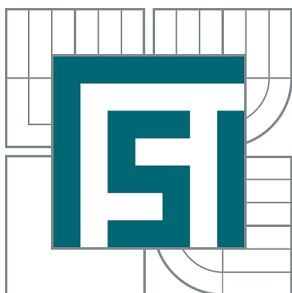


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

GANTTOVY DIAGRAMY

GANTT CHARTS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN KANTAR

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. SIMEON SIMEONOV, CSc.

BRNO 2014

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Martin Kantar

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

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 diplomové práce:

Ganttovy diagramy

v anglickém jazyce:

Gantt charts

Stručná charakteristika problematiky úkolu:

Analýza funkcí Ganttových diagramů.

Návrh programového vybavení pro Ganttovy diagramy za využití OOP.

Cíle diplomové práce:

1. Analýza funkcí Ganttových diagramů.

2. Návrh programového vybavení pro Ganttovy diagramy za využití OOP.

3. Naprogramování DLL knihovny pro Ganttovy diagramy.

Seznam odborné literatury:

REMBOLD, U. Computer Integrated Manufacturing and Engineering. Wokingham: Addison-Wesley, 1994, 640 s. ISBN 02-015-6541-2.

MOORE, Ron. Selecting the right manufacturing improvement tools: What tool? When?. Burlington: Butterworth-Heinemann, 2007, xxii, 390 s. ISBN 07-506-7916-6.

PINEDO, Michael L. Planning and scheduling in manufacturing services. New York: Springer Science, 2005, 506 s. ISBN 03-872-2198-0.

John Wiley and Sons, Inc. Brandimarte P., Villa A. Modeling manufacturing system: from aggregate planning to real-time control. Berlin: Springer, 1999, 215 s. ISBN 35-406-5500-X.

HARRISON, David K. Systems for planning and control in manufacturing: systems and management for competitive manufacture. 1st ed. Oxford: Newnes, 2002, xiv, 297 s. ISBN 07-506-4977-1.

Vedoucí diplomové práce: doc. Ing. Simeon Simeonov, CSc.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2013/2014.

V Brně, dne 17.12.2013

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan fakulty

ABSTRAKT

Diplomová práce je zaměřená na Ganttovy diagramy. Především na jejich využití ve výrobních procesech, jako je plánování a rozvrh výroby. Obsahuje úvod, vysvětlující co to jsou Ganttovy diagramy, k čemu slouží a jaký je jejich princip. Diplomová práce dále obsahuje analýzu a programové možnosti realizace Ganttových diagramů za využití OOP. Programovou a teoretickou analýzu jednotlivých funkcí, které jsou charakteristické pro Ganttovy diagramy, včetně jejich vnitřní logiky a programové implementace. Tyto naprogramované charakteristické funkce Ganttových diagramů jsou zkomponovány ve formě DLL knihovny, která nabízí jednoduchou realizaci Ganttových diagramů. Pomocí této svojí knihovny je dále realizován celý software pro řízení výroby. K práci je přiložen na DVD kromě vytvořené DLL knihovny i samotný software s detailními popisy jednotlivých částí zdrojového kódu.

ABSTRACT

The thesis is aimed at Gantt charts. Primarily on their use in production processes, such as planning and scheduling. The thesis also includes an introduction, explaining what are the Gantt charts, for what are using and what is their principle. The thesis also includes analysis and software possibilities for implementing Gantt charts with using OOP. Programmatic and theoretical analysis of the various functions that are characteristic of Gantt charts, including their internal logic and program implementation. These programmed characteristic functions of Gantt charts were composed in the form of a DLL library, which provides a simple implementation of Gantt charts. By using this library is realized the software for production management. The work is included on the DVD, except the DLL itself software with detailed descriptions of each part of the source code.

KLÍČOVÁ SLOVA

Ganttovy diagramy, řízení výroby, DLL knihovna, OOP, C#.

KEYWORDS

Gantt charts, control of production, DLL library, OOP, C#.

PROHLÁŠENÍ O ORIGINALITĚ

Prohlašuji, že jsem diplomovou práci na téma Ganttovy diagramy vypracoval samostatně s využitím odborné literatury a pramenů, uvedených v Seznamu použité literatury.

Datum

Martin Kantar

BIBLIOGRAFICKÁ CITACE

KANTAR, M. *Ganttovy diagramy*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 49 s. Vedoucí diplomové práce doc. Ing. Simeon Simeonov, CSc..

PODĚKOVÁNÍ

Děkuji tímto panu doc. Ing. Simeonu Simeonovi, CSc. za odborné rady a připomínky v oblasti plánování výroby při vypracovávání diplomové práce.

Obsah:

Zadání závěrečné práce	3
Abstrakt.....	5
Prohlášení o originalitě	7
Seznam použitých symbolů	10
1 Úvod	11
2 Popis Ganttových diagramů.....	12
2.1 Základní princip	12
2.2 Historie	12
2.3 Použití.....	15
2.3.1 Metoda CPM	15
2.3.2 Metoda PERT	15
3 Analýza a návrh programového vybavení.....	16
3.1 Plocha diagramu	16
3.2 Procesní činnosti	19
3.3 Priorita	23
3.3 Souvislost	23
4 Implementace knihovny.....	28
4.1 Vykreslení mřížky.....	28
4.2 Vykreslení zakázky	29
4.3 Vykreslení pracoviště	30
5 Realizace softwaru.....	31
5.1 Hlavní menu	31
5.1.1 Nový soubor	31
5.1.2 Otevření souboru	31
5.1.2.1 Textový soubor	31
5.1.2.2 Databázový soubor	33
5.1.3 Uložení souboru	34
5.1.4 Rychlé uložení	35
5.1.5 Rychlé načtení	36
5.2 Tabulka procesů.....	37
5.2.1 Zobrazená data	37
5.2.1 Vybrání položky	39
5.3 Manipulace s daty.....	40
5.3.1 Úprava	40
5.3.2 Přidávání.....	41
5.3.3 Odebírání.....	42
5.3.4 Filtrování	42
5.3.5 Změna diagramů.....	43
5.4 Rozložení softwaru.....	44
5.5 Výsledný software.....	46
6 Závěr.....	47
Seznam použité literatury.....	48
Příloha	49

SEZNAM POUŽITÝCH SYMBOLŮ

OOP	Object-Oriented Programming
CPM	Critical Path Method
PERT	Programm Evaluation Review Technique
WPF	Windows Presentation Foundation
WF	Windows Form
API	Application Programmin Interface
GDI	Graphics Device Interface
XAML	Extensible Application Markup Language
DVD	Digital Video Disc

1 ÚVOD

V dnešní době jsou stále více kladeny důrazy na ekonomické náklady téměř ve všech oblastech lidských činnostech. Lidé se snaží si zajistit co největší přínos za cenu nejmenších nákladů a tedy dosáhnout maximální efektivity. V dřívějších dobách se pro dosažení lepší efektivity používaly například grafy a tabulky. Toto použití mělo ovšem několik problémů. Mezi hlavní patřilo například to, že takto vytvořený graf na papíře byl v podstatě jen na jedno použití. Vypracování takovýchto grafů dejme tomu třeba každý den by bylo ohromně časově náročné. Jednalo se spíš o jakési statické ukazatele, které sice efektivitu dokázaly vylepšit, ale nedokázaly jejich potenciál využít na maximum. Z tohoto důvodu začaly vznikat různé softwary simulující lidské úkony, které mají za účel lidem usnadnit zjištění, jak této maximální efektivity dosáhnout. Softwary mají oproti běžnému papíru několik výhod. Největší výhodou je, že dovedou informovat uživatele o dynamických informacích, kdežto papír jen o statických. Na papíru nikdy nebude aktuální stav, ale pouze stav, který byl při začátku vypracování. Běžně se totiž stává, že dojde k neočekávaným situacím a je nutné provést přeplánování. Další výhodou samou o sobě je i univerzálnost. Pokud se vytváří nějaký graf na papír, tak se musí většinou začít vždy úplně od znovu, oproti tomu software nabízí už nějakou základní entitu, která je již vypracovaná. V dnešní době už existují tisíce takovýchto softwarů sloužících k usnadnění lidské činnosti. V posledních letech jde vysledovat určitý přesun od klasických softwarů pracujících na počítači na softwary pracujících v internetových prohlížečích. Tento přesun je logickým vyústěním snahy usnadnit uživateli celkovou práci. Odbourává tak nutnost samotný software stahovat a instalovat na počítači, stejně tak se starat o uložená data. Tyto věci bývají v čisté režii zprostředkovateli těchto internetových aplikací. Klasické softwary na počítače ale také nezahálejí. Je to především z toho důvodu, že například firmy potřebují mít svůj software na míru svým požadavkům, stejně tak pokud pracují například s důvěrnými daty, tak pro ně může působit poměrně nebezpečně spoléhat se na zabezpečení těchto dat nějakou cizí osobou.

Cílem této práce je provést analýzu Ganttových diagramů, vytvořit DLL knihovnu pro vykreslování Ganttových diagramů, jakožto poměrně silného nástroje v oblasti vytváření rozvrhů a plánů. Pomocí této knihovny může být poměrně komplikovaná oblast vykreslování lehkou záležitostí. Dalším cílem je vytvoření softwaru, využívající tuto naprogramovanou knihovnu, sloužícího k plánování výrobních procesů. Při tvorbě softwaru byl kladen důraz na co největší univerzálnost pro uživatele. Především co se zpracování dat a jejich samotného nahrání do programu týče. Samozřejmě velkou roli hraje i samotná manipulovatelnost se softwarem, kdy by uživatel nemusel procházet složitým školením, ale měl by být schopen intuitivně software ovládat. Součástí práce je na DVD tedy samotná knihovna i software, který jí využívá. Na DVD jsou přiloženy dále zdrojové kódy těchto obou částí a nějaká uměle vytvořená vzorová data, které program umí zpracovat.

2 POPIS GANTTOVÝCH DIAGRAMŮ

Ganttův diagram je nástroj, sloužící v různých oblastech lidské činnosti, především v oblastech plánování a navrhování. Jeho diagramová struktura je nejčastěji ve formě obdélníků, přičemž jednotlivé obdélníky znázorňují vždy určitou činnost plynoucí v čase – ve směru časové osy. Ganttovy diagramy nabízí poměrně dobrý poměr - co se týče množství informace na jednotku plochu. Je to především z toho důvodu, že pouhý jeden takový obdélník dokáže tím, že je umístěn na časové ose, poskytovat poměrně důležité a lehce čitelné informace i úplnému laikovi. Mimoto dokáže s vhodně zvoleným propojením jednotlivých úkonů informovat o tom, v jakém pořadí se má na čem začít.

Tyto diagramy neslouží pouze k zobrazování událostí, které se staly v minulosti. Ba naopak, z těchto informací z minulosti dokáže predikovat - zobrazovat co se teprve stane [2].

2.1 Základní princip

Princip Ganttových diagramů je poměrně prostý a současně velmi i účelný. A to dokázat vystihnout danou procesní problematiku ve formě diagramů. Tedy přesněji jí nejlépe popsat pomocí obrázku, který je na první pohled lehce čitelný. Než se objevily Ganttovy diagramy, tak se všechny plánované činnosti popisovali pouze slovně, popřípadě ve formě tabulek, což mohlo působit jak na jednu stranu nepřehledně, tak velmi zmatečně a to především z toho důvodu, že slovní pojetí problematiky je sama o sobě poměrně subjektivní záležitost. Přičemž popsanou problematiku může chápat jinak nezkušený člověk v dané oblasti, tak specializovaný pracovník. V podstatě se dá mluvit o diagramu jako univerzálním jazyku, kterému dokáže lehce porozumět jakýkoliv člověk bez potřeby nějakého vyššího vzdělání. To, že jsou Ganttovy diagramy lehce srozumitelné, velice napomohlo jejich samotnému rozvoji. Jejich uživatelé totiž nemuseli trávit dlouhý čas se seznamováním se s jejich problematikou, popřípadě dlouhými školeními.

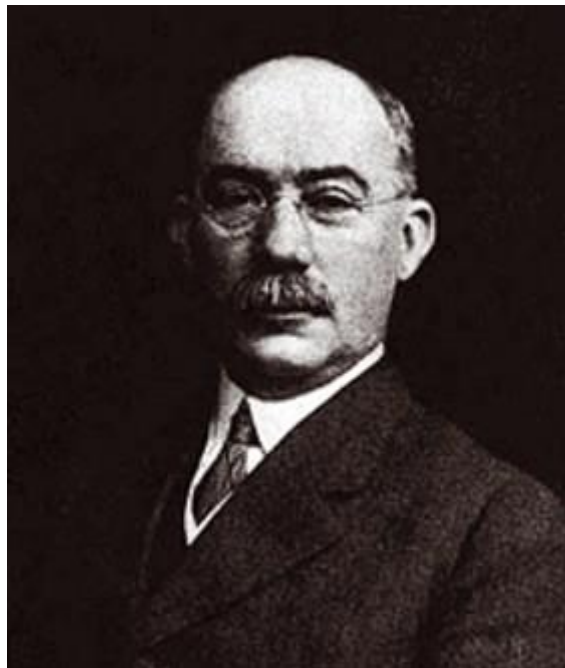
Ganttovy diagramy by měly být tak srozumitelné, že by z nich měla dokázat číst osoba, která je bude obsluhovat, tak i třeba pouhý řadový zaměstnanec. Pokud zaměstnanci budou mít přístup k diagramům například plnění práce, tak budou moc být na svojí práci mezi kolegy pyšní. Stejně tak to může být impulz pro špatné zaměstnance se sebou něco dělat.

2.2 Historie

Ganttovy diagramy přišli na svět v době, kdy byly velmi zapotřebí. V té době se podniky staraly především jen o výrobní provedení, jakost a množství. Nikdo moc nehleděl na čas jako proměnnou veličinou ve výrobě. Za autora těchto diagramů je všeobecně považován americký strojní inženýr Henry Laurence Gantt (1861-1919), který vytvořil svůj první diagram kolem roku 1910. Ovšem ve skutečnosti byl první Ganttův diagram vytvořen už v roce 1896 Karolem Adamieckim, tehdy pod názvem Harmonogram. Adamiecki se ovšem nikdy výrazně nesnažil o propagaci v jiném jazyce než v rodné polštině nebo ruštině. Mnoho lidí z Polska a Ruska dodnes nazývají Ganttův diagram Harmonogramem [1]. Hlavním obdobím, kdy se diagramy začaly hojně využívat v průmyslu a docházelo k jejich zdokonalování, se odehrálo v USA za období světové války.

H. L. Gantt se nikdy nepokoušel o patentování nebo ochranu autorského práva svých diagramů. Nejen dával vzory každému, kdo o ně požádal, ale také je uveřejnil v mnohých odborných člancích. Byl vždy rád, když ostatní mohli těžit z jeho znalostí. [2].

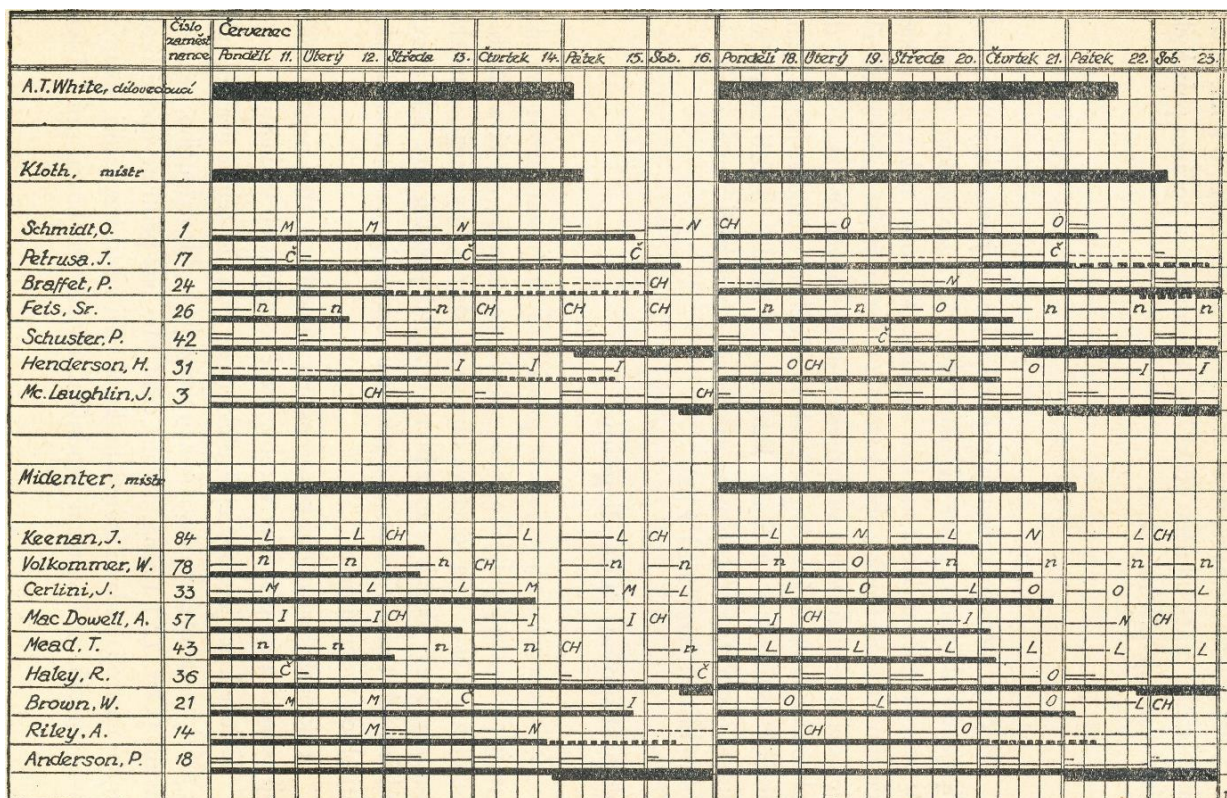
Henry L. Gantt byl také autorem mzdové úkolové soustavy, která se v dnešní době velmi dobře uplatňuje v kombinaci s Gantovými diagramy. Její vznik se datuje kolem roku 1903. V této soustavě jsou zaměstnancům, kteří se podíleli na procesech, které byly dokončeny ve stanovené lhůtě, přidány finanční odměněny. Samozřejmě výše odměny se také odvíjela od doby, kdy byl v předstihu dokončen proces před původně plánovaným ukončením [2].



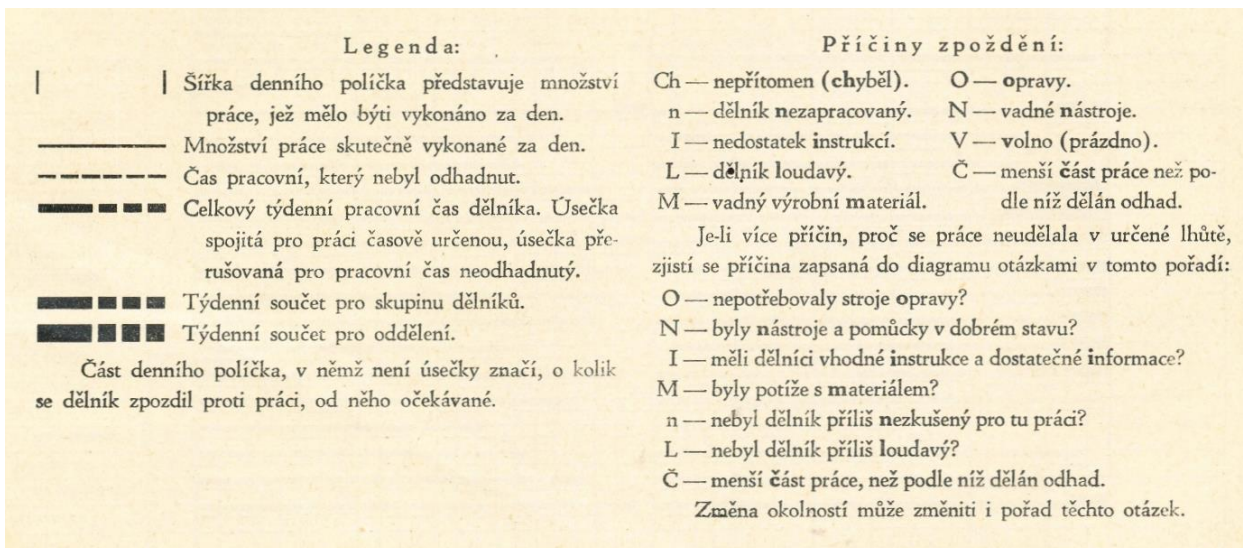
Obr. 1 Henry Laurence Gantt [3].

Ze začátku se Ganttovy diagramy vytvářely ručně - tuží, což bylo velice pracné a při každé změně v diagramu se musely graf překreslit, popřípadě nakreslit znovu. To značně omezovalo jejich dynamické využívání, protože změny v diagramu jsou jedním z hlavních vítaných rysů.

Jeden z prvních Ganttových diagramů, který byl ve své době přeložen do češtiny, je znázorněn na Obr. 2. Tento konkrétní diagram byl vytvořen přibližně už před 90 lety. Diagram zobrazuje jmenovitě vykonanou práci jednotlivých pracovníků v dané firmě. S pomocí přiloženého popisu na Obr. 3 nám diagram podává velké množství informací o průběhu práce v poměrně na malém kousku papíru, což patří mezi jednu z hlavních předností Ganttových diagramů.



Obr. 2 Jeden z prvních Ganttových diagramů [2]



Obr. 3 Popis k diagramu na Obr. 2 [2]

V dnešní době ovšem s příchodem éry osobních počítačů (1980) tyto problémy odpadají. Existuje celá škála sofistikovaných softwarů, které dokáží vykreslovat Ganttovy diagramy a měnit údaje za pochodu.

2.3 POUŽITÍ

Použití Ganttových diagramů je skoro neomezené. Mohou se používat jak v malých podnicích (se dvěma až čtyřmi zaměstnanci), tak i ve velkých podnicích čítajících stovky zaměstnanců. Dokáží popsat jednoduše diagramem chod podniku v určeném časovém horizontu. Slouží k vytvoření plánu výroby tak, aby výsledný plán byl uskutečnitelný v co nejkratší době, tak se i stará o to, jestli práce pokračuje podle rozvrhu, popřípadě pokud ne, tak nalézt přímo příčiny. Umožňují velmi přesně predikovat to, co se také stane a tím lépe odstranit překážky. Nečinnost lidí a strojů bývá většinou největší problém při plánování činnosti či řízení výroby jako takové, tento problém se dá ovšem odstranit pomocí Ganttových diagramů, které tyto nedostatky umí pojmenovat a říct kde se konkrétně nacházejí. V případě tohoto druhu nečinnosti se dají po jejich prvním náznaku navrhnout opatření, která tuto časovou ztrátu dokáží dohnat, jak už například zvýšení počtu pracovníků, prodloužení směny a podobně.

Mezi významné využití patřilo pak například řízení lodní dopravy za 1. světové války. Dále pak často zmiňovaná stavba, kde byly použity Ganttovy diagramy je stavba Hooverovy přehrady (1931-1936), kde sloužila k rozdělování práce pracovníkům a řízení zásob na skladě tak, aby nikdy nemohly pozdržet práci včasným objednáním. Dalším větším projektem, kde byly použity Ganttovy diagramy je plánování mezistátní dálniční sítě napříč Amerikou (1956). Byly ovšem použity i v situacích, kde by to třeba někdo ani nečekal. Příkladem toho je například pozvání samotného Henry Gantta do Evropy polským ministrem financí, za účelem posílení polské měny, což se opravdu v té době podařilo. [2]

Ganttovy diagramy se používají více než 100 let, což je samo o sobě dobrým ukazatel, že patří stále k tomu nejlepšímu, jak zpracovávat plánované činnosti. Doba se ovšem vyvíjí a s ní i Ganttovy diagramy. Mezi tyto zásadnější modifikace v posledních desetiletích patří propojení Ganttových diagramů s optimalizačními algoritmy. Mezi nejčastější metody optimalizace patří techniky CPM a PERT patřící do odvětví síťové analýzy.

2.3.1 Metoda CPM

CPM – neboli metoda kritické cesty byla vymyšlena koncem 50. let minulého století. Jde o deterministický matematický model, který propočítává celkové trvání v rámci celého projektu [4]. Princip metody tkví v tom, že k diagramům přistupuje jako k ohodnocenému síťovému grafu, ve kterém se řeší kritická cesta – což je nejdelší cesta, nemajíc časové rezervy. Tato kritická cesta určuje celkové trvání projektu ve formě nejkratší možné realizace v rámci daného projektu [5]. Jde v podstatě o vytvoření cesty mezi počátečním uzlem – počátečním procesem a koncovým uzlem – koncovým procesem.

2.3.2 Metoda PERT

Tato metoda je zobecněním CPM metody. Oproti CPM pracuje se stochastickým matematickým přístupem, který se používá u složitějších činností s lehčím plánováním, které nemají přesně ohodnocení trvání jednotlivých činností. Pracuje se zde místo přesných hodnot trvání s pravděpodobnostním rozdělením. K tomuto účelu se zde používá beta rozdělení. Tato metoda je sice výpočtům jednotlivých pravděpodobností složitější, ale na druhou stranu i přesnější [6]. Tato problematika síťové analýzy je poměrně obsáhlá jako taková a není dalším obsahem diplomové práce.

3 ANALÝZA A NÁVRH PROGRAMOVÉHO VYBAVENÍ

Jak už bylo uvedeno, hlavní účel Ganttových diagramů je zefektivnění procesů, které Ganttovy diagramy graficky popisují. K tomuto popisu slouží několik hlavních funkcí a částí, které jsou charakteristické pro Ganttovy diagramy a které budou v této kapitole dále rozebrány. Mezi nejdůležitější tyto části patří samotná plocha pro diagramy, diagramy ve formě procesní činnosti, prioritizace jednotlivých činností, souvislost - neboli návaznost mezi jednotlivými procesy a možnost celé manipulace ve formě fronty, která eviduje dané pořadí.

Při návrhu programového vybavení byl použit programovací jazyk C# pro jeho vhodné použití v oblasti OOP za využití programové verze Microsoft Visual C# 2010 Express, která je zdarma běžně dostupná.

Při návrhu vykreslování jednotlivých částí byla použita technika WPF, která je nástupcem klasické WF technologie, která v oblasti především vykreslování má značná omezení. Je to z toho důvodu, že WPF používá při vykreslování Direct3D, což je speciální rozhraní API, takže jsou aplikace méně náročné na procesor. Oproti tomu WF provádí veškeré vykreslení na rozhraní GDI a dost tím zatěžuje procesor. Dále je výhoda WPF oproti WF také v tom, že WF má pozici všech kontrol statickou, kdežto WPF nabízí dynamickou manipulaci, závislou například na velikosti samotného okna, v kterém se nachází. WPF je sice z programového hlediska náročnější, protože vyžaduje i znalost jazyka XAML, oproti tomu nabízí ale mnohem větší volnost ve vytváření, která je v podstatě neomezená. V současné době, kdy se rozrůstá trh s mobilními aplikacemi, je i v těchto situacích vhodné se zaměřovat na WPF technologii, pro její snadnou realizaci přizpůsobení se libovolné velikosti displeje. [7].

3.1 Plocha diagramu

Plánovací plocha diagramu slouží k znázorňování pracovních činností v závislosti na čase. Zmíněný čas je zde znázorněn pomocí časové osy, která se pro přehlednost nejčastěji umísťuje nad samotnými pracovními činnostmi a má horizontální směr pro co nejefektivnější využití plochy. Formát časové osy by měl být rozdělen na stejně velké časové úseky - buňky a volen s přihlédnutím na délky trvání jednotlivých procesních činností tak, aby výsledný diagram byl co nejpřesnější a nejprehlednější i pro čtení samotným okem. Pomyslnou jednotkou na časové ose mohou být například hodiny, dny, týdny, měsíce, roky a podobně. Samozřejmě je vhodnější účelně zobrazovat dostatečný počet časových jednotek v různých kombinacích současně. Pokud by například časová osa podávala informace o pouhých hodinách, tak by uživatel mohl lehce ztratit ponětí o celkovém časovém kontextu - v jakém dni se činnost nachází a podobně. Na Obr. 4 je znázorněna jedna z možností pracovní plochy, která poskytuje veškeré časové informace, které jsou pro uživatele jakkoliv potřebné.

10. března 2014 (pondělí)							
0:00	3:00	6:00	9:00	12:00	15:00	18:00	21:00

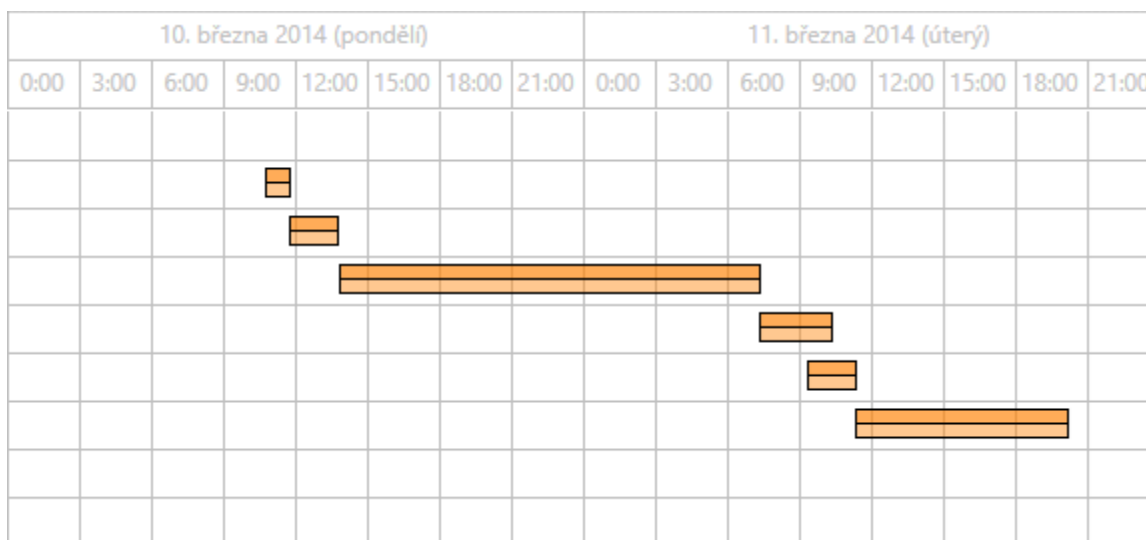
Obr. 4 Jedna z možných pracovních ploch.

Programově v prostředí C# se dá této plochy dosáhnout pomocí spojení dvou kontrolkek Canvas, každá umístěna ve svém ScrollViewer, díky kterému se dá vhodně nastavit sdílené posouvání obou kontrolkek současně. Horní Canvas obsahuje pouze časovou osu úhledně rozdělenou mřížkou, kde každá takto vznikla buňka, podává časové informace o celém svém obsahu. Druhý Canvas (dolní) obsahuje ze začátku pouze samostatnou mřížku, ale později se do něj budou vykreslovat průběhy všech pracovních procesů. Jsou použity dvě plochy, na vykreslování, a ne jedna z toho důvodu, že pokud by uživatel chtěl posouvat (scrollovat) seznam pracovních procesů (spodní Canvas), tak by okamžitě ztrácel pojem o tom, v jaké časovém pásu se zrovna nachází. Z tohoto důvodu je nejpraktičtější nastavit možné scrollování horního Canvasu pouze na horizontální směr a spodního jak na horizontální (sdílený s horní plochou), tak i vertikální, pro možnost posouvat se mezi všemi pracovními činnostmi. Vykreslování mřížek a textů, zobrazujících čas, je možné realizovat pomocí cyklu `for` za použití několika parametrů, které jsou získané z dat pracovních činností.

Mezi tři nejdůležitější parametry při vykreslování patří délka osy X, která nám například u mřížky říká, po drobné korekci na časovou jednotku osy, kolik bude potřeba vertikálních čar pro vykreslení mřížky. Tato velikost délky osy X se dá jednoduše získat výpočtem z rozdílu posledního ukončeného procesu a prvního započatého procesu – převedeno na celkový počet hodin. V tomto případě je převod na hodiny z toho důvodu, že všechny používané měřítka časové osy se dají vyjádřit nejsnadněji v hodinách. Pro získání skutečné velikosti délky osy se musí ještě vypočítaná délka osy vynásobit šířkou jedné buňky, aby byl získán skutečný rozměr. Ve zmíněné korekci časové osy X jde o to, že chceme-li například vykreslovat časovou osu po třech hodinách, jak tomu bylo na Obr 5, tak v tu chvíli pomyslná délka osy X znázorňuje pouze počet hodin od nejdříve započaté činnosti po nejpozději ukončenou činnost to celé vynásobené šířkou jedné buňky. To není úplně přesné, protože v tom případě by šířka jedné buňky byla jedna hodina a nikoli tři. Pro převod tedy postačí tuto délku osy X vydělit onou jednou časové osy – tedy třemi. Pokud by jedna buňka měla znázorňovat například jeden den, tak by stačilo délku osy X vydělit číslem 24 (počet hodin jednoho dne).

Druhým parametrem je délka osy Y, která v porovnání s osou X místo počtu hodin říká jen počet procesních činností – osa je také vynásobená velikostí jedné buňky – tentokrát

V dnešní době se ovšem začínají používat hned dva - z toho důvodu, že jeden (horní) obdélník znázorňuje plán činnosti a druhý (spodní) obdélník znázorňuje skutečný stav činnosti, jak je patrné na Obr. 7. Toto rozdělení má svůj důvod. Může totiž nastat například situace, kdy začátek činnosti nastane dříve - popřípadě později, než bylo původně plánováno. V tom případě by měl mít uživatel stálou možnost vidět - jak si proces ve skutečnosti vede a současně jak by si proces měl vést podle plánu. Na Obr. 7 je znázorněn způsob výsledného vykreslení. V tomto konkrétním případě je činnost procesu ve skutečnosti souměrná s plánem, což je optimální stav – vše se děje tak, jak bylo naplánováno. Pro vykreslení skutečnosti není špatné - volil pouze slabší odstín barvy vykreslující plán, aby bylo zřejmé, že daná skutečnost k plánu patří. Nastavení barvy plánu je čistě v režii uživatele a může si barvy jednotlivých procesů libovolně měnit pro svojí přehlednost.



Obr. 7 Jednotlivé průběhy činností.

Pracovní činnost nám říká, jak dlouho bude daný proces trvat - přesněji jaký je časový rozdíl mezi zahájením a ukončením procesu.

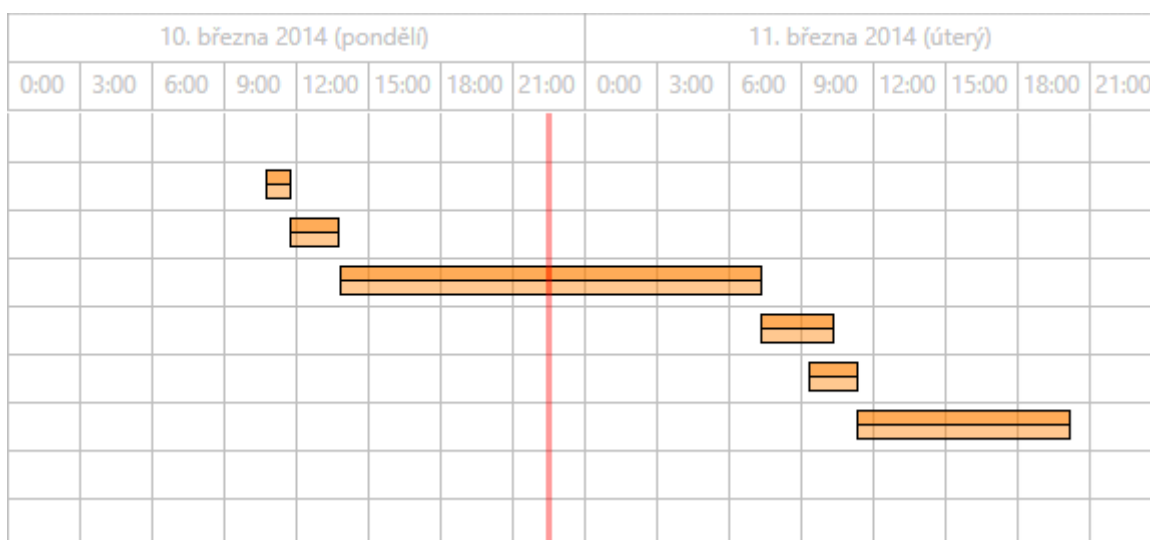
Z programového hlediska, pokud známe nejmladší položku (zmíněnou u plochy diagramu), tak není problém zjistit její vzdálenost od daného začátku činnosti, jak je předvedeno na Obr. 8. Následně samotné tzv. délku trvání procesu se vypočítá rozdílem mezi koncem a začátkem konkrétního procesu. Tento rozdíl je počítán v sekundách pro maximální přesnost (s přesností na sekundy) a následně je opět zpět vynásoben na hodiny.



Obr. 8 Zjištění vzdálenosti.

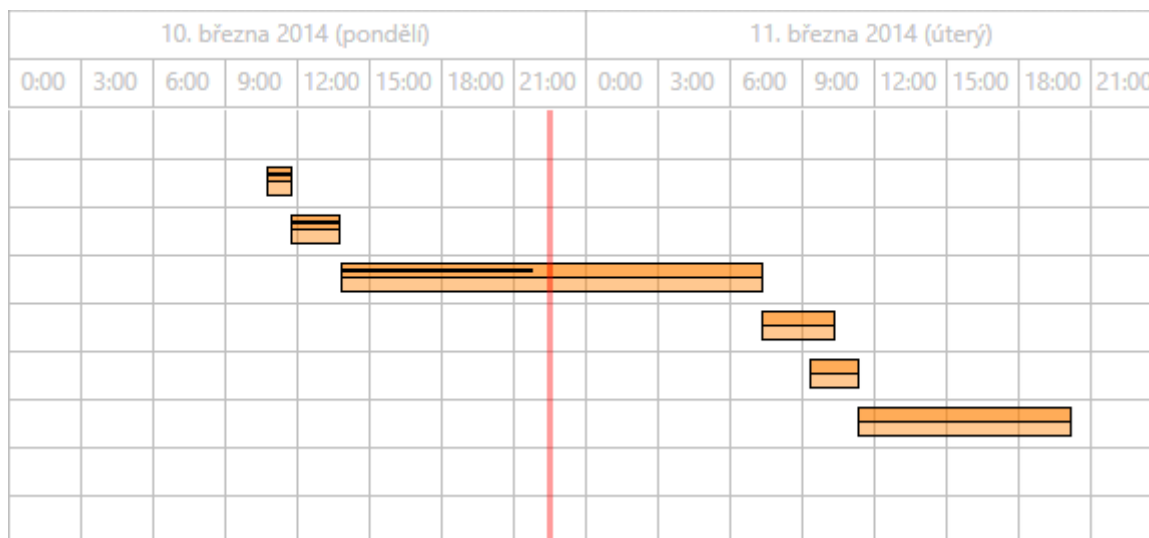
Vykreslování činností se dá realizovat pomocí cyklu for, kdy jsou všechny jednotlivé položky rozděleny podle zakázek. Každá zakázka tedy obsahuje pouze svoje procesy. Následně jsou vykresleny jednotlivé zakázky, přičemž se s každou vykreslenou činností posouvá Y-nová souřadnice o jeden řádek, aby každý řádek neobsahoval více jak jednu činnost.

Je také vhodné zobrazovat v diagramu současný čas, pro lepší orientaci. To se dá realizovat podobně jako délky trvání procesů. S tím rozdílem, že začátek konkrétní činnosti bude nastaven na současné datum (včetně času). Délka trvání procesu bude zde představovat pouhou šířku dneška. Oproti běžné činnosti bude dále obsahovat proměnnou výšku, která se bude rovnat počtu položek vynásobených velikostí mřížky Y, Obr. 9.



Obr. 9 Zobrazení dneška v diagramu.

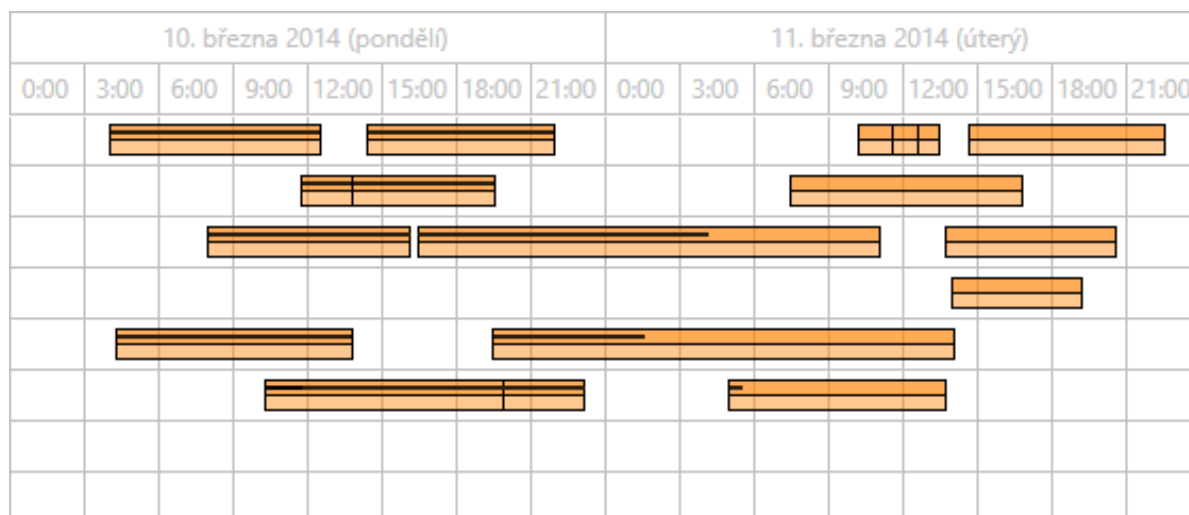
Další potřebnou věcí při vykreslování činností je také to, z jaké části je proces již hotov. Při vykreslování se dá například mluvit o tom, z kolika procent je již činnost hotová. Vykreslení může probíhat obdobně jako při vykreslení činnosti s tím, že hotové množství se nejčastěji vykresluje do celkové délky trvání procesu (Obr. 10). Tato informace nám v porovnání se současností (dneškem) může podat přesný stav činnosti. Toto zobrazení nám může už názorně ukázat poměr mezi množstvím požadované práce, vykonané práce a předpokládanou dobou, ve které by měla být činnost ukončena.



Obr. 10 Zobrazení hotové práce.

V dřívějších dobách se množství hotové činnosti zapisovalo pomocí dvou čar nad sebou, což mohlo vést k určité nepřehlednosti.

Zobrazování více obdélníkových diagramů v jednom řádku se často používá v situacích, kdy dochází k zobrazování Ganttových diagramů pracovišť (Obr. 11). V této situaci každý řádek znázorňuje jedno pracoviště. Prázdné mezery mezi jednotlivými obdélníky v jednom řádku znázorňují čas, kdy se na konkrétním pracovišti nepracuje. Programově se dá toto vykreslení realizovat velmi podobně jako u Ganttova diagramu. S tím, že se všechny procesy rozdělí podle pracovišť a následně dojde k vykreslení vždy celého pracoviště a následné posunutí na další řádek.



Obr. 11 Vykreslení Ganttových diagramů pracovišť.

3.3 Priorita

Priorita nás informuje o tom, jak je daný proces důležitý v porovnání s ostatními. Zvýšení priority nějakému procesu obvykle znamená snížení priority jiného procesu. Číselně se to dá vyjádřit tak, že u dvou procesů jednomu zvýšíme práci na něm na 120% a druhému ji snížíme na 80%. Samozřejmě to není nejlepší způsob řešení, protože tímto způsobem může dojít i k ovlivnění ostatních procesů. Nejčastěji se zvýšení priority řeší více přesčasy, aby nedošlo k ovlivňování ostatních procesů.

Pro programovou realizaci by byl potřeba nějaký algoritmus, který prováděl reorganizaci jednotlivých činností podle priority.

3.4 Souvislost

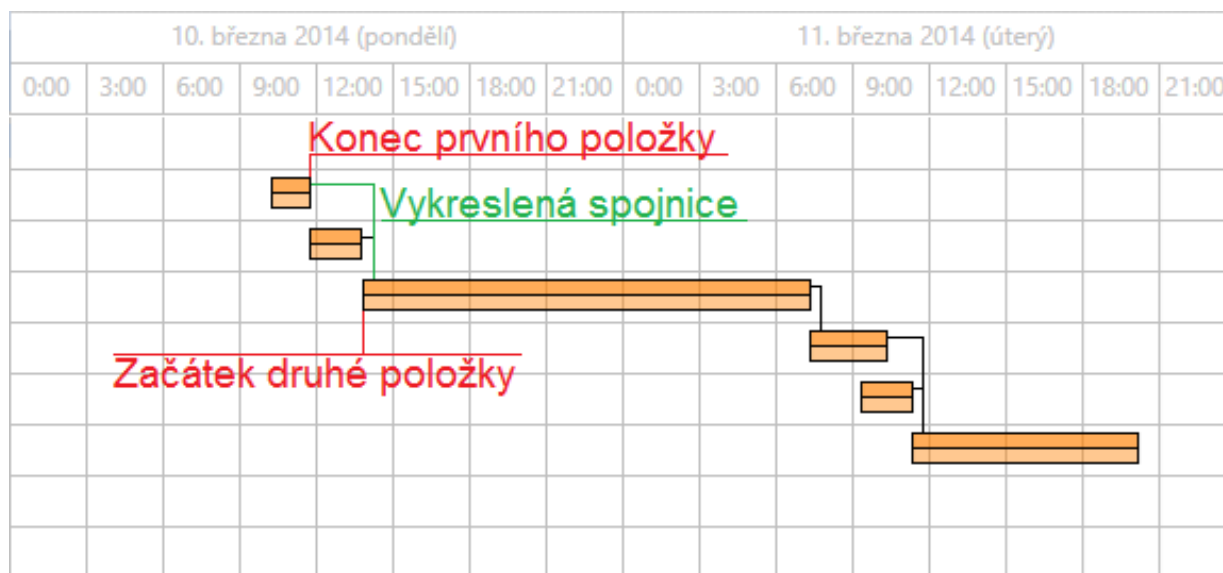
Souvislost nám říká, jestli na zahájení některého procesu musí dojít k ukončení jiného procesu. Nebo stejně tak může znázorňovat, v jakém pořadí se jednotlivé procesní činnosti mají provádět. Příkladem může být například výroba nějaké součástky, kdy součástka putuje mezi jednotlivými pracovišti. Součástka by nemohla být na dvou pracovištích současně.

Souvislosti mezi jednotlivými pracovními činnostmi je všeobecně jeden ze základních aspektů, které Ganttovy diagramy odlišují od harmonogramů. Díky těmto souvislostem se diagram stává dynamický nástroj, schopen jednoznačně vyjádřit v jakém pořadí se pracovní činnosti mají realizovat.

Na Obr. 12 je znázorněn jeden ze způsobů, jak znázorňovat souvislosti mezi jednotlivými procesními činnostmi.

Programově se dá vytvoření spojitostí realizovat například formou Listu pro spojitosti, jehož každá položka bude mít informace, z které položky vede cesta do které. Každá tato položka v tomto Listu spojitosti bude reprezentována pouhými dvěma čísly, znázorňující čísla procesních činností, mezi kterými je spojení vytvořeno. Toto číslo procesní činnosti je ve skutečnosti číslo, pod kterými je uložena daná činnost v Listu pracovních činností.

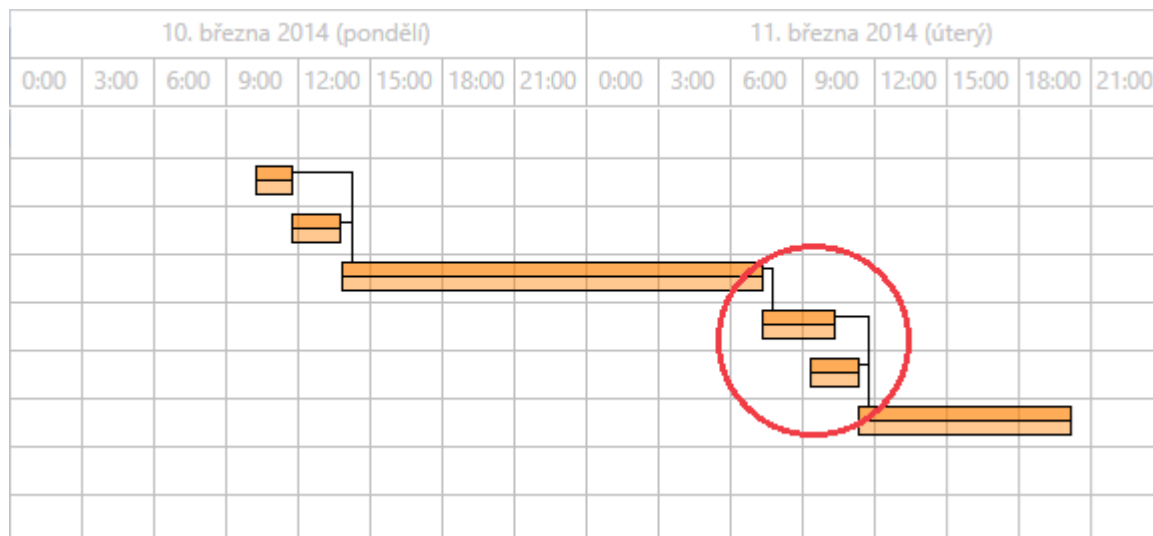
Při vykreslení se pomocí těchto dvou čísel procesních činností dají dohledat potřebná data, určující X-ové souřadnice, mezi kterými se spojení realizuje. Pomocí prvního čísla se dá získat datum začátku spojení, kde ono číslo udává konec položky pod tímto číslem. Druhé číslo pak udává datum konce spojnice, tedy začátek procesní činnosti pod tímto číslem. Na Obr. je znázorněn příklad, kde je vykreslená spojnice z první procesní činnosti do třetí. Při znalosti začátku a konce tzv. spojnice je potřeba už jen určit směr spojení těchto dvou bodů. Protože ve výrobě většinou při skončení jednoho procesu okamžitě začíná jiný proces, tak je nejlepší volit směr vykreslené spojnice do tzv. pravého úhlu (viz. Obr. 14). Při tomto vykreslení zaujímá vykreslená spojnice nejméně plochy a působí velmi přehledně.



Obr. 14 Znázornění způsobu oddělování zakázek.

Při vytváření samotných spojníc je potřeba vycházet z faktu, že nesmí nastat situace, kdyby pomyslná první položka skončila v daném datum a následující položka by začala ještě před koncem oné první položky. Tato situace je znázorněna na Obr. 15. V tomto případě je potřeba rozlišovat co se spojuje s čím. Samotné propojování dvou činností v řadě za sebou není příliš správné. Jelikož je potřeba aby si program byl schopen sám rozpoznat co má s čím spojit, je potřeba aby obsahoval nějakou vnitřní logiku.

Programově se dá tato vnitřní logika naprogramovat tak, aby se spojnice nevytvářely mezi každou následující položkou, ale pouze mezi položkami, kde první položka končí dříve (nebo ve stejnou dobu) jako začíná druhá položka. Z první položky nalezení druhé položky, lépe řečeno nalezení „první“ následující položky, jejichž začátek je až po konci první za položky. Toto hledání se dá vhodně realizovat pomocí rekurzivního volání. Samozřejmě pokud v dané zakázce existuje pouze jedna procesní činnost, tak k vytvoření spojnice vůbec nedojde.



Obr. 15 Vytváření správných spojníc.

3.5 Fronta

Fronta slouží k odstranění problému, kdy přesně nevíme, na čem by se mělo v konkrétní den pracovat, kde můžeme s procesem počkat a kdy je to už neodkladné. Fronta zde nahrazuje jakousi lidskou entitu, která by jinak musela velmi subjektivně rozhodovat v jakém pořadí pracovat, aby byla práce co nejefektivnější. Navíc zmíněná entita nemusí mít dostatečné vědomosti a znalosti o všech pracovištích. Pokud bychom měli víc takovýchto pracovníků, kde by se každý věnoval svému pracovišti, kterému dobře rozumí, tak by často docházelo k neshodám mezi sebou, co se rozvrhu práce týče a mohlo by to vést protichůdným způsobům řešení. Kdyby taková osoba přece jen existovala, tak by byla až příliš nenahraditelná a její odchod by způsobil nedorozumění problémy a zastavil veškeré práce. Tím, že se nám o toto rozhodování stará fronta odpadá veškeré dohadování, jestli je ona osoba dostatečně kvalifikovaná na rozhodování nebo ne, navíc fronta nám nikam nemůže odejít. Do fronty se nám vkládají jednotlivé procesy v takovém pořadí, aby došlo k co největšímu počtu dokončených procesů v co nejefektivnějším čase. Zaměstnanci díky tomu mohou vědět v jakém pořadí pracovat. Pokud jde například o nějaké výrobní podniky, tak nám to může říkat v jaké pořadí obsluhovat jednotlivé stroje, aby nedocházelo k časové prodlevě a stroje byly maximálně využity. V podnicích to může vypadat tak, že vedoucí zná položky ve frontě na daných pracovištích na několik dní dopředu.

Programově by se dala fronta realizovat nějakým vhodným algoritmem schopným prohledávání stavového prostoru.

4 IMPLEMENTACE KNIHOVNY

Naimplementovaná knihovna slouží k samotnému vykreslení Ganttových diagramů. Obsahuje tedy v sobě několik metod pro vykreslení. Jsou zde tři stěžejní metody, patří sem metoda na vykreslení mřížky, zakázky a pracovišť. Tyto metody v sobě dále zahrnují další metody, které jsou také zakomponovány v knihovně. Dále knihovna obsahuje kromě třídy na vykreslení další tři třídy pro práci s objekty. Jedná se o objekt plocha – nesoucí informace na jakou plochu se bude kreslení provádět, jaká je jednotka ve směru X a Y na kreslicí ploše a aktuální hodnotu Y – udávající aktuální místo, kam se v daný okamžik má kreslit. Dalším objektem je zakázka, která sdružuje všechny procesy v rámci dané zakázky, dále si propočítá z procesů dané zakázky její přesný začátek a konec pomocí nalezení nejdříve započaté činnosti a nejpозději skončené činnosti. Dalším objektem je pracoviště, které funguje naprosto stejně jako zakázka s tím rozdílem, že nesdružuje procesy v rámci zakázky, ale v rámci pracovišť. Posledním objektem je položka. Jednotlivé položky znázorňují konkrétní procesy. Každá takováto položka si nese velké množství informací o sobě, jako je název zakázky, název výrobku, číslo dané operace v rámci zakázky, její plánovaný konec a začátek, jaká část položky je jich hotova, prioritu položky, počet kusů, její skutečný začátek a konec, dále pak barvu, kterou má být daná pracovní činnost vykreslena a informace o tom, jestli se má daná položka zobrazovat.

4.1 Vykreslení mřížky

Tato metoda, jak název napovídá, slouží k vykreslení mřížky. V knihovně jsou hned dvě metody s tímto názvem, ovšem s jinými typy parametrů. Jedna tato mřížka požaduje pět parametrů, jsou to počet buněk mřížky na šířku (delkaosyX), následně pak velikost jedné buňky na šířku (velikostMrizkyX) a na výšku (velikostMrizkyY), dále pak parametr určující kam se má vykreslení provést – jedná se o kontrolku cava, posledním parametrem je pak datum nejmladší položky, určující datum na začátku mřížky ve směru osy X – v tomto případě se jedná o údaj datového formátu.

Možná může působit poměrně matoucím dojmem proč je u mřížky potřeba datum nejmladší položky. Je to z toho důvodu, že nejde v tomto případě o čistou mřížku jako takovou. Při vykreslování mřížky dochází také k vykreslování textů do jednotlivých buňkách (Obr. 16). Tyto texty jsou přesně vycentrované do buňky, v které se nahází. K tomuto účelu vycentrování jsou v knihovně použity metody pro získání rozměrové výšky a šířky textu, takže je následně možné při znalosti těchto dvou parametrů při vycentrování přistupovat jako k obdélníku, který pomyslně ohraničuje text.

10. března 2014 (pondělí)							
0:00	3:00	6:00	9:00	12:00	15:00	18:00	21:00

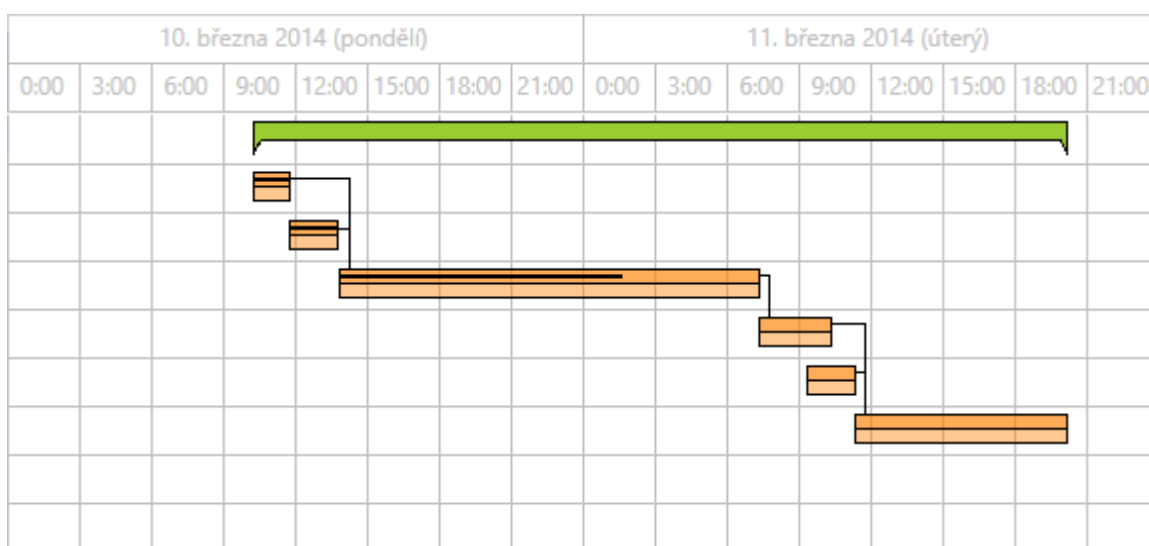
Obr. 16 Vykreslené mřížky s textem.

Při vykreslení této mřížky hraje důležitou roli šířka jedné buňky. Podle její velikosti se rozhoduje, jak bude tato mřížka vypadat. Pokud by byla buňka již příliš malá, tak dojde ke spojení s dalšími buňkami. Stejně tak pokud by byla buňka již příliš velká, tak dojde k jejímu rozdělení ve dvě buňky. Toto rozhodování se děje uvnitř samotné metody a podle rozhodnutí se zavolá zvolená metoda na vykreslení. Existují čtyři různé rozložení této mřížky s textem, přičemž existuje základní nastavení, z něhož ostatní možnosti vykreslení pouze dědí a přepisují si pouze změněné parametry. Pomocí dědičnosti se zde dá poměrně dobře zamezit zbytečné redundanci zdrojového kódu.

Druhá mřížka obsahuje také pět parametrů, má ale ovšem místo parametru znázorňující nejmladší položku, informaci o počtu potřebných řádků mřížky (*delkaosyY*). Ve výsledku jde tedy o klasickou mřížku. I tato mřížka ovšem obsahuje automatické přizpůsobení podle šířky jedné buňky, založenou na úplně stejném principu jako prvně zmíněná mřížka.

4.2 Vykreslení zakázky

Jelikož není problém si jednotlivé procesní činnosti roztrždit do skupin podle jména zakázky, tak je velice vhodné si vykreslovat vždy celou zakázku jako celek. Jako vstupními parametry této metody jsou list zakázek – která obsahuje celý seznam se všemi zakázkami, dalším parametrem je pak již zmíněná plocha definující kam se bude vykreslování provádět a informaci na jakém řádku se vykreslování zrovna nachází, ve smyslu jak moc od vrchu se má zrovna kreslit, třetím a posledním parametrem je datum nejmladší položky – která se stará o informace ve směru osy X, ve smyslu jak moc od levého kraje se má kreslit. V této metodě se nejdříve provede seřazení jednotlivých zakázek podle jejich začátku – je to z toho důvodu, aby při vykreslení šly zakázky od vrchu dolů chronologicky. Každá takováto zakázka mě vlastnost určující její výsledný začátek a konec výpočtem z jednotlivých položek. Po seřazení zakázek dochází k samotnému vykreslení. Funguje to na způsob, kdy se projedou cyklem všechny zakázky, přičemž u každé zakázky se provede metoda na vykreslení trvání celé zakázky (*VykresleniTrvaniZakazky*). Tato metoda vychází z faktu znalosti začátku a konce celé zakázky. Na Obr. 17 se jedná o ten zelený obdélník, který je na krajích ostře zkosen.

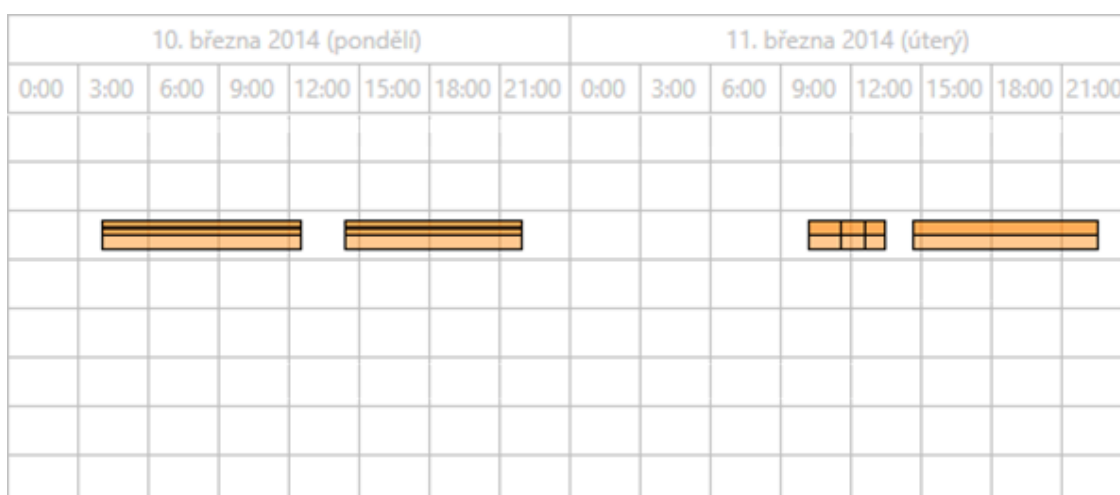


Obr. 17 Vykreslení zakázky.

Po vykreslení trvání celé zakázky dojde k posunutí na další řádek, kde už se začíná provádět cyklus na vykreslení všech procesních činností v rámci dané zakázky. Jde přesněji o vykreslování jejich trvání (VykresleníTrvani). Tato metoda je zde volána hned dvakrát po sobě. Z toho důvodu, že jeden obdélník znázorňuje plán procesní činnosti, na Obr. ten tmavě oranžový, a druhý obdélník skutečný stav procesní činnosti, na Obr. ten světle oranžový. Tato metoda používá především na vykreslení údaje z dané činnosti jako je začátek, konec barva a informace o samotné ploše. Následně dojde k zavolání metody pro vykreslování již hotové části (VykresliHotovo). V principu jde v podstatě o stejný obdélník jako při vykreslení jedné procesní činnosti. S tím rozdílem, že se vzdálenost, jako proces, má vynásobit procentem hotové části. Dále zde dojde už jen k zúžení a zaměření Y-nové hodnoty na vnitřek procesní činnosti znázorňující plánované trvání. Všechny tyto metody vykreslující obdélník využívají principu dědičnosti, přičemž si jen ze základního obdélníkového diagramu upravují parametrické hodnoty. Dále se ve vykreslení zakázky provede ještě vytvoření spojitosti mezi jednotlivými procesy (VykresleníSpojnice). Tato metoda má v sobě určitou vnitřní logiku, která pomocí rekurzivního volání hledá, co se dá s čím spojit. Vychází totiž z faktu, že nelze vytvořit spojení mezi procesní činností, jejíž začátek začal dřív než předchozí činnosti konec. V této metodě tedy dochází k hledání validního spojení. Tato metoda má v sobě další logiku takovou, že zabráňuje spojovacím čarům v přechodu přes obdélníkový diagram. Po vykreslení těchto objektů dojde k posunutí na další řádek. Toto se děje tak dlouho dokud nejsou všechny procesy všech zakázek vykresleny.

4.3 Vykreslení pracoviště

Vykreslení pracoviště je založeno na úplně stejném principu. S tím rozdílem, že vstupním parametrem je místo listu zakázek, list pracoviště. Samotný vykreslovací princip je v podstatě identický s vykreslováním zakázek. V tomto případě zde ovšem nedochází k žádnému vykreslování spojitostí mezi jednotlivým zakázkami a po vykreslení procesu nedochází k posunutí na další řádek. Posunutí na další řádek se provede až ve chvíli vykreslení všech činností daného pracoviště.



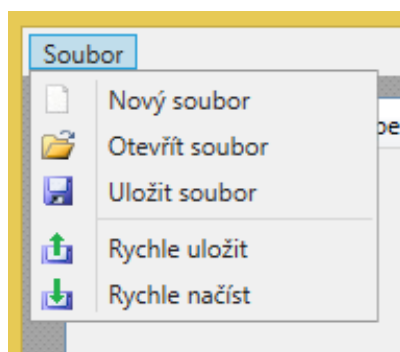
Obr. 18 Vykreslení procesů v rámci jednoho pracoviště

5 REALIZACE SOFTWARE

Vytvořený software využívá naimplementovanou DLL knihovnu a skládá se z několika hlavních částí, které jsou následně vysvětleny. Při tvorbě byl kladen hlavně důraz na univerzálnost použití a požadavky, které jsou při řízení výroby nezbytností. Jeho univerzálnost je poměrně široká i co se týče využití, takže je možné ho použít i na jiná plánovaná odvětví, i když byl zaměřen na plánování výroby.

5.1 Hlavní menu

Pod tímto názvem se skrývá menu zobrazující možné operace se samotným souborem. Toto menu je umístěno v levém horním rohu a obsahuje několik možností, jak je patrné z Obr. 19. Jak je patrné z obrázku, tak menu označené jako Soubor (protože se jedná o práci čistě se souborem) obsahuje základní manipulace se souborem jako je vytváření úplně nového souboru od znova, otevírání a ukládání souboru a mimo to i rychlé volání pro uložení, popřípadě načtení dat.



Obr. 19 Hlavního menu programu.

5.1.1 Nový soubor

Nový soubor slouží k tomu, že si smaže současné – zobrazené procesy. V podstatě nebude obsahovat žádná nahraná nebo jakkoliv jinak vytvořená data. Tato možnost umožňuje začít čistě nanovo – jakoby uživatel znovu zapnul program a měl možnost vše naplánovat od znova.

5.1.2 Otevření souboru

Tato možnost je asi nejvíce obsáhlá ze všech funkcí, které software nabízí a umožňuje uživateli ohromné množství, co se týče způsobu manipulace s daty. Program hned nabízí širokou škálu typů souboru, které dokáže zpracovat. Mezi databázovými soubory to jsou pro Excel .xls soubory, pro Access .accdb a .mdb soubory, mezi textovými soubory jsou to pak soubory s koncovkou .txt a .tbl. U datových souborů je zvolen jiná přístup než u textových.

5.1.2.1 Textový soubor

U textového souboru se zobrazí okno pro nahrávání procesních dat (Obr. 20). Je zde možnost eventuálně změnit soubor, tedy přesněji i samotného přístup k němu za databázový popřípadě obráceně. Ve vedlejší kontrolce pro text se zobrazuje zvolená cesta k souboru, aby

uživatel věděl, co přesně chce otevřít. Pod tímto jakoby prvním řádkem je potřeba zvolit, jakým způsobem jsou v tomto textovém souboru oddělena data. Může se jedna například o tabulátor, středník, čárku nebo mezeru, které jsou v rychlé volbě. Je zde ovšem možnost použít jiný oddělovací znak, v tom případě se zobrazí nová kolonka, kam uživatel může napsat libovolný znak - popřípadě text, kterým jsou data oddělena. Pokud takto navolí je potřeba data ještě správně identifikovat. V textových souborech totiž neexistují tabulky, které by data podle něčeho rozdělávala. Jde spíše o jakousi směs dat oddělených jedním znakem odrážkováný. Při identifikaci byl kladen důraz na jednoduchou manipulaci, když uživatel pouze vybírá data. Je zde sedm nejdůležitějších údajů, která každá činnost musí mít. Jsou to zakázka – tedy přesněji její název, výrobek, číslo operace, pracoviště, začátek, konec a počet kusů. Jak je patrné na Obr. tyto údaje se musí vybrat pomocí tzv. kombinovaných seznamů, které jsou pod každou položkou a jejíž hodnotu budou následně reprezentovat. Při výběru oddělovacího znaku jsou do těchto kombinovaných seznamů zapsána data, která se v daném souboru nachází na prvním řádku, přičemž se bere v potaz oddělovací znak mezi jednotlivými údaji v souboru.

Obr. 20 Nahrávání procesních dat z textového souboru

Uživatel tímto jednoduchým způsobem může velmi efektivně identifikovat data v souboru. Pokud například u zakázky vybere ze seznamu druhou položku, program to pochopí tak, že u každého řádku je na druhém místě název zakázky. Možnost odeslání dat se uživateli nezpřístupní do té doby, než identifikuje takto všechna data. Další poměrně šikovnou funkcí je zde možnost použít použitou konfiguraci. Tato funkce funguje tak, že pokud uživatel už jednou odešle nějaká data, tak program si zapamatuje zvolenou konfiguraci. Tedy přesněji jak se daný soubor jmenoval, dnešní datum, jaký byl použit oddělovací znak a také především informaci co se na kolikátém řádku nachází. Pokud následně bude potřeba otevřít například soubor stejného typu – tedy přesněji jeho datového rozložení, tak bude stačit zde vybrat danou konfiguraci podle názvu. K názvu je zde přidáno i datum, pokud by byly vytvořeny dvě stejné konfigurace v různých dnech. Po použití konfigurace a odeslání dat se nová konfigurace nevytváří, protože tam už v podstatě taková existuje. Pokud ovšem uživatel vše navolí ručně a odešle data, tak vznikne nová konfigurace. Zapamatování konfigurace probíhá tak, že se na pozadí tato konfigurace ukládá do složky planovaniVyroby, daný soubor má pak název konfigurace.csv. Každá konfigurace v tomto souboru zaujímá jeden řádek a její první položka udává označení konfigurace – tedy její název a datum vytvoření konfigurace. Tyto označení konfigurace jsou pak uživateli nabídnuta. Pokud by uživatel chtěl určitou konfiguraci, která by nesesedla na vybraný soubor, který chce otevřít, tak by byl varován, že dané soubory si nejsou podobné. Pokud by uživatel zadal některá data špatně – co se především datového typu týče, tak nebude schopen data odeslat. Jde především o situace, kdy by zadal místo potřebného čísla například u počtu kusu text a podobně.

5.1.2.2 Databázový soubor

Základní struktura okna je velmi podobná jako u textového, je zde ovšem několik nových možností (Obr. 21). Hned první změnou je to, že se nevybírám oddělovací znak, ale název tabulky daného souboru, která se vybírá ze seznamu. Do tohoto seznamu jsou nahrány všechny tabulky daného souboru, takže stačí jen jednoduše vybrat, v které jsou data.

Obr. 21 Nahrávání procesních dat z databázového souboru

Při výběru tabulky může ovšem nastat situace, kdy je potřeba nahrát data z více tabulek. K tomuto účelu zde slouží tlačítko „+“ (plus), které po kliknutí vygeneruje další kontrolní seznam tabulek a uživatel díky tomu bude moci vybrat data ze dvou tabulek. Po přidání prvního seznamu se mu dále zobrazí i tlačítko „-“ (mínus), která má obrácenou funkci, tedy odebere jeden tzv. kontrolní seznam. Uživatel má možnost si takto vygenerovat až maximálně 6 nových kontrolních seznamů (Obr. 22). Tento počet je dán faktem, že je potřeba vyplnit v rámci položky sedm údajů, takže každý další kontrolní seznam by byl už nadbytečný.

Obr. 22 Maximální využití tabulek u databázového souboru

Pokud následným výběrem uživatel vyplní všechny vygenerované kontrolní seznamy. Dojde k tomu, že se do kontrolních seznamů, znázorňující název zakázky, výrobek, číslo operace, pracoviště, začátek, konec a počet kusů, nahrají všechny názvy sloupečků ve vybraných tabulkách. Samozřejmě pokud uživatel nebude generovat žádný další kontrolní seznam pro novou tabulku, a nechá si jen základní, tak dojde k výběru názvu sloupečků jen z něj.

Oproti textovým souborům je tu u možnosti vpravo nahoře pro náhled na vybrané tabulky. Tato možnost se dá aktivovat v případě vyplnění všech kontrolních seznamů pro výběr tabulek, přičemž dojde k rozšíření okna a jeho následného rozdělení mezi počet vybraných tabulek (Obr. 23). Tato možnost je především z důvodu praktičnosti, kdy se uživatel může podívat, jaké sloupečky se v dané tabulce nachází.

Nahrání procesních dat

Změnit soubor: C:\Users\raven\Desktop\data\pro Tagy.acodb Použít použitou konfiguraci: ☒ pro Tagy.acodb (12. 5. 2014) Náhled na tabulku: ☒

Vyberte název tabulky: Tabulka1 Tabulka2 Tabulka3

Zakázka: Zakazka Výrobek: Vyrobenek Číslo operace: CiskoOperace Pracoviště: Pracoviste Začátek: Zacek Konec: Konec Počet kusů: PocetKusu

Odeslat Zrušit

Zacatek	Konec
1.3. 2014 6:30:00	1.3. 2014 9:00:00
1.3. 2014 9:00:00	1.3. 2014 14:00:00
1.3. 2014 14:00:00	1.3. 2014 15:00:00
1.3. 2014 6:30:00	1.3. 2014 9:00:00
1.3. 2014 14:00:00	2.3. 2014 7:00:00
2.3. 2014 7:00:00	2.3. 2014 8:00:00
1.3. 2014 9:00:00	1.3. 2014 11:30:00
2.3. 2014 7:00:00	2.3. 2014 12:00:00
2.3. 2014 12:00:00	2.3. 2014 13:00:00
1.3. 2014 9:00:00	1.3. 2014 11:30:00

Pracoviste	PocetKusu	Vyrobenek
SU_2	10	part_0-10
FR	10	part_0-20
MONT	10	part_0
SU_1	10	part_0-10
FR	10	part_0-20
MONT	10	part_0
SU_2	10	part_0-10
FR	10	part_0-20
MONT	10	part_0
SU_1	10	part_0-10

CiskoOperace	Zakazka
O1.10	O1
O1.20	O1
O1.30	O1
O2.10	O2
O2.20	O2
O2.30	O2
O3.10	O3
O3.20	O3
O3.30	O3
O4.10	O4

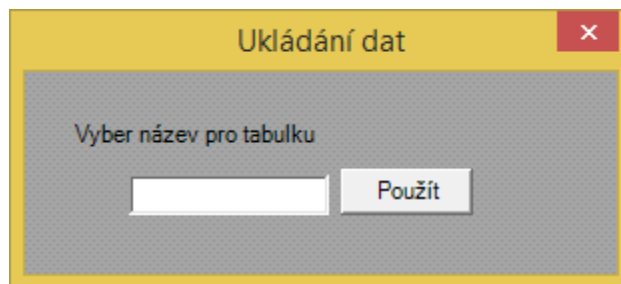
Obr. 23 Náhled na tabulky v rámci jednoho souboru

Způsob ukládání konfigurace je obdobný jako u textových souborů, v tomto případě se ukládají i čísla vybraných tabulek, takže uživatel je schopen otevírat pomocí vytvořené konfigurace i takto složitě propojená rozhraní. U textových souborů jsou tyto vygenerované kontrolní seznamy brány jako prázdné, protože v té chvíli neexistují.

Obecně je tento způsob propojování tabulek poměrně složitou disciplínou, protože při odeslání dat musí dojít u každé vybrané položky, znázorňující sloupeček, k dohledání podle názvu sloupečku o jakou tabulku se vůbec jedná - tedy jestli se v dané tabulce tento sloupeček nachází či nikoli. Po nalezení názvu tabulky pro jednotlivé data je možné odeslat ona data sloupeček po sloupečku.

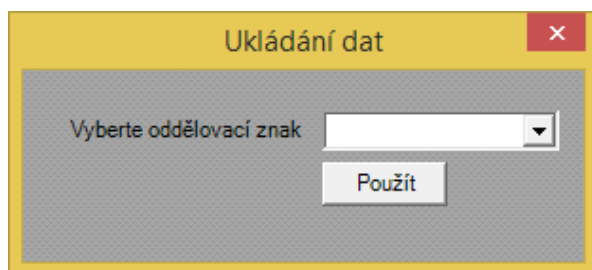
5.1.3 Uložení souboru

Pod tímto pojmem se ukrývá přesněji uložení dat, se kterými uživatel v době ukládání pracuje. Ukládání může být dvojího druhu, buď uložení do textového, nebo databázového souboru. Při ukládání dochází k uložení pouze těch nejdůležitějších informací, jako jsou: název zakázky, název výrobku, číslo operace, pracoviště, kde by se měla práce končit, její začátek a konec, jaká část je již hotová (v procentech), její priorita, počet potřebných kusů, vepsaná poznámka a její skutečný začátek a konec. Výběrem názvu nového souboru a místa uložení to ovšem nekončí. Pokud jde o ukládání do databázového souboru – jako je Excel má uživatel dále možnost si vybrat jak se bude jmenovat tabulka, do které data ukládá (Obr. 24). Při tomto druhu ukládání je preferován formát .xls pro jeho větší univerzálnost. Druhým způsobem uložení může být tedy do textového formátu. Zde je upřednostňován základní textový formát .txt a také formát .tbl, který se často používá k ukládání textových dat v podobných programech pro řízení výroby.

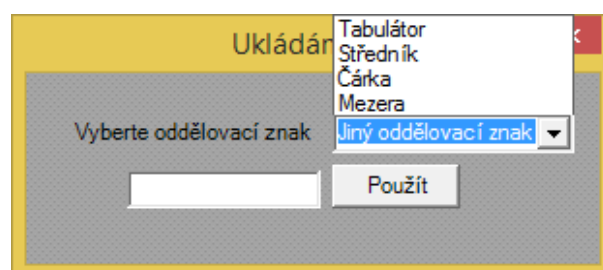


Obr. 24 Databázové ukládání

Při tomto druhu ukládání má uživatel možnost si vybrat jakým znakem chce oddělovat data v souboru (Obr. 25). Mezi základní oddělovací znaky, která jsou hned v nabídce, patří: oddělení tabulátore (asi nejčastěji používaný), středníkem, čárkou a mezerou. Samozřejmě je vhodné použít takový oddělovací znak, který se následně nebude vyskytovat v žádném datovém údaji. To by mohlo vést k nesrozumitelnosti při následné snaze data znovu otevřít. Z tohoto důvodu se používá nejčastěji tabulátor, který se běžně v textových datech nevyskytuje. Další možností, kromě těchto 4 základních je použít tzv. jiný oddělovací znak. Po výběru této možnosti se objeví nová kolonka pro vyplnění (Obr. 26), kam se může napsat libovolný znak – popřípadě text, kterým budou data následně v souboru oddělena.



Obr. 25 Textové ukládání.



Obr. 26 Ukládání s jiným oddělovacím znakem.

Ukládání dat může například sloužit v situacích, kde jsou firemní data během procesu upravena a je následně potřeba tyto data sdílet s ostatními pracovišti. Další možností může být situace, kdy by bylo potřeba zpětně dohledat data z některého dne. To se může například hodit při nějaké důkladnější analýze, jak si výroba vedla v určitém časovém horizontu. Pomocí těchto údajů se dají hledat problémy, které se na pracovišti často objevují. Tyto problémy se následně mohou lehce pojmenovat a odstranit.

5.1.4 Rychlé uložení

Možnost rychlého uložení nabízí uživateli možnost si veškeré datové informace, které má v současnosti zanesené v programu uložit pouhým jedním kliknutím do tzv. schránky pro rychlé manipulaci. Ve skutečnosti se jedná o textový soubor ve složce planováníVýroby s názvem procesy.csv. Uložení je obdobné jako při uložení souboru s tím rozdílem, že zde se nemusí nic zadávat a program vše udělá sám. Při prvním uložení se v této složce vytvoří nový již zmíněný soubor procesy.csv, při každém dalším rychlém uložení je tento soubor přepsán novým, takže je možné mít rychle uložená jen jedna data. Takto rychle uložená data v sobě nesou více informací než při klasickém uložení souboru. V tomto případě se ukládá i barvy jednotlivých procesů. Při běžném uložení jsou tyto informace o barvě poměrně zbytečné

a můžou působit pro uživatele, který by se díval do souboru, spíš matoucím dojmem. V tomto případě uživatel nemá ovšem přímý přístup do souboru, pokud si ho tedy nenajde ve složce, a pro samotný program tato data matoucí nejsou, protože ví - co se kde v souboru nachází. Tato funkce může vhodně sloužit například v situacích, kdy si chceme průběžně ukládat změněná data a nechceme při tom generovat nový soubor. Další dobrou možností by nemuselo být udělení automatické volání této metody pro rychlé uložení. Například v nějakém časovém intervalu, kdy by se nám v podstatě zálohovaly současná data. Mohla by se zde objevit i situace, kdy není možné provést rychlé uložení. Tato situace se může stát, pokud uživatel nemá přístupová práva k zápisu do místa umístění software. Pokud by tato situace nastala, tak o ní bude uživatel informován prostřednictvím varovné hlášky.

5.1.5 Rychlé načtení

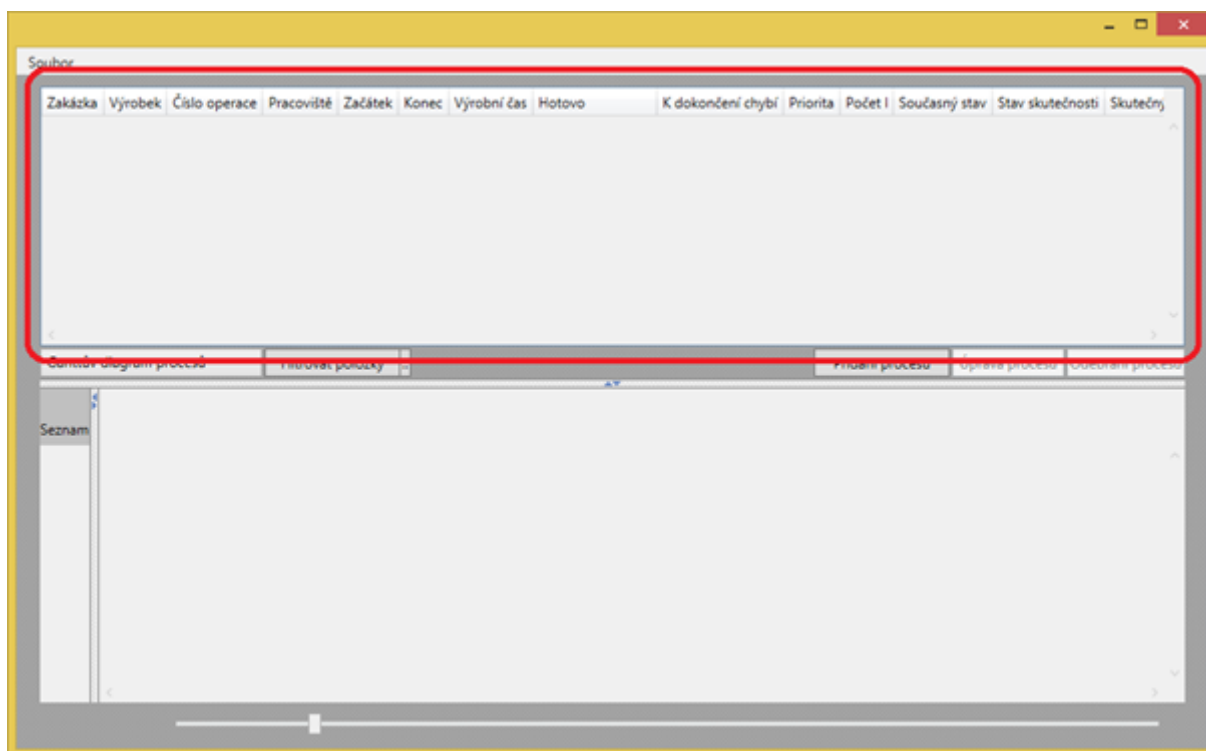
Rychlé načtení je úzce spojeno s rychlým uložením. Pokud uživatel provedl rychlé uložení aspoň jednou, tak je možné provést rychlé uložení, které načte data z okamžiku rychlého uložení. Pokud k žádnému uložení nedošlo a uživatel by chtěl rychle načíst data, objeví se varovné hlášení, že soubor pravděpodobně ještě neexistuje. Tato funkce je koncipována tak, že pokud by uživatel provedl rychlé uložení a například vypnul software, tak pak při rychlém načtení by pracoval v místě, kde provedl poslední rychlé uložení. Tato funkce má i jiný ještě potenciál. Pokud by pracovník přišel ráno k pracovišti – zapnul si počítač a chtěl by se podívat na plánovanou práci, tak by nemusel provádět manipulaci s klasickým otevřením souboru. V tomto případě by provedl pouhé jedno kliknutí pro rychlé načtení, přičemž by na něj ve schránce čekala aktuální data.

Tento princip má ještě větší potenciál. Dejme tomu, že ve firmě bude několik pracovišť, která všechna pracují se stejnými daty. Tato data by mohla být nahrána na nějaké centrálním serveru firmy a při použití funkce rychlého načtení by se data načítala přímo z tohoto firemního serveru. V tom případě by už nemohla nastat situace, kdy by někdo neměl například nejaktuálnější data, popřípadě by nevěděl, že nějaké jiné pracoviště se opozdilo. Pokud by pracovníci neměli přístup pro manipulaci s daty, tedy přesněji všechna data by se měnila automaticky – strojově nebo popřípadě by se o ně starala jedna osoba (jde především o změny ve styku aktuálních stavů). Tak v tomto případě by stačilo, kdyby se na jednotlivých pracovištích, dejme tomu každou minutu, prováděla automaticky funkce pro rychlé načtení. Takto rychle nahrána data z firemního serveru by byla vždy aktuální. Jejich aktuálnost by byla v podstatě zajištěná funkcí rychlého uložení volanou každou minutu na firemní server. K tomuto ukládání by stačil jeden zapnutý tento software upravený tak, aby se funkce rychlého uložení volala každou minutu sama od sebe za použití časovače – ať už došlo k nějaké změně či nikoli.

Pomocí těchto dvou funkcí pro rychlé načtení a rychlé uložení je tedy možné koordinovat tzv. online výrobu v celé firmě velmi jednoduše a hlavně efektivně – za použití pouhého textového souboru.

5.2 Tabulka procesů

Tato tabulka se nachází v horní polovině programu (viz. Obr. 27) a slouží k zobrazování informací o jednotlivých procesních činnostech. Pokud například uživatel bude chtít znát informace o některé procesní činnosti, tak po kliknutí na onu položku dojde nalezení oné položky v tabulce. Uživatel tak nemůže ztratit informace o tom, na který proces se zrovna dívá. Informace v tabulce jsou seřazena v takovém pořadí, kdy první sloupeček poskytuje nejdůležitější data a směrem doleva jejich důležitost klesá.



Obr. 27 Umístění tabulky s procesy.

5.2.1 Zobrazená data

Některá tato data jsou přímo získaná ze souboru – popřípadě z „ručně“ vytvořených procesních činností. Některá zobrazená data jsou ovšem vypočítaná a graficky znázorněna v tabulce. Základní sloupce tabulky, jejíž data jsou pouze „tlumočení“ dat z procesních činností jsou: zakázka, výrobek, číslo operace, pracoviště, začátek, konec, priorita, počet kusů, skutečný začátek a skutečný konec. Mezi data, které se pro zobrazení nejdříve musí vypočítat, patří například výrobní čas – což je údaj, který informuje o celkové délce procesu. Zjednodušeně řečeno se jedná o velikost získanou odečtením začátku dané činnosti od jejího začátku. V tomto případě je zde použit naprogramovaný převodník, který nezobrazuje pouze surová data, ale dává jim určitý řád, díky kterému působí přehledně. Jedná se o rozdělení na tři úseky: dny, hodiny a minuty. Velice podobným údajem je i údaj informující kolik chybí k dokončení, v tomto případě je ovšem ještě od výrobního času odečten procentuální podíl již hotové činnosti. Pokud je činnost hotová ze 100%, tak se místo výpisu textu zobrazí text hotovo (Obr. 28).

K dokončení chybí	
16 hodin 40 minut	
16 hodin 40 minut	
22 hodin 40 minut	
Hotovo	
1 dni 23 hodin 40 minut	
1 dni 2 hodin 10 minut	

Obr. 28 Sloupeček tabulky kolik chybí k dokončení.

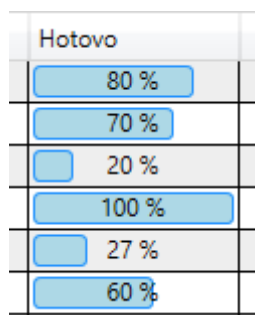
Mezi další takto proměnná data v čase patří zobrazení současného stavu. V tomto sloupečku se provádí výpočet toho, jaké má daná činnost zpoždění – popřípadě předstih oproti plánované skutečnosti. Pokud má například pracovní činnost předstih, tak dojde ke změně barvy tohoto textu na zelenou barvu a před text znázorňující velikost se připiše plus. Pokud má činnost zpoždění, tak bude mít text červenou barvu a před onen text se připiše mínus. Dalším stavem, který zde může nastat, je už zmíněné dokončení činnosti. Tyto stavy jsou patrné na Obr. 29.

Současný stav	
- 2 hodin 32 minut	
- 1 hodin 39 minut	
- 27 minut	
Hotovo	
+ 18 minut	
+ 3 hodin 4 minut	

Obr. 29 Sloupeček tabulky zobrazující současný stav.

Dalším zajímavým sloupečkem je sloupeček stavu skutečnosti, který podává informace, jak si vede současný stav vůči plánu. Optimálního stavu, kterého je možné zde dosáhnout je „souběžně s plánem“. To znamená, že vše jde tak, jak bylo naplánováno. Zobrazení zpoždění skutečnosti oproti plánu, popřípadě předstih oproti plánu, se zobrazuje obdobně jako v předešlých příkladech – tedy barevně a se znamínkem na začátku.

Posledním sloupečkem, co má zobrazená data závislá na datech činnosti je zobrazení hotové části. Přesněji se jedná o procentuální podíl již hotové části vůči celku. Jedná se o vykreslení zaobleného trojúhelníku do buňky, kde šířka tohoto rámečku odpovídá šířce buňky krát hotový podíl (Obr. 30). Samozřejmě není možné dosáhnout více jak sta procent.



Obr. 30 Zobrazení z jaké číslí je činnost hotova.

5.2.2 Vybrání položky

Položka zobrazená v tabulce se dá vybrat třemi různými způsoby. Může být buď vybrána prostřednictvím stromu položek (menu se zakázkami – položkami nalevo) zobrazující názvy zakázek a položek v rámci dané zakázky, kliknutím na vykreslený Ganttův diagram, kdy se daný vykreslený obdélník vztahuje k celému řádku, ve kterém leží nebo jednoduše kliknutím na položku v tabulce. Takto vybraná položka se barevně odliší od ostatních položek v tabulce (Obr. 31). Pokud tato položka má nějakou napsanou poznámku, tak se zobrazí přidáný řádek, sloužící k zobrazení dané poznámky (Obr. 31). Pokud vybraná položka nemá napsanou žádnou poznámku, tak se tento přidáný řádek vůbec nezobrazí. Pro zrušení zobrazené položky je před poznámkou přidáné tlačítko pro schování tzv. detailu daného procesu.

Řídicí systém	vyr_1-10	Analýza problematiky	22_A1	10.03.2014 10:46	10.03.2014 11:46	1 hodin	100 %	Hotovo
Řídicí systém	vyr_1-20	Specifikace požadavků	22_B1	10.03.2014 11:46	10.03.2014 13:46	2 hodin	100 %	Hotovo
Řídicí systém	vyr_1-30	Předběžný návrh	22_B1	10.03.2014 13:47	11.03.2014 07:17	17 hodin 30 minut	100 %	Hotovo
Řídicí systém	vyr_1-40	Prototypování	a	11.03.2014 07:17	11.03.2014 11:17	4 hodin	90 %	24 m
Zde se zobrazuje vepsaná poznámka								
Řídicí systém	vyr_1-50	Vyhodnocování	22_C3	11.03.2014 07:17	11.03.2014 11:17	4 hodin	60 %	1 hod
Řídicí systém	vyr_1-60	Úprava návrhu	22_C3	11.03.2014 11:17	12.03.2014 06:47	19 hodin 30 minut	0 %	19 hr

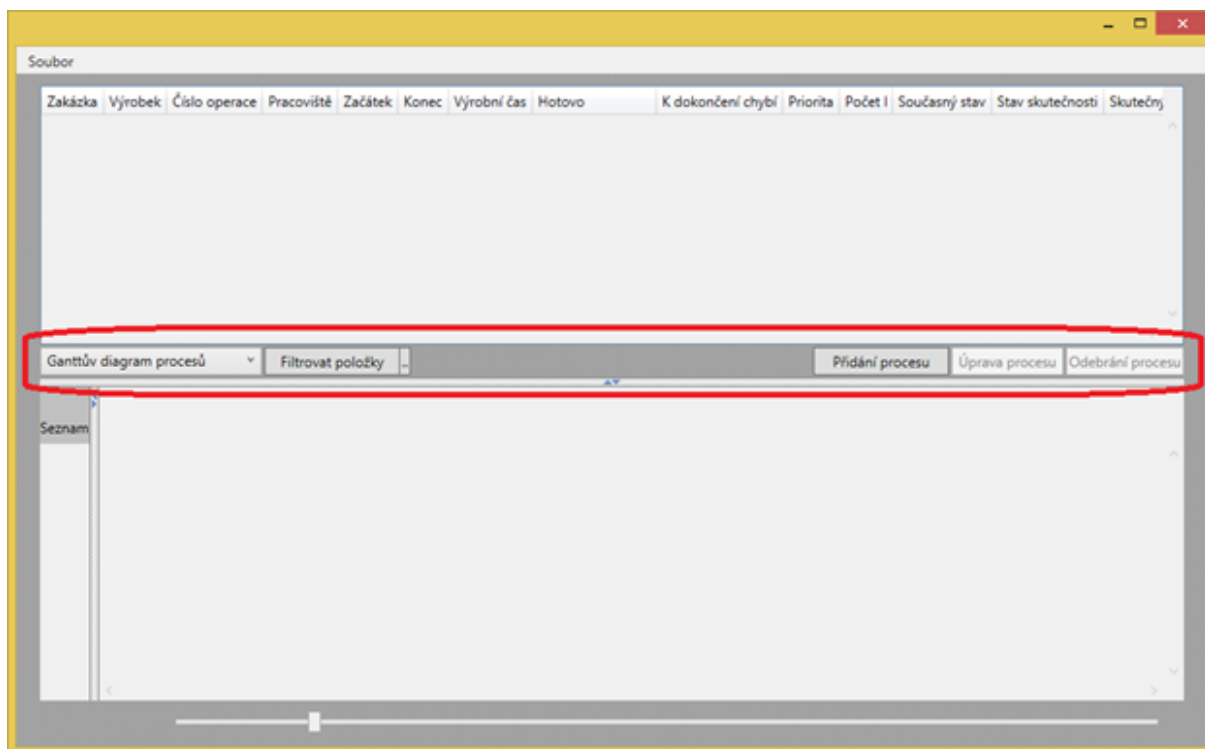
Obr. 31 Vybraná položka v tabulce s daty.

Pokud dojde k takovému výběru položky – dojde k zpřístupnění nových možností, jak s touto vybranou položkou manipulovat. Tím je myšlená možnost použití tlačítek na úpravu a odebrání daného procesu. Pokud žádná položka není vybrána – tedy není co měnit nebo odebrat, tak jsou jinak tato tlačítka pro manipulaci znepřístupněna. Pokud například uživatel si vybere nějakou položku jak už podle názvu položky (v menu se seznamem zakázek – položek nalevo) nebo kliknutím na daný diagram, tak dojde k dohledání té dané položky v tabulce, aby uživatel měl okamžitý náhled na informace dané položky.

Tato tabulka nemusí vždy zobrazovat všechna data, ale pouze Ta, která uživatel chce. Pokud například provede filtrování, které je obsahem další kapitoly, nebo schová data v rámci zakázky pomocí panelu s názvy zakázek – položek, tak se ona data nebudou zobrazovat. Vždy se tedy zobrazují v tabulce pouze data, kterou jsou v danou chvíli vykreslena.

5.3 Manipulace s daty

Ve skutečnosti jde především o možnost vytvářet, upravovat a odebírat data. Kontrolky, které slouží pro manipulaci s daty, se nachází uprostřed okna – mezi tabulkou s daty a plochou pro vykreslování položek (Obr. 32). Mezi tyto tzv. procesní manipulace patří mimo jiné i možnost filtrovat libovolně data a změna Ganttových diagramů mezi procesy a prací.



Obr. 32 Tlačítek pro manipulaci s diagramem.

5.3.1 Úprava procesní činnosti

Jelikož se data jednotlivých procesních činností mohou za chodu měnit, je vhodné, aby uživatel měl možnost upravovat data, která se během činností změnila. K tomuto přepisování slouží jednoduchý formulář, který se aktivuje po kliknutí na tlačítko Úprava procesu, za předpokladu předem vybraného procesu. Formulář, jak je patrné na Obr. 33, zobrazuje současná aktuální data zvoleného procesu, kde uživatel může jednoduše přepsat změněná data. Se změnou dat jako jsou zakázka, výrobek, číslo operace, pracoviště a počet kusů se příliš počítat nedá, ovšem jejich úprava není vyloučená. Zajímavější data pro úpravu jsou už například plánovaný začátek, plánovaný konec, skutečný začátek a skutečný konec. Tyto data se mohou měnit v závislosti na několika faktorech – například pokud dojde ke zpoždění a následnému přeplánování. Dalším zajímavým údajem je poznámka, která umožňuje sdělovat informace, týkající se konkrétního procesu. Pokud například dojde ke zpoždění, tak zde může být uveden důvod, proč k němu došlo – popřípadě kdo je za něj zodpovědný. Dalším důležitým údajem je procentuální vyjádření – kolik procent z daného procesu je již

dokončeno. Aby bylo možné zjišťovat aktuální stav procesu vůči skutečnosti. Dalším údajem je priorita, jak už bylo řečeno, jedná se pouze o informační údaj v tomto případě. Posledním údajem, který se zde nachází, je barva, která určuje barvu, kterou bude plánovaná procesní činnost vykreslena. Tato změna barvy může velice zpřehlednit diagramy. Můžeme třeba stejným pracovištěm nastavit stejnou barvu a tím docílit toho, že při prvním pohledu bude hned patrné, přes jaká pracoviště zakázka bude putovat. Další možností je například jednotlivým procesům nastavit stejnou barvu v dané zakázce.

Při pokusu o dokončení úprav je provedená kontrola, zda daná data odpovídají svým datovým typům. Tedy aby například počet kusů nebyl vyjádřen slovně, konec nepředbíhal začátek a podobně.

Obr. 33 Formulář pro úpravu procesu.

5.3.2 Přidání procesní činnosti

Tato možnost je zde především z toho důvodu, že se nám neplánovaně může objevit úplně nová procesní činnost nebo například stav, kdy si potřebujeme naplánovanou procesní činnost rozdělit na dva procesy. V tom případě by došlo k úpravě onoho procesu a přidání nového tak, aby navazoval na upravený proces. Další vhodnou možností využití této možnosti je, pokud nemáme vstupní data ve formě souboru nebo máme, ale v takovém typu souboru, který program nepodporuje. Pomocí této možnosti je možné vytvořit libovolné množství nových procesních činností.

Formulář na přidávání nových pracovních činností se velmi podobá tomu na upravování pracovních činností, jak je patrné z Obr. 34. Oproti němu nemá však vyplněné nic a časové položky jsou nastaveny na současnost.

Položka výrobek a číslo operace by mělo být úplně unikátní a neměl by existovat žádný jiný proces se stejným názvem. Pokud by k tomu totiž došlo, tak by mohl být problém s celkovou identifikací procesů, protože ta provádí pomocí těchto unikátních identifikátorů. Pro možnost přidání procesu je potřeba vyplnit položky z prvního řádku – tedy nemohou být tato data reprezentována bez textu.

Obr. 34 Formulář pro přidání procesu.

5.3.3 Odebrání procesní činnosti

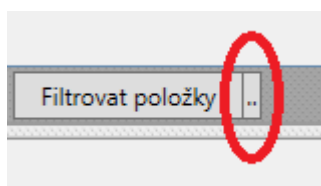
Tato možnost slouží, jak už název napovídá k trvalému odstranění libovolné – vybrané položky. Tato možnost se dá především uplatnit v situacích, kdy došlo ke zrušení nějaké zakázky nebo změně technologie, se kterou byl proces původně plánován řešit.

5.3.4 Filtrování procesů

Filtrování procesů slouží k zobrazování pouze těch procesů, které chceme. Tato schopnost se především uplatňuje v situacích, kdy máme velké množství procesů a my chceme třeba zobrazovat pracovní činnosti dané zakázky. U několika položek je možné vybrat rozmezí, v kterém se položky, které chceme zobrazovat, nacházejí. Pokud ve formuláři nějakou položku nevyplníme, tak se podle ní filtrovat nebude. U položek znázorňují datum/čas je potřeba provést zaškrtnutí (Obr. 35). Je to z toho důvodu, že pokud by uživatel musel zadávat celé datum a čas bez jisté „navigace“ tak by mohl mít problém s vyplněním. Takto jsou data pro datum/čas již předvyplněná na současnost a uživatel jen jednoduše přepíše to, co potřebuje.

Obr. 35 Formulář pro filtrování procesů.

Pokud by uživatel provedl filtrování a chtěl svoje rozhodnutí vrátit do původního stavu, tak má dvě možnosti. Může opět zahájit filtraci a nic nevyplnit a pouze filtrovat – tedy ve smyslu, že nezadá žádná omezující data a tím pádem vykreslí vše. Druhým způsobem je, že použije malé tlačítko umístěné napravo od tlačítka pro filtraci (Obr. 36). Po stisknutí tohoto tlačítka se zobrazí všechny pracovní činnosti.

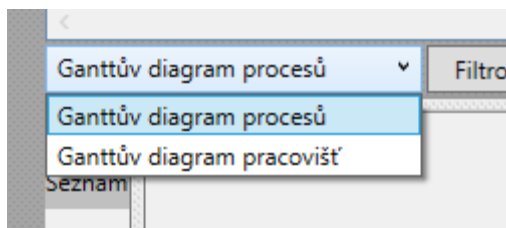


Obr. 36 Zrušení filtrovaných procesů.

5.3.5 Změna diagramu

Tato možnost umožňuje přepínat mezi Ganttovými diagramy znázorňující procesy a pracoviště (Obr. 37). Při samotném vykreslování je vždy bráno v potaz jaký text se v této kontrolce nachází, protože tento text určuje co, a v jakém pořadí se bude vykreslovat. Při zvolení Ganttových diagramů pracovišť dojde k zablokování možnosti upravování procesů a odebrání procesů. Je to z toho důvodu, že provádění úprav všech položek daného pracoviště by nemělo smysl, stejně tak jako smazání celého pracoviště. Další věcí, která se tzv. deaktivuje, je možnost při kliknutí na položku v seznamu položek zobrazení oné položky v tabulce. Je to z toho důvodu, že je možné zobrazit vždy pouze jednu položku a dané

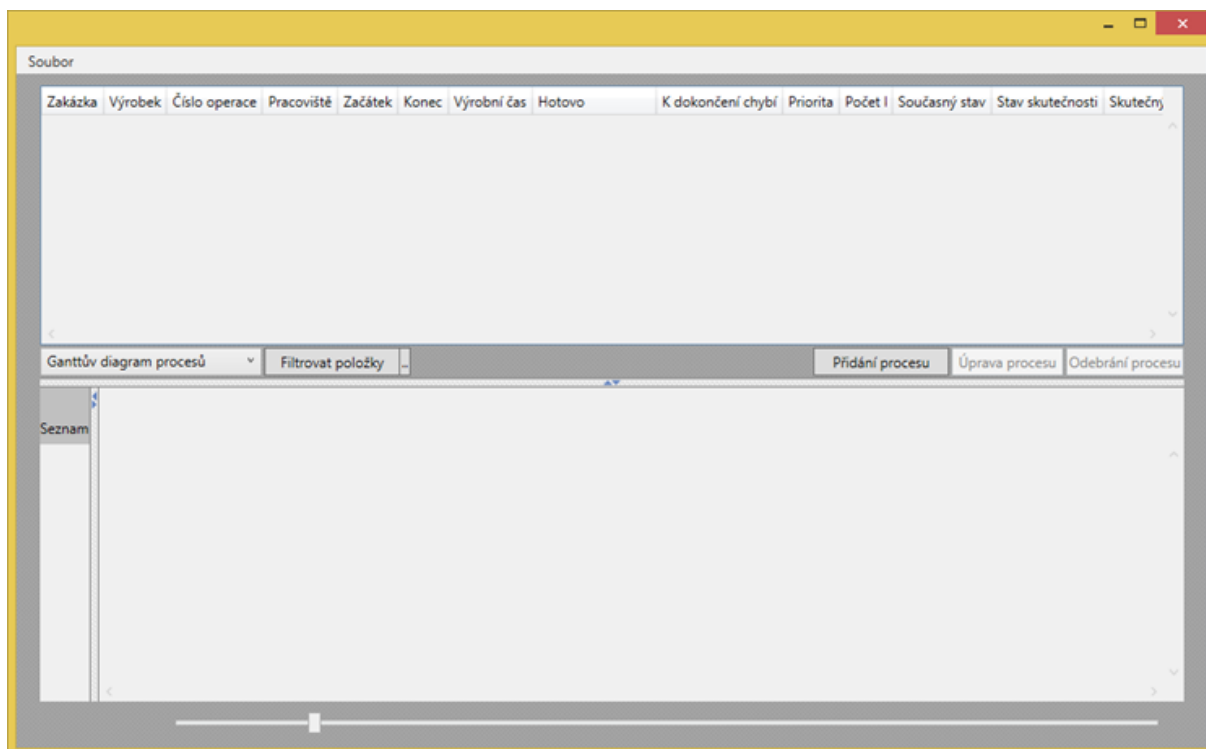
pracoviště může zastupovat libovolné množství položek.



Obr. 37 Zobrazení možnosti přepínání mezi diagramy.

5.4 Rozložení softwaru

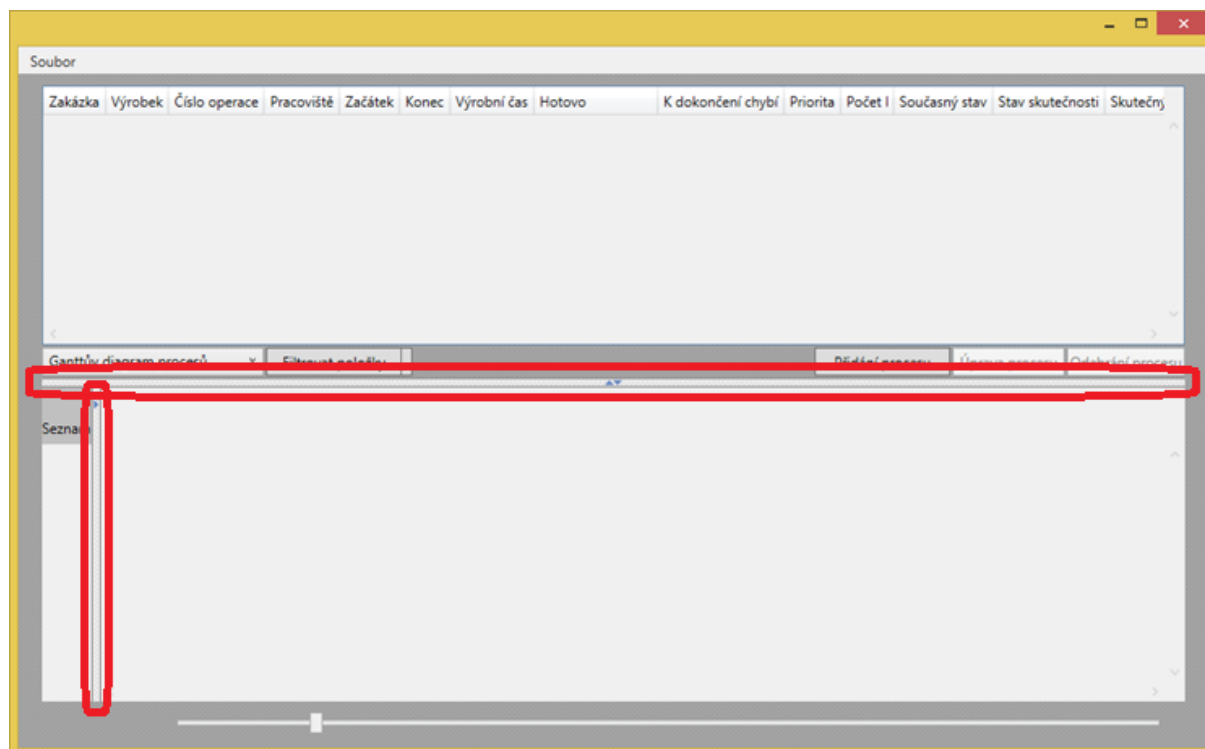
Rozložení realizovaného softwaru (Obr. 38) může na první pohled působit velmi staticky, což není ovšem pravda. Při manipulaci s velikostí okna, ať už jde o maximalizaci okna či ruční nastavení velikosti tažením za okraj okna, si okno zachovává základní rozložení. V tomto základním nastavení rozložení zaujímá přibližně 35% výšky okna tabulka s data, 50% vykreslování diagramů a zbylých 15% zaujímají kontrolky a okrajové mezery.



Obr. 38 Rozložení softwaru.

Tento poměr určitě není špatný. Může ovšem nastat situace, kdy data uživatele příliš nezaujmají, a uživatel chce mít větší grafický náhled, popřípadě se chce podrobně podívat na data v celkovém kontextu. K tomu účelu je v softwaru zakomponován tzv. oddělovač, který nabízí uživateli nastavení rozložení dle vlastní libosti. Realizovaný software obsahuje rovnou dva tyto oddělovače (Obr. 39). Vertikální oddělovač – tedy ten větší, co se může posouvat nahoru a dolů umožňuje nastavit poměr mezi tabulkou s daty a vykreslenými diagramy. Ve výsledku je tedy možné, jednoduchým posunutím, se úplně oprostít od tabulky s daty

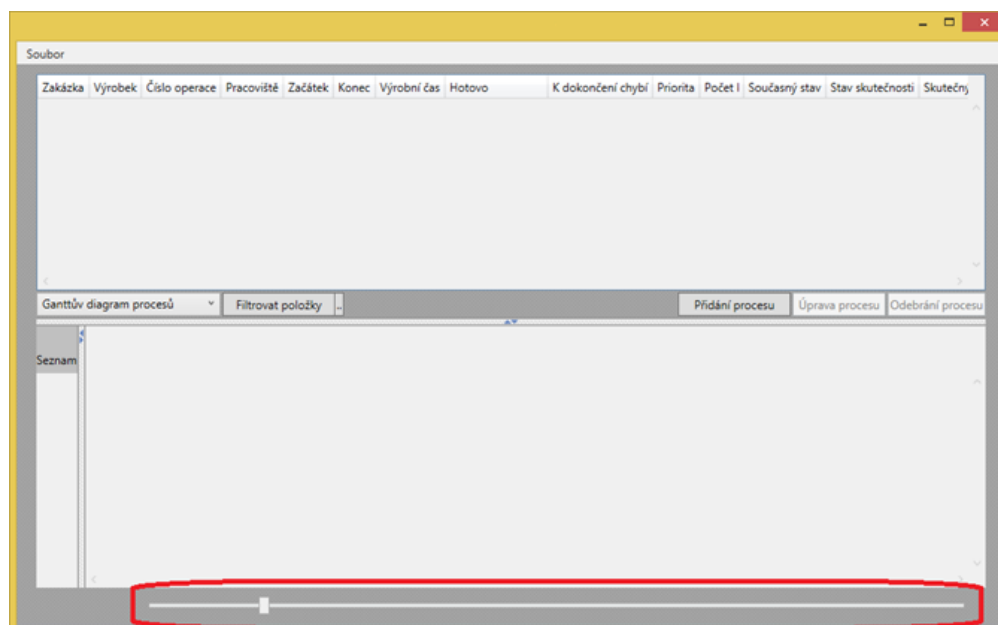
popřípadě vykreslenými diagramy a využít tak plně zobrazovanou plochu, která je zrovna potřeba. Horizontální oddělovač – tedy ten menší, co se může posouvat pouze doleva a doprava je poněkud skromnější. Umožňuje pouze manipulaci se zobrazenými názvy položek popřípadě zakázek. Pokud by měla například nějaká položky příliš dlouhý název, tak by bylo zbytečné kvůli jejímu celému názvu omezovat plochu na kreslení. Z tohoto důvodu má uživatel možnost si nastavit, zda potřebuje vidět celý název nebo jen jeho část – popřípadě jak moc velkou.



Obr. 39 Ukázka oddělovačů v softwaru.

Jak již bylo zmíněno, pro lepší manipulaci s měřítkem časové osy je zde v softwaru zakomponován posuvník (Obr. 40), jehož aktuální hodnota udává velikost časové jednotky osy.

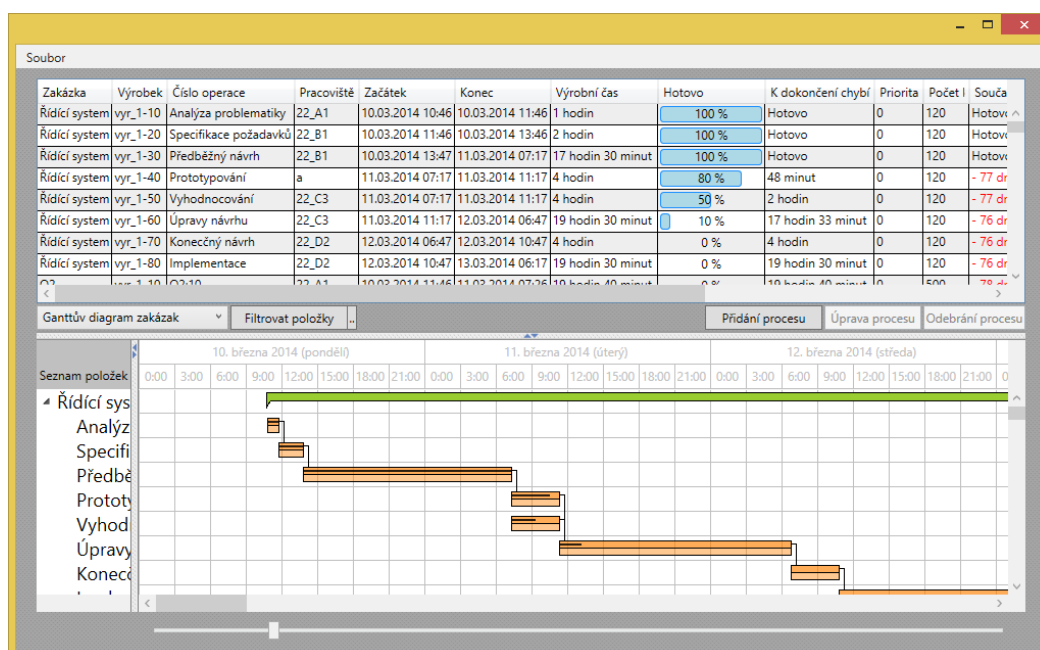
Ke změně - překreslení měřítka (nemusí se jednat jen o časové jednotky) dochází vždy až po uvolnění posuvníku, aby se předešlo zbytečnému překreslení tzv. všeho při posunutí o zanedbatelnou vzdálenost – což může být poměrně vhodné například v situacích, kdy bychom měli ohromné množství pracovních procesů. Další možností je vytvořit posuvník pouze o, v tomto případě, čtyřech stavech, mezi kterými by se jen přepínalo a ne posouvalo.



Obr. 40 Vyznačen posuvník v programu

5.5 Výsledný software

Na Obr. 41 je znázorněn, jak vypadá výsledný software v praxi. Software je možný k použití v různých oblastech lidské činnosti související s plánováním. Jak už ve formě klasických Ganttových diagramu – například plánování výroby, což byl jejich hlavní účel, tak může sloužit i jako pouhý harmonogram pro zobrazování naplánovaných činnosti – například ve formě klasického rozvrhu. Při použití softwaru jako harmonogramu jde v podstatě o nevyužívání jeho dynamických vlastností. Tento software byl vyvíjen v operačním systému Windows 8, byl však i otestován ve Windows 7, kde neprojevil žádné selhání.



Obr. 41 Výsledný software v praxi.

6 ZÁVĚŘ

V diplomové práci jsem provedl analýzu funkcí Ganttových diagramů a nastínil způsob jejich programové realizace za využití OOP. Z těchto mých návrhů realizace jsem následně naprogramoval DLL knihovnu, určenou pro vykreslování Ganttových diagramů s pomocí programovacího jazyka C#. Tato knihovna je součástí DVD i včetně jejího celého zdrojového kódu, který je důkladně okomentován. Tato knihovna byla programována v programu Microsoft Visual C# 2010 Express.

Pomocí této knihovny jsem následně naprogramoval celý software, určený pro vytváření Ganttových diagramů výrobních procesů. Funkční software je také součástí DVD a je zde přiložen i celý zdrojový kód tohoto softwaru včetně důkladného okomentování. Na DVD jsou také přiložena některá uměle vytvořená data, které je schopen zpracovat. Tato data tu nejsou z toho důvodu, že by software byl omezen na vstupní data, ale z hlediska toho, že pokud by si někdo chtěl onen software cvičně vyzkoušet, tak by si musel pracně data vymýšlet, což může být poměrně komplikované pro někoho, kdo nikdy tato data teřba neviděl. Software je jinak ve skutečnosti velmi univerzální, co se struktury zpracování dat týče. Pokud mají data aspoň nějaké vnitřní uspořádání tak nemá problém s jejich zpracováním.

SEZNAM POUŽITÉ LITERATURY

- [1] Wallace Clark and Henry Gantt (1922) [*The Gantt chart, a working tool of management*](#). New York, Ronald Press.
- [2] CLARK, Wallace. *Ganttovy diagramy: pomůcka k řízení podniků a práce*. Praha: Orbis, 1931, 141 s.
- [3] Chatfield, Michael. "Gantt, Henry Laurence (1861-1919)." *History of Accounting: An International Encyclopedia*, edited by Michael Chatfield and Richard Vangermeersch. New York: Garland Publishing, 1996. P. 269.
- [4] WALTER, Jaromír, Stanislav VEJMOLA and Petr FIALA. *Aplikace metod síťové analýzy v řízení a plánování*. 1. vyd. Praha: SNTL - Nakladatelství technické literatury, 1989. 282 s. ISBN 80-03-00101-3.
- [5] JIRCHÁŘ, Václav. Metoda CPM/PERT. In: *Metoda CPM/PERT* [online]. 2009 [cit. 2014-05-30]. Dostupné z: http://kmlinux.fjfi.cvut.cz/~jahodfra/ztgb/prezentace/cpm_pert.pdf
- [6] Winston W. Royce (1970). "[Managing the Development of Large Software Systems](#)" in: *Technical Papers of Western Electronic Show and Convention* (WesCon) August 25–28, 1970, Los Angeles, USA.
- [7] ČÁPKA, David. Úvod do WPF. In: *Úvod do WPF* [online]. [cit. 2014-05-30]. Dostupné z: <http://www.itnetwork.cz/c-sharp-tutorial-wpf-uvod-a-prvni-formularova-aplikace>

PŘÍLOHA

Příložené DVD obsahuje několik složek, jsou to:

DLL Knihovna – tato složka obsahuje funkční knihovnu připravenou na použití.

Zdrojový kód knihovny – obsahuje celý zdrojový kód s komentářem .

Software – obsahuje spouštěcí .exe soubor s knihovnou, kterou potřebuje (po vytvoření konfigurace nebo rychlé uložení se zde vygeneruje nová složka s danými údaji).

Zdrojový kód softwaru – obsahuje kompletní zdrojový kód celého softwaru.

Příložená data – V této složce jsou dva typy souborů, jsou to: „Soubor pro více tabulek.accdb“ a „Soubor pro textový přístup.tbl“, které program může například otevřít, jde zde vložen i komentář v souboru „komentář k obsluze.txt“ pro jejich obsluhu.