

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Vizualizace EEG/ERP signálu v EEG/ERP Portálu

Plzeň, 2012

Lucie Koloušková

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 21. června 2012

Lucie Koloušková

Abstract

This thesis deals with the visualization of EEG/ERP signal gained from neuroinformatic experiments. Data and metadata gained from these experiments are stored in the database and the EEG/ERP Portal is its user interface. The visualization is essential, because the data are stored in special binary non-transparent format. Therefore the visualization is a suitable solution, which will provide the online preview of the recorded brain activity.

The next part of this thesis describes the HTML5 technology that is used to create the visualization and the reasons why the HTML5 was chosen as well. There is also HTML5 comparison with other available technologies. Finally, future extensions of the EEG/ERP visualization are proposed.

Abstrakt

Tato práce pojednává o vizualizačním nástroji pro zobrazení EEG/ERP signálu, který se získává v rámci natáčení neuroinformatických experimentů. Data a metadata získaná z těchto experimentů jsou ukládána v databázi, nad níž běží EEG/ERP Portál. Samotná data, tedy průběh vln zaznamenaný při experimentu, jsou uložena v běžně nečitelné binární podobě. Vizualizace tohoto signálu je vhodným řešením, protože poskytuje online náhled zaznamenané mozkové aktivity.

Část práce je také zaměřena na popis a výběr technologie HTML5, která je použita pro vytvoření vizualizace. Výběr dokládám srovnáním s jinými dnes dostupnými technologiemi. Závěrem uvádím některé další možnosti rozšíření současné funkcionality.

Obsah

1	Úvod	2
2	EEG/ERP experiment	3
2.1	Metoda evokovaných potenciálů	3
2.2	Získaná data a metadata	4
3	EEG/ERP Portál	7
3.1	Koncept	7
3.2	Technologie	8
4	Specifikace problému	10
4.1	Požadavky	10
4.2	Přehled technologií	11
4.2.1	Adobe Flash	11
4.2.2	SVG	13
4.2.3	HTML5 a Canvas API	15
4.3	Porovnání technologií	16
5	HTML5 a Canvas API	20
5.1	Vize a principy HTML5	20
5.1.1	Sémantické elementy	20
5.1.2	Odstranění zásuvných modulů	21
5.2	Specifikace	22
5.3	Podpora webových prohlížečů	23
5.4	Canvas API	24
5.4.1	Podpora Canvas	24
5.4.2	Práce s rozhraním	25
6	Vizualizace EEG/ERP signálu	28
6.1	Knihovna pro načítání signálu	28
6.2	Kontrola podpory prohlížeče	31
6.3	Proces vizualizace	32
6.4	Možnosti rozšíření	34
7	Závěr	35
A	Porovnání kódu SVG animace pomocí JavaScriptu a SMIL s Canvas API	41
B	Ukázka hlavičkového souboru VHDR	44
C	Ukázka souboru s markery VMRK	46

1 Úvod

V této práci jsem se snažila najít optimální řešení pro vizualizaci EEG/ERP signálu, který se získává z neuroinformatických experimentů prováděných v laboratoři spadající pod Katedru výpočetní techniky a informatiky (dále jen KIV) Fakulty aplikovaných věd Západočeské univerzity v Plzni. Výsledná data jsou uložena v binární podobě v souboru s příponou `eeg` a pro přehrání signálu je potřeba desktopová aplikace.¹ Pro rychlý náhled je takový přístup nepraktický a zdlouhavý, protože uživatel musí věnovat čas stažení binárního souboru se záznamem experimentu, spuštění aplikace na svém počítači a otevření souboru, přičemž pomímám založení projektu pro daný experiment a podobné podružné kroky. Vzhledem k dnešním dostupným technologiím se nabízí očividná možnost, a to poskytnout uživateli prostředek pro online přehrávání zaznamenaného signálu přímo v rámci EEG/ERP Portálu, tedy ve webovém prohlížeči.

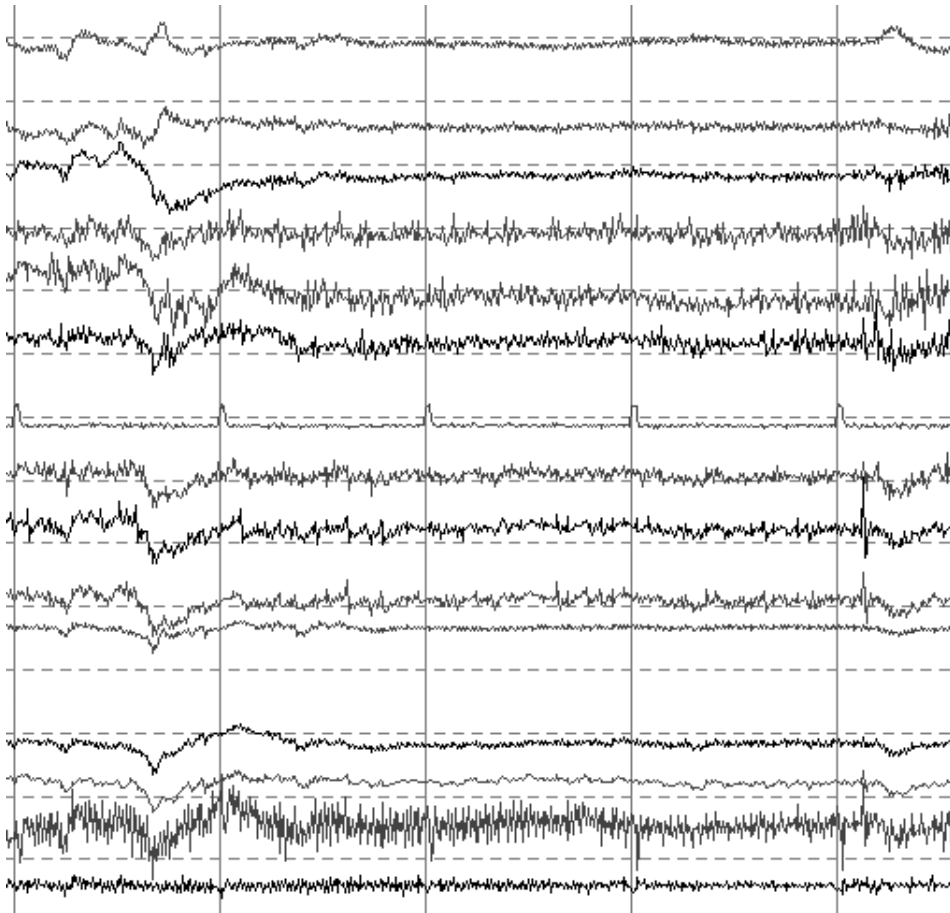
Současné trendy na poli vývoje webu vedou k co největší uživatelské přívětivosti, celkové přehlednosti a logickému uspořádání a především k interaktivitě, pokud je v daném případě žádoucí. EEG/ERP Portál tímto případem je, protože funguje jako úložiště pro data, jejichž sdílení by mělo být v rámci vědeckých skupin co nejjednodušší. Navíc pojem interaktivita nutně neznamená spoustu appletů či robustní aplikaci ve Flashi. Jedním z nadějných nástupců je HTML5 a právě na jeho komponentě Canvas je ve spojení s JavaScriptem a frameworkem jQuery postavena celá vizualizace.

V následujících kapitolách je popsán EEG/ERP Portál a jeho využití v rámci neuroinformatického výzkumu na světové úrovni. Další blok je věnován čistě popisu HTML5, jeho přednostem, ale i nedostatkům. Zde se také dostanu k podrobným informacím o komponentě Canvas, jak z hlediska specifikace dle W3C, tak z pohledu praktického použití. Poslední do pořadí nikoliv však do významu je sekce věnovaná samotné vizualizaci EEG/ERP signálu, kde je naznačena struktura dat, proces načtení a konečně jejich vizualizace.

¹Konkrétně v našem případě se jedná o Brain Vision Analyzer.

2 EEG/ERP experiment

Na KIV Fakulty aplikovaných věd Západočeské univerzity v Plzni funguje výzkumný tým neuroinformatiků. Data potřebná pro jejich výzkum se získávají v rámci EEG/ERP experimentů, při nichž je subjekt měřen podle konkrétního scénáře. Experimenty jsou založeny na elektroencefalografii (EEG) a na metodě evokovaných potenciálů (ERP²).



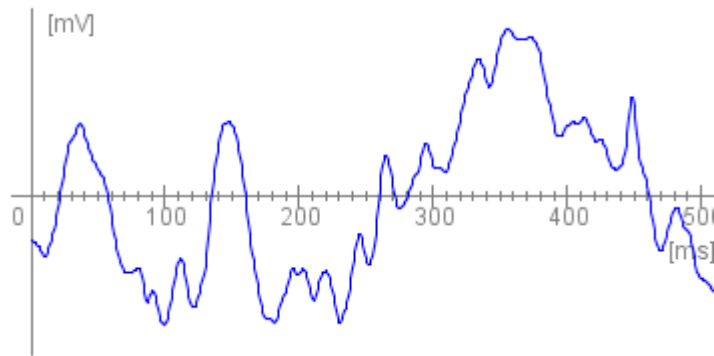
Obrázek 1: Ilustrativní ukázka EEG signálu.

2.1 Metoda evokovaných potenciálů

Měření EEG má vedle nízkých nákladů a neinvazivnosti také své nevýhody. Jednou z nich je fakt, že výsledkem měření je sumace řádově stovek různých neurálních

²Zkratka ERP bývá v literatuře interpretována dvěma způsoby - Event-Related Potential nebo Evoked Response Potential.

signálů. Výsledný encefalogram obsahuje také artefakty³ způsobené např. mrknutím, svalovým záškubem nebo polknutím, které u subjektu nelze uspokojivě eliminovat. Druhý problém je ten, že EEG snímá mozkovou aktivitu elektrodami přiloženými na skalpu a průchod signálu lebkou snižuje jeho amplitudu řádově na mikrovolty.



Obrázek 2: Ukázka ERP vlny.

Jedním ze způsobů jak určit konkrétní nervovou odezvu vyvolanou smyslovým nebo motorickým stimulem je metoda evokovaných potenciálů. Reakce subjektu na vnější podnět se projeví změnou elektrické aktivity nervového systému. Vzhledem k měřenému EEG je amplituda evokovaného potenciálu výrazně menší, proto je pro spolehlivější detekci odezvy stimul v průběhu měření opakován, aby byl nepříznivý poměr signál:šum eliminován dostatečným počtem vzorků. Mnohonásobné zaregistrování odpovědi na podnět zajistí dostatečný počet epoch, tedy oblastí kolem předpokládaného výskytu potenciálu, pro zprůměrování. Do průměru se nepočítají epochy s artefakty, protože jejich amplituda může být až stokrát větší oproti potenciálu a přítomnost artefaktů ve vzorku by tak zkreslila průběh vlny evokovaného potenciálu nebo zcela znemožnila jeho detekci.

2.2 Získaná data a metadata

Data a metadata jsou během experimentu zaznamenávána softwarem Vision Recorder. Výstup tvoří 3 soubory, v nichž jsou od sebe odděleny obecné informace o měření (např. počet kanálů), záznam naměřené mozkové aktivity a výčet markerů.

- **EEG** - záznam mozkové aktivity na jednotlivých kanálech

³Artefakt je signál, který nepochází z mozku. Dle původů rozlišujeme oční, EKG a EMG artefakty.)

- **VHDR** - hlavičkový soubor s obecnými informacemi jako je počet kanálů, interval vzorkování nebo formát dat
- **VMRK** - seznam markerů spolu se záznamem jeho typu, umístění a relevantnosti vzhledem ke kanálu

Každý experiment je popsán hlavičkou. Jedná se o ASCII soubor s příponou `.vhdr`, jehož formát má strukturu jako INI soubor operačního systému Windows. Sestává ze sekcí, které člení klíčové atributy s přiřazenou hodnotou podle jejich významu; příkladem sekce je `[Common Infos]` pro obecné informace o počtu kanálů nebo formátu zaznamenaných dat. Pro klíčové atributy platí pravidlo, že tento klíč musí být jedinečný v rámci jedné sekce. Kompletní soupis předdefinovaných sekcí a klíčů s výčtem hodnot je k dispozici v manuálu [15, strana 182]. Ukázka VHDR souboru je v Příloze A.

Soubor s příponou `.vmrk` je strukturován podobně jako hlavičkový soubor VHDR. Tento soubor obsahuje výčet markerů, přičemž u každého z nich jsou zaznamenány následující informace:

- typ;
- popis;
- pozice;
- délka trvání;
- číslo kanálu, k němuž je výskyt markeru vztažen;
- datum.

Pokud je číslo kanálu 0, znamená to, že marker je vztažen ke všem měřeným kanálům. Kompletní soupis předdefinovaných sekcí a klíčů s výčtem hodnot je k dispozici v manuálu [15, strana 186]. Ukázka VMRK souboru je v Příloze B.

Formát dat obsažených v souboru s příponou `.eeg` je definován v hlavičkovém souboru. Hodnoty jsou uloženy v binární podobě, přičemž formát definovaný atributem `BinaryFormat` může být následující:

- **IEEE_FLOAT_32** - desetinné číslo s plovoucí čárkou, jednoduchá přesnost, 4 byty na hodnotu;
- **INT_16** - 16-bitový signed integer;
- **UINT_16** - 16-bitový unsigned integer.

Pokud není definováno jinak, výchozí hodnota je `INT_16`. Dalším důležitým atributem pro správnou interpretaci binárních hodnot je způsob uložení, který je definován v `UseBigEndianOrder`; `YES` pro big-endian nebo `NO` pro little-endian. Jako výchozí hodnota se bere `NO`.

Posledním rozhodujícím parametrem je orientace dat, která je specifikovaná klíčem `DataOrientation`.

- **VECTORIZED** - jako první jsou všechna data pro první kanál, poté pro druhý atd.;
- **MULTIPLEXED** - první hodnota pro každý kanál, druhá hodnota pro každý kanál atd.

Výchozí hodnota je `MULTIPLEXED`.

3 EEG/ERP Portál

3.1 Koncept

Primárním důvodem vývoje EEG/ERP Portálu byla absence softwaru pro ukládání a správu dat a metadat získaných z EEG/ERP experimentů. Dalším podnětem byla také obecná doporučení INCF⁴, mezi jejíž členy patří i Česká republika. Tato doporučení byla ustanovena na konferenci v prosinci 2007 (Sustainability of Neuroscience Databases) a patří mezi ně např. vytvoření odpovídající webové infrastruktury pro vzájemnou komunikaci mezi výzkumnými skupinami a sdílení dat a metadat z prováděných experimentů.

V současné době neslouží EEG/ERP Portál výhradně jako datové úložiště, ale poskytuje i další funkce komunitního portálu jako je např. možnost sestavení a správy výzkumných skupin nebo rezervace laboratoře na konkrétní datum a čas. Systém nabízí kompletní správu experimentů, scénářů, dat i metadat, která lze také stáhnout. Správou je myšleno vytváření, editace a mazání.

Aplikace EEG/ERP Portálu je přístupná pouze registrovaným uživatelům, jejichž možnosti jsou omezeny sadou restrikcí, která je určena rolí uživatele v systému. Portál disponuje následujícími uživatelskými úrovněmi:

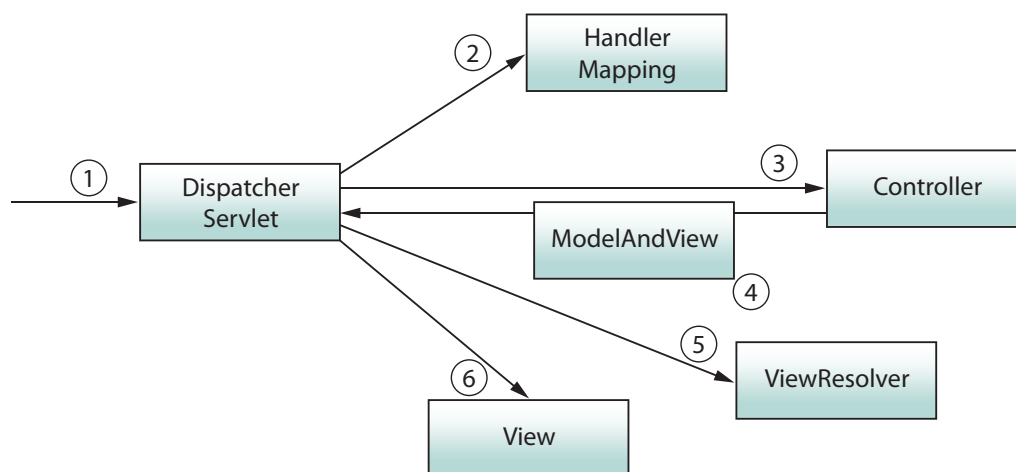
- **Neregistrovaný** - uživatel bez účtu má k dispozici pouze informace z homepage a možnost registrace.
- **Registrovaný** - má pouze práva pro čtení a download dat a metadat v rámci EEG/ERP experimentů. Nemůže vytvářet vlastní scénáře ani experimenty.
- **Experimentátor** - oproti registrovanému uživateli může tento vytvářet a dále spravovat vlastní experimenty. Má přístup k osobním údajům měřeného subjektu.
- **Administrator skupiny** - experimentátor, který vytvořil vlastní výzkumnou skupinu, je automaticky jejím administrátorem. Může měnit oprávnění uživatelů uvnitř skupiny a také může přidat nového člena.

⁴International Neuroinformatics Coordinating Facility je mezinárodní organizace se sídlem ve Stockholmu.

EEG/ERP Portál je od roku 2011 registrován v NIF⁵ na úrovni 2.5. V rámci této úrovně jsou o EEG/ERP Portálu poskytovány jednak obecné informace, jako je koncept, vývojářský tým a kontaktní údaje, ale také data z experimentů a scénáře. Toto začlenění mezi zdroje NIF vede především ke sdílení informací v širším okruhu vědeckých skupin, které se rovněž zabývají neuroinformatickým výzkumem.

3.2 Technologie

Jedná se o webovou aplikaci psanou v JSP. Portál je postaven na 3-vrstvové a MVC architektuře za pomoci frameworku Spring MVC (vizte Obrázek 3) s pluginy Spring Security a Spring Social. Datová část EEG/ERP Portálu využívá databázi běžící na systému Oracle. Objektově relační mapování nad touto databází zajišťuje framework Hibernate.



Obrázek 3: Workflow ve Spring MVC aplikaci.

Architektura MVC frameworku Spring zajišťuje vysokou úroveň abstrakce a tedy i dobrou údržbu a rozšiřitelnost aplikace. Tzv. životní cyklus znázorněný na Obrázku 3 lze popsat následovně:

- *DispatcherServlet* (1) zaregistruje příchozí požadavek.
- *DispatcherServlet* hledá odpovídající obsluhu pro požadavek pomocí *HandlerMapping* (2).

⁵Neuroscience Information Framework je katalog neuroinformatických zdrojů.

- Jakmile je nalezen controller, *DispatcherServlet* předá požadavek ke zpracování komponentně *Controller* (3).
- Na základě řídicí logiky controller předá odpověď v objektu *ModelAndView* (4) zpět do *DispatcherServlet*.
- Nyní *DispatcherServlet* přiřadí view pomocí *ViewResolver* (5).
- Posledním krokem je spárování modelu se získaným view v komponentě *View* (6).

Spring Social je plugin pro Spring framework, který umožňuje propojení mezi webovou aplikací a API sociální sítě. K dispozici jsou konektory pro Facebook, Twitter, LinkedIn a GitHub. Pomocí Spring Social je do EEG/ERP Portálu integrována možnost přihlášení přes Facebook účet nebo LinkedIn.

Autentifikaci a restrikci přístupu na základě uživatelských rolí zajišťuje modul Spring Security. Omezení přístupu je definováno konfiguračním XML souborem. Informace o uživateli (jako např. přihlašovací jméno a heslo) jsou spolu se sadou pravidel pro přístup uloženy v databázi, v konfiguračním XML se proto definice práv získávají SQL dotazem. Restrikce je možné uplatnit jak na celou webovou stránku, tak jen na její část.

HTML stránky prezentační vrstvy jsou generovány pomocí JSP coby nadstavbou nad JavaServlety s využitím JSTL [5]. Vizuální část je dotvořena externími CSS styly, které zajišťují vzhled a pozicování. Některé sekce EEG/ERP Portálu obsahují interaktivní prvky ovládané JavaScriptem; sem lze zařadit i použité komponenty jQuery UI.

Framework Hibernate zajišťuje objektově-relační mapování mezi Java objekty a entitami v relační databázi běžící na systému Oracle. Perzistentní vrstvu tvoří POJO objekty.

4 Specifikace problému

Otázka vizualizace EEG/ERP signálu získaného při měření experimentu nebyla dosud řešena, protože neuroinformatická skupina na KIV Fakulty aplikovaných věd ZČU používá pro své potřeby software Brain Vision Analyzer, který poskytuje možnost vizuální rekonstrukce EEG/ERP signálu.

Začleněním EEG/ERP Portálu mezi neuroinformatické zdroje NIF Registry se rozšiřuje pole působnosti i mezi další výzkumné skupiny, protože v rámci NIF jsou poskytovány scénáře i samotné experimenty. Nelze se ani spoléhat, že ostatní neuroinformatické skupiny používají stejný software, proto je vizualizace EEG/ERP signálu, který je uchovávan v binární podobě `.eeg` souboru, logickým řešením, jak poskytnout bližší informace o průběhu experimentu.

Před realizací vizualizace EEG/ERP signálu je nutné potřebná data nejprve načíst, aby bylo možné s nimi dále manipulovat. Jejich vnitřní struktura je detailně popsána v sekci 2.2.

4.1 Požadavky

Hlavním požadavkem na vizualizaci EEG/ERP signálu je její začlenění do EEG/ERP Portálu, aby byla přístupná přímo ve webové aplikaci, kde bude sloužit pro online náhled zaznamenaného signálu. Tím odpadá nutnost stažení datových souborů a použití Brain Vision Analyzeru či jiné aplikace pro zobrazení EEG signálu.

Vizualizace EEG/ERP signálu formou interaktivní animace byla druhým požadavkem. Tento přístup poskytuje uživateli značně variabilní ovládání - pozastavení a opětovné spuštění vizualizace nebo změna rychlosti vykreslování. Ovládání všech uvedených funkcí musí být co nejjednodušší. S tímto a také s předchozím požadavkem souvisí nároky na kompatibilitu s co nejširším spektrem používaných webových prohlížečů, protože právě na jejich interpretaci závisí každá část frontendu webové aplikace.

Proces načítání dat a metadat EEG/ERP signálu bude realizován v programovacím

jazyku Java, na kterém je postaven i EEG/ERP Portál. Dále je vhodné pamatovat na možné budoucí využití načítání těchto dat i v rámci jiných aplikací a z toho důvodu koncipovat celý proces jako samostatnou knihovnu. Struktury, ve kterých budou načtená data a metadata uložena, ani proces načítání nejsou blíže specifikovány, společnými rozhodujícími parametry jsou však efektivnost, časová náročnost a v případě datových struktur i přístupnost.

4.2 Přehled technologií

Vizualizace EEG/ERP signálu závisí především na výběru technologie, protože její limity určují zároveň hranice vizualizace. Začlenění do EEG/ERP Portálu zužuje výběr na oblast webových technologií, které musí splňovat další dvě podmínky: kompatibilitu napříč spektrem používaných prohlížečů a možnost vytvoření interaktivní vizualizace na bázi animace. Zmíněná kritéria splňuje Macromedia Flash, SVG nebo HTML5 s použitím Canvas API. Účelem této podkapitoly je seznámení s jednotlivými technologiemi, představení výhod i záporů každé z nich a závěrem jejich vzájemné porovnání, na jehož základě je vybrána jedna pro realizaci vizualizace EEG/ERP signálu.

4.2.1 Adobe Flash

Flash, nebo také formálně Adobe Flash a ještě předtím Macromedia Flash, je multiplatformní platforma pro tvorbu interaktivního obsahu webu, tzn. animace, přehrávače audia a videa. Dříve, řádově před deseti lety, byl Flash prakticky jediným nástrojem pro tvorbu RIA aplikací, ovšem s nástupem nových technologií se jeho použití začalo omezovat jen na vybrané prvky webu či na online hry, kde je stále v dominantní pozici. Vývoj započala Macromedia, která byla později odkoupena společností Adobe Systems, a proto jsou všechny produkty Macromedia uváděny pod známkou nového vlastníka.

Flash disponuje vlastním objektově orientovaným skriptovacím jazykem Action Script (momentálně verze 3.0), který vychází ze standardizovaného ECMAScriptu. Umožňuje definování interakcí, animace, zpracování dat. Obsahuje jak základní, tak i kom-

plexní datové typy jako např. Array, Vector nebo Shape [13]. Jazyk je kompilován do byte-kódu, který je prováděn pomocí ActionScript Virtual Machine (AVM)⁶. Kompilátor pro ActionScript je obsažen ve vývojovém prostředí Adobe Flash CS3 (případně CS4) nebo Adobe Flex Builder. Výsledný byte-kód je spolu s grafickými prvky součástí výstupního SWF souboru a ten je spouštěn buď přehrávačem Flash Player nebo programem AIR.

V rámci flashových animací je možné využít rastrovou i vektorovou grafiku. K dispozici jsou i textové objekty, s nimiž lze na časové ose dále manipulovat, a to podle předem daného scénáře (automatické vykreslení popisků os) nebo na základě uživatelských akcí. Pro přehrávání takových animací na webu je potřeba zásuvný modul Flash Player, jehož poslední vydaná verze je 11.2. Jelikož se nejedná o nativní součást webového prohlížeče, je zde problém platformní závislosti. Společnost Adobe nabízí instalační balíčky pro Windows, Mac OS X, Linux a Solaris; pro Windows 7 a distribuce Linuxu jsou k dispozici 32 i 64-bitové verze. Zároveň je zde archiv, kde jsou dostupné i starší verze Flash Player. Soupis operačních systémů, pro které je dostupná aktuální a předchozí verze pluginu Flash Player, je uveden v Tabulce 1.

Verze FP	Operační systémy
Flash Player 11.x	Mac OS X 10.4+ (procesory Intel) Windows XP/Vista/7 Windows Server 2003/2008 Red Hat 5+ Open SUSE 11+ Ubuntu 9.10+
Flash Player 10.1.102.64	Mac OS X 10.4+ (procesory Power PC) Windows 2000

Tabulka 1: Dostupnost posledních dvou verzí Flash Player pro operační systémy.

ActionScript 2.0 již poskytoval komplexní datový typ XML a metody pro manipulaci s ním. Tím byla zaručena možnost základní konfigurace či předávání potřebných hodnot pro vykreslení animace soubory ve formátu XML. Nový ActionScript 3.0 nabízí revidovaný XML objekt založený na standardu E4C; uzly a atributy jsou zpřístupněny jinak než ve verzi 2.0.

⁶AVM je nativní součástí přehrávače Flash Player nebo aplikace AIR.

4.2.2 SVG

Standard SVG je vyvíjen od roku 1999 a v současné době (květen 2012) je unifikován W3C konsorciem ve verzi 1.2. Formát SVG je explicitně navržen pro oblast webové grafiky, kde měl poskytnout možnost interpretace vektorové grafiky, což doposud nebylo realizovatelné. Vedle plnohodnotné podpory vektorové grafiky dává SVG k dispozici také nástroje pro práci s rastrovými obrazy a pro pokročilou manipulaci s textovými objekty. Nejdůležitějším aspektem, kvůli kterému je SVG adeptem pro vizualizaci EEG/ERP signálu, je schopnost vytváření interaktivních animací, přičemž lze kombinovat rastrové i vektorové prvky.

Zkratka SVG označuje jak formát souboru, tak i značkovací jazyk založený na struktuře XML, s jehož pomocí je tvořen zápis grafických objektů. Jednotlivé objekty webové grafiky formátu SVG jsou součástí DOM⁷, který umožňuje přístup či modifikaci obsahu dokumentu, struktury nebo stylu. Provázanost SVG objektů s DOM hraje roli především při renderování grafiky ve webovém prohlížeči. Pokud je atribut SVG objektu modifikován, prohlížeč tuto změnu detekuje v DOM a automaticky překreslí scénu. Další výhodou je přístup ke všem grafickým objektům scény, díky čemuž lze s každým objektem manipulovat v reálném čase např. pomocí JavaScriptu. Naproti všem výhodám reprezentace v DOM je zde i podstatný zápor, a to paměťová náročnost, protože při načtení stránky si webový prohlížeč alokuje paměť pro uchování stromu DOM, kde jsou uloženy všechny elementy dokumentu.

Animované SVG je možné vytvořit za použití následujících technologií:

- JavaScript - umožňuje přístup k objektům grafiky a jejich modifikaci;
- CSS3 - rozšířené selektory a nová pravidla pro animace;
- SMIL - extenze XML přímo určená pro publikaci multimédií na webu.

Animování SVG pomocí JavaScriptu je možné díky reprezentaci grafických objektů scény také v DOM stromu, skrz nějž lze k těmto objektům přistupovat a manipulovat s nimi v reálném čase podle předem definovaného scénáře nebo na základě odezvy na uživatelské akce. Navíc tato varianta není těsně svázaná s typem a verzí používaného webového prohlížeče, alespoň co do podpory JavaScriptu, ve své podstatě se totiž jedná o programovou manipulaci s atributy elementů scény [11].

⁷Document Object Model je objektový model dokumentu reprezentovaný stromem.

CSS3 nabízí rozšířené selektory a nová pravidla přímo určená k animování objektů. Tento způsob je opět závislý na manipulaci s DOM objekty, tentokrát však s minimálním použitím JavaScriptu, jehož použití se omezuje pouze na definice obsluhy akce vyvolané uživatelem jako je např. kliknutí myši na objekt. CSS3 zavádí ještě jednu podstatnou novinku, a to tzv. moduly. Modul je samostatný souhrn standardů vyčleněný mimo hlavní specifikaci a jedním z modulů je i animace. Bohužel je animování na bázi CSS3 silně závislé na podpoře prohlížečů, která není v současné době tak rozsáhlá, aby dokázala pokrýt alespoň 80% uživatelů. Obecně lze říci, že nové funkce obsažené v CSS3 mají nejlepší podporu v prohlížečích na jádře WebKit⁸; detailní přehled na Obrázku 4.

	IE	Firefox	Chrome	Safari	Opera
		3.6			
	6.0	9.0			
	7.0	10.0			
	8.0	11.0	17.0	5.0	
Aktuální verze	9.0	12.0	18.0	5.1	11.6
Následující verze	10.0	13.0	19.0	5.2	12.0
Budoucí verze		14.0	20.0		

■ nepodporuje
■ podporuje
■ není znám o

Obrázek 4: Přehled podpory CSS3 animace, celkem 53.91%. [2]

Protože ještě není vydaná finální verze standardu a zodpovědná skupina W3C konsorcia stále provádí změny ve specifikaci, používají vývojářské týmy webových prohlížečů pro prozatímní implementaci podpory tzv. vendor prefixes (přehled uveden v Tabulce 2), které odlišují způsob interpretace v konkrétním prohlížeči [8].

Vendor prefix	Jádro	Prohlížeč
-moz-	Gecko	Mozilla Firefox
-ms-	Trident	Internet Explorer 9
-o-	Presto	Opera
-webkit-	WebKit	Safari, Chrome

Tabulka 2: Přehled vendor prefixes pro podporu CSS3.

Posledním možným způsobem animace SVG je použití SMIL, jazyka na bázi XML vyvinutého W3C konsorciem čistě pro účel vytváření a publikování multimediálního obsahu na webu. SMIL je příkladem deklarativní animace⁹ zaměřené výlučně na

⁸Na jádře WebKit jsou založeny např. prohlížeče Chrome nebo Safari.

⁹Vztaženo k deklarativnímu programování, princip je stejný.

stavový popis nikoliv však průběhu, jelikož způsob provedení je ponechán implementaci na straně klienta. SMIL animaci lze rozdělit na několik typů podle toho, co je u daného objektu modifikováno:

- číselné atributy elementu (rozměry, zakřivení);
- transformační elementy (translace, rotace);
- barvy (výplň, přechod, alpha);
- pohyb (lineární, po křivce).

	IE	Firefox	Chrome	Safari	Opera
		3.6			
	6.0	9.0			
	7.0	10.0			
	8.0	11.0	17.0	5.0	
Aktuální verze	9.0	12.0	18.0	5.1	11.6
Následující verze	10.0	13.0	19.0	5.2	12.0
Budoucí verze		14.0	20.0		

■ nepodporuje
■ podporuje
■ není známo

Obrázek 5: Přehled podpory SMIL animace, celkem 53.93%. [2]

4.2.3 HTML5 a Canvas API

HTML5 navazuje na HTML 4.01, odstraňuje některé zastaralé elementy, zjednodušuje pravidla korektního zápisu značek a obsahuje také zcela nové tagy. Vyjma změn značkovacího jazyka přináší i nová API, některá začleněná přímo do standardu HTML5, jiná mají vlastní dokument specifikace.

Nové rozhraní Canvas API je specifikováno vlastním standardem mimo HTML5, přesto se do něj často začleňuje. Jedná se o tzv. plátno poskytující možnost interaktivního kreslení přímo uživatelem, renderování předem definované scény nebo vytváření animace. Canvas pracuje s rastry, nepodporuje vektorovou grafiku. Jeho rozhraní poskytuje metody pro manipulaci s kontextem, nikoliv však samotný nástroj, ovládání plátna (kreslení, mazání, změna velikosti a pozice) je proto možné jen JavaScriptem. Canvas dobře reaguje na externí CSS styly i inline pravidla, podporuje i průhlednost [10].

Původně bylo kreslicí rozhraní představeno společností Apple, která jej vyvinula pro použití ve svém operačním systému Mac OS X pro WebKit komponenty¹⁰. Později bylo rozhraní Canvas implementováno webovými prohlížeči s jádrem Gecko a Operou a následně jeho další vývoj převzala skupina WHATWG, která začala s jeho standardizací v nově navržené specifikaci tzv. next-generation webových technologií.

Canvas API poskytuje také možnost umístění textu do kontextu plátna. Pro Gecko 1.9.1 lze definovat přesný typ použitého fontu, jehož deklarace odpovídá syntaxi CSS pravidla `font-family`. Jako výchozí font se používá `sans-serif` o velikosti 10px. Dalším atributem písma je umístění základní dotažnice¹¹, které lze nastavit např. na `top`, `middle` a další. Výchozí hodnotou tohoto atributu `textBaseline` je `alphabetic`.

Plátno je k dispozici v DOM pod svým jedinečným ID, má-li ho definováno. Přes tento identifikátor k němu lze přistupovat pomocí JavaScriptu a manipulovat s jeho kontextem. Kontextem je myšlen celý souhrn grafických objektů zobrazených na plátně (text, křivky, tvary, rastrové obrázky), není ale možné přistoupit zvlášť k jednotlivým objektům [4, Kapitola 2]. Jakmile je vykreslena např. křivka, nelze s ní dále jakkoliv manipulovat; není možné změnit její barvu, tah nebo pozici, lze pouze vymazat obdélníkovou oblast, ve které se na plátně vyskytuje.

4.3 Porovnání technologií

Pro objektivní posouzení vhodnosti dané technologie se bere v potaz podpora napříč prohlížeči. Parametrem pro zkoumání podpory v konkrétní verzi je její procentuelní zastoupení na trhu, přičemž nejmenší podíl činí 0.63%. Celkem tato statistika zahrnuje 89.9% uživatelů na internetu. Dalším srovnávacím měřítkem je výkon - zda je animace plynulá a má odpovídající hardwarové nároky. Pro srovnání výkonnosti poslouží částicový benchmark.

U SVG hraje roli zejména podpora ze strany prohlížečů, jejíž přehled je na Obrázku 6. Oproti Canvas API (viz Obrázek 7) je celková podpora o cca 10% nižší. Další

¹⁰WebKit komponentami jsou realizovány např. widgety na plochu a také tvoří část prohlížeče Safari.

¹¹Základní dotažnice neboli účárí písma je pomyslná linka, na kterou desadají jednotlivá písmena.

minus SVG je jeho paměťová náročnost, protože uchovává objekty scény v DOM. Možnost vykreslení EEG/ERP signálu ve vektorových křivkách je pro současný záměr vizualizace nevyužitelná, proto se varianta použití SVG zamítá.

	IE	Firefox	Chrome	Safari	Opera
		3.6			
	6.0	9.0			
	7.0	10.0			
	8.0	11.0	17.0	5.0	
Aktuální verze	9.0	12.0	18.0	5.1	11.6
Následující verze	10.0	13.0	19.0	5.2	12.0
Budoucí verze		14.0	20.0		

■ nepodporuje
■ podporuje
■ není známo

Obrázek 6: Přehled podpory SVG, celkem 65.79%. [2]

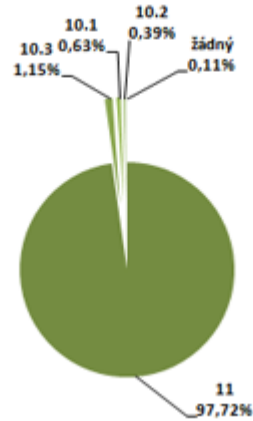
	IE	Firefox	Chrome	Safari	Opera
		3.6			
	6.0	9.0			
	7.0	10.0			
	8.0	11.0	17.0	5.0	
Aktuální verze	9.0	12.0	18.0	5.1	11.6
Následující verze	10.0	13.0	19.0	5.2	12.0
Budoucí verze		14.0	20.0		

■ nepodporuje
■ podporuje
■ není známo

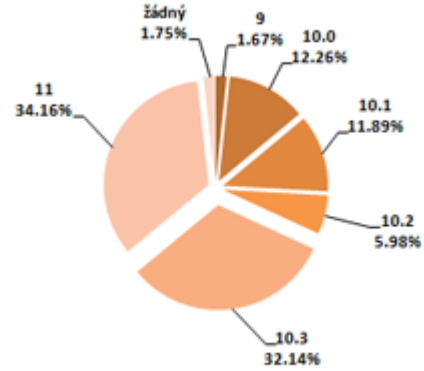
Obrázek 7: Přehled podpory Canvas API, celkem 75.18%. [2]

Pro zbývající dvojici technologií, Flash a Canvas API, nelze uplatnit kritérium podpory stejným způsobem. Flash má značnou nevýhodu v tom, že k přehrání animací v něm vytvořených potřebuje zásuvný modul Flash Player. Ne vždy je tento plugin na počítači k dispozici nebo může být zastaralý. Statistika založená na údajích z RIA Stats [14] na Obrázku 8 ukazuje, že aktuální verze (Flash Player 11) ne vždy jednoznačně převažuje nad těmi staršími; pouze v Chrome tomu tak je, protože obsahuje nejnovější Flash Player přímo v instalačním balíčku. V tomto ohledu je lepší volbou Canvas API. Navíc technologii Flash nepodporují mobilní zařízení Apple, kdežto mnoho mini-prohlížečů pro Android, Symbian nebo iOS implementují podporu Canvas API, byť v menším rozsahu než u desktopových prohlížečů.

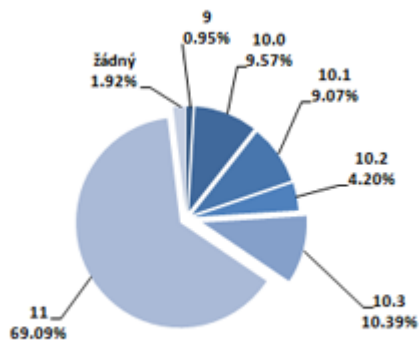
Posledním srovnávacím kritériem je výkon testovaný v celé pětici prohlížečů mnou navrženým částicovým benchmarkem. Animace pohybu jednotlivých částic je zapísána ekvivalentním způsobem v ActionScriptu 3.0 pro Flash, SMIL pro SVG a po-



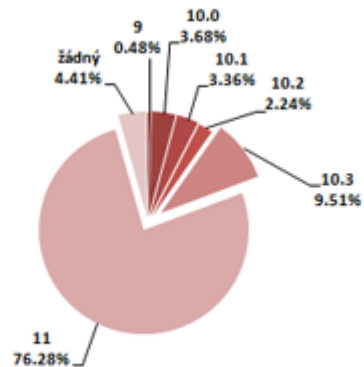
(a) Chrome



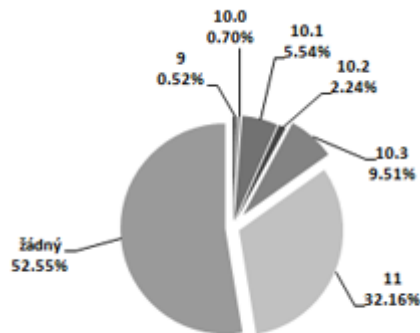
(b) Firefox



(c) Internet Explorer



(d) Opera



(e) Safari

Obrázek 8: Podíl verzí Flash Player v prohlížečích (a), (b), (c), (d) a (e).

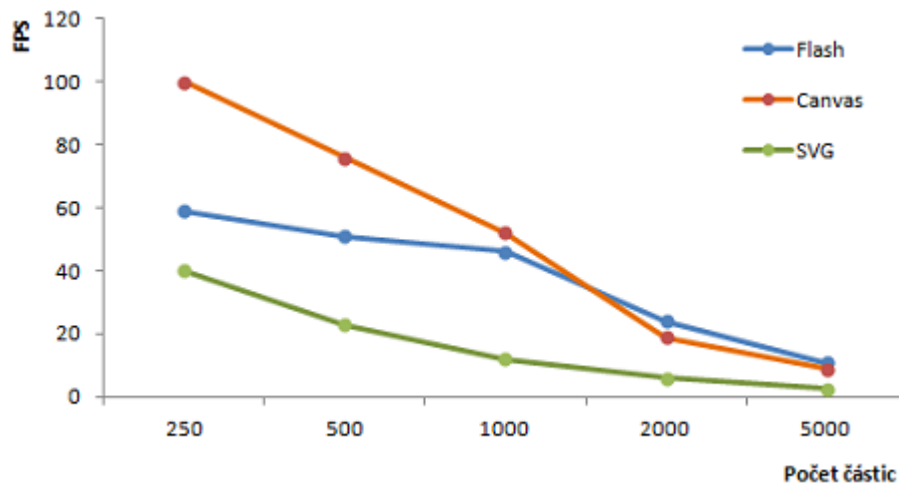
mocí JavaScriptu pro Canvas API. Technologie jsou otestovány stejnou sadou: 250, 500, 1000, 2000 a 5000 částic. V Tabulce 3 jsou uvedeny hodnoty váženého průměru z naměřených hodnot FPS, přičemž za váhové koeficienty posloužilo procentuelní

zastoupení daného prohlížeče na trhu.

Částice	Canvas	Flash	SVG
250	100	59	49
500	76	51	23
1000	52	46	12
2000	19	24	6
5000	9	11	3

Tabulka 3: Přehled vendor prefixes pro podporu CSS3.

Výsledky benchmarku jasně ukazují, že do 1000 grafických objektů scény je výkonější Canvas API. Subjektivním hodnocením je i plynulost animace, která byla pro první 3 sady testů u Canvas i Flashe srovnatelná. Hardwarové nároky (systémová paměť a vytížení procesoru) se začaly rapidně zvyšovat až při 2000 animovaných částic, ovšem při předchozích sadách si renderování grafiky vystačilo s dosud alokovanými zdroji. Oproti Canvas a Flash Playeru výrazně zaostává SVG, proto se jeho vyřazení jen potvrzuje. Závěr je takový, že pro vizualizaci EEG/ERP signálu bude použita technologie Canvas API ve spojení s JavaScriptem, jelikož tato kombinace vyšla dobře v obou kritériích.



Obrázek 9: Výsledky benchmarku pro Flash, Canvas a SVG.

5 HTML5 a Canvas API

HTML5 je dalším milníkem ve vývoji webových standardů a přestože ještě není ustanovena jeho finální specifikace¹², vývojářské týmy webových prohlížečů implementují jeho nové funkce již dnes. Jeho vize sebou nese mnoho nových funkcí, přičemž některé z nich dosud chyběly obzvláště při vývoji rozsáhlejších webových aplikací, kde se principiálně očekává vyšší interaktivita mezi rozhraním a uživatelem.

5.1 Vize a principy HTML5

Nová vize HTML5 je založena na principech kompatibility, interoperability a co možná nejobecnějšího přístupu spolu s upřednostněním uživatelů. Kompatibilita se netýká zcela nových API a komponent, které nemají historického předchůdce v dřívějších verzích HTML, více jde o poskytnutí nové struktury s tím, že bude stejným způsobem interpretován starší zápis i ten nový. Příkladem je zavedení nového tagu `<header>` [7] pro označení záhlaví webové stránky. Na základě analýzy milionů různých stránek, za níž stojí Google, bylo zjištěno, že vysoké procento programátorů a kodérů používá pro označení záhlaví tag `div` s ID `header`. Nový tag poskytuje nástroj tam, kde je už teď naznačena zažitá struktura.

5.1.1 Sémantické elementy

HTML5 definuje nové tagy, které popisují obsah dokumentu i na základě jeho sémantiky. Používání těchto nových značek, jejichž ilustrační soupis je uveden v Tabulce 4, nepřináší vyjma oddělení formy od obsahu žádné okamžité výhody, napomáhá však zjednodušení a zpřehlednění kostry webové stránky a v budoucnu lze počítat s indexací těchto elementů vyhledávači [4, Kapitola 1].

Všechny tyto elementy jsou plně stylovatelné pomocí CSS, což podporuje oddělení formy od obsahu i po vizuální stránce. Rovněž je každý ze zmíněných tagů přístupný přes selektování metodou JavaScriptu `getElementById`. Takový přístup je poněkud

¹²Rok 2012 byl ohlášěn jako termín pro CR. Druhým důležitým datem je rok 2022, který představuje adepta pro PR [6].

Element	Popis
article	obsah článku nezávislý na zbytku stránky
aside	související obsah nebo citace
footer	obsah zápatí celé stránky nebo jen její sekce
header	obsah záhlaví celé stránky nebo jen její sekce
nav	navigace nebo obecně jakékoliv menu
section	sekce webové stránky

Tabulka 4: Nové sekční elementy dostupné v HTML5.

zastaralý, dnes se více využívají selektory frameworků JQuery, MooTools nebo Prototype. Společně se sémantickými prvky je v HTML5 k dispozici také nový způsob vyhledávání elementů v rámci dokumentu, a to pomocí rozhraní *API Selector*.

Celkově přichází HTML5 s konceptem mikroformátů. Jedná se o způsob anotace obsahu strojově čitelnými značkami, které vyjadřují sémantický význam textu (soudnice sídla firmy, telefonní kontakt apod.). V této souvislosti se mluví o vizi Webu 3.0. Výhledová prospěšnost je zřejmá - zobecnění a zavedení sémantiky může napomoci zvýšení relevantnosti výsledků vyhledávání.

5.1.2 Odstranění zásuvných modulů

Zjednodušení interoperability spočívá v rozšíření specifikace a jejím zpřesnění¹³. Jak se postupně tvořil dokument specifikace, spousta nativních funkcí byla z HTML5 vyčleněna do samostatných standardizovaných dokumentů, např. Web Storage nebo Canvas 2D. Nová rozhraní API a funkce HTML5 poskytují podporu pro realizaci takových postupů, jež byly dříve možné výhradně s použitím externích pluginů a appletů, např. webové rozhraní pro kreslení. Používání zásuvných modulů sebou nese typická rizika:

- Není vždy možné na daném počítači nainstalovat požadovaný zásuvný modul.
- Zásuvné moduly mohou být cíleně deaktivované nebo přímo blokové (např. Apple iPad se nedodává s pluginem pro Flash).
- U zásuvných modulů existuje potenciální bezpečnostní riziko.¹⁴

¹³Současná podoba specifikace standardů HTML5 včetně vyčleněných rozhraní API čítá přes 900 stran.

¹⁴Např. metodou clickjackingu je možné vzdáleně přes Flash plugin aktivovat webovou kameru či mikrofon bez uživatelského vědomí.

- Může být problém začlenit obsah zásuvných modulů do zbytku dokumentu tak, aby data obsažená v pluginech byla transparentní pro webové vyhledávače.

Kvůli těmto důvodům se vize HTML5 drží paradigmatu bez zásuvných modulů, čímž si udržuje nezávislost na médiu a platformě. Tam, kde byl původně potřeba zásuvný modul, dnes nastupuje HTML5 a nativní podpora nových funkcí ze strany prohlížečů.

5.2 Specifikace

Za specifikaci HTML5 nesou zodpovědnost hned 3 organizace, přičemž každá nese odpovědnost za jinou část:

1. **WHATWG** - jelikož tato organizace spolupracuje se společnostmi vyvíjející prohlížeče, zodpovídá i za vývoj HTML a API pro další development webových aplikací. Otevřená spolupráce se společnostmi jako je Google, Apple nebo Opera přináší poznatky přímo z praxe a tím činí HTML5 použitelným od samého začátku.
2. **W3C** - součástí konsorcia je i skupina zodpovědná za dodání specifikace HTML5.
3. **IETF** - HTML5 definuje zcela nové rozhraní API WebSocket, které je postavené na protokolu WebSocket a právě ten je dílem IETF. Proto se jedna z pracovních skupin IETF podílí obzvláště na vývoji API WebSocket a některých dalších rozhraní.

Jak už jsem uvedla výše, HTML5 sebou ve specifikaci přináší nové tagy a API; především pro snazší začlenění a správu multimediálního obsahu. Také je k dispozici rozhraní geolokace či globální podpory všech světových jazyků, což byl doteď problém především z typografického hlediska.

Další změnou je vypuštění mnoha zastaralých značek; obzvláště takových, které suplovaly funkci stylů a které jsou plně nahraditelné užitím CSS. Mezi takové patří např. `center` nebo `font`. Nový standard přináší také zjednodušenou `DOCTYPE` hlavičku a přehlednější definici kódování .

5.3 Podpora webových prohlížečů

Inovativní přístup HTML5 je postaven především na podpoře webových prohlížečů. Jejich vývoj vykročil směrem k řešení problémů, jakými bylo kupříkladu zpracování dat a serializace pomocí JSON nebo validní interpretace nových značek.

Když pominu jednoduché prezentace bez jakékoliv řídicí logiky či interaktivity, pak v tom zbytku najdeme zpravidla aspoň jeden externí skript (v zásadě může být i inline vložený), který pomocí JavaScriptu zajišťuje kontrolu formulářových polí, detekci prohlížeče nebo se stará o vizuální efekty. Nehledě na to, že API poskytovaná HTML5 jsou ovládaná JavaScriptem. Stejně jako HTML tagy a CSS styly jsou i soubory JavaScriptu interpretovány prohlížečem. Právě proto je jednou z nejvýznamnějších změn poslední doby evoluce jádra JavaScriptu/ECMAScriptu.

V počátcích patřil JavaScript mezi ryze interpretované jazyky, kdežto nejnovější verze kompilují na úrovni jádra prohlížeče do strojového kódu, což oproti interpretaci poskytuje zrychlení v několika řádech.¹⁵ Tento tah byl učiněn zejména kvůli rostoucím nárokům webových aplikací typu RIA.

Prohlížeč	Název jádra	Popis
Safari 5	Nitro	Od verze Safari 4 se jádro dočkalo optimalizace byte-kódu. Obsahuje nativní context-threaded překladač.
Chrome 5	V8	Jádro dostupné od Chrome 2 je založené na garbage-collection, což zajišťuje vysokou paměťovou škálovatelnost bez přerušení.
IE 9	Chakra	Zaměřeno na kompilaci kódu na pozadí. Výrazný posun oproti IE8.
Firefox 4	JägerMonkey	Kombinace interpretaci s kompilací na základě trace-tree.
Opera 10.60	Carakan	Byte-kód založený na registrech a selektivní kompilace.

Tabulka 5: Jádra JavaScriptu jednotlivých webových prohlížečů.

¹⁵Snahy o kompilování JavaScriptu začaly v roce 2006, kdy Adobe přišlo s kompilačním jádrem JIT (Just-In-Time) a s virtuálním strojem pro ECMAScript (kódové označení Tamarin). Oba nástroje byly začleněny do projektu Mozilla.

Interpretační jádro JavaScriptu každého prohlížeče je svým způsobem unikátní, protože každé z nich je založené na trochu jiném přístupu. Více informací k jednotlivým technologiím používaným pětici nejrozšířenějších prohlížečů je uvedeno v Tabulce 5.

5.4 Canvas API

Jedná se o rozhraní umožňující dynamicky generovat a vykreslovat grafiku, obrázky i složitější animace. Historicky bylo možné podobné pseudoplátno realizovat výhradně s pomocí zásuvných modulů pro Flash, SVG nebo VML, protože neexistoval vhodný HTML element, natož pak nativní podpora prohlížeči.

5.4.1 Podpora Canvas

Podpora HTML elementu Canvas ze strany webových prohlížečů je výborná a pro účely vizualizace zcela dostačující. Aktuální verze pěti nejpoužívanějších prohlížečů prošly v testech podpory s pozitivním výsledkem a jejich vývojářské týmy i nadále slibují postupné zlepšování podpory v každé nadcházející verzi.

	IE	Firefox	Chrome	Safari	Opera
		3.6			
	6.0	9.0			
	7.0	10.0			
	8.0	11.0	17.0	5.0	
Aktuální verze	9.0	12.0	18.0	5.1	11.6
Následující verze	10.0	13.0	19.0	5.2	12.0
Budoucí verze		14.0	20.0		

■ nepodporuje

■ podporuje

■ není známo

Obrázek 10: Přehled podpory Canvas API, celkem 75.18%. [2]

Při pohledu na podporu (viz Obrázek 10) ve dvou releasech předcházejících aktuální verzi prohlížeče je na tom nejhůř Internet Explorer. Jelikož pro Canvas API neexistuje vyjma knihovny ExplorerCanvas žádná adekvátní alternativa, není zde možnost optimalizace pro IE 8 a nižší. Zastoupení právě této verze na trhu měsíc od měsíce klesá průměrně o 0,3 až 0,4 %, zatímco podíl prohlížeče IE9 stejnou měrou roste. Je tedy logickým předpokladem, že uživatelé přecházejí na novější verzi téhož prohlížeče.

Knihovna ExplorerCanvas se objevila v době, kdy poslední release Internet Exploreru neobsahoval podporu rozhraní Canvas, tedy v době verze 8. Za vznikem knihovny stojí Google, nicméně její vývoj končí revizí #73 z března 2010. [9] Od té doby přineslo rozhraní Canvas řadu nových funkcí, s jejichž podporou v této náhradě nelze počítat. ExplorerCanvas je čistě na bázi JavaScriptu a poskytuje jen omezenou podporu transformací 2D kontextu plátna, neobsahuje žádnou podporu práce s textem, tudíž např. popisky os souřadnicového systému by byly v podání ExplorerCanvas neproveditelné. Z uvedených důvodů nedostatečné podpory základních funkcí plátna jsem se rozhodla neimplementovat náhradu ExplorerCanvas pro IE 8.

2012	IE	Firefox	Chrome	Safari	Opera
Březen	18.9%	36.3%	37.3%	4.4%	2.3%
Únor	19.5%	36.6%	36.3%	4.5%	2.3%
Leden	20.1%	37.1%	35.3%	4.3%	2.4%

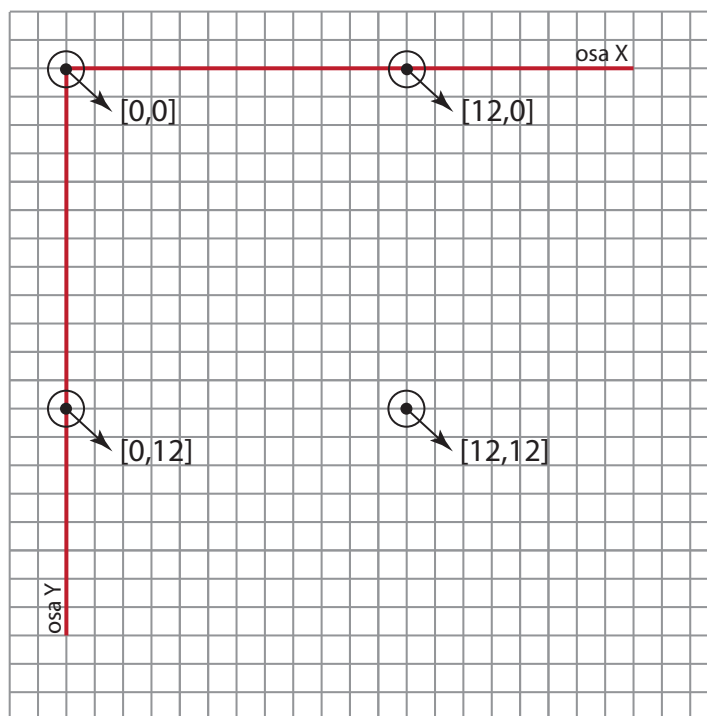
Tabulka 6: Procentuelní zastoupení prohlížečů na trhu v roce 2012.

Z Tabulky 6 navíc jasně vyplývá postupný ústup IE. Jeho podíl na trhu zvolna přebírají ostatní prohlížeče, zejména tedy Chrome¹⁶, a všechny obsahují minimálně ve dvou posledních verzích maximální podporu HTML5 Canvas API. Explicitně jsem se zajímala o podporu v desktopových prohlížečích, některé klony portované na mobilní slibují podporu mnoha novinek obsažených v HTML5 a mezi nimi je i canvas. Pro detailnější přehled doporučuji navštívit web *Can I Use?* [2], kde je množství různých aktualizovaných přehledů.

5.4.2 Práce s rozhraním

Canvas lze chápat jako kreslicí plátno se souřadnicovým systémem, který je ilustrován Obrázkem 11. Počátek je definován v levém horním rohu na souřadnicích $[0, 0]$. Hodnoty souřadnic se zvyšují po jednotlivých pixelech horizontálně na ose X a vertikálně na ose Y.

¹⁶Uvedená statistika je přebraná z webu w3schools.com [1], kde je možné dohledat údaje až z roku 2002.



Obrázek 11: Souřadnicový systém canvas.

Element `<canvas>` je přístupný v DOM na základě svého unikátního id. Pro programové použití plátna je dále nutné získat jeho kontext. Vizualizace EEG/ERP signálu využívá 2D kontext. K manipulaci s plátnem slouží metody poskytující Canvas API, přičemž průběh takové akce lze přirovnat k databázové transakci, protože na začátku se transakce zahájí, provedou se požadované změny, a následně se transakce dokončí provedením tzv. commitu.

Canvas API poskytuje metody pro modifikaci kontextu plátna a lze je rozdělit podle použití do několika skupin:

- transformace;
- cesty;
- tah a výplň;
- rastry.

Implementace Canvas API interně používá transformace přímo ve svém jádru, bez ohledu na právě vykonávané transakce - jedná se o vnitřní optimalizaci. Transformace umožňují operace nad souřadnicovým systémem, např. translaci, rotaci a škálování. Tyto změny upravují výchozí pozici souřadnicového systému.

```
var context = canvas.getContext('2d');  
//ulozeni aktualniho stavu  
context.save();  
//posun kontextu vpravo dolu  
context.translate(70,140);  
//skalovani - dvojnásobny zoom  
context.scale(2,2);  
...  
//obnoveni puvodniho stavu kontextu  
context.restore();
```

Výpis 1: Ukázka transformací kontextu.

Cesta patří mezi základní grafické objekty, může být jednoduchá i složená. Metoda `beginPath` značí počátek cesty a metoda `closePath` cestu uzavírá. Použití cest má význam zejména pro pozdější definici tahu nebo výplně, protože u cesty je plátno schopné rozlišit vnitřní a vnější část vytvořeného objektu. Pro vytvoření cesty slouží následující metody:

- `moveTo` - přesun bez kreslení;
- `lineTo` - přímka;
- `quadraticCurveTo` - kvadratická křivka;
- `bezierCurveTo` - Beziérová křivka;

Vlastnosti křivek určují dodatečné řídicí body a úhly.

Rozhraní Canvas umožňuje nastavení stylu tahu cesty. Lze nastavit šířku tahu v pixelech pomocí `lineWidth`. Vlastnost `lineJoin` definuje vzhled spojnic cest, konkrétní hodnoty jsou `round`, `bevel` a `mitter`. Atribut `strokeStyle` akceptuje anotaci RGB barvy hexakódem, barevný přechod a dokonce i rastrový obraz. Obdobně funguje i vizuální definice výplně, jen s tím rozdílem, že styl je určen atributem `fillStyle`.

6 Vizualizace EEG/ERP signálu

Vizualizaci EEG/ERP signálu lze do jisté míry považovat za samostatný modul, jehož výhodou je variabilita využití a rozšiřitelnost. Navržené řešení není pevně svázané právě s vykreslováním EEG/ERP signálu, ale lze jej použít pro vizualizaci dalších transformací provedených nad signálem, jejichž výsledek nemusí být nutně spojený v čase - může jít o průměrování epoch, počítání waveletové transformace nebo FFT.

Modul vizualizace je umístěn na stránce detailu experimentu, kam svým kontextem spadá. Jeho zobrazení je podmíněno existencí datových souborů VHDR a EEG. Pokud jsou tyto soubory přiloženy u experimentu, je k dispozici plátno pro vykreslení EEG/ERP signálu, v opačném případě je celý modul automaticky skryt. Tato rozhodovací logika je realizována testováním podmínky v JSTL.

6.1 Knihovna pro načítání signálu

Knihovna je naprogramovaná v jazyce Java a pro načítání dat potřebných k vizualizaci EEG/ERP signálu využívá ještě další knihovny. Jelikož VHDR i VMRK soubory mají strukturu shodnou s INI soubory operačního systému Windows, je pro jejich načítání použita knihovna *commons-configuration-1.8*. Konkrétně je využívána třída *HierarchicalINIConfiguration*, která je určena speciálně pro parsování INI souborů. Počítá se s následujícím možným obsahem:

- **sekce** - deklarována v [] vždy výhradně na jednom řádku;
- **parametr** - položka sekce ve formátu `key=value`;
- **komentář** - řádky uvozené ";".

Třída `Vhdr` zajišťuje načtení hlavičkového souboru. Jelikož jsou předem definovány názvy sekcí i parametrů, stačí je zadefinovat jako jazykové konstanty (vizte Výpis 2), s jejichž pomocí lze získat informace z již rozparsovaného hlavičkového souboru v hierarchické struktuře.

Použití *HierarchicalINIConfiguration* je mnohem vhodnější než definovat vlastní pravidla a algoritmus parsování, stejně tak i datovou strukturu, v níž by rozparsovaná

data měla být uložena. HierarchicalINIConfiguration uchovává data z rozparsovaného souboru ve stromové struktuře, přičemž sekce jsou přímými potomky rootu a parametry jsou až potomky jednotlivých sekcí.

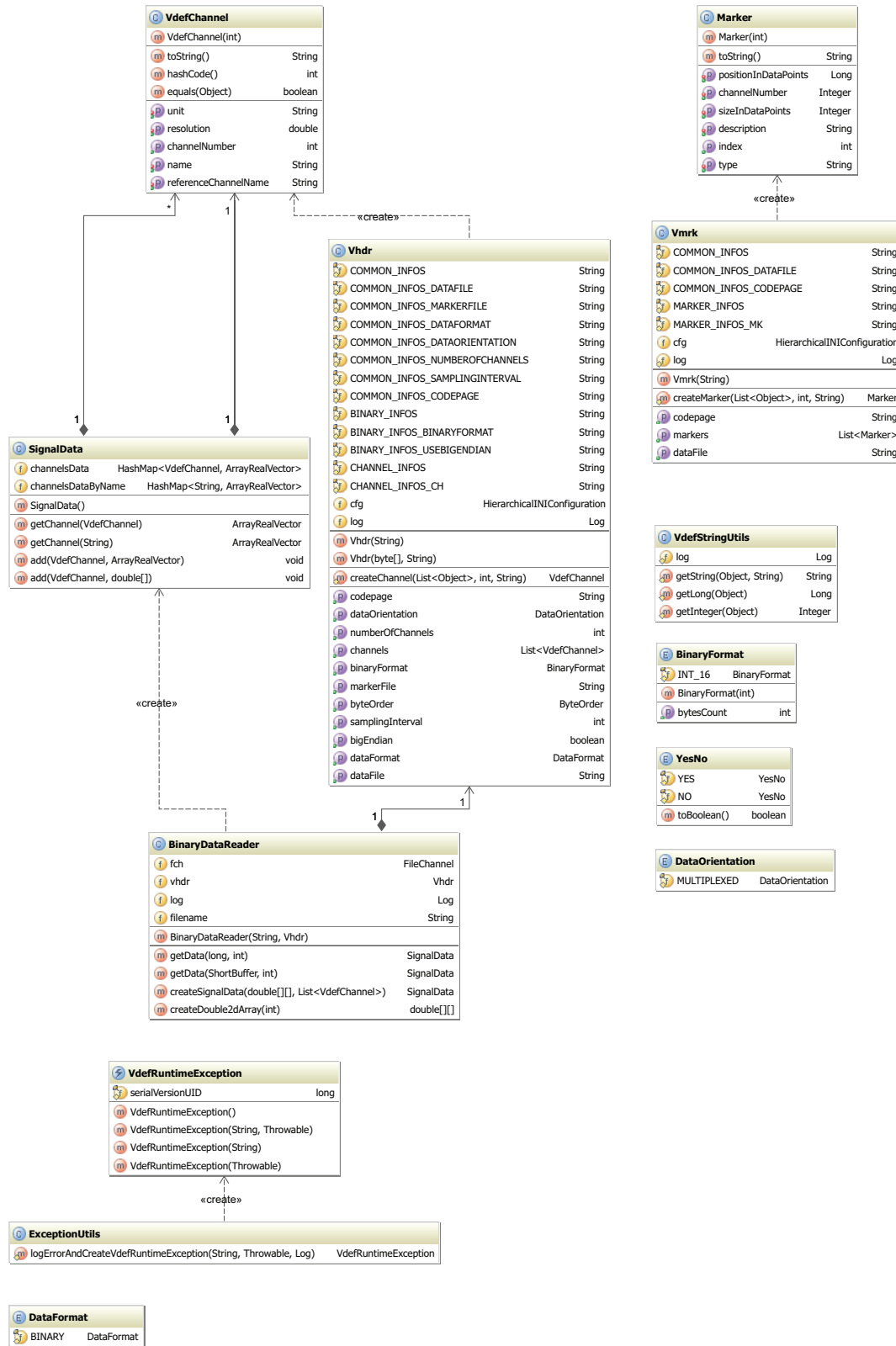
```
COMMON_INFOS = "Common Infos ";
COMMON_INFOS_DATAFILE = COMMON_INFOS + ".DataFile ";
COMMON_INFOS_MARKERFILE = COMMON_INFOS + ".MarkerFile ";
COMMON_INFOS_DATAFORMAT = COMMON_INFOS + ".DataFormat ";
COMMON_INFOS_DATAORIENTATION = COMMON_INFOS + ".DataOrientation ";
COMMON_INFOS_CODEPAGE = COMMON_INFOS + ".Codepage ";
BINARY_INFOS = "Binary Infos ";
BINARY_INFOS_BINARYFORMAT = BINARY_INFOS + ".BinaryFormat ";
BINARY_INFOS_USEBIGENDIAN = BINARY_INFOS + ".UseBigEndian ";
CHANNEL_INFOS = "Channel Infos ";
CHANNEL_INFOS_CH = CHANNEL_INFOS + ".Ch ";
```

Výpis 2: Definice konstant pro parametry a sekce VHDR.

Při načítání sekce [Channel Infos] z VHDR souboru je zároveň vytvářen seznam kanálů ve struktuře `List<VdefChannel>`. Třída `VdefChannel` představuje reprezentaci jednotlivých kanálů; uchovává informaci o jménu, číslu, rozlišení, referenčním označení a jednotkách.

Na základě zjištěných parametrů, které jsou nutné pro správnou interpretaci hodnot uložených v datovém souboru EEG, lze tato data načíst. Načtení a přiřazení dat k jednotlivým kanálům zajišťuje třída `BinaryDataReader`. S ohledem na zachování vztahu dat ke kanálu jsou načtené hodnoty uchovány v mapě `HashMap<VdefChannel, ArrayRealVector>`, kde `VdefChannel` určuje kanál a `ArrayRealVector` reprezentuje data z EEG souboru.

Knihovna pro načítání signálu obsahuje ještě několik tříd, které zajišťují mapování parametrů obsažených ve VHDR souboru. Jedná se např. o přetypování direktivy `UseBigEndian` na logickou hodnotu boolean. Dále je zde připravena třída pro načtení VMRK souboru spolu se strukturou pro reprezentaci seznamu markerů.



Obrázek 12: Class diagram knihovny.

6.2 Kontrola podpory prohlížeče

Jako každá webová technologie má i HTML5 svá omezení. Tím mám na mysli především podporu ze strany prohlížeče, jelikož právě na něm závisí vykreslení webové aplikace u klienta.

Základním kamenem modulu pro vizualizaci je element canvas, který zajišťuje vykreslení EEG/ERP signálu. O podpoře tohoto elementu v rámci nejpoužívanějších prohlížečů a dále jejich jednotlivých verzích jsem psala v předchozí sekci 5.4.1. Na základě zjištěných informací si mohu dovolit zacílit kontrolu na schopnost prohlížeče interpretovat tag `<canvas>`, který se vyskytuje v HTML kódu detailu experimentu. Pro práci s rozhraním je potřeba získat jeho kontext. Pokud není tag `<canvas>` podporován, není ani interpretován, a proto nelze získat jeho kontext. Na tomto předpokladu je postavena detekce prohlížeče s podporou Canvas API.

Kontrola probíhá ve dvou krocích. Tím prvním je pokus o získání kontextu plátna pomocí funkce `isCanvasSupported()` (pro konkrétní strukturu vizte Výpis 3).

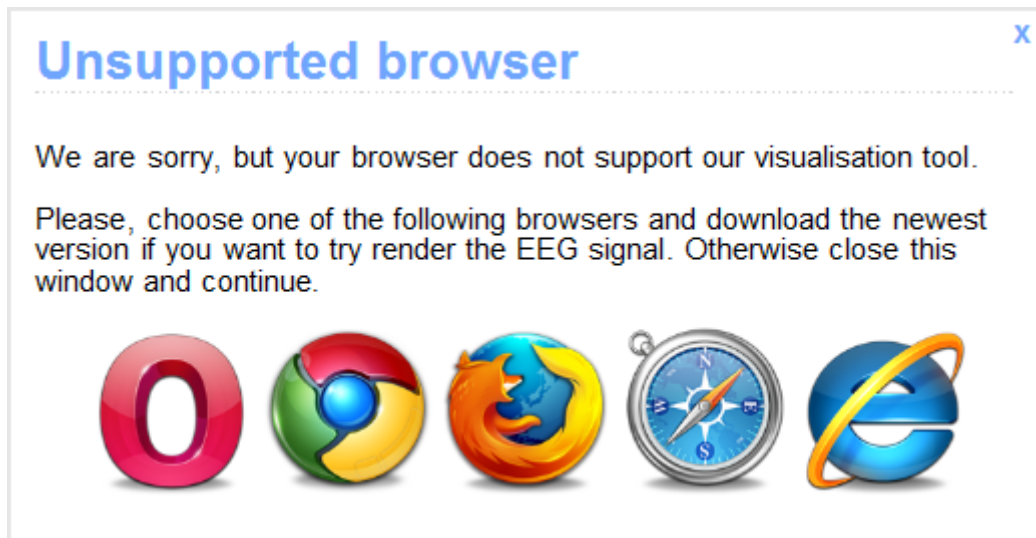
```
function isCanvasSupported() {  
    var elem = document.getElementById( 'canvas' );  
    return !( elem.getContext && elem.getContext( '2d' ) );  
}
```

Výpis 3: Funkce pro kontrolu podpory prohlížeče.

V případě nepodporovaného prohlížeče by druhým krokem mělo být zobrazení alternativního obsahu. Jak jsem však již uvedla výše, nemá cenu vydávat se touto cestou, protože žádný jiný statický prvek nedokáže nahradit `<canvas>`. Aplikace se tedy podrží zobrazení informativního hlášení a doporučí uživateli aktualizaci prohlížeče, pokud má zájem prohlédnout si online vizualizaci EEG/ERP signálu. Okno s hláškou je na Obrázku 13.

Kontrolu je nutné provést hned po načtení stránky, což zajišťuji detekcí přímo v těle `$(document).ready(function())`¹⁷, jejíž kompletní struktura i s komentáři je uvedena ve Výpisu 4. Zobrazení tohoto pop-up okna je řešeno pomocí jQuery a aby

¹⁷Znak \$ naznačuje anotaci jQuery, pomocí níž jsem některé části kódu realizovala.



Obrázek 13: jQuery pop-up okno kontroly nepodporovaných prohlížečů.

uživatelé co nejméně obtěžovalo, jsou nastaveny hned 3 možnosti jeho zavření:

1. křížkem v rohu okna;
2. kliknutím mimo prostor pop-up okna;
3. klávesou ESC.

```

if (!isCanvasSupported()) {
    centerPopup();
    loadPopup();
    $("#visualization").hide();
}

```

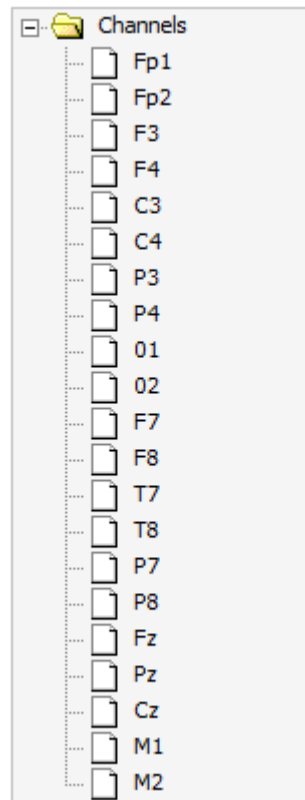
Výpis 4: Fragment kódu pro ošetření .

6.3 Proces vizualizace

Potřebná data pro vizualizaci jsou získána pomocí knihovny pro načítání signálu a předány v objektech JSP stránce. Konkrétně jde o seznam kanálů, na nichž probíhalo měření EEG/ERP experimentu, a pole navzorkovaných hodnot po dobu 10 sekund.

Před samotným vykreslením je nutné vybrat kanál, jehož naměřený EEG/ERP signál se bude vizualizovat. Pro přehledné zobrazení seznamu kanálů slouží strom,

jehož ukázka je na Obrázku 14. Jde o externí komponentu *dhtmlxTree*, která je poskytována pro open source projekty pod GPL licencí, proto je možné ji začlenit do EEG/ERP Portálu. Interaktivitu a renderování stromu zajišťuje JS knihovna.



Obrázek 14: Strom se seznamem kanálů.

Položky stromu mají definovanou akci vázanou na kliknutí levým tlačítkem myši. Tato událost spustí obslužnou JS funkci zajišťující nejprve inicializaci plátna¹⁸ a následné vykreslení hodnot předaných parametrem v podobě pole.

Vizualizace EEG/ERP signálu je realizována postupnou animací pomocí Canvas API. V obslužném kódu JavaScriptu se využívají metody rozhraní `lineTo`, `moveTo` a `stroke`. Signál je reprezentován hodnotami napětí, které jsou vynášeny na ose Y. Kroky na ose X jsou ekvidistantní, protože vzorkování probíhá se stejnou frekvencí po celou dobu experimentu. EEG/ERP signál lze z pohledu vykreslení považovat za cestu složenou z přímků, proto lze pro vizualizaci použít metodu `lineTo`. Síla tahu je nastavena na 1 pixel a spojení mezi jednotlivými liniemi vypadá nejplynuleji při atributu `round`.

¹⁸Inicializace znamená vyčištění kreslicí plochy, vykreslení os s popisky a určení měřítka pro osu Y.

Pod plátnem je k dispozici ovládací lišta, s jejíž pomocí lze animaci pozastavit a opět spustit. Také je zde umístěn slider, který ovládá rychlost vykreslení EEG/ERP signálu. Slider má nastavené rozmezí hodnot od 1 do 100, přičemž jako výchozí je nastavena hodnota 43. Pro změnu rychlosti není potřeba animaci pozastavit, zrychlení či zpomalení je plynulé i za běhu. Rozložení lišty je zobrazeno na Obrázku 15.



Obrázek 15: Inicializované plátno s ovládací lištou.

6.4 Možnosti rozšíření

Modul vizualizace EEG/ERP signálu lze do budoucna obohatit dalšími funkcemi:

- **Markery** Vizualní umístění markerů by bylo určitě ku prospěchu věci především z informativního hlediska, protože v jeho okolí je možné sledovat odezvu subjektu na vnější podnět.
- **Interaktivita** Zvýšení interaktivity značně rozšíří možnosti uživatele, který si teď může signál jen přehrát, ale víc s ním prakticky nenadělá. Přiblížení určitého výseku časové osy a jeho následný export se zdá být praktické.
- **Filtry** Pro některé další vědecké účely se na signál aplikují různé filtry, jejichž výstup by šel také vizualizovat.

7 Závěr

Na základě zadání jsem se seznámila s EEG/ERP Portálem vyvíjeným na KIV. Dále jsem prozkoumala a zhodnotila dostupné technologie umožňující vizualizaci EEG/ERP signálu v rámci webové aplikace. Hodnocením podle daných kritérií prošla nejlépe možnost HTML5 a Canvas API, proto byla zvolena pro implementaci vizualizace. Výsledkem vývoje není jen samotný vizualizační nástroj začleněný do EEG/ERP Portálu, ale také knihovna umožňující načítání signálu z datových souborů.

Výsledné řešení není vázané na jakékoliv pluginy a zároveň je cross-browser kompatibilní. Jsou zde také možnosti dalšího vylepšení - např. zobrazení markerů, vykreslování více signálů najednou pro porovnání. Nejužitečnější funkcí by bylo bezpochyby zoomování výseku signálu pro detailnější náhled. Dalším vylepšením by mohla být realizace plátna ve stylu tzv. skleněné tabule. Postup je takový, že se přes aktuální webovou stránku vyvolá průhledný filtr, který se následně použije jako nová kreslicí plocha pro vizualizaci. Takto lze dosáhnout mnohem větších rozměrů plátna, než když je začleněno do layoutu EEG/ERP Portálu.

Seznam zkratek

- AIR** Adobe Integrated Runtime - prostředí umožňující provoz internetových aplikací i v offline režimu.
- CR** Candidate Recommendation - kandidát na doporučení je verze specifikace stále velmi blízka pracovnímu draftu. Jedná se o druhou úroveň schvalovacího cyklu pro přijetí nového standardu W3C konsorciem.
- EEG** elektroencefalogram - měření mozkové aktivity pomocí elektrod přiložených na lebce.
- ERP** Event-Related Potential - evokovaný potenciál je odezva vyvolaná smyslovým nebo motorickým podnětem.
- FFT** Fast Fourier Transformation - algoritmus pro výpočet diskrétní Fourierovy transformace a její inverze.
- HTML5** HyperText Markup Language - značkovací jazyk pro hypertext. Patří mezi jazyky pro tvorbu webových stránek.
- IETF** Internet Engineering Task Force - organizace zabývající se novými návrhy standardů a zlepšováním prostředí Internetu jako takového.
- JSON** JavaScript Object Notation - platformě nezávislý způsob zápisu dat.
- JSP** JavaServer Pages - technologie tvorby webových stránek v jazyce Java.
- MVC** Model-View-Controller - softwarová architektura dělící aplikaci na datový model, uživatelské rozhraní a řídicí logiku.
- PR** Proposed Recommendation - navrhovaná doporučení znamenají především fázi posledních změn na základě vyjádření komunity. Na této úrovni je dokument specifikace odeslán na W3C Advisory Council k finálnímu schválení.
- RIA** Rich Internet Application - webová aplikace zaměřená především na interaktivitu a formu prezentace obsahu.

SMIL Synchronized Multimedia Integration Language - jazyk z rodiny XML navržený W3C konsorciem speciálně pro vytváření a publikování multimediálního obsahu na webu.

SVG Scalable Vector Graphics - škálovatelná vektorová grafika patří do rodiny značkovacích jazyků. Zároveň je SVG i souborový formát popisující vektorový obraz pomocí XML.

SWF - souborový formát používaný Adobe Flash pro enkapsulaci multimedií, grafiky a ActionScriptu. Soubor je spustitelný v rámci Flash Playeru.

VML Vector Markup Language - zastaralý jazyk na bázi XML, dříve používaný pro reprezentaci vektorové grafiky na webu.

W3C World Wide Web Consortium - mezinárodní konsorcium pro vývoj a standardizaci prostředků WWW.

WHATWG Web Hypertext Application Technology Working Group - organizace založená v roce 2004 spolupracující s tvůrci prohlížečů a dalšími zainteresovanými stranami.

WWW World Wide Web - obecně vžitě označení pro internetovou aplikaci (nejčastěji na bázi protokolu HTTP). Obvykle se dále nerozlišuje podle použité technologie nebo architektury aplikace.

XML Extensible Markup Language - značkovací jazyk vyvinutý a standardizovaný W3C konsorciem.

Reference

- [1] Web Statistics and Trends. *W3schools* [Online] 5, 2012. <http://www.w3schools.com/browsers/browsers_stats.asp>.
- [2] Compatibility tables for support of HTML5, CSS3, SVG and more in desktop and mobile browsers. *Can I Use?* [Online] 5, 2012. <<http://caniuse.com/#feat=canvas>>.
- [3] **Experti komunity jQuery: jQuery Kuchařka programátora.** CPress. 2010. ISBN 978-80-251-3152-7.
- [4] **Lubbers, P.; Albers, P.; Salim, F.: HTML5 Programujeme moderní webové aplikace.** CPress. 2011. ISBN 978-80-251-3539-6.
- [5] **Hall, M.: Java servlety a stránky JSP.** Neocortex spol. s.r.o. 2001. ISBN 80-86330-06-0.
- [6] Maturity Levels of the Recommendation Track *W3C Process Document* [Online] 5, 2012. <<http://www.w3.org/2005/10/Process-20051014/tr.html#q74>>.
- [7] HTML5 <header> Tag *W3schools* [Online] 6, 2012. <http://www.w3schools.com/html5/tag_header.asp>.
- [8] **Hogan, B. P.: HTML5 a CSS3 Výukový kurz webového vývojáře.** CPress. 2011. ISBN 978-80-251-3576-1.
- [9] ExplorerCanvas *ExplorerCanvas at Google Project Hosting* [Online] 4, 2012. <<http://code.google.com/p/explorercanvas/>>.
- [10] **Meyer, E. A.: CSS kompletní průvodce.** Zoner Press. 2007. ISBN 978-8086815-64-0.
- [11] **Goldsteinová, A.; Lazaris, L.; Weylová, E.: HTML5 a CSS3 pro webové designéry.** Zoner Press. 2011. ISBN 978-80-7413-166-0.
- [12] **Kroužek, J.; Domes, M.: CSS Kapesní přehled.** CPress. 2006. ISBN 80-251-0773-6.
- [13] **Adobe Creative Team: Adobe Flash CS3 Oficiální výukový kurz.** CPress. 2008. ISBN 978-80-251-2109-2.
- [14] RIA Statistics *Real World Stats of RIA plugin deployments* [Online] 6, 2012. <<http://riastats.com/>>.
- [15] Brain Vision Analyzer User Manual *Brain Products GmbH* [Online] 6, 2012. <<http://campus.uni-muenster.de/fileadmin/einrichtung/biomag/EEG-Labor/VisionAnalyserManual.pdf>>.

Seznam obrázků

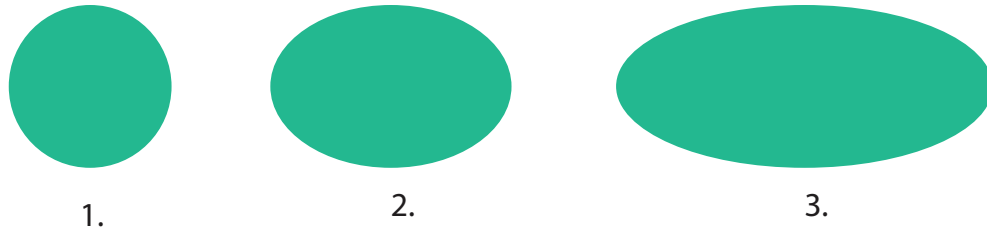
1	Ilustrativní ukázka EEG signálu.	3
2	Ukázka ERP vlny.	4
3	Workflow ve Spring MVC aplikaci.	8
4	Přehled podpory CSS3 animace, celkem 53.91%. [2]	14
5	Přehled podpory SMIL animace, celkem 53.93%. [2]	15
6	Přehled podpory SVG, celkem 65.79%. [2]	17
7	Přehled podpory Canvas API, celkem 75.18%. [2]	17
8	Optional caption for list of figures	18
9	Výsledky benchmarku pro Flash, Canvas a SVG.	19
10	Přehled podpory Canvas API, celkem 75.18%. [2]	24
11	Souřadnicový systém canvas.	26
12	Class diagram knihovny.	30
13	jQuery pop-up okno kontroly nepodporovaných prohlížečů.	32
14	Strom se seznamem kanálů.	33
15	Inicializované plátno s ovládací lištou.	34
16	Příklad animace elipsy: 1-výchozí stav; 2-polovina animačního inter- valu; 3-konečný stav.	41

Seznam tabulek

1	Dostupnost posledních dvou verzí Flash Player pro operační systémy.	12
2	Přehled vendor prefixes pro podporu CSS3.	14
3	Přehled vendor prefixes pro podporu CSS3.	19
4	Nové sekční elementy dostupné v HTML5.	21
5	Jádra JavaScriptu jednotlivých webových prohlížečů.	23
6	Procentuelní zastoupení prohlížečů na trhu v roce 2012.	25

A Porovnání kódu SVG animace pomocí JavaScriptu a SMIL s Canvas API

Na Obrázku 16 je naznačen průběh jednoduché animace elipsy, kdy se v nekonečné smyčce postupně mění délka hlavní osy. Porovnání kódu této animace zapsané pomocí jazyka SMIL a podruhé JavaScriptem je k dispozici níže.



Obrázek 16: Příklad animace elipsy: 1-výchozí stav; 2-polovina animačního intervalu; 3-konečný stav.

SMIL	JavaScript
<pre> <svg xmlns="http://www.w3.org/2000/svg"> <ellipse cx="150" cy="75" rx="10" ry="40" fill="blue"> <animate attributeName="rx" dur="4s" values="10;110;10" repeatCount="indefinite" /> </ellipse> </svg> </pre>	<pre> <svg xmlns="http://www.w3.org/2000/svg" onload="startup(evt)"> <script> <![CDATA[var xmlns="http://www.w3.org/2000/svg" var E; function startup(evt){ E=document.getElementById("E"); oscillate(10,110,10,1); } function oscillate(thin,wide,curx,dir){ E.setAttributeNS(null,"rx",curx); curx+=dir; if (curx>wide curx<thin) { s="oscillate(" window.setTimeout(s,10); dir=-dir; } } </script> <ellipse id="E" cx="150" cy="75" rx="10" ry="40" fill="blue" /> </svg> </pre>

HTML5 a Canvas API
<pre><script type="text/javascript"> window.onload = function () { var cnv=document.getElementById("draw"); var context = cnv.getContext("2D"); oscillate(10,110,1,context); } function oscillate(thin, wide, dir, ctx){ ctx.fillStyle = "rgb(200,0,0)"; ctx.arc(110,110,thin,0,Math.PI*2,true); var act = thin + dir; if (act<wide act>thin) { setTimeout(draw(act, wide, dir, ctx), 50); } else { dir=-dir; setTimeout(draw(act, wide, dir, ctx), 50); } } </script> <canvas id="draw"></canvas></pre>

B Ukázka hlavičkového souboru VHDR

Brain Vision Data Exchange Header File Version 1.0
; Data created by the Vision Recorder

[Common Infos]
Codepage=UTF-8
DataFile=bo01.eeg
MarkerFile=bo01.vmrk
DataFormat=BINARY
; Data orientation: MULTIPLEXED=ch1,pt1, ch2,pt1 ...
DataOrientation=MULTIPLEXED
NumberOfChannels=21
; Sampling interval in microseconds
SamplingInterval=1000

[Binary Infos]
BinaryFormat=INT_16

[Channel Infos]
; Each entry: Ch<Channel number>=<Name>,<Reference channel name>
,
; <Resolution in "Unit">,<Unit>, Future extensions..
; Fields are delimited by commas, some fields might be omitted (
empty).
; Commas in channel names are coded as "\1".
Ch1=Fp1,,0.1,uV
Ch2=Fp2,,0.1,uV
Ch3=F3,,0.1,uV
Ch4=F4,,0.1,uV
Ch5=C3,,0.1,uV
Ch6=C4,,0.1,uV
Ch7=P3,,0.1,uV
Ch8=P4,,0.1,uV

[Comment]

A m p l i f i e r S e t u p

Number of channels: 21
Sampling Rate [Hz]: 1000
Sampling Interval [uS]: 1000

Channels

#	Name	Phys. Chn.	Resolution / Unit	Low Cutoff [s]
	High Cutoff [Hz]	Notch [Hz]		
1	Fp1	1	0.1 uV	10
		1000	Off	

2	Fp2	2	0.1 uV	10
		1000	Off	
3	F3	3	0.1 uV	10
		1000	Off	
4	F4	4	0.1 uV	10
		1000	Off	
5	C3	5	0.1 uV	10
		1000	Off	
6	C4	6	0.1 uV	10
		1000	Off	
7	P3	7	0.1 uV	10
		1000	Off	
8	P4	8	0.1 uV	10
		1000	Off	

S o f t w a r e F i l t e r s

#	Low Cutoff [s]	High Cutoff [Hz]	Notch [Hz]
1	Off	45	Off
2	Off	45	Off
3	Off	45	Off
4	Off	45	Off

Impedance [kOhm] at 10:25:31 :

Fp1: Out of Range!
 Fp2: Out of Range!
 F3: Out of Range!
 F4: Out of Range!
 C3: Out of Range!
 C4: Out of Range!
 P3: 1
 P4: 2

C Ukázka souboru s markery VMRK

Brain Vision Data Exchange Marker File , Version 1.0

[Common Infos]

Codepage=UTF-8

DataFile=bo01.eeg

[Marker Infos]

; Each entry: Mk<Marker number>=<Type>,<Description>,<Position
in data points> ,

; <Size in data points> , <Channel number (0 = marker is related
to all channels)>

; Fields are delimited by commas, some fields might be omitted (
empty).

; Commas in type or description text are coded as "\1".

Mk1=New Segment,,1,1,0,20080401110334996621

Mk2=Jedna ,J 1,2278,1,0

Mk3=Nula ,N 1,3776,1,0

Mk4=Jedna ,J 1,5275,1,0

Mk5=Nula ,N 1,6777,1,0

Mk6=Jedna ,J 1,8279,1,0

Mk7=Jedna ,J 1,9781,1,0

Mk8=Jedna ,J 1,11283,1,0

Mk9=Jedna ,J 1,12786,1,0

Mk10=Jedna ,J 1,14288,1,0

Mk11=Nula ,N 1,15790,1,0

Mk12=Jedna ,J 1,17292,1,0

Mk13=Jedna ,J 1,18794,1,0

Mk14=Jedna ,J 1,20297,1,0

Mk15=Jedna ,J 1,21799,1,0

Mk16=Jedna ,J 1,23301,1,0