Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Josef Závišek

Editor elektrotechnických schémat

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. Jan Kofroň

Studijní program: Informatika, obor Programování

2008

Děkuji panu RNDr. Janu Kofroňovi za vedení mé práce a za čas, který mi v průběhu její tvorby věnoval. Chtěl bych vyjádřit dík také svým rodičům za podporu a pomoc při sestavování práce i během celého studia.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne Josef Závišek

OBSAH

1 ÚVOD....6
2 ELEKTROTECHNICKÁ DOKUMENTACE....8
2.1 Požadavky na elektrotechnickou dokumentaci....8
2.2 Základní pravidla pro tvorbu schémat....9
2.3 Elektrotechnické značky....10
3 SOFTWAROVÉ ŘEŠENÍ, PROGRAMOVÁ DOKUMENTACE....12
3.1 Windows Presentation Foundation....12
3.1.1 Úvod....12
3.1.2 Výhody Windows Presentation Foundation....12
3.1.3 Závěr o WPF....17
3.2 Základní struktura programu....17
3.3 Vnitřní struktura....19
3.3.1 Reprezentace výkresu....19
3.3.2 Grafické prvky....19
3.4 Editace výkresu....21
3.4.1 Volby akcí....21
3.4.2 Transformace zadaného bodu....22
3.4.3 Uchopování objektů....23
3.4.4 Zoomování....23
3.4.5 Vykreslování grafických prvků....24
3.4.6 Rozpoznání klávesových zkratek a příkazů....24
3.4.7 Zjištění velikosti UIElementu....25
3.4.8 Kopírování grafických objektů....26
3.4.9 Undo a Redo....26
3.4.10 Ukládání a otevírání....27
3.5 Vizuální efekty v aplikaci....28
3.5.1 Animace tlačítek....28
3.5.2 Stylování a ikony....29
3.5.3 Dockování panelů....29
3.5.4 Vytvoření efektu odrazu....30
4 UŽIVATELSKÁ DOKUMENTACE....31
4.1 Instalace programu....31

4.2 Spuštění programu....31
4.3 Popis pracovního prostředí....31
4.4 Začátek práce s programem....33
4.4.1 Vytvoření nového projektu....34
4.4.2 Vytvoření nového výkresu....34
4.4.3 Vytvoření nové součástky a zařazení do knihovny....34
4.5 Editace schématu a součástky....34
4.5.1 Rozdíl mezi schématem a součástkou....34
4.5.2 Kreslení grafických prvků....35
4.5.3 Transformace grafických prvků....36
4.5.4 Vkládání součástek do výkresu....36
4.5.5 Výběr....37
4.5.6 Změna vzhledu objektů....37
4.5.7 Zoomování....37
4.5.8 Uchopování....37
4.5.9 Krokování a pravoúhlé kreslení....38
4.5.10 Ovládání programu pomocí klávesnice....39
4.6 Uložení do souboru....41
4.5.7 Tisk dokumentu....41
5 Závěr....42
5.1 Srovnání s ostatními editory....42
5.2 Možná vylepšení....44
LITERATURA....46
PŘÍLOHY....47
Obsah přiloženého CD....47

Název: Editor elektrotechnických schémat
Autor: Josef Závišek
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: RNDr. Jan Kofroň
e-mail vedoucího: Jan.Kofron@mff.cuni.cz

Abstrakt:
Předmětem mé práce je implementace grafického vektorového editoru elektrotechnických schémat. Editor je založen na koncepci CAD systémů a je navržen tak, aby splňoval normy, kterými se elektrotechnická dokumentace musí řídit. Práce se však neomezuje pouze na elektrotechniku. Obsahuje rozšiřitelnou knihovnu součástek, a tudíž může sloužit i pro tvorbu výkresů hydraulických, pneumatických, rozvodů plynu apod. V práci je mimo jiné také popsána nová technologie Windows Presentation Foundation, nad kterou byl editor vystavěn. Jsou také nastíněny přednosti a nové možnosti této technologie.
Klíčová slova: editor schémat, elektrotechnika, WPF, elektrodokumentace

Title: Editor of electrotechnica schemes
Author: Josef Závišek
Department: Department of Sofware Engineering
Supervisor: RNDr. Jan Kofroň
Supervisor's e-mail address: Jan.Kofron@mff.cuni.cz

Abstract:
The main task of this work is the implementation of a graphic vector electrotechnical editor. The editor is based on concept of CAD systems and is designed to accomplish standards that need to be followed. However work is not focused only on the electrotechnics. It includes also extensible component library and that's why it can be used for designing other types of technical documentation as well. The editor is developed on the base of the new technology Windows Presentation Foundation which is also described. The work demonstrates also new advantages and possibilities of this technology.
Keywords: scheme editor, electrotechnics, WPF, electrodocumentation

1 ÚVOD

Mezi nejdůležitější činnosti projektantů, techniků a odborných pracovníků technických oborů patří technické kreslení. Technické výkresy, ať už stavební, strojní, elektrotechnické či jiné, patří k základním částem technické dokumentace požadované právními předpisy při plánování, výrobě, uvádění do provozu, provozování a užívání technických zařízení.

V dnešní době zaujímá výpočetní technika v konstrukčním procesu nezpochybnitelné místo při výrobě a konstrukci. Na rychlost a kvalitu vývoje, konstrukce nebo technologickou přípravu výroby jsou kladeny velmi náročné požadavky. Vyhovět těmto požadavkům a obstát v tvrdé konkurenci na trhu není jednoduché. Pro zvládnutí těchto nároků jsou vyvíjeny nejrůznější CAD systémy (Computer Aided Design – počítačová podpora konstruování). Práci inženýrů a techniků v nejrůznějších firmách si dnes prakticky nelze bez použití těchto nástrojů představit.

V českých podnicích je dodnes používáno a požadováno tzv. konvenční zpracování elektrodokumentace. Jedná se o vytvoření nové nebo překreslení dosavadní dokumentace prostřednictvím univerzálního kreslícího programu. Elektrotechnikům jsou kromě CAD nástrojů dnes dány k dispozici i tzv. CAE nástroje (Computer Aided Engineering). Ty nenesou pouze grafickou informaci, ale umí provádět navíc také kontroly bezchybnosti projekčních dokumentů. To však může být nežádoucí při překreslování starých výkresů, které často nevyhovují novějším normám. Mnohdy projektanti nechtějí používat tyto nástroje i z důvodu neslučitelnosti ověřených výpočtových metod s výpočtovými metodami těchto programů. Proto jsou nástroje, které nesou pouze grafickou informaci, stále zapotřebí a ty mohou být vhodným doplňkem CAE systémů.

Hlavním cílem této práce je představit vektorový grafický editor ElectroCAD, který je zaměřen na grafické zpracování elektrotechnické dokumentace. Mimo jiné se práce zabývá novou technologií Windows Presentation Foundation, na níž je program postaven. V druhé kapitole je nastíněna důležitost tvorby elektrotechnické dokumentace, její dělení a předpisy, na které je nutno při návrhu aplikace přihlížet. Jsou uvedeny také důležité pojmy z elektrotechniky. Třetí kapitola se zabývá softwarovým řešením. První část této kapitoly se věnuje blíže použité technologii. Další část pak způsobu řešení důležitých funkcí programu. Čtvrtá kapitola obsahuje uživatelskou

dokumentaci. Ta slouží nejen pro orientaci uživatele programu při návrhu schématu, ale také pro představení všech funkcí programu. V závěrečné části je zmíněno porovnání s dosavadními programy a jsou popsána možná budoucí vylepšení programu. Text je prokládán obrázky pro názornou ukázku popisovaných objektů.

2 ELEKTROTECHNICKÁ DOKUMENTACE

2.1 Požadavky na elektrotechnickou dokumentaci

Dokumentace staveb

Prováděcí vyhláška o dokumentaci staveb uvádí ve svých přílohách jaký musí být obsah a rozsah projektové dokumentace pro ohlášení stavby, pro provádění stavby a rozsah a obsah projektové dokumentace skutečného provedení stavby.

Pro zařízení silnoproudé elektrotechniky jsou zde požadavky na zakreslení silnoproudých rozvodů a zařízení do půdorysů, na výkresovou dokumentaci světelných a napájecích rozvodů včetně zásuvkových okruhů, na schémata rozvaděčů, blokové schéma hlavních napájecích rozvodů či výkresy uzemňovací soustavy.

Pro bleskosvody vyhláška uvádí požadavky na výkresy provedení jímací soustavy, napojení různých kovových dílů a konstrukcí k jímací soustavě, na napojení jímačů na uzemňovací soustavu a na propojení zemničů.

Pro slaboproudá zařízení vyhláška vymezuje požadavky na výkresy pro přehledné zakreslení veškerých zařízení do půdorysů, bloková schémata obsahující počet a logickou polohu jednotlivých koncových prvků.

Pro zařízení obvodů měření a regulace, automatického systému řízení a elektrické požární signalizace jsou požadovány funkční a montážní schémata jednotlivých technologických a funkčních celků a způsob uložení kabelových vedení.

Dokumentace zařízení

Požadavky na dokumentaci zařízení vyplývají také ze zákona č. 22/1997 Sb., který uvádí potřebné doklady nutné pro posouzení shody výrobku s technickými předpisy, na jejímž základě může výrobce nebo dovozce uvést příslušný výrobek na trh.

Elektrická zařízení nízkého napětí musí dle zákona obsahovat koncepční návrh, výrobní výkresy a schémata součástí, podsestav a obvodů, popisy a komentáře nutné pro srozumitelnost výkresů a schémat a funkcí elektrického zařízení.

Dokumentace pro zařízení v provozu musí být podle předpisů vedena, uchovávána a doplňována. Provozní dokumentace musí být uchovávána po celou dobu provozu

zařízení a jakékoliv změny na strojích, technických zařízeních a v technologiích musí být zaznamenány do jejich technické dokumentace. [1]
Z uvedeného přehledu je vidět závažnost tvorby technických výkresů a kreslení schémat, které má své zásady a svá pravidla.

2.2 Základní pravidla pro tvorbu schémat

Elektrotechnická schémata jsou jakýmsi mezinárodním jazykem elektrikářů a je nutno řídit se podle stanovených norem. Normy jsou schvalovány Evropskou komisí pro normalizaci – CENELEC. Členy této komise jsou národní elektrotechnické komitéty Belgie, Dánska, Finska, Francie, Irska, Islandu, Itálie, Lucemburska, Německa, Nizozemska, Norska, Portugalska, Rakouska, Řecka, Spojeného království, Španělska, Švédska a Švýcarska. Česká republika tyto normy přebírá a překládá prostřednictvím technických normalizačních komisí Českého normalizačního institutu.

Základní pojmy a definice při tvorbě technické dokumentace

Obvody

vícepólové znázornění: znázornění, ve kterém je každý spoj vyznačen vlastní čarou

jednopólové znázornění: znázornění, při kterém jsou dva nebo více spojů znázorněny jedinou čarou

Způsoby uspořádání schématu

funkční uspořádání: způsob, při kterém jsou značky komponent umístěny ve schématu tak, aby byly zřetelné jejich funkční souvislosti

topografické uspořádání: způsob, při kterém jsou značky komponent umístěny tak, že jejich vzájemné umístění ve schématu odpovídá fyzickému umístění

Třídění dokumentů

a) dokumenty vyjadřující funkci

přehledové schéma: zobrazuje hlavní vzájemné vztahy nebo spojení uvnitř systému, instalace, zařízení atd., může být provedeno i jednopólově

blokové schéma: přehledové schéma používající převážně blokové značky

mapa sítě: schéma znázorňující síť na mapě, např. elektrárny, transformátorovny, silnoproudá vedení apod.

funkční schéma: schéma, které zobrazuje detaily funkce systému, instalace, zařízení, software

logické schéma: funkční schéma se značkami převážně logických prvků

obvodové schéma: znázorňuje provedení obvodů systému, instalace, zařízení bez zřetele na umístění, fyzické rozměry nebo tvary (viz obr. 2.1)

b) dokumenty o umístění

situační plán: zobrazuje celkový návrh staveniště

montážní výkres: výkres znázorňující rozmístění komponent

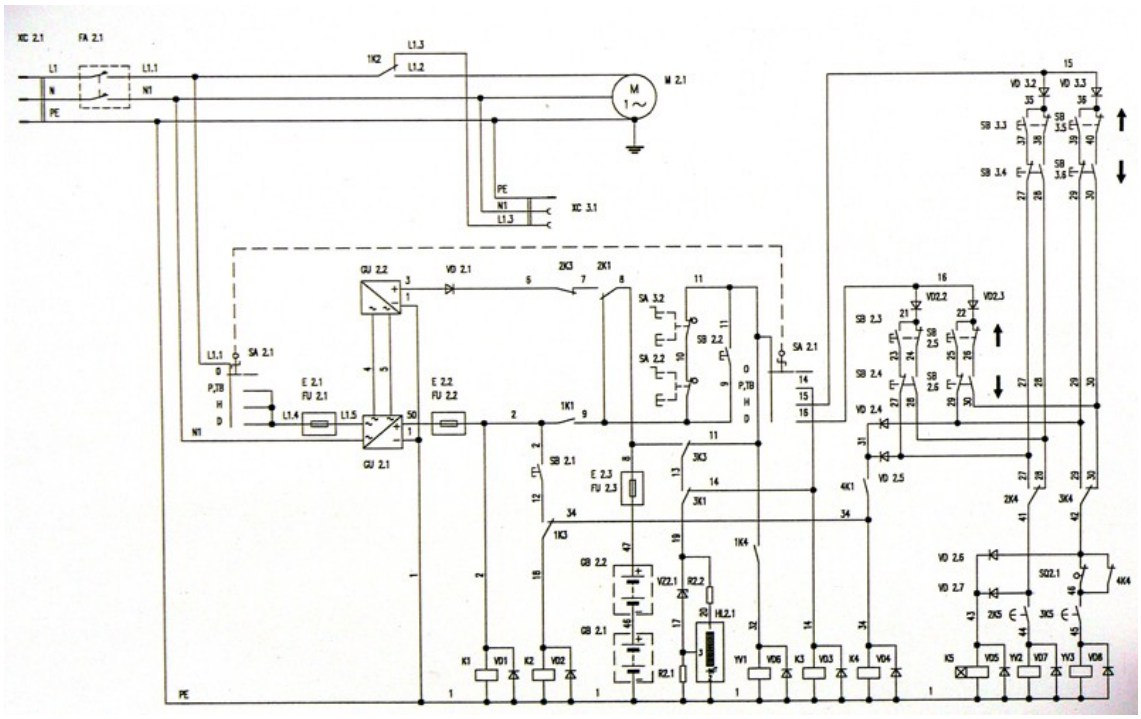
montážní schéma: výkres znázorňující propojení mezi jednotlivými elektrickými předměty

c) dokumenty pro zapojování

zapojovací schéma: schéma znázorňující seznam spojů instalace nebo zařízení

schéma vnitřního zapojení: schéma znázorňující spoje uvnitř konstrukční jednotky

kabelové schéma: schéma podávající informaci o kabelech a vodičích – označení, umístění, zakončení a pod.



Obrázek 2.1 Obvodové schéma

2.3 Elektrotechnické značky

Elektrotechnické značky jsou uvedeny v normě ČSN EN 60617. Všechny značky se zakreslují do mřížky, jejíž rozteč je 2,5mm. Tato velikost však není závazná a značky

mohou být zvětšovány a zmenšovány. Vzájemné poměry značek by se měly při zmenšování a zvětšování zachovávat. Značky se mohou natáčet nebo zobrazovat zrcadlově, jestliže se tímto jejich význam nezmění. Pro značky vodičů se může použít různá tloušťka čáry. Pro zdůraznění jasnosti se značky obvykle uvádějí včetně čar spojů. Pokud není stanoveno jinak, uvedené uspořádání je jen jedním příkladem ze způsobů, kterými se dají kreslit čáry spojů. Doplňková informace může být přidána k většině značek. V systémech CAD je doplňující požadavek, aby každá značka měla referenční bod umístění v průsečíku rastru (mřížky). [2]

Základní názvosloví

Značka – je obrázek, znak nebo písmeno obvykle používané ve schématu pro znázornění předmětu nebo pojmu

Prvek značky – je jednoduchý obrazec s určitým významem, který se musí zkombinovat s jinými obrazci pro vytvoření úplné značky

Všeobecná značka – je obvykle jednoduchá značka společná pro typově příbuzné předměty

Doplňková značka – je značka přidaná k jiné značce pro poskytnutí doplňkové informace

Bloková značka – je jednoduchá značka představující sestavu předmětů a určená pro indikaci funkce soustavy bez udání podrobností o předmětech a bez uvažování všech spojení

3 SOFTWAROVÉ ŘEŠENÍ, PROGRAMOVÁ DOKUMENTACE

Program je napsán v jazyce C# ve vývojovém prostředí Microsoft Visual Studio 2008. Po dlouhém rozhodování a s přihlížením na požadavky na program se stala základem platforma .NET 3.5 s novou technologií Windows Presentation Foundation. Jelikož velká část programu využívá vlastností a předností této technologie, budu se v dalším textu zabývat nejprve popisem této technologie.

3.1 Windows Presentation Foundation

3.1.1 Úvod

Když poprvé vyšel .NET, přinesl s sebou také spoustu nových technologií. Ať už nový způsob ve psaní webu (ASP.NET), tak nový způsob připojování se k databázím (ADO.NET), anebo nové úsporné programovací jazyky (C# a VB .NET) [3]. Jednou z těchto nových technologií byly také Windows Forms. Přestože Windows Forms jsou komplexním nástrojem pro vývoj aplikací, jsou založeny na nezbytných částech Windows, které se během posledních let příliš nezměnily. Pokud chceme vytvořit tlačítko s úplně jiným vzhledem, je nutno naprogramovat vlastní ovládací prvek a vykreslovat každou jeho polohu. Každý ovládací prvek má navíc vlastní oblast, do které může zasahovat. Výsledkem toho je, že není možno vykreslovat např. stín, který zasahuje do jiné oblasti. Windows Presentation Foundation je odpovědí na tyto nedostatky. Obsahuje sadu ovládacích prvků, na které jsme zvyklí i z Windows Forms, ale všechny texty, ohraničení i výplně pozadí vykresluje samostatně. WPF umožňuje programátorovi kontrolovat a měnit způsob, kterým je každá část obrazovky vykreslována. Pomocí těchto vlastností může programátor jednoduše dát ovládacím prvkům nový vzhled a to často i bez psaní kousku kódu.

3.1.2 Výhody Windows Presentation Foundation

Široká integrace

Vývojáři aplikací pro MS Windows, kteří by chtěli použít ve své aplikaci 3D grafiku, video a bohaté možnosti prohlížení dokumentů spolu s 2D grafikou a ovládacími prvky,

by se museli učit několika nezávislým technologiím se spoustou nekonzistentností a pokoušet se smíchat tyto technologie dohromady bez přílišné podpory vývoje. WPF zastřešuje všechny tyto oblasti s důsledným programovým modelem. Je možné aplikovat stejné druhy efektů na různé typy medií. Mnoho technik z jedné oblasti lze využít ve všech ostatních.

Nezávislost na rozlišení

Obyčejně zvýšení rozlišení obrazovky znamená většinou také to, že se aplikace stane příliš malá, texty nečitelné a grafické prvky nerozeznatelné. Díky WPF je možné vyvinout grafické rozhraní, které vypadá smysluplně jak na obrazovce mobilu, tak na padesátipalcové televizní obrazovce. WPF umožňuje programátorům zvětšovat či zmenšovat prvky na obrazovce nezávisle na rozlišení obrazovky. Mnohé z tohoto je díky tomu, že WPF klade důraz na vektorovou grafiku.

Hardwarová akcelerace

Přestože je WPF nová technologie, je vystavěna nad Direct3D. Obzvláště obsah ve WPF aplikaci – ať už 2D, 3D, grafický nebo textový – je konvertován do 3D trojúhelníků, textur a ostatních Direct3D objektů a následně vykreslován hardwarově. To znamená (na rozdíl od systémů založených na GDI), že WPF aplikace těží z výhod hardwarové akcelerace jako např. vyhlazování a samozřejmě lepší výkon (díky práci procesorům grafické karty GPU a tudíž menšího zatížení hlavního procesoru CPU) [4]. Toto také zajišťuje, že všechny WPF aplikace – nejen náročné hry – dostanou maximum výhod nového hardwaru a ovladačů, které se typicky zaměřují na způsobilost pro 3D. Neznamená to ale, že WPF potřebuje pro svůj běh dokonalý hardware; WPF zahrnuje také softwarové vykreslování. To umožňuje provozovat nové vlastnosti WPF i bez hardwarové podpory.

Deklarativní programování

Po více než dvacet let Win16/Win32 programy používaly deklarativní zdrojové skripty k definici rozložení dialogů a menu [4]. WPF dostává deklarativní programování na další úroveň se zavedením nového rozšiřitelného aplikačního jazyku XAML. Kombinace WPF a XAMLu je podobné používání HTML pro definování uživatelského rozhraní s neuvěřitelnou výmluvností. Tato výmluvnost rozšiřuje hranice pro definování uživatelského rozhraní. WPF totiž využívá XAML i pro reprezentaci 3D modelů a dalších. Výsledek toho je, že grafickým designérům je dána větší moc přímo se podílet

na vzhledu aplikace a taktéž na chování ovládacích prvků, kde už se obvykle očekává psaní kódu pro obsluhu událostí.

Bohaté možnosti pro nastavení a skladby uživatelského rozhraní

Ovládací prvky ve WPF jsou velmi jednoduché pro skladbu. Je možné vytvořit ComboBox, jehož obsahem jsou animovaná tlačítka nebo menu vyplněné videoklipy. Tyto úpravy mohou znít dost děsivě, ale důležité je, že není potřeba psát tuny kódu k docílení něčeho takového, co si programátor dříve neuměl ani představit. WPF také ulehčuje možnost, jak dávat aplikacím různé skiny a styly.

Jednoduché uvedení do provozu

WPF poskytuje možnosti, jak jednoduše zprovoznit aplikaci (např. za použití Windows Installer nebo ClickOnce) nebo jak hostovat aplikaci např. ve webovém prohlížeči. Ačkoliv většina této podpory není ve WPF nová a byla již ve Windows Forms, je to pořád důležitá součást technologie. Jeden nový aspekt je, že WPF je postaven nad ClickOnce, což umožňuje přímou integraci WPF aplikace s webovým prohlížečem a jeho navigačním systémem. [4]

Porovnání WPF s minulostí

WPF celkově neposkytuje něco, co se nedalo udělat za použití asembleru. Otázkou je ale, kolik snahy je nutno vyvinout pro dosažení požadovaného cíle. Jestliže se rozhodneme vytvořit aplikaci, která je ekvivalentní WPF aplikaci, musíme se nejen starat o vykreslování pixelů na obrazovku a o interakci vstupních zařízení, ale také o podporu lokalizace, která je již ve WPF vytvořena.

Porovnání s DirectX

WPF poskytuje vysokoúrovňovou abstrakci nad DirectX. WPF přebírá popis scény a tvarů a snaží se o nejlepší způsob, jak tyto objekty vykreslit pomocí dostupných hardwarových zdrojů [4]. Neznamená to však, že by bylo DirectX mrtvé. DirectX je pořád v jistých případech vhodnější technologie než WPF pro pokročilé vývojáře, kteří potřebují pro vykreslování složitých 3D těles maximální výkon.

Nevýhoda volby DirectX oproti WPF je obrovský vzrůst náročnosti vývoje. K této náročnosti důležitou částí přispívá nutnost testovat aplikaci na každé kombinaci driveru (popř. GPU), pro kterou chce programátor v aplikaci vybudovat podporu. Toto testování je již ve WPF implementováno a vývojář se tak může soustředit na jiné důležité záležitosti. WPF aplikace mohou dokonce ovlivňovat GPU i skrz vzdálenou plochu. Je

nutné také poznamenat, že DirectX může být navíc použito i zároveň s WPF v jedné aplikaci.

Porovnání s Windows Forms

WPF se celkově spíše hodí na multimediální aplikace. Občas vládne názor, že Windows Forms je technologie, která se hodí pro aplikace v oblasti podnikání nejlépe. Tento názor ale pochází spíše z dob prvních verzí WPF, kde ještě chyběla spousta ovládacích prvků, jako například TreeView, ListView nebo třeba OpenFileDialog. Tyto chybějící části dělaly vývoj s WPF složitější než při použití Windows Forms.

Windows Forms mají dodnes některé užitečné ovládací prvky, které ve WPF stále chybí. Přesto má WPF přesvědčivé vlastnosti, které stojí za to využít (jako např. bohatá podpora budování layoutu aplikace či nezávislost na rozlišení). Takže pokud uživatel nepožaduje běh aplikace na Windows 98, kde WPF není podporováno a Windows Forms ano, pak WPF není určitě špatnou volbou. Poznamenejme také, že Microsoft investuje více do WPF než do Windows Forms. WPF tak má být prezentační platformou budoucnosti.

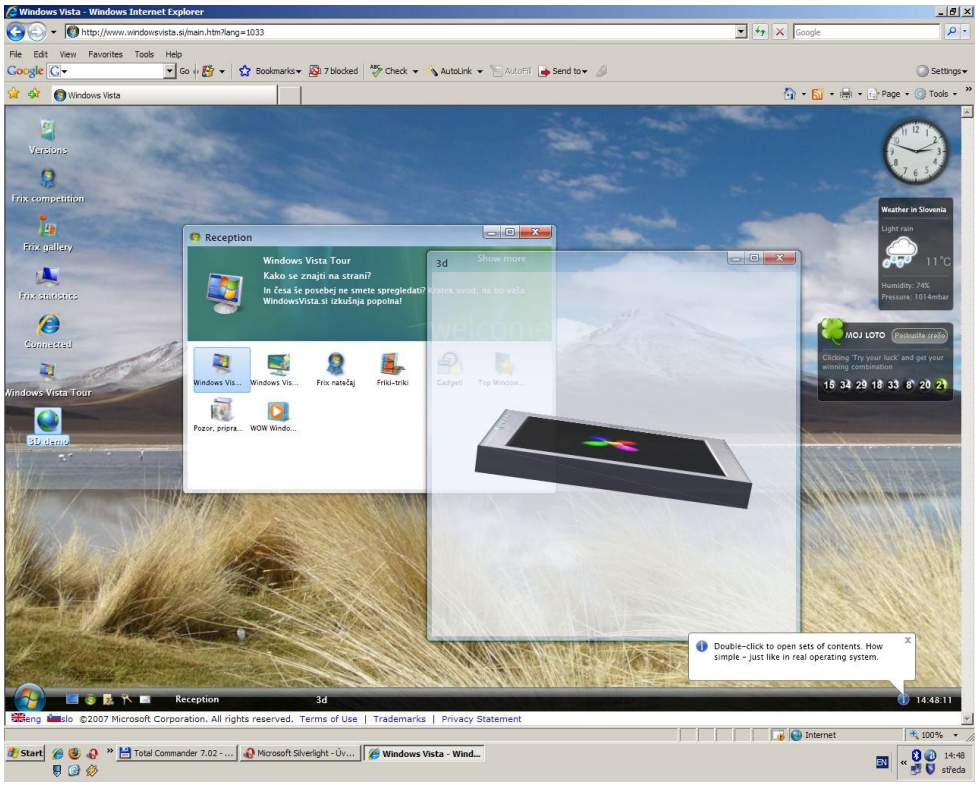
Porovnání s Adobe Flash

Flash je asi nejrozšířenější technologie pro budování bohatého obsahu webů. Pokud se rozhodneme vložit flashový obsah do webových stránek, můžeme předpokládat, že návštěvník již má ve svém počítači instalovaný potřebný přehrávač. Pokud ne, tak si ho lehce a rychle stáhne. Aplikace WPF taktéž mohou běžet v okně prohlížeče. WPF má výhodu v lepších nástrojích pro vývoj, v bohatší sadě prvků, v široké podpoře programovacího jazyka a plném přístupu k podložní platformě. Přesto prohlížení WPF obsahu na webu potřebuje platformu Windows a .NET Framework 3.0 nebo vyšší.

Aby Microsoft překonal závislost na platformě, vydal Windows Presentation Foundation Everywhere (WPF/E) – dnes známé pod názvem SilverLight. V době psaní této práce byla vydána druhá verze SilverLight 2.0. Kromě jazyků C# a Visual Basic podporuje navíc XAML a Javascript. To vše je nezávislé na platformě. Díky možnosti přehrávat videa až v HD kvalitě přes celou obrazovku, je SilverLight jasně vpředu, co se streamování týče. Do budoucna budou tyto produkty jistě ještě několik roků existovat vedle sebe a budou si navzájem velmi konkurovat. Použití lze vidět na obrázcích 3.1 a 3.2.

Obrázek 3.1 Použití technologie Silverlight na webových stránkách

http://www.mediapreview.tv/



Obrázek 3.2 Simulátor Windows Vista na http://www.windowsvista.si/main.htm

WPF na starších verzích Windows

Na Windows Vista je WPF preinstalováno. Kromě Vist je ale také možné provozovat WPF na Windows XP SP2 a Windows Server 2003. Ovšem dvě vlastnosti nebylo možné provozovat na starších verzích Windows:

- 3D objekty jsou anti-aliasovány jen na Windows Vista nebo pozdějších verzích
- okna, která nemají obdélníkový tvar, jsou akcelerována jen na Windows Vista nebo pozdějších verzích

Samozřejmě ovládací prvky mají jiná defaultní témata, která odpovídají tématu operačního systému. Jedním významným vylepšením ve Windows Vista, jež pomáhá WPF aplikacím, je nový model ovladačů. Virtualizuje seznam akcí jednotek GPU a pomáhá systému pracovat výkonněji v době, kdy několik běžících programů využívá zároveň jednotky GPU. Běh více WPF nebo DirectX aplikací může způsobit zamrznutí systému Windows XP. Tento případ není ale již možný ve Windows Vista.

3.1.3 Závěr o WPF

WPF technologie se snaží zkombinovat dobré vlastnosti z různých oblastí, jako jsou např. DirectX (3D a hardwarová akcelerace), Windows Forms (vývojářská produktivnost), Adobe Flash (mocný nástroj pro animace) a HTML (deklarativní značkování a jednoduchost) [4]. První verze WPF udělala velký pokrok v uvědomění si tohoto cíle, ačkoliv samozřejmě nebyla dokonalá. Můžeme narazit na stránky výkonu (pomalý start aplikace, určité hardwarově nepodporované efekty) nebo na oblasti, kde všechny znaky WPF nejsou dořešeny (jako třeba 3D, audio a video). Celkově je ale WPF mocný nástroj, který zvyšuje možnosti a výkonnost při tvorbě nových aplikací.

3.2 Základní struktura programu

Struktura složky, ze které se program spouští, je velmi jednoduchá. Obsahuje spouštěcí soubor, soubor s nápovědou a složku *Icons* s ikonami, které jsou v programu použity. Dále pak složku *Examples* s příklady výstupů a *Library*, která obsahuje soubory s předdefinovanými součástkami. Každý soubor zastupuje jednu součástku a každá složka přísluší jedné kategorii, popř. podkategorii. Součástky jsou standardně typu *ecom*, výkres typu *ecad* a soubor s projektem pak typu *eprj*. Struktura těchto souborů je popsána v části „Ukládání a otevírání.“ Rozšíření knihovny se tedy děje pomocí nahrání

dalších výkresů součástek (soubory typu *ecom*) do složky *Library*, resp. do některé z podsložek (podkategorií). Jelikož WPF ve verzi 3.5 doposud nepodporuje MDI aplikace, je otevření více dokumentů najednou řešeno pomocí panelu záložek.

Při spuštění programu je vytvořena instance třídy *WndMain*, která reprezentuje hlavní okno programu. Při startu jsou kontrolovány parametry předané programu a popř. provedena příslušná obslužná akce (např. otevření výkresu předaného pomocí příslušných parametrů). To proto, aby program mohl při instalaci zaregistrovat přípony souborů, které přísluší k programu a následně je pak přímo otevírat.

Spolu s hlavním oknem je vytvořena schránka, tedy instance třídy *Gr_Clipboard,* pro kopírování a vkládání grafických objektů. Pro načtení knihovny součástek je vytvořena třída *CompTree*, která je potomkem třídy *TreeView*. Ta má při vytvoření za úkol vyhledat úložiště součástek a načíst hlavní kategorie.

Třída hlavního okna si také udržuje položku openedProject_, která je typu *Project*. Tato třída udržuje seznam výkresů náležících k danému projektu a obsahuje metody pro manipulaci s nimi. Kromě projektu obsahuje také seznam otevřených dokumentů *List<Document>*.

Hlavní třídou programu je třída *Document*, jejímu popisu je věnována další část. Slouží pro reprezentaci výkresů jak součástky tak schématu. Je vytvořena při startu programu nebo při vytvoření či otevření nového výkresu a obsluhuje akce kreslení. Pro kreslení se používají grafické třídy, které dědí od předka *Gr_Object* a jsou rozebrány podrobněji v dalším textu.

Další důležitou částí je třída *TextFontInputDialog*. Má standardní chování FontDialogu a používá se při vkládání textu do výkresu. Oproti obyčejnému FontDialogu navíc umožňuje zadávat text, jehož velikost přepočítává v závislosti na zoomu výkresu. Pro uchování vybraného fontu a jeho vlastností používá třídu *FontChoice*.

Třídy *WndNewProject*, *WndPrintSet* a *WndSettings* reprezentují další okna programu a to pro nastavování parametrů nového projektu, tisku a velikosti krokování. *InputBox* je třída zastupující univerzální okno používané pro zadávání vstupů od uživatele.

Třídy *UndoRedoCommand* a *URChange* slouží pro obsluhu akcí undo a redo. Jejich funkce jsou podrobně rozebrány v části „Undo a Redo“. Další třídy jako např. *FontFamilyComboBoxItem*, *ColorComboBoxItem*, *TreeViewItemWithIcon* jsou rozšířením tříd pro zobrazování položek TreeView či ComboBoxů. Za zmínku stojí ještě statická třída *Gr_Settings*, která udržuje nastavení celé aplikace.

3.3 Vnitřní struktura

3.3.1 Reprezentace výkresu

Stěžejní třídou je třída *Document*, která je potomkem *System.Windows.Controls.Canvas* a zastupuje jeden výkres (ať už součástky nebo schématu). Pomocí konstruktoru třídy *Document* je výkres napojen na několik objektů:

Informační Textblock – vypisuje nápovědu pro aktuálně prováděnou akci

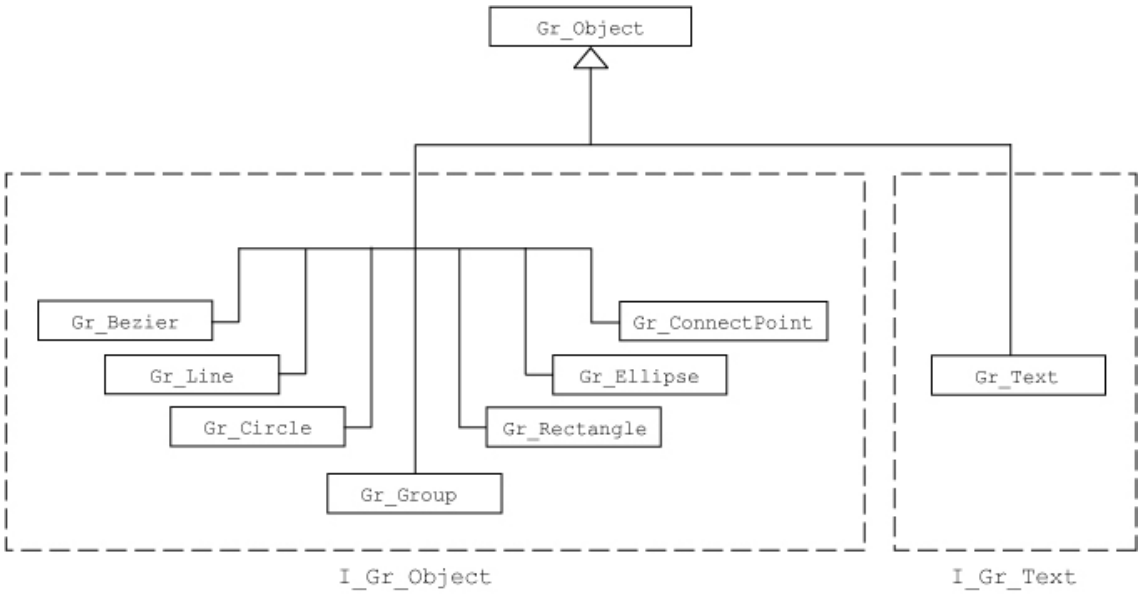
TextBlock souřadnic – používá dokument při zobrazování aktuálních souřadnic na výkresu

ComboBox barev, tloušťky čáry a stylu čáry – při výběru některého grafického prvku zajistí třída *Document* přednastavení těchto ComboBoxů v závislosti na vybraném objektu

Třída v sobě nese *List<Layer>*. Třída *Layer* (hladina) slouží pro uchovávání nakreslených grafických objektů. V programu je doposud použita jen jediná hladina pro každý dokument. Je implementována zvláště kvůli budoucímu rozšíření programu o možnost správy grafických objektů ve vrstvách – tedy o zamykání, schovávání vrstev či změně vlastností všech prvků příslušné hladiny.

3.3.2 Grafické prvky

Základem grafických prvků je třída *Gr_Object*, od které dědí všechny třídy reprezentující grafické prvky. Jsou to *Gr_Bezier* pro bézierovu křivku, *Gr_Circle* pro kružnici, *Gr_Ellipse* pro elipsu, *Gr_Line* pro čáru, *Gr_Rectangle* pro obdélník, *Gr_Text* pro text a *Gr_ConnectPoint* pro reprezentaci přípojného bodu. Od třídy *Gr_Object* dědí navíc také třída *Gr_Group*, která zastupuje skupinu grafických prvků. Je implementována tak, že uvnitř obsahuje *List<Gr_Object>* a jednotlivé metody volá pro každý prvek v tomto listu. Tato třída je použita pro implementaci součástky a obsahuje navíc některé metody pro správu skupiny prvků. Návrh grafických tříd je znázorněn na obr. 3.3.



Obrázek 3.3 Diagram grafických tříd programu

Všechny grafické třídy mají díky dědičnosti od *Gr_Object* předepsanou sadu funkcí. Hlavní z nich jsou funkce (jsou uvedeny bez vstupních parametrů):

`addToCanvas()`

`removeFromCanvas()`

Funkce pro umístění grafického prvku na plátno nebo jeho odebrání z plátna. Je nutno si dát pozor, zda daný prvek už není na daném plátně nebo není logickým potomkem jiného elementu. V takových případech by program skončil chybou.

`redraw()`

Je funkce pro překreslení grafického prvku na plátně. Je využita při zoomování a posunu celého obsahu plátna pomocí nástroje *Ruka*. Zoomování je popsáno podrobněji v dalším textu. Nástroj *Ruka* funguje velice jednoduše. Je posunováno pouze s umístěním offsetu. Jelikož všechny grafické objekty mají souřadnice relativní k tomuto bodu, jsou při volání této funkce `redraw()` všechny překresleny v nové poloze a uživateli se jeví, jako by posouval celým výkresem.

`select()`

`deselect()`

Slouží k výběru daného grafického prvku. *Gr_Select_Type* je výčtový typ určující vzhled vybraného prvku – gstNone (všechny čáry vybraného prvku jsou přerušované) a gstDefault (oproti předchozímu jsou navíc doplněny čtverce označující významné body grafického prvku).

`isClose()`

Je funkce, která určuje, zda leží daný grafický prvek v blízkosti zadaného bodu. Používá se při výběru prvku, kdy je kontrolován každý grafický prvek plátna. Pokud některý z těchto prvků vrátí jako výsledek true, je kontrola pozastavena a na tomto grafickém prvku je zavolána funkce `select()`.

`getCatchPoint()`

Je funkce, která kontroluje takzvané uchopovací body. Podle výčtového typu *WhatToCatch* je zadáno, které body reagují na uchopování. Funkce vrací true, jestliže se daný objekt přichytil ke kurzoru myši, dále pak odkaz na sebe a typ uchopovacího bodu (střed čáry, konec čáry, nejbližší bod atd.).

`setPos()`

Nastaví umístění grafického objektu na zadanou pozici tak, aby jeho střed ležel na této pozici. Této funkce se využívá při vkládání součástky do výkresu, kdy je součástka tažena kurzorem myši.

`move(), moveFrom()`

`rotate(), rotateAt(), rotateFrom(), rotateAtFrom()`

`changeFrom()`

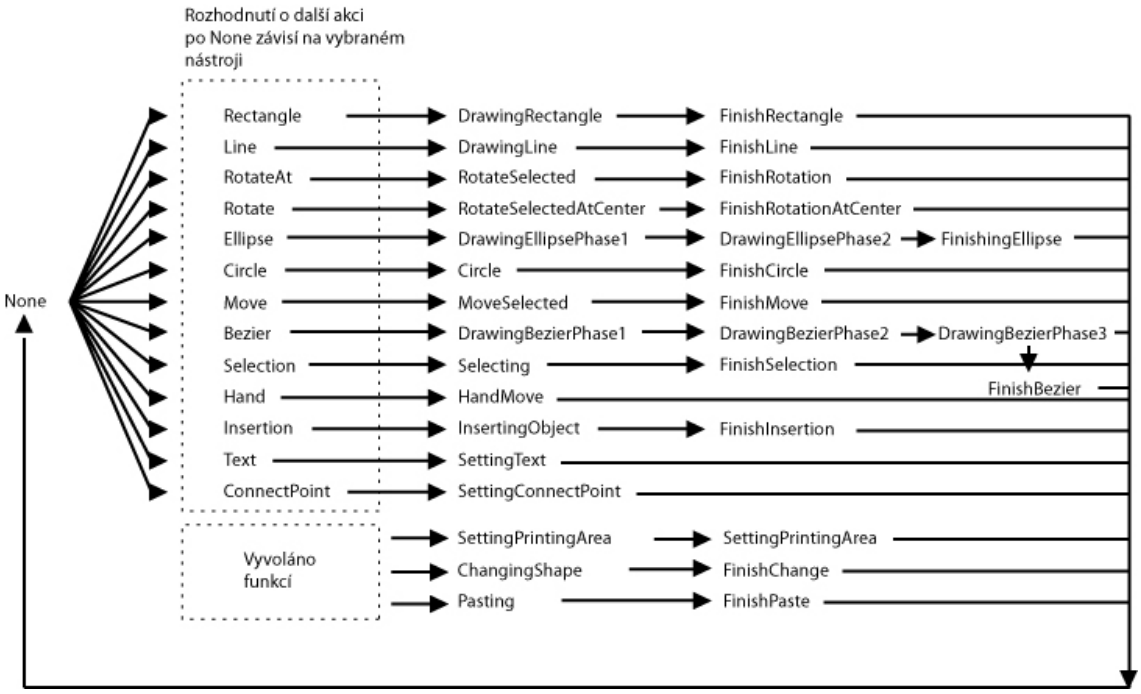
Jsou funkce sloužící pro transformaci grafických objektů. Transformace se při editaci výkresu týkají vždy vybraných prvků. Ty jsou nejprve naklonovány pomocí funkce `Clone()`. U funkcí s přídavkem „From“ je jedním z parametrů objekt, od kterého se má daná transformace provést. Díky tomu má uživatel možnost vidět jak původní podobu objektu tak i jeho transformovanou podobu.

3.4 Editace výkresu

3.4.1 Volby akcí

Při editaci výkresu je nutné správně rozlišovat mezi různými typy akcí (výběr objektu, kreslení objektu, transformace atd.). K tomu slouží dva výčtové typy. První výčtový typ se jmenuje *SelectedToolType* a určuje, který nástroj je právě vybraný. Druhý je *ActionType* a určuje právě probíhající akci. Zvolení správného typu akce zajišťuje funkce `chooseAction()`, jejíž prvním parametrem je předchozí akce a druhým zvolený nástroj. Program přechází mezi jednotlivými stavy v závislosti na probíhající akci.

Výběr akce je znázorněn na obrázku 3.4 Změna akce probíhá vždy při volání funkce `doMouseDown()`, která je volána buď po stisknutí tlačítka myši nad výkresem, nebo po potvrzení správného vstupu z klávesnice.



Obrázek 3.4 Diagram přechodu mezi jednotlivými stavy programu

3.4.2 Transformace zadaného bodu

Uživatel může zadat bod při kreslení pomocí dvou způsobů. Prvním způsobem je určení bodu z klávesnice. Takto zadaný bod je přepočítán do skutečných souřadnic na plátnu. Nejprve je vynásoben velikostí zoomu a pak posunut podle polohy počátku soustavy souřadnic (offsetu). Dále už nepodléhá žádným transformacím. Pokud uživatel nezadá bod ale délku, je tento bod dopočítán pomocí funkce `getPointInDirection()` a následně transformován do souřadnic plátna. Druhý způsob zadání bodu je pomocí kurzoru myši. Takto zadaný bod prochází několika fázemi vyhodnocení:

Určení souřadnice

Nejprve se vyhodnotí souřadnice relativní k aktuálnímu výkresu. Toho je docíleno pomocí vestavěné funkce `GetPosition()`, která se volá na argumentech události spojené s akcí myši (čili např. MouseEventArgs, MouseButtonEventArgs).

Vyhodnocení krokování

Pokud je aktivní volba krokování, pak jsou souřadnice předány funkci `setStepCoordinates()`, která určí nejbližší souřadnice spadající do mřížky určené

velikostí kroku v *x*-ovém a *y*-novém směru. Toho je docíleno tak, že každá souřadnice je vydělena velikostí kroku v příslušném směru. Tím určíme počet kroků od počátku soustavy souřadnic. To je ale většinou reálné číslo a pokud chceme, aby výsledná souřadnice byla celočíselným násobkem kroku, musíme určit nejbližší pozici, která takovým celočíselným násobkem je. Proto vezmeme dolní a horní celou část z počtu kroků, pronásobíme velikostí kroku v daném směru a určíme, která z nově vypočítaných souřadnic je nejbližší k původní.

Vyhodnocení pravoúhlého kreslení

Při aktivní volbě pravoúhlé kreslení a v závislosti na probíhající akci jsou souřadnice přepočítány pomocí funkce `getOrthoCoord()`. Tato funkce je volána jen ve fázích, kdy již byl zadán bod, od kterého by mohly být nové souřadnice spočítány. Nejdříve je spočítána horizontální a vertikální vzdálenost aktuálních souřadnic od zadaného bodu. Pokud je vertikální vzdálenost menší než horizontální, je vrácen nový bod. Tomu je ponechána *x*-ová souřadnice, ale *y*-ová souřadnice je nastavena na *y*-ovou souřadnici prvního zadaného bodu. V opačném případě je mu ponechána *y*-ová souřadnice a změněna *x*-ová.

3.4.3 Uchopování objektů

Hlavní třídou v implementaci uchopování je třída *Gr_CatchPoint*. Každý otevřený dokument si udržuje jednu instanci této třídy. Tato třída v sobě nese typ uchopovacího bodu, který určuje, k jaké části objektu byl kurzor myši přichycen. Tento typ je uchován pomocí výčtového typu *CatchPointType*. Třída *Gr_CatchPoint* navíc v sobě nese referenci na objekt, který zareagoval na přichycení a souřadnice uchyceného bodu. Při pohybu myši je volána událost, ve které se vždy volá funkce `checkCatchPoints()`. V této funkci jsou procházeny všechny grafické objekty a na nich je volána funkce `getCatchPoint()`. Ta mimo jiné vrací true, pokud objekt reaguje na přichycení a tím je kontrola objektů ukončena. Uchopovací bod je zobrazen pomocí funkce `show()` třídy *Gr_CatchPoint* a výsledná souřadnice je přepsána souřadnicí uchyceného bodu.

3.4.4 Zoomování

Ve WPF jsou implementovány transformace pro zoomavání (pomocí třídy *ScaleTransform*). Ty však nebylo možno použít, jelikož při této transformaci jsou

zároveň modifikovány i tloušťky čar zvětšovaných objektů. To je však pro aplikaci s přesným rýsováním poněkud nežádoucí.

Zoomování je tedy implementováno samostatně. Každý grafický prvek obsahuje jednak své skutečné souřadnice na plátně a jednak své reálné souřadnice. Reálné souřadnice se vztahují k bodu offset. Ten značí počátek soustavy souřadnic a všechny grafické prvky mají své reálné souřadnice relativní vůči tomuto počátku. Při zoomavání záleží na poloze kurzoru na plátně. Aby bylo zajištěno, že se prvky zvětšují kolem zoomavaného bodu, je nejprve vypočítán pohyb offsetu (počátku soustavy souřadnic). Následně jsou vynásobeny reálné souřadnice každého grafického prvku velikostí zoomu a posunuty v závislosti na pohybu offsetu.

3.4.5 Vykreslování grafických prvků

WPF používá nový druh Properties takzvané Dependency Properties. Ty závisí v jednom okamžiku na více poskytovatelích. Tito poskytovatelé mohou např. pomocí animace průběžně měnit svoji hodnotu. Největší přínos Dependency Properties spočívá v možnosti oznámit změnu své hodnoty. Většina Properties ve WPF je Dependency. Ty mohou být měněny přímo z XAMLu bez použití procedurálního kódu. Kdykoliv se změní hodnota Dependency Property, WPF může automaticky spustit několik obslužných metod, které závisí na hodnotě metadat dané Property. Jednou z takovýchto metod může být například překreslení příslušného objektu.

Díky tomu není nutné se starat o překreslování jednotlivých grafických objektů. K vykreslení grafického prvku s novými rozměry, novou pozicí, barvou či stylem stačí pouze změnit příslušné Dependency Property. V programu tedy není potřeba implementovat žádnou funkci na re-rendering. Každý *Gr_Object* v sobě nese příslušný objekt z jmenného prostoru *System.Windows.Shapes* (např. *Ellipse*) a tomuto objektu jsou měněny jeho Dependency Properties.

3.4.6 Rozpoznání klávesových zkratek a příkazů

Tak jako WPF přináší více infrastruktury do Properties, přináší více infrastruktury i do systému událostí. Objevuje se nový typ událostí, tzv. Routet Events. Uživatelské rozhraní je ve WPF konstruováno pomocí takzvaných logických a vizuálních stromů objektů. Routet Events jsou navrženy pro fungování na těchto stromech. Pokud je tedy vyvolána tato událost, může putovat tímto stromem podle tří strategií – probublávání (událost je vyvolána nejprve na zdroji, pak na každém prvku směrem vzhůru ve stromě

dokud nedoputuje ke kořenu v logickém stromu), tunelování (událost je vyvolána na kořenu, pak na každém prvku směrem dolů ve stromě dokud nenalezne zdroj) a přímá strategie (událost je vyvolána pouze na zdroji – téměř stejné jako u běžných událostí, ale mohou se zapojovat do dalších mechanismů, jež jsou spojeny s Routet Events). Většina událostí je prvního typu, ale často jsou párovány s událostmi druhého typu. Ty mají podle konvence předponu „Preview“ a jsou ve WPF spouštěny těsně před svými partnery.

Toho je využito pro rozeznávání, zda uživatel použil klávesovou zkratku, či zda chtěl zadat příkaz nebo hodnotu z klávesnice. Nejprve je spuštěna událost PreviewKeyDown, kdy program vyhodnotí, jestli daná klávesa neoznačuje rychlou volbu nástroje. Pokud ano, je nastaven příznak události *Handled* na true. To označuje, že daná událost již byla zpracována a není nutné ji propagovat dále. Událost KeyDown se v takovémto případě vůbec nespustí. Tím odpadá nutnost opravy textu ve vstupním *TextBoxu*. Naopak pokud klávesa nepřísluší žádné klávesové zkratce, je událost propuštěna dál. Po vyvolání události PreviewKeyDown se vyvolá i událost KeyDown a dojde k zapsání písmena do vstupního TextBoxu.

Po vepsání příkazu a následného potvrzení klávesou Enter je zavolána funkce `useInput()` třídy *Document*. Je rozpoznán typ příkazu a zavolána jemu příslušející funkce v závislosti na probíhající akci.

3.4.7 Zjištění velikosti UIElementu

WPF má velmi důmyslný layout engine. Všechny elementy ve WPF, které jsou zahrnuty do procesu layoutu aplikace, dědí od třídy *System.Windows.UIElement*. Rodičové a potomci, kteří jsou odvozeni od této třídy, navzájem při tomto procesu spolupracují při určování jejich výsledné pozice a rozměrů. Zjištění velikosti se běžně provádí přes Property *ActualWidth* a *ActualHeight*. Problém nastane, když chceme velikost UIElementu zjistit přímo v metodě, kdy byl tento UIElement vytvořen. Pro zjištění velikosti je nutno dát totiž příležitost WPF layout enginu pro přepočítání rozměrů.

K této skutečnosti bylo potřeba přihlédnout při vykreslování rámečku kolem textu. Objekt *Gr_Text* v sobě proto nese pomocné body, které určují, kde se aktuálně rámeček nachází. Při tvorbě textu je asynchronně volán delegát, který se postará o přepočítání *ActualHeight* a *ActualWidth* a do konstruktoru třídy *Gr_Text* jsou předány již správné

hodnoty. Ty jsou pak uloženy do pomocných bodů a díky tomu není nutné volat delegáta při každé změně zoomu znovu.

3.4.8 Kopírování grafických objektů

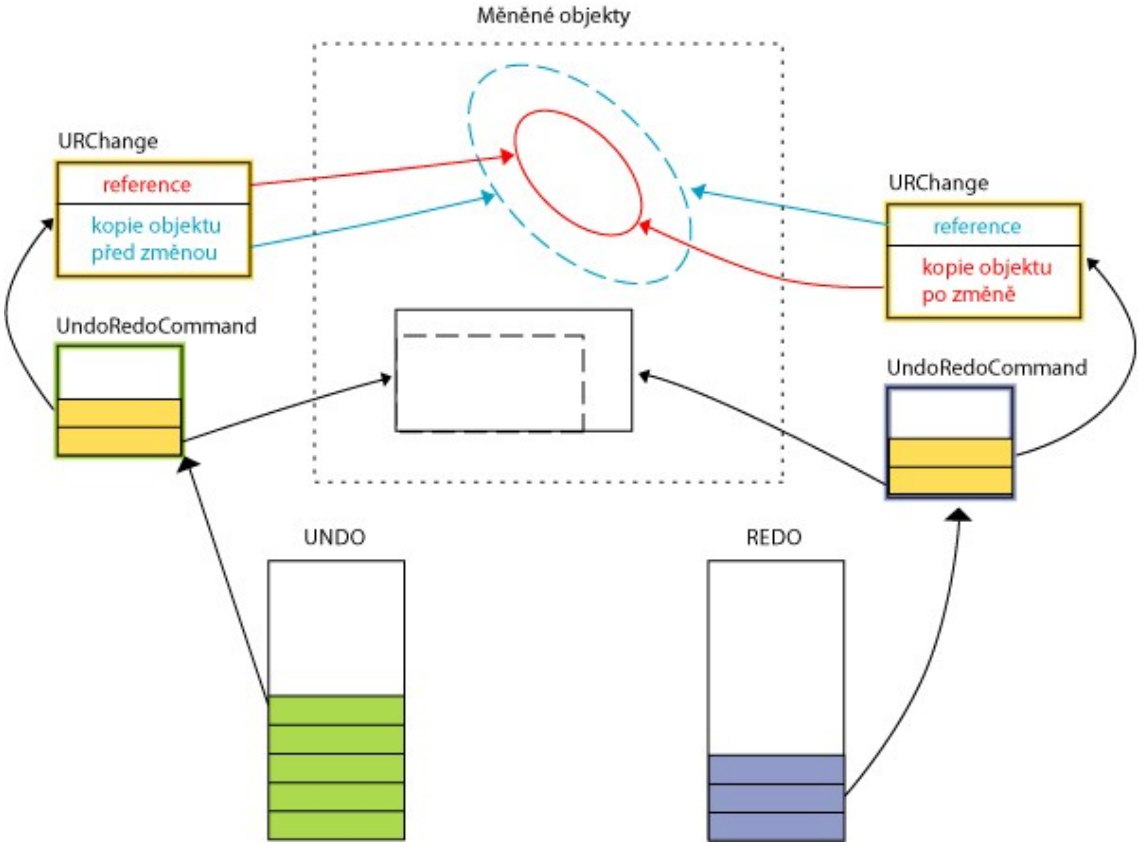
Pro kopírování grafických prvků se využívá třída *Gr_ClipBoard*. Tato třída je instanciována při startu programu. Kopírování se týká vybraných objektů a aktivního dokumentu (při více otevřených dokumentech). Třída *Gr_ClipBoard* nese *List<Gr_Object>*, který je vždy před novým kopírováním vymazán použitím funkce `clear()`. Na každém z vybraných objektů je zavolána funkce `Clone()`, která vytvoří kopii objektu. Tato kopie je přidána do clipboardu funkcí `addItem()`.

3.4.9 Undo a Redo

Možnost vrátit provedené změny vpřed a zpět patří k základním vlastnostem programu. Implementace těchto příkazů je řešena pomocí dvou zásobníků undoStack_ a redoStack_. Ty jsou typu *URStack*, což je zásobník s možností odebírat i z jeho dna. To proto, aby bylo možné omezovat počet vrácených kroků a při přesáhnutí limitu mazat nejstarší provedenou akci ze dna zásobníku. Každá instance třídy *Document,* reprezentující jeden výkres, má své vlastní zásobníky pro undo/redo operace.

Kdykoliv je tedy provedena nějaká akce, která podléhá undo/redo operaci, vytvoří se nová instance třídy *UndoRedoCommand* a je přidána na vrchol zásobníku undoStack_. Tato třída uchovává změny jedné akce. Jelikož program umožňuje transformaci několika grafických prvků zároveň, může být těchto změn i více. Každá změna je instancí třídy *URChange*, která obsahuje referenci na změněný prvek a jeho předchozí stav. Pokud uživatel zažádá o vrácení změny, je z vrcholu zásobníku undoStack_ vybrána položka, na ní zavolána funkce `doUndo()` a přesunuta do zásobníku redoStack_. Funkce `doUndo()` prochází všechny instance třídy *URChange* a na nich volá jejich funkci `doUndo()`, která vrátí změnu jednoho grafického prvku. Existují 3 stavy třídy *URChange*. Prvním je situace, kdy se jedná o změnu již existujícího prvku. V takovém případě je nahrazen grafický prvek, na nějž odkazuje reference, za jeho původní podobu. Druhý stav nastává v případě, kdy byl grafický prvek na plátně vytvořen a neexistuje jeho předchozí podoba. V tomto případě funkce `doUndo()` smaže tento prvek z plátna výkresu. Opačným případem je třetí stav, kdy byl prvek smazán z plátna a funkce `doUndo()` ho naopak vrátí zpět na plátno. Opačně se chová funkce

`doRedo()` sloužící pro obnovení změny. Průběh operace undo a redo je znázorněn na obrázku 3.5.



Obrázek 3.5 Průběh operace undo-redo

3.4.10 Ukládání a otevírání

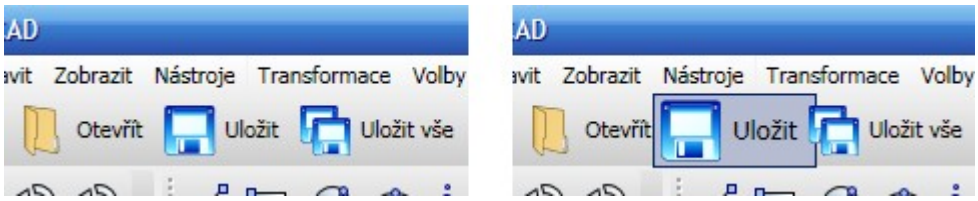
Ukládání dokumentu je implementováno pomocí serializace. Protože je každý grafický prvek serializovatelný (každý potomek *Gr_Objekt* implementuje rozhraní *ISerializable*), stačí uložit všechny grafické objekty daného dokumentu do souboru či streamu. Při serializaci jsou navíc uloženy i další informace. Spolu s grafickými objekty je do souboru zaserializována také instance třídy *DocumentInfo*, která v sobě nese typ dokumentu určený pomocí výčtového typu *DocumentType* a dále pak jméno autora a verzi dokumentu. Pomocí toho může být při otevírání jednoduše rozpoznáno, zda se jedná o výkres součástky, schématu nebo jiný typ souboru. Soubor součástky se totiž liší od výkresu právě příznakem *DocumentType* nastaveným na hodnotu Component (výkres má tuto hodnotu nastavenou na Drawing). Třída *Document* má více metod pro ukládání i otevírání dokumentu. Dokument může být uložen jak do souboru zadaného pouze cestou (metoda `SaveToFile()`), tak do streamu (metoda `SaveToStream()`). Při otevření i uložení dokumentu je uložena do privátní položky fullName_ cesta úložiště.

Toho se využívá při volání funkce `Save()`, která uloží dokument do stejného místa, odkud byl otevřen nebo uložen dříve. Třída *Document* obsahuje také Property *IsDirty*, která indikuje, zda proběhly v dokumentu nějaké změny a je potřeba jej znovu uložit. Všechny funkce pro uložení i otevření dokumentu vrací typ bool pro indikaci, zda se daná akce zdařila. Soubor typu *eprj*, tedy soubor zastupující projekt v sobě nese seznam dokumentů, které tomuto výkresu patří. I do tohoto typu souboru se serializuje třída *DocumentInfo*, která má výčtový typ *DocumentType* nastavený na hodnotu Project.

3.5 Vizuální efekty v aplikaci

3.5.1 Animace tlačítek

Pro animace jsou použity třídy ve jmenném prostoru *System.Windows.Media.Animation*, které slouží k popsání a aplikování animace bez nutnosti manuální práce. Animace ve WPF jsou nezávislé na čase. Znamená to, že pokud se zvýší výkon hardwaru, nebude animace rychlejší, ale bude více plynulá [4]. WPF umí pozměňovat frame rate v závislosti na daných podmínkách. Tyto třídy pro animaci avšak mají svá omezení – umožňují měnit pouze jednotlivé Dependency Properties daného prvku.



Obrázek 3.6 Ukázka animovaného tlačítka

Animace tlačítek je vytvořena přímo v XAML kódu pomocí tzv. event triggerů, které umožňují reagovat na vzniklou situaci, v našem případě na události MouseEnter a MouseLeave. Event triggeru jsou předány animace, které se mají při dané události provést. Ty jsou vepsány do tzv. StoryBoardu, který spouští sekvenci animací. Nastavením vlastností AccelerationRatio a DecelarationRatio je ovlivněno zrychlování animace v jejím průběhu. Délka trvání animace je nastavená pomocí parametru Duration. V programu jsou použity dvě animace – pro změnu RenderTranform.ScaleX a RenderTranform.ScaleY. Tato animace je zapsána do stylu s názvem SimpleAnimated a je aplikována na všechna tlačítka toolbarů. Ukázka animace je zobrazena na obrázku 3.6.

3.5.2 Stylování a ikony

Jedna z nejvíce pozitivně hodnocených vlastností WPF je možnost dávat všem objektům radikálně odlišný vzhled. K tomu se ve WPF využívají styly, šablony, skiny a témata. V programu jsou použity převážně styly. Style je reprezentován třídou *System.Windows.Style* a její hlavní funkcí je shromažďovat hodnoty jednotlivých Properties, které jsou pak aplikovány na prvky, kde je daný styl použit [5]. Na prvek může být aplikován styl, který má i více definovaných Properties než daný prvek. Jednoduše se nastaví ty vlastnosti, které jsou obsaženy jak ve stylu tak i v Properties daného prvku. Na styl se odkazuje jménem, které je definováno v atributu x:key. Všem stylům jsou nastavena omezení pomocí vlastnosti TargetType, kde definujeme typ prvku, na který může být použit (např. Buton). Styly je možné také přepisovat. Pro vytvoření výjimečného tlačítka, které se od našeho stylu liší pouze barvou pozadí, lze tuto barvu pozadí nastavit přímo u tohoto tlačítka a tomuto nastavení bude dána přednost před barvou pozadí definovanou ve stylu. Styly jsou umístěny do ResourceDictionary, které jsou následně zahrnuty do Aplication.Resources. Tak je možné se na ně odkazovat jak už z XAMLu či z procedurálního kódu pomocí jména stylu. V programu jsou použity styly ze dvou ResourceDictionary – každý z nich v jenom souboru (SimpleStyles.xaml a Icons_dictionary.xaml). V druhém jmenovaném jsou zahrnuty návrhy jednotlivých ikon toolbarů, popsané pomocí XAMLu. Díky tomuto vektorovému formátu je pak možné libovolně zmenšovat či zvětšovat takovéto ikony bez ztráty kvality vzhledu. Toho si lze všimnout na obr. 3.6.

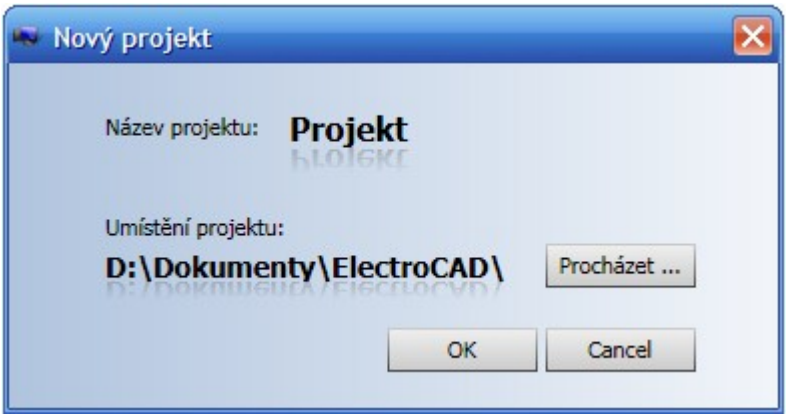
3.5.3 Dockování panelů

K vytvoření dokovatelných panelů je použito tří tabulek tzv. Grid. Hlavní plocha dokumentu obsahuje jednu velkou tabulku a nijak nezahrnuje dva postranní panely. Levý a pravý panel jsou samostatnými tabulkami. Pokud uživatel najede na tlačítko na postranní liště, je zviditelněna tabulka nastavením Property *Visibility* na hodnotu Visible. Naopak při pohybu kurzoru mimo tabulku je nastavena *Visibility* na Collapsed, což má za následek kromě zneviditelnění tabulky také to, že tabulka není zahrnuta do procesu layoutu. Při zadockování panelu je do hlavní tabulky přidán sloupec pomocí třídy *ColumnDefinition*. Ten má spolu s příslušející tabulkou nastavenou vlastnost *SharedSizeGroup* na stejnou hodnotu, což zajistí sdílení rozměrů prostoru. Proto při

změně šířky panelu pomocí *GridSplitteru* je změněna i šířka sloupce v tabulce hlavního obsahu.

3.5.4 Vytvoření efektu odrazu

Při vytváření nového projektu je otevřeno okno, ve kterém se zadává název nového projektu a úložiště. Lze si všimnout, že zadané texty (jméno projektu nebo cesta k úložišti) mají pod sebou viditelný odraz (viz obr. 3.7). Toho je docíleno díky Data Bindingu. Základem Data Bindingu je třída *System.Windows.Data.Binging*. Pomocí této třídy můžeme na sebe napojit dvě Property různých objektů a ty udržovat synchronizované. V programu je tedy zobrazen obdélník pod textem, jehož výplň je napojena na výplň příslušného TextBoxu. V kombinaci s transformací pro obrácení textu a aplikováním průhlednosti je docíleno efektu odrazu. Při změně textu ve vstupním TextBoxu je automaticky díky Data Bindingu změněno i pozadí odrazu a tak vypadá odraz reálně.



Obrázek 3.7 Efekt odrazu v dialogovém okně nového projektu

4 UŽIVATELSKÁ DOKUMENTACE

4.1 Instalace programu

Program ElectroCAD je určen pro systém Windows XP, Windows Vista a vyšší. K běhu programu je zapotřebí instalace .NET Framework 3.5. Pokud tedy na cílovém počítači tato instalace chybí, je nutné Framework nainstalovat nejdříve. Instalační soubor je dodáván spolu s instalačním programem .NET Framework 3.5.

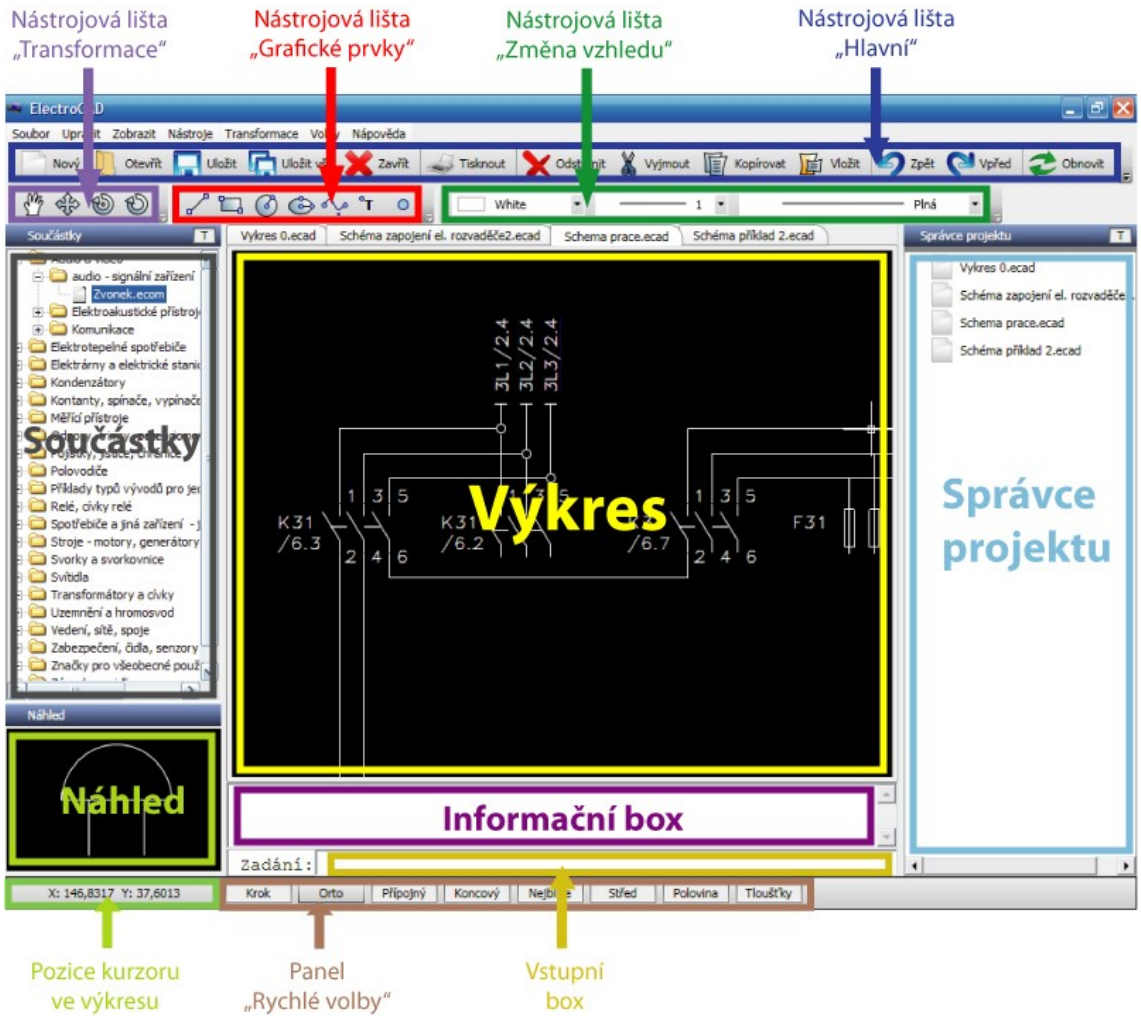
Vlastní program se instaluje spuštěním samoinstalačního balíčku s názvem ElectroCAD_setup.exe. Průvodce instalací vás vyzve k umístění úložiště programu, vytvoří zástupce na ploše a také ve Start menu. Dále registruje nové přípony souborů, které souvisejí s programem.

4.2 Spuštění programu

Program se spouští souborem ElectroCAD.exe, který se nachází v instalačním adresáři programu. Po spuštění se objeví hlavní okno aplikace. Pro rychlou ukázku je v instalačním adresáři složka s názvem *Examples*, kde si můžete prohlédnout možné výstupy programu.

4.3 Popis pracovního prostředí

Pracovní prostředí je rozděleno do několika částí, viz obr. 4.1. V horní části se nachází menu s nástrojovými lištami. Hlavní nástrojová lišta obsahuje volby pro standardní funkce, jako jsou vytvoření nového výkresu, otevření existujícího, kopírování, vkládání, změny vpřed a zpět, tisk apod. Všechna tato tlačítka mají své ekvivalenty také v menu. Menu obsahuje některé volby navíc (např. vytvoření a zavření projektu, volbu *Uložit jako* a další).



Obrázek 4.1 Části hlavního okna aplikace

Nástrojová lišta „Transformace“ (viz obr. 4.2) slouží pro aplikování transformací na vybrané prvky. Pomocí nástrojů této lišty lze s vybranými prvky výkresu otáčet a posouvat. První nástroj této lišty je nástroj *Ruka*. Tento nástroj slouží k posouvání celého výkresu a lze ho volit přidržením klávesy mezerník. Ostatní transformace jsou popsány podrobněji v dalším textu. Všechny jsou opět přístupné i z nabídky menu.

Obrázek 4.2 Nástrojová lišta „Transformace“

Nástrojová lišta „Grafické prvky“ (viz obr. 4.3) slouží k volbě jednotlivých kreslících nástrojů. Můžeme kreslit čáru, obdélník, kružnici, elipsu, bézierovu křivku a tzv. přípojný bod. Způsob kreslení jednotlivých nástrojů je opět podrobněji rozebrán v dalším textu.

Obrázek 4.3 Nástrojová lišta „Grafické prvky“

Nástrojová lišta „Změna vzhledu“ (viz obr. 4.4) slouží ke stylování jednotlivých grafických prvků. Nabídka barev je složena ze všech standardních barev MS Windows. K výběru stylu čáry je k dispozici sada typů čar a předdefinovaných šířek.

Obrázek 4.4 Nástrojová lišta „Změna vzhledu“

V levé části okna programu je panel součástek, kde je možné vybrat z knihovny předdefinovaných součástek a ty následně vkládat do výkresu. Tento panel je dockovatelný, což znamená, že je možno jej schovat pomocí tlačítka v horní části tohoto panelu, a tak zvětšit plochu výkresu. Po výběru součástky je vždy součástka zobrazena v oblasti náhledu. Plocha výkresu se nachází uprostřed okna aplikace. Pozadí výkresu je standardně nastaveno na černou barvu, která neunavuje příliš zrak. Při tisku je invertována černá barva za bílou a naopak. V pravé části okna aplikace je opět dockovatelný panel „Správce projektu.“ Zde jsou zobrazeny aktuálně otevřené výkresy nebo výkresy otevřeného projektu.

Pod oblastí výkresu se nachází informační box a vstupní box. Do informačního boxu program vepisuje aktuálně prováděné akce a nápovědu pro uživatele. Do vstupního boxu se zadávají body, délky a příkazy. Vstup z klávesnice je podrobněji rozebrán v dalším textu.

Ve spodní části programu je zobrazována aktuální pozice kurzoru ve výkresu. Ta záleží na posunu výkresu a zoomu. Vedle tohoto informačního textu je panel „Rychlé volby.“ Pomocí tohoto panelu je možné zapínat a vypínat pomocné funkce při kreslení. Tyto volby jsou dostupné i z menu programu.

4.4 Začátek práce s programem

Po spuštění programu máte na výběr z několika možností:

4.4.1 Vytvoření nového projektu

Pokud chcete vytvořit novou elektrotechnickou dokumentaci, která bude obsahovat více výkresů, zvolte *Soubor – Nový – Projekt*. Zadejte cílovou složku a název projektu. Aktivní může být vždy pouze jeden projekt. Před otevřením či vytvořením nového projektu jsou vždy uzavřeny všechny výkresy. Pokud je některý výkres pozměněn, dotáže se program před zavřením na jeho uložení.

4.4.2 Vytvoření nového výkresu

Pro vytvoření samostatného výkresu zvolte v menu *Soubor – Nový – Výkres* nebo použijte tlačítko *Nový výkres* na hlavním panelu nástrojů. Pokud je otevřen projekt, bude tento výkres zařazen do aktuálního projektu.

4.4.3 Vytvoření nové součástky a zařazení do knihovny

Knihovnu součástek můžete libovolně rozšiřovat, popř. staré součástky nahrazovat novými podle nejnovějších norem. V menu zvolte *Soubor – Nový – Součástka*. Vytvořenou součástku ukládejte do adresáře *Library* do příslušné kategorie. Pokud budete chtít vytvořit novou kategorii, vytvořte v adresáři *Library* novou složku a součástku umístěte do ní. Každá složka zastupuje jednu kategorii. Struktura složek tedy odpovídá struktuře kategorií a podkategorií zobrazených v knihovně. Pro načtení nové součástky do panelu součástek musíte znovu obnovit strom součástek. To provedete pomocí příkazu *Zobrazit – Znovu načíst součástky* nebo tlačítkem *Obnovit* na hlavní nástrojové liště nebo pomocí klávesové zkratky F5.

4.5 Editace schématu a součástky

4.5.1 Rozdíl mezi schématem a součástkou

Editace součástky je téměř stejná jako editace výkresu. Hlavní rozdíl je v umístění počátečního bodu. Při návrhu součástky je vždy zobrazen pomocný kříž, který určuje počáteční bod (střed součástky). Tento středový bod je důležitý při vkládání součástky do výkresu, kdy je tažena za tento střed. Je vhodné, aby přípojné body byly pokud možno umístěny na osách procházející počátkem. Při krokování pak budou tyto přípojné body ležet v mřížce, která je určená *x*-ovou a *y*-ovou velikostí kroku.

U výkresu je počáteční bod umístěn v levém horním rohu obrazovky. Jeho poloha se však při zoomování a posouvání mění. Všechny objekty mají relativní souřadnice právě k tomuto bodu.

4.5.2 Kreslení grafických prvků

Čára

Po zvolení nástroje *Čára* je nutno určit umístění počátečního bodu čáry. Tento bod lze zadat kliknutím levého tlačítka myši nebo pomocí zadání bodu z klávesnice. V následující fázi máte několik možností. První z nich je určení druhého bodu čáry pomocí kliknutí či zadání z klávesnice. Další možností je určení směru čáry pomocí tažení myši a následného zadání délky čáry z klávesnice. Program automaticky vypočítá druhý bod čáry tak, aby ležel ve zvoleném směru a úsečka měla požadovanou délku.

Obdélník

První fáze je stejná jako při kreslení čáry. Určíte první roh obdélníku. Druhý bod můžete opět zadat pomocí tlačítka myši či vstupem z klávesnice. Je ale možné také pomocí tahu myši určit směr úhlopříčky (a tak i poměr stran) a následně zadat její délku. Program dopočítá velikost strany *a* a *b* obdélníku tak, aby měl požadované vlastnosti.

Kružnice

Při zvolení nástroje *Kružnice* nejprve určujete její střed ať už z klávesnice nebo kliknutím. Poté zadáte poloměr a to buď vytažením kružnice do požadované velikosti, nebo zadáním délky poloměru z klávesnice.

Elipsa

Kreslení elipsy je rozděleno do tří fází. V první fázi určujete levý bod hlavní poloosy elipsy. Ve druhé fázi určujete šířku elipsy a její natočení. Šířku opět můžete zadat pomocí klávesnice. V další fázi uvedete velikost vedlejší poloosy. Je to vzdálenost od středu elipsy po aktuální bod. Tuto velikost lze opět zadat z klávesnice.

Bézierova křivka

Používá se pro kreslení hladkých křivek. Určuje se zadáním 4 bodů. První dva z nich určují počáteční a koncový bod křivky. Další dva fungují jako jakési magnety a určují, kterými směry bude křivka vychýlena. Všechny body a také vzdálenost počátečního bodu a koncového bodu lze zadat z klávesnice.

Text

Pro vepsání textu do výkresu nejprve určíte levý horní okraj umístění textu. Po zadání tohoto bodu bude otevřeno okno, kde zvolíte font, styl, velikost a barvu textu. Vepíšete požadovaný i víceřádkový text. Velikost textu v dialogovém okně je vždy počítána podle aktuálního zoomu a zadané velikosti. Text tedy bude mít po potvrzení tlačítkem OK stejnou velikost, jaká je zobrazena v dialogovém okně. Pro rychlé stisknutí tlačítka OK je možno využít klávesovou zkratku Ctrl+Enter. Upravení textu se provádí dvojklikem na již vytvořený text. Bude znovu otevřeno okno stejné jako při zadávání nového textu s předvyplněnými hodnotami.

Přípojný bod

Přípojný bod se používá zejména při tvorbě součástek. Slouží k označení míst součástky, ve kterých se součástka zapojuje do obvodu. K těmto místům pak program hledá uchycení (viz uchopování). Lze použít i v běžném výkresu a zadává se pouze určením pozice. Tento grafický prvek se netiskne.

4.5.3 Transformace grafických prvků

Transformace se týkají vždy vybraných objektů. Nejprve je tedy nutno označit grafické prvky, které mají být transformovány. Poté vyberete z nástrojové lišty transformace příslušný nástroj – posun, rotaci nebo rotaci ve středu. Při aplikování posunu nejprve určíte bod, od kterého se bude vzdálenost a směr posunu určovat. Poté tažením myši umístíte transformované prvky do požadované pozice. Při rotaci nejprve určíte střed otáčení stiskem tlačítka myši nebo z klávesnice. Poté tažením myši nebo zadáním určíte úhel natočení. Transformace rotace ve středu slouží zejména k natočení více součástek o stejný úhel. Každý objekt se otočí kolem svého středu o zadaný úhel.

4.5.4 Vkládání součástek do výkresu

Knihovna součástek je k dispozici v levé části okna programu. Všechny součástky v knihovně jsou navrženy podle aktuálních norem. Nejdříve vyberte požadovanou součástku v kategoriích knihovny. Kliknutím na název součástky se součástka zobrazí v náhledu. Pro vložení do výkresu přetáhněte součástku z náhledu na plochu výkresu a umístěte součástku do požadované polohy.

4.5.5 Výběr

Nástroj výběr není na žádném panelu nástrojů. Tento nástroj se volí automaticky po každé akci nebo po zrušení akce klávesou Escape. Existují dva způsoby výběru. První je výběr jednoho grafického objektu. Po kliknutí tlačítkem myši program nalezne objekt ležící v blízkosti určeného bodu. Pokud nenalezne žádný objekt, přechází do druhého režimu výběru – hromadného. V tomto způsobu výběru zadáte ohraničující rám. Objekty, které padnou do rámu, jsou označeny.

Takto označené objekty jsou vykreslovány přerušovanou čarou. Navíc jsou vykresleny jejich hlavní body pomocí rámečků. Prostřednictvím těchto bodů lze s objekty manipulovat v závislosti na typu vybraného objektu. Pomocí středového bodu lze každý objekt posouvat. Ostatní slouží ke změně šířky, výšky, délky či tvaru objektu.

4.5.6 Změna vzhledu objektů

Grafickým objektům lze nastavovat barvu, styl čáry nebo tloušťku čáry. To se provádí pomocí nástrojové lišty „Změna vzhledu.“ Změna barvy, stylu nebo tloušťky se aplikuje vždy na vybraný objekt. Pokud není vybrán žádný objekt, použije se nové nastavení vzhledu pro každý nově nakreslený grafický prvek.

4.5.7 Zoomování

Zoomování se provádí kolečkem myši. Lze ho v programu aplikovat kdykoliv, ať už v průběhu kreslení, transformace nebo při jiné akci. Záporný zoom je omezen a nelze zmenšovat výkres do nekonečna. Kladný zoom není omezen.

4.5.8 Uchopování

Velkou předností programu jsou takzvané uchopovací body, pomocí nich je možno kreslit přesně a urychlit návrh schématu. Uchopovací body jsou nabízeny v průběhu tvorby výkresu. Existují dva režimy uchopování:

Režim při výběru

V tomto režimu jsou nabízeny pouze základní body vybraných objektů pro editaci těchto objektů. Pokud tedy např. vyberete čáru, jsou nabízeny k uchycení dva koncové a středový bod. Koncové body slouží pro změnu čáry, středový bod pro posun čáry. Podobně je tomu i u jiných grafických objektů.

Režim při kreslení nebo editaci grafického objektu

V tomto režimu jsou nabízeny všechny zvolené uchopovací body. Existuje několik typů těchto bodů (viz obr. 4.5):

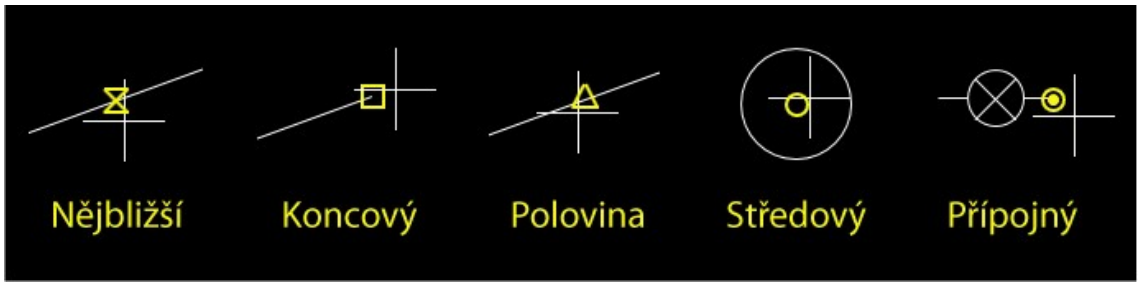
nejbližší – označuje nejbližší bod na grafickém objektu od aktuální pozice kurzoru

koncový – označuje místa konců čar, rohů a jiných důležitých míst

polovina – označuje střed čáry, bézierovy křivky a středy čar obdélníku

středový – označuje střed kružnice, elipsy a obdélníku

přípojný – označuje místa součástek, kde jsou umístěny přípojné body



Obrázek 4.5 Typy uchopovacích bodů

Každý typ uchopovacího bodu lze kdykoliv v průběhu kreslení vypnout pomocí lišty „Rychlé volby“ ve spodní části programu nebo v menu „Volby.“

4.5.9 Krokování a pravoúhlé kreslení

Elektrotechnická schémata se kreslí pravoúhle nebo v mřížce. Proto je tato možnost v programu nezbytná. Krokování i pravoúhlé kreslení se zapíná v menu „Volby“ nebo na liště „Rychlé volby“ a je opět možno je nastavit v jakémkoliv okamžiku – např. i v průběhu kreslení grafického objektu.

Krokování uchytává kurzor do mřížky, jejíž rozteč se nastavuje buď příkazy _set stepX a _set stepY nebo kliknutím pravým tlačítkem myši a výběrem *Nastavení velikosti kroku* z kontextového menu. Rozteč mřížky musí být vždy kladné číslo (i reálné) a udává se v milimetrech. Při výběru se krokování automaticky vypíná. Je tedy aktivní jen při kreslení, vkládání a transformaci objektů.

Pravoúhlé kreslení se zapojuje v průběhu kreslení. Uchytává *x*-ovou nebo *y*-ovou souřadnici podle předešlého bodu (či jiného významného bodu – záleží na typu kresleného objektu). Hodí se zejména pro kreslení pravoúhlých spojů, posunech v jednom směru, rotaci o úhel, jehož velikost je násobkem 90° a jiné.

Krokování a pravoúhlé kreslení lze libovolně kombinovat dohromady. Pozor – pokud je zapnuto pravoúhlé kreslení a zároveň je zobrazen uchopovací bod, má uchopování přednost a může porušit pravoúhlé kreslení.

4.5.10 Ovládání programu pomocí klávesnice

Velkou část programu lze ovládat pouze použitím klávesnice. Můžete volit jednotlivé nástroje pomocí příkazů, zadávat body, délky a úhly. Příkaz vždy začíná uvozovacím znakem _. Nástroje je možné také volit pomocí klávesových zkratek. Volba pomocí klávesových zkratek značně urychluje práci a je vřele doporučována. Program automaticky rozpoznává mezi tím, zda se jedná o příkaz, zadání délky a úhlu z klávesnice nebo klávesovou zkratku.

Seznam příkazů a klávesových zkratek:
_line [L] – vybere nástroj čára
_rectangle [R] – vybere nástroj obdélník
_circle [C] – vybere nástroj kružnice
_ellipse [E] – vybere nástroj elipsa
_text [T] – vybere nástroj text
_bezier [B] – vybere nástroj bézierova křivka
_connectPt [P] – vybere nástroj přípojný bod

Volba transformačních nástrojů pomocí příkazů a klávesových zkratek:
_move [M] – posun vybraných objektů
_rotate – rotace ve středu zvoleného objektu
_rotateAt [O] – rotace kolem zadaného středu

Další příkazy:
_undo [Ctrl+Z] – vrátí akci o krok zpět
_redo [Ctrl+Y] – posune akci o krok vpřed
_group [Ctrl+G] – vytvoří z vybraných objektů součástku
_ungroup [Ctrl+Shift+G] – rozloží součástku na grafické prvky
_del [Del] – smaže vybrané prvky

Další klávesové zkratky:

[Ctrl+S] – uloží aktuální dokument

[Ctrl+Shift+S] – uloží všechny otevřené dokumenty

[Ctrl+O] – otevře dokument

[Ctrl+P] – vytiskne aktuální dokument

[Mezerník] – posune výkres (přidržením)

[Escape] – zruší aktuálně prováděnou akci

[F1] – zobrazí nápovědu k programu

Zadávání bodů

Bod se zadává pomocí určení souřadnice *x* a *y*, které se oddělují středníkem a zadání se potvrzuje stiskem klávesy Enter (např. 35,2;160,5). Bod 0;0 na počátku odpovídá levému hornímu rohu kreslícího plátna v případě výkresu, nebo středu určeného pomocným křížem při kreslení součástky. Tento bod však během editace může různě cestovat v závislosti na tom, do jaké míry a kde zoomujeme výkres. Pro výstup na tiskárnu není tento bod rozhodující, protože před tiskem je uživatel vyzván k zadání okrajů výkresu pro konkrétní typ stránky.

Zadávání bodu není nijak závislé na použití uchopovacích bodů, krokování nebo použití pravoúhlého kreslení.

Zadávání délky

Délku určíte zadáním čísla a potvrzením klávesou Enter. Je možné zadat délku čáry, délku poloměru kružnice, délku hlavní a vedlejší poloosy elipsy apod. Ne vždy je však zadání délky povoleno. Záleží na konkrétní situaci. Lze ji zadat např. při kreslení čáry, když je zadán první bod, ale nelze ji zadat při určování prvního bodu čáry. O tom, zda je možno délku zadat, informuje informační box. Zadání neplatné délky (např. záporné) je oznámeno taktéž v informačním boxu a není provedena žádná akce.

Zadávání délky nezávisí na krokování, či na použití uchopovacích bodů, ale je závislé na použití pravoúhlého kreslení. Pokud tedy zadáte délku při pravoúhlém kreslení, dopočítá se délka podle zvoleného směru.

Zadávání úhlů

Úhly se zadávají ve stupních a je možno zadat je při aplikování transformace rotace nebo rotace ve středu. Úhel může být jak záporný (proti směru hodinových ručiček), tak kladný (ve směru hodinových ručiček).

Zadávání úhlu nezávisí na použití uchopovacích bodů, krokování či pravoúhlého kreslení.

4.6 Uložení do souboru

Každý výkres lze jednotlivě uložit do souboru pomocí volby v menu *Soubor – Uložit* (pro uložení do předešlého úložiště) nebo *Uložit jako* (pro uložení do nového souboru). Volbou *Uložit vše* proběhne uložení všech otevřených výkresů. Pokud některý výkres nebyl dosud uložen, budete požádáni o zadání jména a úložiště výkresu.

4.5.7 Tisk dokumentu

Při tisku program nejprve nabídne dialogové okno pro nastavení základních parametrů tisku. V tomto okně můžete nastavit libovolné zvětšení či zmenšení výkresu, velikost okrajů a typ papíru. Po potvrzení tlačítka OK je v závislosti na nastavených parametrech vykreslen obdélník s nastavenými rozměry stránky. Ten slouží pro určení umístění výkresu na tisknuté stránce. Po umístění vyzve program k určení tiskárny a nastavení dalších parametrů tisku a následně je dokument odeslán do tiskové fronty.

5 Závěr

Cílem této bakalářské práce bylo vytvořit program pro kreslení elektrotechnických schémat a prozkoumat možnosti nové technologie WPF. Výsledkem je program ElectroCAD a jeho výhody ve velké části vycházejí právě z použité technologie. Ve srovnání s většinou podobných programů disponuje hardwarově akcelerovanou grafikou. Díky WPF je použit antialiasing na všechny grafické prvky a tak odpadá nevzhledná kostrbatost čar. Předností programu je automatické uchopování grafických prvků, pomocí něhož je možno kreslit přesně a rychle. Program obsahuje bohaté možnosti ovládání pomocí rychlých klávesových voleb a zadávání délek, úhlů a bodů, což může výrazně přispět k urychlení tvorby výkresu. Nutno poznamenat, že díky stylování a použití animací je také docíleno příjemného vzhledu celé aplikace.

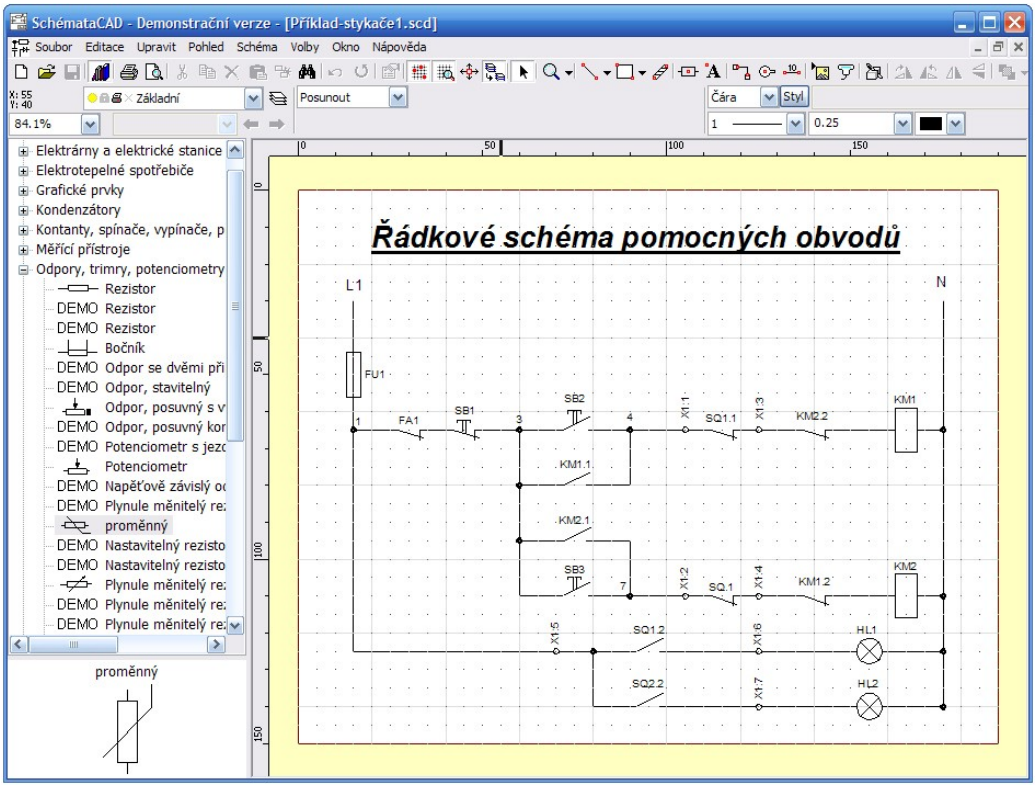
5.1 Srovnání s ostatními editory

SchémataCAD [6]

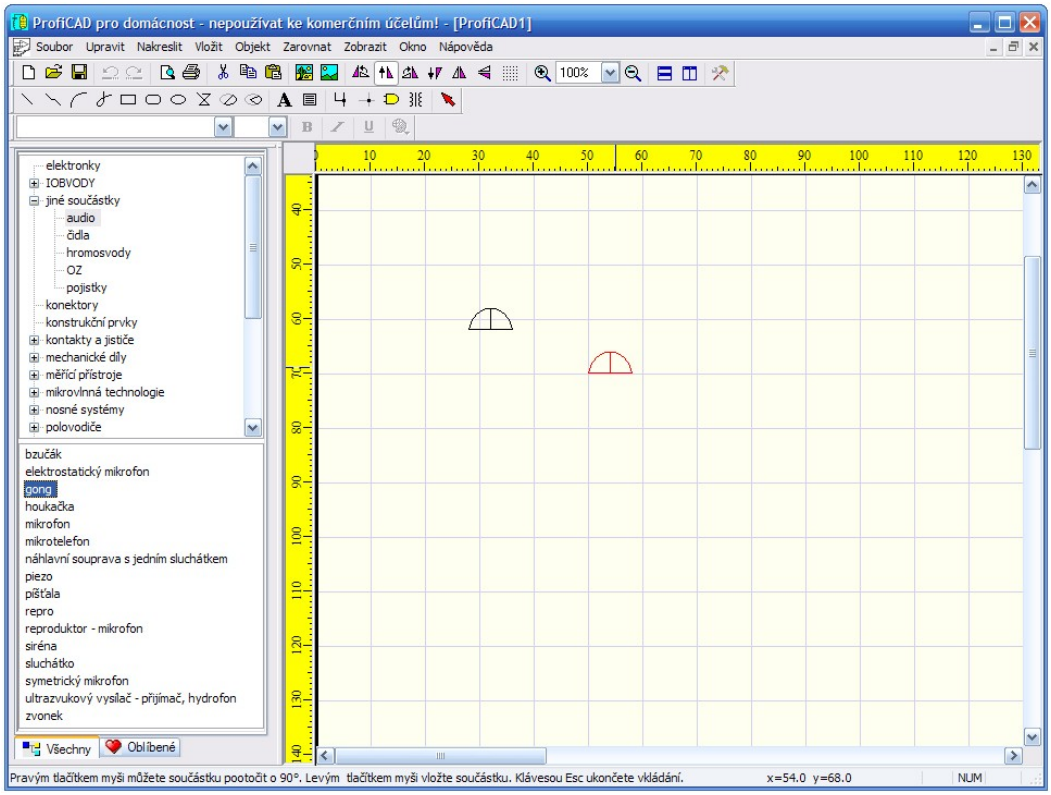
Je český nástroj pro tvorbu elektrotechnických schémat. Má velmi jednoduché ovládání a je cenově dostupný. Má bohatou knihovnu součástek, které je možno pomocí mřížky umísťovat na plochu výkresu. Nevýhodou je, že nepodporuje uchopovací body a bez použití mřížky nelze přesně kreslit. Délky, úhly a souřadnice nemohou být zadávány z klávesnice přímo v průběhu kreslení. Oproti ElectroCADu nemá hardwarově podporovanou grafiku a grafické prvky nejsou vyhlazované. Okno programu SchémataCAD je zobrazeno na obrázku 5.1.

ProfiCAD [7]

Je jednoduchý cenově dostupný program. Skládá se ze dvou samostatných částí – editor součástek a editor schémat. Obsahuje bohatou knihovnu a podporuje automatické číslování vložených součástek. Nevýhody jsou obdobné jako u předchozího programu. Nepodporuje kreslení ve více hladinách, nemá akcelerovanou grafiku, není možné zadávat body a délky v průběhu kreslení. Nepodporuje kreslení uchopovacími body a grafické objekty nejsou vyhlazované. Editor schémat programu ProfiCAD je zobrazen na obrázku 5.2.



Obrázek 5.1 Editor elektrotechnických schémat SchémataCAD

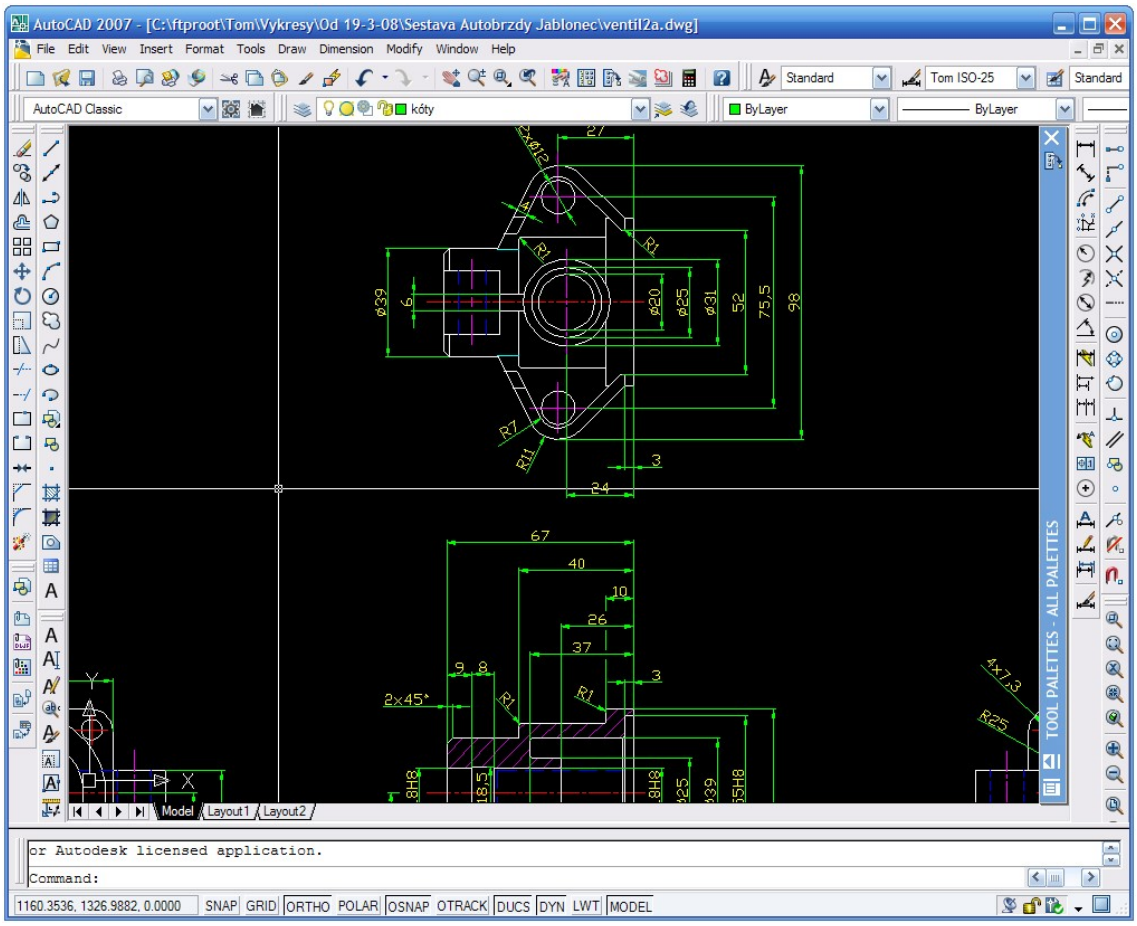


Obrázek 5.2 Editor elektrotechnických schémat ProfiCAD

AutoCAD [8]

AutoCAD (viz obr. 5.3) je špičkou mezi systémy CAD a stal se asi nejpoužívanějším nástrojem pro tvorbu technických výkresů vůbec. Program je vystavěn nad DirectX a tudíž disponuje hardwarově akcelerovanou grafikou. V průběhu kreslení pomáhá uživateli pomocí uchopovacích bodů, pomocí guidelines a zobrazování dalších pomocných informací. Je možné zadávat délky, úhly a další příkazy pomocí klávesnice. Podporuje kreslení jak ve 2D tak i ve 3D.

Nezaměřuje se na elektro a součástky potřebné pro elektrotechnická schémata je nutno do programu nahrát. Kvalita programu odpovídá jeho ceně, a proto se tak stává pro potřebu menších firem finančně nedostupný.



Obrázek 5.3 Software pro 2D a 3D projektování a konstruování AutoCAD

5.2 Možná vylepšení

Na vývoji programu se bude dále pokračovat. Je několik oblastí, které jsou zamýšleny implementovat v dalších verzích programu. Jsou to např.:

Hladiny

Program je přichystán pro budoucí implementaci kreslení ve více hladinách. Veškeré kreslení zatím probíhá pouze v jediné hladině. Pomocí hladin by bylo možno spravovat větší množství grafických prvků zároveň. Každá hladina by měla nést nastavení své barvy, stylu a tloušťky čáry. Mohlo by být možné zneviditelňovat, zamykat, zmrazovat, mazat, kombinovat apod.

Provázání součástek s výkresy

Při návrhu elektrotechnických schémat se často pro složitější objekt použije bloková značka a systém zapojení se následně rozkreslí podrobněji v dalším výkresu. Zajímavou vlastností programu by mohlo být propojení součástek s odkazem na výkres, kde je tato součástka podrobněji rozkreslena.

Automatické vkládání popisků

Některé součástky mají podle norem přidělené označující písmeno. Toto písmeno bývá doplněno pořadovým číslem, které označuje počet stejných součástek ve výkresu. Tento popisek by mohl program automaticky generovat a sledovat počty součástek na výkresu.

Výpis seznamu součástek

Program by automaticky mohl generovat seznam použitých součástek daného výkresu, spolu s jejich názvy a označením. Tento seznam pak exportovat např. do xls tabulky pro Excel.

Export

Zamýšleným rozšířením programu je export do souborů obrázků standardních typů (např. jpg, bmp, gif atd.), dále pak do dokumentu pdf pro Acrobat Reader nebo do výkresů pro program AutoCAD typu dwg a dwf.

Podpora vkládání bitmapových obrázků

Schémata jsou často zakreslována do půdorysů a stavebních plánů. Program by v budoucnu mohl implementovat možnost vložení bitmapy do výkresu.

LITERATURA

[1] Česká technická norma ČSN EN 61082-1–4: *Zhotovování dokumentů používaných v elektrotechnice,* Český normalizační institut, 1998

[2] Česká technická norma ČSN EN 60617-1–13: *Zhotovování dokumentů používaných v elektrotechnice,* Český normalizační institut, 1995

[3] Matthew MacDonald: *Pro WPF in C# 2008: Windows Presentation Foundation with .NET 3.5, Second Edition*, Apress, 2008

[4] Adam Nathan with Daniel Lehenbauer: *Windows Presentation Foundation Unleashed*, Sams publishing, 2007

[5] Laurence Moroney: *Foundations of WPF: An Introduction to Windows Presentation Foundation,* Apress, 2006

[6] Software pro elektrotechniky, projektanty a konstruktéry: URL http://www.elmer.cz

[7] Editor elektrotechnických schémat ProfiCAD: URL http://www.proficad.cz

[8] Software pro 2D a 3D projektování a konstruování AutoCAD: URL http://usa.autodesk.com

PŘÍLOHY

Obsah přiloženého CD

Na přiloženém CD najdete:

Spustitelnou verzi programu:	`bin`
Instalační soubory:	`instal`
Ukázkové příklady výkresů:	`example`
Uživatelskou dokumentaci:	`man/user`
Programátorskou dokumentaci:	`man/prog`
Zdrojové kódy:	`src`
Text této práce v PDF:	`bc`