

**Evropský polytechnický institut, s.r.o. v Kunovicích**

**Studijní obor: Elektronické počítače**

# **DIGITÁLNÍ ZPRACOVÁNÍ A PŘENOS VIDEA**

**(Bakalářská práce)**

**Jan Válek, DiS.**

**Duben 2007**

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně pod vedením RNDr. Davida Skoupila a uvedl v seznamu literatury všechny použité literární a odborné zdroje.

Olomouc, duben 2007

Jan Válek, DiS.

Děkuji svému vedoucímu práce RNDr. Davidu Skoupilovi a oponentu Ing. Vladimíru Grohmannovi za odborný dohled, který mi poskytli při zpracování mé bakalářské práce. Dále pak RNDr. Stanislavu Michalovi a Ing. Jiřímu Burdovi za poskytnutí užitečných internetových odkazů a literatury. Své rodině a kamarádům za podporu.

Olomouc, duben 2007

Jan Válek, DiS.

# Obsah

<b>Obsah .....</b>	<b>2</b>
<b>Úvod.....</b>	<b>6</b>
<b>1 Přehled možností komprimace videa .....</b>	<b>7</b>
1.1 <i>Bezeztrátové .....</i>	7
1.1.1 RAW .....	7
1.1.2 HuffYUV.....	8
1.2 <i>Ztrátové kodeky .....</i>	8
1.2.1 Indeo® Video 5.10.....	8
1.2.2 Microsoft H.261 a H.263.....	8
1.2.3 Microsoft Video 1 .....	8
1.2.4 MJPEG .....	8
1.2.5 MPEG-1 .....	9
1.2.6 MPEG-2 .....	10
1.2.7 MPEG-3 .....	11
1.2.8 MPEG-4 .....	11
1.2.9 ASF, WMV .....	12
1.2.10 Quicktime.....	12
1.2.11 RealVideo .....	12
1.2.12 DivX 3.11a Alpha .....	13
1.2.13 DivX 4, DivX 5 .....	13
1.2.14 XviD .....	14
1.2.15 DV.....	15
<b>2 Streamování videa.....</b>	<b>16</b>
2.1 <i>Úvod do streamování.....</i>	16
2.2 <i>Možnosti streamování .....</i>	17
2.3 <i>Kódování .....</i>	17
2.4 <i>Parametry streamování .....</i>	18
<b>3 Popis dostupných možností progresivního downloadu .....</b>	<b>22</b>
3.1 <i>Využití ve výuce.....</i>	22
3.2 <i>Komerční využití.....</i>	23
<b>4 Vkládání příkazů do mediálního streamu .....</b>	<b>Chyba! Záložka není definována.</b>
4.1 <i>Macromedia Flash .....</i>	27
4.1.1 Macromedia Flash Media Server .....	27
4.1.2 Flash aplikace a streamované video .....	28
4.2 <i>Synchronized Multimedia Integration Language.....</i>	28
4.3 <i>JavaScript.....</i>	30

4.4	<i>SMIL - základní elementy a konstrukce</i> .....	31
4.4.1	SMIL 1.0.....	31
4.4.1.1	Hlavička dokumentu .....	32
4.4.1.2	Tělo dokumentu.....	32
4.4.1.3	Časová osa a synchronizace .....	33
4.4.1.4	Kontrola obsahu.....	33
4.4.1.5	Odkazy.....	34
4.4.2	SMIL 2.0.....	35
4.5	<i>SMIL - tvorba prezentace v praxi</i> .....	36
4.6	<i>SMIL - Timed Interactive Multimedia Extensions for HTML</i> .....	37
4.6.1	Specifikace HTML+TIME.....	38
4.6.1.1	Atributy (X)HTML prvků .....	38
4.6.1.2	Integrace HTML+TIME 2.0 do (X)HTML .....	39
4.6.2	Příklady HTML+TIME pro MSIE 5.5 a 6 .....	40
4.6.3	XHTML+SMIL Profile .....	40
4.7	<i>SMIL - animace a prezentační efekty</i> .....	41
4.8	<i>Využití a zhodnocení</i> .....	42
<b>5</b>	<b>Popis dostupných aplikací pro práci se streamovanými daty</b> .....	<b>44</b>
5.1	<i>RealNetworks RealSystem</i> .....	44
5.2	<i>Windows Media</i> .....	44
5.3	<i>Apple QuickTime</i> .....	45
5.4	<i>MPEG</i> .....	45
	<b>Závěr</b> .....	<b>46</b>
	<b>Resumé</b> .....	<b>48</b>
	<b>Použité internetové zdroje</b> .....	<b>49</b>
	<b>Použitá literatura</b> .....	<b>49</b>

# Úvod

V současné době jde vývoj multimedií, výpočetní techniky a konektivity k internetu neuvěřitelně dopředu. To, co se před pár lety mohlo zdát neuvěřitelné, je dnes běžnou záležitostí. Analogové videokamery a fotoaparáty již byly vystřídány z větší části digitálními, ovšem vývoj se nezastavil a na trhu se objevují stále nové přístroje. Digitální fotoaparáty s větším rozlišením, digitální videokamery podporující nové standardy jako je např. HDV stále ve větším rozsahu naplňují trh. Tento směr vývoje je zcela jistě správný. S digitálním videem po jeho nahrání do počítače se dá dělat téměř cokoli a téměř bez ztráty na kvalitě. Je možné video upravovat, zpracovávat, převádět do jiných formátů, vypalovat na DVD a díky dnešním možnostem rychlého internetu i přenášet do celého světa. A to dokonce i realtime. Vypálené DVD disky lze pohodlně archivovat, aniž by docházelo ke ztrátě kvality.

Sledování multimedií přes internet se stává stále populárnější a to nejen jako komerční záležitost, ale trůfám si říci, že tato odvětví se stále více přesouvají i do rezortu vzdělávání. Studenti zůstávají doma, zatím co kantor přednáší v učebně. Všeobecně se takovému přenosu multimedií přes internet říká streamování.

Hlavním zájmem této práce je především objasnit pojem streamování, nastínit jeho možnosti a využití především v oblasti online prezentací. Nevyhnu se ani zmínce o různých metodách komprimace digitálního videa, přednosti a výhody jednotlivých komprimčních algoritmů, které s touto problematikou úzce souvisejí.

V této práci jsem se snažil vybrat z nepřehledného množství informací z dané oblasti jen ty nejdůležitější a postavit je do srozumitelné formy. Ve výsledku by práce měla posloužit jako návod, jak streamovat multimediální data přes internet.

# 1 Přehled možností komprimace videa

Video není nic jiného než sekvence po sobě jdoucích obrázků. Pokud bychom chtěli takové video uložit do počítače a dále zpracovávat, upravovat stříhovými programy a konvertovat do vhodných formátů (pro export na jiná média určená pro formy přehrávání mimo počítač např. DVD, či pro vysílání na internetu), narazíme na problém s velikostí surového videa.

Pokud bychom vzali v úvahu video v rozlišení 720x576 bodů (klasické DV rozlišení u normy PAL<sup>1</sup>) v barevném formátu RGB (což je standardně 3x8bitů, tedy 3 Bajty na jeden obrazový bod) a snímkovou frekvencí 25Hz. Jednoduchým výpočtem zjistíme, že k uložení jedné sekundy videa potřebujeme  $720 \times 576 \times 3 \times 25 = 31\,104\,000 \text{ B} \approx 30 \text{ MB}$ . Současné běžně používané pevné disky počítačů reálně nedosahují takové přenosové rychlosti. Navíc budeme-li počítat se 120GB diskem, pak by se na takový disk vešlo zhruba 63 minut záznamu.

Pokud bychom přesto na disk video v takovém formátu stáhli, už by nám nezbylo místo na jeho samotné zpracování. Proto je potřeba video ještě před jeho zpracováním nějakým způsobem upravit. K tomu slouží takzvané kodeky (zkratka slov KOMPRESOR-DEKOMPRESOR). Kodek je prakticky mechanismus, který snímky daného videa zakóduje (za pomoci různých kódovacích algoritmů, o kterých bude řeč v dalších kapitole) do menší podoby a při přehrávání videa jej zase v reálném čase dekoduje.

Různé kodeky se liší kvalitou, rychlostí a výslednou velikostí komprimovaného videa, která bývá většinou v poměru k nekomprimovanému originálu 1:4 – 1:100. Kodeky můžeme dále rozdělit na ztrátové z bezztrátové.

## 1.1 Bezeztrátové kodeky

### 1.1.1 RAW

RAW není vlastně kodek, ale již zmíněný nekomprimovaný formát. Pro plný PAL (720x576) má datový tok kolem 29,7 MB/s a pro poloviční PAL (352x288) má datový tok 7,2 MB/s.

---

<sup>1</sup> PAL (phase alternation line) je jeden ze standardů kódování barevného signálu pro televizní vysílání. Další jsou například SECAM nebo NTSC.

### 1.1.2 HuffYUV

Tento kodek komprimuje video s použitím Huffmanova kódování. V nejlepším případě komprimuje až na 40% původní velikosti. Zvládá kompresi obrazu v barevném formátu RGB<sup>2</sup> i YUV<sup>3</sup>. Tento kodek je velmi rychlý a je zdarma.

## 1.2 Ztrátové kodeky

### 1.2.1 Indeo® Video 5.10

Tento kodek byl vyvinut společností Intel. Má poměrně dobrou kvalitu obrazu. Lze nastavit, aby každý snímek byl klíčový<sup>4</sup>. Při nastavení kvality na 100% je výsledný obraz téměř k nerozeznání od nekomprimovaného.

### 1.2.2 Microsoft H.261 a H.263

H.261 je standard pro videokonference a videotelefonii přes ISDN. Umožňuje regulovat tok dat v závislosti na propustnosti sítě. Přenos dat je 64kbit/s nebo 128kbit/s (dva kanály ISDN). Kodek H.263 implementuje vyšší přesnost při pohybu než H.261. Jeho použití je pro monitorovací systémy a pro videokonference s velkou obrazovkou.

### 1.2.3 Microsoft Video 1

Tento kodek je standardní součástí všech operačních systémů firmy Microsoft od verze Windows 95. Kvalitou výsledného obrazu je ovšem velice špatný. I při nastavené 100% kvalitě je pozorovatelné čtverečkování a jiné nepříjemné vady v obraze. Kodek je navíc poměrně pomalý a takto zakódované video je dokonce větší než stejné video zkomprimované bezztrátovým kodekem HuffYUV!

### 1.2.4 MJPEG

Kompresní kodek MJPEG (Motion JPEG) je založen na kompresi jednotlivých snímků použitím komprese JPEG. Tento kodek má většinou volitelný kompresní poměr v rozmezí 6:1 do 16:1. Při kompresním poměru 1:8 je kvalita obrazu stále ještě velmi dobrá a datový tok se pohybuje kolem 4 MB/s a dosahuje tak dobrého poměru kvalita/velikost. Velikou předností tohoto kodeku je, že každý snímek je komprimován samostatně a je tedy

---

<sup>2</sup> Barevný model RGB neboli červená-zelená-modrá je aditivní způsob míchání barev používaný ve všech monitorech a projektorech.

<sup>3</sup> YUV je barevný model používaný v televizním vysílání v normě PAL. Model k popisu barvy používá tříprvkový vektor [Y,U,V], kde Y je jasová složka a U a V jsou barevné složky.

<sup>4</sup> Klíčový snímek je snímek, od kterého se odvíjí další snímky. Na ostatních snímcích se ukládají pouze obrazové změny oproti klíčovému snímku.



vždy klíčový. Proto je tento kodek velmi vhodný pro střih videa na počítači. Zároveň je implementován hardwarově v mnoha polo-profesionálních zachytávacích kartách a zachytávání pak funguje bezproblémově i na velmi pomalých počítačích (dostačuje CPU 300MHz). Častou nevýhodou takto hardwarově implementovaného kodeku je nemožnost přehrát zachycené video na jiném počítači bez tohoto hardware. Softwarový kodek komprimující video kodekem MJPEG je například PICVideo MJPEG Codec.

### **Výhody**

- každý snímek je klíčový, ideální pro střih
- bývá implementován hardwarově
- podpora prokládaného obrazu
- poměrně vysoká kvalita obrazu
- bývá implementován hardwarově
- 

### **Nevýhody**

- vysoké zatížení CPU
- velký datový tok

## **1.2.5 MPEG-1**

MPEG je zkratkou pro Motion Pictures Experts Group. Cílem práce této skupiny bylo standardizovat metody komprese videesignálu a vytvořit otevřenou a efektivní kompresi. Formát MPEG-1 byl dokončen v roce 1991 a jako norma přijat roku 1992 - ISO/IEC-11172. Byl navržen pro práci s videem o rozlišení 352x288 bodů a 25 snímků/s při datovém toku 1500kbit/s. Parametry komprese MPEG-1 jsou srovnávány s analogovým formátem VHS. Formát MPEG-1 se stal součástí tzv. „White Book“, což je definováno jako norma pro záznam pohyblivého obrazu na CD (74 minut videa).

MPEG komprese používá ke kompresi videa I, P a B snímky. I snímky (Intra Pictures) jsou snímky klíčové, jsou komprimovány obdobně jako MJPEG, ale navíc s možností komprimovat různé části obrazu různým stupněm komprese. P-snímky (Predicted Pictures) jsou kódovány s ohledem na nejbližší předchozí I nebo P-snímek. B-snímky (Bidirectional Pictures) jsou pak dopočítávané jako rozdílové snímky mezi nejbližším předchozím I nebo P-snímekm a nejbližším následujícím I nebo P-snímekem. Celá sekvence snímků (od jednoho I po další I snímek) se pak nazývá GOP (Group of Pictures) a standardní MPEG stream pro VCD, SVCD a DVD používá pořadí IBBPBBPBBPBBPBBPBB. Přesto MPEG stan-

dard neurčuje žádná pravidla a omezení pro vzdálenost I a P snímků. Kompresi navíc umožňuje kdykoliv ukončit GOP a předčasně tak použít další sekvenci GOP začínající snímkem I. Toto vede především ke zlepšení kvality videa. Komprimované video obsahující proměnlivé vzdálenosti mezi klíčovými snímky se pak nazývá VKI (Variable Keyframe Interval). Počet I, P a B snímků lze většinou nastavit, záleží na implementaci kompresoru. Z pohledu zabíraného místa pak I snímky zabírají nejvíce místa, po nich jsou P snímky a úplně nejméně místa zabírají snímky B. Kompresi MPEG-1 se nehodí pro střih videa z důvodu vzdálených klíčových snímků. Většina stříhových programů však umožňuje export do formátu MPEG-1. Tento kodek je totiž jeden z nejrozšířenějších formátů a lze jej softwarově přehrát téměř na každém počítači a stejně tak na 95% všech stolních DVD přehrávačích. Tento formát lze také streamovat. Bohužel v dnešní době je již tento kodek zastaralý, přesto je to nejkompatibilnější formát. Co se týče kvality je v porovnání s jinými kodeky na tom poněkud hůře, protože abysme dosáhly dobré kvality obrazu, potřebuje mnohem více bitů na kompresi než u jiných kodeků (DivX, XviD)

### **Výhody**

- vysoká podpora přehrávačů softwarových i hardwarových, kompresor i dekompresor je zdarma
- používá se pro Video CD
- vhodný i pro stream videa

### **Nevýhody**

- nepodporuje prokládané snímky
- nízká kvalita při nízkém datovém toku
- jen konstantní datový tok
- nevhodný pro střih

## **1.2.6 MPEG-2**

Po dokončení MPEG1 standardu jej začali lidé používat, a snažili se jej používat i na vyšší rozlišení. Narazili ale na několik problémů, kvůli kterému byl MPEG1 nepoužitelný. Kompresi MPEG1 zvládá komprimovat pouze celé snímky. Nepodporuje však kompresi snímků prokládaných. Formát MPEG-2 byl dokončen v roce 1994 a stal se standardem pro kompresi digitálního videa. Byl navržen tak, aby dosahoval vysílací kvality videa. Oproti MPEG-1 přináší komprese MPEG-2 podporu pro prokládané snímky, tedy

přerušky. Dále proměnlivý datový tok, což umožňuje v náročnějších scénách videa použít více bitů pro kompresi a naopak v klidnějších scénách se použije méně bitů. Samozřejmě dále podporuje i konstantní datový tok.

Při stejném datovém toku a plném rozlišení (720x576) dosahuje MPEG2 mnohem vyšší kvality obrazu než MPEG1 komprese. Nevýhodou komprese MPEG2, je na druhou stranu velmi vysoké zatížení procesoru při přehrávání, a prakticky žádný rozdíl v kvalitě oproti MPEG1 kompresi při nízkých rozlišeních. Pro streamování v nízké kvalitě je tedy vhodnější komprese MPEG1, zatímco pro plné rozlišení a vysoké datové toky zase MPEG2.

### Výhody

- používá se pro SVCD, DVD
- používá se pro digitální vysílání (DVB - Digital Video Broadcast)
- Vysoká kvalita při vysokém datovém toku (6Mbit/s a více)
- podpora proměnlivého datového toku

### Nevýhody

- pro osobní počítače nutnost hardware či software přehrávače, při softwarovém přehrávání je vysoké zatížení procesoru
- nízká kvalita při nízkém datovém toku
- nevhodný pro stříh

## **1.2.7 MPEG-3**

Pro HDTV (High Definition TV) měl být určen MPEG-3. Jeho vývoj byl ale zastaven, protože pro požadavky HDTV plně postačuje formát MPEG-2.

## **1.2.8 MPEG-4**

MPEG-4 byl vyvinut opět společností Motion Picture Experts Group. Není to již přesná definice komprese a komprimačních algoritmů, nýbrž je to množina parametrů a vlastností, které musí kompresor splňovat, aby byl MPEG-4 kompatibilní. Známe tedy různé implementace MPEG-4, které vybírají z definice MPEG-4 vždy to, co je pro daný formát vhodnější. Kodeky využívající způsoby komprese MPEG-4 jsou např. Microsoft MPEG-4 v1, v2 a v3, DivX 4, DivX 5, XviD a další

### 1.2.9 ASF, WMV

Firma Microsoft si všimla úspěchů na poli streamovaného videa, kterých dosahovaly společnosti Apple a RealNetworks svými formáty Quicktime, MOV a RM, a vyvinula vlastní formát ASF (Advanced Streaming Format), určený především pro stream videa. ASF je formát i komprese, vychází z formátu AVI a dovoluje použít pouze kompresi Microsoft MPEG4. Firma Microsoft uvedla i formát WMV, který je novější verzí ASF. Komprese ASF částečně implementuje MPEG4, nepodporuje totiž B-snímky.

#### Výhody

- vhodný pro stream

#### Nevýhody

- uzavřenost formátu, nemožnost využití jinými programy než firmy Microsoft, zakázáno převádění do jiného formátu z formátu ASF
- maximální rozlišení 352x288; kompresor zahazuje snímky aby dodržel datový tok; formát doplňuje soubor ASF o nadbytečné data, aby udržoval konstantní datový tok proudu, čímž se zvětšuje velikost celkového souboru až o 25% oproti přímé kopii video proudu

### 1.2.10 Quicktime

Quicktime je formát vyvinutý firmou Apple, který byl v dřívější době, kdy mu nekonkuroval MPEG velmi zajímavý a používaný. Je přenositelný mezi PC a Macintosh platformami, používá kompresi 5:1 až 25:1. Dnes se používá například na prezentačních CD a pro video streaming. Přesto v dnešní době již tento formát netrhá žádné rekordy a nelze jej příliš doporučit pro použití v praxi. Nutný je také přehrávač, který ale není součástí operačních systémů, a je tedy nutné si jej z webových stránek firmy Apple stáhnout. Také firma Apple tvrdí, že Quicktime je plně kompatibilní MPEG4 kodek, ale v přehrávači Quicktime verze 6.0 nebylo možné přehrát opravdový MPEG4 stream.

### 1.2.11 RealVideo

Real Video a Real System G2 jsou formáty komprese vyvinuté firmou Real Networks. Má podobné vlastnosti jako Quicktime, ale je více zaměřen na kompresi streamovaného videa.

### 1.2.12 DivX 3.11a Alpha

DivX 3.11a Alpha je nelegální a upravená verze kodeku ASF MS-MPEG4v3. Microsoft v beta verzi tohoto kodeku umožňoval ukládání videa do formátu AVI, ale ve finální verzi toto zakázal. Přesto se jednomu počítačovému pirátovi podařilo upravit finální kodek tak, aby umožňoval dále kompresi do formátu AVI. Vznikem tohoto nelegálně upraveného kodeku byly také odstraněny některé špatné vlastnosti kodeku ASF. Již nebylo omezeno maximální rozlišení na 352x288.

Tento kodek, přestože je nelegální, zahýbal světem digitálního videa na počítačích. Na 1CD se jeho pomocí podaří uložit až 1 hodina filmu ve velmi uspokojivé kvalitě. Snížením datového toku lze samozřejmě nahrát více, ale na úkor kvality. V dnešní době je ale tento nelegální kodek již překonán a není tedy důvod jej nelegálně používat.

### 1.2.13 DivX 4, DivX 5

Skupina lidí majících prsty v upraveném kodeku DivX 3.11 Alpha se rozhodla vytvořit vlastní kodek. Z výchozího projektu nazvaný OpenDivX vyšla první verze nazvaná DivX 4, která byla sice dostupná i se zdrojovými kódy, ale kvalita kodeku nedosahovala kvalit kodeku DivX 3.11a. Kodek DivX 4 podporuje několik variant komprese. Jednoprůchodová s daným datovým tokem, jednoprůchodová s danou kvalitou a dvouprůchodová. První zmíněná varianta komprese se snaží při kompresi videa dodržet daný datový tok. Mnohdy jej ale nedodrží a vytvoří kódované video mnohem větší než předpokládané. Varianta komprese s danou kvalitou pak komprimuje tak, aby kodek dosáhl dané konstantní kvality. Nevýhodou je nepředvídatelná velikost souboru. Poslední varianta je dvouprůchodová komprese. Provádí se dvěma průchody komprimovaného videa. Při prvním průchodu se analyzuje komprimované video a zapisují se získané informace do logovacího souboru. Při druhém průchodu se využívá informace z prvního průchodu a efektivněji se využívá datový tok. Pro scény s vyšší komprese DivX 4 používá I a P snímky. Podporuje také proměnlivou vzdálenost I-snímků (VKI).

#### Verze DivX 5 kodeku

- *DivX 5.0.2 Standard* - standardní verze, která je zdarma. Neobsahuje všechny vymoženosti plné verze
- *DivX 5.0.2 Pro GAIN* - plná verze, která do počítače nainstaluje software pro zobrazování internetových reklam.

- *DivX 5.0.2 Pro* - plná verze, bez reklam. Cena 30 USD

Od verze DivX 5 je již kodek uzavřený, bez zdrojových kódů. Kodek je kompatibilní s MPEG-4, komprimuje do formátu MPEG-4 Simple Profile a zvládá přehrávání předchozích verzí kodeku DivX, MPEG-4 Simple Profile, MPEG-4 Advanced Simple Profile a H.263 (videokonference). DivX 5 používá pokročilejší techniky při kompresi a oproti DivX verze 4 dosahuje zlepšení kvality až o 25% při zachování velikosti souboru.

DivX 5 má integrované některé nástroje/filtry v sobě a umožňuje tak přímo při kompresi změnit rozměry obrazu, aplikovat filtr rozprokládání, ořezat obraz a jiné. Dále implementuje algoritmy pro zvýšení komprese využitím tzv. psychovizuálního modelu. Při něm se dosahuje lepší komprese bez znatelné ztráty kvality a to díky znalostem o lidském vizuálním systému. Implementuje obousměrnou kompresi, tedy B-snímky. Dále tzv. globální kompenzaci pohybu, což je algoritmus, který optimalizuje kompresi pro panorámování, roztmívání obrazu, přibližování, náhlé změny jasu (exploze), stagnující plochy (voda) a další. Kodek také umí export čistě do MPEG-4 formátu a konverzi mezi ním a AVI formátem. Jak je vidět kodek DivX 5 toho přinesl poměrně dost, ale jeho nevýhodou je placená/reklamová plná verze.

### **Výhody**

- vysoká kvalita videa při nízkém datovém toku
- lze použít jakékoliv rozlišení dělitelné 4 až do 1920x1088
- pokročilé kompresní techniky
- podpora barevných formátů YUV a RGB
- MPEG-4 kompatibilní

### **Nevýhody**

- vysoké zatížení CPU
- přehrávatelné jen na osobních počítačích

## **1.2.14 XviD**

V momentě, kdy se OpenDivX stal uzavřeným, se toto nelíbilo některým programátorům pracujícím na OpenDivXu, vzali si zdrojové kódy, ještě otevřeného OpenDivXu, a začali vyvíjet vlastní verzi kodeku nazvanou XviD. XviD je opět MPEG-4 kompatibilní kodek a implementuje mnoho vlastností MPEGu 4, bohužel zatím nepodporuje obousměr-

né kódování (B-snímky). Kodek XviD obsahuje mnoho nastavení a k dosažení kvalitního výstupu je potřeba vědět o tomto kodeku opravdu hodně a správné nastavení kodeku je poměrně obtížné. Také umí produkovat MPEG-4 kompatibilní datové proudy.

### **Výhody**

- kodek je zdarma
- vysoká kvalita videa při nízkém datovém toku
- lze použít jakékoliv rozlišení dělitelné 4 až do 1920x1088
- velké množství nastavení kodeku
- podpora barevných formátů YUV a RGB
- MPEG-4 kompatibilní

### **Nevýhody**

- vysoké zatížení CPU
- přehrávatelné jen na osobních počítačích

## **1.2.15 DV**

Dlouho jsem váhal, kam zařadit tento kodek. V mnoha pramenech se píše, že je to kodek bezztrátový, realita je ovšem jiná. Aby dosáhl v minulé kapitole zmíněného konstantního poměru 5:1, tak dochází ke ztrátě informace. Přesto je kompresní algoritmus velmi dokonalý a obraz lze srovnávat s kompresemi bezztrátovými.

### **Výhody**

- každý snímek je klíčový
- vysoká kvalita obrazu srovnatelná s profesionálními zařízeními

### **Nevýhody**

- velký datový tok

## 2 Streamování videa

Jednoduše by se dalo říct, že streamování je jednosměrný přenos multimediálních dat po digitálních sítích v reálném čase. V reálném čase znamená, že na straně přijímače je možné okamžitě zobrazovat příchozí data bez čekání na stažení celého příspěvku.

### 2.1 Úvod do streamování

Při streamování není datový tok ukládán na disk příjemce, ale je rovnou přehráván. Z této vyplývá, že tento způsob přenosu multimediálních dat přináší jisté výhody, ale i nevýhody:

- Nemusíme čekat na stažení celého multimediálního souboru, který bývá často dosti objemný. Multimediální obsah je k dispozici téměř okamžitě. V praxi pak příjemce multimediálního obsahu nemusí čekat na stažení celého například několikahodového záznamu, ale může s jeho přehráváním začít okamžitě, popřípadě se může pohybovat po časové stopě záznamu. Je tedy možné si záznam během pár minut projít a zjistit, zda je v něm obsaženo to, co příjemce hledá.
- Při streamování nejsou data ukládána na disk, čili není možné si záznam spustit později z lokálního disku bez možnosti připojení k internetu. Existují ovšem aplikace, kterými lze streamovaná data zachytávat a ukládat na lokální disk pro pozdější použití, kdy nebudete mít přístup k síti. Takovými programy jsou například CoCSoft Stream Down nebo MPlayer.
- Pro příjem streamovaných multimediálních dat je třeba připojení určité kvality. Pro příjem zvuku v kvalitě hlasu dostačuje rychlost připojení okolo 40 kbit/s. Pro přenos kvalitního stereo zvuku (hudby) je potřeba již rychlosti připojení okolo 140 kbit/s a nevyšší nároky na přenosovou kapacitu linky jsou kladeny pro přenos obrazových dat, u kterých je potřeba nejméně rychlosti 200 kbit/s. V samotném streamování se ve finále volí kompromis mezi kvalitou a dostupností. U zvukového záznamu můžeme volit počet kanálů (mono - hlas, stereo - hudba) a bitovou rychlost (pro záznam zvuku plně dostačuje 32 kbit/s na kanál, u hudebního záznamu potom alespoň 64 kbit/s na kanál). U videa lze zvolit jednak snímkovací frekvenci (u prezentací postačuje 10 – 15 snímků za sekundu), velikost obrazu (volíme s ohledem na čitelnost či srozumitelnost záznamu) a kompresi. Pro dosažení nejlepších vý-



sledků používáme nejmodernější komprimační formáty a nastavením voleným pro konkrétní případ.

- V případě pomalého nebo kolísavého připojení může při přehrávání multimediálního streamu docházet k výpadkům či dokonce k znemožnění přehrávání. Možným řešením bývá poskytování daného obsahu ve více přenosových rychlostech, kde nižší přenosové rychlosti jsou na úkor kvality obsahu.

## 2.2 Možnosti streamování

- Režim **On-demand** – neboli na žádost, kdy multimediální obsah celý umístěný na serveru (úložišti) a klientskému počítači (příjemci) je zasílána pouze požadovaná část obsahu.
- Režim **On-line** nebo též **Live Stream** – multimediální obsah je teprve v reálném čase vytvářen na streamovacím serveru a přenášen prostřednictvím vysílacího serveru příjemcům. Jedná se o živé vysílání například televize či rádia po internetu.

Aby bylo možné data serverem streamovat, je potřeba je překódovat do formátu vhodného pro streamování – do formátu podporovaného streamovacím serverem. Nejčastěji RealMedia, WindowsMedia a QuickTime. Tyto formáty jsou kvalitativně srovnatelné a také nejrozšířenější.

Předností formátů WindowsMedia je jejich nativní podpora v OS Windows, nelze je ovšem přehrát v jiných OS. Předností formátů RealMedia je nezávislost na OS, vždy je ale třeba na straně klienta instalovat podporu pro přehrávání. QuickTime je již méně rozšířený a je vázaný jen na některé OS (Windows, Mac). Na server lze umístit i některé běžné multimediální formáty, a to MPEG-1, MPEG-2, Flash, MP3. Nevýhodou formátů MPEG je ale až 5x vyšší datový tok než u RealMedia či WindowsMedia, a to při srovnatelné výstupní kvalitě obrazu.

## 2.3 Kódování

K překódování slouží producer<sup>5</sup>. Překódovaná data se buď přímo umístí na vysílací server (režim On-demand), nebo se data můžou dekodovat v reálném čase a na vysílací server se umísťují postupně (režim On-line). U On-demand režimu je situace jednodušší, jelikož pro překódování videa nejsme tlačeni časem a můžeme si dovolit např. víceprůcho-

---

<sup>5</sup> Jedná se o výkonný počítač

dové zpracování videa – na úkor doby zpracování. Na rozdíl od On-line režimu, kde data musí být vysílacímu serveru poskytovány nepřetržitě s minimálním zpožděním, které bývá mnohdy i v řádů minut.

Pro kódování multimediálního obsahu do konkrétního streamovacího formátu se užívá program obecně nazývaný jako encoder. Pro kódování audia/video do formátů Windows-Media se používá program Windows Media encoder, pro překódování do formátů Real-Media se používá Real Producer Basic, oba tyto programy jsou na uvedených odkazech volně ke stažení.

Po nainstalování a spuštění konkrétního encoderu se navolí:

- Zdroj multimediálních dat – nejčastějším zdrojem bývá audio/video soubor, další možností je multimediální zařízení, např. mikrofon, kamera
- Parametry kódování – především se volí velikost požadovaného výstupního datového toku (udáno v kb/s) a typ datového toku (CBR/VBR tj. konstantní/variabilní), kvalita kódování (nízká/střední/vysoká, jedno/vícecestné). Možné je i upravit vlastnosti audia/video (vzorkovací frekvence a mono/stereo zvuku, rozlišení a počet snímků/s obrazu).
- Cíl překódovaných dat – nejčastějším cílem bývá uložení do souboru, při On Line režimu streamování se překódovaná data hned průběžně posílají na streamovací server.

Po překódování je možné si streamovaný obsah z lokálně uloženého souboru přehrát a tak hned posoudit dosaženou kvalitu, při nepřijatelné kvalitě je nutné upravit parametry kódování, viz níže.

Následně je třeba překódovaný soubor umístit do Vámi přiděleného adresáře na streamovacím serveru, multimediální obsah poté bude dostupný prostřednictvím URL odkazu z webu. Váš URL prefix spolu s návodem pro připojení adresáře Vám bude zaslán při registraci.

## **2.4 Parametry streamování**

Vliv volitelných parametrů kódování na výslednou kvalitu streamovaného multimediálního obsahu:

**Velikost datového toku** – Čím vyšší je datový tok, tím více se kvalita výstupního obsahu blíží kvalitě vstupní. Snahou je ale i co nejvíce omezit tento tok, aby byl přehratelný i

při pomalejším připojení klienta. Velikost datové toku se volí odděleně pro audia a video, přijatelné kvality audia lze dosáhnout už od toku 32 kb/s, u videa s rozlišením 320x240 při 30 fps může být kvalita přijatelná už od cca 200 kb/s. Čím je ale ve scéně videa více pohybu, tím obecně je i potřebný vyšší datový tok.

**Typ datového toku** – U konstantního toku je fixně stanovena jeho velikost, bez ohledu na to, kolik je ve scénách pohybu. U variabilního toku se velikost toku dynamicky přizpůsobuje množství pohybu ve scéně, takže jde o efektivnější způsob zakódování. Dočasné špičky ve variabilním toku ale mohou u pomalu připojených klientů způsobovat výpadky při přehrávání. Proto pokud to nastavení umožňuje, je vhodné u variabilního toku kromě průměrné velikosti toku nastavit i jeho min./max. velikost.

**Kvalita kódování** – Při více průchodovém kódování videa je dosaženo trochu vyšší výstupní kvality, kódování ale trvá více-násobně delší dobu. Při On-Line kódování lze použít pouze jednorůchodové kódování. Při volbě nízká/střední/vysoká kvalita kódování se volí různé algoritmy pro jeden kódovací průchod, čím vyšší požadovaná kvalita, tím více je třeba i času na zakódování.

**Rozlišení obrazu** – Čím vyšší rozlišení, tím více je zachyceno detailů v obraze, ale tím vyšší je i potřebný datový tok. Nemá smysl nastavovat vyšší rozlišení výstupu, nežli je rozlišení vstupu, při horší kvalitě vstupu naopak bývá vhodnější výstupní rozlišení zmenšit. Při On Line režimu streamování často není k dispozici natolik výkonný počítač, který by v reálném čase zvládl kódovat vyšší rozlišení.

**Vzorkovací frekvence audia** – Čím vyšší vzorkovací frekvence, tím vyšší bývá kvalita výstupního zvuku, ale tím vyšší je i potřebný datový tok. Nemá smysl nastavovat vyšší vzorkovací frekvenci výstupu, nežli je vzorkování vstupu. Pro zachycení mluveného slova stačí vzorkování 8-16 kHz, pro CD kvalitu se užívá 44 kHz.

**Počet snímků/s** – Čím vyšší počet snímků/s, tím plynulejší je výsledný obraz, ale tím jsou i vyšší nároky na datový tok. Pro nejvyšší plynulost stačí nastavit 24-30 snímků/s, při velmi malé dynamice nebo při oželení plynulosti obrazu lze omezit jejich počet až na 1-10 snímků/s.

Zpřístupnění multimediálního streamovaného obsahu uživateli je realizováno nejčastěji prostřednictvím klasického hyper-linkového odkazu na podadresář vysílajícího serveru, kde je daný obsah umístěn. Pro zajištění funkčnosti je většinou ještě třeba na straně klienta jednorázově nainstalovat podporu pro daný streamovací formát.

Využití streamovaných multimedií nachází nejvyšší využití především v oblasti prezentací, živého vysílání, vzdělávání a e-learningu, kde se nejvíce zúročuje hlavní přednost

streamování, a tou je okamžitá dostupnost multimediálního obsahu přes síť prostřednictvím webového linku.

Streamovací systém se skládá z několika částí. Data jsou streamována pomocí streamovacího serveru a přes síť doručována na klientské stanice. V případě on-demand přenosu může mít každý klient jiný požadavek a tudíž vygeneruje na serveru svůj datový tok.

V případě on-line přenosů by teoreticky bylo možné, aby sítí procházel jeden stream, který by odebírali všichni klienti. Ten způsob přenosu se nazývá multicast. Není však možné jej použít v síti internet, kde lze použít pouze unicast, při kterém každý klient na streamovacím serveru vygeneruje svůj vlastní datový tok. Síť a streamovací server je pak zatěžován v závislosti na počtu posluchačů. Jestliže například bitová rychlost jednoho toku bude 300 kbit/s (nižší kvalita videa) a připojili bychom 30 klientů, dostaneme celkový bitový tok odcházející ze serveru 9 Mbit/s.

Z výše uvedeného vyplývá, že požadavky na kvalitu a rychlost připojení nejsou pouze na straně klientů, ale také na straně streamovacího serveru. Server by tedy měl být umístěn v místě s kvalitním připojením s velmi vysokou rychlostí nejlépe na páteřní síti. Jistě také nebude rozumné, aby si každá organizace zřizovala svůj vlastní streamovací server pro svých několik videí a platila drahé vysokorychlostní připojení k internetu. Výhodnější bude sdílení jedné infrastruktury a pronajímání. Samozřejmě bude nutné pořídit výkonnější server a rychlejší připojení, ale celkové náklady budou daleko nižší než kdyby jednotlivé organizace pořizovaly vlastní techniku.

Při streamování on-demand problémy s dopravou dat ke streamovacímu serveru nebudou. Data se upraví a přes internet nahrají do úložiště streamovacího serveru (v nejhorším případě by bylo možné je zaslat do streamovacího střediska na CD, DVD nebo jiném nosiči). Horší situace nastává v případě on-line přenosů, kdy je potřeba data na streamovací server dopravit v reálném čase. Zde se ukazuje jako výhodné oddělení produceru od streamovacího serveru – principiálně by to totiž mohl být jeden počítač. V některých méně náročných aplikacích je možné takto pracovat. Producer zachycuje data ze vstupních zdrojů, v reálném čase je konvertuje do vhodného formátu s příslušnými parametry a přes síť posílá do streamovacího serveru. Ten se podobně jako při on-demand přenosu stará o distribuci ke klientům. Producer může generovat více kvalit a tím rychlostí datových toků. Zpracování multimediálních dat on-line (především videa) je velmi náročné na výkon produceru zvláště v případech, kdy je třeba generovat více než jednu kvalitu výstupu. Nároky na rychlost propojení mezi producerem a streamovacím serverem nebudou vysoké, protože spojením prochází pouze jedna kopie každé kvality.

Pro kódování multimediálního obsahu do konkrétního streamovacího formátu se užívá program obecně nazývaný jako encoder. Pro kódování audia/video do formátů Windows-Media se používá program Windows Media encoder, pro překódování do formátů Real-Media se používá Real Producer Basic, oba tyto programy jsou na uvedených odkazech volně ke stažení.

### 3 Možnosti využití streamovaných dat

Využití streamování multimediálních dat má široké využití. Multimediální obsah se dá sdílet nejen přes internet, ale i přes lokální síť a to v mnohem vyšších kvalitách díky vyšší propustnosti sítě.

#### 3.1 Využití ve výuce

Elektronická podpora výuky je v dnešní době velice často využívána. Studenti i učitelé mají z tohoto způsobu výuky mnoho výhod (pro studenty je to např. studium libovolným tempem kdykoli a kdekoli, pro učitele je to, po vynaložení jistého počátečního úsilí, pomoc a ulehčení práce např. při opravování úkolů). Elektronickou podporu výuky lze zevrubně rozdělit na dvě části - online a offline výuku. Offline přístup je založen na distribuci výukových materiálů na určitých datových médiích (v současné době nejčastěji na CD), zatímco online výuka nejčastěji využívá funkcí implementovaných v systémech pro řízení výuky (LMS - Learning Management System). Tyto systémy jsou softwarové aplikace určené pro řízení a dohled všech aktivit vztahujících se k výukovému procesu organizace. Mezi základní vlastnosti LMS patří: administrace uživatelů a tříd, distribuce výukových materiálů, automatické vyhodnocování testů, podpora různých komunikačních nástrojů (např. diskuzní fórum, e-mail, videokonference) a další.

Základem elektronické podpory výuky jsou samozřejmě elektronické výukové kurzy. Aby byl výukový kurz úspěšný, je nutné studenta zaujmout; student nesmí být jen pasivní příjemce informací, ale musí být průběžně aktivizován a musí být zavedena jistá forma zpětné vazby. Toho lze docílit pomocí různých interaktivních prvků ve výukovém kurzu (např. slepé mapy) nebo jinými prostředky (např. diskusní fórum, chat).

Tato zpráva je zaměřena na možnosti využití streamovaného videa v elektronické podpoře výuky, zejména ve spolupráci s dalšími typy médií (obrázky, animace a text), protože samostatné přehrávání streamovaného videa nebývá problém. Příkladem takovéto spolupráce mohou být např. přednášky, které jsou doprovázeny prezentacemi a animacemi. Všechny níže uvedené metody jsou hodnoceny s ohledem na elektronickou podporu výuky a na možnosti distribuování pomocí systému pro řízení výuky, protože tyto systémy mívají jistá omezení, která mohou znemožnit použití dané metody.

Pro zhodnocení vhodnosti použití některé z metod v našich podmínkách jsem uvažoval několik kritérií (vycházel jsem aktuálních podmínek, takže např. na CESNETu je již provozován streamovací server, a proto není potřeba pořizovat další).

Softwarové a hardwarové požadavky. V tomto kritériu jsou uvažovány požadavky na softwarové a hardwarové vybavení potřebné pro poskytování synchronizace streamovaného videa. Dále je třeba uvažovat také potřebné programové vybavení na straně klienta nutné pro využití dané metody.

Tvorba kurzů. Sebelepší metoda spolupráce videa s ostatními médii nemá šanci na úspěch, pokud tvorba materiálů využívající danou metodu bude tak náročná, že ji uživatelé nebudou používat. Proto je potřeba, aby tvorba výukových materiálů byla po nainstalování potřebného programového vybavení byla co nejjednodušší.

Podpora v prohlížečích. Výukové materiály jsou distribuovány studentům prostřednictvím videostreamu přes Internet, kde si je může student prohlédnout. Studenti si mohou prohlížet videa na mnoha systémech, a proto je důležitá podpora použité metody alespoň v nejpoužívanějších prohlížečích.

### **3.2 Komerční využití**

V dnešní době již existuje plno komerčních serverů, kde kdokoli může publikovat své videonahrávky. Za tuto situace může především pokročilá technologie v oblasti zpracovávání digitálního videa a počítačových sítí, kdy většina lidí vlastní hardware dostačující ke zpracovávání digitálních videí a má dobrou konektivitu pro publikování a sledování streamovaných dat.

Mezi nejznámější a nejrozšířenější servery pro sdílení videa jsou např.:

- Youtube.com
- Metacafe.com
- Video.google.com
- iFilm.com
- putfile.com

Také i spousta TV stanic začala v posledních letech vysílat svůj program pomocí streamu přes internet. Ne jen, že většina stanic podporuje realtime sledování programu, ale některé stanice si vytvářejí i vlastní archív pořadů, které je možné sledovat i mimo jejich běžný vysílací čas. Některé nekomerční televize vysílají dokonce pouze přes internet. Mezi nejrozšířenější televize u nás, které vysílají přes internet, jsou např. Óčko či ČT24.

### 3.3 Vlastní stream – ukázka

V předchozích kapitolách rozebírám problematiku digitálního videa a jeho použití pro streamování pouze teoreticky. V této podkapitole uvedu praktický příklad realizace streamovaného videa a způsob jeho prezentace na www stránkách.

#### 3.3.1 Vhodný formát pro streamování

Aby bylo možné video streamovat, je k tomu zapotřebí streamovací server s patřičnou aplikací, která bude video stream vytvářet. Z mého pohledu vyšel v dnešní době nejlépe formát WMV ne z hlediska kvality či velikosti video souboru, ale čistě jen z hlediska dostupnosti. Je doba, kdy většina uživatelů internetu používá právě operační systém Windows od firmy Microsoft, který je již přímo dodáván s přehrávačem video souborů WMV – Windows Media Player.

Video zvolené pro streamování jsem tedy převedl do formátu WMV. Zvolil jsem dvě různé kvality. První datový tok 500kbit/s a druhý 100kbit/s.

Ted' jsem se ocitl před problémem, kde rozchodit streamovací server. Pro zkušební účely samozřejmě nezáleželo na rychlosti připojení serveru, ale již jsem přemýšlel, co by bylo nejvhodnější do budoucna pro školu, kdyby se rozhodla poskytovat studentům video streamy přednášek. Proto jsem se obrátil přímo na CESNET<sup>6</sup> a možnosti využití jejich videoserveru<sup>7</sup>. Jejich reakce byla velice vstřícná a video ukázky mi na server umístili. Dále je CESNET schopný spolupracovat se školou a poskytovat EPI s.r.o. streamy jejich přednášek. Kromě možnosti využívat jejich server pro streamování On-demand, kdy je vytvořený video soubor uložený na jejich serveru, podporují i možnost Live streamu, kdy se streamovaná data posílají na CESNET, odkud jsou dále publikována. Existují dvě možnosti, jak takto stream publikovat a vhodnost použití jednoho či druhého závisí na možnostech serveru.

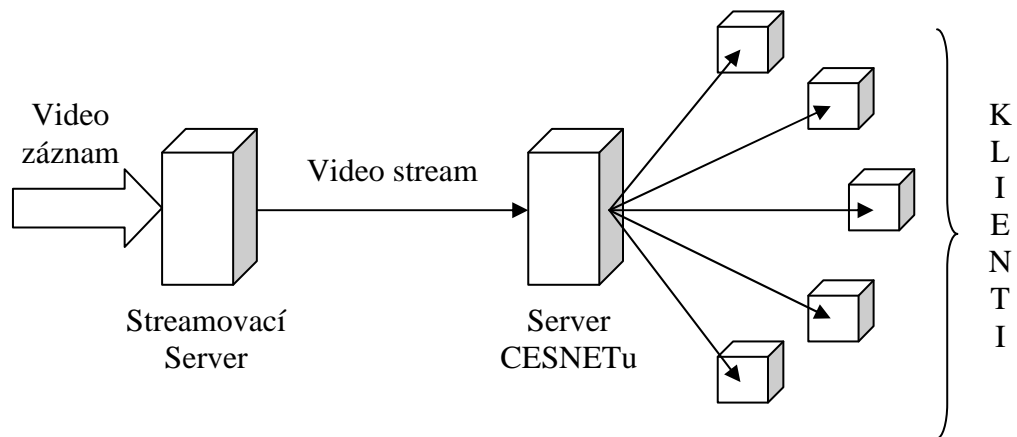
**První možností** (Obrázek 1.) je vytvářet stream již na vysílací straně (což klade vyšší nároky na hardware) a vytvořený stream pak jen přes CESNET přeposílat (řetězit) klientům. Toto se dá využít například u zmiňovaného WMV.

---

<sup>6</sup> Provozovatel páteřní sítě v České republice ([www.cesnet.cz](http://www.cesnet.cz)).

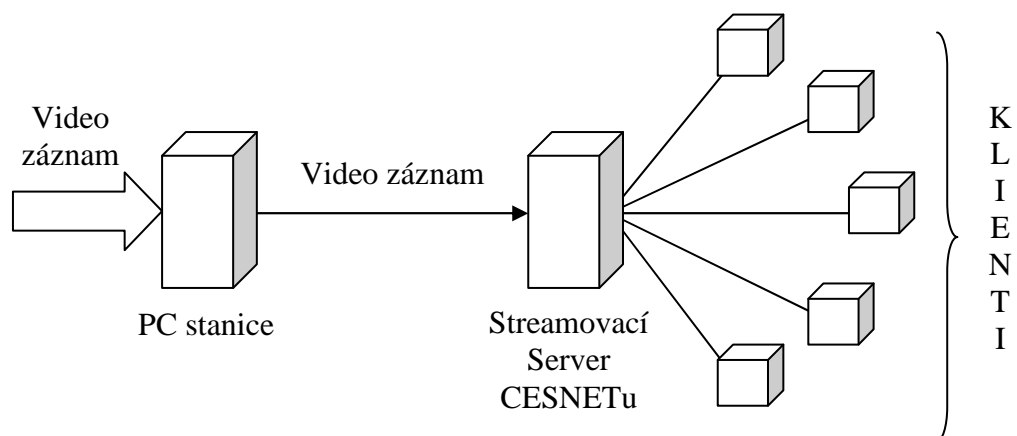
<sup>7</sup> Streamovací server CESNETu ([videoserver.cesnet.cz](http://videoserver.cesnet.cz)).





Obrázek 1.

**Druhá možnost** (Obrázek 2.) je posílat surová video data na video server CESNETu, kde budou následně překódována a poskytována jako stream klientům. V tomto případě již slouží počítač na vysílací straně pouze jako prostředník a nejsou kladeny tak vysoké požadavky na hardware.



Obrázek 2.

### 3.3.2 Spuštění streamu na www stránkách

Video stream na www stránkách spustíme prostřednictvím vloženého objektu, v našem případě media playeru. Slouží k tomu (X)HTML tag **OBJECT**, vložený do kódu stránky mezi **BODY** nebo mezi **HEAD** tagy. **OBJECT** je párový tag a má mnoho volitelných parametrů, které přímo pracují s daným vloženým objektem a ovlivňují nejen jeho vizuální podobu na stránkách, ale i jeho chování a funkci. K základním vlastnostem tagu **OBJECT** patří **TYPE** a **DATA**, které nám definují formát streamovaných dat a cestu ke streamovanému souboru. Základním kódem pro vložení video streamu může být:

```
<OBJECT type="video/x-ms-wmv"  
DATA ="http://server/stream.wmv" >  
</OBJECT>
```

K tagu OBJECT se vztahují i další důležité vlastnosti a to WIDTH a HEIGHT, které definují rozměry videa:

```
<OBJECT type="video/x-ms-wmv"  
WIDTH=" 320 " HEIGHT=" 240 "  
DATA ="http://server/stream.wmv" >  
</OBJECT>
```

Mezi tagy OBJECT můžeme i vkládat nepárové tagy PARAM, kterými přímo ovlivňujeme chování vloženého objektu. U media pleyeru se dají použít např. následující:

```
<OBJECT type="video/x-ms-wmv"  
WIDTH=" 320 " HEIGHT=" 240 "  
DATA ="http://server/stream.wmv" >  
<PARAM name="AUTOPLAY" VALUE="false" >  
<PARAM name="CONTROLLER" VALUE="true" >  
</OBJECT>
```

Parametr AUTOPLAY nám říká, zda bude video spuštěno hned po jeho načtení, nebo zda se bude čekat na spuštění videa uživatelem. Parametr CONTROLLER rozhoduje, zda se v objektu videa budou aktivní ovládací prvky přehrávače či nikoli.

## 4 Vytváření online prezentací

### 4.1 Macromedia Flash

Využití aplikací vytvořených v prostředí Macromedia Flash streamovaného videa a dalších médií je z pohledu elektronické podpory výuky dobrá volba, protože v něm lze vytvořit efektní multimediální a interaktivní výukové materiály a navíc lze přehrávač Flash aplikací nainstalovat pro téměř všechny prohlížeče, takže vytvořené aplikace si může student prohlédnout kdekoli. Streamované video je možné v aplikacích Flash využít díky relativně novému produktu firmy Macromedia, který se nazývá Flash Media Server.

#### 4.1.1 Macromedia Flash Media Server

Macromedia Flash Media Server (dále jen FMS) kombinuje v sobě mnoho funkcí, které umožňují vytvářet mnoho různých interaktivních multimediálních aplikací pracujících v reálném čase (např. video na přání, videokonference), jejichž výhodou je, že tvoří kompaktní celek a nenastávají případy, kdy se např. video otevře v novém okně, či samostatně v přehrávači. S ohledem na téma této zprávy je ale pro nás podstatná podpora funkcí streamovacího serveru. Pro streamování videa používá FMS svůj vlastní kodek (soubory mají příponu flv), který je kvalitativně srovnatelný s běžně používanými kodeky. Pro překodování videa do tohoto proprietárního formátu lze použít kodér dodávaný s vývojovým prostředím Flash. Při vlastním přenosu FMS detekuje kvalitu připojení klienta a podle výsledku automaticky přizpůsobuje kvalitu přenášeného videa a tím i potřebnou přenosovou rychlost.

Flash Media Server může být provozován na následujících operačních systémech:

- Windows 2000 Server;
- Windows 2003 Server Standard Edition;
- Linux Red Hat Enterprise Version 3.0;
- Linux Red Hat Enterprise Version 4.0.

Hardwarové požadavky na server jsou následující:

- Pentium 4, 3.2GHz (Dual Xeon nebo lepší);
- 1 GB paměti RAM;
- 1 Gb/s ethernetová síťová karta.

### **4.1.2 Flash aplikace a streamované video**

Pro vytvoření výukového kurzu, který bude obsahovat Flash animace či aplikace, je potřeba mít nainstalovaný software Macromedia Flash. S využitím funkcí výše zmíněného Flash Media Serveru je možné v těchto aplikacích jednoduše spouštět streamovaná videa, která jsou umístěna na FMS. Současně s tím je, díky již vytvořeným komponentám, možné toto video snadno ovládat (spouštět, zastavovat, spouštět od jisté pozice apod.) Je tedy možné všechny animace a aplikace provázat s odpovídajícím videem a reagovat tak na aktuální pozici videa nebo změnit pozici videa podle interakce uživatele.

Použití této metody v elektronické podpoře výuky oproti ostatním zde zmíněným metodám má jednu velkou výhodu: vlastní Flash aplikaci je možné spustit téměř ve všech typech prohlížečů. Výhodou je také to, že celá aplikace a tedy i veškerá synchronizace je prováděna v rámci jednoho objektu (Flash aplikace), takže si Flash přehrávač veškeré řízení provádí sám. Nevýhodou je, že pro tvorbu vlastní aplikace je potřeba mít nainstalován program Macromedia Flash a pro rychlou tvorbu výukových kurzů je potřeba dobrá znalost tohoto prostředí. Další drobnou nevýhodou je to, že video, které má být streamováno musí být ve formátu flv. Ale protože se s Macromedia Flash dodává i kodér do flv formátu, který umožňuje překodovat formáty asf, avi, dv, mov, mpeg a wmv, považuji tuto nevýhodu za drobnost. S tím souvisí i to, že nelze použít běžné streamovací servery, ale musí být použit FMS.

## **4.2 Synchronized Multimedia Integration Language**

Synchronized Multimedia Integration Language (SMIL) je jazyk vydaný konzorciem W3C, který je specializovaný na multimédia. První specifikace SMIL 1.0 byla vydána v roce 1999. SMIL se snaží odstranit nedostatky běžných HTML stránek (jejich "statičnost") a umožňuje tvorbu interaktivních aplikací a animací. SMIL umožňuje integrovat multimediální objekty do HTML stránek. Pracuje již s vytvořenými multimediálními objekty a "jen" popisuje jejich pozici, dobu zobrazení a jejich chování. Základní funkce jazyka lze shrnout do následujících bodů:

- přesný popis vzhledu prezentace a umístění jednotlivých objektů;
- popis změn prezentace v čase;
- popis interaktivity objektů;
- parametrizace prezentace.

SMIL je značkovací jazyk a celý je založen na specifikaci XML. Z toho vyplývají jistá specifika:

- nutnost rozlišovat malá a velká písmena;
- všechny tagy musí být řádně ukončeny ( <tag> ... </tag> nebo <tag />).

SMIL je vybaven relativně malým počtem různých XML elementů. Přesto je s ním možné vytvořit bohaté interaktivní multimediální prezentace. Kořenovým uzlem je tag <smil>. Ten obsahuje dvě části: <head> a <body>.

Sekce <head> popisuje vzhled prezentace, definuje možná umístění objektů a charakterizuje různé možnosti zobrazování a skrývání mediálních objektů. Dále mohou být v této části umístěna meta-data (např. informace o autorovi prezentace, době platnosti, klíčová slova).

V sekci <body> jsou pak uvedeny všechny použité objekty a jejich vzájemná synchronizace - objekty se mohou buď zobrazovat paralelně nebo sekvenčně (tagy <par> a <seq>). Tyto možnosti se dají různým způsobem kombinovat.

SMIL rozlišuje několik typů médií. Typy médií, používaný tag a používané formáty pro dané médium (možnosti použití závisí na použitém přehrávači) jsou uvedeny v následujících bodech:

- text (<text>) ... txt, html;
- streamovaný text (<textstream>) ... jen pro Real Player;
- obrázek (<img>) ... gif, jpg, png;
- animace (<animation>);
- zvuk (<audio>) ... wav, au, mp3;
- video (<video>) ... mpg, avi, mov, rm.

SMIL umožňuje také řízení chodu prezentace pomocí událostí. To umožňuje určitý objekt spustit (ukončit) v případě, že jiný objekt byl spuštěn, ukončen nebo že na něj uživatel kliknul. Použití událostí se definuje atributy begin a end u odpovídajících objektů.

Další funkcí jazyka SMIL je možnost přizpůsobit prezentaci určitým vstupním parametřům. Těmi parametry mohou být: použité rozlišení a barevná hloubka nastavení zobrazení, rychlost připojení k Internetu, jazyk uživatele a další. Tato parametrizace umožňuje

např. přehrávat videa odpovídající rychlosti připojení, zobrazit texty v jazyce, kterému uživatel rozumí, a mnoho dalších možností, které vylepšují kvalitu celkové prezentace.

Prezentace vytvořené pomocí jazyka SMIL mohou být přehrávány buď pomocí java apletu přímo v prohlížeči (v tomto případě jsou zde omezení týkající se možných zobrazovaných formátů - např. nelze zobrazit videa) nebo pomocí externího programu (např. Real Player). Tento externí program může být implementován jako plug-in v prohlížeči a pak lze přehrávat prezentace také přímo v okně prohlížeče.

Díky tomu, že jazyk SMIL pracuje s jednotlivými objekty jako s celkem (nezajímá se o obsah jednotlivých objektů) a že je popisován jazykem založeným na XML, lze vytvořené prezentace velice snadno měnit a pro tvorbu a editaci prezentací si lze vystačit s obyčejným textovým editorem (to samozřejmě neplatí pro úpravu samotných objektů). Oproti předchozí metodě je tu ale větší problém s rozšiřitelností. V současné době existuje Real Player pro většinu operačních systémů, takže jediným problémem zůstává, že v některých případech se musí prezentace otvírat v externím okně, což není pro výukový kurz nejlepší řešení.

### **4.3 JavaScript**

JavaScript je skriptovací programovací jazyk, který je zpracováván a prováděn na straně klienta, a umožňuje řídit chování prohlížeče. Vkládá se buď přímo do HTML kódu nebo se zapisuje do externích souborů, na které je pak v HTML kódu odkaz. Základní vlastnosti JavaScriptu jsou:

- pracuje pouze na straně klienta (má omezené funkce - např. nemůže přistupovat k souborům);
- rozlišuje malá a velká písmena;
- má základní podporu objektového programování;
- je interpretovaný - nemusí se kompilovat;
- nepoužívá typovou kontrolu.

S využitím JavaScriptu lze docílit podobných výsledků jako při použití jazyka SMIL a java apletu. Navíc je při této kombinaci možné použít i video, které je spouštěno přehrávačem (je ale potřeba, aby bylo možné tento přehrávač ovládat pomocí JavaScriptu).

Při využití JavaScriptu je největší problém se správnou funkcí napsaného kódu v různých prohlížečích. Zatím se podařilo úspěšně propojit streamované video s dalšími typy

médií jen v Internet Exploreru, což je i přes to, že je tento prohlížeč nejčastěji používaný, nepostačující. Proto by se měly najít možnosti pro zobrazování v jiných prohlížečích. Další nevýhodou používání JavaScriptu je náročná úprava ve výsledné prezentaci. Nejlepším řešením je vynaložení určitého počátečního úsilí a vytvoření API rozhraní, které by zajišťovalo správnou funkci ve většině prohlížečích a zároveň by umožňovalo jednoduše vytvářet a měnit požadované prezentace. Při použití JavaScriptu není potřeba speciální programové vybavení a je možné používat jednoduchý textový editor.

#### **4.4 SMIL - základní elementy a konstrukce**

V předchozí kapitole jsem se věnoval jeho podpoře v různých programech a teoreticky jsem přiblížil multimediální technologii SMIL. Tentokrát bych vám rád přiblížil některé ze základních prvků jazyka SMIL.

##### **4.4.1 SMIL 1.0**

Nejprve se podíváme na specifikaci SMIL 1.0. Jde o velmi jednoduchý jazyk, který sestává z nemnoha značek. Nejdůležitější částí SMIL dokumentu je DTD, která vypadá následovně:

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 1.0//EN"  
"http://www.w3.org/TR/REC-smil/SMIL10.dtd">
```

Kořenovým elementem dokumentu je prvek smil. SMIL dokument je rozdělen do dvou částí (podobně jako dokument (X)HTML). První část je tvořena elementem head a druhá elementem body. Výsledný dokument může mít následující podobu:

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 1.0//EN"  
"http://www.w3.org/TR/REC-smil/SMIL10.dtd">  
  
<smil>  
  <head>  
    hlavička dokumentu  
  </head>  
  <body>  
    tělo dokumentu  
</body> </smil>
```

#### 4.4.1.1 Hlavička dokumentu

Hlavička dokumentu obsahuje obecné informace a nastavení celého dokumentu. V hlavičce se nastavuje pozadí, rozměry prezentace, pozice jednotlivých prvků a podobně. Kromě toho může element head obsahovat několik dalších elementů:

- meta - použití elementu je podobné jako v dokumentech (X)HTML
- layout - element layout nastavuje rozvržení dokumentu

Pro určení rozvržení dokumentu může element layout obsahovat další dva prvky - root-layout a region. Element root-layout nastavuje vlastnosti celého dokumentu pomocí několika atributů - "background-color" (barva pozadí), "height" (výška okna), "width" (šířka okna), "title" (titulek). Element region vytváří imaginární obdélníky, do kterých bude později možné nahrát multimédia. Zkrátka u vloženého obrázku nastavíte, do kterého regionu se má nahrát. Prvek region má tyto atributy:

- id - identifikátor regionu
- background-color - barva pozadí
- left - pozice od levého okraje okna
- top - pozice od horního okraje okna
- height - výška regionu
- width - šířka regionu
- z-index - tento atribut řeší překrývání dvou regionů (stejně jako v CSS)

Elementy region i root-layout jsou prázdné elementy.

#### 4.4.1.2 Tělo dokumentu

Tělo dokumentu je tvořeno elementem body. Následující elementy vkládají jednotlivá média:

- img – obrázek
- audio – zvuk
- video – video
- text - textový objekt (HTML, TXT)



Pro tyto elementy existuje mnoho společných atributů:

- src - URL zdroje
- alt – popisek
- dur - trvání v časové ose
- begin - začátek v časové ose
- end - konec v časové ose
- region - id regionu, ve kterém se má objekt zobrazit
- top, left, height, width – rozměry

#### 4.4.1.3 Časová osa a synchronizace

Atributy "begin", "dur" a "end" pracují s časovou osou. Časová osa začíná nulou (od spuštění dokumentu). Atribut "begin" určuje sekundu, od které má být objekt zobrazen. Atribut "dur" určuje dobu trvání objektu, případně lze také použít atribut "end", který nastavuje sekundu ukončení objektu.

Jednotlivé objekty je možné seskupovat a následně synchronizovat pomocí elementů seq a par, které je možné i vnořovat.

- seq - definuje sekvenční zobrazení (pokud obsahuje více jak jeden objekt, zobrazuje objekty postupně, jeden po druhém, přičemž atribut "begin" nastavuje časovou prodlevu od předchozího objektu)
- par - definuje paralelní zobrazení objektů (objekty budou zobrazeny současně, přičemž atribut "begin" nastavuje časovou prodlevu od spuštění prezentace)

#### 4.4.1.4 Kontrola obsahu

Kontrolou obsahu se rozumí rozhodnutí o vhodném médiu. SMIL je schopný zjišťovat vlastnosti klientského počítače (jazyk, rozlišení a další) a závisle na zjištěních poskytovat alternativy. Obecný zápis vypadá takto:

```
<switch>
  <element hodnotici_atribut="hodnota" />
  <element hodnotici_atribut="hodnota2" />
</switch>
```

Použitelných atributů existuje několik:

- system-bitrate - rychlost připojení v bps (28 800, 56 000...)
- system-language - systémový jazyk (en, sk, cz...)
- system-screen-size - rozlišení monitoru (640x480, 800x600, 1024x768...)
- system-screen-depth - barevná hloubka monitoru

Příklad nakonec, určení jazyka klienta:

```
<switch>
  <video src="video-cz.mpg" system-language="cz" />
  <video src="video-sk.mpg" system-language="sk" />
  <video src="video-fr.mpg" system-language="fr" />
  <video src="video-en.mpg" />
</switch>
```

Konstrukce zjistí, zda se má přehrávat česky, slovensky nebo francouzsky a pokud ani jedna možnost nevyhovuje, bude použit anglický snímek.

#### 4.4.1.5 Odkazy

Použití elementu `a` je velmi obdobné použití v (X)HTML, hodnotou atributu "href" je URL nebo "id" jiného objektu. SMIL přidává atribut "show", který může nabývat hodnot "replace", "new" a "pause". V případě použití "replace" bude aktuální prezentace pozastavena a nahrazena novou, při použití "new" bude prezentace otevřena v novém okně a původní bude pokračovat dál, při použití "pause" se nová prezentace otevře v novém okně a původní je pozastavena. Obsahem elementu `a` může být některý z objektů (img, video, audio...) nebo synchronizační elementy (par a seq).

```
<a href="http://www.tilt.cz" show="new">
  
</a>
```

Druhý možný odkaz se vytváří elementem `anchor`, který se podobá klikacím mapám v HTML. Atribut "coords" vymezuje obdélníkovou oblast, určenou hodnotami "top", "left" (souřadnice horního rohu) a "bottom", "right" (dolní roh).

```

  <anchor href="http://www.tilt.cz" show="new" coords="10,
10, 100, 100" />
</img>
```

#### 4.4.2 SMIL 2.0

Podle všeho, naučit se používat SMIL 1.0 není žádný zvláštní problém, stačí si jen umět spočítat na prstech časy všech objektů a výsledek je zaručen. Bohužel rozdíl mezi specifikací SMIL 1.0 a SMIL 2.0 je propastný. V novější specifikaci přibylo mnoho stránek.

SMIL 2.0 přidává především moduly pro interaktivitu (uživatel může reagovat na prezentaci - měnit parametry objektů) a animace (změna barvy, parametru, pohyb po křivce). Závažnou změnou je modularizace, tedy rozdělení prvků SMILu do skupin - modulů, což umožňuje použít vybrané vlastnosti SMILu k doplnění jiného formátu. Kupříkladu animační moduly jsou použity v SVG. SMIL 2.0 obsahuje celkem 45 modulů rozdělených do deseti skupin:

- Animation Modules: Moduly obsahující elementy a atributy k začlenění animace do dokumentu. Některé animační moduly přebírá SVG (samozřejmě s jistými odlišnostmi).
- Content Control Modules: Moduly obsahující atributy a elementy pro kontrolu obsahu (rozhodovací konstrukce). Jde o moduly, které zjišťují informace o systému návštěvníka - rychlost připojení, rozlišení, jazyk a podobně. Závisle na zjištěných datech je možné poskytnout návštěvníkovi odpovídající obsah, například audio v jím preferovaném jazyce. Část tohoto modulu byla obsažena již ve SMIL 1.0, a to v podobě elementu switch pro rozhodovací konstrukci.
- Layout Modules: Moduly pro návrh (rozložení) prezentace. Obsahem jsou zejména elementy root-layout, layout a region. Tento modul nepopisuje pouze rozvržení dokumentu, ale také hlasitosti audio a video médií. Tento modul popisuje spíše vzhled než sémantiku.
- Linking Modules: Moduly určené k tvorbě odkazů (ze SMIL 1.0 přetrvávají elementy anchor, a).
- Media Object Modules: Moduly sloužící k začlenění médií do dokumentu (audio, video, img a mnoho dalších elementů).
- Meta-information Modules: Metainformační moduly reprezentované podobně jako v (X)HTML elementem meta a nově elementem metadata, který pracuje s RDF.

- Structure Modules: Strukturní moduly popisují rozčlenění dokumentu (elementy smil, head, body a jejich atributy).
- Timing and Synchronization Module: Časovací a synchronizační moduly (par, seq a další nové elementy a atributy).
- Time Manipulations Module: Moduly pro změny působení času. Obsahem jsou například atributy, které nastavují rychlost přehrávání médií - zrychlení či zpomalení oproti normálu.
- Transition Effects Module: Modul pro tvorbu přechodů (změna barvy v určitém časové úseku).

Výsledkem použití všech modulů je velmi efektivní multimedialní prezentace srovnatelná třeba s často používaným Power Pointem.

#### **4.5 SMIL - tvorba prezentace v praxi**

V této kapitole se budeme věnovat tvorbě multimedialní prezentace se vším, co k ní patří, na praktickém příkladě. Využijeme při tom většinu informací, které jsme shromáždili v předchozích kapitolách.

Před tím, než webdesignér začne s tvorbou prezentace, měl by se rozhodnout, jaký formát zvolí. My pomíneme program PowerPoint a další jemu podobné, takže nám zůstanou dvě možnosti:

Vytvořit prezentaci v jazyce SMIL. Bude ji možno zobrazit v některých multimedialních přehrávačích (Apple Quick Time Player, Real Player, Adobe SVG Player).

Použít formy HTML+TIME. Tím vznikne prezentace, kterou bude možné zobrazit v internetovém prohlížeči (Microsoft Internet Explorer 5.5 a vyšší).

Rozhodnutí mezi těmito formáty je poněkud zúžené faktem, že každý typ prezentace je určen pro jiné zařízení. Můžeme se však rozhodovat alespoň závisle na rozšíření přehrávačů a prohlížečů. Microsoft Internet Explorer 6.0 je momentálně nejrozšířenějším prohlížečem. Oproti tomu Apple Quick Time Player je obvyklým přehrávačem pro platformu Apple, zatímco Real Player moc rozšířený není (jde o obdobu Winampu či Windows Media Playeru). Dále je třeba si uvědomit, kolik uživatelů je schopno zobrazit prezentaci vytvořenou v PowerPointu - není jich sice tolik, ale prohlížeč pro tento formát je dostupný zdarma (podporuje verze PowerPoint 97, 2000, 2002 a 2003).

Potrženo a sečteno, většině uživatelů vyjdete vstříc formátem HTML+TIME, avšak bude-li existovat též obdoba ve formátu SMIL, vaše prezentace bude čitelná pro ještě větší

část uživatelů. A o to se dnes pokusíme, vytvoříme krátkou prezentaci ve SMILu a následně ji vytvoříme též v HTML+TIME.

Prezentace ve SMILu a prezentace v HTML+TIME se neliší jen koncovým zařízením, ale existují mezi nimi ještě další důležité rozdíly. Prezentace ve SMILu musí být validní, jinak ji přehrávač není schopen přehrát. To neplatí pro HTML+TIME v Microsoft Internet Exploreru, i když ani ta by samozřejmě neměla obsahovat chyby. Dalším rozdílem jsou textové prvky, SMIL pracuje s textem jak s objektem, proto musí být veškerý používaný text uložen s koncovkou .txt, případně .html (webová stránka, avšak ta je pro některé přehrávače nečitelná). HTML+TIME pracuje s textem naprosto normálně. Takže za fungující prezentací ve SMILu se skrývá více práce než za toutéž prezentací ve formátu HTML+TIME, je však kvalitnější a do budoucna i přístupnější.

#### **4.6 SMIL - Timed Interactive Multimedia Extensions for HTML**

Timed Interactive Multimedia Extensions for HTML, zkráceně HTML+TIME, je výsledkem spolupráce společností Microsoft, Macromedia, Compaq/Dec a Digital Renaissance. HTML+TIME je rozšířením SMILu do standardních prohlížečů, především Microsoft Internet Exploreru 5 a vyšších. Tento formát zapojuje časování a synchronizaci do dokumentů (X)HTML.

HTML+TIME 1.0 staví na specifikaci SMIL 1.0. Později se objevila specifikace HTML+TIME 2.0, která je podporována v MSIE 5.5. Tato specifikace staví na HTML+SMIL language profile podle W3C. HTML+TIME 2.0 zapojuje do prohlížeče některé moduly SMILu 2.0, zejména Timing and Synchronization module, Animation module, Media Object module a Integration module.

HTML+TIME se liší od SMILu v několika bodech:

- některé prvky a atributy chybí, protože jsou zbytečné - zejména jde o vizuální část SMILu, kterou nahrazuje CSS
- některé prvky jsou přidány
- integrace do zdrojového kódu stránky se provádí odlišným způsobem

HTML+TIME úspěšně integruje SMIL do některých prohlížečů, což umožňuje jeho efektivní použití ve firemních sítích. V budoucnu lze snad předpokládat rozšíření HTML+TIME i do dalších produktů.

## 4.6.1 Specifikace HTML+TIME

HTML+TIME vypouští některé elementy z původní specifikace SMIL 1.0, z důvodu zbytečnosti:

- smil - v HTML je nahrazen elementem html
- layout - rozvržení je prováděno pomocí CSS
- region - nahrazen CSS
- root-layout - nahrazen CSS
- text - text není v HTML tvořen objektem
- anchor - obrázkové mapy nejsou podporovány

Dále mizí také řada atributů (zejména v XHTML), které jsou nahrazeny CSS - top, left, height, width, background-color, z-index a další. Zato se objevuje několik nových atributů určujících chování dokumentu (beginEvent, endWith, endEvent, repeatDur, viz dále). Mnoho původních (X)HTML prvků je ale doplněno o atributy SMILu, aby s nimi bylo možno manipulovat na časové ose (například p, div, span, h1-6, strong, em a další).

Nakonec byly původní SMIL elementy (už jich nezbylo moc) doplněny o namespace t (t:seq, t:par, t:media, t:img, t:video), jejich funkce však zůstala stejná. Rovněž atributy byly ve specifikaci HTML+TIME doplněny o namespace t (t:dur, t:src, t:begin, t:end), avšak pokud není použit, žádné problémy nenastávají, spíše naopak, při jeho použití některé věci špatně fungují.

### 4.6.1.1 Atributy (X)HTML prvků

Nyní se podíváme na význam nových atributů (X)HTML prvků. Z původních čistě (X)HTML atributů je nejdůležitějším atribut id, který umožňuje absolutní pojmenování prvku a pozdější odvolání na něj.

- beginWith - hodnotou je id elementu, který se má zobrazit ve stejnou chvíli
- endWith - hodnotou je opět id elementu, který se skryje ve stejnou chvíli
- beginAfter - id elementu, se kterým se začne manipulovat po skončení manipulace s aktuálním elementem
- beginEvent - hodnotou je id (form.onSubmit) nebo name (document.onLoad) události, závisle na ní se pak element zobrazí
- endEvent - ukončení závislé na události

- repeat - počet opakování zobrazení, hodnota "indefinite" způsobí nekonečné opakování
- repeatDur - doba opakování (sec), hodnota "indefinite" způsobí nekonečné opakování

Pokud je hodnotou atributu čas v sekundách, je možné použít i desetinné číslo s desetinnou tečkou (0.5 = půl sekundy).

#### 4.6.1.2 Integrace HTML+TIME 2.0 do (X)HTML

Dokument (X)HTML bude nutné doplnit o několik informací, zejména o deklaraci typu dokumentu. Každý element ohraničující část kódu s HTML+TIME musí mít nastavenou třídu time. Výsledný dokument vypadá takto:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML+SMIL //EN" "html-
plussmil.dtd">
<html xmlns:t="urn:schemas-microsoft-com:time">
<?import namespace="t" implementation="#default#time2">
<style>
.time {behavior: url(#default#time2);}
</style>
<body>
HTML dokument
  <div class="time">
    HTML+TIME
  </div>
HTML dokument
</body>
</html>
```

## 4.6.2 Příklady HTML+TIME pro MSIE 5.5 a 6

Blikající text, ať už pomocí elementu blink nebo pomocí CSS, byl po čase z dobrého důvodu zavržen. Avšak pomocí HTML+TIME můžeme blikající text vytvořit nanovo (náhled):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML+SMIL //EN" "html-
plusmil.dtd">
<html xmlns:t="urn:schemas-microsoft-com:time">
<head>
  <?import namespace="t" implementation="#default#time2">
  <title>Blikající text</title>
  <style>
    .time, p {behavior: url(#default#time2)}
  </style>
</head>
<body>
  <div style="width: 200px;height: 40px;">
    <t:seq class="time" dur="0.2" repeatDur="1000">
      <p begin="0" dur="0.1"> Blikající text </p>
    <t:seq>
  </div>
</body>
</html>
```

Celý vtip spočívá v tom, že "t:seq", sekvence trvající 0.2 s a opakující se po dobu 1000 s, obsahuje odstavec textu, kterému je na časové ose vyhrazen úsek 0.1 s. Znamená to, že 0.1 sekundy je vidět text a dalších 0.1 sekundy vidět není.

HTML+TIME je možné aplikovat i na VML, pomocí jednotlivých sekvencí nebo prostřednictvím atributů:

```
<v:shape class="time" dur="5s" begin="4s" style='width: 60;
height: 60' stroke="true" strokecolor="black" stro-
keweight="1" fill="true" fillcolor="white" coordorigin="0
0" coordsize="60 60">
<v:path v="m 30,0l 60,15,60,45,30,60,0,45,0,15x e" />
</v:shape>
```

## 4.6.3 XHTML+SMIL Profile

S MSIE 6 přišel nový profil pro integraci SMILu do prohlížečů, nazvaný XHTML+SMIL Profile, který integruje podmnožinu specifikace SMIL 2.0. Profil zahrnuje podporu animací, kontroly obsahu, mediálních objektů, časování, synchronizací a přechodů. Některé moduly SMIL 2.0 chybí, ale většinou jsou nahrazeny CSS. Dlužno říci, že



MSIE6 sice XHTML+SMIL podporuje, avšak ne zcela korektně, mnohem lépe si vede s HTML+TIME.

Základní hlavičky XHTML+SMIL dokumentu vypadají následovně:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+SMIL //EN"
"http://www.w3.org/2001/SMIL20/WD/xhtmllplusmil.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:smil="http://www.w3.org/2001/SMIL20">
```

## Budoucnost SMILu

Svět multimediálních prezentací vypadá zbytečně zamotaně, což je způsobeno existencí mnoha profilů a technologií. Nejlepší bude shrnout vývoj popisovaných formátů v několika bodech:

- **1998** - Specifikace SMIL 1.0
- **1998** - Specifikace HTML+TIME - implementuje prvky SMIL 1.0 do některých prohlížečů (především MSIE5), některé elementy a atributy jsou nahrazeny CSS
- **2000** - HTML+SMIL language profile - profil vycházející z pracovního návrhu SMIL 2.0
- **2000** - HTML+TIME 2.0 - implementace většiny modulů SMIL 2.0
- **2001** - SMIL 2.0
- **2001** - Specifikace XHTML+SMIL Profile (Workig Draft) - později částečně podporována v MSIE6

Jazyk SMIL může najít uplatnění zejména ve firemních sítích, na samotném internetu se zatím moc často nepoužívá. Nutno podotknout, že multimediální zprávy MMS rovněž vycházejí z jazyka SMIL, takže je možné, že se jeho doménou stanou mobilní zařízení. SMIL přináší univerzální, výkonné a přitom nenáročné řešení, které je pro mobilní platformu mnohem důležitější než pro oblast PC.

## 4.7 SMIL - animace a prezentační efekty

SMIL, neboli Synchronized Multimedia Integration Language, umožňuje vytvářet působivé multimediální prezentace obsahující video, hudbu, obrázky i text. V tomto článku se seznámíme s metodami animace objektů ve SMILu a se základními prezentačními efekty, které nám v budoucnu poslouží při tvorbě komplexních prezentací.

Prezentace slouží k předvedení informací a k těmto účelům jsou mnohdy použity prezentační efekty. Kdykoli text zdánlivě spadne, vletí a odsune se, vždy za tímto dějem stojí prezentační efekty. Stejně tak je možné vytvářet prezentace v jazyce SMIL a obdobně lze použít prezentační efekty a animace. V předchozích kapitolách jsem několikrát teoreticky popisoval tvorbu a možnosti animací.

Animační modul SMILu, který je mimochodem integrován i do SVG, je souborem několika druhů animací a dalších efektů. Co je to ve skutečnosti animace? Na webu se mnohdy setkáváme s animovaným GIFem, animace je výraz zahrnující oživení a pohyb. A tak animace může blikat, animace se může odsouvat z dokumentu nebo může poskakovat přes celé okno. Možnosti animací ve SMILu lze obsáhnout v několika bodech:

Přesun objektu po křivce (křivka je vymezena dvěma body a časovým úsekem mezi nimi).

Změna atributu - změna barvy, velikosti, polohy a podobně. Změnou atributu lze dosáhnout například plynulé změny barvy z černé na bílou (přes odstíny šedi) nebo postupného zvětšování objektu.

Rollover - záměna obrázku po najetí myši, často využíváno na webových stránkách.

Střídání několika obrázku (animovaný GIF), ale i jiných prvků.

Animace jsou podporovány až ve SMIL 2.0 a jejich problematika je natolik rozsáhlá, že jsem raději zvolil animace v HTML+TIME, které jsou výrazně jednodušší a momentálně lépe využitelné. Všechny zdrojové kódy jsou proto psány v HTML+TIME a lze je bez problémů zobrazit v Internet Exploreru 5.5 a 6.0.

## **4.8 Využití a zhodnocení**

Všechny tři výše uvedené metody jsou použitelné, ale každá z nich má jisté výhody a nevýhody a volba nejvhodnější metody záleží na volbě hodnotících kritérií.

Využití JavaScriptu se nabízí jako jedno z možných a celkem jednoduchých řešení. Při této metodě není nutné instalovat nějaké zvláštní programové vybavení, JavaScript je prohlížeči podporován a jeho schopnosti jsou pro jednoduché prezentace postačující. Při použití JavaScriptu však narazíme na několik problémů. Každý prohlížeč může interpretovat JavaScript vlastním způsobem (největší rozdíly jsou asi mezi Internet Explorerem a ostatními prohlížeči), dále je pro možnost synchronizace streamovaného videa s ostatními položkami pomocí JavaScriptu nutný přehrávač videa, který je možné pomocí JavaScriptu ovládat.

Budeme-li hodnotit metody z hlediska použitelnosti v různých prohlížečích a operačních systémech, bude nejlepším řešením Macromedia Flash Media Server, protože Fla-

sh přehrávače jsou dostupné do téměř všech typů prohlížečů. Další výhodou je široké pole různých možností, které lze vytvořit (např. vytvořit interaktivní animaci a podle interakce spouštět video od určité pozice). Nevýhodou této metody je, že vyžaduje dobrou znalost prostředí Macromedia Flash, ve kterém se výukové materiály vytvářejí. Dále je nutné mít nainstalovaný Flash Media Server. Protože je v současné době na CESNETu provozován streamovací server, nejeví se pořízení MFS jako vhodné, a proto tuto metodu musíme z našeho pohledu zavrhnout. Pokud si ale organizace musí pořídit vlastní streamovací server a má programátory, kteří ovládají tvorbu Flash animací, je FMS pro tento případ asi nejlepší volbou.

Z pohledu jednoduchosti tvorby a editace výukových materiálů je z uvedených metod jednoznačně nejlepší jazyk SMIL. Díky jeho struktuře založené na XML je velice jednoduché upravovat jednotlivé prezentace a nabízené možnosti jsou obvykle postačující. Nevýhodou je zatím menší rozšiřitelnost oproti Macromedia Flash a případné zobrazování prezentace do nového okna v externí aplikaci (např. Real Player). Pokud dojde k výraznějšímu rozšíření tohoto jazyka, je SMIL velmi vhodným kandidátem pro synchronizaci streamovaného videa s ostatními typy médií.

## **5 Popis dostupných aplikací pro práci se streamovanými daty**

V současné době se pro streaming používá několik formátů – RealSystem firmy RealNetworks, Windows Media firmy Microsoft, QuickTime firmy Apple a dále celá rodina MPEG formátů.

### **5.1 RealNetworks RealSystem**

Firma RealNetworks je průkopníkem streamingu na Internetu a její produkty jsou stále špičkou v této oblasti. V současné době je aktuální verze 9.0, výrazné změny však nastaly zejména s příchodem verze G2 (6.0) v roce 1999. K dispozici jsou obvyklé tři základní komponenty vysílacího řetězce – RealProducer pro encodování videa, RealServer pro jeho distribuci a RealPlayer pro přehrávání. Audio, video, sekvence obrázků, Flash a další formáty je možné kombinovat do výsledné prezentace pomocí jazyka SMIL (Synchronised Multimedia Language).

### **5.2 Windows Media**

Společnost Microsoft začala s vývojem vlastního řešení přibližně ve stejné době jako RealNetworks, po nástupu verze RealSystem G2 se však stalo technologicky zaostalejším. Verze 7.0 však již odstranila většinu nedostatků a produkty se po delší době opět staly zcela srovnatelné. Výhodou řešení Microsoftu je především jeho cena, protože všechny komponenty a nástroje jsou zdarma dostupné jako součást operačního systému kategorie Server. Distribuční řetězec tvoří aplikace Windows Media Encoder, Windows Media Server a Windows Media Player.

Nejmladším systémem je systém Microsoft Streaming Media (MS Netshow). Výhodou a zároveň nevýhodou Netshow je těsná integrace s Windows. Kódovací software je volně stažitelný z WWW serveru Microsoftu a je také součástí Windows 2000. Totéž platí pro vysílací server. Management vysílacího serveru je plně integrován do správy Windows (u ostatních systémů jde o WWW rozhraní).

Systém Windows Media je samozřejmě vhodný především pro vysílání obsahu (systém jednoho vysílače a mnoha příjemců).

Kodér je navázán na rozhraní DirectDraw/DirectShow, tudíž funguje s většinou dodávaných produktů. Výkon kódovacího počítače je obdobný jako v případě ostatních produktů. Stejně jako systém Real Networks podporuje tzv. bandwidth negotiation, kdy je možné posílat na server v jednom balíku více videostreamů v různé kvalitě, přičemž klient a ser-

ver dohodnou kvalitu přenášeného videa v závislosti na dostupné přenosové kapacitě. Netshow na rozdíl od systému Real Networks pouze ve videočásti. Audiočást bohužel tuto vlastnost neobsahuje, což znamená velký problém zvláště pro vysílatele rádií.

Protože je vysílací server integrován s Windows, není jeho nastavení problémem. Bohužel vlastníci ostatních operačních systémů přijdou zkrátka. S vysílacím serverem se pojí jeden problém. Ostatní platformy fungují tak, že kódér navazuje spojení se serverem (je to i logické, neboť uživatel obvykle sedí poblíže kódovacího pracoviště, zatímco je server zastrčený někde v hostingovém centru ISP). U Netshow ovšem navazuje spojení server, což s sebou přináší problémy v případě, že je kódér provozován například uvnitř organizace za firewallem. V takovém případě je uskutečnění přenosu takřka nemožné, neboť umožnění vysílání znamená bezpečnostní problém na straně vysílací organizace.

### **5.3 Apple QuickTime**

Společnost Apple zvolila jiné obchodní zaměření svého produktu a tak do nedávné doby nebyla k dispozici serverová část řetězce. I dnes se tento formát používá především pro distribuci předem připravených záznamů. V Evropě není příliš zastoupen díky menšímu rozšíření počítačů Macintosh než ve Spojených státech.

### **5.4 MPEG**

Formáty MPEG-1 a MPEG-2 se pro streaming vysoce kvalitního videa používají již relativně dlouhou dobu, díky vyšším datovým nárokům ale nejsou vhodné pro přenosy po Internetu. Uplatnění tak nalézají především při distribuci digitálního televizního vysílání DVB, jako náhrada klasických přenosových tras na vysílače, pro distribuci a kontribuci videopříspěvků v reálném čase a podobně. Pro přenos na Internet je určen zatím poslední člen rodiny – MPEG-4, který v současné době prodělává bouřlivý vývoj.

## Závěr

Tato práce je pouze zlomkem toho, jak se dají zpracovávat a přenášet multimediální data. Tato problematika je rozsáhlá hlavně díky své rozšířenosti a díky svým možnostem. Pokud půjde vývoj dopředu tímto tempem, můžeme se v budoucnu určitě těšit na zajímavé věci.

První kapitola pojednává o možnostech komprimace videa, kde jsem video kodeky rozdělil do dvou základních skupin. Bezeztrátové a ztrátové. Hlavním cílem byla ukázka velkého množství různých kodeků a jejich srovnání. Je to jen výběr těch nejzákladnějších z nich.

V druhé kapitole se zaměřuji již na hlavní problematiku a tou je streamování. Jedná se o možnost sledování videa přes počítačovou síť. Streamování můžeme rozdělit na dvě základní skupiny, které jsou podrobně popsány. Dnešní doba rychlého připojení na internet není neobvyklou věcí, ale jsou i takové domácnosti nebo místa, kde provider neposkytuje tak velké rychlosti připojení. Při streamování se bere ohled především na dostupnost daného streamu, proto jsem se nezaměřil jen na parametrické nastavení videa co se nejlepší kvality týče. Zmínil jsem se i o možnostech poskytování streamu v různých kvalitách právě pro širší spektrum uživatelů.

V další kapitole jsem se pokusil nastínit využití právě progresivního downloadu. Jeho využití je velice široké od vzdělávání po komerční využití. Streamování se dá využít i jako videokonference.

Ve čtvrté kapitole se zaměřuji na samostatnou práci se streamovanými daty na straně příjemce. Uvádím zde několik možnosti vytváření online prezentací. Zmiňuji se o jazyku SMIL, kdy příjemce může daný stream nejen spustit, ale může vytvořit online prezentace, které v sobě můžou zahrnovat několik streamů najednou, texty, obrázky a jiná data.

V poslední kapitole se věnuji různým aplikacím, pracujících se streamovanými daty. Tato kapitola rozšiřuje kapitolu o streamování.

Bylo velice těžké vybrat nejdůležitější informace z tak obrovského množství dostupné literatury. Především v kapitole o jazyku SMIL, jsem zmínil jen základy pro práci multimediálními daty.

Tato práce by měla sloužit jako úvod do vytváření online prezentací s podporou multimediálního streamu. Srovnává různé druhy kodeků a jejich využitelnost. Ukazuje možnosti streamování multimediálních dat a hlavně ukazuje na to, jak z tyto data zpracovat do onli-

ne prezentace. Je to úvod například do vytváření internetových portálu, kde si studenti budou moci prohlížet přednášky.

## **Resumé**

Hlavním cílem práce bylo ukázat směr vývoje dnešních multimedií a možnosti práce s nimi. Nejen co se týče zpracování na počítači, ale i jejich publikování přes počítačovou síť. Další věcí bylo vytvořit návod k vytváření online prezentací pro studenty, kteří by měli možnost sledovat výuku odkudkoli, kde mají přístup k internetu.

V práci se mi podařilo vytvořit srovnání různých druhů kodeků a jejich využitelnost. Zmínil jsem se o streamování a o možnostech vytváření online prezentací. Tyto informace by mely posloužit čtenářům v porozumění dané problematice a schopnosti vytvořit vlastní multimediální prezentaci.



## Použité internetové zdroje

1. [http://www.pvdmedia.com/streaming\\_ptatese.htm](http://www.pvdmedia.com/streaming_ptatese.htm)
2. <http://www.civ.cvut.cz/info/info.php?&did=443>
3. <http://www.civ.cvut.cz/info/info.php?&did=447>
4. <http://www.fi.muni.cz/~xpavlov/xml/examples/bc3/bc3.xhtml>
5. <http://www.cesnet.cz/doc/techzpravy/2005/streamvideo/>
6. <http://gimli.mysteria.cz/kompresse/algoritmy.html>
7. [http://en.wikipedia.org/wiki/Streaming\\_media](http://en.wikipedia.org/wiki/Streaming_media)
8. <http://interval.cz/clanky/smil-zakladni-elementy-a-konstrukce/>
9. <http://interval.cz/clanky/smil-jazyk-pro-multimedialni-prezentace/>
10. <http://www.mediacollege.com/video/streaming/overview.html>
11. <http://www.root.cz/zpravicky/twibright-hyperluma-lepsi-kodovani-vidoa/>
12. <http://ronja.twibright.com/hyperluma.php>
13. <http://interval.cz/clanky/smil-timed-interactive-multimedia-extensions-for-html/>
14. <http://interval.cz/clanky/smil-animace-a-prezentacni-efekty/>
15. <http://interval.cz/clanky/smil-tvorba-prezentace-v-praxi/>
16. [http://www.pvdmedia.com/streaming\\_technologie.htm](http://www.pvdmedia.com/streaming_technologie.htm)

## Použitá literatura

1. Long, Ben (Schenk, Sonja): 2005. *Velká kniha digitálního videa*. Portál: CP Books
2. Vlček, Karel: 2004. *Kompresse a kódová zabezpečení v multimediálních komunikacích*. Portál: Praha
3. Myslín, Josef: 2005. *Digitální video v praxi*. Portál: Computer Media
4. Dunn, Jason R.: 2003. *Digitální video*. Portál: Computer Press
5. Stolarz, Damien: 2004. *Mastering Internet Video: A Guide to Streaming and On-Demand Video*. Portál: Addison-Wesley Professional
6. Lee, Jack: 2005. *Scalable Continuous Media Streaming Systems: Architecture, Design, Analysis and Implementation*. Portál: Wiley
7. Wootton, Cliff: 2005. *A Practical Guide to Video and Audio Compression: From Sprockets and Rasters to Macro Blocks*. Portál: Focal Press

# Evropský polytechnický institut, s.r.o.

## *Záznam o bakalářské práci*

<b>Autor</b>	Jan Válek, DiS.
<b>Název DP</b>	Digitální zpracování a přenos videa
<b>Studijní obor</b>	Elektronické počítače
<b>Rok obhajoby</b>	2007
<b>Počet stránek</b>	45
<b>Počet příloh</b>	0
<b>Vedoucí BP</b>	RNDr. David Skoupil
<b>Oponent BP</b>	Ing. Vladimír Grohmann
<b>Anotace</b>	Shrnutí problematiky digitálního videa. Možnosti a návody pro vytvoření mediálního streamu a online multimedialních prezentací a jejich aplikace do prostředí internetu.
<b>Klíčová slova</b>	Video, stream, kodeky, komprimace, SMIL
<b>Místo uložení</b>	Knihovna EPI, s.r.o.
<b>Signatura</b>	