	<input type="text" value="volno"/>	<input type="text" value="volno"/>

Obr. 12. Formulář pro vytvoření nového stroje.

Vysvětlivky:

- 1) *Název stroje.* Pole pro zadání názvu stroje. Název stroje musí být unikátní, tzn. nelze použít název již existujícího stroje.
- 2) *Priorita stroje.* Označuje parametr výroby. Je možno vybrat z následujících parametrů: bombír, délka, forma, profil, předpětí, rozteč, typ zubu, vrchol, šířka. Tento parametr určuje to, zda se bude při změně priority mezi dvěma operacemi přičítat čas nutný k přenastavení stroje. Například pokud bude určitý stroj zpracovávat operaci „A“ s prioritou profil rovnu „10“ a následující operace „B“ na tomto stroji bude mít nastavenou taktéž prioritu profil, ovšem ta bude rovna „20“, bude se k operaci „B“ přičítat čas (viz čas přenastavení), nutný k přenastavení stroje.
- 3) *Čas přenastavení.* Určuje časový přírůstek, který se přičte k následující operaci při změně priority stroje.

- 4) *Druh stroje.* Výběr druhu stroje. Vzhledem k tomu, že může existovat více strojů, které provádějí stejnou činnost, je nutné určit druh stroje. Druhy strojů se určují u každé katalogové operace (viz 8.3.2). Nabídku druhů strojů lze editovat v sekci druhy strojů (viz 8.4.3).
- 5) *Výchozí pracovní doba.* Tato část slouží k nastavení výchozí týdenní pracovní doby stroje. Pro každý den v týdnu je možno nastavit časy od kdy a do kdy bude stroj pracovat. Vybrat lze z několika časů podle toho, jaký pracovní cyklus bude stroj vykonávat. Toto nastavení je důležité pro automatické načítání týdenní pracovní doby strojů (viz 8.4.4).

Editace stávajícího stroje

Postup při editaci stávajícího stroje je totožný se postupem pro vytvoření nového stroje.

Smazání stávajícího stroje

Pro smazání stávajícího stroje je nutné v seznamu strojů vybrat příslušný stroj a kliknout na ikonu smazat. Následně se zobrazí detail stroje, a tlačítko *Smaž*, kterým se smazání stroje potvrdí.

8.4.3 Druhy strojů

Část druhy strojů slouží ke správě druhů strojů. V této části lze zobrazit všechny stávající druhy strojů, založit nový druh, editovat a smazat stávající druhy.

Vytvoření nového druhu

Pro vytvoření nového druhu je potřeba kliknout na tlačítko *Vytvořit nový druh*. Poté se do formuláře zadá název druhu a potvrdí tlačítkem *Uložit*. Název druhu stroje musí být unikátní, tzn. nelze použít název již existujícího druhu.

Editace stávajícího druhu

Postup při editaci stávajícího druhu je totožný se postupem pro vytvoření nového druhu.

Smazání stávajícího druhu

Pro smazání stávajícího druhu je nutné v seznamu druhů strojů vybrat příslušný druh a kliknout na ikonu smazat. Následně se zobrazí detail stroje a tlačítko *Smaz*, kterým se smazání stroje potvrdí.

8.4.4 Pracovní doby strojů

Najdi: přihlášen: Dudr Tomáš | [odhlásit](#)

Hlavní strana Zákazníci Výroba Účty Sklad Katalog Plánování výroby

Plánování výroby > Pracovní doby strojů

Pracovní doby strojů

[Načíst výchozí pracovní doby](#) **1**

[Smazat pracovní doby](#) **2**

3
← 18. pracovní týden →

Stroje 4	Neděle 5 2.5.2010 nastav volno 6	Pondělí 3.5.2010 nastav volno	Úterý 4.5.2010 nastav volno	Středa 5.5.2010 nastav volno	Čtvrtek 6.5.2010 nastav volno	Pátek 7.5.2010 nastav volno	Sobota 8.5.2010 nastav volno
balení nožů	volno volno	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	volno volno
Baracuda	22:00:00 23:59:59	00:00:00 23:59:59	00:00:00 23:59:59	00:00:00 14:00:00 Uložit 7 Strono	00:00:00 23:59:59	00:00:00 22:00:00	volno volno
BAS 100	volno volno	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	volno volno
BAS 210	volno volno	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	volno volno
BAS 60	volno volno	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	volno volno
broušení čelistí	volno volno	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	06:00:00 14:00:00	volno volno
EKV 2000	22:00:00 23:59:59	00:00:00 23:59:59	00:00:00 23:59:59	00:00:00 23:59:59	00:00:00 23:59:59	00:00:00 22:00:00	volno volno
jiné	22:00:00 23:59:59	00:00:00 23:59:59	00:00:00 23:59:59	00:00:00 23:59:59	00:00:00 23:59:59	00:00:00 22:00:00	volno volno

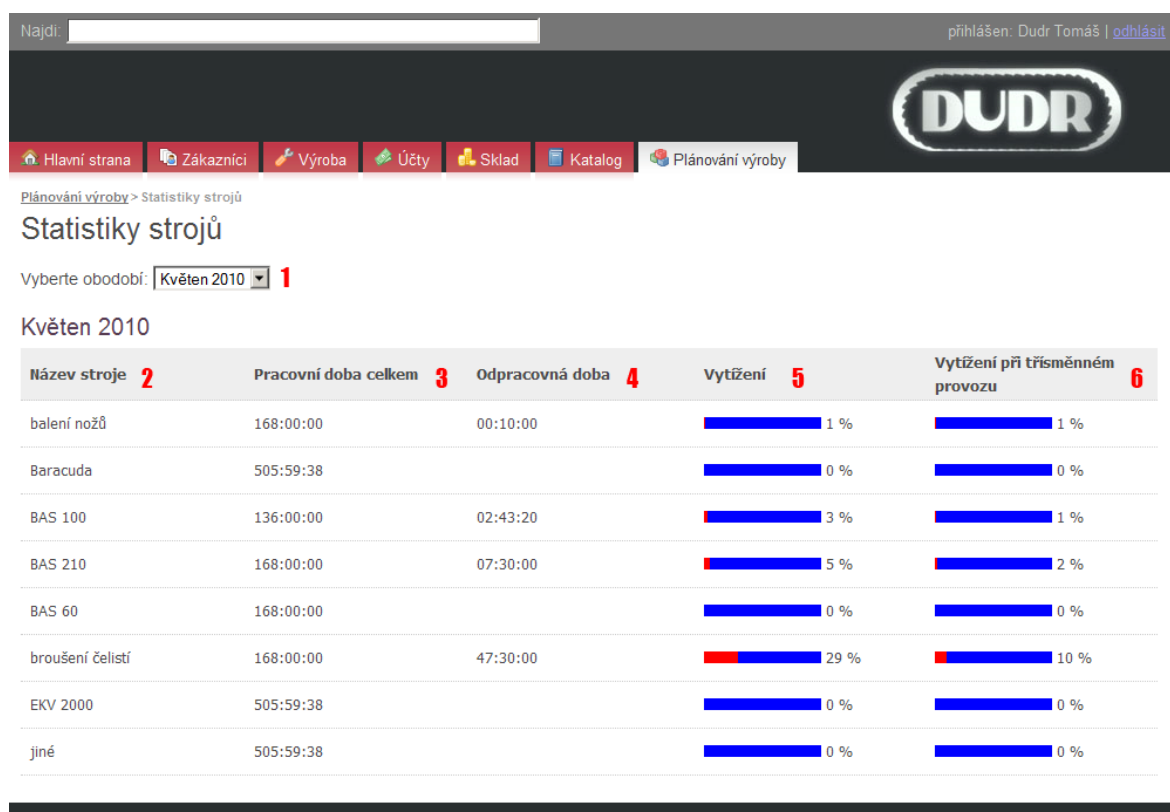
Obr. 13. Formulář pro nastavení pracovní doby strojů.

Vysvětlivky:

- 1) *Načíst výchozí pracovní doby.* Tlačítko pro načtení výchozí týdenní pracovní doby všech strojů do aktuálně zobrazeného pracovního týdne. Výchozí pracovní doba se zadává v sekci stroje (viz 8.4.2).
- 2) *Smazat pracovní doby.* Tlačítko pro smazání pracovních dob všech strojů z aktuálně zobrazeného pracovního týdne.
- 3) Tlačítko pro přechod mezi pracovními týdny.

- 4) *Stroje*. Seznam všech stávajících strojů.
- 5) Zobrazení názvu dne v pracovním týdnu a příslušného data. Vzhledem k tomu, že pracovní týden firmy začíná nedělí, prvním dnem týdne je proto taktéž neděle.
- 6) *Nastav volno*. Tlačítko pro nastavení volna všem strojům v daný den. Použitelné tehdy, je-li některý pracovní den státní svátek, popřípadě nastane-li jiná neočekávaná situace, kdy je potřeba výrobní provoz v daný den přerušit.
- 7) Kliknutím na časový údaj dne kteréhokoliv stroje můžeme jednoduše editovat pracovní dobu. Na výběr máme několik časů, podle toho zvolíme pracovní cyklus stroje, nebo nastavíme stroji volno.

8.4.5 Statistiky strojů



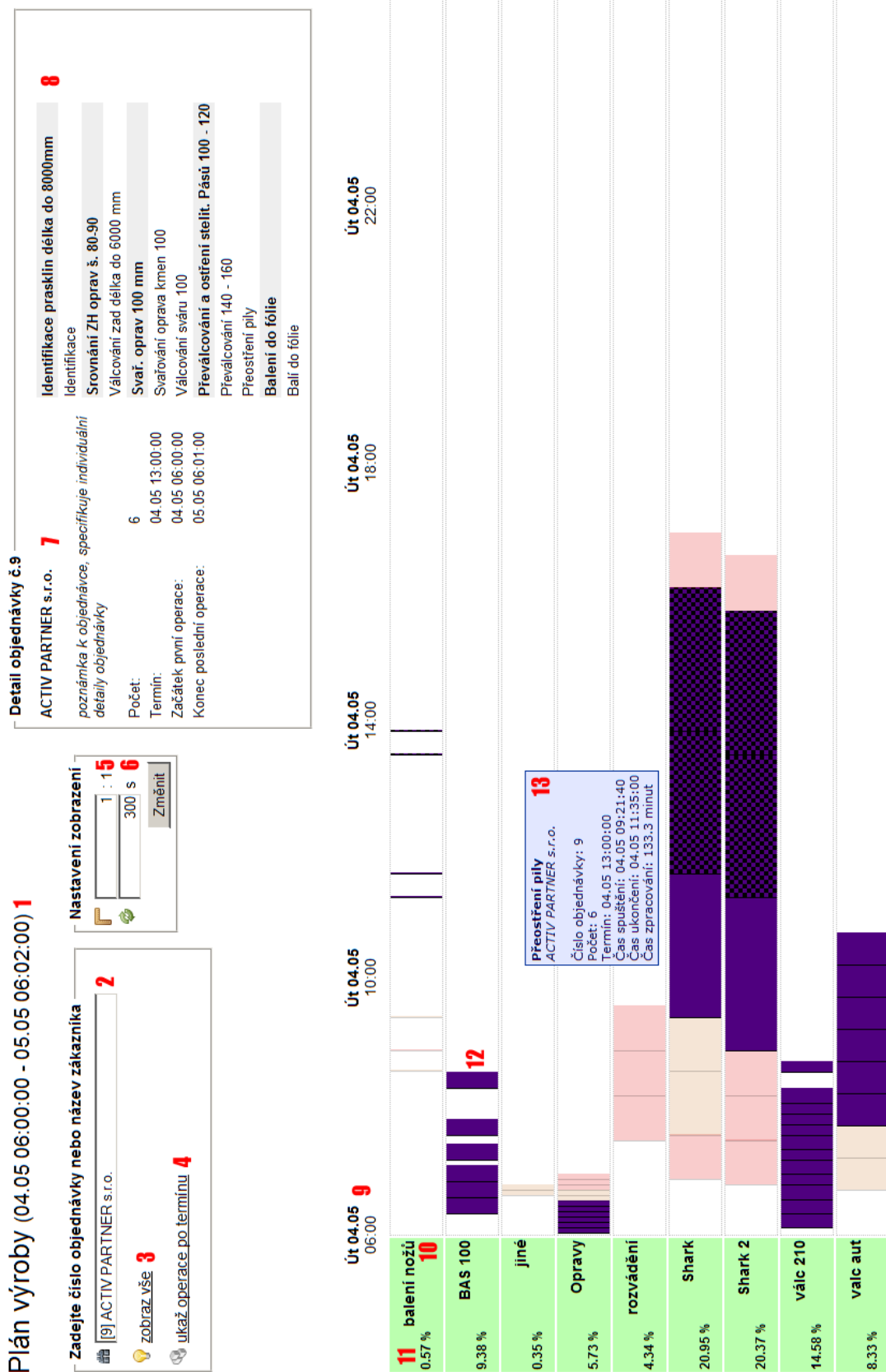
Obr. 14. Zobrazení měsíčních statistik využitelnosti strojů.

Vysvětlivky:

- 1) *Vyberte období*. Nabídka výběru období, pro které chceme zobrazit statistiku využitelnosti strojů.
- 2) *Název stroje*. Seznam všech stávajících strojů.

- 3) *Pracovní doba celkem.* Celková měsíční pracovní doba stroje, vypočítaná podle zadané pracovní doby v sekci Pracovní doby strojů (viz 8.4.4).
- 4) *Odpracovaná doba.* Celková měsíční odpracovaná doba stroje, vypočítaná podle časů všech operací, které se na daném stroji provedly.
- 5) *Vytížení.* Procentuální vyjádření poměru celkové měsíční pracovní doby a celkové měsíční odpracované doby.
- 6) *Vytížení při třísměnném provozu.* Procentuální vyjádření poměru teoretické celkové měsíční pracovní doby při třísměnném provozu a celkové měsíční odpracované doby. Teoretická pracovní doba při třísměnném provozu bere v potaz veškeré státní svátky a dny pracovního volna.

8.4.6 Plán výroby



Obr. 15. Plán výroby

Vysvětlivky:

- 1) *Plán výroby.* Zobrazení času začátku a konce plánu výroby (tj. začátek první operace a konec poslední operace).
- 2) Pole pro vyhledávání objednávek. Můžeme je vyhledávat buď podle čísla objednávky nebo podle názvu zákazníka. Po výběru hledané objednávky se její katalogové operace zvýrazní (respektive všechny ostatní operace změní svou průhlednost). Toto pole funguje jako „našeptávač“ (viz 8.3.1).
- 3) *Zobraz vše.* Tlačítko pro zobrazení všech operací. Použijeme tehdy, chceme-li zrušit výběr jedné objednávky.
- 4) *Zobraz operace po termínu.* Tlačítko pro zvýraznění všech operací, které přesáhly termín dokončení (čas dokončení operace je větší než termín dokončení operace).
- 5) Pole pro zadání měřítka zobrazení plánu výroby. Zadaný údaj představuje velikost jedné minuty v pixelech (tzn. kolik pixelů představuje v plánu výroby jedna minuta). Tento parametr můžeme volit v rozsahu 1 - 10.
- 6) Pole pro zadání obnovovací doby plánu výroby. Zadaný údaj představuje počet sekund, po kterém se automaticky aktualizuje okno plánu výroby.
- 7) *Detail objednávky.* Vybereme-li pomocí pole pro vyhledávání určitou objednávku, zobrazí se zde vybrané detaily objednávky (název zákazníka, poznámka k objednavce, počet kusů, termín dokončení, čas první operace, čas poslední operace).
- 8) *Detail objednávky.* Vybereme-li pomocí pole pro vyhledávání určitou objednávku, zobrazí se zde příslušné katalogové výrobky a k nim náležící katalogové operace v pořadí, jakém se postupně zpracovávají.
- 9) Časová osa. Zobrazuje datum, zkratku dne a čas.
- 10) Zobrazuje název stroje.
- 11) Zobrazuje aktuální vytížení stroje.
- 12) Katalogová operace. Každá katalogová operace představuje barevně označený obdélník (šířka obdélníku představuje čas zpracování operace, barva obdélníku představuje barevné rozlišení objednávek).

- 13) Detail katalogové operace. Po najetí kurzoru myši na určitou katalogovou operaci se zobrazí okno s detailem operace (popis katalogové operace, název zákazníka, číslo objednávky, počet kusů, termín dokončení, čas zahájení operace, čas ukončení operace, čas zpracování operace).

8.4.7 Nastavení plánování

Najdi: přihlášen: Dudr Tomáš | [odhlásit](#)

DUDR

Hlavní strana Zákazníci Výroba Účty Sklad Katalog Plánování výroby

Plánování výroby > Nastavení plánování

Nastavení plánování

Plánovací pravidlo: **1**

Výchozí měřítko v plánu výroby: : 1 (kolik pixelů v plánu výroby bude představovat 1 minuta) **2**

Výchozí obnovovací doba plánu výroby: sekund **3**

4

Obr. 16. Formulář pro editaci nastavení plánování.

Vysvětlivky:

- 1) *Plánovací pravidlo.* Nabídka s výběrem plánovacího pravidla. Toto pravidlo se použije v plánovacím algoritmu pro plánování operací na stroje. Můžeme zvolit následující pravidla:
 - *EDD.* Earliest due date (viz 3.3.1)
 - *LPT.* Longest processing time (viz 3.3.2)
 - *SPT.* Shortest processing time (viz 3.3.3)
 - *WSPT.* Weighted shortest processing time (viz 3.3.4)
 - *MSLACK.* Minimal slack (viz 3.3.5)
 - *ATC.* Apparent tardiness cost (viz 3.3.6)
 - *EECR.* Extended enhanced critical ratio (viz 3.3.7)
 - *DUDR.* (viz 3.3.8)
- 2) *Výchozí měřítko v plánu výroby.* Pole pro nastavení výchozího měřítka plánu výroby. Zadaný údaj představuje velikost jedné minuty v pixelech (tzn. kolik

pixelů představuje v plánu výroby jedna minuta). Tento parametr můžeme volit v rozsahu 1 - 10.

- 3) *Výchozí obnovovací doba plánu výroby.* Pole pro nastavení výchozí obnovovací doby plánu výroby. Zadaný údaj představuje počet sekund, po kterém se automaticky aktualizuje okno s plánem výroby.
- 4) *Ulož.* Tlačítko pro uložení provedených změn.

8.5 Terminálová část aplikace

Terminálová část aplikace běží na průmyslových panelových počítačích umístěných u strojů ve výrobních prostorách. Slouží k odepisování provedených operací dle plánu výroby.

The screenshot shows the terminal application interface. At the top, it displays the date and time: "Neděle 09. 05. 2010 | 16:53" and the user is logged in as "Dudr Tomáš".

On the left side, there is a sidebar menu with several items, each with a red number indicating its status or count:

- Zapsané činnosti **1**
- Shark (1/3) **2**
- lis 100
- valc aut
- Baracuda
- lis 210
- nože
- pěchování
- rozvádění
- EKV 2000
- BAS 100
- BAS 210
- Nezakázková čin...
- Přestávka **3**

The main area is divided into two sections:

Identifikace **4**

- 11 | AGROFOREST a.s. Identifikace prasklin délka do 6000mm **2ks** **5** -4 dny
- 9 | ACTIV PARTNER s.r.o. Identifikace prasklin délka do 8000mm **5ks** -5 dny
- 10 | IMET-CZ, spol. s r.o. Identifikace prasklin délka do 10000mm **3ks** včera

9 | ACTIV PARTNER s.r.o. detaily **6**

1.5.2010 12:38
 č.zakázky: 26; č.operace: 48
 termín: **4.5.2010 13:00** (-5 dní)
 způsob dopravy: nezadáno
 Identifikace (Opravy)
Iden. pras. délka do 8000mm

Buttons: **7** zahájit činnost, zapsat

Parametry

- délka: 8000
- rozteč/zuby: 10/800
- profil: topholz
- forma: 6,5 **8**
- šířka stelitu: 20
- typ zuby: Z3

Počty

- Σ Celkem: **5 ks**

Poznámka

- poznámka k objednávce, specifikuje individuální detaily objednávky

Obr. 17. Terminálová část aplikace.

Vysvětlivky:

- 1) Zapsané činnosti. Po kliknutí na toto tlačítko zobrazíme seznam naposledy zapsaných (tzn. provedených) katalogových operací.

- 2) Seznam dostupných strojů pro aktuálně přihlášeného pracovníka. Každý pracovník má způsobilost ovládat jen určité stroje. Po výběru konkrétního stroje zobrazíme seznam dostupných operací (bod 5) ke zpracování v pořadí, jaké bylo vypočítáno plánovacím algoritmem.
- 3) *Nezakázková činnost a přestávka*. První tlačítko slouží k provedení činností, které nejsou součástí objednávek. Druhé tlačítko slouží k nahlášení příchodu do práce, odchodu na přestávku, příchodu z přestávky a odchodu z práce.
- 4) Název katalogové operace.
- 5) Seznam dostupných katalogových operací, informace o počtu kusů a počtu dní do konce termínu dokončení objednávky.
- 6) Detailní informace o vybrané katalogové operaci.
- 7) *Zahájit činnost a zapsat*. Prvním tlačítkem zahájíme zpracování katalogové operace, druhým tlačítkem potvrdíme její dokončení a odepíšeme z plánu výroby.
- 8) Parametry objednávky, počet kusů a poznámka k vybrané katalogové operaci.

9 ANALÝZA STÁVAJÍCÍHO SYSTÉMU PRO PLÁNOVÁNÍ LOGISTIKY

9.1 Popis

Současný způsob práce s nástrojem pro plánování logistiky je následující.

Při zadávání objednávky do systému se tato objednávka zároveň ručně vloží do Google Kalendáře (dále jen kalendář) a jako datum se uvede termín dodání hotové objednávky zákazníkovi.

Každý den v týdnu je v kalendáři rozdělen na více částí. Tyto části znamenají určité trasy, na kterých se v ten konkrétní den budou rozvážet hotové objednávky (vždy jeden řidič a vozidlo přísluší k jedné trase). Do této konkrétní trasy a dne se tedy vloží nová objednávka. Objednávku musí mistr výroby v průběhu její realizace sledovat a pomocí barevných štítků jí nastavuje stavy (nová, snad se stihne, stihne se část, stihne se, pro opravy, nestihne se, hotovo).

Dále pak v části systému pro správu úkolů je nutné ručně vytvořit úkol. Každý úkol může obsahovat jednu či více objednávek. Podle hotovosti jednotlivých objednávek lze poté vytvářet jednotlivé plány rozvozu. Ty vytvoříme zadáním parametrů plánu výroby (datum rozvozu, vozidlo a zaměstnanec, nastavení času návštěvy prvního zákazníka) a přiřazením jednotlivých úkolů. Plány rozvozu fungují přes API rozhraní systému Google Maps a obsahují funkce jako výpočet nejkratší trasy dle zadaných úkolů, dále výpočet potřebného času pro projetí trasy či výpočet začátku jízdy tak, aby byl zaměstnanec v zadanou dobu u prvního zákazníka. Nakonec je možné celou trasu ze systému exportovat do mobilního GPS zařízení.

Jak je vidět, skoro všechny tyto úkony jsou neautomatizované a nutí mistra výroby věnovat spoustu času pouze sledováním jednotlivých objednávek, značit jejich stavy do kalendáře a vytvářet plány rozvozu na jednotlivé dny. Jediné, co je v této části zpracováno dobře a použitelně, je logistická část zajišťující generování plánů rozvozu do map.

9.2 Navržené řešení

Navržené řešení spočívá v odstranění nutnosti používání kalendáře a ve vytvoření podobné funkční komponenty v současném informačním systému. Jednotlivé objednávky by se pak automaticky řadily do plánů a tras rozvozů podle jejich aktuální hotovosti.

Tedy řešením je vytvořit novou komponentu typu kalendář a tuto propojit s plánem výroby. Tím, že se plánovací algoritmus spouští automaticky každou půlhodinu (a potažmo tedy i plán výroby je aktualizován s touto periodou), lze vcelku s velkou přesností určit termín dokončení objednávek a podle tohoto předpokládaného termínu dokončení poté objednávky automaticky přiřadit ke konkrétnímu dni rozvozu a trase. Použití nástroje pro vytváření plánů rozvozu, který je v současné době implementován pomocí API rozhraní Google Maps, by se tak zachovalo a pouze propojilo s nově vytvořenou komponentou typu kalendáře reagující na aktuální stav zpracování objednávek ve výrobě.

ZÁVĚR

Cílem diplomové práce bylo vytvořit nástroje pro správu katalogových informací a pro plánování a rozvrhování výroby, následně prozkoumat existující informační systémy dostupné na trhu, a podle analýzy funkčních a nefunkčních požadavků firmy DUDR – pilové pásy vybrat takový informační systém, který vyhovuje zadaným požadavkům. Dále pak navrhnout a naprogramovat uvedené nástroje do vybraného informačního systému, analyzovat stávající systém pro plánování logistiky a navrhnout jeho inovační řešení.

Před samotným návrhem a implementací těchto nástrojů bylo nutné získat potřebné teoretické znalosti týkající se plánování a rozvrhování výroby, identifikovat typ konkrétního plánovacího problému firmy a pomocí této identifikace určit metodu řešení plánovacího algoritmu. To vše bylo provedeno v teoretické části práce.

Praktická část byla zaměřena na analýzu existujících informačních systémů a její vyhodnocení, na návrh a vývoj obou požadovaných nástrojů a v neposlední řadě na rozbor systému pro plánování logistiky a jeho možné inovační řešení. Z úvodní analýzy existujících informačních systémů bylo zjištěno, že žádný ze zkoumaných produktů nevyhovuje zadaným požadavkům, a proto bylo nutné integrovat nově vyvíjené nástroje do stávajícího informačního systému používaného ve firmě. Zde se ukázalo, jak je důležité, spolu s programem, vytvářet taktéž programovou dokumentaci. Vzhledem k chybějící dokumentaci k tomuto systému bylo obtížné pochopit, jak celý systém funguje a značná část času byla věnována reverznímu inženýrství. Poslední kapitoly praktické části se tedy věnují tvorbě programové a uživatelské dokumentace.

Výsledkem diplomové práce jsou nové komponenty informačního systému vyhovující všem funkčním a nefunkčním požadavkům plynoucích z analýzy firemních procesů a z konzultací se zástupci firmy během tvorby projektu.

Spokojenost s výsledky této práce vyústily v další spolupráci s firmou. Konkrétně se bude jednat o realizaci návrhu inovačního řešení systému pro plánování logistiky uvedenou v poslední kapitole, a dále o vytvoření nástroje pro plánování pracovních směn.

CONCLUSION

The goal of this master's thesis was to create tools for managing products as well as the planning and scheduling of manufacturing, then to explore existing information systems accessible in the marketplace. According to analyses of functional and non-functional requirements of the company DUDR – pilove pasy, an information system was chosen that suits their requirements. Further design and implementation tools to the chosen information system, analyse the existing system for planning logistics and design its innovative solution.

Before design and implementation, these tools were necessary to get theoretical knowledge concerning planning and scheduling manufacturing, identifying concrete planning problems of the company and by the help of this identification determine methods of solving planning algorithms. That was performed in theoretical part of this work.

The practical part focused on the analysis of existing information systems and their evaluation, on design and development of both required tools and lastly on the analysis of systems for scheduling logistics and its possible innovative solutions. Opening analyses of the existing information systems ascertained that no surveyed product suited the engaged requirements, that is why it was necessary to integrate newly developed tools to the current information system used in the company. It is shown here how important it is in conjunction with programs to create program documentation too. With regards to missing documentation of this information system it was difficult to understand how the whole system worked and considerable time was devoted to reverse engineering. The last chapters of the practical part were then devoted to the creation of program and user documentation.

The resulting diploma thesis involves new components to information systems suitable to all functional and non-functional requirements resulting from the analysis of company processes and from counsel with deputies of the company during the project's creation.

Satisfaction with results of this work flow into subsequent cooperation with the company. Concretely, this will act as realization proposals for innovative solutions of the system for planning logistics mentioned at the last chapter and then about creating tools for scheduling work shifts.

SEZNAM POUŽITÉ LITERATURY

Monografie

- [1] ADAMS, J., BALAS, E., ZAWACK, D., The Shifting Bottleneck Procedure for Job Shop Scheduling. In *Management Science*. 3rd edition. [s.l.] : INFORMS, 1988. Focussed Issue on Heuristics. s. 391-401. Dostupný z WWW: <<http://www.jstor.org/stable/2632051>>.
- [2] BASL, J.; BLAŽÍČEK, R. *Podnikové informační systémy : Podnik v informační společnosti* . 2. vydání. Praha : Grada Publishing, 1998. Funkcionalita podnikových informačních systémů - ERP, s. 66. ISBN 978-80-247-2279-5.
- [3] BLAZEWICZ, J., et al. *Handbook on Scheduling : From Theory to Applications*. New York : Springer, 2007. 647 s. ISBN 978-3-540-28046-0.
- [4] BLAZEWICZ, J., et al. *Scheduling Computer and Manufacturing Processes*. 2nd edition. New York : Springer, 2001. 485 s. ISBN 978-3-540-41931-0.
- [5] BRUCKER, P. *Scheduling Algorithms*. 5th edition. New York : Springer, 2007. 371 s. ISBN 978-3-540-69515-8.
- [6] CHIANG, T. C.; FU, L. C. Solving the FMS scheduling problem by critical ratio-based heuristics and the genetic algorithm. *Proc. of IEEE Conference on Robotics and Automation*. 2004, vol. 3, s. 3131 – 3136.
- [7] CHIANG, T. C.; FU, L. C. Using Dispatching Rules for Job Shop Scheduling with Due Date-based Objectives. *International Journal of Production Research*. 2007, vol. 45, s. 3245 - 3262. Dostupný také z WWW: <<http://www.informaworld.com/10.1080/00207540600786715>>.
- [8] CHLEBOVSKÝ, V. CRM : Od šanonu, pastelek a diáře k sofistikovanému e-systému. *SystemOnLine : Zpravodajský portál časopisu IT Systems* [online]. 2002, 3, [cit. 2010-05-26]. Dostupný z WWW: <<http://www.systemonline.cz/clanky/crm-od-sanonu-pastelek-a-diare-k-sofistikovanemu-e-systemu.htm>>.

- [9] ČAPATÝ, L., *Rozvrhování výroby pomocí metod lokálního hledání*. Brno, 1998. 54 s. Diplomová práce na Fakultě strojní Vysokého učení technického v Brně na Ústavu automatizace a informatiky. Vedoucí diplomové práce Jiří Dvořák.
- [10] MAIER, P., *Moderní metody rozvrhování výroby*. Brno, 2004. 90 s. Disertační práce na Fakultě strojní Vysokého učení technického v Brně na Ústavu automatizace a informatiky. Vedoucí diplomové práce Jiří Dvořák.
- [11] MORTON, T.; PENTICO, D. W. *Heuristic Scheduling Systems : With Applications to Production Systems and Project Management*. New York : John Wiley & Sons, 1993. 720 s. ISBN 978-0471578192.
- [12] PARKER, R. G. *Deterministic Scheduling Theory*. 1st edition. London : Chapman and Hall, 1996. 296 s. ISBN 978-0412996818.
- [13] PINEDO, M. L. *Planning and Scheduling in Manufacturing and Services*. 2nd edition. New York : Springer, 2009. 537 s. ISBN 978-1-4419-0909-1.
- [14] PINEDO, M. L. *Scheduling : Theory, Algorithms, and Systems*. 3rd edition. New York : Springer, 2002. 678 s. ISBN 978-0-387-78934-7.
- [15] PINEDO, M. L.; SESHADRI S. *Handbook of Industrial Engineering : Technology and Operations Management*. 3rd edition. New York : John Wiley & Sons, 2001. Chapter 63, Scheduling and Dispatching, s. 1718-1740. ISBN 978-0-471-33057-8.
- [16] RUDOVÁ, H. *PA167 Rozvrhování* [online]. Brno : Fakulta informatiky, 2009. 559 s. Průsvitky k přednášce Rozvrhování. Masarykova Univerzita, Fakulta informatiky. Dostupné z WWW: <<http://www.fi.muni.cz/~hanka/rozvrhovani/prusvitky/all.pdf>>.
- [17] SOCHOR, J., Údržba softwaru. *Zpravodaj ÚVT MU* [online]. 2006, 6, 3, [cit. 2010-05-26]. s. 15-20. Dostupný z WWW: <<http://www.ics.muni.cz/zpravodaj/articles/61.html>>. ISSN 1212-0901.

Internetové zdroje

- [18] *About Gantt Charts* [online]. 2009 [cit. 2010-05-26]. Gantt Charts. Dostupné z WWW: <<http://www.ganttchart.com/>>.

- [19] AJAX. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 11.10.2005, last modified on 1.5.2010 [cit. 2010-05-26]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/AJAX>>.
- [20] *BlueErp* [online]. c2007 [cit. 2010-05-24]. Dostupné z WWW: <<http://open.mitija.com/blueerp/>>.
- [21] *Dibi is Database Abstraction Library for PHP 5* [online]. c2008 [cit. 2010-01-27]. Dostupné z WWW: <<http://dibiphp.com/cs/>>.
- [22] *Dolibarr Home* [online]. c2009 [cit. 2010-05-24]. Dostupné z WWW: <<http://www.dolibarr.org/>>.
- [23] Framework. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 22.5.2006, last modified on 2.5.2010 [cit. 2010-05-26]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Framework>>.
- [24] *Free Web Based NolaPro : Accounting and business software* [online]. c2010 [cit. 2010-05-25]. Dostupné z WWW: <<http://www.nolapro.com/index-5.html>>.
- [25] *FrontAccounting* [online]. c2010 [cit. 2010-05-25]. Dostupné z WWW: <<http://frontaccounting.net/wb3/>>.
- [26] *Naval Research Logistics : an International Journal* [online]. c2010 [cit. 2010-05-26]. Wiley. Dostupné z WWW: <<http://eu.wiley.com/WileyCDA/WileyTitle/productCd-NAV.html>>.
- [27] *Nette Framework* [online]. c2008 [cit. 2010-05-27]. Dostupné z WWW: <<http://nettephp.com/cs/>>.
- [28] Open Source Initiative. *Open Source Initiative* [online]. c2010 [cit. 2010-05-26]. The Open Source Definition. Dostupné z WWW: <<http://www.opensource.org/docs/osd>>.
- [29] *PhpDocumentor : The complete documentation solution for PHP* [online]. c2009 [cit. 2010-05-14]. Dostupné z WWW: <<http://www.phpdoc.org/>>.
- [30] *PhreeBooks Accounting : Web Based Open Source Accounting (ERP)* [online]. c2009 [cit. 2010-05-25]. Dostupné z WWW: <<http://www.phreebooks.com/>>.

- [31] *Prototype JavaScript framework : Easy Ajax and DOM manipulation for dynamic web applications* [online]. c2007 [cit. 2010-05-15]. Dostupné z WWW: <<http://prototypejs.org/>>.
- [32] *Tine 2.0 : Open Source Groupware and CRM* [online]. c2009 [cit. 2010-05-25]. Dostupné z WWW: <<http://www.tine20.org/>>.
- [33] *UzERP : An open source ERP solution* [online]. c2009 [cit. 2010-05-25]. Dostupné z WWW: <<http://www.enterprisegroupwaresystem.org/>>.
- [34] *Vtiger Open Source CRM* [online]. c2010 [cit. 2010-05-25]. Dostupné z WWW: <<http://vtiger.com/>>.
- [35] *W3C Document Object Model* [online]. c2005 [cit. 2010-05-26]. Document Object Model . Dostupné z WWW: <<http://www.w3.org/DOM/Overview>>.
- [36] *WebERP : What Is webERP ?* [online]. c2010 [cit. 2010-05-25]. Dostupné z WWW: <<http://www.weberp.org/>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

- AJAX** Asynchronous JavaScript and XML - obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání. Na rozdíl od klasických webových aplikací poskytují uživatelsky příjemnější prostředí, ale vyžadují použití moderních webových prohlížečů.
- API** Application Programming Interface - označuje v informatice rozhraní pro programování aplikací.
- B2B** Business-to-business - obecně lze B2B definovat jako obchodní vztah mezi dodavatelem na jedné straně a odběratelem, který dodané produkty dále využije ve svém podnikání na straně druhé. Důležité je, že na straně odběratele nefiguruje koncový spotřebitel.
- CRM** Customer Relationship Management - informační systémy zajišťující správu dat souvisejících s péčí o zákazníky.
- DOM** Document Object Model - platformě a jazykově-neutrální rozhraní, které dovoluje programům a skriptům dynamicky zpřístupnit a aktualizovat obsah, strukturu a styl dokumentů.
- ERP** Enterprise Resource Planning - aplikace, které představují softwarová řešení užívaná k řízení podnikových dat a pomáhající plánovat celý logistický řetězec od nákupu přes sklady po výdej materiálu, řízení obchodních zakázek od jejich přijetí až po expedici, včetně plánování vlastní výroby a s tím spojené finanční a nákladové účetnictví i řízení lidských zdrojů.
- Framework** softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji.
- Drag & Drop** V překladu znamená „táhni a pusť“. Pomocí této technologie můžeme jednoduše přetahovat objekty. Uchytíme nějaký objekt, tažením přesuneme na jinou pozici a upustíme na cílové pozici, vše pomocí kursoru myši.

- MySQL MySQL je multiplatformní databáze. Komunikace s ní probíhá pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.
- Nette Nette je výkonný framework pro tvorbu webových aplikací v PHP 5.
- PHP Hypertext Preprocessor - skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což lze využít při tvorbě webových aplikací.

SEZNAM OBRÁZKŮ

<i>Obr. 1. Vývojový diagram plánovacího algoritmu.</i>	56
<i>Obr. 2. E-R diagram části databáze.</i>	66
<i>Obr. 3. Náhled na vygenerovanou programovou dokumentaci.</i>	69
<i>Obr. 4. Formulář pro přihlášení do aplikace.</i>	71
<i>Obr. 5. Hlavní okno aplikace.</i>	71
<i>Obr. 6. Zobrazení katalogových výrobků a jejich parametrů.</i>	73
<i>Obr. 7. Formulář pro editaci katalogového výrobku.</i>	74
<i>Obr. 8. Zobrazení katalogových operací a jejich parametrů.</i>	76
<i>Obr. 9. Formulář pro editaci katalogové operace.</i>	77
<i>Obr. 10. Hlavní menu modulu plánování výroby.</i>	79
<i>Obr. 11. Zobrazení strojů a jejich parametrů.</i>	81
<i>Obr. 12. Formulář pro vytvoření nového stroje.</i>	82
<i>Obr. 13. Formulář pro nastavení pracovní doby strojů.</i>	84
<i>Obr. 14. Zobrazení měsíčních statistik vytíženosti strojů.</i>	85
<i>Obr. 15. Plán výroby.</i>	87
<i>Obr. 16. Formulář pro editaci nastavení plánování.</i>	89
<i>Obr. 17. Terminálová část aplikace.</i>	90

SEZNAM TABULEK

<i>Tab. 1. Struktura tabulky katalogovy_vyrobek.</i>	53
<i>Tab. 2. Struktura tabulky katalogova_operace.</i>	53
<i>Tab. 3. Struktura tabulky operace_k_vyrobku.</i>	54
<i>Tab. 4. Struktura tabulky skladova_karta.</i>	54
<i>Tab. 5. Struktura tabulky sklad.</i>	55
<i>Tab. 6. Struktura tabulky stroje.</i>	60
<i>Tab. 7. Struktura tabulky stroje_druh.</i>	60
<i>Tab. 8. Struktura tabulky stroje_pracovni_doba.</i>	61
<i>Tab. 9. Struktura tabulky stroje_pracovni_doba_default.</i>	62
<i>Tab. 10. Struktura tabulky plan_vyroby.</i>	63
<i>Tab. 11. Struktura tabulky plan_vyroby_nastaveni.</i>	64

SEZNAM PŘÍLOH

P I Obsah přiloženého CD

PRÍLOHA P I: OBSAH PŘILOŽENÉHO CD

CD obsahuje následující adresářovou strukturu:

/Phpdocumentor_CZsablona/	český překlad šablony Phpdocumentoru
/program/SQL/	SQL skript pro vytvoření databázových tabulek
/program/www/	zdrojové kódy webové aplikace
/programova_dokumentace/	programová dokumentace ve formě HTML stránek
/text/	text diplomové práce (formáty DOC a PDF)
/uzivatelska_dokumentace/	uživatelská dokumentace ve formátu PDF