

network streetart prezentace projektu, ver. 1.2 / 29.1. 07

Prokop Bartoníček (VŠUP), David Havlík (ČVUT), Andrej Hazucha (ČVUT), Karel Šimek (ČVUT)

network streetart

prezentace projektu, ver. 1.2 / 29.1. 07

autoři projektu:

Prokop Bartoníček (VŠUP), (www.pebe.cz): kreativa, design, produkce

David Havlík (ČVUT), (rewko@centrum.cz): programování serverové části

Andrej Hazucha (ČVUT), (david.havlik@zs1.nmnm.cz): progr. grafické části

Karel Šimek (ČVUT), (karel.fel@centrum.cz): progr. komunikační části

úvod: Prokop Bartoníček: koncept

Řešit smysl tvorby ve veřejném prostoru by bylo asi zbytečné, v tom směru bylo myslím řečeno již mnoho. Snahou projektu je propojit veřejný fyzický prostor s veřejným prostorem virtuálním. Spojit ulice měst společným médiem, které umožní kolemjdoucí na moment spojit v jednom tvurčím procesu.

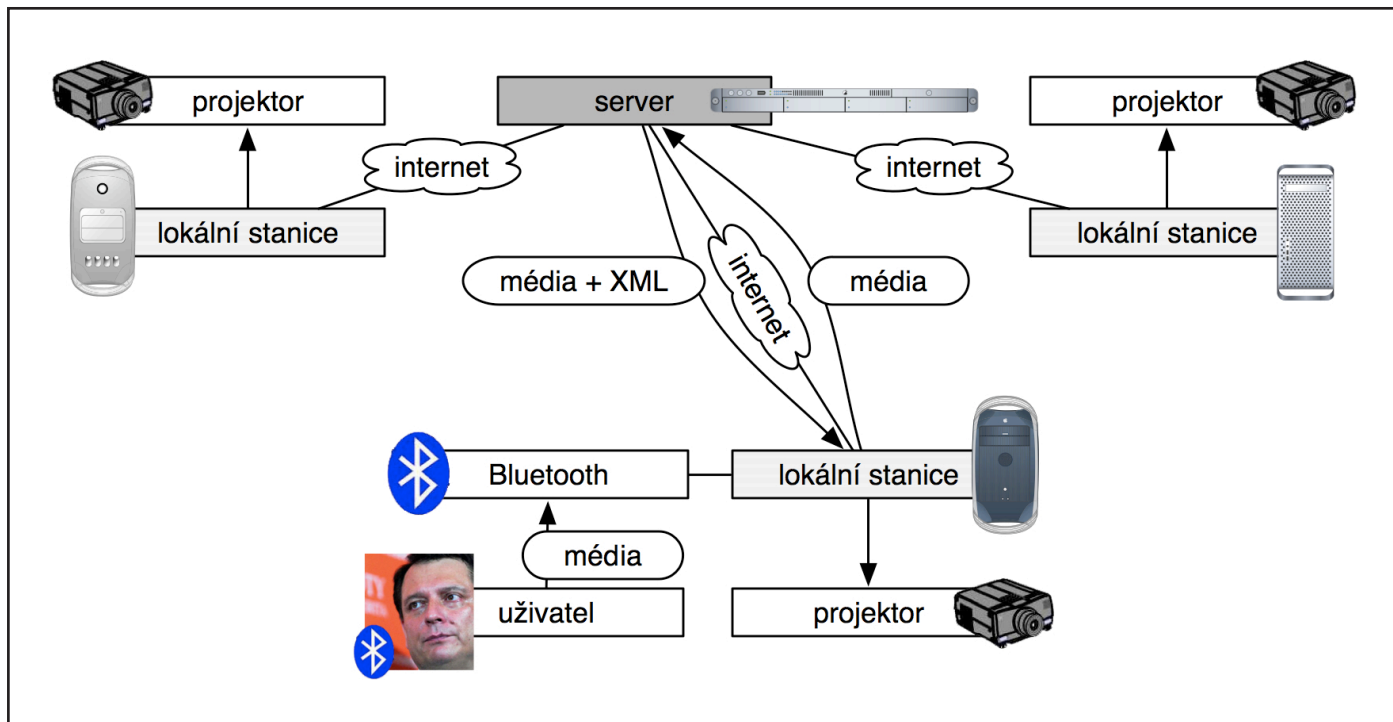
Projekt Network Streetart bude interaktivní instalace ve veřejném prostoru, citující fenomén streetartu a grafity. Půjde o dvě či více projekcí přímo na stěny domů měst, spojené přes internet. Účastníci v blízkosti jedné ze zdí odešlou grafiku, fotografii, či video smyčku ze svého mobilního zařízení pomocí technologie Bluetooth na skrytou stanici. Ta automaticky soubor přijme a odešle materiály na vzdálený server, který bude centrem celého projektu. Server zajistí grafické procesy, které materiály upraví pro projekci a odešle všem stanicím zapojených do projektu. Ty zasadí materiály do kompozic, všude po „světě“, prolnou také doprovodné informace o čase zrodu příspěvku se jménem zdroje a lokací domovské stanice. Server pak bude shromažďovat údaje o vývoji historie promítaného obrazu, které bude možno prohlížet na webové stránce projektu.

Snahou je vytvořit systém, do kterého by se mohl zapojit kdokoli. Postačí stáhnout z webu projektu program pro stanici, tedy nainstalovat jednoduše ovladatelný software, zapnout Bluetooth, a připojit projektor nasměrovaný na stěnu protějšího domu.

technické řešení studentů ČVUT:

V této části se pokusíme přiblížit řešení jednotlivých částí systému.

schéma systému Network Streetart, P. Bartoníček:



provizorní vizualizace koncové projekce, A. Hazucha:



1] David Havlík: serverová část

uživatelské role:

uživatel: Aktivní divák, osoba, která odesílá datová média na lokální stanici prostřednictvím Bluetooth rozhraní svého telefonu.

provozovatel: Osoba, která má na starosti provoz, správu jedné lokální stanice, např. galerie, klub, knihkupectví, módní butik, tvůrčí komunita

administrátor: Osoba, která má na starosti provoz, správu serveru, v tomto případě autoři projektu.

slovník pojmů:

server: Počítačová stanice, se kterou jsou propojeny všechny lokální stanice, slouží jako archiv přijatých médií, řídí a synchronizuje činnost celého systému.

lokální stanice: Počítačová stanice, která zprostředkovává příjem médií od uživatelů a serveru, odesílá média na server a zobrazuje média na projekční plochu.

projekční plocha: Zeď domu v městské ulici, panel v industriální výstavbě, na které se zobrazují média pomocí projektoru.

bluetooth: Bezdrátové rozhraní pro odesílání a přijímání dat .

datové médium: Datové soubory grafiky typu bmp, jpg, gif atd. a videí typu mpeg-4 a 3G atd.

fresh média: Soubory typu média, která ještě nebyla zobrazena na “zdi”.

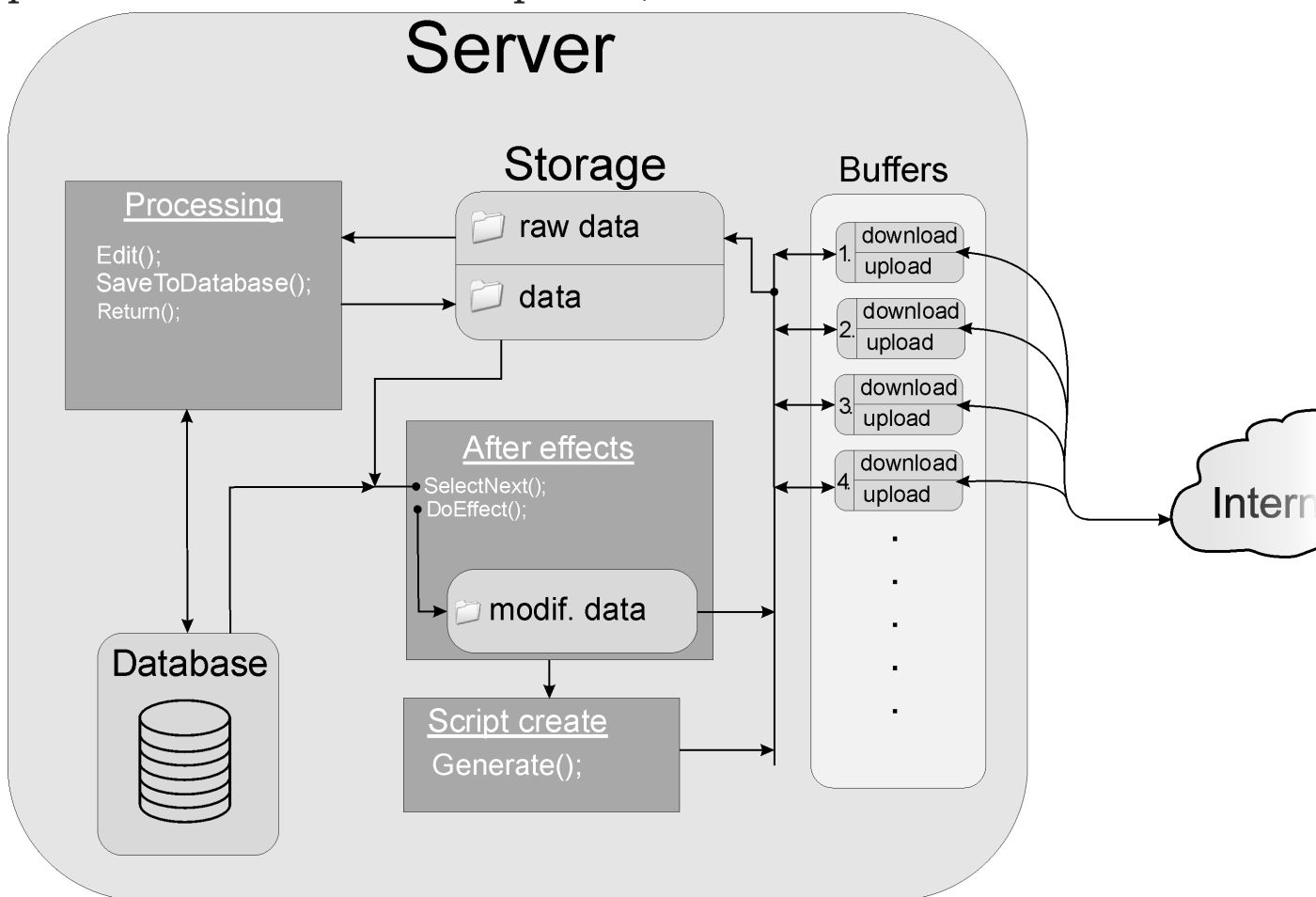
zobrazovací skript: XML skript, který určí, kde a jak se má dané médium zobrazit na “zdi”.

časové kvantum: Zatím blíže nespecifikovaná časová perioda, za kterou se odesílá na lokální stanice řídicí XML skript.

stručný popis funkce:

Uživatel odešle pomocí Bluetooth rozhraní mobilního telefonu/PDA/notebooku datové médium na lokální stanici. Stanice médium přešle na server. Server každé přijaté médium primárně upraví (komprimace apod.), uloží na datové úložiště a údaje o médiu zaznamená do databáze. Dále pak server každému takovému prvotně zpracovanému datovému médiu přidá after efekty (maska pro klíčování, orámování, ořez apod.). Takto zpracovaná média se zpětně rozesílají do všech lokálních stanic. Po nějakém časovém kvantu (periodě cca 3-5 min.) se vytvoří řídicí XML skript, ve které budou obsaženy údaje o tom kam a jak se mají jednotlivá média na promítacím plátně zobrazit. Skript je rozeslán na všechny lokální stanice. Po doručení vytvoří všechny lokální stanice na základu skriptu a doručených médií nový obraz který promítnou na zobrazovací plochu. Tento skript slouží mimo jiné i jako synchronizační prvek pro všechny lokální stanice v systému začleněné. Mezi vytvořením a opětovným reloadováním obrazu na projekční ploše se videa cyklicky opakují a obrázky jsou promítány staticky.

přibližné schéma serverové aplikace, D. Havlík:



popis jednotlivých částí:

buffry: jsou datová úložiště, kam se ukládají média, která se momentálně přijímají nebo odesílají, po dokončení přenosu směrem „upload“ se data z buffru odstraní, pro směr „download“ se přemístí do úložiště „raw data“, kam se ukládají média, která nebyla primárně zpracovaná a na toto zpracování teprve čekají

storage: datové úložiště, obsahující dvě podsložky, jednu pro data bez primárního zpracování „raw data“, druhá pro data již primárně zpracovaná „data“

database: MySQL databáze pro archivaci údajů, jednak o přijatých médiích, ale i dat pro správu celého systému.

processing: Řídí průběh primárního zpracování médií, archivaci dat do databáze a přesun mezi podsložkami „raw data“ a „data“.

after effects: Aplikuje na jednotlivá média after efekty podle parametrů jednotlivých médií. Takto zpracovaná média jednak ukládá do datového úložiště „modified data“ a také je rozesílá do všech stanic. Po uplynutí časového kvanta volá část „Skript create“.

script create: Uchovává informace o médiích odeslaných v předchozím časovém kvantu. Na základě této informace a informace o počtu nových modifikovaných médií vytvoří nový řídicí skript. V případě, že počet nových médií je nízký a nezaplnil by celou plochu projekčního plátna, vybere média aktuálně promítaná, která zůstanou pro konstrukci nového obrazu. Výběr provede na základě stáří jednotlivých médií (stářím je myšlena doba jakou je médium archivováno na serveru). Po konstrukci skriptu jej rozešle jednotlivým stanicím.

ukázka obnovy médií v praxi:

Máme časové kvantum 5 min.

čas: 12:05-12:10

Projektory na všech lokálních stanicích zobrazují obrázky a videa dle níže uvedeného schématu, kde číslo za typem média označuje identifikační číslo média v serverové databázi a ve druhém řádku čas, kdy bylo médium na server přijato.

schéma zobrazovaného plátna – čas 12:05-12:10

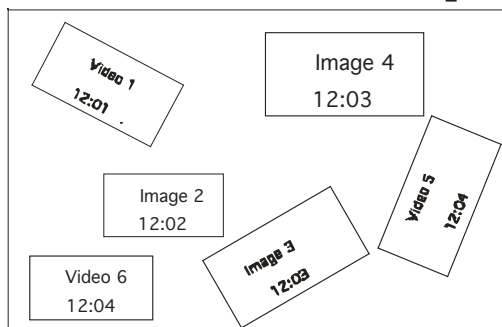
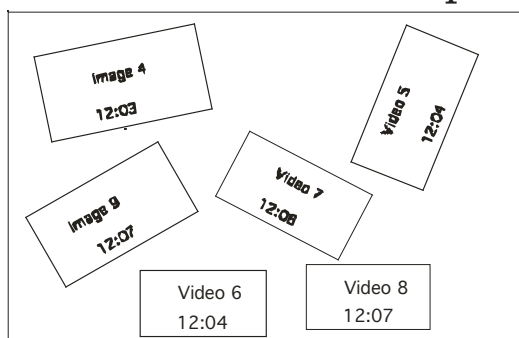


schéma zobrazovaného plátna – čas 12:10-12:15



Za tento čas od 12:05 do 12:10 byly na server doručeny pouze tři nová média. To je na konstrukci zcela nového obrazu málo, jelikož by byl velmi řídko pokryt. Proto se tři nová média doplní do plného pokrytí projekční plochy médii, která byla již částí obrazu vysílaného v čase 12:05-12:10.

přibližná ukázka řídicího skriptu:

Ukázka se vztahuje k situaci vyobrazené na schématu z 12:05-12:10.

```
<script>
  <time=12:05 length=300/>
    <media>
      <medium>
        <id=1/>
        <type="avi"/>
        <length=20/>
        <location x=20 y=1256/>
        <rotation=270>
      <medium/>
      <medium>
        <id=2/>
        <type="jpg"/>
        <length=0/>
        <location x=200 y=300/>
        <rotation=0>
      <medium/>
      <medium>
        <id=3/>
        <type="jpg"/>
        <length=0/>
        <location x=700 y=50/>
        <rotation=30>
      <medium/>
      <medium>
        <id=4/>
        <type="bmp"/>
        <length=0/>
        <location x=850 y=1256/>
        <rotation=0>
      <medium/>
      <medium>
        <id=5/>
        <type="avi"/>
        <length=50/>
        <location x=1200 y=300/>
        <rotation=800>
      <medium/>
      <medium>
        <id=6/>
        <type="avi"/>
        <length=120/>
        <location x=50 y=20/>
        <rotation=0>
      <medium/>
    <media/>
</script>
```

2] Andrej Hazucha: grafická část

after effects:

účel: Upravit jas a kontrast média a pokusit se docílit toho, aby dané médium vypadalo dobře na “zdi”.

vstupní data: Fresh média, která byla zaslaná na server a byla zkontrolovaná, zda jsou vhodná k zobrazení z formálního hlediska, nikoli obsahového. Dále pak základní informace o daném médiu získané při přenosu.

zpracování: Vyvážení jasu a kontrastu bude nutné, jelikož se bude promítat na zeď. Aplikace se pokusí rozeznat vhodnou část média, kterou lze považovat za pozadí, tedy odlišit “papír” od “kresby” a nahradit ho alfa kanálem pro klíčování na stěnu. V této fázi se také do obrazu prolne animovaný popisek s informací o médiu: místo přijetí, čas přijetí, název mateřského zařízení. Pro manipulaci s materiály/médií se pravděpodobně využije knihovna Allegro nebo SDL. Medium je po úpravách uloženo do souboru jako kopie, která je pak rozesílána na lokální stanice.

výstup: Nové soubory obsahující upravené médium - zdokonalené pro zobrazení na zdi.

script create:

účel: Vytvoření XML souboru popisujícího zobrazovanou projekci. Tento XML soubor je v podstatě datovou reprezentací projekce. Je posílán jednotlivým lokálním stanicím, které dle XML sestaví výstupní obraz.

vstupní data: Databáze médií a XML výstupní soubor z předešlého běhu aplikace, tak aby algoritmus věděl, která média jsou zobrazována a kde.

zpracování: Aplikace je zodpovědná za výběr nejnovějších souborů z databáze. Pak musí najít rozumné rozložení médií na plátně. Poloha médií, které na plátně byla i předtím, by se moc měnit neměla. Pak aplikace vygeneruje XML soubor a server ho může rozesílat lokálním stanicím.

výstup: XML soubor, obsahující seznam médií (relativní cestu - médium by již mělo na dané lokální stanici existovat), které se mají zobrazit a jejich transformace (poloha, rotace, atd.).

3] Karel Šimek: komunikační část

komunikace, sdílená paměť a rizika:

Jelikož je projekt na komunikaci založen, nelze tuto jeho část podcenit. Z analýzy vyplývá, že 80% problémů bude způsobeno právě sítí. První problém se týkal použitých technologií. Dnešní internet je založen na tzv. TCP/IP modelu a ten se díky svoji rozmanitosti a robustnosti prosadil nejvíce. Je odvozen z ještě podrobnějšího OSI modelu. Pro transportní vrstvy existují dvě možná řešení. TCP a UDP. TCP je connection-oriented a stará se jak o znovuposlání dat při jejich nedoručení, tak o kontrolu rychlosti a integrity (přirovnání k rozhovoru po telefonu). UDP nechává tyto problémy vyšším vrstvám TCP/IP modelu. V našem případě pak aplikaci samotné. Struktura UDP framu (rámce) je tak podstatně jednodušší a nemusí zahrnovat tolik kontrolních mechanismů. Na druhou stranu je pak o něco náročnější vytvoření programu samotného, kdy je potřeba tyto mechanismy zahrnout do programu. Ať již použitím knihovny nebo vlastním návrhem.

V obou případech lze dosáhnout rozumné spolehlivosti a případné chyby okamžitě řešit opakováním požadavku popř. odesláním chyby do logu či na email správce.

Během návrhu se objevila potřeba implementovat vícevláknový transportní systém s vlastním zásobníkem. Jeho výhodou je exklusivita obsluhy každé připojené lokální stanice, která tak nemusí čekat na ostatní komunikující lokální stanice. Odtud nám tedy vychází dva modely pro komunikaci. První serverový, s podporou více vláken, a druhý klientský, který je vždy připojen k maximálně jednomu serveru. Síť je na komunikaci více kanály připravena.

Dále bylo potřeba vyřešit hazardy, které vznikají při používání sdílené paměti. Lokální stanice odesílají na server média a zapisují jejich záznamy do databáze. Zároveň server čte a odesílá soubory s médii zpět lokálním stanicím společně s kontrolním skriptem obsahujícím zobrazovací instrukce. V tom případě však nemusí být splněny podmínky pro konzistenci databáze. To by mohlo vést k neplatným operacím, duplicitám, chybám a dokonce i ke ztrátě dat.

Řešení existuje několik. V systémech linux i windows jsou implementovány funkce, které umožňují provedení atomické instrukce a s jejich pomocí lze dosáhnout toho, že v jeden okamžik pouze jeden proces přistupuje k databázi. Řeší se to za pomoci semaforů a zámků (MUTEXy), které jsou implementovány do operačního systému. Nyní lze databázi používat více lokálními stanicemi, kdy každá nejprve provede:

Lock () ; zamkne, pokud nelze zamknout tak čeká,

Work () ; provede operace s DB,

Unlock () ; co nejdříve odemkne a umožní tak přístup ostatním procesům.

analýza rizik:

latence: Pro běžné použití postačí v řádu desítek ms, ovšem při provozu na bezdrátových sítích latence rychle stoupá na stovky ms až sekundy (při vysoké zátěži)!. V našem projektu jsou klienti nuceni vyčkat dokud k nim nedorazí jak instrukce, tak média. Tedy výsledkem je, že zobrazovaná data v různých lokalitách s různou kvalitou připojení, mohou být zobrazena s prodlevou. Pokud není jeden z klientů schopen přijmout data dostatečně rychle, a je-li k dispozici další balík dat a skript, pak se skript uloží na zásobník a čeká. Velikost zásobníku je omezena a v případě naplnění jsou nejstarší neodeslané položky mazány na úkor nových a je zaslána zpráva, která na toto upozorňuje.

chyby sítě: Skript je ve formátu XML obsahuje i kontrolní součet CRC pro ověření jeho integrity. Lze tedy jeho rekalkulací ověřit integritu přenesených dat. Ostatní případy (late collisions, broadcast-storms) jsou již vyřešeny na nižších úrovních.

hardware: TCP i UDP lze s úspěchem provozovat na všech ethernetových médiích (UTP, OPTIKA, SERIAL, MODEM). Pro náš projekt postačí běžná osmižilová nestíněná kroucená dvojlinka (UPT kabel třídy 5e kompatibilní i s 1Gbps sítěmi).

závěr:

V druhém semestru náš team čeká mnoho práce.