

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

EVIDENCE A ROZBOR DÉLEK TRVÁNÍ PRACOVNÍCH ÚKONŮ PŘI VÝROBĚ FOREM

EVIDENCE AND TIME CONSUMPTION ANALYSIS OF INDIVIDUAL OPERATIONS DURING
MOULDING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ ŠKRABAL

VEDOUCÍ PRÁCE
SUPERVISOR

ING. RADEK POLIŠČUK, PH.D.

BRNO 2008

ZADÁNÍ ZÁVĚREČNÉ PRÁCE

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2007/08

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Škrabal Ondřej

který/která studuje v bakalářském studijním programu

obor: Aplikovaná informatika a řízení (3902R001)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Evidence a rozbor délek trvání pracovních úkonů při výrobě forem

v anglickém jazyce:

Evidence and time consumption analysis of individual operations during moulding

Stručná charakteristika problematiky úkolu:

Úkolem studenta je návrh a realizace aplikace pro objektivní časomíru jednotlivých pracovních úkonů při výrobě forem ve společnosti zadavatele a pro následnou analýzu získaných dat.

Cíle bakalářské práce:

Cílem práce je návrh aplikace v prostředí Microsoft Visual C# 2008, určené pro sběr časoměrných údajů při průmyslové výrobě. Skutečný průběh pracovního procesu bude v klíčových bodech zaznamenáván On-line čtečkou čárových kódů, přičemž k získaným časosběrným datům bude před zápisem do databáze kromě časového razítka doplňován kód příslušné operace. Manažerský modul aplikace pak zadavateli umožní analyzovat případné rezervy ve výrobě a objektivizovat doposud jen odhadované náklady, účtované jednotlivým obchodním partnerům.

Seznam odborné literatury:

Kent, J.: Visual C# 2005 bez předchozích znalostí, Computer Press, Brno 2007, ISBN: 978-80-251-1584-8.

Bayer, J.: C# 2005 - velká kniha řešení. Computer Press, Brno 2007, ISBN: 978-80-251-1620-3.

Nagel, C.: C# 2005 - Programujeme profesionálně, Computer Press, Brno 2007, ISBN: 80-251-1181-4.

Vedoucí bakalářské práce: Ing. Radek Poliščuk, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2007/08.

V Brně, dne 11. 12. 2007

L.S.



doc. RNDr. Ing. Míloš Šeda, Ph.D.
Ředitel ústavu



doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

LICENČNÍ SMLOUVA

(na tomto místě je vloženo licenční ujednání)

ABSTRAKT

Práce se zabývá ekonomicko-výrobním problémem v konkrétním strojírenském provozu. Řešení je založené na čtení čárových kódů, programovacím jazyku C# a datovém uložišti MS SQL Server 2005 Express Edition. Přináší reálné výsledky hodinových cen výrobních operací daného provozu založené na analýze nasbíraných dat, a vedle toho doporučené postupy k budoucímu navýšení zisku.

ABSTRACT

This work deals with economical production problem in particular mechanical service. The solution is based on capture of barcode, C# programming language and database engine MS SQL Server 2005 Express Edition. It provides the user with real results of per hour cost analysis in given process and it also offers recommendations for increasing profit based on analysis of measured data.

KLÍČOVÁ SLOVA

Identifikace, čárový kód, snímač čárových kódů, databázový systém, C#, systém sledování výroby, časová a ekonomická analýza výroby.

KEYWORDS

Identification, barcode, bar-code scanner, database system, C#, system of production monitoring, time and economic analysis of production.

PODĚKOVÁNÍ

Touto cestou bych chtěl poděkovat vedoucímu bakalářské práce Ing. Radkovi Poliščukovi, Ph.D. za věcné rady při vypracování této bakalářské práce. Dále pak svým rodičům za podporu, zvláště svému otci Ing. Jaromírovi Škrabalovi za rady při realizaci projektu, a také všem zaměstnancům firmy Octopus, spol. s r.o.

OBSAH:

	Zadání závěrečné práce.....	3
	Licenční smlouva.....	5
	Abstrakt.....	7
	Poděkování.....	9
1	Úvod.....	13
2	Nástroje pro návrh systému sledování výroby.....	15
2.1	Identifikační prvky.....	15
2.2	Čárové kódy	16
2.2.1.	Vlastnosti čárových kódů.....	16
2.2.2.	Základní prvky čárového kódu.....	16
2.2.3.	Typy nejčastěji používaných čárových kódů.....	17
2.3	Snímače čárových kódů.....	19
2.4	Vybraný snímač čárových kódů a vybraný typ čárového kódu.....	19
2.5	Programovací prostředky.....	22
2.5.1.	Struktura aplikačního řešení.....	22
2.5.2.	Databázové systémy.....	23
2.5.3.	Vývojová platforma.....	24
3	Návrh systému sledování výroby.....	29
3.1	Koncept klient/server.....	29
3.2	Datová analýza.....	31
3.3	Aplikace pro sběr dat.....	32
3.3.1.	Výpočet pravděpodobnosti vzniku chyby při sběru dat.....	34
3.3.2.	Funkční analýza aplikace pro sběr dat.....	34
3.4	Aplikace pro zadávání a analýzu dat.....	38
3.4.1.	Funkční analýza aplikace pro zadávání a analýzu dat.....	38
4	Ekonomická a časová analýza.....	45
4.1	Časová analýza.....	45
4.2	Ekonomická analýza.....	46
5	Závěr.....	53
	Seznam použité literatury.....	55

1 ÚVOD

Malé a střední firmy, ať už strojírenského, elektrotechnického, farmaceutického, dřevařského, sklářského nebo i jiného průmyslu, stále více využívají analýzy časové a ekonomické i jiné úspornosti své vlastní výroby pomocí automatizovaných systémů používající různých druhů identifikačních zařízení, počítačového hardwaru a softwaru. Díky výsledkům takové „real-time“ analýzy jsou manažeři firmy schopni rychle a pružně rozhodovat o objemu a ceně zakázek, časové a ekonomické vytíženosti strojů, nástrojů, lidských zdrojů, výrobků, apod. Navíc se jejich rozhodování neredukuje jen na pouhou znalost zpětné vazby na minulé zakázky, ale i na reflexi zakázek, které jsou v daném provozu zrovna ve výrobě. Tímto se může stát sledování své vlastní výroby účinnou zbraní na poli konkurenceschopnosti. Nejenže může zefektivnit samotnou výrobu, ale může být dobrým vodítkem pro vedení firmy tam, kde poptávka převyšuje nabídku a správné a rychlé rozhodnutí o počtu přijatých nebo odmítnutých zakázek může znamenat navýšení zisku.

Práce vznikla v reakci na požadavek zobjektivizovat dosud jen odhadované náklady na výrobu vstřikovacích a vyfukovacích forem plastikářského potažmo strojírenského průmyslu. Práce může oslovit jedince a skupiny z řad malých a středně velkých firem, institucí, spolků apod. Může sloužit jako vodítko při řešení stejného nebo podobného problému.

Firma Octopus, spol. s r.o. se zabývá dimenzováním a designem výrobků, konstrukcí a výrobou forem, materiálovým inženýrstvím, plastikářskou technologií, apod. Výroba vstřikovacích a vyfukovacích forem nepatří mezi sériové výroby, ale specializované či kusové. Každá forma je z hlediska konstrukce a výroby originálem. Zavedení normované výroby je tedy finančně náročnější a pro tak malou firmu (asi 10 zaměstnanců) se ekonomicky nevyplatí. Z tohoto důvodu jsem byl jako student ÚAI osloven k návrhu finančně nenáročného řešení, které by pomohlo objektivizovat dosud jen odhadované náklady daného provozu (Obr. 1).



Obr. 1 Stanice pro sledování výroby.

Možnou variantou řešení je nahlédnout do účetnictví firmy a pomocí některého tabulkového procesoru typu „Excel“ spočítat náklady firmy a vyhodnotit případné ceny forem. Toto řešení ale nevyhovuje všem požadavkům zadavatele. Bylo zvoleno komplexnější řešení založené na interní špionáži (též interní strategické zpravodajství). Doby výrobních činností (frézování, broušení, vrtání, soustružení apod.) na jednotlivých dílech formy a další data mohou být zaznamenávána některým čtecím zařízením a příslušná aplikace je může zpracovávat do podoby použitelných informací. Nespornou výhodou tohoto řešení je možnost rozšiřovat zpravodajství na další části jako je docházkový systém, účetnictví, online obchod, apod., nicméně tyto položky měla firma již vyřešené

jinými způsoby nebo je nepožadovala.

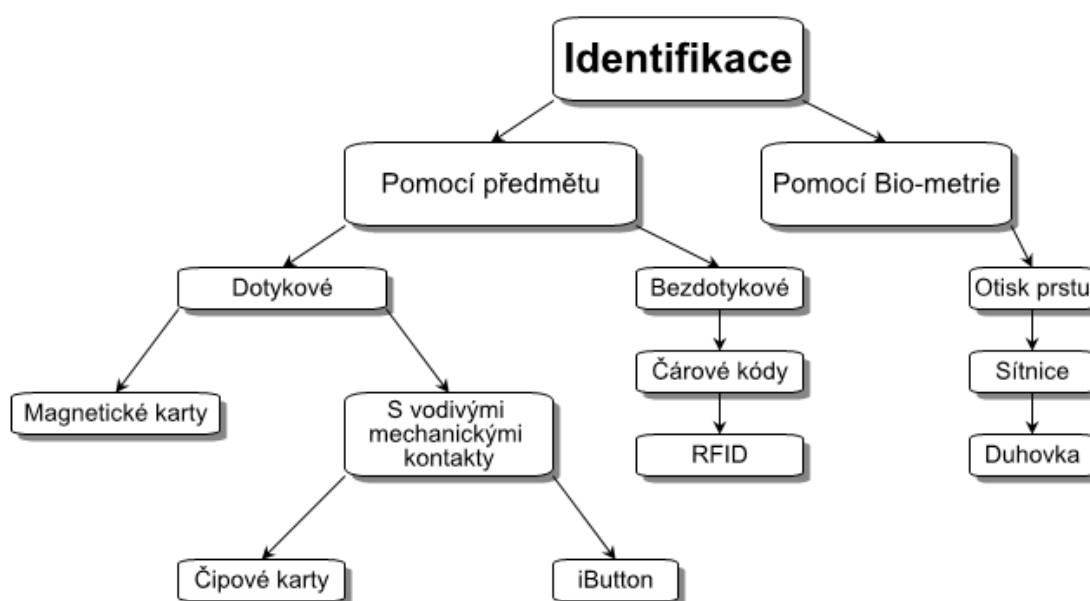
Práce je členěna logicky do kapitol tak, aby čtenář nejdříve získal potřebné teoretické znalosti o nasazených technologiích, a poté je předvedeno konkrétní řešení. Druhá kapitola tedy představuje možnosti v oblasti čtecích zařízení a také možnosti v oblasti softwarového vybavení. Kapitola se podrobněji zabývá čárovými kódy a čtečkami čárových kódů. Dále se zde čtenář seznámí okrajově s jazykem C#, platformou .NET Framework a datovými uložišti. Třetí kapitola je úvodem do uživatelského prostředí, které zajišťuje obsluhu snímání a zadávání dat do databáze. Čtvrtá kapitola je věnována časové a ekonomické analýze s přihlédnutím na vytvořenou aplikaci. V závěru práce jsou představeny návrhy na řešení vzniklého problému výroby.

2 NÁSTROJE PRO NÁVRH SYSTÉMU SLEDOVÁNÍ VÝROBY

Kapitola nabízí různé možnosti, jak řešit návrh systému sledování výroby. Systematicky podává teoretický základ pro vybrané technologie řešeného projektu v konkrétní firmě. Nejdříve jsou představeny fyzické prvky systému, dále na ně navazují prvky softwarového vybavení.

2.1 Identifikační prvky

Identifikačních prostředků, které se využívají k automatické identifikaci, tedy k získávání dat bez použití kláves, je celá řada. Je možné si vybrat z různých druhů identifikačních prvků, které můžeme dělit podle několika hledisek, jak znázorňuje Obr. 2. Cílem této práce není podrobná studie o identifikačních zařízeních, ale jsou zde zmíněny výhody a nevýhody jednotlivých technologií vzhledem k nasazení do strojírenské výroby, a to s přihlédnutím na řešený projekt.



Obr. 2 Dělení podle způsobu identifikace. [6]

V první řadě je potřeba určit výchozí podmínky, ke kterým bude celý problém vztahován. Záměrem bylo ve strojírenské dílně monitorovat:

- a) výrobní dílce
- b) výrobní operace
- c) osoby v závislosti na výrobních operacích a dílcích

Identifikace pomocí Bio-metrie se pro náš řešený problém nehodí. Díky své technologii* se úzce specializuje na identifikaci živočichů, popřípadě osob. Ve strojírenské praxi je naprosto nevhodná vzhledem k přítomnosti prachu, oleje a jiných nečistot na prstech pracovníků. Používají se optoelektrické, kapacitní, teplotní a elektroluminiscentní snímače otisků prstů nebo snímače sítnice i duhovky. Výhoda (u snímačů otisků prstů), kterou by tato technologie v čistém prostředí přinesla, je její dobrá cenová dostupnost. Dále není potřebné vytvářet speciální přenosné médium, protože je přímo tvořené předmětem zájmu, v našem případě osobou a jeho částmi těla.

Další možností je využití některé technologie z řady dotykové identifikace. Mohly by to být

* Bio-metrie - identifikace/verifikace osob podle jedinečných fyziologických znaků člověka

magnetické karty, čipové karty nebo iButton. Opět by se tato řešení dala využít spíše jen pro identifikaci osob. Další nevýhodou je poměrně vysoká cena nosného média, která se pohybuje v řádu desítek až stovek korun. Toto řešení je spíše vhodnější pro provozy, kde nehrozí mechanické poškození média přenášejícího informace. Vhodné jsou jeho různé aplikace v bankovním, například v podobě platebních karet, sim karet mobilních telefonů nebo bývá použit u docházkových systémů větších firem nebo institucí. Pro strojírenskou výrobní praxi by obstál iButton díky své mechanické odolnosti, ale jen těžko si lze představit řešení spojení tohoto média s nějakým výrobním dílcem nebo výrobní operací.

Poslední skupinou, která nám na poli identifikace zbývá, je bezdotyková technologie zastoupená čárovými kódy a RFID. Obě tyto technologie čárové kódy a RFID je možné uplatnit pro účely našeho záměru. Obecně je možné říci, že při volbě jednoho nebo druhého systému bude záležet na konkrétních podmínkách daného strojírenského provozu. Hlavní výhodou RFID je, že nemusí být nutně přímá viditelnost pro čtení a zapisování do RFID tagů (čipů). Dále pak odolnost a variabilita média, mobilita, rychlost pořízení informace, snížení chybovosti, atd. U čárových kódů je větší nebezpečí poškození a chybového čtení a nutnost přímé viditelnosti. Přesto má tato technologie stále své nesporné výhody jakou je například nízká cena média. Pro tisk čárových kódů postačí obyčejná laserová nebo inkoustová tiskárna, takže náklady na médium přenášející informaci jsou téměř zanedbatelné. V našem případě to byl hlavní důvod při rozhodování, kterou technologii nasadit do provozu, protože nutnost přímé viditelnosti nebyl aspekt, který by byl v našem případě na obtíž. Čárové kódy tak byly zvoleny hlavně pro jejich cenovou dostupnost. Tato kapitola čerpá z [6].

2.2 Čárové kódy

Čárové kódy se skládají z tmavých čar a světlých mezer natištěných svisle ve vodorovném směru vedle sebe, přičemž jsou čteny snímači čárových kódů. Čáry i mezery mají definovanou určitou šířku a jsou nositelem informace. Snímače čárových kódů vyzařují ve většině případů červené světlo, které je tmavými čarami pohlcováno a světlými mezerami odraženo. Odražené světlo je pak transformováno na posloupnost elektrických impulsů s různou šířkou a následně porovnáno s tabulkou příslušných kombinací. Pokud je při porovnávání nalezena příslušná posloupnost, je označena za odpovídající znakový řetězec. Skupiny čar na začátku a na konci čárového kódu mají speciální úlohu – slouží k synchronizaci čtecího zařízení, které podle nich generuje signál start/stop. Na základě technické specifikace se pro správný chod čtení doporučuje ochranné světlé pásmo bez potisku za a před synchronizačními čarami. Data, která je možné zapsat a pak následně číst z čárového kódu, mohou obsahovat téměř cokoli: číslo zaměstnance, číslo výrobku, číslo výrobní operace, jméno osoby, apod. [1]

2.2.1. Vlastnosti čárových kódů

Jednou z hlavních výhod čárových kódů je přesnost. Pokud budeme zadávat data ručně vznikne chyba každé třísté zadání, při použití čárových kódů bude chyba zaznamenána někdy kolem miliontého snímání dat. Navíc mohou být tyto chyby téměř eliminovány na základě kontrolní číslice, která je vypočítána z předchozích číslic kódu.

Rychlost je další neméně důležitou výhodou, která hraje při rozhodování, jak snímat a spravovat vyráběné položky, hlavní roli. Snímače čárových kódů jsou nejméně třikrát rychlejší než ruční zadávání dat.

Čárové kódy mohou být natištěny na jakýkoli materiál rezistentní proti vlhkosti, kyselinám, výkyvům počasí, atd., dále je můžeme přizpůsobit velikosti výrobku. Flexibilita je tedy další důležitou vlastností.

Díky všem těmto atributům můžeme předpokládat vyšší produktivitu a efektivnost práce. Tato kapitola čerpá z [1] a [5].

2.2.2. Základní prvky čárového kódu

- X - šířka modulu – jedná se o nejmenší možnou šířku čáry nebo mezery
- R - světlé pásmo – jeho šířka je nejméně 2,5 mm, ale doporučuje se desetinásobek

šířky modulu

- H - výška kódu – jedná se o svislý rozměr pásu kódu, doporučená délka je 10% délky pásu pro ruční čtení, 20% délky pásu pro čtení se skenerem, minimum je stanoveno na 20 mm a výjimku tvoří čárový kód EAN, který má doporučenou 75% délku pásu.
- L - délka kódu - jedná se o délku pásu od první značky Start až po poslední značku Stop, nepřičítá se světlé pásmo
- C - kontrast – je definován jako rozdíl jasu odrazu pozadí a jasu odrazu čáry vydělený jasem odrazu pozadí. Pro dobře čitelný kód by měl být větší než 0,7. Tato kapitola čerpá z [2].

2.2.3. Typy nejčastěji používaných čárových kódů

Existuje velké množství čárových kódů, do kterých můžeme zakódovat data. V závislosti na typu najdeme kódy, do kterých je možné zakódovat jen číslice, kódující jen znaky jako např. 1, 8, 9, atd. nebo kódy umožňující kódovat jen základní abecedu. Existují ale také alfanumerické typy, které kódují číslice i abecední znaky. Našli bychom i typy čárových kódů, které dokáží kódovat jen sadu 127 nebo 255 znaků z ASCII tabulky. Obecně se pak typy čárových kódů liší délkou a strukturou kódu, způsobem kódování a v neposlední řadě typickým oborem použití. Pokud bychom čárové kódy porovnávaly z hlediska délky, zjistili bychom, že některé mají proměnnou délku a jednotlivé znaky jsou poskládány za sebou nebo pod sebou podobně jako slova ve větě. Jiné mají pevnou délku jako např. EAN 8 s 8 číslicemi nebo EAN 13 s 13 číslicemi. Počty typů kódů, které se v současnosti používají, jsou v desítkách, proto uvádím jen jejich zlomek z celkového počtu, a to ještě s přihlédnutím na případné využití v oblasti průmyslu. [4]

EAN 13 a EAN 8

Tyto kódy se v největší míře používají v obchodní síti, kde se tímto způsobem označuje zboží. Zakódovány mohou být číslice 0 až 9 a každá číslice je reprezentována dvěma čarami a dvěma mezerami. Jak už bylo uvedeno, obsahují 8 číslic pro EAN 8 nebo 13 pro EAN 13 číslic (Obr. 3). Každý stát zapojený do mezinárodního sdružení European Article Numbering (EAN) může tento čárový kód používat. První 2-3 číslice kódu určují stát, ČR má číslo 859. Dalších 4 - 6 číslic určuje dodavatele nebo výrobce, následující jsou rezervovány pro zboží. Poslední číslice má funkci kontrolní, umožňuje ověřit správnost dekódování. [3], [5]



Obr. 3 Čárový kód EAN 13. [3]

UCC/EAN 128, Code 128

Rodina těchto kódů umožňuje kódovat větší množství informací, než tomu je u EAN 8 a 13. Jsou to alfanumerické kódy proměnné délky. Z kódů můžeme vyčíst datum balení, datum výroby, maximální nebo minimální trvanlivost, délku, šířku, hmotnost, objem, plochu, číslo výrobku, atd. Tyto kódy nacházejí uplatnění v průmyslu a označují se jimi obchodní nebo logistické jednotky.

Pomocí kódů UCC/EAN 128 můžeme kódovat až 96 znaků z ASCII tabulky a dále 11 funkčních znaků. Každý údaj zakódovaný v tomto kódu má svůj vlastní aplikační identifikátor (AI), který přesně stanovuje typ údaje. Každý znak je tvořen čarami nebo mezerami o celkové šířce 11 modulů. UCC/EAN 128 tvoří speciální standardizovanou verzi Code 128.

Do Code 128 (Obr. 4) je možné zakódovat všechny tisknutelné znaky standardu ASCII, tedy 128 znaků. Vždy tři čáry a tři mezery znázorňují jeden znak a celková šířka znaku je opět 11 modulů. Šířka čáry nebo mezery může být 1, 2, 3 nebo 4 moduly. Tato část práce čerpá z [3] a [5].



Obr. 4 Čárový kód CODE 128. [3]

CODE 39

Tento kód je alfanumerický (Obr. 5), je schopen zakódovat písmena A až Z, číslice 0 až 9 a sedm speciálních znaků. Každý znak je tvořen pěti čarami a čtyřmi mezerami, je sestaven z 3 širokých a 6 tenkých čar nebo mezer. Znaky start a stop nám zde reprezentuje znak „hvězdičky“. CODE 39 nepodporuje malé písmena, ty jsou automaticky na vstupu přeměněna na velké písmena. Kód byl navržen v roce 1974 pro průmysl a obchod. Díky své spolehlivosti (k chybě dojde asi po přečtení 30 miliónů znaků) se rozšířil do automobilového, zbrojařského, a dalších odvětví průmyslu, používá se mimo jiné např. ve zdravotnictví. [3], [5]



Obr. 5 Čárový kód CODE 39. [3]

Kódy typu 2 z 5

Čárový kód Industrial 2/5 byl navržen pro interní průmyslové aplikace v roce 1968 firmou Identicon Corporation. Tento kód dovoluje poměrně velkou hustotu zápisu, může to být až 8 znaků na 1 cm a jeho délka se může měnit v závislosti na hustotě zápisu nebo počtu znaků. Kóduje pouze numerické číslice 0 až 9 a je tvořen znakem start a stop. Každý znak tohoto kódu je znázorněn pěticí čar, tři jsou úzké a dvě široké. Zajímavostí je, že světlé pásy nenesou informaci. Šířku světlého pásu se doporučuje nastavit rovnu šířce modulu X. Šířka širokého a úzkého elementu odpovídá poměru 3:1. Příklad kódování je v Tab. 1, přičemž 0 představuje úzký a 1 široký element. Tyto kódy jsou vhodné pro nekvalitní tisk, podklad, ztížené podmínky čtení nebo špatně přijímající barvu díky širokému tolerančnímu pásmu. [2]

Znak	C1	C2	C3	C4	C5
0	0	0	1	1	0
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0
Start	1	1	0		
Stop	1	0	1		

Tab. 1 Kódovací tabulka. [2]

2.3 Snímače čárových kódů

Snímače čárových kódů se nejčastěji dělí na snímače obsahující prvek CCD (tento typ snímače byl použit v našem projektu) a na laserové.

U laserových snímačů se čte jedním nebo více paprsky, které jsou emitovány laserovými diodami. Jejich nespornou výhodou jsou lepší dekodovací schopnosti a schopnost snímat čárové kódy z většího odstavu.

Snímače CCD (Charge-Coupled Device) jsou jednodušší. Jednorozměrný obraz dopadající na plochu snímače je vyhodnocen řádkovým senzorem. Modernější digitální snímače obsahují maticový senzor CCD. Tyto přístroje pak pracují na podobném principu jako digitální fotoaparát. Obraz s kódovým obrazcem je vyfotografován a uložen do paměti. Dále pak zpracován přímo přístrojem nebo předán do počítače, kde je identifikován pomocí metody rozpoznávání obrazců. K CCD technologii více v použité literatuře [10] a [12].

Snímače čárových kódů můžeme dělit také podle míry inteligence a autonomnosti, komunikace s nadřazeným zařízením (s PC nebo PLC), kolik má komunikačních kanálů nebo jaké rozhraní používá (např. RS-232, RS-485, USB, Ethernet). Dalším důležitým dělením je rozdělení snímačů na stacionární a mobilní snímače. Tato kapitola čerpá také z [4].

Při volbě snímačů čárových kódů jsou důležité další parametry:

- maximální snímací vzdálenost
- možnosti orientace čteného kódu
- mechanická odolnost
- teplotní odolnost
- ostatní

2.4 Vybraný snímač čárových kódů a vybraný typ čárového kódu

Pro účely snímání čárových kódů byl vybrán mobilní přístroj s obchodním označením „Čtečka ČK CCD-MT9062, 90mm, RS232 černá“ (Obr. 6) a typ čárového kódu CODE 128 popsán v kapitole 2.2.3. Na trhu je možné najít velké množství druhů snímačů čárových kódů. Liší se svými parametry a v neposlední řadě svou cenou.



Obr. 6 ČK CCD-MT9062, 90mm, RS232 černá. [7]

Parametry vybraného snímače:

CCD čtečka čárových kódů se sériovým rozhraním RS232 (RS232 viz. použitá literatura [9]):

SPECIFIKACE (převzato z publikace [7]):

Šířka čtecího pole: 120 mm

Hloubka čtecího pole: 0 - 50 mm

Rozlišení: 4mils (0,1 mm)

PSC (kontrastní poměr): minimálně 0,45

Rychlost snímání: 45 snímků/sec

Zdroj světla: 660 nm červená LED (viditelné světlo)

Snímač: CCD 2088 pixelů

Napájecí napětí: 5 V \pm 5 %

Proudový odběr dekódování: - 65 mA \pm 10 mA

Proudový odběr klidový: - 35 mA \pm 5 mA

Rozměry: (v x š x h) 185,5 x 89 x 27 mm

Hmotnost bez kabelu: 120 g \pm 5 g

Kabel: výměnné kabely, délka 210 cm

Pracovní teplota: 0°C až 50°C

Skladová teplota: -20°C až 60°C

Relativní vlhkost: 20% až 80%, bez kondenzace

Shoda dle normy FCC class A a CE (EN55022 CLASS B, En50082-1)

Čtecí režimy: se spouští, nepřetržitý (bez spouště)

Rozhraní: RS232

Dekódování č.kódů: UPC-E, EAN-8, EAN-13, ISBN, ISSN, Interleave 2 of 5,

Code 11, Code 32, Code 39, Code 93, Code 128,

Codabar, MSI/Plessey, BC-412, JAN, China Post Code

EPP31901 Napájecí redukce 5V pro CCD skener - záslepka

EPP31902 Napájecí redukce - kabel pro CCD skenery

Nastavení snímače čárových kódů

Pro správnou funkci snímače čárových kódů bylo potřeba provést nastavení parametrů tohoto snímače. K tomuto účelu nám slouží Programovací příručka [8] přiložená výrobcem k zařízení. Jak píše samotný výrobce v této příručce, před nastavováním snímače je potřebná určitá obeznámenost s problematikou čárových kódů. Toto obeznámení bylo provedeno v rámci předchozích podkapitol. Samotná konfigurace se provádí pomocí typizovaných čárových kódů, které je možné najít v příručce výrobce. Pokud např. potřebujeme nastavit snímač pro čtení čárových kódů CODE 128, v příručce najdeme odpovídající čárový kód a jednoduše ho čtečkou načteme, čímž proběhne nastavení.

Postup nastavení snímače pro náš projekt: (zmíněny jen ty nejdůležitější nastavení)

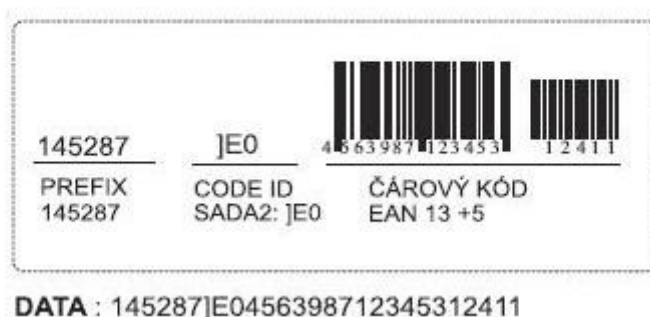
- 1) Obnovení továrního nastavení: DEFAULT
- 2) Typ počítače: PC – AT
- 3) Typ rozhraní (Obr. 7): RS232



Obr. 7 Náhled na výběr typu rozhraní. [8]

- 4) Přenosová rychlost: 9600 kb.
- 5) Režim čtení:
TRIGGER MODE – LED se rozsvítí po stisknutí spouště a zhasne po uvolnění spouště nebo po načtení čárového kódu.
- 6) Nastavení přesnosti: 8 (1 až 9).
- 7) Prefix a surfix*:
Prefix „#“, uživatelem zvolený znak nebo řetězec znaků (max 16), který skener posílá před načteným čárovým kódem. Surfíx „\$“, uživatelem zvolený znak nebo řetězec znaků (max 16), který skener posílá po načtení čárového kódu.
- 8) Typ čárového kódu (Obr. 8):

PŘÍKLAD : 1. Prefix: 145287,
2. Code ID: předdefinovaná sada2,
3. Typ čárového kódu : EAN 13+5



Obr. 8 Příklad struktury konkrétního čárového kódu. [8]

V příručce jsou definovány jednotlivé typy čárových kódů, každý čárový kód je opatřen

* Přípustné znaky jsou z ASCII tabulky uvedené v příručce

svým CODE ID. Dále jsou předdefinovány pro každý kód dvě sady (Obr. 9), přičemž současně lze použít jen jednu sadu identifikátorů typu čárového kódu. CODE ID je možné nastavit i vlastní, kdyby znaky předdefinované v sadě1 nebo sadě2 nevyhovovaly. Byl zvolen CODE 128 a ponechána sada1 a tímto bylo nastaveno CODE ID: „K“.

VÝBĚR SADY IDENTIFIKÁTORŮ TYPU KÓDU					
TYP ČÁROVÉHO KÓDU	SADA 1	SADA 2	TYP ČÁROVÉHO KÓDU	SADA 1	SADA 2
MSI	O	JM0	CODABAR	N	JF0
EAN 8	S	JE0	DELTA Code	G	
UPC-E	E	JE0	LABEL Code	C	
UPC-A	A	JE0	UK PLESSY	P	JPO
EAN 13	F	JE0	MATRIX 2 OF 5	Y	JX0
Code 93	L	JG0	FULL ASCII Code 39	D	JA0
Code 11	J	JH0	STANDARD Code 39	M	JA0
TELEPEN	U	JB0	IATA 2 of 5	R	JR0
EAN 128	T	JC1	INDUSTRIAL 2 OF 5 (Code 2 of 5)	V	JS0
Code 128	K	JC0	China Post Code (Toshiba Code)	H	JX0
Code 32 (Code 39 PARAF)	B	JX0	INTERLEAVED 2 OF 5	I	JJ0

Obr. 9 Přehled přednastavených identifikátorů k jednotlivým čárovým kódům. [8]

9) Délka kódu: min. délka (5), max. délka (48).

10) Další nastavení:

Příručka poskytuje širokou škálu nastavení, z důvodů rozsahu zde nejsou uvedeny. [8]

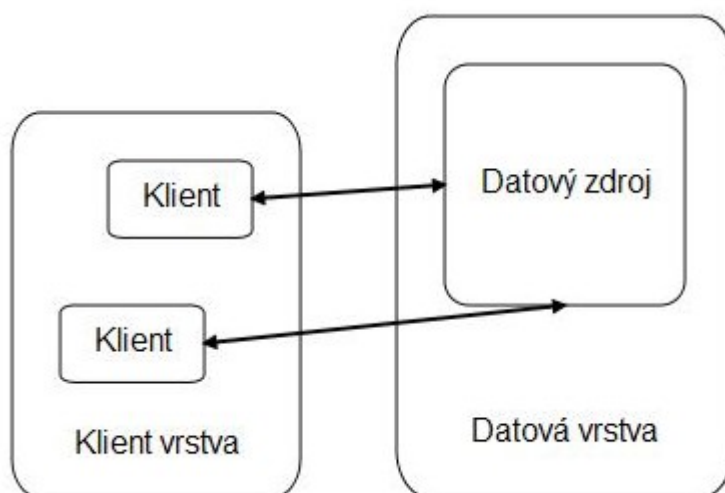
2.5 Programovací prostředky

Podkapitola představuje jak možnosti řešení pro návrh aplikace v některém z programovacích jazyků, tak možnosti pro ukládání dat v některém z datových úložišť. Podrobněji se zabývá vybraným řešením, tedy službou MS SQL Server 2005 Express Edition, jazykem C# a vývojovým prostředím Microsoft Visual C# 2008 Express Edition. Předvedena je také struktura aplikačního řešení v našem projektu.

2.5.1. Struktura aplikačního řešení

Pro náš konkrétní případ postačí dvouvrstvá architektura aplikačního řešení. Tato struktura se skládá z klientské a datové vrstvy, jak je patrné z Obr. 10. Klienti obsahují většinu aplikační logiky. Tato logika pracuje nad datovým zdrojem, kterým může být souborový systém, informační systém, webová služba, obecně jiná aplikace nebo tak, jak je to v našem případě, kde je zdrojem dat relační databáze. Klienti používají dotazovací jazyk SQL pro práci s daty uloženými v databázi.

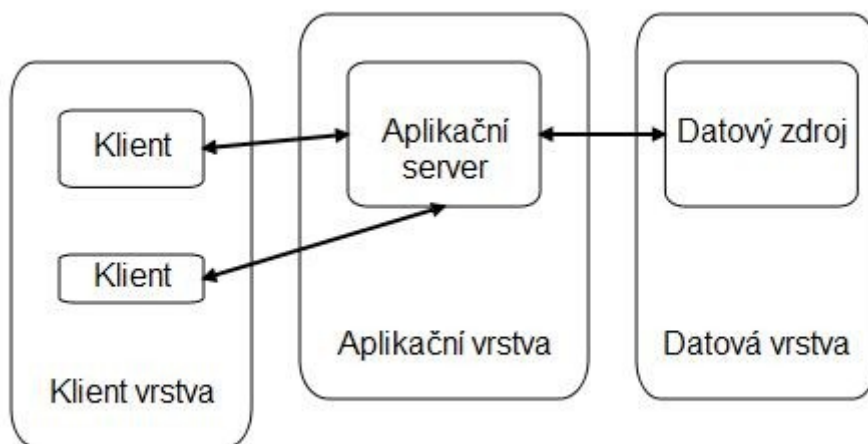
Při řešení komplexnějších problémů, jako je spojení docházkového a skladového systému, průmyslové špiónáže vlastní výroby a v neposlední řadě internetového obchodu nebo firemního informačního systému, dvouvrstvá architektura nestačí. Výkonové nároky na klientské počítače a nároky datových přesunů na samotnou datovou vrstvu jsou neudržitelné, proto se využívá modelu tří vrstev. Takto specifikovaná architektura obsahuje střední vrstvu (middle tier).



Obr. 10 Dvouvrstvá architektura. [11]

Třívrstvá architektura, tak, jak je ji možné vidět na Obr. 11, se dělí na klientskou, aplikační a datovou vrstvu. Tímto se klientským počítačům uleví od zátěže, jelikož je značná část jejich aplikační logiky přesunuta na aplikační server (aplikační vrstva). Díky tomuto přesunutí je možné snadnější sdílení aplikační logiky a lepší správy a dostupnosti. Dále se tímto redukuje datový přenos, který nově probíhá hlavně mezi aplikačním serverem a datovým zdrojem. Aplikační server může nabídnout řadu služeb mezi které patří rozložení zátěže nebo výpadku. Tímto byli klienti v podstatě ochuzeni o aplikační logiku a začalo se jim říkat „tenčí klienti“. Naopak klienti obsahující aplikační logiku si vydobyli název „tlustí klienti“. Samozřejmě tenčí klienti nějakou aplikační logiku obsahují, ale ta je oproti tlustým klientům zanedbatelná.

Naše dvě vytvořené aplikace budou tedy tlustými klienty v kontextu dvouvrstvého modelu dále jen klienti. Kapitola čerpá z [11].



Obr. 11 Třívrstvá architektura. [11]

2.5.2. Databázové systémy

Skládají se z databáze a ze systému řízení báze dat. To znamená, že databázový systém obsahuje jednak fyzicky a logicky uspořádaná data, ale také softwarové prostředky umožňující přístup a manipulaci s uloženými daty. Pro definici datových struktur vznikly programové nástroje typu DDL (Data Definition Language) a pro manipulaci s daty vznikly nástroje typu DML (Data Manipulation

Language). U objektových databází se používají nástroje typu ODL (Object Definition Language). Pokud bychom chtěli rozdělit databázové systémy podle aktuálních trendů, použili bychom dělení na databáze relační, objektové, postrelační a XML* databáze. Liší se jak v programovém přístupu k datům, tak v postupu při vytváření logického členění tabulek, respektive dat. Do našeho konkrétního projektu byl zvolen relační model. V praxi je tento model stále více používán díky své jednoduchosti a rychlosti. Další významné dělení databázových systémů je podle jejich velikosti:

- malé databáze jako jsou PC Fand, Paradox, dBase, apod. Tyto databáze se používají většinou na lokálních PC a jsou často využívány jen jedním uživatelem.
- střední databáze jako jsou MS Access a Visual FoxPro atd.. Používají se jak pro jedno PC, tak pro větší počet počítačů v síti. Lze zde najít některé rysy velkých databází.
- velké databáze jako jsou Oracle, Sybase, MS SQL Server, Caché, IBM DB2, Informix, MySQL, atd. Tyto databáze jsou již komplexními databázovými systémy v pravém slova smyslu. Obsahují RDBMS (Relation Data Base Management systém) tedy systém řízení báze dat. Navíc zajišťují vysoký výkon, bezpečnost a v neposlední řadě zpracování velkého množství dat. Takto velké databázové systémy pracují na principu klient/server a to znamená, že samotný databázový stroj (databáze) je umístěna na databázovém serveru (výkonný počítač). Přístup k tomuto serveru je zajišťován na základě klientských aplikací běžících na jednotlivých PC v síti.

V rámci projektu byl použit MS SQL Server 2005 Express Edition (podrobněji v kapitole 2.5.4.) vzhledem k jednoduché návaznosti na Visual Studio, jazyk C# a pracovního prostředí .NET Framework. Tato kapitola čerpá z [13].

2.5.3. Vývojová platforma

Pro tvorbu klienta, který bude tvořit rozhraní mezi uživatelem a databází je možné použít některý z objektově orientovaných jazyků podporujících přístup k datům v databázi: Javu, C#, VB nebo C++, PHP, Delphi, Python i jiné programovací jazyky. Při rozhodování, který programovací jazyk použít pro napsání aplikace, bude záležet na konkrétních znalostech architektů, programátorů, stejně jako firemních záměrech. Pro práci s relačními databázemi se používá standardizovaný strukturovaný dotazovací jazyk SQL (Structure Query Language).

Při výběru jazyka a pracovního prostředí pro náš projekt bylo vycházeno z několika předpokladů:

- interní projekt v průmyslové výrobě
- finanční dostupnost, nejlépe podpora studentů
- dobrá podpora výukových materiálů
- možnost pozdějšího provázání s webovým rozhraním
- programátorovy dosavadní znalosti některého OOP jazyka – spíše jen teoretické základy filosofie OOP

Vzhledem k tomu, že dosavadní znalosti programátora byly spíše teoretické, nebylo nutno vázat se na některý OOP programovací jazyk. Bylo možné vybrat jazyk, který bude vyhovovat uvedeným předpokladům a bude dostatečně kopírovat filosofii objektově orientovaného programování.

První možností je skriptovací jazyk PHP, který patří k nejrozšířenějším nástrojům k vytváření webových aplikací. Některé výhody tohoto nástroje jsou: snadná dostupnost výukových materiálů,

* XML – rozšiřitelný značkovací jazyk (z anglického Extensible Markup Language)

velká podpora pro komunikaci s MySQL, možnost návaznosti na web. Nicméně tento jazyk nebyl zvolen vzhledem k orientaci PHP hlavně na webové aplikace, dále menší podpoře frameworků, horší podpoře OOP a v těchto směrech neucelené koncepci.

Pro návrh a implementaci větších informačních systémů bude vhodné řešení založené na specifikaci J2EE (Java Enterprise Edition) vycházející ze skupiny firem jako jsou Sun Microsystems, IBM, Oracle, HP a dalších, nebo řešení založené na pracovním prostředí .NET firmy Microsoft. Výhodou Javy bude její možná použitelnost na jakémkoliv OS s JVM (Java Virtual Machine), naproti tomu nevýhodou .NET je jeho možné použití jen na OS Windows nebo Linux v rámci Mono* projektu. Obě řešení jsou ale srovnatelná. Technologie založená na Javě nabízí otevřenější přístup a lepší možnost volby při rozhodování, který databázový nebo operační systém použít. Naproti tomu platforma .NET lépe odráží nejnovější přístupy a trendy. Ucelenost technologie a podpora výukových materiálů je srovnatelná. Práce se nebude pouštět do hodnotících soudů, která platforma či specifikace je lepší, protože vždy záleží na konkrétní situaci.

Byl zvolen .NET a jazyk C#. Tato kapitola se dále zabývá tímto pracovním prostředím o něco podrobněji, stejně jako vývojovým prostředím Microsoft Visual C# 2008 Express Edition, a také službou MS SQL Server 2005 Express Edition - podrobnosti o tomto produktu byly zařazeny až nyní, vzhledem k výběru prostředí .NET a jazyka C#. Tato kapitola čerpá z [14] a [15].

Pracovní prostředí .NET Framework

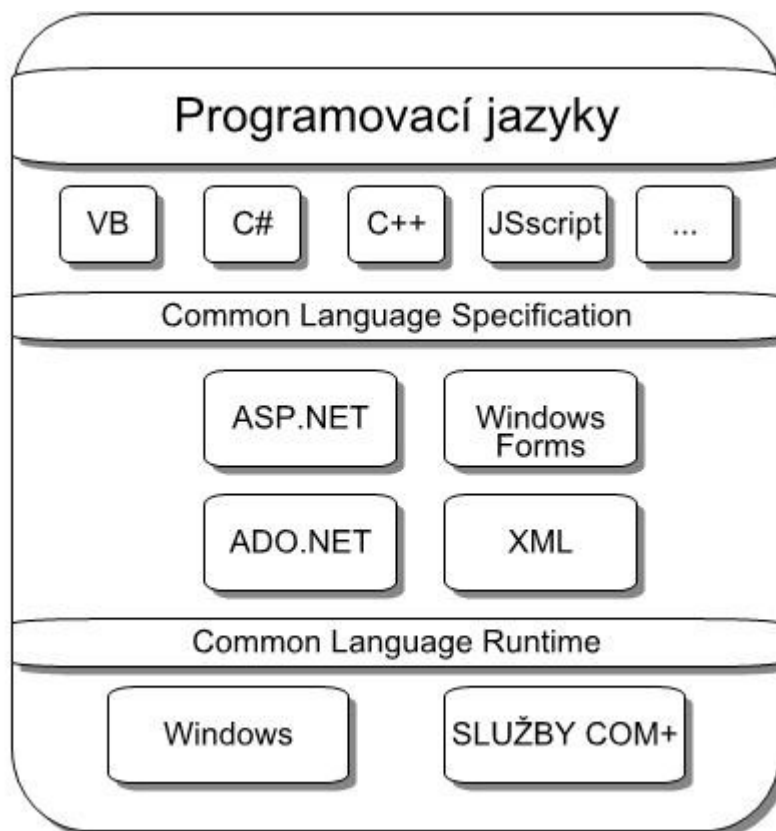
Prostředí .NET Framework je knihovna tříd, funkcí, atd., rozsáhlá asi jako Windows API*. Zajišťuje podobné možnosti jako Windows API, ale navíc tyto možnosti rozšiřuje na snadnější přístup k databázím, připojení k internetu nebo zprostředkování webových služeb. Je to také operační prostředí tzv. .NET runtime, kde jsou programy spuštěny. Toto operační prostředí zajišťuje další obslužné mechanismy jako je správa spuštěných vláken, podpůrné služby atd. Nicméně je potřeba říci, že operační prostředí .NET není operační systém, ale je to vrstva vložená mezi operační systém Windows (spolu s rozhraním Windows API) a jiné aplikace. Tato vrstva zajišťuje moderní objektově orientovaný přístup.

Dále bude zjednodušeně vysvětlena architektura prostředí .NET, která je patrná z Obr. 12. Základem pracovního prostředí .NET je společné prostředí zpracování označené jako CLR (Common Language Runtime) nebo je také možné používat označení operační prostředí .NET. Spuštěný kód pod CLR se musí řídit určitými pravidly definovanými specifikací .NET a říká se mu tzv. spravovaný kód (managed code). Než je tedy kód (napsaný v jazyce C#, VB nebo jiném) spuštěn, dochází k překladu tohoto kódu do jazyka MSIL (Microsoft Intermediate Language) - dále jen IL - což je podobný jazyk bajtovému kódu jazyka Java založený na číselných kódech. Teprve pak je takto přeložený kód (spravovaný kód) přeložen pomocí CLR modulu do kódu specifického pro danou platformu.

Z popsaného postupu vyplývá několik vlastností. Je to možná nezávislost prostředí .NET na platformě. Přeložený soubor s bajtovými instrukcemi můžeme umístit na jakoukoli platformu. Teprve za běhu dochází ke konečné fázi překladu. Tato koncepce je ale zatím spíše jen teoretická vzhledem k tomu, že pracovní prostředí .NET je zatím pouze pro OS Windows a z části pro Unix v rámci projektu Mono, jak již bylo zmíněno v kapitole 2.5.3. Prostředí používá technologii JIT (Just - In - Time). Kód není přeložen celý, ale jen když je potřeba. Dále překladač JIT při překladu ví, na jakém procesoru bude konečný spustitelný kód spuštěný a může ho optimalizovat pro daný procesor nebo instrukce jeho strojového kódu. Poslední vlastnost, která zde bude uvedena, je interoperabilita programovacích jazyků. To může být volně vysvětleno tak, že do IL kódu může být přeložen kód v C# a zároveň v VB nebo jiném. Tyto jazyky vedle sebe budou bez problémů existovat a kód IL bude správně funkční. Je však nutné upozornit na to, že aby mohlo dojít ke správné interoperabilitě programovacích jazyků, musí být dodrženy určité společné rysy, které určuje specifikace společného jazyka CLS (Common Language Specification).

* Mono - otevřený projekt v počátcích sponzorovaný firmou Novell, aby byl vyvinut opensource – vývojářská platforma pro UNIX verzi Microsoft .NET

* API - application programming interface, tedy obecně rozhraní pro programování aplikací



Obr. 12 Zjednodušená architektura .NET Framework. [14]

CLS je podmnožinou nástrojů prostředí .NET Framework a množinou standardů určujících minimální požadavky pro programovací jazyky, jejich překladače a programátory. Nebudou zde uvedeny všechny tyto základní požadavky, ale jen ty nejdůležitější. Jsou jimi objektová orientace, využití rozhraní, přísné rozlišování mezi hodnotovými a referenčními (odkazovými) typy, přísná typizace dat a další. Je třeba zdůraznit, že je možné psát kód, který nebude s CLS plně kompatibilní. Výsledný IL, pak nemusí být úplně interoperabilní s jinými jazyky. Poslední služba, o které se tato kapitola zmiňuje v rámci pracovního prostředí .NET, je automatická správa paměti (garbage collector). V prostředí .NET se programátor nemusí starat o uvolňování paměti v kódu aplikace, ale zajišťuje to za něj právě automatická správa paměti, čímž se myslí program, který se stará o čištění paměti. Podrobnosti je možné najít v použité literatuře, z které tato kapitola také čerpá [20].

Pracovní prostředí .NET Framework poskytuje opravdu obrovské množství tříd zajišťujících nejrůznější funkce. Takový výčet funkcí není cílem této práce, proto zde nebude uveden.

Jazyk C#

Jazyk C# staví na objektově orientovaném návrhu. Je to jazyk vytvořený od samotných základů zcela od znova, ale je speciálně navržen pro pracovní prostředí .NET Framework. Pokud je správně používán v tomto kontextu, pak je možné dosáhnout vysoké efektivity práce. Podrobnosti o syntaxi a sémantice jazyka se můžete dozvědět v některé ze současně používaných publikací [16], [20].

Záměrem projektu bylo pracovat s daty uloženými v databázi. V prostředí .NET Framework existuje rozhraní ADO.NET, které obsahuje množinu tříd speciálně připravených pro práci s daty a komunikací s databázovými systémy. Práce s ADO.NET v jazyce C# je rychlá a intuitivní.

Microsoft Visual C# 2008 Express Edition

Jedná se o vývojové prostředí pro jazyk C#. Tento produkt je zcela zdarma a je primárně určený pro studenty. Vyznačuje se jednoduchostí a snadnou zvládnutelností. Tato verze je určena

začátečnickům a hodí se pro návrh jednoduchých aplikací. Tvorba a distribuce takto vytvořených aplikací není licenčně omezena. [17]

MS SQL Server 2005 Express Edition

Tato databázová služba vychází z MS SQL Serveru 2005, má některá omezení, je však zdarma. Maximální velikost databáze (souboru s příponou .mdf) je omezena na 4GB. Dále je možné využít jen jednoho procesoru a 1GB operační paměti. Mezi další omezení patří vynechání některých služeb jádra SQL Serveru. Tato omezení jsou vzhledem k velikosti našeho projektu akceptovatelná. [19]

Pro snadnější správu serveru, databáze a zdrojů bylo použito Microsoft SQL Server Management Studio Express, které má opět podobnou licenční politiku jako předchozí produkty. Pro přímou správu zabezpečení povrchu, přístupů a konfigurace sítě byly použity nástroje MS SQL Serveru a to SQL Server Surface Area Configuration a SQL Server Configuration Manager. Podrobnější informace o práci s těmito nástroji jsou uvedeny v kapitole 3.

3 NÁVRH SYSTÉMU SLEDOVÁNÍ VÝROBY

Primárním požadavkem, ze kterého tato práce vznikla, bylo zobektivizovat dosud jen odhadované náklady na výrobu vstřikovacích nebo vyfukovacích forem strojírenské výroby firmy Octopus, spol. s r.o. se sídlem v Březolupech, v okrese Uherské Hradiště.

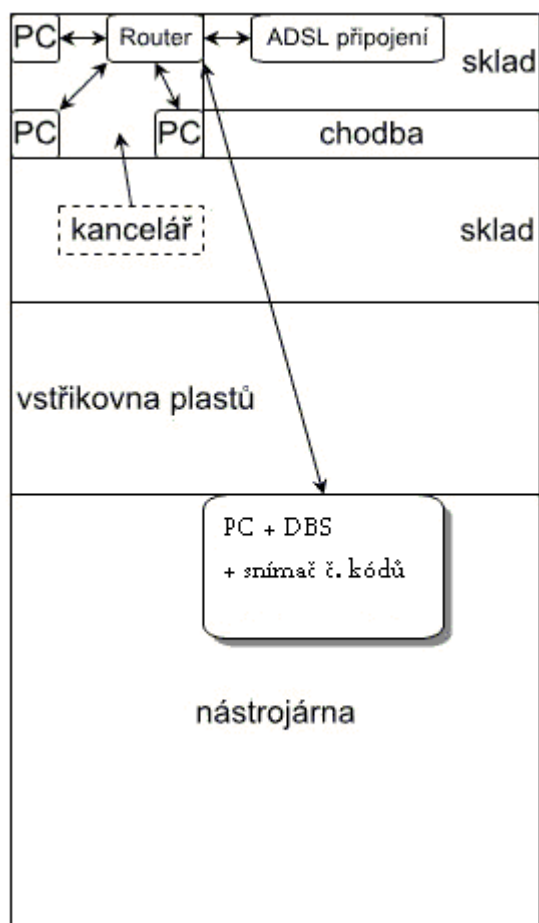
Návrh systému, který by tento požadavek splnil, začal od výběru identifikačního zařízení, přes konzultaci s vedením firmy o jejich představách, co by měl systém všechno sledovat a reflektovat, sestavení logiky datového skladu, návrh sestavení sítě na základě konceptu klient/server, až po realizaci samotných aplikací zajišťujících sběr, zadávání a analýzu dat.

Díky teoretické části by čtenář měl být obeznámen s technologiemi, které byly vybrány a z jakého důvodu byly vybrány, proto se k tomu práce již dále nevrací. Pokud je tato práce čtena v jiném než postupném pořadí kapitol, pak je čtenáři doporučeno zaměřit se na předchozí kapitoly této práce.

3.1 Koncept klient/server

Softwarové řešení obsahuje dvě aplikace, jedna slouží pro sběr dat, druhá je určena pro sekretářku a vedení firmy. Podrobněji se o obou aplikacích čtenář dozví v následujících kapitolách.

Vzhledem k tomu, že obě aplikace běží na počítačích v jiných místnostech, jak je patrné z Obr. 13, bylo potřeba uplatnit zapojení PC do sítě. Kapitola čerpá z [18], [22] a [23].



Obr. 13 Zjednodušený půdorys přízemí budovy firmy.

Počítač v nástrojárně připojený do místní sítě má několik funkcí. Na tomto počítači lze pracovat jako na stanici (softwarová obsluha snímače čárových kódů) a zároveň slouží jako poskytovatel databázových služeb ostatním klientům v lokální síti. Obsahuje tedy databázovou službu

MS SQL Server 2005 Express Edition.

Byla zjištěna potřebná délka kabelu a pořízen kabel s označením UTP 100 BASE-T4 cat. 5. Vzdálenost mezi kanceláří a nástrojárnou byla asi 50 m. Maximální délka segmentu pro Fast-Ethernet a tento typ kabelu je 100m, takže nic nebránilo v natažení kabelu. Při samotné instalaci kabelu bylo nutné vyhnout se velkým vstřikovacím strojům, brusce a jiným nástrahám strojírenského provozu. Kabel byl veden podél trubek již nepoužívaného topení nebo podél stropu tak, aby se vyhnul nebezpečným částem provozu, kde by mohl být snadno přeseknut. [21]

Dalším úkonem bylo nastavení databázového serveru, aby k němu mohli přistupovat okolní klienti v síti. SQL Server obsahuje konfigurační nástroje, díky kterým je možné nastavit služby serveru, komponenty a síťové rozhraní. Tato specifická nastavení pro každý SQL Server určují úroveň zabezpečení a řídicí přístup do povrchových oblastí serveru. Prvním nástrojem je SQL Server 2005 Surface Area Configuration. Z tohoto nástroje lze přistupovat ke konfiguraci na lokálním nebo i vzdáleném počítači. Pod položkou Surface Area Configuration for Services and Connections (patrné z Obr. 14) se skrývá nastavení služeb SQL Serveru.

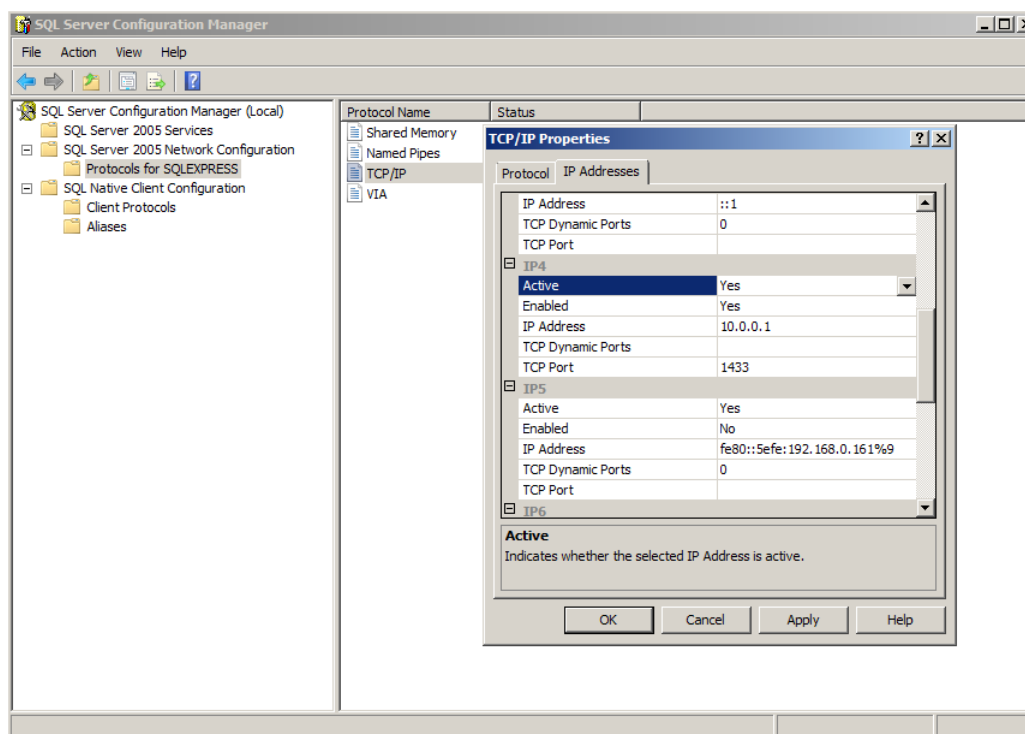


Obr. 14 Hlavní menu nástroje Surface Area Configuration.

U každé služby je možné nastavit, zda má být spuštěna automaticky po startu počítače nebo manuálně. Je možné nastavit tuto službu jako vypnutou. Vzhledem k omezené expresní edici jsme měli k dispozici pouze dvě služby, které jsou našimi aplikacemi využívány, takže jsme je nechali spouštět po startu počítače. Dále je možné u každé služby nastavit způsob připojování klientů k těmto službám. Je možné nastavit pouze lokální přístup nebo vzdálený a lokální. Vzdálená spojení využívají klienti připojující se pomocí aplikací běžících na jiných počítačích, než běží SQL Server. Logicky jsou tak lokální spojení využívána aplikacemi běžícími na stejném počítači jako SQL Server. Pro komunikaci je možné zvolit rodinu protokolů TCP/IP nebo Named Pipes, a nebo taky oboje. V našem případě byly zvoleny protokoly TCP/IP. Implicitně nastavený port, na kterém server naslouchá, je port 1433.

Pod položkou Surface Area Configuration For Features se skrývá spousta dalších nastavení vlastností jednotlivých služeb SQL Serveru. Tato nastavení slouží, stejně jako předchozí, ke zlepšení zabezpečení povrchové oblasti serveru.

Dalším důležitým nástrojem je SQL Server Configuration Manager (Obr. 15), který obsahuje množství nastavení. Kapitola se nebude všemi zabývat. Budou zmíněny jen nastavení ve vztahu k nastavení síťového spojení.



Obr. 15 Okno SQL Server Configuration Manageru.

Pro nastavení síťového spojení ke konkrétní instanci SQL Serveru - v našem případě „SQLEXPRESS“ - stačí vybrat SQL Server 2005 Network Configuration a dále Protocols for .. příslušnou instanci. Dále pak nastavit protokol TCP/IP jako spuštěný a v jeho vlastnostech nastavit IP adresu, na které bude server naslouchat spolu s příslušným portem. V našem konkrétním případě to byla IP adresa 10.0.0.4 a port 1433. Je nutné nastavit Enabled na Yes. Položku Active přepneme na Yes jen tehdy, pokud chceme naslouchat jen na konkrétním portu. Dále je potřeba nahlédnout do složky SQL Native Client Configuration a zvolit pořadí upřednostňovaných protokolů, popřípadě nastavit podle potřeby vlastnost zapnuto/vypnuto jednotlivých protokolů. Je zde také možné konfigurovat jednotlivé protokoly, ale vzhledem k tomu, že nám postačí implicitně nastavené hodnoty nebudou zde tyto vlastnosti zmíněny. Pokud by vznikly při nastavování připojení problémy, podívejte se na použitou literaturu [24], ze které tato podkapitola také čerpá.

Poslední úpravu, kterou je nutné udělat v případě, že Váš Windows používá bránu FireWall, je vytvořit výjimku pro SQL Server 2005 a také pro službu SQL Server Browser, aby nebyly vzdálené požadavky na připojení k serveru odmítnuty, respektive zakázány. Pokud by nebylo jasné, jak tyto nastavení provést, použijte některou literaturu podobnou té, která byla použita zde [23].

Tímto byla ukázána některá hlavní, převážně síťová nastavení nutná ke správnému chodu našeho systému sledování výroby.

3.2 Datová analýza

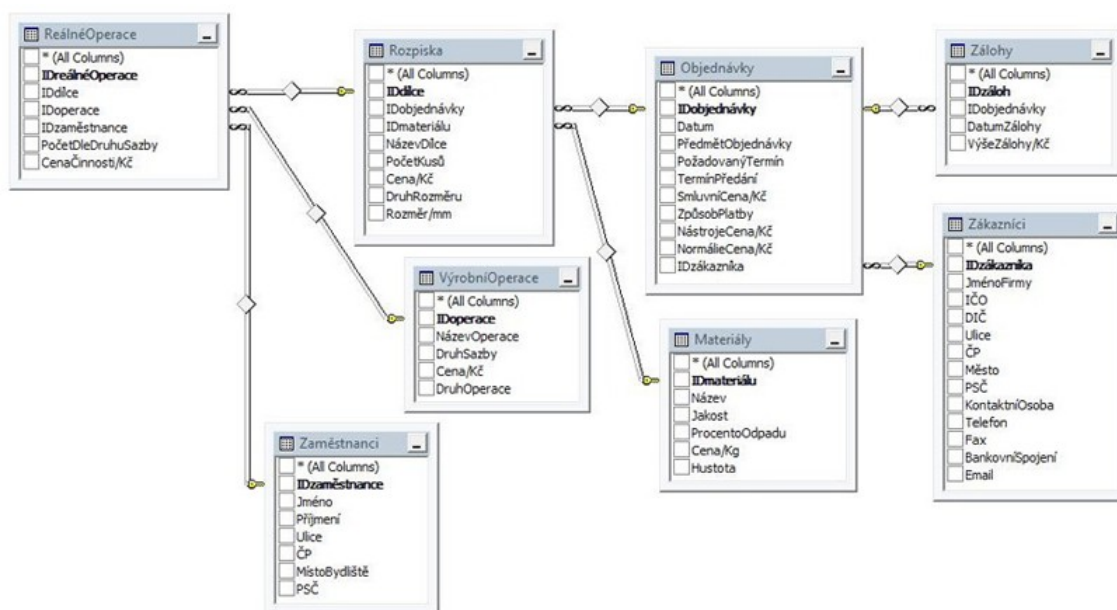
Bylo uskutečněno několik pohovorů s vedením firmy se zaměřením na to, co by chtěli sledovat. Snahou bylo správně analyzovat nejen různé druhy relací, ale konkrétní požadavky firmy. Byly zaznamenány předběžné návrhy tabulek a možné vztahy a jiné požadavky. Pro inspiraci tohoto postupu byla použita literatura [25].

Po takto důkladně provedené analýze byl sestaven relační model pro datové uložení, a to pro konkrétní situaci sledování výroby ve strojírenské firmě. Následně bylo převedeno na skutečné tabulky v relační databázi a byly nastaveny potřebné relace. Navíc byly vytvořeny tzv. Stored

Procedures (uložené procedury), které jsou volané aplikací pro vkládání, aktualizaci nebo mazání dat z databáze. Tyto procedury jsou napsané v Transact-SQL (procedurální jazyk podobný např. Pascalu s možností používat SQL). Výhodou tohoto postupu je například ošetření přiřazení správného identifikačního čísla pro nové položky, apod. Kapitola se nebude o tomto postupu podrobněji zmiňovat. K této práci bylo použito SQL Server Management Studio Express, které obsahuje mnoho užitečných funkcí. Za zmínku snad stojí, že bylo nutné vytvořit fyzický soubor datového uložení, že databáze byla pojmenována „FirmProduction“, a že byl vytvořen účet oprávněný pro přístup k této databázi s možností zápisu a čtení dat z této databáze. Vytvořené tabulky, jak jsou vidět na Obr. 16, je nutné podrobněji popsat (Tab. 2):

Název tabulky	Účel
Objednávky	Podrobné informace o zakázkách
Zálohy	Režie případných záloh pro jednotlivé objednávky
Zákazníci	Podrobnější informace o obchodních partnerech firmy
Rozpiska	Podrobné informace o jednotlivých výrobních dílcích sestavy (formy)
Materiály	Informace o používaných druzích materiálů
ReálnéOperace	Nasnímaná data z provozu a zadané informace o vnějších výrobních operacích
Zaměstnanci	Podrobnější informace o jednotlivých zaměstnancích firmy
VýrobníOperace	Podrobné informace o používaných výrobních operacích uvnitř i vně firmy

Tab. 2 Výchet tabulek v databázi a jejich účel.

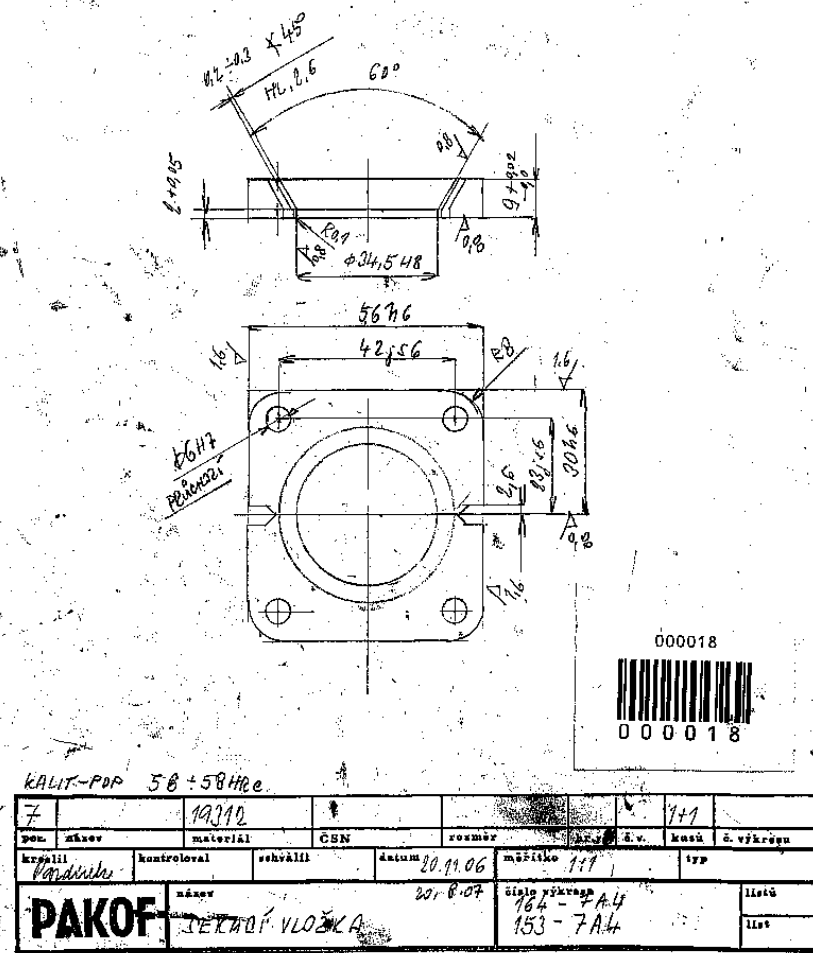


Obr. 16 Relační schéma.

3.3 Aplikace pro sběr dat

Jedná se o jednoduchou aplikaci, která tvoří prostřední článek mezi snímačem čárových kódů (respektive pracovníkem daného provozu) a datovým uložištěm. Návrh aplikace lze udělat několika způsoby. Nutno podotknout, že pracnost jednotlivých návrhů aplikace je srovnatelná. Ve vybraném provozu je potřeba sledovat doby trvání jednotlivých výrobních operací s ohledem, na kterém dílci a kdo danou výrobní operaci provedl.

Každý výrobní dílec má svůj vlastní výrobní výkres, který poskytuje nezbytné informace k výrobě daného dílce. Vzhledem k tomu, že zde mluvíme o výrobě vstřikovacích a vyfukovacích forem je možné tvrdit, že každý dílec je originál (s určitou nadsázkou). Tímto je zajištěno to, že pokud zaměstnanec dostane za úkol provést na daném dílci např. broušení, musí si daný výkres obstarat a podle něj vyrábět. Je tedy účelné opatřit každý takový výkres identifikačním číslem v podobě čárového kódu (Obr. 17).



Obr. 17 Čárový kód s identifikačním číslem na výkrese.

Požadavkem vedení firmy, ve které byl projekt realizován, bylo opatřit čárovými kódy jen výrobní výkresy. Tyto požadavky byly akceptovatelné vzhledem k tomu, že se nejedná o velký provoz (počet zaměstnanců do 10 osob, počet výrobních operací také do 10). Teoretická pravděpodobnost vzniku chyby při zadávání dat v takovém případě je asi 1,3 % (podrobnosti v kapitole 3.3.1), protože do procesu snímání dat vstupuje ruční zadávání pomocí klávesnice. I když se jedná o vyšší pravděpodobnost vzniku chyby, přesto se jedná o pravděpodobnost stále zanedbatelnou. Za celou dobu snímání nebyly zaznamenány výrazné problémy v tomto směru.

Vzhledem k tomu, že se jedná o nástrojářskou dílnu se zaměřením na výrobu vstřikovacích nebo vyfukovacích forem, počet výrobních operací se dramaticky zvyšovat nebude, i kdyby firma zvýšila objem výroby. Se zvýšením objemu výroby ale souvisí zvýšení počtu zaměstnanců. V tomto případě je zadávání identifikačního čísla z klávesnice nemyšlitelné. Proto je vhodné každému zaměstnanci věnovat jeho identifikační číslo v podobě kartičky s čárovým kódem. Z tohoto důvodu byla do aplikace zahrnuta možnost používat čárové kódy identifikující jednotlivé zaměstnance. Při používání tohoto postupu, tedy čtení dvakrát čtečkou a jedenkrát zadávání pomocí klávesnice, je teoretická pravděpodobnost vzniku chyby mnohem menší, a to asi 0,3 % (podrobnosti v kapitole 3.3.1). O realizaci této možnosti snímání v samotné aplikaci je více uvedeno v podkapitole 3.3.2.

Bude tedy záležet, jak moc automatizovaný systém požadujeme. Mohli bychom opatřit všechny sledované prvky čárovými kódy, tedy i výrobní operace v podobě knihy výrobních operací s příslušnými čárovými kódy. Tímto bychom zajistili, že by byla teoretická pravděpodobnost vzniku chyby při zadávání téměř zanedbatelná, a to asi $4 \times 10^{-4} \%$ (podrobnosti v kapitole 3.3.1). Zaměstnanec by nemusel fyzicky zadávat žádné údaje do klávesnice, ať už by se jednalo o jednoduché pohybování pomocí šipek nebo zadávání čísel přes numerickou klávesnici. S aplikací by komunikoval jen pomocí vizuálního kontaktu a se snímačem čárových kódů.

Možností, jak navrhnout aplikaci, je hodně a není cílem je tady všechny vyjmenovat. Účelem bylo upozornit, že pokud bude někdo realizovat jiný projekt někde jinde, bude jeho aplikace vypadat trochu jinak, a to v závislosti na podmínkách konkrétní situace.

3.3.1. Výpočet pravděpodobnosti vzniku chyby při sběru dat

Bylo čerpáno z použité literatury [26]. Pravděpodobnosti pro vznik chyby u jednotlivých prvků (prvek - snímání čárových kódů nebo zadávání dat z klávesnice) jsou uvedeny v kapitole 2.2.1. V našem případě můžeme uvažovat, že se jevy navzájem vylučují, tj. pokud budeme mít třeba variantu o třech prvcích, tak jak bylo uvedeno v kapitole 3.3, nemůže se stát, že chyba vznikne při prvním snímání, při druhém nevznikne, při třetím nevznikne a zároveň na prvním nevznikne, na druhém vznikne a na třetím nevznikne. Nás zajímají všechny možnosti, kdy nastane chyba, tj. pokud alespoň na jednom prvku vznikne chyba. To znamená, že musíme uvažovat všechny jednotlivé kombinace. Chyba může nastat jen u jednoho prvku, může ale nastat na dvou nebo na třech prvcích. Navíc platí, že jedna chyba může nastat jednou u prvního prvku, jednou u druhého, jednou u třetího, apod. Se všemi těmito kombinacemi musíme počítat. Výpočet je ještě delší, pokud budeme mít zadávání o 4 nebo 5 prvcích, jelikož kombinace narůstají. Dále se jedná o jednoduchou matematiku, takže zde podrobný výpočet jednotlivých pravděpodobností uveden není. Jednotlivé jevy, tj. pravděpodobnosti vzniku chyby na jednotlivých prvcích, budeme předpokládat za nezávislé, a proto budeme jednotlivé jevy mezi sebou násobit. Protože se jednotlivé kombinace navzájem vylučují, celková pravděpodobnost bude součet jednotlivých kombinací. Pro přesnější výpočet bychom mohli hledat závislosti mezi jednotlivými prvky systému, ale pro náš případ postačí zmíněné zjednodušení ve formě jednoduchého násobení pravděpodobností.

3.3.2. Funkční analýza aplikace pro sběr dat

Při načtení čárového kódu z výkresu se v prostředí aplikace objeví dialogové okno (Obr. 18.) Zaměstnanec zde zadá své identifikační číslo, přičemž zadávání je ošetřené tak, aby mohl zadávat jen číselné údaje. Pokud zadá a potvrdí jiné, než je v seznamu, nebo své číslo nezadá objeví se varovná zpráva. Zaměstnanec potvrzuje svůj výběr tlačítkem Enter, akci může zrušit tlačítkem ESC.

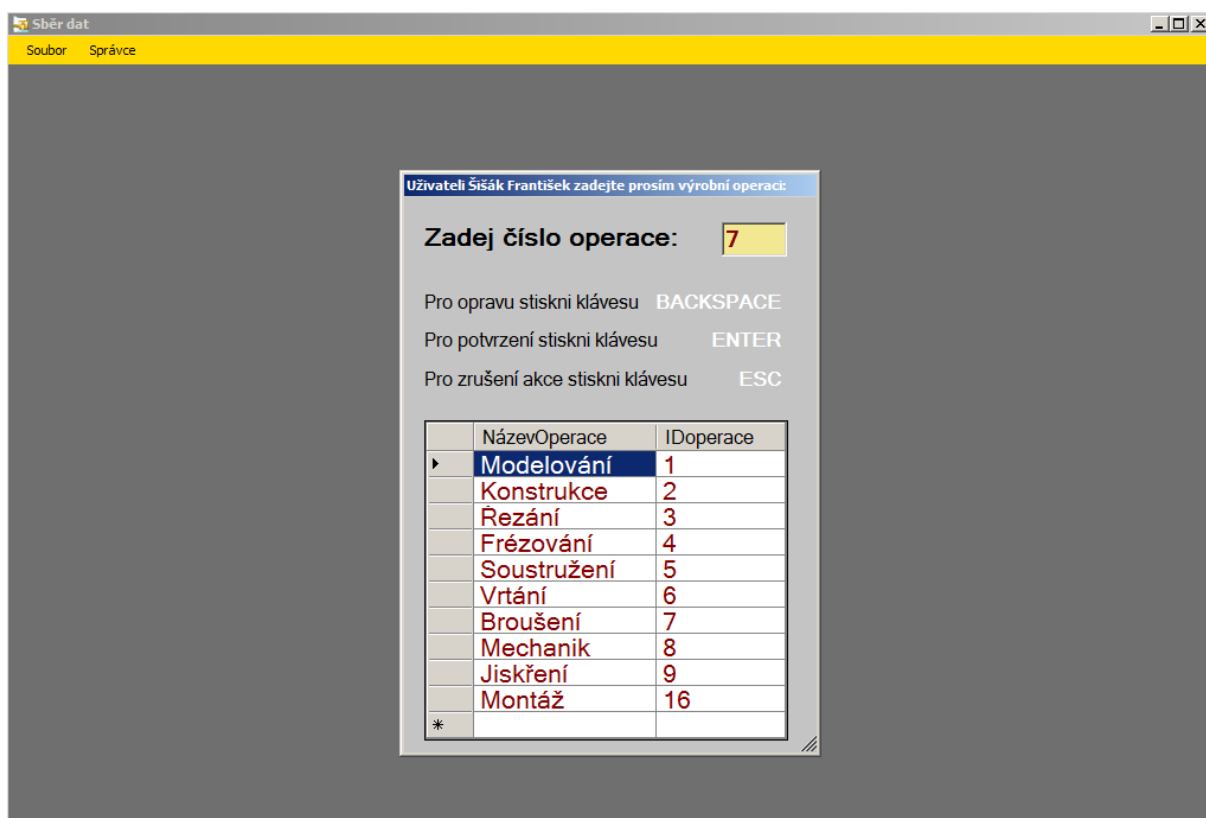
Po potvrzení se objeví další dialogové okno (Obr. 19), kde stejným způsobem zaměstnanec vybere výrobní operaci.

Pokud zadávání proběhne v pořádku, je informace obsahující identifikační číslo zaměstnance, výrobní operace, dílce, čas načtení operace a cenu zapsána do pomocné tabulky v datovém uložišti.

Pro možnost používání identifikačních kartiček zaměstnanci probíhá celá procedura podobně s tím rozdílem, že je v mnohém zjednodušená. Zaměstnanec přijde načte čárový kód ze své kartičky, tím je vynecháno první dialogové okno a zaměstnanec je identifikován na základě svého čárového kódu. Objeví se tedy až druhé dialogové okno, kde zaměstnanec vybere druh výrobní operace. Tentokrát nemusí mačkat klávesu Enter. Pro ukončení procedury stačí, aby přešel čárovým kódem na výkres dílce. Tímto jsou informace uloženy do datového uložisti. Pokud by se snažil ukončit proceduru stiskem klávesy Enter objeví se upozornění, že musí ukončit zadávání dat pomocí čárového kódu na výkresu.

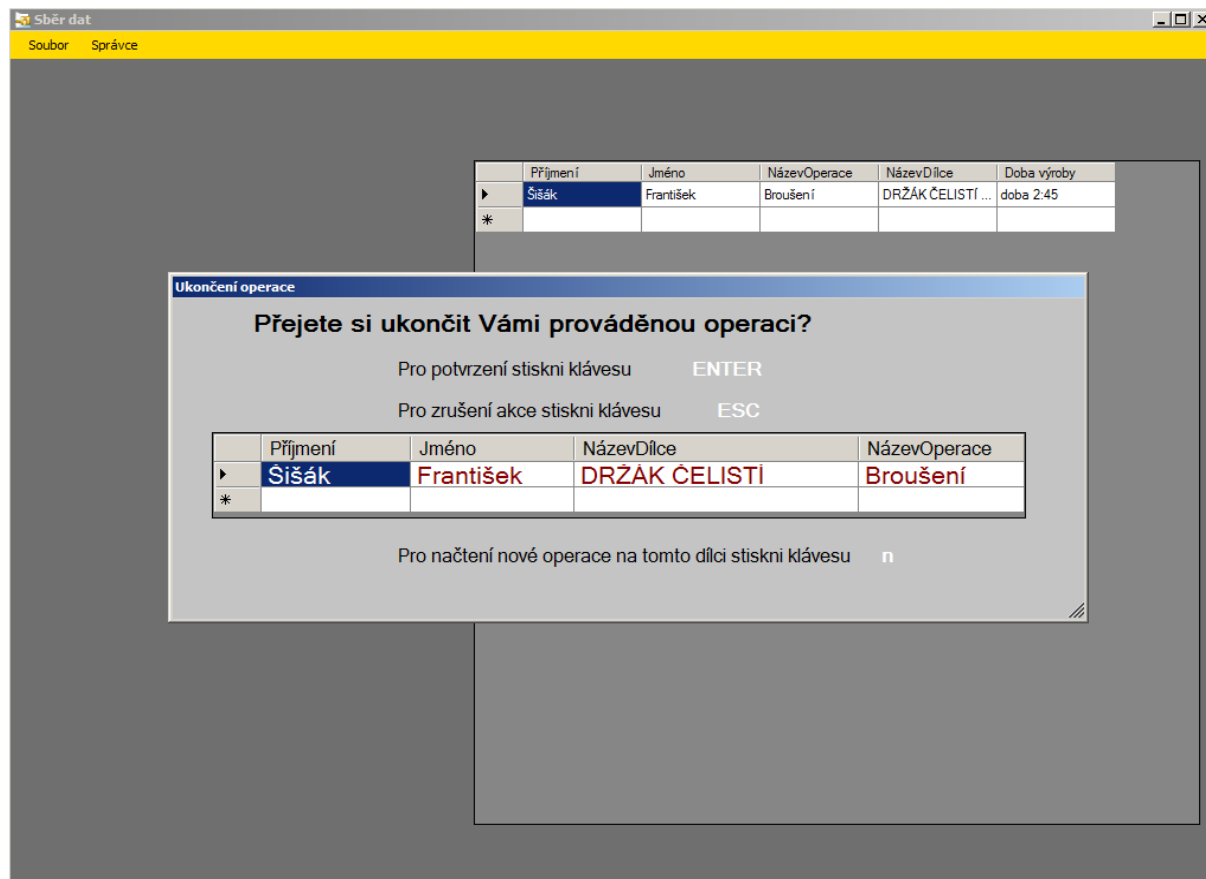


Obr. 18 Dialog pro výběr zaměstnance.



Obr. 19 Dialog pro výběr výrobní operace.

Pokud zaměstnanec ukončuje operaci, nemusí už nic zadávat, stačí, když načte čárový kód, a pokud má dílec načtenou nějakou operaci, je tato operace nalezena v pomocné tabulce v datovém uložišti (tato tabulka nebyla zahrnuta v kapitole 3.2, jelikož slouží jen jako dočasné uložště dat). Pro nalezení dat v pomocné tabulce v datovém uložišti může být použita i identifikační kartička zaměstnance. Následně se objeví dialogové okno (Obr. 20).



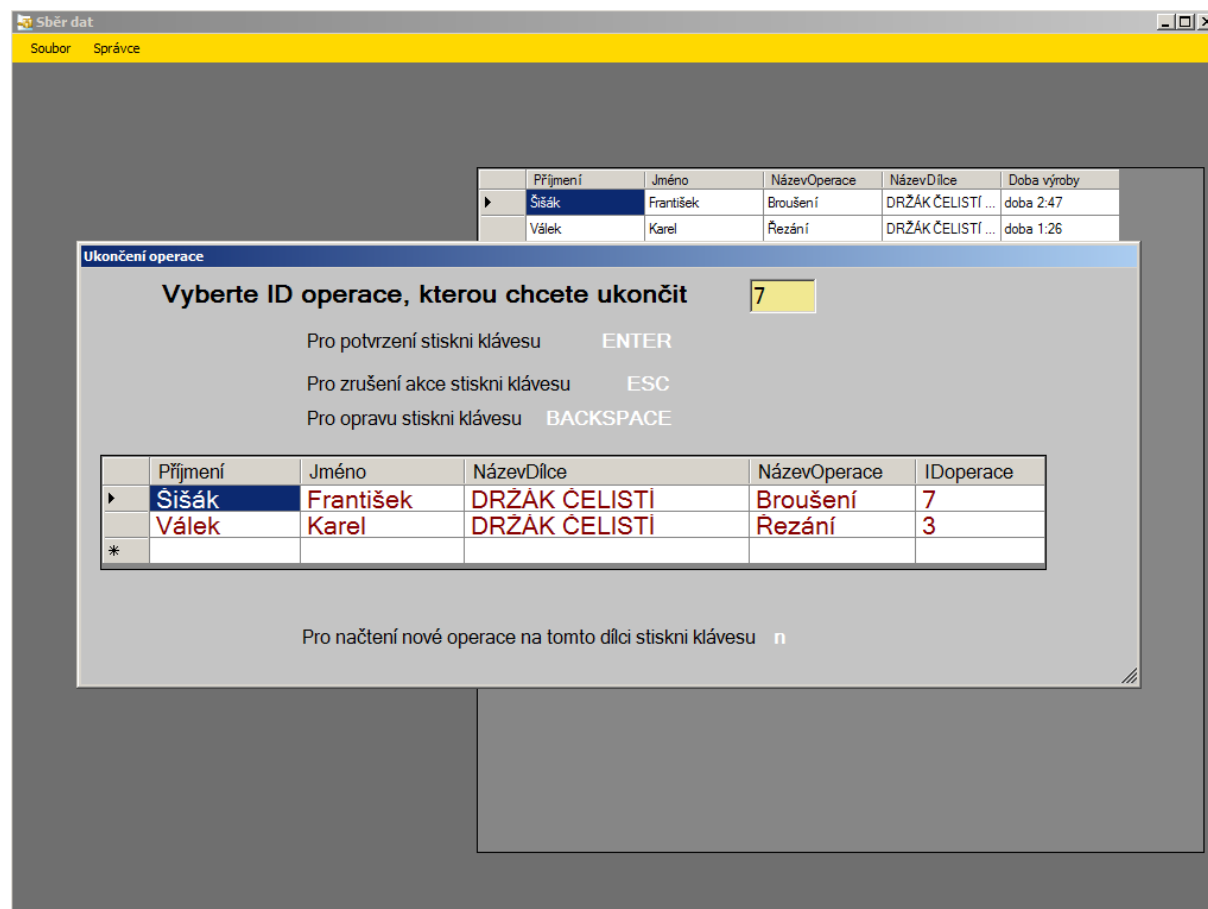
Obr. 20 Dialog pro ukončení jedné operace.

Zaměstnanec již jen potvrdí tlačítkem „Enter“ nebo potvrdí načtením čárového kódu. Může se také rozhodnout zadat další operaci na stejném dílci. V tomto případě stačí, když stiskne písmeno „n“ jako nová operace a následně pokračuje výše popsaná procedura zadání nové operace.

Samozřejmě je funkční i varianta, pokud by se v budoucnosti začalo používat identifikace zaměstnanců pomocí čárových kódů. Tady záleží na tom, jak se zaměstnanec identifikoval. Pokud se identifikoval svou kartičkou objeví se po stisku klávesy „n“ až druhé dialogové okno, kde vybere požadovanou operaci a potvrdí načtením čárového kódu dílce. Pokud se identifikoval čárovým kódem dílce, pak se po stisku klávesy „n“ objeví první dialogové okno, kde se může identifikovat svou kartičkou, až poté následuje výběr operace. Uložit může klávesou „Enter“ nebo načtením čárového kódu dílce.

Dílec tedy může mít načtených více operací (i když se tato možnost zdá nelogická, je možné že zaměstnanec nechá dílec frézovat a jde například modelovat na počítači, nebo jeden zaměstnanec frézuje a druhý stejný dílec modeluje v počítači, apod.) a pro tyto případy se ukončující dialog objeví trochu pozměněný. Řešení je takové, že je možné zadat identifikační číslo operace (Obr. 21), a tím ukončit požadovanou operaci. Při ukončení je ukončovaná operace vymazána z pomocné tabulky v datovém uložišti a zároveň je již příslušná informace - obsahující identifikační číslo zaměstnance, výrobní operace, dílce, doba trvání operace a cena odvozená od druhu výrobní operace a doby jejího trvání uložena do tabulky „ReálnéOperace“ v datovém uložišti.

Samozřejmě v tomto případě nefunguje ukládání pomocí snímání čárových kódů, a to z důvodů nejednoznačné identifikace na ukončení požadované operace.



Obr. 21 Dialog pro ukončení více operací.

Na pozadí aplikace běží tabulka (Obr. 22) obsahující aktuálně načtené operace. Tato tabulka slouží pro kontrolu zaměstnance, jestli se mu podařilo správně data zadat. Navíc je přehledně vidět kdo, na čem a co dělá, a na kolika operacích se pracuje. Stejná tabulka je zahrnuta i v sesterské aplikaci určené pro vedoucí osoby v kanceláři. Zde zase slouží pro kontrolu vedení jak dlouho, na čem a kdo pracuje. Tím je možné sledovat, jestli se někdo neulívá, na čem se pracuje, popřípadě které stroje jsou aktuálně v provozu.

	Příjmení	Jméno	NázevOperace	NázevDílce	Doba výroby
▶	Šišák	František	Broušení	DRŽÁK ČELISTÍ ...	doba 2:49
	Válek	Karel	Řezání	DRŽÁK ČELISTÍ ...	doba 1:28
*					

Obr. 22 Výpis operací v provozu

V aplikaci lze nastavit některé důležité vlastnosti spojení snímače čárových kódů jako je baud rate, jméno portu (COM1, COM2 atd.), prefix, surfix a znaková sada kódu (Obr. 23) - viz. kapitola 2.4.3. Tato nastavení jsou přístupná jen oprávněným uživatelům, kteří se musí přihlásit (viz. přihlašování uživatelů v kapitole 3.4).

The image shows a dialog box titled "Konfigurace spojení". It has five input fields: "BaudRate" with a dropdown menu showing "9600", "PortName" with a dropdown menu showing "COM1", "Prefix" with a text box containing "#", "Surfix" with a text box containing "\$", and "Type-code" with a text box containing "K". At the bottom, there are two buttons: "Uložit" and "Storno".

Obr. 23 Nastavení spojení se snímačem č. kódů.

3.4 Aplikace pro zadávání a analýzu dat

Tato aplikace je již poměrně rozsáhlejší oproti předchozí. Je v ní využito výhod objektově orientovaného návrhu, tedy dědičnosti, polymorfismu, atd., jazyk C# umožňuje použít mnoho skvělých výhod tohoto přístupu k programování. S datovým uložištěm je komunikováno, stejně jako v předchozí aplikaci, pomocí SQL jazyka. Jak již bylo zmíněno, bylo použito rozhraní ADO.NET, které obsahuje mnoho tříd, které zajišťují spojení s SQL Serverem. V této kapitole nebudou zmíněny podrobnosti, jak byla aplikace naprogramována. Cílem práce je spíše ukázat, jak je možné problém sledování výroby vyřešit. I když by tato kapitola mohla projít celou aplikací, je zde rozebrána jen část pro režii s daty. Analýza dat zde rozebírána není, jelikož bude popsána v kapitole 4.

3.4.1. Funkční analýza aplikace pro zadávání a analýzu dat

Kapitola se tedy zabývá jen částí pro zadávání dat. Přičemž se jedná o část, kterou využije sekretářka nebo pomocná síla, která zadává nové objednávky, a k tomu příslušné informace o jednotlivých dílcích. Dále upravuje tabulky se zaměstnanci, materiály nebo přidává výrobní operace, apod. Cílem bylo navrhnout aplikaci tak, aby zadávání a úprava dat byla pohodlná a přehledná.

Po spuštění aplikace se musí uživatel přihlásit pomocí svého příjmení a hesla (Obr. 24). Tato operace byla zařazena z důvodů odstupňování aplikace pro vedoucí provozu a pro pomocnou sílu, která jen zadává data. Je zřejmé, že vedení bude chtít mít kontrolu jak nad daty naplněnými do tabulek, tak nad částí aplikace umožňující analýzu dat. Díky takovému odstupňování aplikace je možné zpřístupnit různým skupinám uživatelů různé funkce nebo definovat jejich osobní styly barev aplikace nebo zobrazení, apod. Přihlášení funguje tak, že pokud je zadáno špatné jméno nebo heslo, uživatel je na to upozorněn, v opačném případě je uvítán uvítací zprávou. Pokud zadá třikrát po sobě chybné ověření, ukáže se mu, jaké jméno a heslo zadal, aby si mohl zkontrolovat, jestli náhodou nezadá heslo velkými písmeny, apod.

The image shows the login screen of the "Sledování výroby" application. At the top, there is a menu bar with "Soubor" and "Správce". Below it, there are two labels: "Příjmení" and "Heslo". The "Příjmení" field contains the text "Root". The "Heslo" field is masked with asterisks. At the bottom, there are two buttons: "Přihlásit" and "Storno".

Obr. 24 Přihlášení do aplikace.

Oprávnění uživatelé mohou měnit, přidávat nebo mazat uživatelské účty (Obr. 25). Pro jednoduchost byly definovány dva odlišné styly udávající barvy aplikace a dvě vrstvy zabezpečení.

Při změně některé položky v tabulce se tlačítko „Aktualizuj data v databázi“ změní na žlutou barvu, aby uživatel věděl, že došlo ke změně oproti datům uloženým v datovém úložišti.

IDheslo	IDzaměstnanec	Heslo	Zabezpečení	Styl
1	8		Vedoucí	1
2	1		Zadavatel	2
*	2			
	3			
	4			
	5			
	6			
	7			
	8			
	9			

IDzaměstnanec	Jméno	Příjmení	Ulice	ČP
8	Jaromír	Škrabal	xxxxx	xxxxx
*				

Obr. 25 Změna zaměstnance vlastního účtu.

Po úspěšném přihlášení se objeví menu, ve kterém se lze snadno pohybovat a vybírat požadované tabulky (Obr. 27). Menu bylo rozděleno na tři vrstvy. V databázi lze najít dva druhy tabulek. První jsou ty, které neobsahují cizí klíče. Tyto tabulky slouží k definování entit jako jsou zákazníci, zaměstnanci firmy, materiály nebo operace. Operace jsou dále rozděleny až v aplikaci podle toho, jestli jsou vnitřní nebo vnější, a to na výrobní operace a kooperace. Tyto tabulky lze tedy najít pod položkou „Základní data“ v první vrstvě menu. Ve druhé vrstvě se vždy objeví specifické položky podle výběru první vrstvy menu. Třetí vrstva obsahuje tlačítka a názvy různých operací užitečných k filtrování, ukládání a mazání dat. Tyto položky se liší podle toho, o jakou záložku se jedná. Například u správce hesel bylo možné jednotlivé účty mazat, u tabulky „Materiály“ to povoleno není. Má to své opodstatnění vzhledem k definovaným relacím mezi tabulkami.

IDoperace	IDzaměstnanec	PočetDleDruhuSaz	CenaČinnosti/Kč
13	7	2,24	207
13	7	2,08	6
13	7	0,65	2
13	7	1	3
10	7	0,26	260
11	7		
11	7		
13	7		
10	7		
11	7		
11	7	0,25	500

Zadej počet dm2

dm2

Uložit Storno

Obr. 26 Příklad předem připraveného dialogu pro zadávání dat.

Sledování výroby
Soubor Správce

Základní data Vstupní data Analýza dat

Zákazníci Zaměstnanci **Materiály** Výrobní operace Kooperace

Načti data z databáze Aktualizuj data v databázi Vyber typ zobrazení

změněné řádky: 0

IDmateriálu	Název	Jakost	ProcentoOdpadu	Cena/Kg	Hustota
1	Dural	Certal	10	230	2800
2	Dural	Tvrzený	15	241	2800
3	Dural	Obyčejný	15	140	2800
4	Měď	Nimonic	15	285	8960
5	Ocel-obyč.	11523	15	35	7800
6	Ocel-kulatina	12050	15	30	7800
7	Ocel-kulatina	14220	15	58	7800
8	Ocel-kulatina	19312	15	94	7800
9	Ocel-kulatina	19436	15	140	7800
10	Ocel-kulatina	19421	15	140	7800
11	Bronz	Tažená	15	270	8900

Obr. 27 Tabulka „Materiály“ obsahuje jen primární klíč.

Další záložkou v první vrstvě menu je záložka „Vstupní data“, která obsahuje tabulky, které již obsahují cizí klíče, a jsou tedy propojovacími tabulkami, a zároveň tak definují některou entitu na základě svých vlastních atributů nebo atributů získaných některou relací. Jak ukazuje Obr. 28 můžeme zde najít tabulku Objednávky, Zálohy, Rozpiska, Kooperace na dílcích nebo Výrobní operace na dílcích. Vzhledem k použitým relacím bylo nutné zajistit, aby výběr zpřesňujících informací byl přehledný (Obr. 30). Není možné, aby si zadavatel pamatoval například cizí klíče všech materiálů v podobě identifikačních čísel. Proto se po kliknutí do příslušné buňky objeví vysouvací roletové menu, ve kterém se objeví specifické informace o daném materiálu, kooperaci, apod. Protože vypisovat všechny informace do tohoto rozbalovacího menu by bylo nesmyslné, a to z důvodů nepřehlednosti, objeví se v dolní části tabulka obsahující další doplňující informace. Takto zadavatel přesně ví, jakou položku z jiné tabulky svazuje s položkou aktuální tabulky.

Tam, kde to nebylo potřebné, byly do vysouvacího menu umístěny jen identifikační čísla. Když přijde nová objednávka, je nutné zadat příslušná data a přilepit čárové kódy na výrobní výkresy. Celé zadávání má svou logiku a aplikace je tomu přizpůsobena tak, aby to bylo přehledné a rychlé. Například tabulka Objednávky a Rozpiska má v třetí vrstvě menu definována speciální filtrovací tlačítka. Pokud zadavatel zadává novou objednávku, je nesmyslné, aby se mu v Rozpisce ukázaly dílce jiných objednávek, nebo v tabulce Kooperace kooperace jiných dílců, atd. Stačí do volné kolonky zadat číslo objednávky a zadavatel je po kliknutí na tlačítko „Zadej kooperace“ přesunut do příslušné tabulky s odfiltrovanými daty. V takto odfiltrované tabulce se zobrazí data pouze pro danou objednávku.

Navíc je toto filtrování uplatněno na vysouvací roletové menu (Obr. 29), takže například pro zadání kooperace na dílci zadavatel nevybírání z tisíců dílců, ale jen třeba z 20 pro danou objednávku, na které zrovna lepší čárové kódy (přesněji na výrobní výkresy).

ID dílce	ID objednávky	ID materiálu	Název dílce	Počet kusů	Cena/Kč	Druh Rozměru	Rozměr/mm	Povol zápis ceny
1	1	3	DRŽÁK ČELISTI ...	2	122	Hranol	20x50x135 ...	<input type="checkbox"/>
2	1	3 jakost Obyčejný	ČELISTI DOLNÍ ...	2	293	Hranol	20x50x325 ...	<input type="checkbox"/>
3	1	1 jakost Čertal	ČELISTI HORNÍ ...	2	293	Hranol	20x50x325 ...	<input type="checkbox"/>
4	1	2 jakost Tvrdý	OPĚRA TĚLA ...	1	143	Hranol	42x70x108 ...	<input type="checkbox"/>
5	1	3 jakost Obyčejný	OPĚRA HRDLA ...	1	11	Hranol	20x30x40 ...	<input type="checkbox"/>
6	1	4 jakost Nimonic	VÝZTUHA ...	1	11	Hranol	10x20x125 ...	<input type="checkbox"/>
7	1	5 jakost 11523	UPÍNACÍ DESKA...	1	42	Hranol	12x70x160 ...	<input type="checkbox"/>
8	1	6 jakost 12050	KOPYTO HRDL...	1	256	Hranol	30x135x140 ...	<input type="checkbox"/>
9	1	7 jakost 14220	KOPYTO UCHA ...	1	16	Hranol	20x25x70 ...	<input type="checkbox"/>
10	1	8 jakost 19312	UPÍNACÍ DESKA...	1	37	Hranol	12x62x160 ...	<input type="checkbox"/>
11	1	9 jakost 19436	MEZIDESKA UC...	1	12	Hranol	12x62x50 ...	<input type="checkbox"/>
12	1	10 jakost 19421	ROZPĚRKA ...	6	64	Válec	14x221 ...	<input type="checkbox"/>
13	2	11 jakost Tažená	TĚLO FORMY L ...	2	8747	Hranol	98x210x300 ...	<input type="checkbox"/>
14	2	12 jakost Obyčejná	HRDLO FORMY ...	2	2508	Hranol	82x98x300 ...	<input type="checkbox"/>

Procento Odpadu	Cena/Kg	Hustota
15	140	2800

Obr. 28 Tabulka Rozpiska.

Dále je nutné říci, že pokud uživatel přechází mezi tabulkami, je varován, pokud některé data pozměnil a neuložil do databáze. Po uložení změněných dat do databáze se vypíše počet změněných řádků do připravené kolonky pod tlačítkem „Aktualizuj data v databázi“. Tímto má zadavatel kontrolu o provedené operaci. Samozřejmě je aplikace ošetřena chráněnými bloky a navíc, tam, kde to bylo možné a vhodné, se objeví varovný dialog, pokud uživatel udělá něco nepatřičného.

IDreálnéOperace	IDdílce	IDoperace	IDzaměstnanec	PočetDleDruhuSaz	CenaČinnosti/Kč	Povol zápis ceny
15	49	11	7	0,734760046005...	1470	<input type="checkbox"/>
16	39	11	7	0,25	500	<input type="checkbox"/>
*	40					<input type="checkbox"/>
	41					<input type="checkbox"/>
	42					<input type="checkbox"/>
	43					<input type="checkbox"/>
	44					<input type="checkbox"/>
	45					<input type="checkbox"/>
	46					<input type="checkbox"/>
	47					<input type="checkbox"/>
	48					<input type="checkbox"/>
	49					<input type="checkbox"/>
	50					<input type="checkbox"/>
	51					<input type="checkbox"/>
	52					<input type="checkbox"/>

Obr. 29 Tabulka kooperací s odfiltrovanými daty.

Nesporným usnadněním pro zadavatele jsou některé předem připravené dialogy (Obr. 32). Obr. 26 také patří mezi předem připravené dialogy, tento dialog se objeví jen u druhu položky s dm^2 , protože například kg jsou automaticky dopočítány. Aplikace dm^2 nedopočítává, protože ne všechny plochy jsou zinkované, apod. Dalším usnadněním jsou vysouvací menu s předem definovanými hodnotami. Například u některých buněk je předem jasné, že zadavatel bude vypisovat opakující se dvě, tři i více hodnot. V takovém případě se vyplatí usnadnit zadavateli práci (Obr. 31).

	IDreálnéOperace	IDdílce	IDoperace	IDzaměstnanec
	1	7	13	7
▶	2	10	p. Zinkování	7
	3	11	10 n.o. Kalení	7
	4	12	11 n.o. Cementace	7
	5	18	12 n.o. Nitridace	7
	6	20	13 n.o. Zinkování	7
	7	21	14 n.o. Elox přírodní	7
	9	29	15 n.o. Elox černý	7
	14	34	11	7
	15	49	11	7
	16	52	11	7
	17	37	11	7
	18	38	10	7
	58	61	11	7

Podrobnosti o vybrané operaci				
	IDoperace	NázevOperace	DruhSazby	Cena/Kč
▶	13	Zinkování	dm2	3
*				

Obr. 30 Přehledné svazování položek mezi tabulkami.

Dalším usnadněním je automatické počítání cen. Tam, kde by uživatel musel něco počítat, to dělá aplikace za něj. Například v uvedené tabulce Rozpiska, jak patrně z uvedeného Obr. 28, je cena dílce vypočtena z informací o rozměru, druhu materiálu a počtu kusů. Cena je vypočtena poté co zadavatel zadá všechny potřebné údaje. Pokud se některý změní, je cena přepočítána. Uvedený postup je uplatněn i v jiných tabulkách, kde se počítalo něco jiného. Samozřejmě platí, že pokud je cena tímto způsobem automaticky dopočítávána, není nutné ani účelné, aby byla takové kolonky udělena možnost zápisu. Někdy je ale potřeba cenu „natvrdo“ zadat cenou neodpovídající výpočtu. Například u kooperace zinkování si sice obchodníci účtují za dm², ale pokud je dílec malý a správná cena by měla být třeba jen 86 Kč, firma zprostředkující tuto službu si naúčtuje minimální taxu například 200 Kč. To samozřejmě odpadá, pokud se nechá pozinkovat více dílců dohromady. Někdy se ale stane, že je potřeba mít dílec hotový rychle, a čekat na další dílce by znamenalo mnohem větší ztráty. Proto byla přidána kolonka „Povol zápis ceny“ (Obr. 29). Po zaškrtnutí příslušného řádku je možné cenu libovolně měnit.

DruhSazby	Cena/Kč
Kg	1000
Kg	2000
Kg	300
kus	
dm2	3
dm2	5

Obr. 31 Usnadnění zadávání.

Dalším příkladem „chytrého“ chování je cílené odfiltrování možných chyb. Pokud uživatel

nezadá „DruhRozměru“ a chce zadat „Rozměr/mm“ objeví se zpráva, že musí nejdříve vybrat „DruhRozměru“, protože na tom závisí vzhled dialogového okna. Logicky je vzhled dialogového okna pro hranol jiný než pro válec. V ostatních částech aplikace jsou tyto možné neduhy zadavatele taktéž ošetřeny podobným způsobem.

PočetKusů	Cena/Kč	DruhRozměru	Rozměr/mm
2	122	Hranol	20x50x135 ...
2	293	Hranol	20x50x325 ...
2	293	Hranol	20x50x325 ...
1	143	Hranol	42x70x108 ...
1			20x30x40 ...
1			10x20x125 ...
1			12x70x160 ...
1			30x135x140 ...
1			20x25x70 ...
1			12x62x160 ...
1			12x62x50 ...
6			14x221 ...
2	18747	Hranol	98x210x300 ...

Zadej rozměry dílce

délka

šířka

výška

Obr. 32 Dialog pro zadání rozměru hranolu.

Při sledování výroby nebyly zaznamenány žádné větší problémy, přesto uvedeme jeden příklad. Vzhledem k tomu, že ve firmě není zatím elektronický docházkový systém, se občas stalo, že se některý zaměstnanec zapomenul na konci směny odhlásit. Nejdříve to bylo řešeno tak, že byla do aplikace pro vedoucí pracovníky umístěna záložka, kde vedoucí vidí aktuální operace v provozu a zároveň je může z prostředí programu ukončit. Pokud například ví, že už někdo odešel a neodhlásil se, odhlásí ho ručně „natvrdo“, a to z pohodlí své kanceláře (Obr. 33). Tento dialog se objeví i při vypínání aplikace, aby upozornil vedoucího při odchodu, a to tím způsobem, že se ho aplikace zeptá před vypnutím, jestli chce nebo nechce aktuálně prováděné operace ukončit. Tímto se problém úplně nevyřešil, jelikož vedoucí není vždy na konci směny přítomen. Pokud se zaměstnanec zapomenul odhlásit, druhý den ráno vznikly operace s 21 hodinami práce. Proto byl do aplikace pod záložku „Vstupní data“ -> „Výrobní operace na dílcích“ umístěn do třetí vrstvy filtr, který podle zvolené objednávky vyfiltruje jednotlivé výrobní operace s větším počtem odpracovaných hodin než je 8 hodin. Takto je vedoucí schopen docela jednoduše vzniklé chyby opravit.

Ukončené operace: 0, zbývající: 2

	Jméno	Příjmení	NázevOperace	NázevDílce	ČasNačtení	Dokonči
	Miroslav	Slavický	Frézování	DRŽÁK ČELISTÍ ...	29.2.2008 13:00	<input checked="" type="checkbox"/>
	František	Šišák	Soustružení	ČELISTÍ DOLNÍ ...	29.2.2008 14:34	<input type="checkbox"/>
*						<input type="checkbox"/>

Obr. 33 Ukončení výrobní operace z prostředí aplikace.

Toto řešení ještě nevyřešilo problém úplně elegantně. Pokud by byl zaveden elektronický docházkový systém, pak by tyto chyby nevznikaly. Jestliže by měl zaměstnanec při odchodu načtené operace, byly by automaticky odhlášeny a s příslušnými informacemi uloženy do datového úložiště. To je ale mimo rozsah této práce. Proto to spíše navrhuji jako další projekt do budoucna. V dané firmě se používá ke sledování docházky nástěnka a fix.

4 EKONOMICKÁ A ČASOVÁ ANALÝZA

Tato kapitola představí část aplikace věnované ekonomické a časové analýze sledování výroby a pokusí se o zhodnocení výstupů z nasbíraných dat v již zmiňovaném konkrétním průmyslovém provozu. Výroba byla sledována po dobu asi 5 měsíců, od začátku listopadu 2007 po konec března 2008. Za tuto dobu se podařilo zkompletovat 11 forem. Takto velká základna dat nám umožní racionálně zhodnotit získané informace. Jedním z úkolů této práce je objektivizovat dosud jen předpokládané náklady na výrobu forem. Ceny materiálů, normálií, atd. jsou účtovány jednotlivými dodavateli. Takové položky sice zaujímají velkou část nákladů, jsou však vzhledem k ceně pevně dané, tímto pro nás nejsou podstatné. Ceny položek, které byly odhadnuty z předchozích zkušeností, byly ceny výrobních operací jako je frézování, vrtání, jiskření, apod. Úkolem tedy bylo objektivizovat náklady na výrobní operace. Vzhledem k tomu, že cena výsledné formy je z určitého hlediska funkcí závislou na určení cen výrobních operací, je tato cena až důsledkem závislým na cenách výrobních operací. Ceny materiálů, normálií, apod. jsou konstantami. Zisk firmy za dané období je nulový, tj. to, co firma získá, utratí beze zbytku na chod firmy. Na základě této skutečnosti a získaných dat můžeme navrhnout navýšení nebo snížení cen výrobních operací. Díky této analýze bude moci vedení firmy reálně ohodnotit budoucí zakázky tak, aby nedošlo k jejich přílišnému nadhodnocení nebo podhodnocení.

4.1 Časová analýza

Než se pustíme do ekonomické analýzy, bude předvedena časová analýza jednotlivých zakázek. Tato část aplikace slouží k rozboru trvání jednotlivých operací a nachází se pod záložkou „Zakázka dle pracnosti“, přičemž je možné sledovat, kolik hodin trvaly jednotlivé operace a jaké procento zaujímají tyto doby z celkové doby výroby dané formy. Tyto skutečnosti umožňují uživateli rozhodnout, která výrobní operace měla na zakázce největší finanční podíl, a vyvodit tak z toho patřičné důsledky. V podobném duchu můžeme sledovat, jak dlouho na dané zakázce pracovali jednotliví zaměstnanci nebo kolik hodin bylo odpracováno na jednotlivých dílech sestavy.

Zakázka dle pracnosti Z. dle ekonomiky Z. dle rozpracovanosti Podrobnější informace o z. Prognóza Operace v provozu

Pro 2 Kanystr 5L, z firmy: xxxxxxxxxxxx, zastoupení panem: xxxxxxxxxxxx Od data 1. listopadu 2007

Operace podle profese Generuj graf

NázevOperace	Počet/hod	% z celkové doby
Frézování	93,4	37,5
Vitání	43,8	17,6
Broušení	30,9	12,4
Montáž	27,6	11,1
Mechanik	23,9	9,6
Soustružení	20,2	8,1
Jiskření	9,5	3,8
Modelování	0,1	0
*		

Operace podle zaměstnanců Generuj graf

Příjmení	Počet/hod	% z celkové doby
Válek	78,7	31,6
Fantomas	47,5	19,1
Čemý	47,5	19,1
Slavický	45,3	18,2
Šišák	20,2	8,1
Škrabal	9,5	3,8
Pšeja	0,6	0,2
*		

Operace podle dílce Generuj graf

NázevDílce	Počet/hod	% z celkové doby
TĚLO FORMY L...	59,9	24
HRDLO FORMY ...	58,8	23,6
DNO FORMY L+...	46,8	18,8
ZÁVITOVÁ VLO...	15,6	6,3
UPÍNAČÍ DESKA...	12,6	5,1
VODÍČÍ POUZD...	9,8	3,9
HORNÍ A DOLNÍ...	8,1	3,2
KOPYTO HORNÍ...	7,2	2,9
VLOŽKA HRDLA...	6	2,4
VODÍČÍ ČEP ...	5,6	2,2
SEKACÍ VLOŽKA...	4,3	1,7

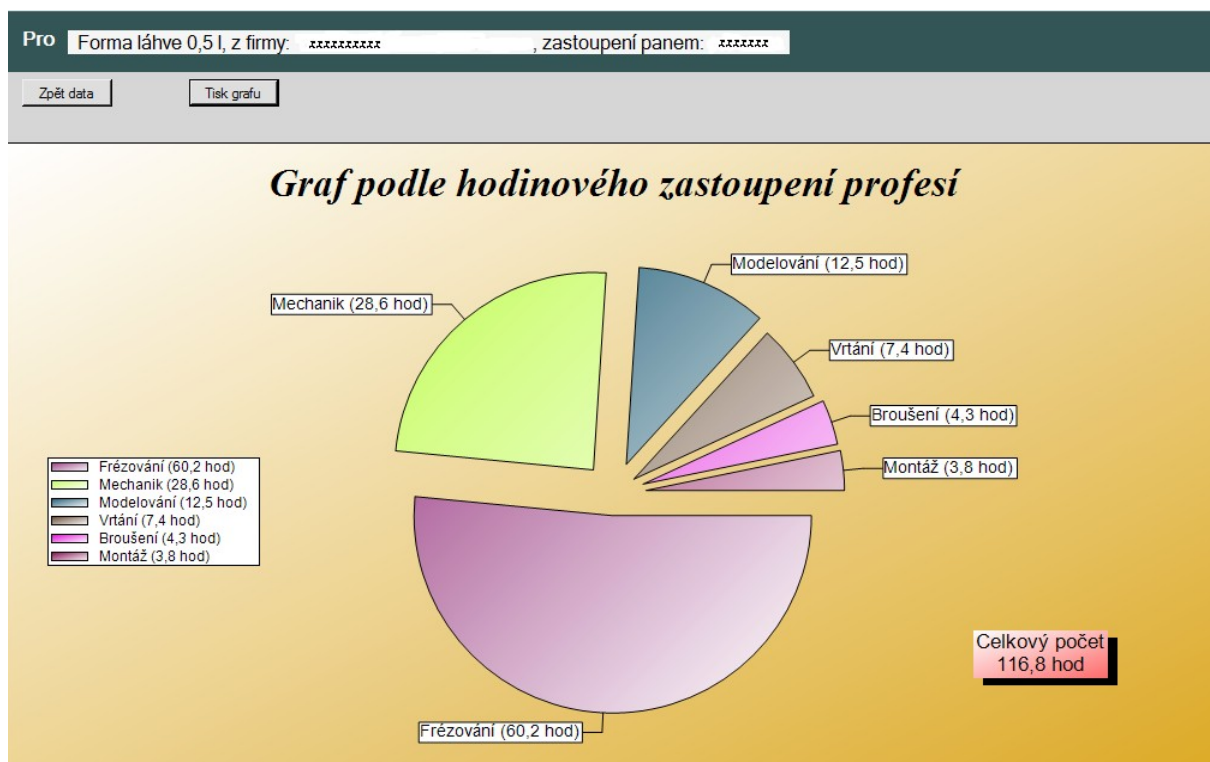
Sumace celkových časů

Datum zadání	Termín Předání	Odpracovaných hodin	Počet dnů	Pracovních dnů
19.11.2007	19.2.2008	249,2	95	31,2
*				

Obr. 34 Časová analýza dle pracovních úkonů na zakázce.

Ze získaných informací je možné pro budoucí zakázky rozhodnout, zda se třeba nevyplatí některé dílce kvůli jejich pracnosti zadat do jiné specializované firmy, kde by vyšly levněji.

Z Obr. 34 je patrné, že obsluha aplikace není složitá. Pomocí kalendáře je možné filtrovat zakázky jen od určitého data tak, aby v seznamu nebylo třeba 500 zakázek a uživatel nemusel tímto seznamem listovat a pracně vyhledávat hledanou zakázku. Po výběru zakázky se objeví vypočtená a vyfiltrovaná data. Uživatel zde může zjistit, jaká je celková doba výroby v hodinách, kolik je to v pracovních dnech (pracovní den je 8 hodin) a v neposlední řadě, kolik dnů trvalo, než byla zakázka předána zákazníkovi. Všechny tyto skutečnosti mohou pomoci při dalším plánování objemu výroby nebo pro zefektivnění výroby. Jak je patrné z Obr. 35, data je možné přehledně zobrazit do grafu.



Obr. 35 Přehledný graf reprezentující hodinové zastoupení výrobních operací.

4.2 Ekonomická analýza

Na rozpracovaných zakázkách je možné sledovat, kolik již bylo vyčerpáno nebo přečerpáno prostředků. K tomuto účelu nám slouží záložka „Z. dle rozpracovanosti“. Opět je možné filtrovat zakázky dle data. Jak je patrné z Obr. 36, ve výběru se objeví jen rozpracované zakázky, tedy zakázky bez zapsaného data předání. Aplikace vyhledá dosud rozpracované položky a ukáže, kolik procent je vyčerpáno ze smluvní ceny.

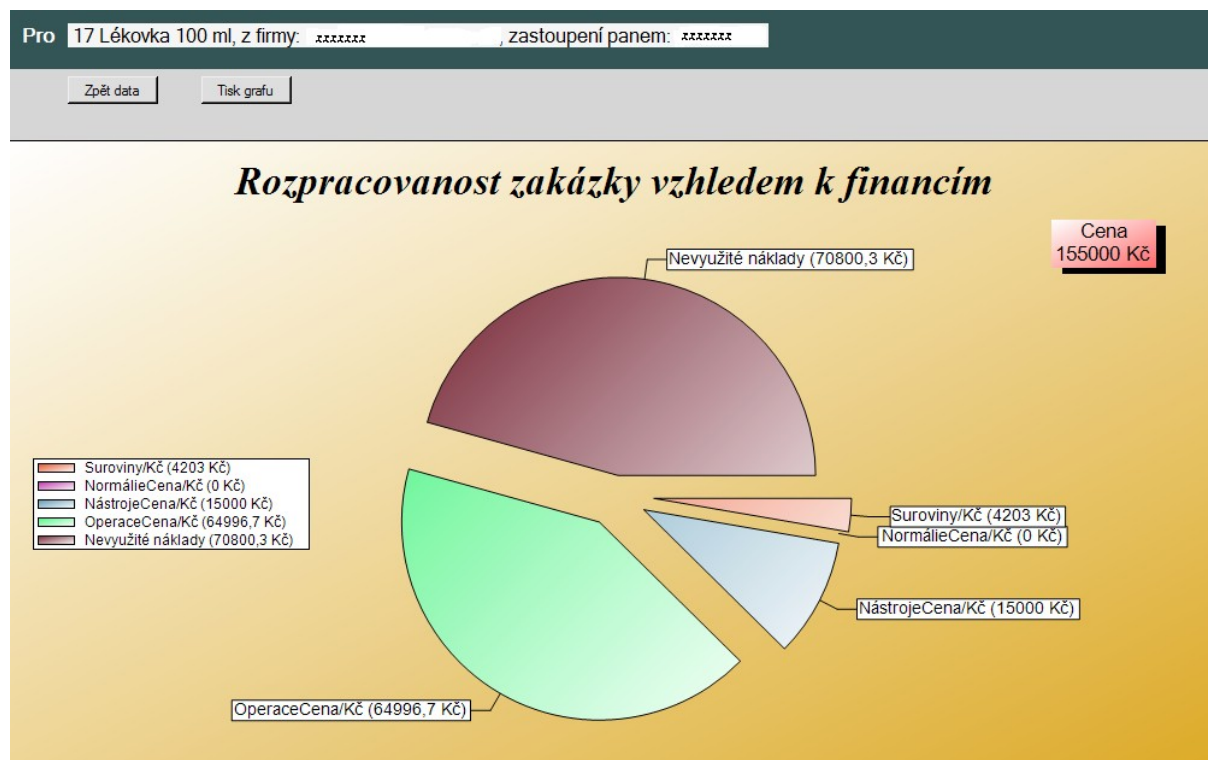
Pro 17 Lékovka 100 ml, z firmy: xxxxxxxx, zastoupení panem: xxxxxxxx Od data 1. listopadu 2007

4 jádro, z firmy: xxxxxxxx, zastoupení panem: xxxxxxxx
 10 razník, z firmy: xxxxxxxx, zastoupení panem: xxxxxxxx
 13 dílce xxxxxxxx, z firmy: xxxxxxxx, zastoupení panem: xxxxxxxx
 17 Lékovka 100 ml, z firmy: xxxxxxxx, zastoupení panem: xxxxxxxx
 18 Forma víčko, z firmy: xxxxxxxx, zastoupení panem: xxxxxxxx
 19 Dílce, z firmy: xxxxxxxx, zastoupení panem: xxxxxxxx

Suroviny/Kč	NormáleCena/Kč	NástrojeCena/Kč	OperaceCena/Kč	SmluvníCena/Kč	Vyčerpáno v Kč	Vyčerpáno v %
4203	0	15000	64996,7	155000	84199,7	54,3
*						

Obr. 36 Vyfiltrovaná data podle kritéria kalendáře a rozpracovanosti zakázky.

Opět je možné vygenerovat graf (Obr. 37), který názorně zobrazuje využití a nevyužití nákladů na vybranou zakázku. Pokud jsou náklady přečerpány, graf zobrazí zápornou částku, která je nad smluvní cenou - jako „Náklady navíc“ - a uvede skutečnou a smluvní cenu. Všechny generované grafy je možné tisknout stisknutím tlačítka „Tisk grafu“. V této části aplikace opět platí, že je aplikace ošetřena tak, aby nedocházelo k chybám. Pokud někde některá data chybí tak, že by to vedlo k chybě, aplikace se s tímto problémem vypořádá.



Obr. 37 Zobrazení nevyužitých nákladů v poměru k již vyčerpaným.

Pod záložkou „Podrobnější informace o Z.“ nalezneme podrobnější informace (Obr. 38) o jednotlivých výrobních operacích, kooperacích, dílcích nebo zálohách. Zde jsou data z datového úložiště vyfiltrována například u výrobních operací tak, že jsou sečteny hodiny a ceny vzhledem k zaměstnanci, výrobní operaci a dílci. Například v datovém úložišti můžeme mít položku Kopyto, frézování, pan „X“ několikrát. Zaměstnanec může na jednom dílci frézovat např. v pondělí, ve čtvrtek a v pátek. Zde se informace zobrazí jako jedna položka. Tímto způsobem jsou filtrovaná i ostatní data záložky. Takto je docíleno přehledné tabulky s potřebnými informacemi, které mohou sloužit jako přehled pracovníků a cen pro obchodní partnery firmy.

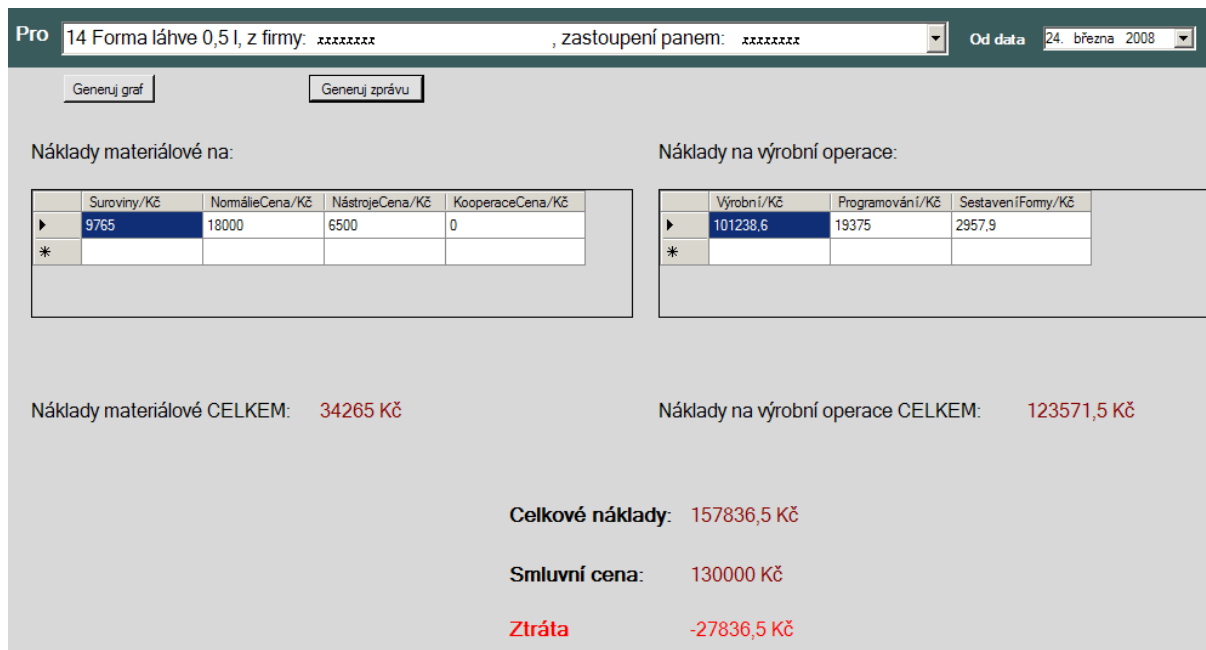
Pro 9 forma čelo kulaté, z firmy: xxxxxxxx, zastoupení panem: xxxxxxxx

Příslušné informace o jednotlivých:

NázevDílce	NázevOperace	Příjme	hSazby	CenaČinnosti/Kč
deska pravá	Montáž	Válek		646,7
deska pravá	Vitání	Čemý	1,52	606,7
táhlo	Broušení	Válek	0,55	220
táhlo	Soustružení	Šišák	6,75	2700
vložka pevná	Frézování	Pšeja	6,65	3990
vložka pevná	Mechanik	Čemý	1,32	526,7
vložka pevná	Montáž	Čemý	8,37	3346,7

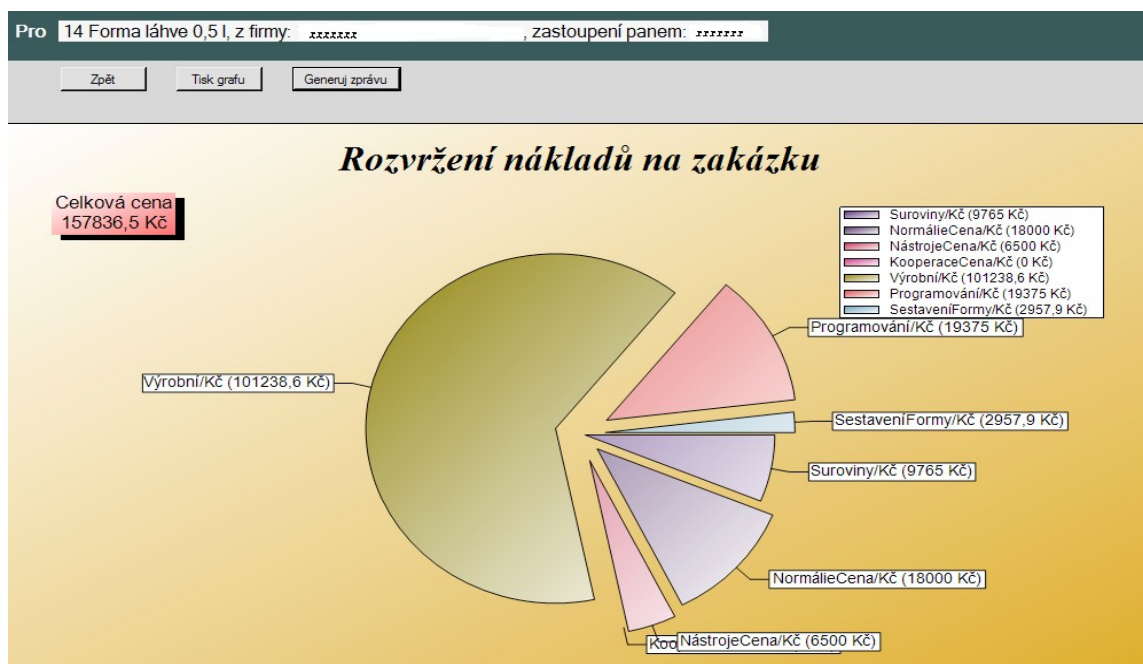
Obr. 38 Část vyfiltrované tabulky.

Ekonomická analýza rozpracované výroby a podrobnější informace o zakázkách mohou také posloužit pro potřebu uživatele sledovat určité specifické informace v průběhu výroby.



Obr. 39 Rozvržení ekonomiky pro lahev 0,5 l.

Pro sledování sumace nákladů na jednotlivé zakázky slouží záložka „Z. dle ekonomiky“ (Obr. 39). Tyto informace mohou být opět zobrazeny do grafu (Obr. 40) nebo si uživatel může vygenerovat zprávu do dokumentu „html“ tlačítkem „Generuj zprávu“ (později zahrnuto do většiny záložek analýz). Zde uživatel může vidět, jak velkou část z hlediska ekonomiky zaujímají položky nakoupené od dodavatelů a položky odpracované v rámci vlastní firmy. Záložka rovnou zobrazí, zda je daná zakázka zisková nebo ztrátová. Na tomto základě může uživatel sledovat, které formy je nutné podrobněji analyzovat například pomocí časové analýzy (kapitola 4.1) v závislosti na zisku nebo ztrátě.



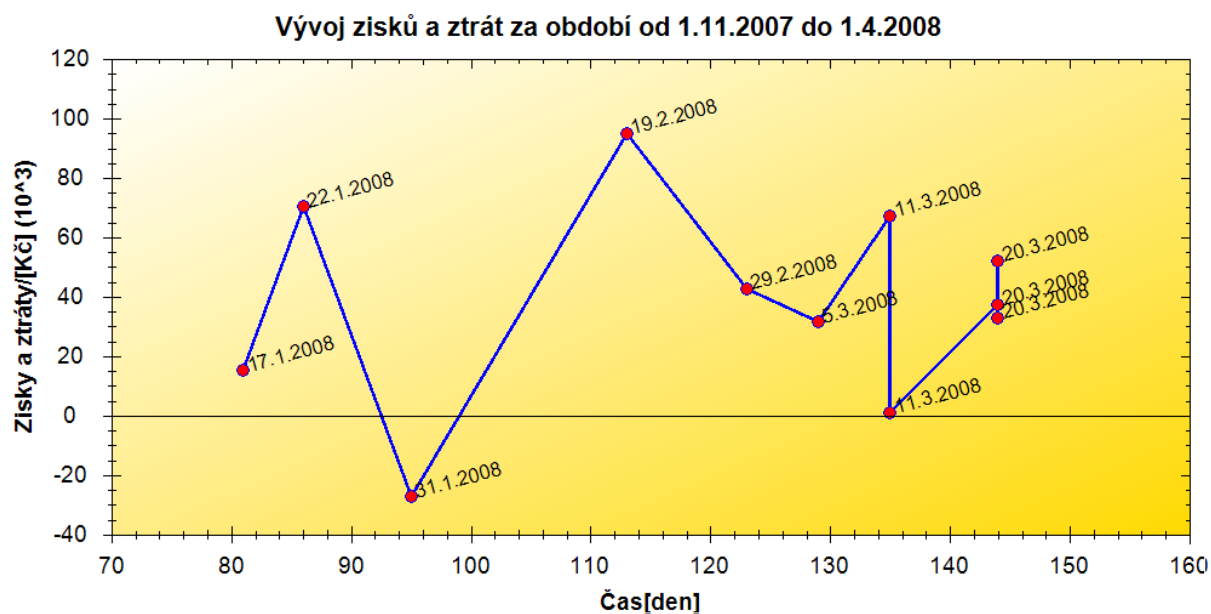
Obr. 40 Graf rozvržení financí na lahev 0,5 l.

Nyní se přesuneme k nejdůležitější části této práce. Vzhledem k tomu, že hodinové ocenění jednotlivých výrobních operací (frézování, soustružení, jiskření, apod.) byly jen odhadnuty, mohou nám všechny zakázky vyjít nadhodnoceny, podhodnoceny nebo správně. Důležitým předpokladem pro ekonomickou analýzu je skutečnost, že firemní zisk je na hodnotě 0 Kč, tj. výdaje firmy se rovnají příjmům. To je dáno vysokým leasingovým zatížením firmy na splácení aut, strojů a budovy, ve které firma sídlí.

Pro zjištění skutečného stavu slouží záložka „Prognóza“. Z Obr. 41 a grafu (Obr. 42) je patrné, že náklady na výrobní operace byly hrubě podhodnoceny, protože suma zisků udává hodnotu 416 554 Kč, a to je vzhledem k nulovým ziskům firmy nesmyslná hodnota. Na základě jednoduchých úvah založených na nástrojích matematiky bylo vypočteno, že náklady na výrobní operace musí být navýšeny asi o 93,75 %. Jak je patrné z Obr. 40 pro výpočet tohoto navýšení stačí zadat skutečné zisky do kolonky za „Předpokládaný zisk“ a stisknout tlačítko „Odhadni“ pro odhad navýšení hodinových cen výrobních operací.

JménoFirmy	KontaktníOsoba	PředmětObjednávky	TermínPředání	Smluvní cena	Celková cena	Zisk/Ztráta	Hodnota(Zisk/Ztráta)
		rám ...	17.1.2008	60000	44891	Zisk	15109
		spodek ...	22.1.2008	125000	54616,7	Zisk	70383,3
		ALEX 1L ...	31.1.2008	25000	52194,7	Ztráta	-27194,7
		Kanystr 5L ...	19.2.2008	280000	184907,7	Zisk	95092,3
		foma čelo přístroje ...	29.2.2008	106000	63296,3	Zisk	42703,7
		foma pouzdra repro...	5.3.2008	100000	68571	Zisk	31429
		foma čelo kulaté f...	11.3.2008	126000	58677,3	Zisk	67322,7
		Upínací deska ...	11.3.2008	16000	14963,3	Zisk	1036,7
		foma zátky ...	20.3.2008	120000	82538,7	Zisk	37461,3
		Forma láhve 0,5l ...	20.3.2008	130000	98028,3	Zisk	31971,7
		Forma láhve 0,3l ...	20.3.2008	120000	68761	Zisk	51239

Obr. 41 Vývoj ztrát a zisků za určité období.



Obr. 42 Graf zobrazující vývoj ztrát a zisků za vybrané období.

K tomuto účelu slouží funkce pod tlačítkem „Zpřesni“ v záložce „Vstupní data -> Výrobní operace na dílcích“. Samozřejmě uživatel nemusí počítat hodnoty po navýšení, stačí, když zadá o

kolik procent chce položky navýšit nebo snížit. Z Obr. 43 je patrné, že zpřesňujeme jen tři hodnoty, i když máme asi deset výrobních operací. Je to v důsledku toho, že některé operace jsou si cenou za hodinu rovny. Po provedených cenových změnách jsme se dostali na nulový zisk, přesněji ztrátu -18,5 Kč, ale to je v kontextu 700 položek zaokrouhlovací chyba, které jsme se dopustili při výpočtu, o kolik procent musíme zvýšit náklady. Jak je patrné z Obr. 44, pokud bychom chtěli v dalším období zisk 100 tisíc, pak za předpokladu, že máme dostatečně reprezentativní vzorek, musíme navýšit stávající ceny výrobních operací o 11,59 %.

Operace	IDzaměstnanec	PočetDleDruhuSaz	CenaČinnosti/Kč	Povol zápis ceny
	3	3	1200	<input type="checkbox"/>
	1	0,016666	9,999600410461...	<input type="checkbox"/>
	7	0	0	<input type="checkbox"/>

Upřesni odhad cen výrobních operací

Předchozí ceny	Nové ceny	
800	<input type="text" value="1550"/>	<input type="button" value="zvedni/sniž o %"/> <input type="text" value="93,75"/>
600	<input type="text" value="1163"/>	
400	<input type="text" value="775"/>	
<input type="button" value="Storno"/> <input type="button" value="Ulož a přepočítej"/>		

1	0,766666650772...	460	<input type="checkbox"/>
1	0,283333331346...	170	<input type="checkbox"/>

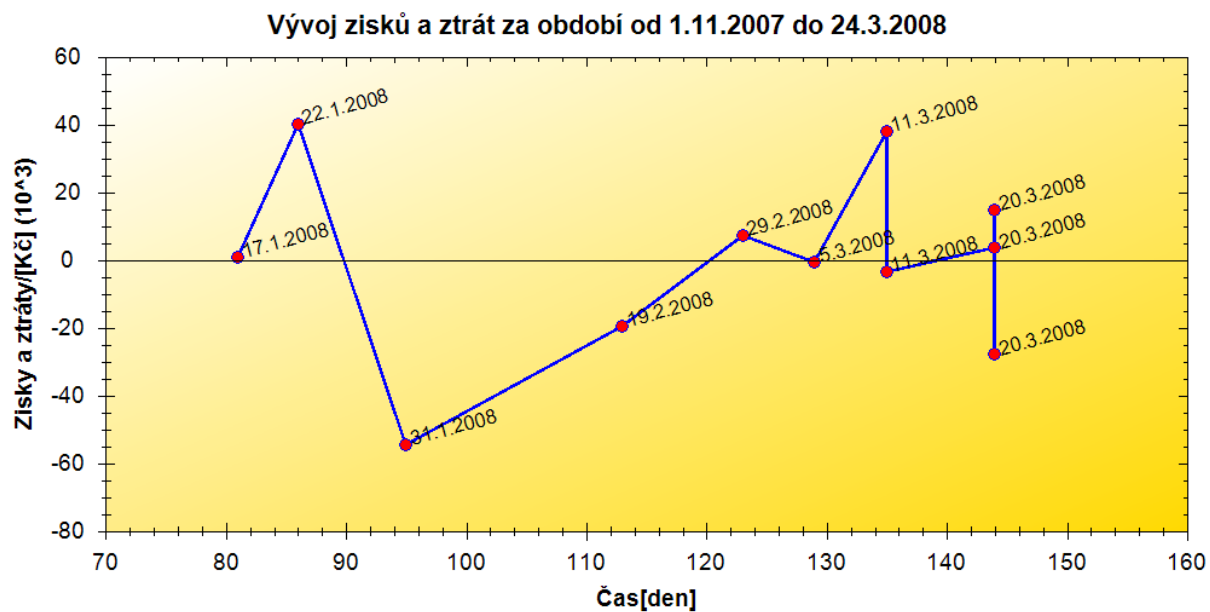
Obr. 43 Tlačítko pro zpřesnění cen výrobních operací.

Graf (Obr. 45) se nám celkem logicky posunul více pod osu udávající zisk nebo ztrátu v Kč. Samozřejmě se neposunul o nějakou konstantu, jelikož se nejedná o funkční závislost. Z Obr. 44 může uživatel zjistit, které formy byly ztrátové, které zase výdělečné, a na základě těchto výsledků rozhodnout, které typy forem se v budoucnu vyplatí vyrábět a které nikoli. I když se jedná o kusovou výrobu a každá forma je originálem, přesto jsou si typově podobné a na těchto základech lze zracionalizovat případné ceny budoucích zakázek.

Generuj graf		Celková ztráta: -18,5 Kč		Předpokládaný zisk: 100000	Odhadni	Navyšte o: 11,59 %	
JménoFirmy	KontaktníOsoba	PředmětObjednávky	TermínPředání	Smluvní cena	Celková cena	Zisk/Ztráta	Hodnota(Zisk/Ztráta)
		rám ...	17.1.2008	60000	59150,4	Zisk	849,6
		spodek ...	22.1.2008	125000	84622,9	Zisk	40377,1
		ALEX 1L ...	31.1.2008	25000	79360,3	Ztráta	-54360,3
		Karystr 5L ...	19.2.2008	280000	299453,9	Ztráta	-19453,9
		forma čelo přístroje ...	29.2.2008	106000	98540,1	Zisk	7459,9
		forma pouzdra repro...	5.3.2008	100000	100521	Ztráta	-521
		forma čelo kulaté f...	11.3.2008	126000	87827,3	Zisk	38172,7
		Upínací deska ...	11.3.2008	16000	19241,5	Ztráta	-3241,5
		forma zátky ...	20.3.2008	120000	116368	Zisk	3632
		Forma láhve 0,5l ...	20.3.2008	130000	157806,5	Ztráta	-27806,5
		Forma lahve 0,3l ...	20.3.2008	120000	105126,6	Zisk	14873,4

Obr. 44 Skutečný vývoj ztrát a zisků za uvedené období.

Vzhledem k tomu, že výroba byla sledována plných 5 měsíců, můžeme tvrdit, že odhad navýšení cen za výrobní operace o 93,75 % je s malou chybou. Pokud se bude sledovat výroba i nadále tak, abychom měli vzorek jednoho kalendářního roku, je možné odhadovat, že navýšení cen se změní maximálně v řádu procent. Nicméně s větším reprezentativním vzorkem dat budeme schopni vytvářet přesnější závěry.



Obr. 45 Graf zobrazující skutečný vývoj ztrát a zisků za vybrané období.

5 ZÁVĚR

Práce se věnovala jak technickým problémům v oblasti řešení systému interního sledování výroby, tak konkrétnímu řešení ve firmě Octopus, spol. s r.o.

Díky této práci budou v daném provozu neziskové typy zakázek, při každé nové objednávce, vyřazeny z výběru budoucí výroby. Tato a další rozhodnutí je možné provádět na základě ekonomické a časové analýzy výroby zahrnuté v aplikaci.

Samotné ceny forem nejsou podhodnoceny ani nadhodnoceny, jelikož vychází z dlouholeté praxe daného výrobního závodu, a jejich cena je také do značné míry dána cenami konkurence. Firma tedy své zisky nemůže zvýšit „uměle“. Formy se mohou lišit konstrukcí, a tedy i pracností. Výstupy z tohoto projektu umožňují dané firmě rozhodnout, které typy forem nebo dílců nevyrábět nebo u kterých je nutno použít jiné výrobní postupy. V některých případech může pracnost vyjít velká na základě vzniklých chyb při výrobě nebo na základě špatného konstručního návrhu a následné nutnosti vyrábět některé dílce znovu. I když se jedná o provoz s dlouhodobou tradicí a tyto chyby se stávají minimálně, přesto by se s nimi mělo do budoucna počítat a malé navýšení cen forem v řádů procent nemusí nutně vést k odrazení zákazníka ke konkurenci a může také znamenat navýšení zisku firmy.

V současné době je v přípravě docházkový systém, který bude v nejbližším čase také zaveden a navázán na stávající systém sledování výroby, jak již bylo zmíněno v kapitole 3.4.1.

SEZNAM POUŽITÉ LITERATURY

- [1] KODYS, spol. s r.o.. *Čárový kód* [online]., 2006 [cit. 2008-02-06].
Dostupné z: http://www.carove-kody.cz/?_ch_sl=google1
- [2] Wikipedie. *Čárový kód* [online]., 05.01.2008 [cit. 2008-02-06].
Dostupné z: http://cs.wikipedia.org/wiki/%C4%8C%C3%A1rov%C3%BD_k%C3%B3d
- [3] Cosmotron Bohemia s.r.o.. *Čárové kódy* [online]., [2008-02-06].
Dostupné z: <http://www.cosmotron.cz/index.php?section=ck>
- [4] Šmejkal, Ladislav. Snímače čárových kódů – přehled trhu. *Automatizace • ročník 50 • číslo 11 • listopad 2007* [online]. [cit. 2008-02-06].
Dostupné z: <http://www.automatizace.cz/article.php?a=1952>
- [5] BELOS Trade, s. r. o.. *Čárový kód* [online]., [cit. 2008-02-06].
Dostupné z: <http://www.carovykod.com/index.php?id=2&lang=cz>
- [6] Benáček, M. *Možnosti identifikace a monitorování pohybu zboží a osob*. Brno, 2007. 51 s.
Bakalářská práce na ÚAI FSI VUT v Brně. Vedoucí bakalářské práce Ing. PAVEL HOUŠKA, Ph.D. Dostupné z: http://autnt.fme.vutbr.cz/SZZ/2007/BP_Benacek.pdf
- [7] Elektroland.cz. *Čtečka ČK CCD-MT9062, 90mm, RS232 černá.*, [cit. 2008-02-06].
Dostupné z: <http://www.elektroland.cz/ctecka-ck-ccd-mt9062-90mm-rs232-cerna/d-99771/>
- [8] EZ One Shot, CCD Scanner. *Programovací příručka*.
- [9] Wikipedie. *RS-232* [online]., 01.01.2008 [cit. 2008-02-06].
Dostupné z: <http://cs.wikipedia.org/wiki/RS-232>
- [10] Wikipedie. *CCD* [online]., 20.01.2008 [cit. 2008-02-06].
Dostupné z: <http://cs.wikipedia.org/wiki/CCD>
- [11] Pichlík, Roman. *Třívrstvá architektura v kostce I.* [online]., 11.11.2004 [cit. 2008-02-06].
Dostupné z: http://www.sweb.cz/pichlik/archive/2004_11_07_archive.html
- [12] Wikipedie. *Fotoelektrický jev* [online]., 03.12.2007 [cit. 2008-02-08].
Dostupné z: http://cs.wikipedia.org/wiki/Fotoelektrick%C3%BD_jev
- [13] Polášek, Marek. Databáze z hlediska podnikových informačních systémů. *IT Systém* [online]. 7-8/2003. [cit. 2008-02-09]. Dostupné z:
<http://www.systemonline.cz/clanky/databaze-z-hlediska-podnikovych-informacnich-systemu.htm>
- [14] Kouba, Tomáš. *Pro web aplikaci je lepší Java nebo .NET?* [online]., 15.5.2006. [cit. 2008-02-09].
Dostupné z: http://www.neo.cz/~tomas/java.net/archive/2006_05_01_archive.html
- [15] Šeda, Jan. *J2EE, .NET a vývoj rozsáhlých systémů.* [online]., 11.02.2003. [cit. 2008-02-08]
Dostupné z: <http://interval.cz/clanky/j2ee-net-a-vyvoj-rozsahlych-systemu-2/>
- [16] Nagel, Christian. *C# 2005 : programujeme profesionálně*. První vydání. Brno: Computer Press, 2006. 1398 s. ISBN 80-251-1181-4

- [17]Microsoft Corporation. *Visual Studio 2008 Express Edition* [online]., [cit. 2008-02-09].
Dostupné z: <http://www.microsoft.com/cze/msdn/produkty/vstudio/ExpressEditions/default.aspx>
- [18]Wikipedie. *Network address translation* [online]., 6.2.2008. [cit. 2008-02-09].
Dostupné z: <http://cs.wikipedia.org/wiki/NAT>
- [19]LANius s.r.o. *Základní informace o MS SQL Server 2005 Express Edition* [online]., [cit. 2008-02-13]. Dostupné z: <http://www.lanius.cz/sql2005.htm>
- [20]Kolektiv autorů (Simon Robinson, K. Scott Allen, Ollie Cornes, Jay Glynn, Zach Greenvoss, Burton Harvey, Christian Nagel, Morgna Skinner, Karli Watson). *C# Programujeme profesionálně*. První vydání. Brno : Computer Press, 2003. 1130 s. ISBN 80-251-0085-5
- [21]Roupec, Jan. *Počítačové sítě*. VUT v Brně FSI. listopad 2002. 83 s.
- [22]Wikipedie. *Router* [online]., 1.2.2008. [cit. 2008-02-09].
Dostupné z: <http://cs.wikipedia.org/wiki/Router>
- [23]Microsoft Corporation. *How to configure SQL Server 2005 to allow remote connections* [online]., [cit. 2008-02-09]. Dostupné z: <http://support.microsoft.com/kb/914277>
- [24]R.Stanek, William. *Microsoft SQL Server 2005 Kapesní rádce administrátora*. První vydání. Brno: Computer Press, 2006. 543 s. ISBN 80-251-1211-X
- [25]Hernandez, Michael J. *Návrh databází*. První vydání. Praha : Grada, 2006. 408 s. ISBN 80-247-0900-7
- [26]Čermák, Pavel. Červinková, Petra. *Odmaturuj z matematiky*. Třetí vydání. Brno : Didaktis, 2004. 208 s. ISBN 80-7358-014-4