

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

APLIKACE PRO EVIDENCI A ŘÍZENÍ STROJÍRENSKÉ VÝROBY

APPLICATION FOR PRODUCTION EVIDENCE AND MANAGEMENT

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

AUTOR PRÁCE
AUTHOR

BC. PAVEL KOUDELA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. RADEK POLIŠČUK, PH.D

BRNO 2011

ZADÁNÍ ZÁVĚREČNÉ PRÁCE

(na místo tohoto listu vložte originál a nebo kopii zadání Vaš práce)

ABSTRAKT

Tématem práce je analýza řízení výroby ve společnosti SLB spol, s r. o. a následný návrh podkladů pro implementaci podpůrné aplikace. Mezi využívané technologie patří strukturovaná analýza, objektově orientované programování, webové technologie a databáze. Serverová i klientská část aplikace jsou platformě nezávislé a využívají moderní programovací přístupy.

ABSTRACT

The aim of this thesis is a production management analysis in company SLB and creating design for supportive application implementation. Used technologies include structured analysis, object oriented programming, web technologies and a relational database. Server as well as client applications are platform independent and use modern programming techniques.

KLÍČOVÁ SLOVA

MES, PHP, MySQL, AJAX, strukturovaná analýza, strojírenská výroba, informační systém

KEYWORDS

MES, PHP, MySQL, AJAX, structured analysis, engineering production, information system

BIBLIOGRAFICKÁ CITACE

KOUDELA, P. *Aplikace pro evidenci a řízení strojírenské výroby*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2011. 57 s. Vedoucí diplomové práce Ing. Radek Poliščuk, Ph.D..

PROHLÁŠENÍ

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a odbornou literaturu využitou při zpracovávání práce řádně cituji s uvedením úplného odkazu.

Bc. Pavel Koudela

PODĚKOVÁNÍ

Na tomto místě bych rád poděkoval Ing. Radku Poliščukovi, Ph.D za ochotu a cenné rady, které mi pomohly při zpracování této práce. Dále děkuji vedení společnosti SLB spol. s r. o. za vytvoření příležitosti k tvorbě popisovaného systému a za spolupráci při analýze, vývoji a testování vyvinuté aplikace. Rád bych také poděkoval ostatním členům vývojového týmu za spolupráci a mé rodině za podporu při dokončování práce.

Obsah:

	Zadání závěrečné práce.....	3
	Abstrakt.....	5
	Bibliografická citace.....	7
	Prohlášení.....	9
	Poděkování.....	11
1	Úvod.....	15
2	Způsoby řízení strojírenské výroby.....	17
2.1	Obcné způsoby řízení.....	17
2.2	Způsob řízení výroby ve zkoumané společnosti.....	18
3	Analýza.....	19
3.1	Požadavky.....	19
3.1.1	Hardware.....	20
3.1.2	Software.....	20
3.1.3	Knihovny.....	20
3.2	Diagram datových toků	21
3.3	Entitně relační diagram.....	29
3.3.1	Tabulka pro evidenci zakázek.....	30
3.3.2	Tabulky součástí.....	31
3.3.3	Tabulky technologických postupů.....	33
3.3.4	Vazební tabulka mezi zakázkou, součástí a technologickým postupem.....	35
3.3.5	Pomocné tabulky.....	35
3.4	Rozdělení funkčních bloků.....	35
3.5	Zhodnocení.....	39
4	Implementace.....	41
4.1	PHP knihovny.....	41
4.2	Knihovny a aplikace v jazyce JavaScript.....	42
4.3	Adresářová struktura.....	43
4.4	Konfigurace systému.....	44
4.5	Zabezpečení.....	44
4.6	Chybová a informační hlášení.....	45
4.7	Indexace tabulek a vyhledávání.....	45
4.8	Moduly obsluhující zadávání výroby.....	45
4.9	Sběr dat z výroby.....	47
4.10	Sledování zakázky.....	48
4.11	Ganttovy grafy zobrazující postup výroby.....	48
4.12	Časové rozvrhy operací.....	49
4.13	Výpočet nákladů výroby.....	50
5	Nasazení aplikace.....	53
5.1	Nasazení systému v řízení výroby.....	53
5.2	Dílenské terminály.....	54
6	Závěr.....	55
	Seznam použité literatury.....	57

1 ÚVOD

Cílem této práce byla analýza stavu řízení a procesů ve výrobě speciálních ložisek, vytvoření popisu probíhajících procesů a příprava podkladů nutných k implementaci, včetně její realizace. Zadavatelem úkolu bylo vedení společnosti SLB spol. s r. o., jejíž sídlo se nachází v Brně. Na výsledném vývoji se dále podílela programátorka Jana Baarová a grafik Martin Kořan.

Společnost SLB, spol. s r. o. se zabývá vývojem a výrobou speciálních ložisek, konstrukcí vstřikovacích forem a produkcí plastových součástí. Založena byla v roce 1991 pětici společníků s rozsáhlými znalostmi z vývoje a výroby ložisek. V roce 2002 získala firma SLB certifikát řízení kvality ISO 9001. Systém řízení kvality obsahuje popisy procesů a podle nich probíhá vývoj, výroba a prodej na dvou pracovištích. Ke konci roku 2010 společnost zaměstnávala 29 zaměstnanců, z toho 15 THP (technicko-hospodářští pracovníci) a 14 pracovníků na dělnických pozicích.



Obr. 1 – Ukázka produkce společnosti SLB spol. s r. o.

Mezi hlavní nedostatky v řízení výroby zkoumaného podniku patří nedostatečné využití informačních technologií. Veškeré dokumenty a procesy jsou evidovány pomocí tabulkových a textových editorů a komplexní pohled na obsazenost výrobních kapacit není k dispozici. Data z výroby jsou sbírána do tabulek na výrobních výkresech, ale kromě archivace dosud nebyla dále centrálně zpracovávána a analyzována. Z této skutečnosti vyplývá cíl práce, kterým je nasazení aplikace pro podporu řízení výroby implementující funkce kapacitního plánování, výpočty nákladů, archivaci pracovních postupů a která bude umožňovat napojení na stávající ekonomický software Company Manager.

Důležitým bodem byla organizace práce ve zmíněném tříčlenném týmu. Mým úkolem na popisovaném systému byla kompletní analýza problému včetně zpracování podkladů pro implementaci aplikace. Po dokončení analytické části práce jsem se dále zabýval implementací autentizace a autorizace uživatelů, sledovacích karet, Ganttových grafů a seznamů úkolů. Na tuto fázi práce navázalo zpracování modulů pro nahrávání externích dokumentů, napojení aplikace na dokumentový server ve společnosti a tvorba konektorů na ekonomický software. Dále jsem do aplikace nasazoval zvolené knihovny a programoval nové metody objektů, které byly realizovány ostatními členy týmu. Slečna Baarová měla na starosti začlenění knihovny jQuery a tvorbu modulů starajících se o zadávání výroby. Grafickou stránku aplikace navrhoval pan Kořan s důrazem na jednoduchost a preferované dotykové ovládání aplikace.

Druhá kapitola se zabývá krátkým pohledem na skutečný stav na poli řízení výroby. Jaký je okolními společnostmi využíván software a jaké jsou důsledky těchto metod. Třetí kapitola zahrnuje požadavky zadavatele a popisuje metody použité k analýze aktuálního stavu. Výsledkem je popis aplikace z pohledu datového, procesního i implementačního ve formě návrhu modulů se základními funkcemi. Čtvrtá kapitola seznamuje s použitými nástroji a výsledným provedením jednotlivých částí programu, na což navazuje kapitola s popisem postupného nasazování vyvinutého systému ve společnosti a problémy vznikající při jeho zavádění. Poslední kapitola shrnuje současné a do budoucna plánované funkcionality. Tato doporučení na rozšíření aplikace jsou výsledkem analýzy aktuálního způsobu provádění vybraných firemních procesů.

2 ZPŮSOBY ŘÍZENÍ STROJÍRENSKÉ VÝROBY

V rámci této práce je zkoumáno řízení výrobních úseků u malých podniků (do 50 zaměstnanců). Aplikované způsoby řízení jsou diametrálně odlišné a obvykle vycházejí ze základních bodů, jejichž výskyt je společný pro většinu výrobních podniků. Nejdůležitějšími prvky v tomto ohledu jsou:

- finanční situace firmy,
- vztah vedení k informačním technologiím a orientace v této oblasti,
- znalosti vedoucích pracovníků,
- přítomnost dílenských mistrů,
- samostatnost a zkušenosti zaměstnanců.

2.1 Obecné způsoby řízení

Při práci v oblasti výroby ložisek jsem měl možnost pozorovat systémy řízení v drobných a malých podnicích. Ve většině drobných podniků, které jsem navštívil, probíhá výroba a její organizace na základě podpory tabulkových editorů, což je obvykle dáno jejich velikostí a strukturou. U malých podniků dochází k rozložení podílu řízení výroby mezi tabulkové procesory, podnikové systémy ERP (Enterprise Resource Planning, [1])/MES (Manufacturing Execution Systems, [2]) a systémy programovanými na míru. Střední a větší společnosti využívají výhradně posledních dvou možností řízení výroby.

Intuitivní řízení

Velká část drobných podniků řídí výrobu bez podpory informačních technologií. Toto se týká především podniků s jedním až pěti zaměstnanci. Z hlediska krátkodobého plánování se jedná o akceptovatelnou metodu, pro větší objem výroby už ale jde o nevhodný přístup.

Řízení s podporou tabulkových procesorů

Tabulkové procesory jsou často využívaným pomocníkem při plánování výroby téměř jakéhokoli typu. V běžných kancelářských provozech jsou využívány k evidenci kontaktů, nabídek, zakázek, poptávek a dalších obdobných agend. V rámci tabulkových editorů lze realizovat i složitější výpočty a zobrazení. Příkladem může být zpracování statistik z ekonomického pohledu na firmu nebo na konkrétní výrobek, ale také různé grafické a textové způsoby plánování třeba i s využitím Ganttových grafů. Problémy těchto řešení jsou především nesoudržnost informací, redundance, uzamykání souborů a obtížně realizovatelné souhrnné pohledy na celkový stav. Jednoznačným kladem je dnes neomezená dostupnost těchto řešení, která lze realizovat i ve volně dostupných kancelářských balících.

Řízení pomocí ERP/MES

Ekonomické systémy, mající implementované výrobní a plánovací moduly s podporou SQL datových úložišť a architekturou client-server, jsou cenově náročná řešení. Vzhledem k povaze tohoto softwaru je evidentní, že s ohledem na univerzalitu bude obsahovat velké množství, pro konkrétního uživatele až irelevantních, parametrů, přičemž i tak často nastává situace, že některý údaj zákazník v agendě postrádá. Výhody „krabicových“ systémů jsou dány faktem, že stejný software je používán mnoha dalšími společnostmi, zatímco obdobné řešení na míru bývá vyvíjeno téměř od počátku. Podnikových systémů tohoto typu je na trhu velké množství. Je možné vybírat z nabídky nadnárodních i českých společností jako jsou například SAP, MS Dynamics, Epicor, Helios Green, Abbra Gx atd.

Podnikový systém na míru

Obvykle prvním aspektem, při zvažování individuálního podnikového systému, je pohled na náklady spojené s vývojem zakázkové aplikace. Tyto investice jsou podmíněny především velikostí společnosti a rozmanitostí požadované funkcionality. V mnoha případech mohou jednoduchá MES řešení na bázi internetových WWW překonávat univerzální proprietární systémy jak z pohledu použitelnosti, tak i z hlediska cenového.

Zákaznické aplikace se vyznačují následujícími prvky:

- zpracovávají přesně to, co zákazník požaduje,
- přidání nové funkcionality je realizováno úpravou nebo napsáním nového modulu,
- zákazník se stává vlastníkem všech zdrojových kódů a může s nimi libovolně nakládat,
- nově vyvinutý systém zpravidla obsahuje velké množství chyb,
- cenové kalkulace se mohou po upřesnění specifikace změnit,
- při nedostatečné podpoře nemusí být nasazení úspěšné.

2.2 Způsob řízení výroby ve zkoumané společnosti

Základem všech metod řízení výrobního podniku je přesná definice pracovních procesů, technologických postupů a delegace pravomocí a zodpovědnosti. Společnost SLB spol. s r. o. je držitelem certifikátu dle ISO 9001 od roku 2002 a zavedené pracovní procesy a postupy odpovídají jednotlivým popisům. Veškeré toky dat i prostředků, které jsou ve zmíněných procesech specifikovány, jsou i reálně prováděny. Pracovníci na jednotlivých pracovištích si ale zároveň vedou svoje agendy, které ještě nebyly v době zkoumání systému procesně zavedeny.

Ve společnosti SLB je aktuálně využíván model „Řízení s podporou tabulkových procesorů“. Ekonomická stránka firmy je svěřena aplikaci Company Manager od COM-SYS SOFTWARE, přičemž hlavním požadavkem vedení je zachování tohoto programu při zavádění MES systému. Mimo působnost ekonomické aplikace jsou externě vedeny agendy poptávek, nabídek, objednávek, dopisů, protokolů a potvrzení objednávek. Výroba jako taková je plánována na základě tabulky zvané „Harmonogram výroby“, upřesňujících časových rozvrhů a informací od zákazníka. Veškeré dokumenty jsou uloženy na souborovém serveru a jsou dostupné všem odpovědným osobám. Zároveň ale platí nevýhody zmíněné v popisu této metody.

3 ANALÝZA

Cílem analytické etapy vývoje bylo vytvoření podrobné specifikace zkoumaného systému. Tento popis je následně použit vývojáři, kteří na projektu pracují.

Základními prvky, které byly při analýze řízení výroby v SLB využity, jsou (kromě specifikace požadavků) diagram datových toků a entitně relační diagram. Při zkoumání ve společnosti probíhající procesy byly využity konzultace i vlastní zkušenosti z chodu této firmy. Jako zdroj zásad při dotazování a postupů při tvorbě specifikace byla využita Yourdonova metoda strukturované analýzy, která obsahuje doporučení a postupy pro tvorbu logických modelů a jejich provázání s datovými modely. Tyto výstupy byly posléze podrobeny dalšímu rozboru, který měl za cíl stanovení funkčních bloků systému. Výsledkem je návrh modulární struktury, která popisuje vstupně/výstupní rozhraní a napomáhá při vývoji systému.

Většina procesů, které momentálně ve společnosti probíhají, je evidována za pomoci tabulek vedených v tabulkovém procesoru, případně formou textových dokumentů předávaných odpovědným pracovníkům. Spotřebovaná práce je evidována na zadních stranách výrobní dokumentace a následně archivována. Tento stav je pro společnost neuspokojivý. Hlavním nedostatkem je v tomto případě nemožnost kontrolovat kapacitní vytížení strojů a na základě těchto informací rozhodovat o případných kooperacích. Úkolem nově vyvíjeného systému tedy bude evidence spotřebované práce, na základě které se budou sestavovat kapacitní plány pro opakovanou výrobu. U nových zakázek budou časy nutné na realizaci úkolů zadávány na základě zkušeností z obdobné produkce. Nasbírané údaje také poslouží jako nástroj pro cenové kalkulace a tvorbu nabídek. Společnost je svými zákazníky neustále tlačena ke snižování cen svých produktů, a proto je nezbytně nutné znát hranici nákladů, pod kterou se výroba stává nerentabilní.

3.1 Požadavky

Na aplikace podporující řízení výroby jsou kladeny rozdílné požadavky podle charakteru daného procesu. Popisovaný systém bude evidovat operace probíhající ve strojírenské výrobě.

Základní požadavky jsou shrnuty do následujících bodů:

- evidence zakázek pravidelně synchronizovaná s aplikací Company Manager,
- evidence výrobků a jejich částí,
- tvorba technologických postupů (TP) s vazbou na konkrétní zakázku,
- generování seznamu operací (úkolů) z TP,
- tvorba časových rozvrhů pro jednotlivé typy operací (soustružení, frézování, vrtání, otvorové broušení atd.),
- přebírání úkolů zaměstnanci pomocí dotykových panelů,
- evidence spotřebované práce pro jednotlivé operace přímo na pracovištích s archivací v databázi pro jednoduché vyhledávání a tvorbu statistik,
- evidence zbylých součástí po ukončení zakázky,
- historie – uchovávání veškerých výrobních údajů. V případě reklamací umožní rychle a jednoduše dohledat veškeré činnosti, které byly při plnění zakázky realizovány,
- rychlé začlenění opakované výroby,
- využití stávajícího serveru s operačním systémem.

Ve firmě je kromě obvyklého kancelářského balíku využíván ekonomický software Company Manager a konstrukční programy Autocad a Inventor od společnosti Autodesk. V ekonomickém softwaru je realizována evidence zakázek, kterou bude systém řízení využívat. Tato aplikace pracuje s databázovými soubory typu dBase, které je možné s pomocí programu dbf2mysql importovat do databázového serveru MySQL. I přes to, že přidávání výrobních zakázek probíhá většinou jen jednou za den, bude pro udržení datové konzistence využívána služba CRON a importovací příkaz bude aktivován každých 30 minut. Celý proces je jednosměrný a veškerá data ve zdroji budou vyvíjené aplikaci přístupná v režimu „pouze pro čtení“.

Popis procesu nakládání s výrobní dokumentací stanovuje povinnost konstrukčního oddělení

vydat nové výkresy pro každou přijatou zakázku. Tímto se eliminují potenciální problémy spojené s neaktuálností výrobní dokumentace.

Prioritou při výběru technologií a následné implementaci je volba otevřených řešení, aby případně vybraná proprietární technologie a nebo nedostupný hardware nemohly jakýmkoliv způsobem omezit modifikovatelnost a rozšiřitelnost aplikace.

3.1.1 Hardware

Při volbě hardwaru (HW) byl kladen důraz na poměr kvalita/cena a na jednoduchou obnovu těchto zařízení v následujících letech. Z počátku byly zvažovány průmyslové počítače s dotykovou obrazovkou. Reference z provozů ve firmě SLB však naznačují, že nasazení průmyslových počítačů by v porovnání s životností a cenou běžného spotřebního zboží nebylo cenově efektivní. V popisovaném provozu dlouhodobě pracují běžné kancelářské počítače bez známek větší poruchovosti.

Jako vhodný HW pro dílenské terminály byl zvolen MSI Wind Top AE2040. Počítač je dostatečně výkonný, je vybaven 20“ velkou dotykovou obrazovkou s optimálním rozlišením a je možné ho zakoupit s adaptérem VESA 100 x 100 pro připevnění na zeď.

Druhou částí hardwarového vybavení jsou tablety určené pro vedoucí výroby. Trh s tablety se v současnosti značně vyvíjí a na trh přichází velmi rychlá zařízení. Prvním typem, který byl pořízen, je tablet PC Acer Aspire 1825PT. Jedná se o zařízení s mobilním nízkonapětovým dvoujádrovým procesorem Intel a kapacitním dotykovým displejem. Tato skutečnost vylučuje použití běžného stylusu a vyžaduje optimalizaci aplikačního softwaru pro ovládání prsty. Výhodou zařízení je hardwarová klávesnice, která umožňuje pohodlné ovládání aplikace tam, kde by dotykové ovládání nebylo vhodné.

3.1.2 Software

Stávající programové vybavení serveru sestává z aplikací dostupných v repozitářích OS Debian GNU/Linux 5.0 (dále jen Debian, [3]). Zmíněná verze obsahuje průběžně aktualizované balíčky aplikací i HTTP serveru.

Instalované balíčky:

- Samba – sdílení dat přes CIFS protokol,
- CUPS – tiskový subsystém,
- Apache 2 – HTTP server,
- PHP 5 – interpret jazyka PHP,
- MySQL 5 – relační databáze,
- MRTG – monitorování zátěže.

Klientem pro přihlášení do aplikace může být jakýkoli WWW prohlížeč podporující standardy dle W3C. Prioritou se stala optimalizace pro prohlížeč Google Chrome. Ten v době psaní této práce vykazoval objektivně nejvyšší výkon při zpracování JavaScriptu, který během zobrazování obsahu na straně klienta generuje největší zátěž. Žádné další speciální softwarové vybavení není pro provoz aplikace nutné.

3.1.3 Knihovny

Zdaleka ne veškerou funkcionalitu je nutné programovat od nuly. Zdroje otevřeného softwaru, obvykle uvolňovaného pod licencí GNU/GPL, jsou obrovskou zásobárnou hotových knihoven, které se nabízí k zapojení do vyvíjených aplikací. Po zhodnocení požadavků byly vybrány následující:

- Pear MDB2 – abstrakční rozhraní pro komunikaci s databázovým serverem, které umožňuje jednoduchou migraci v případě nutnosti, [4]
- PHPMailer – knihovna pro podporu jednoduchého odesílání emailů, [5]
- jQuery – rozsáhlá knihovna s množstvím doplňkových funkcí pro programování v jazyce JavaScript, [6]

- CKEditor – vizuální editor textu, [7]
- html2ps/html2pdf – konvertor HTML 4.0 do přenositelných formátů PostScript a PDF. [8]

3.2 Diagram datových toků

Diagram datových toků (DFD) je rozdělen do vrstev podle úrovně zanoření v systému. Každou nižší vrstvu získáváme dekomponováním jednoho z procesů, který se vyskytuje v nadřazené úrovni.

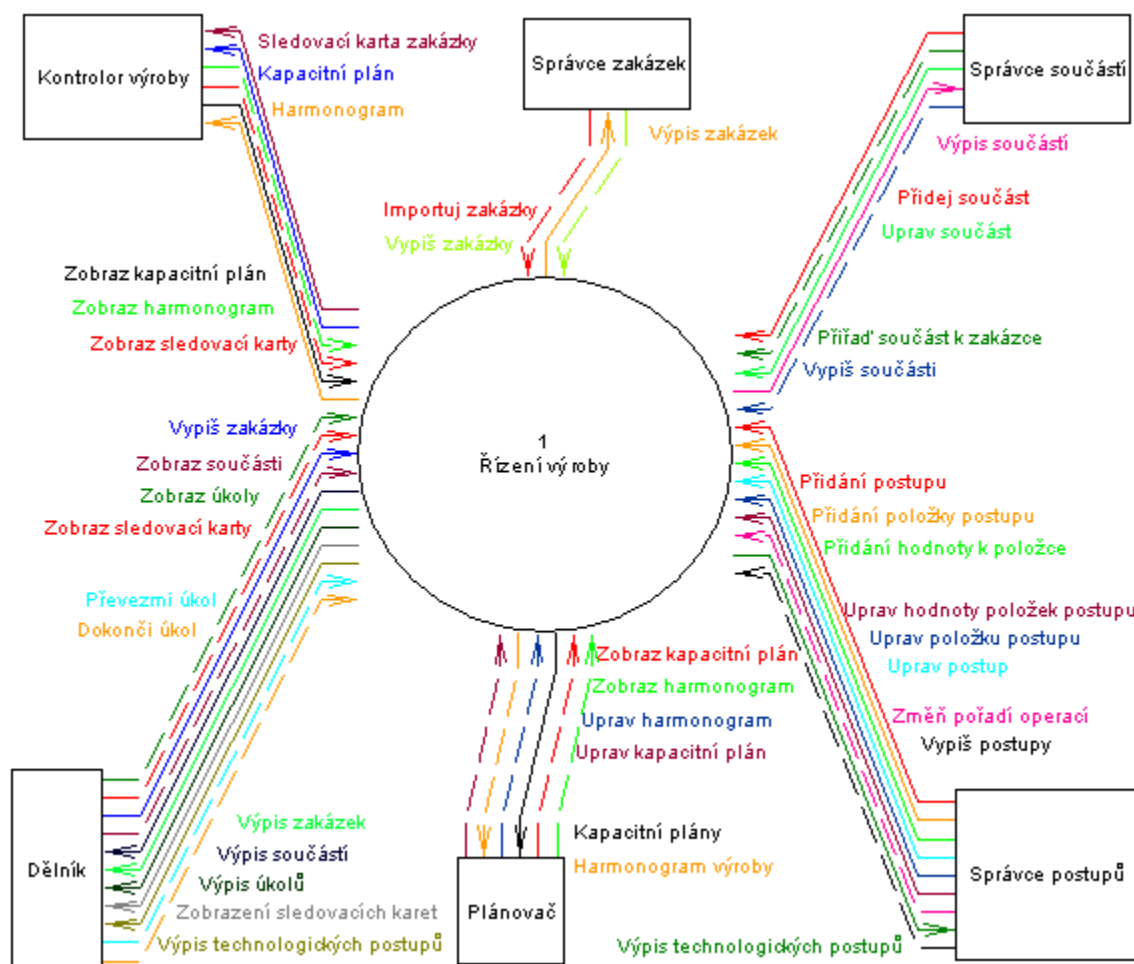
Základní prvky DFD:

- Terminátor – jedná se o externí entity komunikující se systémem z jeho okolí. Při návrhu není možné měnit způsob práce terminátoru, jelikož se jedná o prvek, který není součástí systému.
- Proces – prvek systému provádějící transformaci vstupů na výstupy za určitých podmínek. Proces zpracovává vstupy produkované terminály nebo data získaná z pamětí.
- Tok – způsob znázornění přenosu informací od terminálů k procesům a od procesů k pamětem nebo obráceně. Toky dat, které spojují terminály a procesy, ve skutečnosti znázorňují komunikační rozhraní a v reálném prostředí se může jednat i o fyzické přenosy materiálu nebo dokumentů. V popisovaném systému existují dva typy událostí, které jsou tímto způsobem popisovány. Jedná se o události typu Flow (F), které znázorňují datové toky a události typu Control (C) znázorňující řídicí toky.

Kontextový diagram

Kontextový diagram slouží k jednoduché orientaci v systému a jeho funkcích. Každá událost, kterou generuje některý z terminátorů, musí být obsluhována odpovídajícím modulem a k tomu určenou funkcí. Kontextový diagram také napomáhá programátorovi při specifikaci uživatelských skupin a tvorbě pravomocí, které jsou přesně identifikovatelné na základě událostí.

Hlavní proces, viditelný na první úrovni DFD, zobrazuje systém jako celek a na něj působící terminátory. V diagramu jsou zobrazeny všechny řídicí a datové toky, kterými jednotlivé uživatelské role ovlivňují tento systém.



Obr. 2 – Kontextový diagram

Z počátečního přezkoumání situace ve společnosti vyplynula skutečnost, že v systému se bude vyskytovat celkem šest uživatelských rolí reprezentovaných šesti terminátory. Tyto role jsou ve vyvíjení programu reprezentovány stejným počtem uživatelských skupin. Administrátor systému při vytváření uživatelů rozhodne, kterých skupin budou jednotliví uživatelé členy. Aplikace umožňuje komfortní cestou vytvořit novou pracovní skupinu a přiřadit jí jednotlivé pravomoci v případě, že postupem času dojde k vytvoření nové uživatelské role.

Terminátory:

1. **Kontrolor výroby** – jedná se o skupinu lidí, kteří mohou prohlížet stav jednotlivých zakázek, hlídat kapacitní plány a kontrolovat harmonogram výroby. Uživatelé zajišťující tuto roli nemají žádná další oprávnění. Ve skutečnosti ale budou mít dostupné i všechny ostatní role, takže jejich pravomoci v systému nebudou nijak omezeny.

Seznam událostí:

- Zobraz kapacitní plán C
- Zobraz harmonogram C
- Zobraz sledovací karty C
- Sledovací karta zakázky F
- Harmonogram F
- Kapacitní plán F

2. **Správce zakázek** – tato role má pouze dvě oprávnění, a těmi jsou import zakázek a jejich prohlížení.

Seznam událostí:

- Importuj zakázky C
- Vypiš zakázky C
- Výpis zakázek F

3. **Správce součástí** – uživatelé mající za úkol plnění dat součástí a udržování jejich aktuálnosti. V případě změny na výkrese musí tento pracovník zajistit zkopírování staré součásti, její archivaci a upravení nově vzniklé kopie.

Seznam událostí:

- Vypiš součásti C
- Přidej součást F
- Uprav součást F
- Přiřaď součást k zakázce C
- Výpis součástí F

4. **Správce postupů** – technologické postupy jsou důležitým prvkem v celé aplikaci. Oprávnění pro tuto roli sestávají z přidávání postupů, vkládání položek postupů a jejich ohodnocování předpokládanými časy. V případě, že došlo k úpravě některé ze součástí, je tento uživatel povinen zajistit revizi celého postupu a provést případné změny.

Seznam událostí:

- Přidání postupu F
- Přidání položky postupu F
- Přidání hodnoty k položce F
- Uprav postup F
- Uprav položku postupu F
- Uprav hodnoty položek postupu F
- Změň pořadí operací F
- Vypiš postupy C
- Výpis technologických postupů C

5. **Plánovač** – plánování úzce souvisí s manažerským rozhodováním vztaženým k potvrzování dodacích termínů pro nové zakázky. Úkoly plánovače jsou udržování harmonogramu výroby v konzistentním stavu s ohledem na potvrzené datum dodání a tvorba kapacitních plánů jednotlivých operací tak, aby docházelo k plnění těchto termínů.

Seznam událostí:

- Zobraz kapacitní plán C
- Zobraz harmonogram C
- Uprav harmonogram C
- Uprav kapacitní plán C
- Harmonogram výroby F
- Kapacitní plány F

6. **Dělník** – zaměstnanci pracující na dílenské pozici potřebují dostávat dostatečné informace o stavu aktuálně vyráběných zakázek především z důvodu nutné delší přípravy na některé operace. Ze zmíněného důvodu má tento uživatel dostupné agendy zakázek a součástí včetně jejich technologických postupů. Dělník nemůže do žádné z agend svévolně zasahovat a měnit je. Jeho hlavní povinností zůstává přebírání úkolů a zaznamenávání spotřebované práce.

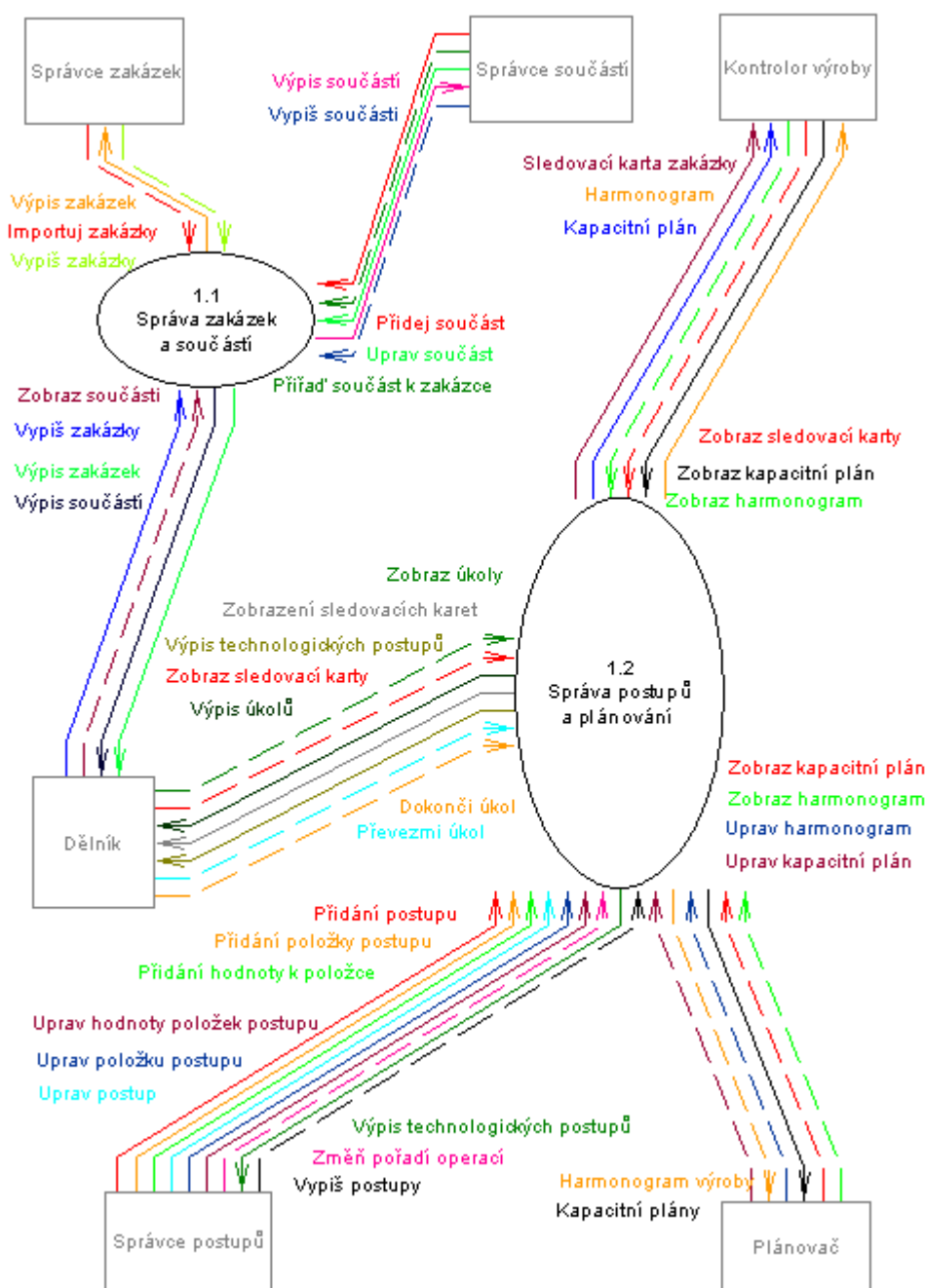
Seznam událostí:

- Vypiš zakázky C
- Zobraz součásti C
- Zobraz úkoly C
- Zobraz sledovací karty C
- Převezmi úkol C
- Dokonči úkol F
- Výpis zakázek F
- Výpis součástí F
- Výpis technologických postupů F
- Výpis úkolů F
- Zobrazení sledovacích karet F

Dekompozice procesu Řízení výroby

Proces *Řízení výroby* je možné rozložit do dvou menších (Obr. 3), a to *Správa zakázek a součástí* a *Správa postupů a plánování*.

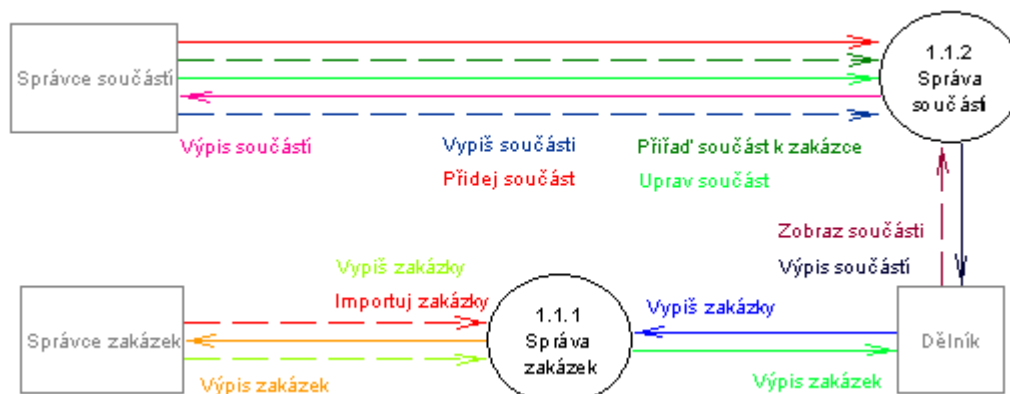
První z jmenovaných obsluhuje veškeré události spojené se správou zakázek a součástí. Proces *Správa postupů a plánování* zpracovává toky generované terminátory *Kontrolor výroby*, *Dělník*, *Plánovač* a *Správce postupů*. Všechny jmenované role produkují události spojené s moduly kapacitního plánování, technologických postupů a sledovacích karet.



Obr. 3 – Dekompozice řízení výroby

Správa zakázek a součástí

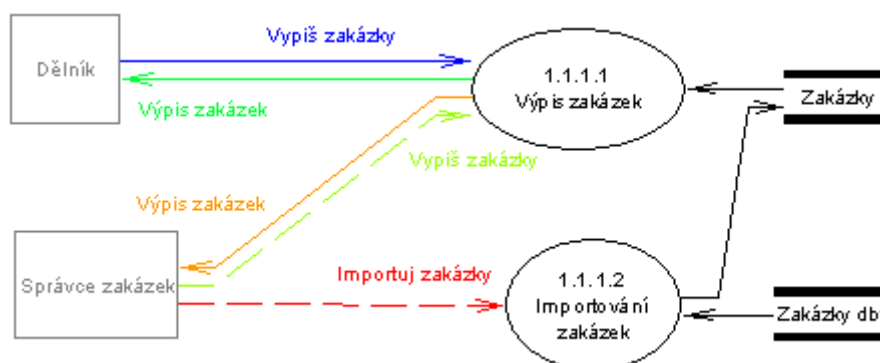
Správa zakázek a součástí je souhrnný proces zaštiťující dva menší moduly systému. Jedná se o nakládání se základními agendami, které jsou v aplikaci použity a tématicky spolu souvisí. Při dekompozici tohoto procesu tedy vznikly dva menší prvky *Správa součástí* a *Správa zakázek*. S prvním jmenovaným komunikují terminátory *Správce součástí* a *Dělník* a proces *Správa zakázek* je ovlivňován terminátory *Správce zakázek* a *Dělník*.



Obr. 4 – Správa zakázek a součástí

Správa zakázek

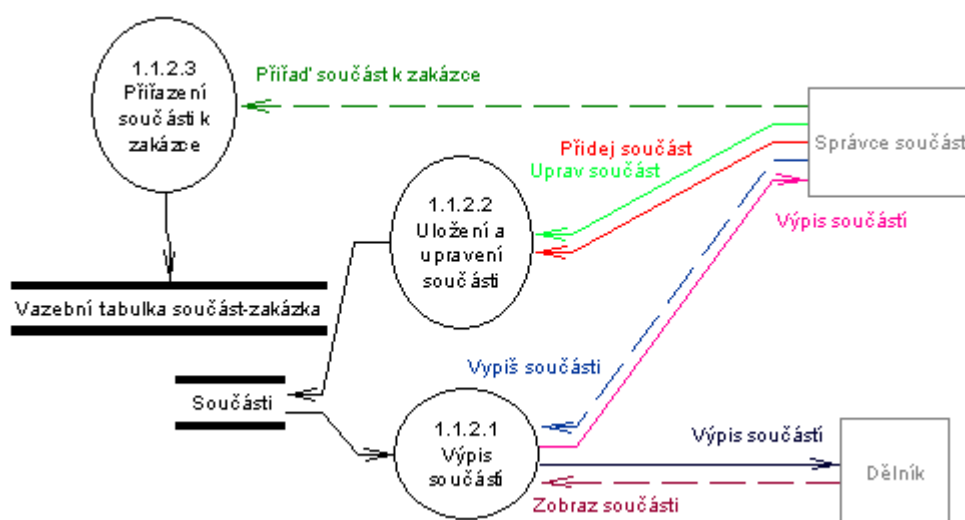
Popisovaný prvek je základní částí systému, která i přes toto výsadní postavení nezastává příliš mnoho funkcí. Základem je proces *Importování zakázek*, který pomocí funkce *exec()* spouští program *dbf2mysql*, zajišťující import souboru ve formátu dbf do MySQL databáze. Jedná se o aplikaci do příkazového řádku spuštěnou se základními parametry, kterými jsou uživatelské jméno k databázi, heslo, jméno databáze a soubor s daty. Proces *Výpis zakázek* potom zajišťuje funkce spojené s vytvořením nečíslovaného seznamu, který představuje základ pro stromovou strukturu.



Obr. 5 – Správa zakázek

Správa součástí

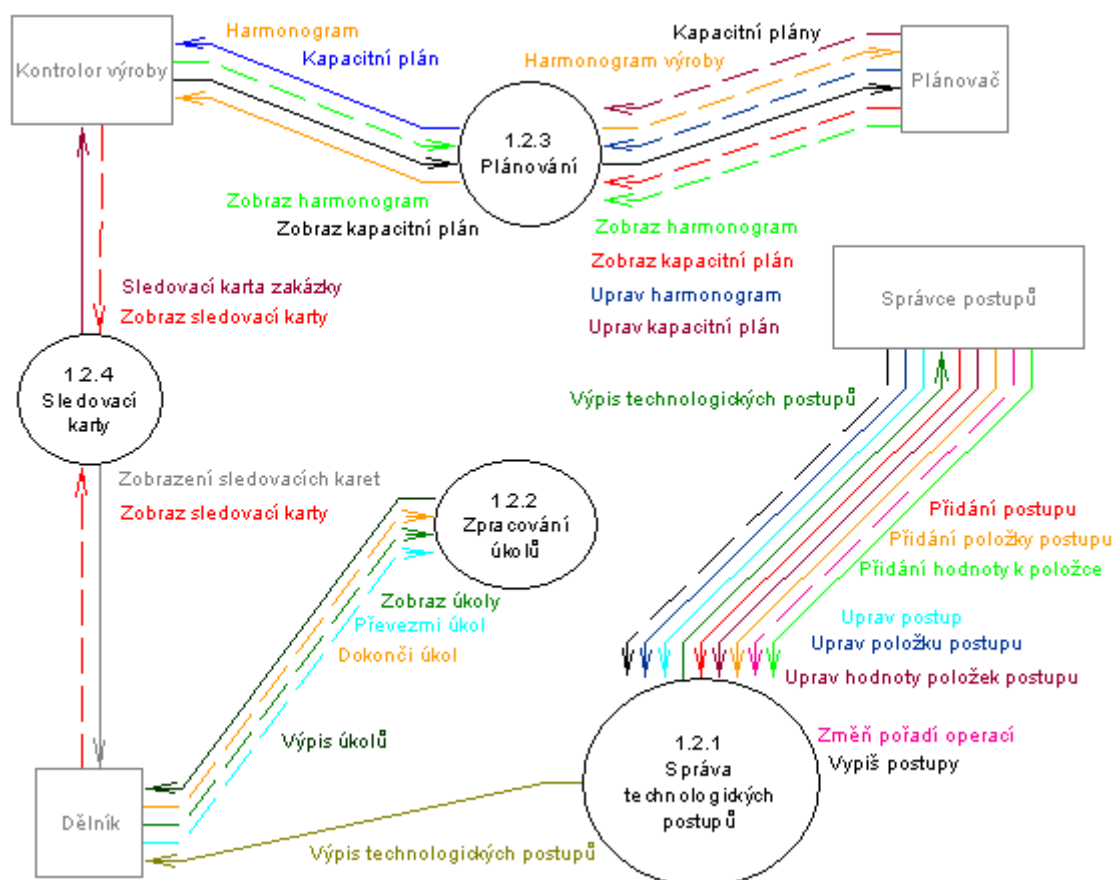
Posledním grafem této větve DFD je *Správa součástí*. Každá realizovaná zakázka může obsahovat libovolné množství položek, které jsou pod její hlavičkou vyráběny. Tvorbu těchto dvojic zajišťuje proces *Přiřazení součástí k zakázce* vkládající na uživatelský pokyn odpovídající informaci do vazební tabulky. Jednotlivé součásti vytváří dvouúrovňovou stromovou strukturu, které v realitě odpovídá vztah sestava-součást. Posledním prvkem na této úrovni DFD je proces obsluhující výpis uložených součástí komunikující, na žádost uživatele, s datovým úložištěm.



Obr. 6 – Správa součástí

Správa postupů a plánování

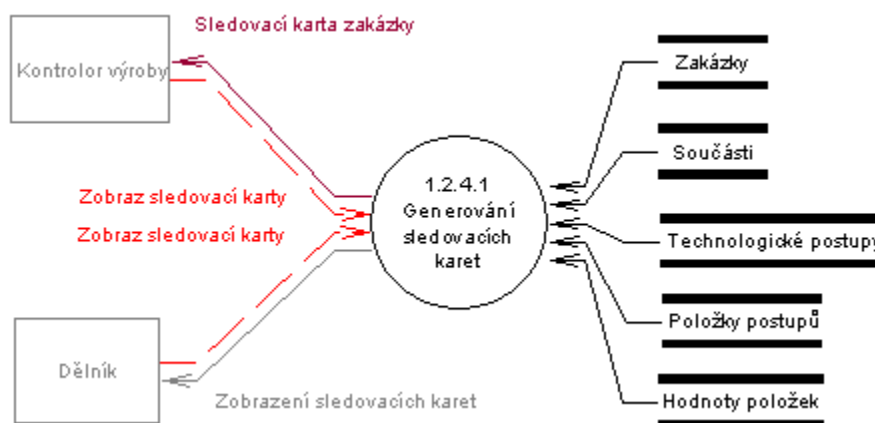
Tento proces je druhou větví diagramu popisující funkce plánování a správy technologických postupů a je dekomponován na čtyři menší podprocesy spravující největší část know-how společnosti. Technologické postupy výroby, navíc se znalostí časových náročností a plánovacím modulem, jsou bezesporu nejvíce ceněnými prvky. Jejich výstupům a funkcím je tedy věnována zvýšená pozornost.



Obr. 7 – Správa postupů a plánování

Sledovací karty

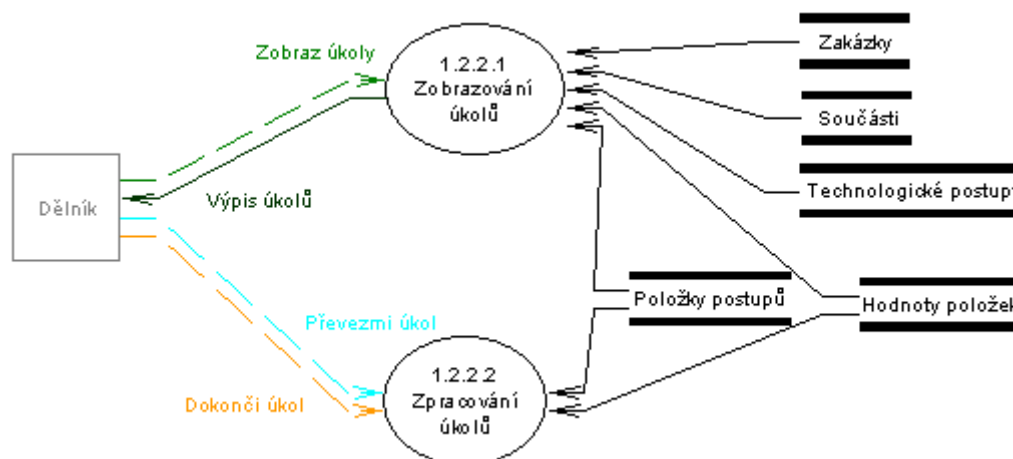
Sledovací karty lze popsat procesem, který zajišťuje generování záznamů s obsahem odpovídajícím technologickému postupu dané součásti. Jelikož je pro generování sledovacích karet potřeba znát detaily téměř ze všech agend, čte proces data z tabulek zakázek, součástí i technologických postupů. Zápis do těchto tabulek není popisovaným procesem realizován.



Obr. 8 – Sledovací karty

Zpracování úkolů

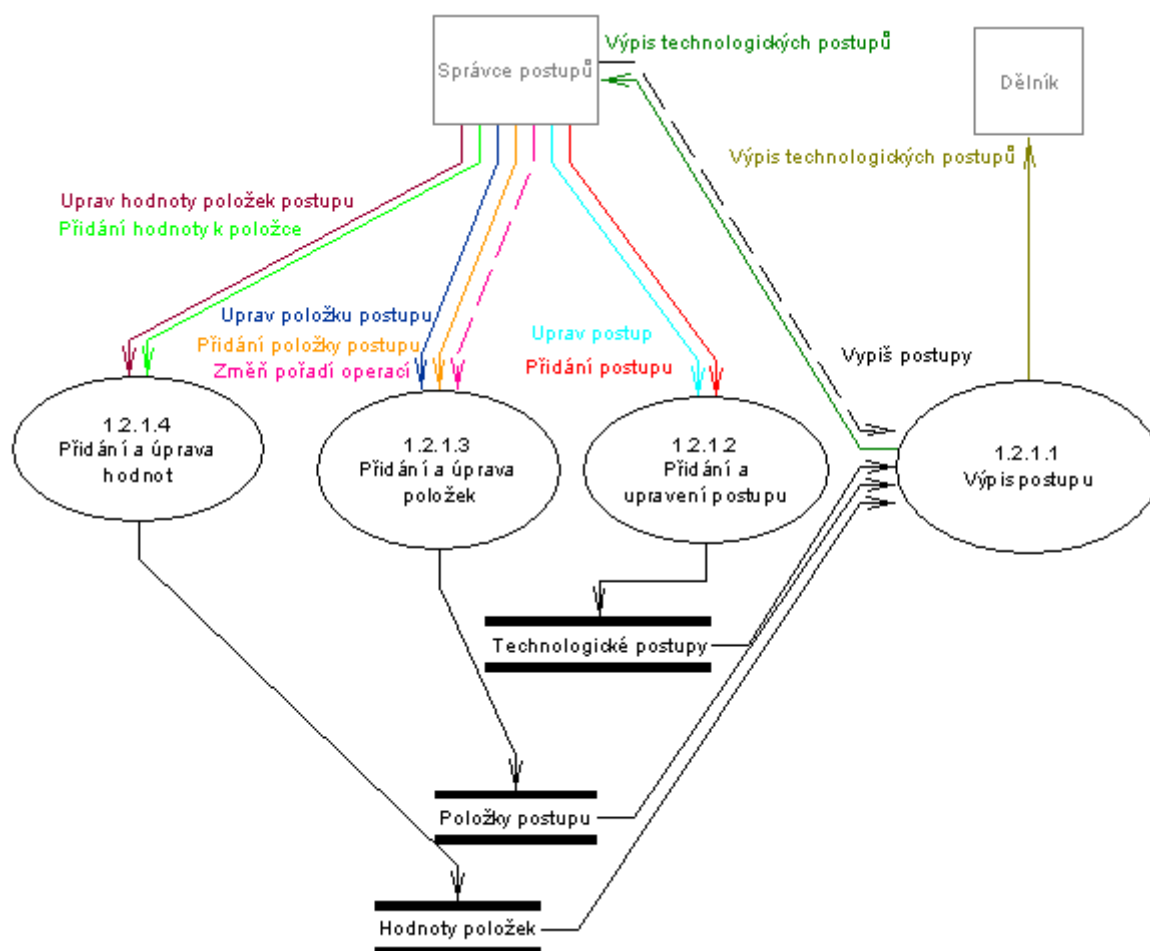
Výpis jednotlivých úkolů, a to přebraných, nepřebraných, dokončených a nebo jejich kombinace, je realizován procesem *Zobrazení úkolů*. Tento proces opět potřebuje přístup k datům ze zakázek, u nichž musí rozlišit, jestli nejsou ukončené, a datům ze součástí a postupů, podle kterých jsou vyráběny.



Obr. 9 – Zpracování úkolů

Správa technologických postupů

Proces *Správa technologických postupů* má přístup ke všem datovým entitám, ve kterých jsou uloženy technologické postupy, jejich položky a hodnoty. V jeho rámci jsou vykonávány veškeré operace s výrobními postupy, včetně přebírání a dokončování úkolů a zároveň sběr údajů o spotřebované práci.



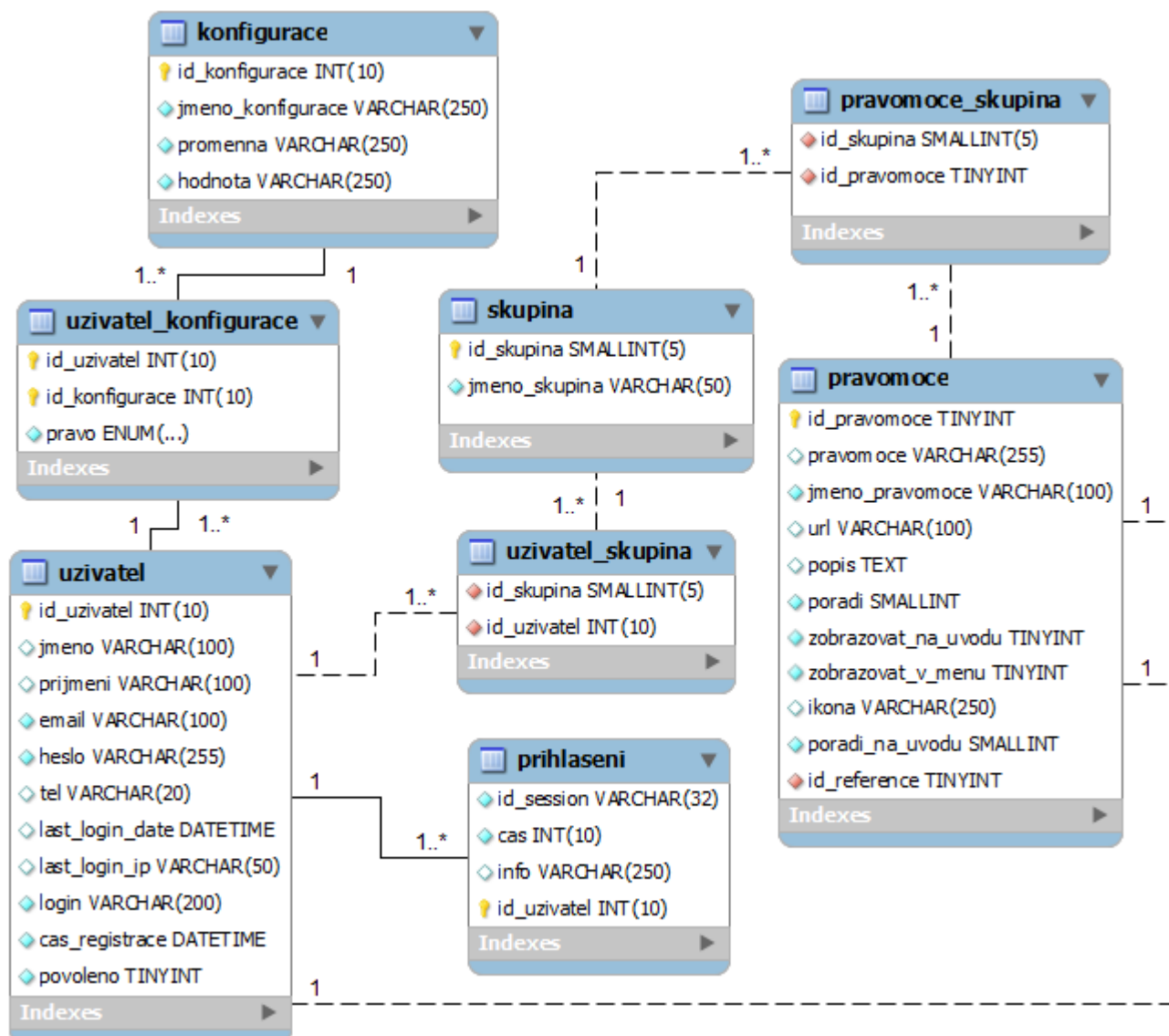
Obr. 10 – Správa technologických procesů

3.3 Entitně relační diagram

Entitně relační diagram (ERD) se v odborných publikacích obvykle označuje jako datový model definující strukturu dat a v čase neměnné parametry. Díky tomu, že nedochází ke změnám datového schématu, zůstávají zachovány i vztahy mezi jednotlivými atributy při transformaci dat v systému. [9] Kompletní návrh databázové struktury popisované aplikace je k dispozici v elektronické podobě.

Prvky ERD:

- **Entita** – je prvek reálného světa a je definována jako jedinečná a nenahraditelná. Její charakteristika je popsána pomocí parametrů, které se v terminologii nazývají atributy. Vzhledem k použití relační databáze jsou v projektu řízení výroby entity reprezentovány tabulkami a jednotlivé atributy reprezentují sloupce tabulek.
- **Vztah** – jedná se o relaci mezi dvěma nebo více entitami, která slouží k zaznamenání kontextu. Vztahy jsou dále definovány kardinalitou, která určuje mohutnost množiny vztahu a nabývá hodnot $m:n$, $1:n$, $n:1$ a nebo $1:1$. Relace typu $m:n$ je nahrazována vazební tabulkou a dvěma vztahy $1:n$.



Obr. 11 – Entitně relační diagram, autentizace, autorizace

3.3.1 Tabulka pro evidenci zakázek

Vzhledem k požadavku napojení aplikace na evidenci zakázek v programu Company Manager je tabulka *vyroba_zakazka* realizována jako pohled do entity importované ze zmíněné aplikace. Původní entita obsahuje celkem 112 atributů a není navrhována dle vhodných metodik. Vzhledem k tomuto faktu je vytvoření pohledu nezbytné pro lepší orientaci a skrytí matoucích a nepotřebných údajů.

vyroba_zakazka:

- id_zakazka* – identifikátor zakázky, který je primárním klíčem tabulky. Jako cizí klíč figuruje ve vazební tabulce *vyroba_zakazka_soucast_postup*,
- nazev* – stručný popis předmětu zakázky,
- pracovnik* – zaměstnanec, který je odpovědný za plnění dané zakázky,
- termin_dod* – termín dodání hotové zakázky. Tento parametr je využit při plánování a stanovuje se podle něj, zda je možné zakázku dokončit v požadovaném termínu,
- stredisko* – označení úseku, kde má být zakázka realizována. V tomto případě bude středisko vždy nabývat hodnoty 3,
- stav* – atribut *stav* je datového typu *enum* a jeho datový slovník nabývá dvou hodnot:

0 = neukončená zakázka, 3 = ukončená zakázka.

Ostatní atributy přesněji popisují konkrétní zakázku a není třeba je zde rozebírat.

3.3.2 Tabulky součástí

V tabulce *vyroba_soucast* jsou evidovány všechny součásti, které společnost SLB vyrábí nebo nakupuje a používá ve svých produktech. Každé hotové ložisko se skládá z daného počtu dílů, což je vztah, který je kvůli dědičnosti zachycen ve vazební entitě *vyroba_soucast_soucast*. K vyráběným součástem je nutné v tabulce *vyroba_soucast_zbytek* uchovávat informaci o zbytcích z předchozí produkce pro jejich následné zařazení do opakované výroby (Obr. 12).

vyroba_soucast:

- *id_soucast* – primární klíč tabulky a jedinečný identifikátor. Jako cizí klíč figuruje ve vazební entitě *vyroba_zakazka_soucast_postup*,
- *nazev* – textové označení součásti,
- *cislo_vykresu* – číslo výkresu identifikuje, podle jaké dokumentace byla součást vyráběna,
- *cislo_zmeny* – zpřesňuje identifikaci dokumentace o číslo revize,
- *cena* – v případě nakupovaných a upravovaných dílů je třeba uchovávat i jejich cenu pro stanovení celkových nákladů na součást,
- *typ* – rozlišení nakupovaných, vyráběných a upravovaných součástí,
- *dodavatel* – textová informace o dodavateli nakupovaných dílů,
- *aktivni* – v případě vydání nové verze výkresu je nutné starou součást archivovat a vytvořit její novější revizi. Stav konkrétní součásti je tedy popisován tímto atributem.

Ostatní atributy udávají časy vzniku a další doplňkové informace.

vyroba_soucast_soucast:

- *id_soucast* – identifikátor hierarchicky nadřazené součásti,
- *id_soucast_podrazena* – identifikátor podřazené součásti,
- *ks_na_komplet* – atribut udávající počet kusů potřebný pro kompletaci celku. Tato položka je plně závislá na primárním klíči popisované tabulky, což je dvojice *id_soucast, id_soucast_podrazena*,
- *poradi* – umístění součásti na kusovníku. Umožňuje rychlou orientaci při prohlížení sestavy a dat v aplikaci.

vyroba_soucast_zbytek:

- *id_soucast* – cizí klíč do tabulky součástí,
- *mnozstvi* – počet zbývajících výrobků,
- *umisteni* – fyzická poloha zbytků ve skladu,
- *stav* – do této položky je možné uvádět informace o stavu dokončení výrobku,
- *poznamka* – doplňková informace,
- *id_zakazka* – chceme-li zbytek využít pro novou výrobu, je třeba ho přiřadit k zakázce. V opačném případě by mohlo dojít k situaci, kdy bude uživatel počítat s jedním zbytkem pro více zakázek,
- *datum_vlozeni, datum_prirazeni* – časové informace.

3.3.3 Tabulky technologických postupů

Technologické postupy jsou základním know-how společnosti, a proto je důležitá jejich řádná archivace a průběžná optimalizace. Jednotlivé pracovní postupy mají jednu až n položek a jednotlivé položky mohou nabývat hodnot zadávaných buďto dělníkem (při ukončování přebraného úkolu), nebo vedoucím výroby při tvorbě nového technologického postupu (Obr. 13).

Při zadávání nové výroby mohou nastat tři situace:

- Zadává se opakovaná výroba, což je nejjednodušší případ. Součásti zůstávají neměnné a technologický postup z minulé výroby se kopíruje 1:1.
- Zadává se opakovaná výroba se změnou součásti nebo postupu. Změněná součást je zkopírována a je upraven atribut *zmena* v tabulce *vyroba_soucast*. Jakékoliv změny v technologickém postupu neovlivní předchozí záznamy díky jejich kopírování.
- Zadává se nová výroba, kdy je ke každé součásti vytvořen nový technologický postup.

vyroba_tecnologicky_postup:

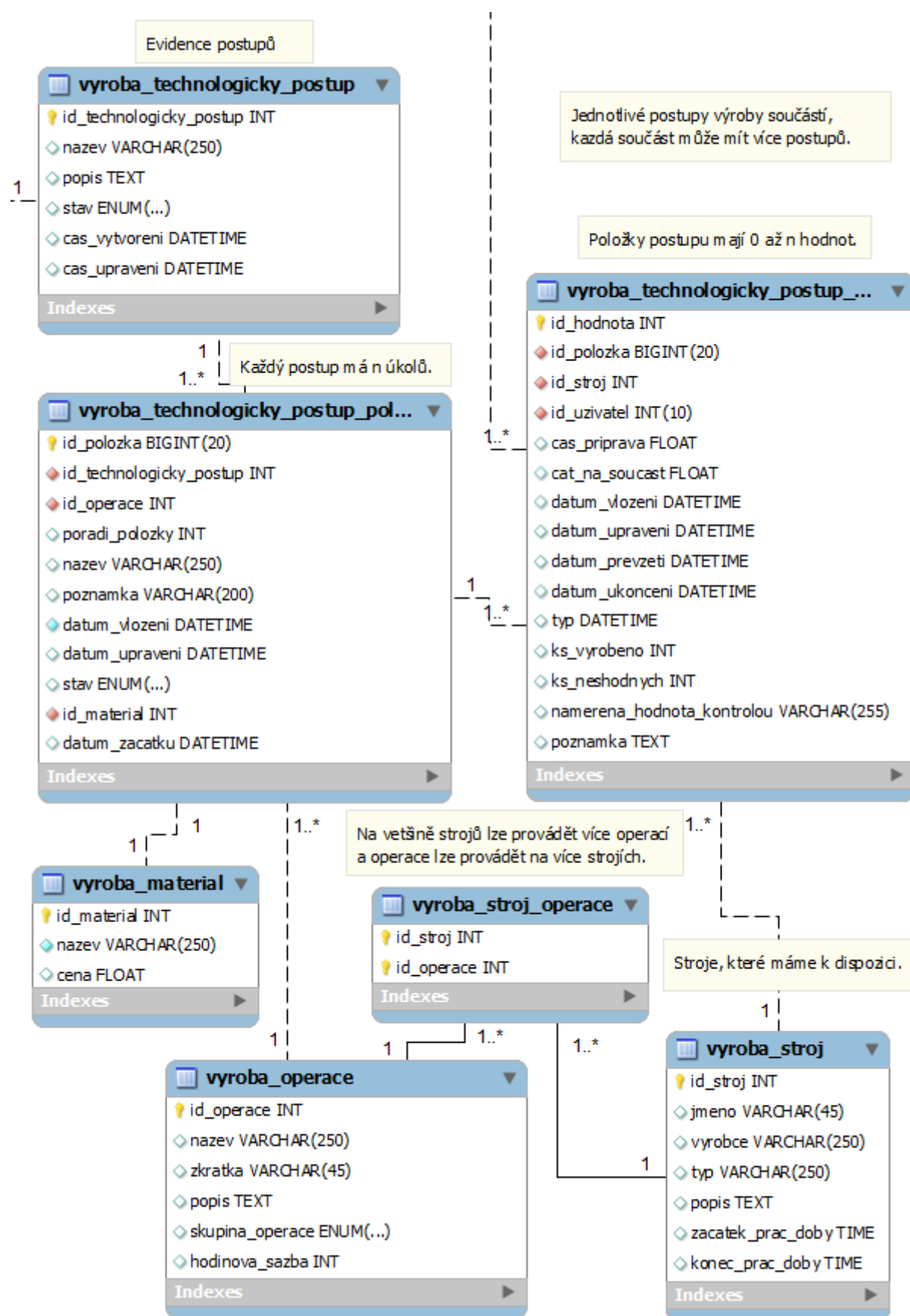
- *id_tecnologicky_postup* – identifikátor technologických postupů a primární klíč tabulky,
- *nazev* – krátké označení, čím se technologický postup zabývá,
- *popis* – sloupec, do kterého je možné vpisovat důležité komentáře týkající se celého výrobního postupu,
- *stav* – atribut *stav* může nabývat hodnoty „ověřený“ pro již odzkoušený postup a hodnoty „nový“ pro nově vytvořený postup, například při přidávání nové součásti,
- *cas_vytvoreni*, *cas_upraveni* – časové hodnoty.

vyroba_tecnologicky_postup_polozka:

- *id_polozka* – primární klíč tabulky identifikující jednotlivé položky,
- *id_tecnologicky_postup* – cizí klíč do tabulky technologických postupů. Každá položka patří do jednoho pracovního postupu,
- *id_operace* – cizí klíč do tabulky operací,
- *nazev* – textové pojmenování položky postupu,
- *stav* – atribut *stav* nabývá tří hodnot (čeká, převzatá, dokončená). Stav položky postupu mohou měnit všichni uživatelé přebíráním a ukončováním jednotlivých úkolů,
- *id_material* – ukazatel do tabulky materiálů.

vyroba_tecnologicky_postup_polozka_hodnota:

- *id_hodnota* – primární klíč tabulky identifikující hodnoty,
- *id_polozka* – každá hodnota patří právě jedné položce postupu,
- *id_stroj* – položka postupu (úkol) může být realizována na více strojích, proto je identifikátor konkrétního zařízení uložen u hodnoty a ne přímo u položky,
- *id_uzivatel* – identifikátor uživatele, který danou hodnotu k položce vytvořil,
- *cas_pripava* – informace o délce trvání přípravy práce,
- *cas_na_soucast* – čas na operaci na jeden kus výrobku,
- *typ* – hodnota může být čtyř různých typů (předpokládaná, dříve dosažená, přebraná a dokončená),
- *ks_vyrobeno* – atribut popisující množství zpracovaných kusů ve shodě s výkresovou dokumentací,
- *ks_neshodnych* – ve výrobě se bohužel vyskytují i výrobky neshodné a tuto informaci o „zmetkovitosti“ je třeba uchovávat pro rozhodování o velikosti zmetkového zajištění při opakované výrobě.



Obr. 13 – ERD, technologické postupy

3.3.4 Vazební tabulka mezi zakázkou, součástí a technologickým postupem

V aplikaci je nutné udržovat informaci, ve které zakázce je daná součást vyráběna. Tento základní údaj musí být také rozšířen o doplňkové parametry, kterými jsou informace o počtu vyráběných kusů, použitém technologickém postupu, termínu dodání zákazníkovi atd.

vyroba_zakazka_soucast_postup:

- *id_zakazka* – cizí klíč do tabulky *vyroba_zakazka*. Společně s atributem *id_soucast* tvoří primární klíč této vazební entity,
- *id_soucast* – cizí klíč do tabulky *vyroba_soucast*,
- *id_tecnologicky_postup* – je parametrem určujícím postup, podle kterého bude daná součást pro konkrétní zakázku vyráběna,
- *pocet_kusu* – atribut určující množství vyráběné součásti,
- *termin_dodani* – dodání hotového výrobku dodavateli,
- *stav* – pro nakupované součásti je třeba v závislosti na zakázce udržovat informaci o jejich aktuálním stavu,
- *potvrzene_doruceni* – termín, kdy má být nakupovaná součást doručena,
- *vykres* – parametr požadovaný jedním z vedoucích výroby. Jedná se o dvoustavovou hodnotu určující, zda ve výrobním oddělení došlo k přebrání výkresu od konstrukce.

3.3.5 Pomocné tabulky

K technologickým postupům je v systému potřeba udržovat další agendy, přičemž jejich naplnění daty bylo vesměs realizováno jednou a dokud nebudou společností zakoupeny nové technologie, nebude jejich změna nutná. Mezi tyto tabulky patří *vyroba_stroj*, *vyroba_operace* a vazební entita *vyroba_stroj_operace*.

Další pomocnou tabulkou je *vyroba_material*, jejímž naplněním při nasazování systému byla pokryta valná většina ve společnosti využívaných materiálů. I do této agendy ale mohou správci součástí v případě potřeby jednoduše doplňovat nové položky.

V návrhu ERD jsou i tabulky řešící evidenci pomůcek a protokolů, tyto však nejsou v aktuální verzi systému využity.

3.4 Rozdělení funkčních bloků

Celý systém řízení výroby lze rozdělit do devíti funkčních bloků. Rozdělení je realizováno na základě diagramu datových toků, který podává komplexní pohled na celou aplikaci. Jednotlivé funkční bloky jsou od sebe odděleny a zapouzdřeny v modulech.

Modul zakázka

Modul *zakázka* slouží k evidenci zakázek a jejich grafickému zobrazení.

Vstupy:

- data zakázek z dbf importu.

Výstupy:

- seznam zakázek,
- podrobné informace o zakázce.

Zajišťované funkce:

- výpis zakázek optimalizovaný pro dotykové ovládání,
- naplnění objektu daty o požadované zakázce.

Využívané datové struktury:

- tabulka *zakazky*,
- pohled *vyroba_zakazka*.

Modul součást

Jak název modulu napovídá, slouží ke zpracování všech akcí spojených s vyráběnými a nakupovanými součástmi. Hlavním úkolem této třídy je akvizice dat výrobních prvků a jejich zpracování do dvouúrovňové stromové struktury. Modul bude ve velké míře ovládán pomocí dotykových panelů, a proto je nutná jeho optimalizace pro tento typ vstupního zařízení.

Vstupy:

- formulář s informacemi o součásti,
- informace o přiřazení součásti k zakázce,
- požadavek ke kopírování, smazání nebo archivaci součásti,
- zbytky vyráběných součástí.

Výstupy:

- seznam součástí,
- detailní informace o konkrétní součásti,
- seznam zbytků součástí.

Zajišťované funkce:

- přidávání a upravování součástí,
- přiřazování součásti k zakázce,
- výpis stromové struktury,
- kopírování, mazání a archivace součástí,
- evidence zbylých položek při dokončování výroby.

Využívané datové struktury:

- tabulky *vyroba_soucast*, *vyroba_soucast_zbytek*,
- vazební entity *vyroba_zakazka_soucast_postup*, *vyroba_soucast_soucast*.

Modul technologický postup

Technologické postupy slouží k uchování informací o způsobu výroby a jako návody při výrobě. Z managerského pohledu se jedná o zdroj informací pro kapacitní plánování, sledování stavu zakázky a o podklady pro výpočet nákladů na výrobu součástí.

Vstupy:

- založení nového nebo úprava stávajícího technologického postupu,
- přidávání nebo úprava položek postupu,
- informace o časové náročnosti dané položky postupu,
- data převzetí a ukončení položky.

Výstupy:

- technologické postupy s položkami a nasbíranými hodnotami.

Využívané datové struktury:

- tabulky *vyroba_tecnologicky_postup*, *vyroba_tecnologicky_postup_polozka*, *vyroba_tecnologicky_postup_polozka_hodnota*, *vyroba_stroj*, *vyroba_operace*, *vyroba_stroj_operace*.

Modul úkol

Důležitým prvkem v celém systému je seznam všech úkolů nebo úkolů převzatých jednotlivými uživateli. Díky tomuto výpisu může vedení společnosti sledovat, kdo právě pracuje na jakém úkolu a to včetně údaje, jak dlouho je již daný úkol vykonáván.

Výstupy:

- seznam nedokončených úkolů,
- seznam úkolů převzatých všemi a nebo pouze konkrétním uživatelem.

Využívané datové struktury:

- agenda pracuje se všemi tabulkami používanými v modulech *zakázka*, *součást* a *technologický postup*,
- ke svému běhu potřebuje i vybrané metody zmíněných modulů, jako jsou například funkce k načtení dat.

Modul sledovací karty

Sledování stavu zakázky je momentálně ve společnosti řešeno formou sledovacích tabulek. Jejich vypovídací hodnota je však poměrně malá, jelikož neobsahují položky technologických postupů, ale pouze seznam předtištěných operací, které jsou obvykle ve výrobě realizovány. Úkolem tohoto modulu tedy bude výpis všech součástí pro danou zakázku včetně položek z jejich technologického postupu. Tím získá vedení společnosti podrobný přehled o již realizovaných operacích a o tom, co na zakázce ještě zbývá dokončit.

Výstupy:

- karty pro sledování výroby.

Zajišťované funkce:

- prostřednictvím modulů *součást* a *technologický postup* zajišťuje výpis detailů o součástech a položkách postupu, stejně jako jejich přebírání a dokončování.

Využívané datové struktury:

- agenda pracuje se všemi tabulkami používanými v modulech *zakázka*, *součást* a *technologický postup*,
- ke svému běhu potřebuje i vybrané metody zmíněných modulů, jako jsou například funkce k načtení dat.

Modul kapacitní plánování

Kapacitní plánování je nesporně jednou z nejdůležitějších částí řízení ve výrobních společnostech. Při návrhu způsobu plánování byla ze strany zadavatele zavržena jeho automatizovaná varianta. Důvodem k tomuto rozhodnutí byla nutnost optimalizace po sobě jdoucích úkolů, kdy z důvodu přípravných časů je lepší, když na sebe navazují obdobné operace a to i za cenu posunutí více upřednostňované zakázky. Výroba ve společnosti SLB je dělena do sedmi základních skupin, pro které je nutné řešit časové harmonogramy.

Skupiny operací:

- řezání,
- soustružení,
- vrtání,
- broušení na plocho,
- broušení na kulato,
- broušení otvorů,
- montáž.

Pro plánování těchto operací bude funkcionalitu zajišťovat modul *plánování*, který vedení společnosti ulehčí rozhodování při potvrzování dodacích termínů u nových zakázek a tvorbě termínových nabídek. Každá zpožděná operace bude navíc barevně odlišena a na základě této informace může vedení firmy upozornit zákazníka na možné nedodržení termínu.

Výstupy:

- časový plán pro každou skupinu operací s možností nastavení měřítka,
- informace o případném prodloužení nebo pozdním započetí operace.

Modul uživatel a autentizace

Pro rozlišení uživatelů je nutné řešit správu uživatelských účtů. Činnosti s tímto spojené má na starost modul *uživatel*, který využívá tři tříd s názvy *Uzivatel*, *Uzivatel_skupina* a třídu *Auth*. Druhá zmíněná zajišťuje navazování uživatelů na pracovní skupiny a třetí se stará o kontrolu přihlašovacích údajů. Při vstupu do aplikace jsou ověřena uživatelská data a proběhne načtení a uložení pracovních skupin do uživatelevo sezení. Při každém obnovení stránky je kontrolována časová expirace a pravost sezení.

Vstupy:

- data nově přidávaných nebo upravovaných uživatelů,
- přihlašovací údaje při probíhající autentizaci,
- seznam skupin, do kterých patří zpracovávaný uživatel.

Výstupy:

- seznamy uživatelů,
- informace o konkrétním uživateli,
- seznam skupin přidělených uživateli,
- výsledky autentizace.

Zajišťované funkce:

- autentizace uživatelů,
- kontrola expirace přihlášení,
- odhlášení uživatele,
- přidávání, úprava, odstraňování a zakazování uživatelů,
- přiřazování a seznam přiřazených skupin.

Využívané datové struktury:

- tabulky *uzivatel* a *uzivatel_skupina*.

Modul pracovních skupin

Pracovní skupiny slouží k rozdělení uživatelských oprávnění na základě rolí, které jsou uživateli v systému zastávány. Člověk může být členem libovolného množství pracovních skupin, přičemž každá skupina má pouze vybrané množství pravomocí podle odpovídající role.

Vstupy:

- data přidávané nebo upravované skupiny.

Výstupy:

- seznam skupin.

Zajišťované funkce:

- přidávání a upravování skupin,
- mazání skupin,
- seznam vytvořených skupin.

Využívané datové struktury:

- tabulka *skupina*.

Modul pravomocí a autorizace

Jak bylo popsáno dříve u modulu pracovních skupin, je nutné rozeznávat uživatele na základě rolí. Jednotlivé role jsou založeny na pravomocích a dvou třídách, které slouží k jejich zpracování – *Pravomoc* a *Pravomoc_skupina*. Oprávnění jsou poté ověřována pomocí třídy *Autorizace*, která provádí kontrolu přístupových práv u akcí toto vyžadujících. Třída *Pravomoc* slouží také k výpisu administračního menu pro správce aplikace, které se skládá právě ze zmiňovaných pravomocí uspořádaných v dvouúrovňové struktuře.

Vstupy:

- data při přidávání a upravování pravomocí,
- navázání pravomocí k uživatelské skupině,
- ověřovaná pravomoc.

Výstupy:

- seznam pravomocí,
- seznam pravomocí pro konkrétní skupinu,
- administrační menu,
- výsledek autorizace.

Zajišťované funkce:

- přidávání a upravování pravomocí,
- přiřazování pravomocí ke skupinám,
- odstraňování pravomocí,
- autorizace uživatele k provedení akce,
- výpis administračního menu.

Využívané datové struktury:

- tabulky *skupina*, *pravomoce*, *pravomoce_skupina*.

3.5 Zhodnocení

Systém musí být vhodně modulárně rozdělen, aby byl jednoduše udržovatelný i po jeho nasazení. Při programování musí být kladen důraz na oddělení funkcionality na úrovni objektů a na maximální omezení duplikace kódu.

V systému budou existovat celkem 4 uživatelské role s různými stupni oprávnění. Jednotlivé pravomoci je nutné volit s rozvahou, aby nedošlo k omezení uživatelů tam, kde to není vhodné. Zároveň nesmí být pracovníkům povoleno zasahovat do věcí, za které nenesou odpovědnost.

Potenciálním problémem může být zavádění na pracovišti, jelikož programy obdobného typu jsou zaměstnanci často odmítány a zavrhovány. Při nasazování je nutné aplikaci důkladně otestovat, odstranit nalezené chyby a všechny uživatele řádně zaškolit. Ke každému terminálu bude přiložen stručný návod k použití a osoby odpovědné za zavádění systému musí být trvale k dispozici.

4 IMPLEMENTACE

K programování aplikace byly zvoleny webové technologie, zejména pro jejich multiplatformní použitelnost, a to jak u serverové, tak klientské části. Jako základ slouží skriptovací jazyk PHP, který od verze 4 podporuje objektové programování. Pro uchovávání dat byl vybrán MySQL server dostupný v komunitní edici zdarma.

Klientská část aplikace je postavena na standardech XHTML 1.1, CSS 2 a vybrané prvky na připravovaném CSS 3. Ve velké míře je využíváno asynchronních dotazů na server pomocí JavaScriptu a XML, což je technologie označovaná zkratkou AJAX (Asynchronous JavaScript and XML) [10]. Ve vyvíjeném systému je kladen velký důraz na rychlou odezvu a uživatelsky přívětivé rozhraní bez opakovaného obnovování celých stránek. Z tohoto důvodu bylo nasazení asynchronních technologií jedním ze základních požadavků.

Na klientských terminálech je důležitým prvkem optimalizace pro dotykové ovládání. Veškerou práci v tomto směru obstará operační systém, který vstupy z dotykové vrstvy zpracovává obdobným způsobem jako jiná vstupní zařízení a události takto vygenerované předává zobrazovacímu serveru. Drobnou odlišností, oproti klasickému programování pro web, je v tomto případě nutnost vyvarování se událostí OnMouseOver a OnMouseOut, které jsou obvykle používány pro vyvolání informačních oken po najetí myši nad konkrétní objekt. Tato funkcionality je pro dotykové ovládání nepoužitelná. Důležitým prvkem aplikace se stala vhodně zvolená velikost ovládacích prvků a jejich grafická názornost.

Základem pro celou aplikaci se stal elastický layout, který využívá jednotek *em* (*em* je definována jako šířka velkého písmena m). Z této definice vyplývá, že změnou velikosti fontu dojde k přepočítání všech rozměrů, které jsou touto jednotkou označeny. Po počátečním testu se toto řešení osvědčilo a veškeré rozměry jsou v kaskádových stylech uváděny v těchto jednotkách. Zmíněný přístup přináší jednu zásadní výhodu, a to možnost přizpůsobení rozvržení stránky velikosti a rozlišení monitoru, na kterém má být aplikace zobrazena. Toto přizpůsobení je realizováno prvkem umístěným v záhlaví systému, který je dostupný v libovolném okamžiku. Velikost je vázána na konkrétního uživatele a rozlišení monitoru tak, aby nebylo nutné nastavení opakovaně měnit.

4.1 PHP knihovny

PHP je vývojáři webových aplikací často využívaný skriptovací jazyk a není tedy překvapením, že pro něj existuje velké množství rozšíření.

Databázová abstrakční vrstva (Pear MDB2)

U vyvíjené aplikace je předpokládán dlouhý životní cyklus a z toho důvodu by mohla být postupem času provozována na novějším a nebo úplně jiném databázovém serveru. Proto je nasazena abstrakční vrstva nad relačním databázovým serverem umožňující jednoduchou migraci v případě potřeby.

PEAR je repositář knihoven, funkcí a aplikací rozšiřujících jazyk PHP. Jeho součástí je i popisovaná databázová abstrakční vrstva MDB2, která představuje objektově orientované API (Application Programming Interface) pro komunikaci s databázovými servery.

Vlastnosti MDB2:

- přenositelnost vyvíjené aplikace.
Mezi podporované databázové servery patří: MySQL, MySQLi, PostgreSQL, Oracle, Querysim, Interbase/Firebird, MSSQL, SQLite,
- různé metody pro načítání řádků,
- podpora transakcí,
- rozhraní pro získávání informací o tabulkách,
- objektově orientované API.

Dionýsos

Dionýsos je redakční systém, na jehož vývoji pracuji od roku 2002. Jedná se o uživatelsky přívětivý publikační nástroj umožňující intuitivní cestou spravovat webové prezentace, dokumentové servery, soukromé galerie a mnoho dalších typů projektů. Jádru aplikace se skládá z pěti modulů, které zajišťují funkcionalitu potřebnou ve většině informačních systémů.

Zajišťované funkce:

- autentizace uživatelů oproti databázi,
- ochrana sezení proti krádeži,
- skupiny uživatelů,
- systém pravomocí přidělovaných pracovním skupinám,
- zálohování dat systému,
- publikační moduly.

PHPMailer

Aplikace implementující SMTP protokol a umožňující odesílání zpráv pomocí libovolně zvoleného poštovního serveru.

Kromě jiného PHPMailer zajišťuje následující funkcionalitu:

- SMTP autentizace,
- modifikaci hlaviček,
- různé způsoby kódování,
- podporu e-mailů s více adresáty,
- HTML e-maily i čistý text (alternativní obsah pro emailové klienty nepodporující zprávy v HTML formátu),
- vkládání obrázků do těla e-mailu,
- přílohy.

html2ps / html2pdf

Předpokladem pro budoucí rozvoj systému je možnost pohodlného tisku a výměny dokumentů vystupujících z aplikace. V první fázi se bude jednat o sdílení technologických postupů se společnostmi zajišťujícími kooperaci, následně půjde o již zmíněné nabídky, poptávky a další dokumenty. Aplikace *html2ps* je kompletní program zajišťující konverzi HTML 4.0 a CSS 2.0 stránek do přenositelného formátu PostScript nebo PDF společnosti Adobe. Pro výměnu dokumentů se jedná o ideální formáty, kdy příjemce vidí přesně to, co mu odesílatel chtěl předat. Vzhledem k rozšířenosti formátu PDF byla zvolena tato varianta.

4.2 Knihovny a aplikace v jazyce JavaScript

JavaScript je objektově orientovaný jazyk používaný ve webových aplikacích a prezentacích. Syntaxí je obdobný programovacím jazykům C/C++, byl vyvinut pod křídly společnosti Netscape a veřejně ohlášen se společností Sun Microsystems v prosinci 1995. [11] Ke konci dvacátého století byl standardizován asociacemi ECMA [12] a ISO [13]. Program v JavaScriptu je obvykle spouštěn na straně klienta a v případě vyvíjené aplikace je použit výhradně tímto způsobem. Moderní webové prohlížeče dokáží některé části jazyka překládat do strojového kódu (obvykle se jedná o cykly), a tím výrazně zrychlit jeho interpretaci.

V posledních letech se ve velké míře využívá asynchronní komunikace klienta se serverem. Tento způsob výměny informací je realizován pomocí *xmlHttp* požadavků a následného vkládání odpovědi obvykle pomocí *innerHTML*. Popsaný programovací přístup je označován již zmíněnou zkratkou AJAX. V řízení výroby jsou tímto způsobem realizovány následující požadavky:

- filtry v zakázkách a součástech,
- našeptávače k filtrům,
- obnovování agend po přidání položky,
- načítání formulářů pro přidávání, úpravy a zobrazení detailů položek,

- přebírání a ukončování úkolů.

Zobrazení probíhající činnosti při průběhu asynchronních požadavků (tam, kde je to žádoucí) je realizováno pomocí animovaného obrázku *prubeh.gif*, kterému odpovídající script mění nastavení css parametru *display* z hodnoty *hidden* na hodnotu *block*. Po ukončení probíhajícího úkolu je tento prvek znovu skryt nastavením parametru zpět na *hidden*. Zobrazení činnosti u asynchronních požadavků je z uživatelského hlediska důležité. Pokud konkrétní požadavek probíhá delší dobu, je nutné dát uživateli na vědomí, že se jeho pokyn zpracovává, protože u tohoto druhu načítání dat nedochází ke klasickému obnovení stránky nebo obrazovky, jak jsou uživatelé z webových aplikací zvyklí.

jQuery

Jedná se o jeden z často používaných frameworků, který rozšiřuje funkce jazyka JavaScript. Základem je oddělení popisu chování stránky od popisu obsahu, čehož jQuery dosahuje principem vyhledání specifikovaných elementů a popsání jejich chování na základě nastalých událostí.

Podporu knihovny jQuery oznámily i velké společnosti v čele s Microsoftem, základem ale zůstává rozsáhlá komunita vývojářů, kteří produkují množství rozšíření obvykle pod licencemi MIT nebo GNU GPL. Díky těmto zásuvným modulům je možné urychlit vývoj aplikací a věnovat více času jejich funkcionalitě.

Prvky a pluginy využívané v popisované aplikaci:

- WidgetTreeList – jQuery plugin zpracovávající nečíslované seznamy. Pomocí generovaných událostí jsou volány jQuery metody, které zajišťují rozbalení nebo sbalení sestaveného stromu. K pluginu byla doprogramována metoda, která vytvoří seznam aktuálně otevřených nodů a po obnovení obsahu dat zajistí jejich znovuotevření.
- Popup window – vyskakující okno sloužící k zobrazení detailů položek, formulářů pro přidávání a úpravy prvků a dalších typů oken.

CKEditor

Jazyk HTML nabízí k editaci textu párovou značku *textarea*, která podporuje pouze základní funkce v podobě vkládání čistého textu. Pokud má uživatel v úmyslu vložit do stránky formátovaný text a nemá odpovídající znalosti značkovacího jazyka HTML, je obvykle odkázán na externí vizuální HTML editory a následné kopírování zdrojového kódu. CKEditor je webový WYSIWYG nástroj umožňující uživatelsky přívětivým způsobem formátovat text vkládaný do webových aplikací.

Důvodem k zařazení obdobného programu do popisované aplikace byla možnost tvorby odkazů a různých poznámek na úvodní obrazovku po přihlášení uživatele. Požadavkem u těchto poznámek byla podpora formátování textu a tvorba tabulek. CKEditor byl mimo jiné vybrán i z důvodu univerzálnosti, podpory standardu XHTML 1.1 a možnosti využití správcem předdefinovaných šablon.

4.3 Adresářová struktura

Rozdělení celé aplikace do adresářů je realizováno s maximálním důrazem na jednoznačnost.

Rozložení adresářů:

- admin – adresář sloužící pro oddělení autorizované části aplikace od neautorizované, protože se v budoucím rozvoji řízení výroby počítá s možností publikace dokumentů a zveřejnění skladových zásob i neautorizovaným uživatelům.
- conf – složka obsahující soubory s konfigurací systému. Hlavním souborem v tomto adresáři je *main.conf*, který obsahuje konfigurační záznamy pro celý program a především proměnné pro připojení databáze. Při načítání vedlejších nastavení je procházen celý podadresář *modul* a všechny soubory v něm nalezené jsou následně zpracovány.
- data – do aplikace mohou být nahrávány doplňkové dokumenty, obrázky a další typy souborů, které jsou uchovávány právě v této složce.

- layout – volba rozvržení a celkový vzhled aplikace je možné přizpůsobit osobním potřebám uživatele. Konfigurace výchozí verze designu je uložena v konfiguračním souboru, nicméně uživatel má možnost tuto volbu přepsat a zvolit si design optimalizovaný pro jeho potřebu. V aplikaci jsou momentálně uloženy dvě verze rozložení, z nichž jedna je optimalizována pro dotykové ovládání a druhá počítá s klasickým ovládáním myši. Vzhled obou layoutů je totožný, optimalizovány jsou pouze velikosti ovládacích prvků tak, aby se na jednu stránku vešlo více informací.
- lib – jak již název napovídá, jedná se o adresář obsahující knihovny a aplikace poskytující dodatečnou funkcionalitu. V knihovnách jsou uloženy i třídy sloužící k autorizaci uživatele, databázový framework usnadňující práci s abstrakční vrstvou MDB2, třída *dionysos* obsahující univerzální funkce, například pro konverzi datumů, parsování URL a v neposlední řadě také třída *rizeni*, která obsahuje funkce využívané napříč všemi moduly v systému.
- modul – tento adresář obsahuje složky se soubory vztažené vždy ke konkrétnímu modulu.

4.4 Konfigurace systému

Aby bylo usnadněno nasazování systému a úprava výchozích proměnných, existují v aplikaci dvě metody pro její přizpůsobení. První z nich je základní konfigurace realizovaná v souboru *main.conf* a v souborech specifických pro jednotlivé moduly. V tomto nastavení jsou využívány dva typy deklarace proměnných a to konfigurace za pomoci funkce *define()*, která musí být spouštěna před odesláním hlaviček klientovi a vytvoří konstantu s požadovaným jménem. Druhým postupem je definice globálních proměnných *\$GLOBALS*, které jsou dostupné ve všech třídách a souborech využívaných v aplikaci. Ve zmíněných souborech najdeme mimo jiné i nastavení pro připojení databáze.

Alternativní metodou pro nastavení systému je uložení proměnné do databáze. K tomuto účelu slouží tabulka *konfigurace* a vazební entita *uzivatel konfigurace*, které uchovávají například volbu designu a velikosti fontu. Tyto proměnné mohou být ukládány ve dvou režimech, které stanoví administrátor a to pro čtení nebo zápis. Všechny operace spojené s touto agendou jsou obstarávány třídou *Konfigurace*, která kromě standardních metod pro výpis formulářů, výpisu seznamu konfigurací, uložení a upravení obsahuje také metodu pro načítání a zpracování konkrétních konfiguračních prvků z databáze. Poslední zmíněná metoda podporuje tři typy proměnných, a to proměnné lokální, globální (definované pomocí *\$GLOBALS*) a konstanty (definované pomocí *define()*). Všechny akce spojené s úpravou konfigurace jsou podmíněny pravomocí *konfigurace-sprava* nebo navázáním konkrétní položky na aktuálně přihlášeného uživatele.

4.5 Zabezpečení

Popisovaná aplikace uchovává pro společnost velmi citlivá data, proto je kladen vysoký důraz na její bezpečnost. Základním prvkem je šifrovaný (256 bitů, algoritmus SHA1) přenos dat s využitím SSL certifikátu, který je podepsán důvěryhodnou certifikační autoritou OptimumSSL. Certifikát je instalován na webovém serveru Apache s aktivovaným rozšířením modSSL, pro které je nastaven virtuální server na portu 443.

Autentizace uživatelů

Dalším prvkem zabezpečení je autentizace uživatelských účtů oproti MySQL databázi. Uživatelská jména a hesla v podobě otisku (hash) jsou uložena v tabulce *uzivatel* a provádí se tedy přímé porovnání pomocí SQL selectu.

Systém oprávnění uživatelů a skupin

V aplikaci se vyskytuje poměrně velké množství operací, které mají být dostupné pouze zvoleným uživatelům. Ideálním způsobem pro řešení oprávnění bylo respektovat role, které budou v systému existovat, vytvoření pracovních skupin na základě těchto rolí a přidělení odpovídajících pravomocí. Stav uživatelských rolí v aplikaci a jejich oprávnění je zřejmý z diagramu datových toků.

V programu je tedy vytvořeno 6 uživatelských skupin s pravomocemi přidělenými dle datových toků na DFD.

Ověřování oprávnění obstarává třída *Autorizace*, která obsahuje funkci *over_modul()*. Tato funkce realizuje kontrolu oproti tabulce *pravomoce* a seznamu přidělených pravomocí načtených na základě příslušnosti do skupin. V případě odepření práva k manipulaci s požadovanou agendou je zaznamenána chyba do hromadných chybových hlášení.

4.6 Chybová a informační hlášení

Zpětná vazba, kterou systém poskytuje ve formě chybových, úspěšných a nebo informačních hlášení, je ukládána do tří globálních polí, která jsou definována na začátku hlavního scriptu *index.php*. Všechna tři pole, tedy *\$GLOBALS['error']*, *\$GLOBALS['ok']* a *\$GLOBALS['informace']*, jsou při výpisu bufferů procházena v cyklu a v případě výskytu záznamu je tento vypsán do odpovídajícího prvku na stránce. Informační hlášení jsou automaticky skrývána po předem nastaveném čase, ale po kliku na odpovídající tlačítko je možné jejich opětovné zobrazení. Pro jednodušší orientaci jsou od sebe jednotlivé typy odděleny barevně – zelená pro úspěšná, červená pro neúspěšná a šedá pro informativní hlášení.

4.7 Indexace tabulek a vyhledávání

S ohledem na předpokládaný objem zpracovávaných dat byla aplikace postavena na tabulkách využívajících formát úložiště MyISAM, který podporuje fulltextovou indexaci obsahu. Fulltextové indexy jsou aplikovány na vybrané sloupce obsahující důležité textové informace v tabulkách zakázek, součástí a technologických postupů.

4.8 Moduly obsluhující zadávání výroby

Zadávání výroby je první činností, kterou je třeba vykonat dříve než bude zahájena realizace produktu. Při zadávání nové výroby mluvíme především o přidávání nových součástí, tvorbě technologických postupů a přiřazování součástí k zakázkám. Hlavními moduly obstarávajícími tyto činnosti jsou *součást* a *technologický postup*.

Modul *součást* je realizován jako jedna třída názvu *Soucast*, která obsahuje metody, jejichž výstupem jsou formuláře pro přidávání nových součástí a jejich případnou úpravu. Data jsou odeslána JavaScriptem pomocí asynchronního požadavku a metody ze stejné třídy určené pro zápis těchto informací se postarají o jejich uložení. Popisovaný modul dále obsahuje metody obsluhující výpis seznamu, mazání a archivaci součástí.

Modul *technologický postup* obsahuje obdobné prvky jako modul *součást*, ale s tím rozdílem, že kromě přidávání technologických postupů je také možné přidávat položky postupů a ukládat k nim nasbírané nebo předpokládané hodnoty.

Součásti i položky technologických postupů je možné přidávat hromadně, což byl jeden ze základních požadavků z důvodu zrychlení práce.

SLB - řízení výroby

hledej 78 % Dotyky

Zakázky - Všechny **Probíhající** Ukončené Zobraz - pouze archiv **Bez archivu** Vše Středisko - Všechny 3

Číslo	Název	Prac.	Vyst.	Organizace	Firma	Objednávka	Prot.	Stav
7880 SLB 41-26	EVA	2011-04-19				07511	0	

Součásti:

Název	Množství	Číslo výkresu	Typ	Termín dodání	Poznámka
SLB 41-26	35	14-SLB 41-26-00-7461	Vyráběná	2011-05-30 00:00:00	
vnější kroužek		14-SLB 41-26-10-7462	Vyráběná		
čep		14-SLB 41-26-20-7463	Vyráběná		
váleček 3x5 G2		ČSN 02 3685	Nakupovaná	VAL3,0x5,0 III	
příložný kroužek		14-SLB 41-26-75-7464	Vyráběná		

Postup:

Název	Stav	Čas vytvoření	Čas upravení	Popis
výroba - č.z.7880	nový	2011-05-10 15:04:43		
Operace Název: Nákup materiálu Vytvořeno/Upraveno: 2011-05-10 / Poznámka:	Stroj/Úživatel Typ / Čas př./Ks Ks/Změřů Datum převzetí Datum ukončení Poznámka	Kooperace / Martin 0K5/0.01Kg - předpokládána / X Tě 15,5	Kooperace / Martin 0K5/0.01Kg - 0/0 X 2011-05-11 14:08:18 2011-05-11 14:08:18 Tě 15,5	
Operace Název: Dělení Vytvořeno/Upraveno: 2011-05-10 / Poznámka:	Stroj/Úživatel Typ / Čas př./Ks Ks/Změřů Datum převzetí Datum ukončení Poznámka	pila / Martin 10/0.1 - předpokládána / X na 1000 mm, nebo zbytky	pila / Martin 10/0.1 - 72/0 X 2011-05-11 14:08:58 2011-05-11 14:08:58 zbytky materiálu	
Operace Název: Soustružení Vytvořeno/Upraveno: 2011-05-10 / Poznámka:	Stroj/Úživatel Typ / Čas př./Ks Ks/Změřů Datum převzetí Datum ukončení Poznámka	Kooperace / Martin 7K5/0Kg - předpokládána / X Málek a syn, O/320/11	Kooperace / Martin 7K5/0Kg - přebrána / X 2011-05-11 14:10:03 Málek a syn O/320/11	
Operace Název: Tepelné zpracování Vytvořeno/Upraveno: 2011-05-10 / Poznámka:	Stroj/Úživatel Typ / Čas př./Ks Ks/Změřů Datum převzetí Datum ukončení Poznámka	Kalení / Martin 0/0 - předpokládána / X QIP, na šroubu		

7857 SLB 85-30; 83-81; 83-82	EVA	2011-04-15				B707492	0	
7854 SLB 08-8, 09-3	EVA	2011-04-13				e-mail	0	
7853 SLB 88-47	EVA	2011-04-12				OV2011/00122	0	
7850 RRK120x80Q	EVA	2011-04-11				OBJ00947	0	
7849 RAK108VQM2/RK	EVA	2011-04-08				e-mailem	0	
7848 SLB 58-2	EVA	2011-04-08				139465	0	
7846 SLB 46-12, 57-10-2	EVA	2011-04-08				B707485	0	
7845 SLB 58-12-1	EVA	2011-04-08				B707453	0	
7842 SLB 44-14	EVA	2011-04-07				B707455	0	

Zobrazeno 10 z celkového počtu 90 Přesun na stránku: << < 1 2 -3- 4 5 > >>

Obr. 14 – Moduly obsluhující zadávání výroby

4.9 Sběr dat z výroby

Důležitým prvkem celého procesu, který následně umožní optimalizaci časových plánů a výpočty výrobních nákladů, je shromažďování dat o spotřebované práci. U těchto činností bylo základním požadavkem přizpůsobení pro ovládání z dotykových obrazovek, protože údaje jsou sbírány přímo v prostředí výroby a zadávat je mají za úkol samotní pracovníci. Formuláře a ovládací prvky v tomto případě obstarávají třídy *Soucast* a *Technologicky_postup* popsané v podkapitole 4.8.

Povinností každého zaměstnance je před započítáním práce přistoupit k terminálu a převzít odpovídající úkol.

Detail položky technologického postupu

Název:	Datum vložení:	Datum upravení:
Tepelné zpracování	2011-05-10 15:03:46	
Poradí operace:	Název operace:	Skupina operace:
40	Tepelné zpracování	tepelné zpracování
Stav:	Zkratka operace:	Popis operace:
čeká	Tep	
Poznámka:		

Stroj/Uživatel: Cementace / Martin
 Typ / Cas př./ks: 0/0 - předpokládána
 Ks/Změtků: /
 Datum převzetí:
 Datum ukončení:
 Poznámka: QIP

Obr. 15 – Přebírání úkolu

Po dokončení práce zvolí zaměstnanec dokončení převzatého úkolu a vyplní hodnoty odpovídající spotřebované práci. Tyto hodnoty jsou porovnány s výpočtem, kolik času zaměstnanec spotřeboval mezi převzetím a ukončením úkolu a rozdíly jsou zaznamenány.

Ukončujete položku: Tepelné zpracování

Uživatel:	Ks vyrobeno:	Ks neshodných:
Pavel Koudela		
Hodnota kontrolou:	Hmotnost:	Poznámka:
	0	

Ukončit

Obr. 16 – Ukončování úkolu

4.10 Sledování zakázky

Na základě dříve popsaných modulů je možné sestavit tabulku sledování zakázky. Ve stávajícím systému řízení jsou tyto karty realizovány papírově a zaznamenává se do nich pouze dokončení operace. Implementací modulu *sledovací karty* získá vedení společnosti mnohem komplexnější pohled na stav zakázky a zjednodušený náhled na technologický postup výroby všech součástí potřebných pro kompletaci zakázky. Ve sledovací kartě je výpis dílů rozdělen na dva typy, kterými jsou vyráběné a nakupované součásti. V případě, že je nakupovaná součást upravována je zahrnuta do obou tabulek (viz Obr. 17).

Modul *sledovací karty* v hojné míře využívá metod z tříd *Soucast* a *Technologicky_postup*. Jedná se především o načítání dat součástí, zjišťování položek technologických postupů a případně načítání detailů položek postupu včetně funkcí přebírání a dokončování úkolu.

VYRÁBĚNÉ: Název součásti	Na 1ks	Potřeba	Rozpr.	Výroba	Vykr.	Seznam operací z TP	Termín dodání: 2011-06-24	Stav
SLB 88-47 14-SLB 88-47-00-7219		1	0	1				0/0
vnější kroužek 14-SLB 88-47-10-7220	1	1	0	2		N. m. Děř So Tep BP BO BK		
vnitřní kroužek 14-SLB 88-47-20-7221	1	1	0	2		N. m. So FR Tep BP BK BO		
příruba 14-SLB 88-47-30-7222	1	1	0	2		N. m. Děř So Tep BP BO BK		
kluzné pouzdro 14-SLB 88-47-30-7223	2	2	0	3		N. m. Děř So BO		

NAKUPOVANÉ: Název součásti	Na 1ks	Potřeba	Sklad	Nákup	Potvrzený termín	Č. Objednávky	Stav
klec volnoběžky DCS475A	1	1	0	2	0000-00-00 00:00:00		Neurčen
šroub ISO 4762-M6x25-12.9 (Skl. Bn ISO 4762 (02 1143))	6	6	0	7	0000-00-00 00:00:00		Neurčen
kolík ISO 2338-Sm6x35-St (Skl. Bn ISO 2338 (02 2140))	3	3	0	4	0000-00-00 00:00:00		Neurčen

vnější kroužek, položka Nákup materiálu, hodnota Kooperace: T6120
vnější kroužek, položka Soustružení, hodnota Kooperace: KOVO-Zaplatil, O/378/10
vnější kroužek, položka Tepelné zpracování, hodnota Kalení: Bodycote
vnější kroužek, položka Bk: broušení spolu s položkou 4
vnitřní kroužek, položka Nákup materiálu, hodnota Kooperace: Tr56 2/14.8/26.6)
vnitřní kroužek, položka Soustružení, hodnota Kooperace: KOVO-Zaplatil, O/378/10
vnitřní kroužek, položka Řezání/dráčkem: výroba dráčky pro pero
vnitřní kroužek, položka Řezání/dráčkem, hodnota Kooperace: hrubý odhad
vnitřní kroužek, položka Tepelné zpracování, hodnota Kalení: Bodycote
příruba, položka Nákup materiálu, hodnota Kooperace: T6130
příruba, položka Soustružení, hodnota Kooperace: KOVO-Zaplatil, O/378/10
příruba, položka Tepelné zpracování, hodnota Kalení: Bodycote
příruba, položka Bk: broušení spolu s položkou č. 1
příruba, položka Bk, hodnota S31: čas u položky č. 1
kluzné pouzdro, položka Nákup materiálu, hodnota Kooperace: T056
kluzné pouzdro, položka BO: broušení nalisované do pozice 1

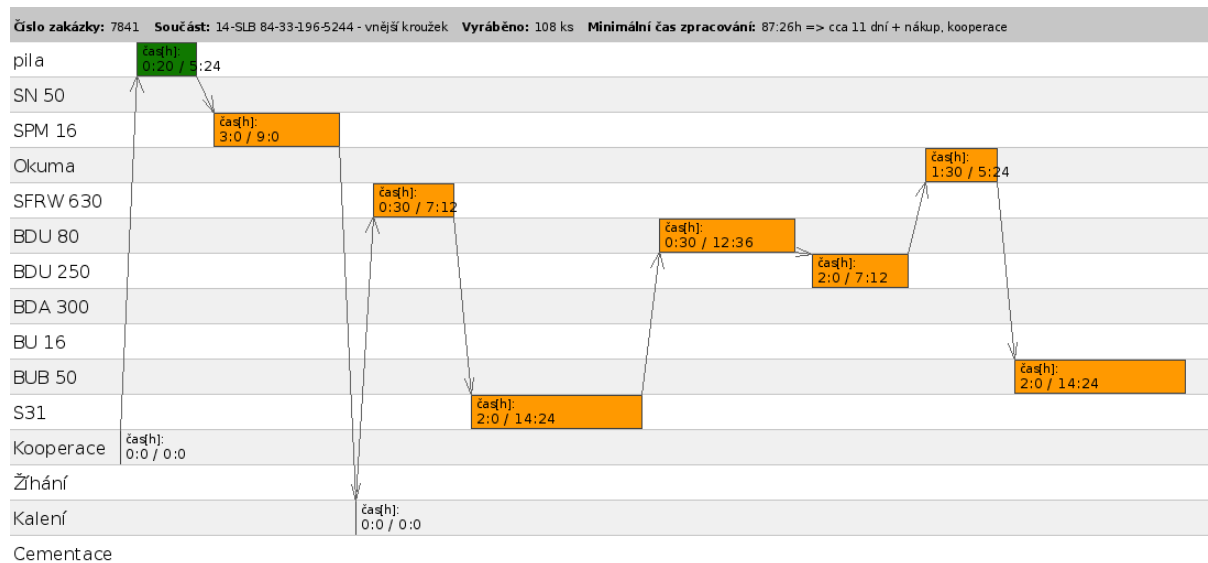
Obr. 17 – Sledovací karty

Pohledy na spotřebovaný čas

Po vstupu do sekce sledování má uživatel na výběr zobrazení sledovacích tabulek podle zakázky, součásti a nebo jejich kombinace. Tímto způsobem je zaměstnancům umožněno kontrolovat všechny realizované zakázky pro konkrétní součást včetně náhledu na vývoj spotřebovaného času. Zmíněná funkce napomáhá monitorování produktivity pracovníků a podporuje manažerské rozhodování v oblasti mezd a personalistiky. V případě testování nových technologií je také možné efektivně pozorovat dopad jejího použití.

4.11 Ganttovy grafy zobrazující postup výroby

Výpis modulu *sledovací karty* lze najít v rozcestníku pod stejnojmennou záložkou. Pokud si uživatel vybere výpis konkrétní součásti, vykreslí se mu na obrazovku také Ganttův graf, který zobrazí plán výroby daného dílu. Ganttovy grafy mohou být kresleny na základě informací z modulu *plánování*, což je čas počátku operace a číslo stroje, na kterém má být součást vyráběna. Tyto dvě informace jsou uchovány u každého dílu, který je zahrnut do plánu výroby. Poloha jednotlivých operací v obrázku je vypočítána na základě informace o jejich počátku, délce trvání a číslu stroje. K odlišení pracovních postupů je využito předem definovaného barevného odstínu.



Obr. 18 – Ganttův graf výroby součásti

4.12 Časové rozvrhy operací

Pro potřeby kapacitního plánování je v aplikaci implementován modul *plánování*, který umožňuje na základě informací z technologických postupů vytváření časových harmonogramů. Časové rozvrhy jsou realizovány pomocí vodorovných seznamů, do kterých jsou umisťovány bloky s jednotlivými operacemi. Délka objektu zobrazujícího jednu operaci je odvislá od času, který má daný úkol ve skutečnosti spotřebovat. Modul plánování využívá k ovládání metodu drag and drop, která je v jazyce JavaScript plně podporována.

Časové plány je možné znázorňovat v různých horizontech. Jedná se o zobrazení dne, týdne, měsíce a 4 měsíců a je možné mezi nimi libovolně přepínat. Z pohledu výlučnosti operací je možné zobrazení případných kolizí, které mohou nastat v situaci, že budou operace jdoucí po sobě naplánovány s překryvem. Tyto stavy jsou znázorněny tak, že dojde ke zčervenání příslušného bloku, pokud se nachází v kolizi. Překryvy operací není možné obecně zakázat, protože je dokonce žádoucí, aby některé činnosti probíhaly současně.

Popisovaný modul tedy obsahuje stejnojmennou třídu *Plánování*, která zajišťuje kompletní funkcionalitu popsanou výše.

Jmenovitě jde o:

- výpis časových plánů,
- vyřazování již hotových úkolů,
- načítání nově přidáných položek,
- ukládání pořadí úkolů,
- zobrazení historie,
- přepínání časového období.

Další funkcionalitu, jako jsou například informace o zakázce, ke které úkol patří, podrobnosti o operaci a technologickém postupu nebo třeba přebírání a ukončování úkolů, zajišťují třídy popsané v předchozích odstavcích.



Obr. 19 – Plánování operací

4.13 Výpočet nákladů výroby

Ekonomický pohled na výrobu je pro vedení společnosti důležitým prvkem. Základními požadavky v tomto ohledu jsou:

- prověření rentability výroby,
- tvorba cenových nabídek,
- kontrola výše marže na výrobcích.

Pro realizaci výpočtů tohoto typu je třeba znát o každé součásti základní informace a postupy výroby. Pokud je díl nakupovaný nebo upravovaný, je nutné znát jeho nákupní cenu. Je-li vyráběný, je třeba mít informace o jakosti materiálu, váze přířezu potřebného pro výrobu, případně znát cenu, kde-li o speciální typ (výkovek, odlitek atd.). K vyráběným a upravovaným součástem je také potřeba výrobní postup s položkami ohodnocenými předpokládaným nebo reálně spotřebovaným výrobním časem.

Zadávání technologických postupů s těmito situacemi počítá a podle typu položky umožňuje zadávání ceny, hmotnosti nebo předpokládaného/spotřebovaného času. U položky materiál je z důvodu jeho kolísající ceny uváděn typ a hmotnost přířezu. V tabulce materiálů je potom uvedena průměrná cena za kilogram. Nakupované součásti mají cenu zadávanou fixně, ale je možné ji kdykoliv měnit. Cena práce je ve společnosti SLB stanovena vzhledem ke stroji, na kterém operace probíhala, a proto je ukládána jako atribut do tabulky *stroj*.

Vypočtené náklady na výrobu součástí je možné zobrazovat ve formě podrobného výpisu pro jednu hlavní součást se všemi detaily o spotřebovaném čase a cenou a nebo ve formě stručného seznamu, ve kterém je vypsán pouze název hlavní součásti a informace o vypočítané ceně. Do stručného výpisu jsou také zaznamenány díly, u nichž je nákupní cena i cena výroby rovna nule, protože je zde splněn předpoklad, že by mohlo jít o nevhodně vyplněnou položku.

SLB - řízení výroby

Zakázka: 7841 Součást: SLB 84-33 Suma všech součástí

Legenda: **Předpokládá** **Dokončená**

Č.z.: 7841 Název SLB 84-33 Firma: **Technické služby Brno** Č. obj.: B707445 Vystavená: 2011-04-06 Protokol: 0 Stav:

Název	Číslo výkresu	Operace	Stroj	Ks	Čas přip	Čas součást	Cena	Hmotnost	Cena přip	Cena součást	Poznámka hodnota	Poznámka položka
SLB 84-33	14-SLB 84-33-008-5243			1			0					
Mala suma součást: cena příprava: OKČ, OKČ cena součást: OKČ, OKČ cena kooperace: OKČ, OKČ												
vnější kroužek	14-SLB 84-33-196-5244			1			0					
		Nákup materiálu	Kooperace	1	0 0	0 0	0 0	1.95 0	0 0	0 0	Tr62/11,75/(38,5)	0
		Dělení		1	0 0	0 0	0 0	0 0	0 0	0 0		0
		Soustružení		1	0 0	0 0	0 0	0 0	0 0	0 0		0
		Tepelné zpracování	Kalení	1	0 0	0 0	0 0	1.3 0	0 0	0 0		28,8
		BP		1	0 0	0 0	0 0	0 0	0 0	0 0		28,8
		BK		1	0 0	0 0	0 0	0 0	0 0	0 0		28,8
		BO		1	0 0	0 0	0 0	0 0	0 0	0 0		28,8
		BO		1	0 0	0 0	0 0	0 0	0 0	0 0		28,8
		Soustružení		1	0 0	0 0	0 0	0 0	0 0	0 0		28,8
		BK		1	0 0	0 0	0 0	0 0	0 0	0 0		28,8
Mala suma součást: cena příprava: OKČ, OKČ cena součást: OKČ, OKČ cena kooperace: 28,8OKČ, OKČ												
hřídel	14-SLB 84-33-296-5245/3			1			0					
		Nákup materiálu	Kooperace	1	0 0	0 0	0 0	1.31 0	0 0	0 0	T635	0
		Dělení		1	0 0	0 0	0 0	0 0	0 0	0 0		0
		Soustružení		1	0 0	0 0	0 0	0 0	0 0	0 0		0
		Tepelné zpracování	Kooperace	1	0 0	0 0	0 0	0.8 0	0 0	0 0		0
		BK		1	0 0	0 0	0 0	0 0	0 0	0 0		0
		Soustružení		1	0 0	0 0	0 0	0 0	0 0	0 0		0
Mala suma součást: cena příprava: OKČ, OKČ cena součást: OKČ, OKČ cena kooperace: OKČ, OKČ												

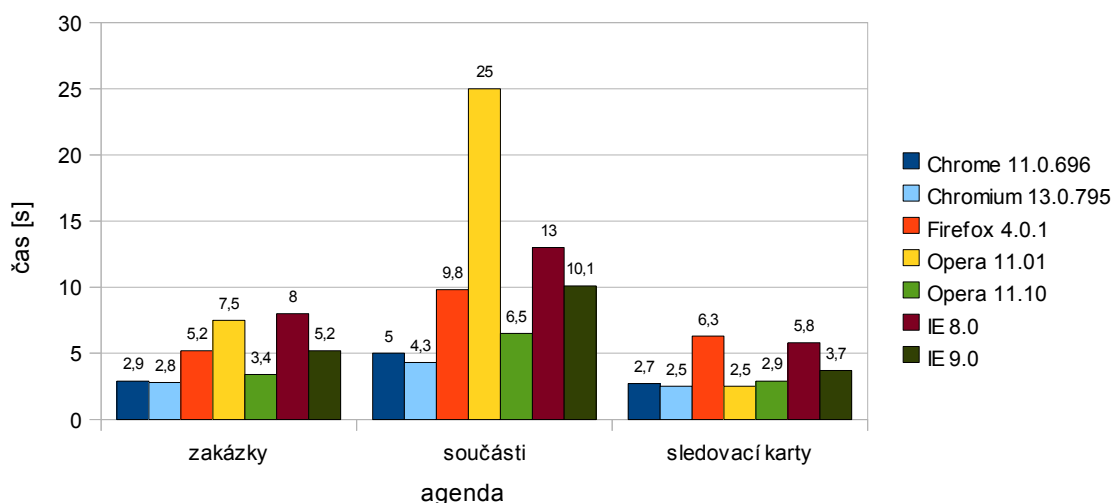
Obr. 20 – Výpočty nákladů na výrobu

5 NASAZENÍ APLIKACE

V závěru vývoje popisovaného programu přišlo na řadu nasazení v prostorách společnosti SLB. Při zavádění informačních systémů nastává řada standardně definovatelných problémů, kterým je nutné předcházet optimalizací zaváděcího procesu. V tomto případě bylo přistoupeno k tzv. pilotnímu zavádění, při kterém je systém zprovozněn na jednom zkušebním pracovišti a testován v reálných podmínkách. Tímto způsobem je možné identifikovat potenciálně problematické části aplikace, aniž by došlo ke „znechucení“ všech zaměstnanců případnou nedokonalostí nově nasazovaného softwaru. Ve firmě pracují lidé všech věkových kategorií (rozmezí cca 20 až 70 let), proto bylo nutné zvolit poměrně pozvolné zavádění aplikace do provozu.

5.1 Nasazení systému v řízení výroby

Startovacím pracovištěm, na kterém byl umístěn první dotykový terminál, se stala kancelář vedoucích výroby. Z důvodu současného zavádění nového konstrukčního softwaru nebyla v první fázi realizace systému implementována funkcionality pro automatické načítání kusovníků. Veškeré zadávání součástí tedy bylo nutné realizovat ručně za pomoci formulářů k tomu určených. Zadávání dat probíhalo na klasických počítačích bez využití dotykových displejů, proto byl k tomuto účelu přednostně využíván design optimalizovaný pro standardní ovládání s využitím klávesnice a myši. Počáteční plnění daty bylo i vzhledem k implementaci dávkového zadávání nových součástí poměrně rychlé. Původní cíl – přidávání pouze nově přichozích zakázek – byl proto pozměněn na systém označitelný jako „last in first out“, tj. do agendy byly postupně zadávány zakázky s nejstaršími daty založení. Při zadávání nových dílů je využívána agenda zakázek, která poskytuje komplexní pohled na seznam aktivních zakázek včetně vyráběných součástí a technologických postupů. V této části aplikace se nejvíce projevuje náročnost funkcí na klientské straně. Veškeré klientské funkce jsou napsány ve skriptovacím jazyce JavaScript a jeho svižnost je značně ovlivněna interpretem tohoto jazyka. Aktuálně nejrychlejším se v tomto směru jeví engine V8 [14] použitý v prohlížeči Google Chrome [15]. Z tohoto důvodu bylo rozhodnuto o jeho nasazení na dotykové terminály a všechna ostatní pracoviště, kde to bylo možné.



Obr. 21 – Průměrná rychlost prohlížečů při zobrazení jednotlivých agend

Dalším krokem po prvotním nasazení byla optimalizace designu aplikace. Při zadávání technologických postupů se projevila nutnost vytvoření kompaktnějšího formuláře přidávání položek. Téměř každý technologický postup obsahuje více než pět položek a při použití grafikem navrženého rozložení bylo nutné ve velké míře posouvat stránku. Protože množství záznamů přidávaných naráz není omezeno a vzhledem k využití aplikace na zařízeních s různým rozlišením, nelze optimalizovat

formuláře tak, aby bylo možné posouvání stránek plně eliminovat.

Po prvním měsíci používání se ukázalo, že nejdůležitějším prvkem systému budou sledovací karty, které poskytnou nejrychlejší a nejkompaktnější pohled na aktuální stav výroby. V prostředí sledování je možné realizovat úpravy součástí a postupů, přebírání a dokončování úkolů a vyplňování dat dodání u nakupovaných součástí. Dalším důležitým prvkem, který byl do sledování přidán v průběhu testování, je možnost alokace zbytků ke konkrétní zakázce, stejně tak jako jejich okamžitá úprava v případě nutnosti.

5.2 Dílenské terminály

Jako dílenské terminály jsou nyní použity počítače integrované do monitoru s dotykovou obrazovkou. Zásadním důvodem pro výběr spotřebního hardwarového vybavení byl předpoklad jeho rychlé a bezproblémové dostupnosti nyní i v následujících letech. U popisovaných počítačů je aktuálně příznivá i cena, která se odvíjí od využití hromadně vyráběných komponent. Zvoleny byly počítače s infračervenou dotykovou technologií, která je sice zatížena větším zpožděním [16], ale díky které je možné terminál ovládat jak prsty, tak i libovolnými jinými předměty. Dotykový LCD počítač je již z výroby krytý sklem a proti poškrábání byl dodatečně opatřen i vyměnitelnou fólií. Umístění terminálů na dílnách, kde již před instalací nebyl přebytek prostoru, bylo poměrně obtížné a na všech místech způsobovalo jistá omezení. Jako výsledné řešení byly vedením společnosti vybrány VESA držáky pro umístění na zeď, čemuž byl přizpůsoben i výběr konkrétních terminálů. Vybraná ramena umožňují pohyb terminálu v rozmezí několika desítek stupňů ve vertikálním i horizontálním směru.



Obr. 22 – MSI AE2040 – terminál umístěný v dílně na stěně

6 ZÁVĚR

Nasazená aplikace, která podporuje procesy probíhající při řízení výroby, kompletně splňuje požadavky zadavatele a znamená významný posun vpřed v organizaci popisované strojírenské produkce. Díky postupnému vývoji, který byl započat provedením podrobné analýzy systému a následným pravidelným testováním funkcionality, byl vytvořen modulárně řešený program usnadňující sledování a řízení kusové výroby speciálních ložisek. Aplikace se vyznačuje vysokou dostupností, jednoduchou rozšiřitelností a optimální složitostí, která nevnáší do zpracovávaných úkolů zbytečnou nejednoznačnost.

Uživatelské role systému jsou striktně odděleny a neoprávněný uživatel se tedy nemůže dostat k citlivým informacím. K implementaci zmíněné funkcionality byla využita správa uživatelských skupin s rozdělováním pravomocí. Celkové bezpečnosti aplikace byla věnována velká pozornost právě s ohledem na charakter zpracovávaných údajů. Přístup k datům je možný i z Internetu, a proto byl nasazen důvěryhodnou certifikační autoritou podepsaný SSL certifikát pro šifrované spojení, který celkové zabezpečení dále zvyšuje.

Výsledný program je navržen pro dva typy ovládání – standardní (myš, klávesnice) a dotykové. K návrhu, implementaci i provozu byl využit volně dostupný software uvolněný především pod licencemi GNU/GPL a MIT. Díky využitým knihovnám probíhal celý vývoj poměrně rychle a bez větších problémů. Používání programu je intuitivní, v návrhu datových struktur jsou pouze zadavatelem požadované atributy a nedochází tedy ke vzniku nepřehlednosti. Kompletní uživatelské prostředí bylo vytvořeno na míru a je předmětem neustálého vývoje v podobě přemísťování prvků pro dosažení ideální ovladatelnosti. V rámci prací na popisovaném systému bylo implementováno celkem devět modulů, které mezi sebou navzájem využívají a dědí funkce pro dosažení optimálních výstupů bez duplikace kódu. Přibližně 80 % událostí je v aplikaci realizováno za pomoci asynchronní komunikace klienta se serverem. Zmíněný přístup pomáhá šetřit přenosovou kapacitu sítě a výrazně zvyšuje uživatelskou přívětivost tím, že nedochází k neustálému obnovování stránky s aplikací. S tímto souvisí i jedna z nejobtížnějších částí práce, kterou bylo vytvoření funkce pro aktualizaci stromové struktury při asynchronně provedených úpravách. Tento úkol byl vyřešen naprogramováním univerzální metody pro všechny objekty, které stromovou strukturu využívají. Funkce se podle zadaných parametrů rozhodne, který strom a která jeho část má být obnovena, čímž je dosaženo optimálních výsledků z pohledu rychlosti i uživatelské přívětivosti.

Požadavky kladené vedením společnosti na systém byly splněny a v mnoha případech bylo dokonce dosaženo rozsáhlejší funkcionality, než bylo původně plánováno. Podkladem k tomuto tvrzení je aktivní využívání vyvinuté aplikace v době dokončování této práce společně s navrhováním a implementací nových modulů. Základním aspektem, který plně uspokojil zúčastněné strany, je velmi rychlý vývoj nových funkcí, které mohou čerpat z velkého množství metod v již hotových objektech.

Jedním z nově připravovaných prvků je evidence nabídek a poptávek ve vztazích dodavatel-odběratel. V tomto modulu, který přímo navazuje na řízení výroby, jsou navrženy čtyři číselné řady (dle stávajícího popisu procesu) pro nabídky, poptávky, objednávky a potvrzení objednávek. Tyto dokumenty budou napojeny na součásti a zakázky, čímž se zajistí ještě rychlejší informovanost vybraných zaměstnanců. Dalším zvažovaným prvkem, souvisejícím s implementací nových funkcí a modulů, je podpora archivace metrologických údajů přímo v elektronické složce zakázky. Veškeré, metrologickou laboratoří kontrolované, parametry vyráběných produktů jsou momentálně archivovány v písemné formě a hledání v takovýchto záznamech je časově náročné. Některé metrologické přístroje navíc produkují protokoly v elektronické podobě a jejich archivace by měla pozitivní důsledek na skladovací prostory, pohodlnější a rychlejší vyhledatelnost záznamů a úsporu nezanedbatelných nákladů na tisk.

SEZNAM POUŽITÉ LITERATURY

- [1] Wikipedia. *ERP: Enterprise Resource Planning* [online]. 8.5.2011 [cit. 11.5.2011]. Dostupné z: <http://cs.wikipedia.org/wiki/Enterprise_resource_planning>.
- [2] Wikipedia. *MES: Manufacturing Execution Systems* [online]. 6.5.2011 [cit. 11.5.2011]. Dostupné z: <http://en.wikipedia.org/wiki/Manufacturing_execution_system>.
- [3] *Debian GNU/Linux: O projektu Debian* [online]. 15.2.2011 [cit. 26.2.2011]. Dostupné z: <<http://www.debian.org/intro/about>>.
- [4] Alberton, Lorenzo. *PHP Pear: MDB2* [online]. 29.8.2010 [cit. 6.3.2011]. Dostupné z: <<http://pear.php.net/package/MDB2>>.
- [5] Prevost, Andy; Jagielski, Jim. *PHPMailer manual* [online]. 2008 [cit. 7.4.2011]. Dostupné z: <<http://phpmailer.worxware.com/>>.
- [6] *jQuery documentation* [online]. 2011 [cit. 5.4.2011]. Dostupné z: <<http://docs.jquery.com>>.
- [7] Knabben, Frederico. *CKEditor Documentation* [online]. 3.3.2011 [cit. 7.4.2011]. Dostupné z: <<http://docs.cksource.com/>>.
- [8] Tufat. *html2ps and html2pdf specification* [online]. 6.9.2010 [cit. 16.4.2011]. Dostupné z: <http://www.tufat.com/s_html2ps_html2pdf.htm>.
- [9] Sochor, Jan. *Analýza a návrh systémů. Brno, Fakulta informatiky Masarykovy univerzity*. 2002 [cit. 17.4.2011]. Dostupné z: <<http://www.fi.muni.cz/~sochor/PB007/>>.
- [10] Schutta, T. Nathaniel; Asleson, Ryan. *AJAX: vytváříme vysoce interaktivní webové aplikace*. Computer Press, 2006. 269 s. ISBN: 80-251-1285-3.
- [11] Wikipedia. *JavaScript* [online]. 3.5.2011 [cit. 8.5.2011]. Dostupné z: <<http://cs.wikipedia.org/wiki/JavaScript/>>.
- [12] *Ecma international* [online]. [cit. 10.5.2011]. Dostupné z: <<http://www.ecma-international.org/>>.
- [13] *ISO: International Organization for Standardization* [online]. 2011 [cit. 10.5.2011]. Dostupné z: <<http://www.iso.org/iso/about.htm>>.
- [14] Majda, David. *V8: JavaScript uvnitř Google Chrome* [online]. 22.1.2009 [cit. 12.5.2011]. Dostupné z: <<http://zdrojak.root.cz/clanky/v8-javascript-uvnitř-google-chrome/>>.
- [15] Google. *Google Chrome* [online]. 2010 [cit. 12.5.2011]. Dostupné z: <<http://www.google.com/chrome/intl/cs/more/index.html>>.
- [16] Vekobs s.r.o.. *Technologie: Infrared touchscreens* [online]. 2011 [cit. 12.3.2011]. Dostupné z: <http://www.vekobs.cz/cz_technologie.htm>.
- [17] Micro-Star Int'l Co., Ltd.. *Wind Top AE2040* [online]. 2011 [cit. 12.3.2011]. Dostupné z: <<http://eu.msi.com/product/aio/Wind-Top-AE2040.html>>.
- [18] Majda, David. *SpiderMonkey: rychlá kompilace JavaScriptu do nativního kódu* [online]. 15.1.2009 [cit. 12.5.2011]. Dostupné z: <<http://zdrojak.root.cz/clanky/spidermonkey-rychla-kompilace-javascriptu/>>.
- [19] Resig, John. *JavaScript a Ajax: moderní programování webových aplikací*. Brno: Computer Press, 2007. 360 s. ISBN: 978-80-251-1824-5.
- [20] Gilmore, J. W.. *Velká kniha PHP 5 a MySQL: kompendium znalostí pro začátečníky a profesionály*. Zoner press, 2005. 711 s. ISBN-80-86815-20-X.
- [21] Koudela, Pavel. *Internetová prezentace SKAS MU*. Brno, 2008. 36 s. Bakalářská práce na Fakultě informatiky Masarykovy univerzity. Vedoucí diplomové práce Mgr. Jan Holeček. Dostupné z: <http://is.muni.cz/th/98541/fi_b/>.
- [22] Rosebrock, Eric; Filson, Eric. *Linux, Apache, MySQL a PHP: instalace a konfigurace prostředí pro pokročilé webové aplikace*. Grada, 2005. 344 s. ISBN 80-247-1260-1.
- [23] Brandejs, Michal. *Linux: praktický průvodce*. Konvoj, 2003. 312 s. ISBN 80-7302-050-5.
- [24] Kosek, Jiří. *HTML: tvorba dokonalých WWW stránek*. Grada Publishing, 1998. 296s. ISBN 80-7169-608-0.
- [25] Kosek, Jiří. *PHP: tvorba interaktivních internetových aplikací*. Grada Publishing, 1999. 502s. ISBN 80-7169-373-1. Dostupné z: <<http://www.kosek.cz/php/php-tvorba-interaktivnich-internetovych-aplikaci.pdf>>.