Masarykova univerzita

Fakulta Informatiky

Bakalářská práce

Sdílená pracovní plocha

Martin Gracík

2008

Prohlášení

Prohlašuji, že tato práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

……..

Poděkování

Děkuji vedoucí své práce Evě Hladké a Jiřímu Denemarkovi za odbornou pomoc při vývoji aplikace sdílené pracovní plochy a psaní této práce.

Shrnutí

Obsahem této bakalářské práce je uvést čtenáře do problému kolaborativní spolupráce více uživatelů pomocí sítě internet a převést existující sdílenou pracovní plochu ze systému Linux do Microsoft Windows. Aplikace by měla být jednoduše použitelná pro běžného uživatele a mít tím pádem nativní vzhled. Zároveň však musí umět komunikovat se svými předchůdci pomocí standardizovaného protokolu.

V úvodu práce charakterizuji možné způsoby kolaborativní spolupráce více uživatelů pomocí sítě internet a technologie k tomu určené. Popisuji zde síť MBone, výhody vyplývající z užívání této technologie i způsob jak se k ní připojit.

Pro vývoj aplikace pro systém MS Windows jsem si vybral jazyk C++ a knihovnu wxWidgets. Uvádím zde, jak jsem postupoval při volbě technologie, podle čeho jsem se rozhodoval, a také nastiňuji hlavní rysy této knihovny.

V dalších kapitolách rozebírám implementaci, hlavní objekty aplikace a nejdůležitější metody v ní používané, hlavně ty, které jsou potřeba pro vykreslení objektů. Dále popisuji hlavní rozvržení aplikace a její základní ovládání.

Klíčová slova

Multicast, MBone, C++, wxWidgets, OGL

Obsah

1 Úvod

Již v pravěku lidé zjistili, že budou-li při lovu spolupracovat, mají větší šanci na úspěch. A nejinak je tomu i dnes. Téměř všichni vědci pracují v týmech, protože víc hlav víc ví. Dříve se však museli lidé scházet, aby spolu mohli diskutovat. Přestože máme rychlé dopravní prostředky, zabírá cestování stále hodně času, což je v dnešní době uspěchané době veliký problém. Proto se mnoho vědců stále snaží najít způsob, jak ulehčit a urychlit vzájemnou komunikaci. Jedním z řešení je využití internetu, který je dnes dostupný téměř na každém kroku. Jedná se o takzvané kolaborativní prostředí, což je virtuální prostředí určené pro setkání a vzájemnou komunikaci více subjektů.

Pro provoz takového systému potřebuje každý uživatel klientskou aplikaci, pomocí které bude pracovat, přístup k internetu a spojení se serverem, který řídí veškerou komunikaci. V dnešní době lze přenášet několik druhů informací, mezi něž patří zvuk a obraz. Důležitou součástí kolaborativní komunikace více uživatelů je pak takzvaná sdílená pracovní plocha, což je jakási virtuální tabule. Na tuto tabuli pak může kreslit více uživatelů najednou, přičemž změny provedené jedním uživatelem okamžitě vidí i ostatní účastníci. Jelikož tuto aplikaci využívají zejména lidé, kteří počítačům moc nerozumí, je velmi důležité, aby ovládání bylo co nejjednodušší a nejpřirozenější. Proto jsem si jako obsah své bakalářské práce zvolil vytvoření sdílené pracovní plochy pro operační systém Microsoft Windows. Ačkoliv je tento operační systém ve světě stále nejrozšířenější (hlavně mezi počítačovými laiky), neexistuje pro něj zatím podobná aplikace. Jak jsem již zmínil dříve, je velmi důležité, aby byla aplikace snadno ovladatelná. Zároveň však musí být kompatibilní s předchozími verzemi pro systém Linux, které využívají již standardizovanou síťovou komunikaci.

2 Videokonference

2.1 Videokonference

Videokonference je interaktivní spojení dvou a více, osob, při kterém zúčastnění nějakým způsobem komunikují. Nabídka způsobů komunikace je velice široká a závisí především na programovém vybavení zúčastněných. V zásadě se dá videokonference rozdělit na několik částí.

2.1.1 Přenos zvuku

Přenos zvuku je nejdůležitější složkou a zároveň je ze všech nejchoulostivější z hlediska přenosu a zpracování. Aby byla komunikace vedena na přijatelné úrovni, nesmí docházet k větším výpadkům a zpožděním.

2.1.2 Přenos obrazu

Přenos obrazu je nedílnou součástí videokonference. *„Význam interaktivního přenosu vzhledu a chování vašich partnerů je především psychologický. Pokud svůj protějšek nevidíte, vnímáte dialog s ním jako odtažitý a vzdálený. Jestliže však na monitoru sledujete, jak se váš partner tváří a co dělá, velmi se prohlubuje dojem přímého dialogu s ním a příliš se neliší od situace, kdy dotyčný sedí u vedlejšího stolu. Psychologové navíc tvrdí, že řeč přenáší pouze část informace při lidském dialogu. Zbytek komunikace obstarávají prvky vizuální - gestikulace, grimasy a podobně. I z tohoto pohledu je tedy význam přenášeného obrazu značný.“* (Satrapa, Wimmer, & Adamec, 1999)

2.1.3 Sdílená pracovní plocha

Občas je při konferenci vhodné doplnit výklad určitým nákresem či diagramem. Přesně to umožňuje sdílená pracovní plocha. Je to grafická tabule (čistá, nebo s grafikou na pozadí), na kterou může kterýkoliv z účastníků nakreslit, či napsat cokoliv. Tyto objekty se okamžitě zobrazují ostatním zúčastněným, kteří je mohou libovolně měnit.

2.2 MBone videokonference

MBone videokonference je zvláštní případ IP konference. IP konference ke svému provozu využívá protokol IP, který je velice rozšířen zejména díky své jednoduchosti a snadné implementaci. Největší problém tohoto protokolu ovšem je, že nebyl navržen pro tyto účely. Sám totiž nezajistí, že poslané pakety dorazí v určeném pořadí, do určité doby, nebo že vůbec

dorazí do cíle. Jak jsem již naznačil dříve, MBone konference využívá IP protokolu, ovšem k adresování používá skupinové (multicast) adresy. Odtud také název MBone (Multicast backBone).

2.2.1 Skupinové adresy

Skupinové adresy se používají k přenosu multimediálních dat mezi několika účastníky. Bez nich by objem dat byl několikanásobně větší. Vysílající může posílat data skupině tvořící žádného, jednoho nebo více účastníků, a stačí mu k tomu pouze znát skupinovou adresu. Uživatelé mohou do dané skupiny kdykoliv přijít, nebo z ní odejít. Výhody skupinového adresování si můžeme předvést na videokonferenci mezi čtyřmi účastníky. Pokud by využívali standardního protokolu, posílala by se data od každého počítače k serveru a zpátky samostatně celou cestu, což by znamenalo 8 datových toků. Pokud by ale využili skupinové adresy, data by se na společné cestě posílala pouze jednou a rozdělila se až v nejzazším možném místě. Tím se ušetří až 50% kapacity sítě.

Skupinové adresování má však i své nevýhody. Nejvíce zmiňovaný problém je jeho dostupnost. Většina internetové infrastruktury multicastové vysílání podporuje, to však neznamená, že jej lze považovat za standard. Stále je totiž mnoho oblastí, kde není multicast dostupný. Navíc je jednou z prvních služeb, které se vypínají, začnou-li být s internetovým přenosem problémy. „*Pro přenosy s velkým objemem dat může být limitující i maximální dosažitelná rychlost. Při testování multicastu pro účely gridového počítání bylo naměřeno, že na linkách s šířkou pásma 1 Gb/s je možné dosáhnout přenosové rychlosti maximálně kolem 40 Mb/s, což je méně, než vyžadují např. dva videostreamy ve formátu DV.*“ (Hladká & Denemark, Web Jiřího Denemarka, 2005) Ze samotné podstaty multicastového vysílání pak vyvstává druhý problém, a tím je robustnost. Multicastové zprávy se co nejdelší možnou cestu posílají pouze jednou a až na konci své cesty se kopírují a rozesílají mezi své příjemce. To sice zmenšuje objem dat, který je nutné transportovat, ovšem přerušení vedení mezi dvěma uzly, nebo ztráta paketu můžou znamenat velké ztráty informací.

2.2.2 Síť MBone

„*Síť MBone vznikla právě za účelem distribuce skupinově adresovaných dat v Internetu. Jedná se o virtuální síť, která je podmnožinou Internetu. Jejími členy jsou sítě, zařízení a počítače podporující skupinové adresování. Aktuální data o velikosti MBone se nám nepodařilo zjistit, nicméně v roce 1996 do ní bylo zapojeno přibližně 1700 sítí z dvaceti zemí. Odpovídala tak zhruba velikosti Internetu z roku 1990.*“ (Satrapa, Wimmer, & Adamec, 1999) Chcete-li se připojit do sítě MBone, lze to udělat dvěma způsoby. Pokud je Vaše počítačová síť součástí MBone, můžete se napojit přímo, pokud ne, lze to provést pomocí tunelu.

2.2.2.1 Připojení tunelem

„Pojmem „tunel“ se obecně označuje průchod určitých speciálních datových paketů částí sítě, která jim nerozumí nebo nemá potřebné vlastnosti. V našem případě funguje tunel tak, že se váš počítač spojí s některým vzdáleným směrovačem nebo počítačem, který je součástí MBone. Tento spoj povede zcela běžným Internetem. Kdykoli se má odeslat skupinově adresovaný datagram do sítě MBone, Váš počítač jej „zabalí“ do obyčejného IP datagramu. Jeho adresátem je druhý konec tunelu a ve svém těle nese původní skupinově adresovaný datagram. Tak vznikne zcela běžný datagram, který ke své přepravě nevyžaduje žádné nadstandardní vlastnosti. Zcela obvyklým způsobem bude doručen směrovači na konci tunelu. Ten jej rozbalí, čímž se z tunelu vynoří skupinový datagram, který pak dále odešle obvyklými postupy dalším členům MBone. V opačném směru v.e funguje obdobně - vzdálený směrovač zabalí skupinový datagram, pošle tunelem vašemu, který jej rozbalí a zpracuje. Pomocí takovýchto tunelů jsou běžně propojeny některé části sítě MBone. Váš tunel pochopitelně může vést kamkoli. Vzhledem k objemu přepravovaných dat je však záhodno, aby končil pokud možno co nejdříve a vedl dostatečně dimenzovanými linkami. „ (Satrapa, Wimmer, & Adamec, 1999)

2.3 Reflektory

Ačkoliv je zapojení do multicastové sítě nejjednodušší cestou, jak provozovat konferenci, může být uživatel připojen do sítě, kde multicast není podporován. Poté musí použít jiný způsob, jak nahradit multicastové služby. Řešením problému nedostupnosti multicastu pak může být použití prvku, který bude sbírat data z jednoho zdroje a přeposílat ostatním účastníkům a zpracovávat přicházející data v závislosti na jejich odesílateli a příjemci. Nejpoužívanější název pro tento bod je reflektor nebo zrcadlo.

2.3.1 Protokol RTP/RTCP

„Protokol RTP (Real-time Transport Protocol) a s ním úzce související protokol RTCP (RTP Control Protocol) poskytují služby pro přenos dat s real-time charakteristikami jako je např. interaktivní video a audio. Mezi poskytované služby patří identifikace typu přenášených dat, sekvenční číslování paketů, časové značky a kontrola doručení. Pro vlastní přenos dat je využíván protokol nižší úrovně, kterým je v převážné většině případů protokol UDP. RTP žádným způsobem neřeší spolehlivost doručení dat ani nezaručuje, že data budou doručena ve správném pořadí a v garantovaném čase. Poskytuje ovšem dostatek informací k tomu, aby byl příjemce schopen zrekonstruovat přijatá data v pořadí, v jakém byla odvysílána.

Vzhledem k tomu, že RTP protokol byl navržen primárně pro skupinové multimediální konference (i když samozřejmě nevylučuje jiné možnosti použití), zahrnuje také mechanismy pro identifikaci jednotlivých účastníků. Každý RTP paket obsahuje ve své hlavičce 32b identifikátor zdroje paketů (SSRC, synchronization source), který si každý klient generuje náhodně a tak, aby byl jedinečný v rámci jedné relace. Pokud klient přijme paket se shodným

SSRC (čili detekuje konflikt), vygeneruje si náhodně jiný identifikátor. Z toho vyplývá, že v průběhu jedné relace může být stream vysílaný od jednoho uživatele identifikován několika různými SSRC. Jednotlivé streamy s potenciálně různými identifikátory pocházející od stejného uživatele jsou spolu provázány pomocí SDES (source description) RTCP paketů s detailnějšími informacemi o vysílající straně. Tyto pakety kromě uživatelem nastavitelného jména a emailové adresy obsahují také aplikací generované kanonické jméno (CNAME, canonical end-point identifier), které je pro daného uživatele konstantní a mělo by být shodné i pro streamy generované různými aplikacemi (např. audio i video).“ (Hladká & Denemark, Web Jiřího Denemarka, 2005)

3 Analýza požadavků

Sdílená pracovní plocha by měla umět zprostředkovat vizuální komunikaci mezi dvěma a více uživateli. Kterýkoliv z uživatelů by měl mít možnost vytvářet a posouvat základní objekty, ať už je vytvořil on sám, nebo někdo jiný. Jelikož se jedná pouze o kreslící plochu, ostatní druhy komunikace (textová, zvuková, popř. obrazová) nejsou implementovány, ale lze je nahradit běžně používanými aplikacemi třetích stran, např. ICQ, MSN, GTalk.

Dle zadaných požadavků se jako nejlepší jeví systém klient – server. Data jsou uchovávána na serveru, zatímco klientské aplikace slouží k vytváření a zobrazení daných objektů.

Mým úkolem bylo navrhnout a implementovat klientskou část a připravit rozhraní pro síťový protokol. Budu tedy této části věnovat více pozornosti.

3.1 Klíčové pojmy

3.1.1 Server

Serverová část má za úkol udržovat informace o dané scéně a zprostředkovat komunikaci mezi klienty. Poté, co se klient připojí k serveru, dostane veškeré informace nutné k vykreslení scény do současné podoby. Dále server přijímá informace od uživatelů, ukládá a přeposílá ostatním zúčastněným. Jelikož spolu komunikují pomocí UDP protokolu, musí nutně obsahovat mechanismus, který zajistí, aby všichni klienti spolehlivě dostali veškeré zprávy. Pokud zjistí, že někdo nemá kompletní scénu, musí ihned přeposlat chybějící informace.

3.1.2 Klient

Klientská aplikace musí umět komunikovat se serverem – posílat mu informace o provedených akcích a naopak měnit scénu podle informací přicházející ze serveru. Musí obsahovat intuitivní grafické uživatelské rozhraní, aby práce s ním byla pro uživatele přirozená a jednoduchá. Hlavní požadované funkce jsou vytváření a přesouvání základních grafických objektů (obdélník, elipsa, přímka, šipka, štětec a text), vkládání obrázků a přibližování.

3.2 Výběr technologií

Původní sdílená pracovní plocha využívala knihovnu imlib2 systému Linux, jejíž nahrazení bylo hlavním úkolem mé bakalářské práce.

3.2.1 C++

Výběr programovacího jazyku C++ byl zřejmý již zpočátku. Aplikace musí být schopná komunikovat se svými předchůdci, a proto musí používat síťový protokol, který je již standardizován. Tento protokol je napsán v jazyce C a jeho zdrojové kódy jsou veřejně přístupné. Bylo by tedy zbytečné a zároveň velmi namáhavé tento kód přepsat do jiného jazyka. Proto jsem musel vybírat mezi C++ knihovnami, aby byla implementace protokolu co nejjednodušší.

Chvíli jsem uvažoval i nad knihovnou MFC, která je podporována přímo firmou Microsoft, ale nakonec jsem se pro ni nerozhodl hned ze dvou důvodů. Jedním z nich je příliš složitá práce s bitmapou, která je ve sdílené pracovní ploše velmi důležitá. Ten druhý byl nemožnost portace, která sice není zadáním mé práce, ovšem při úspěšném nasazení na operačním systému MS Windows by se mohlo hodit použití i na jiných platformách.

3.2.2 Knihovna wxWidgets

Důležité aspekty při výběru technologií na vývoj sdílené pracovní plochy byly nativní prostředí v operačním systému Microsoft Windows, pro který je sdílená pracovní plocha primárně určena, a zároveň snadná implementace síťového protokolu. Ze všech možných řešení jsem proto zvolil multiplatformní knihovnu wxWidgets, konkrétně verzi 2.8.

wxWidgets je C++ knihovna určená zejména pro vytváření grafického uživatelského rozhraní pro více platforem. Od verze 2 podporuje operační systémy Microsoft Windows, Unix s GTK 1.x nebo 2.x, Unix s X11, Mac OS X a OS/2. Knihovna wxWidgets byla původně vytvořena v Institutu aplikované umělé inteligence Edinburské univerzity a byla určena pouze pro vnitřní účely. Později však byly zdrojové kódy zveřejněny. Lze ji využít volně pro nekomerční i komerční účely, zdrojové kódy se mohou libovolně upravovat, a jsou vyvíjeny mnoha uživateli. Nedílnou součástí této knihovny je výborně zpracovaná fulltextová dokumentace s mnoha příklady.

Navíc, na rozdíl od všech ostatních, má wxWidgets implementováno i spoustu tříd, které se netýkají jenom grafického uživatelského rozhraní. Umí pracovat například s datem a časem, se sítí a dokonce i OpenGL.

3.2.2.1 Nativní vzhled

Nejdůležitější vlastností této knihovny je nativní vzhled. Umí vykreslit veškeré ovládací prvky (např. posunovací lišty, comboboxy) a základní dialogy přesně tak, jak vypadají na dané platformě. Většina volně dostupných řešení sice dovoluje vytvoření interaktivního uživatelského rozhraní, ovšem nevypadá stejně jako ostatní aplikace a tudíž může hodně uživatelů odradit. Uživatelé jsou zvyklí na určitý vzhled a chování programů, a pokud se něco

jen trochu liší, bývají k tomu skeptičtí. Sdílená pracovní plocha je program určený primárně pro běžného uživatele, a tudíž je kladen důraz na obvyklý vzhled a jednoduché ovládání.

3.2.2.2 Portabilita

Další nespornou výhodou použití wxWidgets, na rozdíl od MFC, je její multiplatformní využití. Její aplikační programové rozhraní (API) je stejné, nebo velmi podobné pro všechny podporované platformy. To znamená, že programátor píše program, který je po malých úpravách (pokud vůbec) přeložitelný na MS Windows, Linuxu i Mac OS X, což je v porovnání s kompletním přepisováním celého kódu ohromná úspora nejen času. Navíc to znamená, že programátor nemusí znát aplikační programové rozhraní pro každou platformu.

Z toho vyplívá, že sdílená pracovní plocha vyvinutá ve wxWidgets bude po malých úpravách lehce portovatelná i na ostatní operační systémy (např. Linux, Mac OS X), ačkoliv byla vyvinuta primárně pro Microsoft Windows. Portovatelnost ovšem není zadáním mé bakalářské práce, proto jsem aplikaci napsal tak, aby byla co nejrychlejší a nejstabilnější právě na MS Windows.

3.2.2.3 OGL

OGL (Object Graphic Library) v překladu znamená knihovna grafických objektů. Díky této knihovně, která je součástí wxWidgets, je možné vytvářet základní grafické objekty (například obdélník, elipsu, přímku, šipku) a dále s nimi manipulovat. Tento datový model se přesně hodí pro naší sdílenou pracovní plochu, jelikož kreslení grafických objektů a manipulace s nimi je její podstatou.

3.2.3 Vývojové nástroje

3.2.3.1 wxPack

Balík wxPack[1] nainstaluje do MS Windows vše důležité pro vývoj aplikací využívajících wxWidgets. Obsahuje zdrojové kódy knihovny wxWidgets, vývojové prostředí wxFormBuilder[2] určené pro vývoj wxWidgets aplikací (zejména grafického uživatelského rozhraní), a integraci wxWidgets do MS Visual Studia.

[1] http://wxpack.sourceforge.net/
[2] http://wxformbuilder.org/

3.2.3.2 Microsoft Visual Studio 2005

Microsoft Visual Studio 2005[3] je dle mého názoru nejlepší a nejkomplexnější vývojové prostředí současnosti. Programuji v něm (nebo jeho předchůdci) již několik let a proto jsem při výběru technologií neváhal ani chvíli.

3.2.3.3 wxFormBuilder

Kompletní integrace wxWidgets do MS Visual Studia sice existuje, ale bohužel je placená. Proto jsem si vybral wxFormBuilder, který je zadarmo a integruje se do MS Visual Studia alespoň částečně, což pro mé účely plně postačuje.

wxFormBuilder je WYSIWYG[4] editor sloužící pro návrh uživatelského rozhraní pro knihovnu wxWidgets. Lze v něm jednoduše vkládat dialogy a ovládací prvky a zároveň velmi rychle nastavovat jejich parametry. Tento nástroj jsem však použil pouze na začátku k základnímu návrhu aplikace, protože umí pracovat pouze se základními třídami, zatímco já jsem potřeboval pracovat se svými odvozenými třídami.

[3] http://msdn.microsoft.cz/vstudio/
[4] z anglického What You See Is What You Get – co je vidět na obrazovce, to bude i konečný výsledek

4 Návrh systému

Systém komunikace spočívá v zasílání zpráv mezi serverem a klienty vazbou 1:n. Server musí být vždy dostupný, jinak by klienti nebyli schopni komunikovat mezi sebou. Práce na klientovi samotném však nesmí být na serveru závislá. Není-li server dostupný, musí aplikace pracovat jako samostatná jednotka s plnou funkcionalitou.

4.1 Server

Server je nejdůležitější částí síťové komunikace. Bez něj by se klienti nebyli schopni dorozumět mezi sebou. Udržuje totiž informace o právě probíhajícím sezení. Pokud se chce klient zúčastnit konference, musí se připojit k serveru, který ho autorizuje, poznamená si jeho adresu a zašle mu aktuální podobu pracovní plochy. Pak už jen čeká na zprávy od klientů, ukládá si je a přeposílá ostatním účastníkům. Jelikož komunikace probíhá pomocí protokolu UDP, musí server obsahovat mechanismus, který zajistí, že všichni klienti dostanou všechny zprávy. Pokud ne, musí být schopný chybějící informace přeposlat tak, aby měli všichni účastníci co nejaktuálnější podobu pracovní plochy.

4.2 Klient

Klientská aplikace musí uživateli umožnit vytváření základních grafických objektů, a to obdélník, elipsu, úsečku, šipku, text a volné tahy štětcem. Poté může uživatel s již vytvořenými objekty pohybovat po ploše, vytvářet nové záložky, měnit barvu štětce a přibližovat pohled. Dále musí být uživateli umožněno načítat na plochu obrázky. Při každém takovém načtení se vytvoří nová záložka. Ta se vytvoří i v případě, že si uživatel přiblíží pohled. Pokaždé, když uživatel něco změní, nebo přidá, musí klientská aplikace poslat definovanou zprávu serveru, který si ji uloží a přepošle ostatním účastníkům. Stejně tak opačně – dostane-li aplikace zprávu ze serveru, musí být schopna zprávu analyzovat a vykonat určené příkazy.

4.3 Komunikace

Komunikace mezi klienty probíhá prostřednictvím serveru, kterému posílají předem definované zprávy. Jedná se o datové struktury obsahující akci, která byla provedena, a potřebné parametry (např. souřadnice). Tyto struktury jsou již standardizovány a zdrojové kódy jsou volně přístupné. Aplikace tudíž musí posílat stejné informace jako její předchůdci, kvůli zachování kompatibility. Implementovat síťovou komunikaci ovšem nebylo náplní mé bakalářské práce, proto ji nebudu dále rozebírat.

5 Implementace

Základ aplikace tvoří okno (frame), do kterého je vložen hlavní panel s menu, nástrojovou lištou, stavovým řádkem a tab kontrolem. Do tab kontrolu se při vytvoření nového listu vloží dialog třídy CWbdCanvas, který spravuje vykreslování vložených geometrických obrazců.

Jelikož jsem si musel pro vlastní účely upravit některé z metod knihovny OGL, nemohl jsem je do projektu přidat již zkompilované. Musel jsem tedy upravit zdrojové kódy (což je v souladu s licenčními podmínkami), vložit je do projektu a společně s ním kompilovat.

5.1 Základní třídy

5.1.1 CWbdProxy

CWbdProxy je třída spojující všechny části aplikace. Při spuštění se vytvoří jedna jediná její instance. Všechny ostatní třídy mají jako povinný parametr konstruktoru ukazatel právě na tuto instanci, který si vnitřně uloží. Tím je zajištěno, že všechny třídy mají k CWbdProxy přístup a tudíž mohou využívat jejich metod. To je výhodné zejména při ukládání a čtení globálních parametrů, jako jsou barva a tloušťka čáry, nebo zvolený kreslící nástroj. Jelikož se metody pro nastavování těchto parametrů dají volat z více míst, jsou opatřeny zámkem (třída Mutex), aby je bylo možné nastavovat vždy pouze z jednoho místa.

Všechny dialogy obsahují metodu GetProxy(), která vrátí referenci na globální instanci třídy CWbdProxy.

5.1.1.1 Void NewTab(wxFrame)*

Tato metoda přidá do aplikace nový panel. Pokud má ukazatel wxFrame hodnotu NULL, vytvoří se zcela nový, pokud však jako parametr předáme již vytvořený panel, provedou se pouze následující kroky – přidání panelu na kreslící plochu (nová záložka v Tab kontrolu) a do vektoru panelů. Tuto metodu potřebujeme volat z více míst. Může ji volat aplikace při příchozí zprávě, hlavní okno při zvolení položky z hlavního menu (Nový panel a Otevřít obrázek), nebo přímo kreslící panel (při zvětšování obrázku). Proto je tato metoda v CWbdProxy, kde je dostupná pro všechny.

5.1.1.2 void SetToolType(eToolType)

Tato metoda nastavuje aktuálně zvolený nástroj na hodnotu předanou jako parametr. Jedná se o výčtový typ eToolType. Je volána ze třídy CWbdFrame po kliknutí na požadovaný prvek v nástrojové liště.

5.1.1.3 eToolType GetToolType()

Metoda GetToolType vrací aktuálně zvolený nástroj a to hodnotu výčtového typu eToolType. Je volána zejména z vnitřního dialogu, aby věděl, jaký obrazec se nyní bude kreslit.

5.1.1.4 SetBrushColor(wxColour), SetPenColor(wxColour)

Tyto dvě metody nastavují barvu štětce (pera) na požadovanou hodnotu předanou jako parametr wxColour.

5.1.1.5 wxBrush& GetBrush(), wxPen& GetPen()

GetBrush a GetPen vrací reference na obecně platný štětec (pero) a jsou volány při vykreslování objektů. Přesněji při nastavení kreslící plochy předtím, než bude daný objekt vykreslen.

5.1.2 CWbdApp

CWbdApp je třída odvozená od wxApp a reprezentuje aplikaci samotnou. Při své inicializaci vytvoří hlavní okno, kterému předá referenci na právě vytvořenou instanci CWbdProxy.

5.1.3 CWbdFrame

CWbdFrame je třída vyděděná ze třídy wxFrame, což je okno, jehož velikost a pozice můžou být libovolně měněny. Obsahuje titulek okna se základními tlačítky (minimalizovat, maximalizovat a zavřít), menu, stavovou lištu, panel nástrojů a tabcontrol.

Třída CWbdFrame obsluhuje události přicházející při výběru položky z hlavního menu nebo lišty nástrojů. Po zvolení kreslícího nástroje z lišty se zavolá metoda třídy CWbdProxy, která zajistí jeho přepnutí v rámci celé aplikace.

CWbdFrame ale hlavně analyzuje zprávy přicházející ze sítě. Ty mají definovanou strukturu a obsahují číslo aktivního panelu, typ akce, která se má provést, a její parametry. CWbdFrame si díky CWbdProxy najde ukazatel na příslušný panel, a v něm zavolá danou metodu.

5.1.4 CWbdCanvas

CWbdCanvas je nejdůležitější část aplikace. Stará se o vykreslování objektů a práci s nimi. Je vyděděná ze třídy CShapeCanvas, což je základní dialog knihovny OGL a jsou v něm ošetřeny základní události pro ovládání myší. Ty jsem sice musel pro své potřeby upravit, ale jako inspirace, jak zacházet s objekty, posloužily výborně.

5.1.4.1 wxDiagram

Aby byl dialog schopný zacházet s grafickými objekty, musí k němu být připojen objekt wxDiagram. Ten uchovává informace o vytvořených objektech, o jejich velikosti a pozici a spoustě dalších parametrech. Zároveň nabízí základní metody pro práci s nimi. Mezi nejdůležitější patří přidat objekt na plochu nebo ho z něj naopak odstranit a hledání objektů, kterého je využito zejména při zjišťování, zda uživatel nekliknul na nějaký objekt. Jestli ano, funkce předá ukazatel na něj a my s ním můžeme dále pracovat. Objekt wxDiagram je implicitně vytvořen a připojen v konstruktoru třídy CWbdCanvas. Bez něj by totiž nebyla možná jakákoliv práce s objekty.

5.1.4.2 Kreslení

Metody kreslení byly vytvořeny tak, aby je bylo možné zavolat jak po akci uživatele, tak po příchozí zprávě ze serveru, a to dvěma způsoby.

DrawShape(beginPoint, endPoint, toolType, bDragged, *pDC)

Jeden ze způsobů volání je obecný, kdy se pro vykreslení objektu zavolá metoda DrawShape a parametr toolType (který nabývá hodnoty eToolType) určí, který tvar se má vykreslit. Parametry beginPoint a endPoint určují počáteční a koncový bod. Hodnota bDragged je typu BOOL a určuje, zda uživatel táhne myší a objekt se má tudíž vykreslit pouze dočasně, nebo zda se má vytvořit a uložit do wxDiagramu. Tento parametr je defaultně nastaven na FALSE pro snadnější volání. Ukazatel na kontext zařízení (Device Context) určuje kam se bude objekt vykreslovat. Je-li nulový (defaultně), kreslí se na klientské okno.

DrawRectangle(beginPoint, endPoint, bDragged, *pDC)

Druhý způsob vykreslení spočívá ve volání odlišné funkce pro každý tvar. Parametry jsou stejné jako v předchozím případě, pouze chybí typ tvaru, který je již obsažen v názvu metody (DrawRectangle, DrawArrow, DrawLine…). Jelikož ještě nevím přesně, jaké zprávy budou ze serveru přicházet a jaké budu posílat, zvolil jsem implementaci obou těchto metod, které se dají volat nezávisle na sobě.

6 Aplikace WBD

6.1 Hlavní rozvržení aplikace

Díky wxWidgets má aplikace nativní vzhled. Vypadá jako jakákoliv jiná aplikace systému MS Windows. Po spuštění se objeví hlavní okno sdílené pracovní plochy obsahující menu, nástrojovou lištu, stavovou lištu a tabcontrol.

Hlavní menu obsahuje položky sloužící k vytvoření nové záložky, otevření obrázku do nové záložky a otevření konfiguračního dialogového okna. Toto okno ovšem zatím není implementováno, protože není hotová síťová část aplikace, a proto ho budu navrhovat až s ohledem na množství a druh konfigurovatelných položek. Určitě ale bude komunikovat s třídou CWbdProxy, která udržuje veškerá nastavení programu.

Stavový řádek se skládá ze dvou částí. Jedna ukazuje souřadnice bodu na ploše, nad kterým se nachází kurzor, zatímco druhá část ukazuje nápovědu k jednotlivým položkám v menu a v nástrojové liště.

6.2 Ovládání

Po spuštění aplikace, by se mělo objevit konfigurační okno připojení k serveru (které ještě není implementováno). Poté, co se uživatel připojí k serveru, vyžádá si od něj aktuální podobu pracovní plochy, která se ihned vykreslí.

Celá aplikace se ovládá myší. V nástrojové liště si uživatel může zvolit nástroj, se kterým chce pracovat, mezi něž patří kreslení libovolného tvaru volnou rukou (štětec), šipky, čáry, obdélníku, elipsy a textu. Dále jsou zde k dispozici nástroj „Lupa“, s jejíž pomocí lze přiblížit libovolný výřez plochy, nástroj přesunutí a kopírování, které slouží k přesunu (resp. kopírování) nakreslených objektů. Poslední dva prvky nástrojové lišty slouží k nastavení barvy a tloušťky čáry. Kreslení probíhá tak, že uživatel stiskne levé tlačítko myši na počátečním bodu a táhne myší do koncového. Poté co pustí tlačítko, objekt se vytvoří a je možné s ním libovolně pohybovat.

Uživatel může pracovat s několika pracovními plochami, mezi kterými lze přepínat pomocí záložek zobrazených na vrchu kreslící plochy, nebo pomocí seznamu záložek zobrazených v pravé části okna. Záložky jsou prozatím pojmenovány „Záložka x“, kde „x“ je její pořadové číslo – kvůli zpětné kompatibilitě. V budoucnu by však mohlo být pojmenování libovolné.

7 Závěr

Portace sdílené pracovní plochy se dle mého názoru vydařila. Aplikace je napsaná tak, aby připojení síťové složky bylo co nejjednodušší. Stačí posílat definované zprávy a aplikace vše vykoná sama. Hlavním důvodem portace bylo přiblížit vzdálenou pracovní plochu i běžným uživatelům používající systém Microsoft Windows, což se díky knihovně wxWidgets a jejímu nativnímu vzhledu podařilo.

Další kroky při vývoji by měli směřovat k připojení síťové komunikace mezi klientskou aplikací a serverem, což bude práce pro Karola Pála z Fakulty Informatiky Masarykovy Univerzity. Dále by se měly vyrobit ikonky pro nástrojovou lištu pro lepší orientaci uživatelů a konfigurační dialog pro nastavení aplikace, zejména síťové komunikace.

8 Literatura

Encryption implementation for WBD. (20. Srpen 2001). Získáno 13. Říjen 2007, z CESNET: http://www.cesnet.cz/doc/techzpravy/2001/08/

Highfield, J. (nedatováno). *Whiteboard Tool*. Získáno 13. Říjen 2007, z http://www-mice.cs.ucl.ac.uk/multimedia/software/wbd/

Hladká, E. (nedatováno). *Komunikační portál.* Získáno 12. Listopad 2007, z Masarykova univerzita - Ústav výpočetní techniky: http://www.ics.muni.cz/zpravodaj/clanky_tisk/239.pdf

Hladká, E., & Denemark, J. (2005). Získáno 12. Listopad 2007, z Web Jiřího Denemarka: http://sitola.fi.muni.cz/~xdenemar/files/doc/2005-Olomouc/text.pdf

Nets Group, Computer Science, University College London. (29. Červen 2007). *SUMOVER Project*. Získáno 13. Říjen 2007, z SUMOVER Project: http://www.cs.ucl.ac.uk/research/sumover/

Satrapa, P., Wimmer, M., & Adamec, P. (1999). *Videokonference po síti MBone.* Praha: CESNET z.s.p.o.

Smart, J. (Září 1998). *Object Graphics Library 3.0.* Získáno 13. Říjen 2007, z http://biolpc22.york.ac.uk/wx/contrib/docs/html/ogl/ogl.htm

Smart, J., Hock, K., & Csomor, S. (2005). *Cross Platform GUI Programming with wxWidgets.* Prentice Hall PTR.

Smart, J., Roebling, R., Zeitlin, V., & Dunn, R. (2007). *wxWidgets 2.9.0: A portable C++ and Python GUI toolkit.* Artificial Intelligence Applications Institute.