Karolina Zbyrowska
Wydział Informatyki i Zarządzania
Wyższej Szkoły Informatyki w Łodzi

Promotor: dr Grażyna Sobiczewska

PORÓWNANIE TECHNOLOGII HTML5 I FLASH

Streszczenie – Na przestrzeni ostatniego dziesięciolecia obserwowaliśmy rozkwit sieci WWW, do którego przyczyniły się idee takie, jak Web 2.0, Rich Internet Applications (RIA) i Semantyczny Internet. Uważa się, że strony Web 2.0 zmieniły paradygmat interakcji między użytkownikami, a właścicielami serwisu, oddając tworzenie treści w ręce użytkowników. Prezentowany artykuł odnosi się do tej ważnej tematyki.

1. Wprowadzenie

Formę „cyberświata" ukształtowały technologie, takie jak Flash, Ajax, JavaScript, a przede wszystkim język HTML, ponieważ jest osnową każdej strony internetowej. Jego najnowsza odsłona, czyli HTML5, wprowadza wiele innowacji, które w przyszłości mogą zaowocować kolejnymi zmianami w możliwościach i funkcjonalności Internetu. Tim Armstrong, dyrektor generalny AOL, powiedział, że przyszłością Internetu nie są algorytmy, a zawartość i design. Znakomicie wpisuje się w tą ideę HTML5 z nowymi znacznikami semantycznymi, podkreślającymi ważkość treści, CSS3, umożliwiający nadanie zawartości atrakcyjnej szaty graficznej oraz Flash, szeroko stosowany do tworzenia użytecznych i interesujących wizualnie witryn i aplikacji sieciowych.

Wspomniane wyżej technologie są tematem rozważań w niniejszej pracy dyplomowej. Pomiędzy ich zastosowaniem zarysowuje się od niedawna część wspólna, która jest podstawą do ich porównania w niniejszej pracy. W pracy tej próbuję również znaleźć odpowiedzi na pytania, która technologia daje większe możliwości, jakie są między nimi podobieństwa i różnice, czy mogą ze sobą konkurować oraz jaka jest szansa, że w przyszłości jedna wyprze drugą.

2 Technologia HTML

HTML jest językiem unifikującym sieć WWW. Od jego pojawienia się upłynęło 20 lat, podczas których ewoluował i rozwijał się, umożliwiając tworzenie bogatych stron internetowych z wykorzystaniem grafiki, stylów CSS i multimediów. Specyfikacja HTML zawiera zbiór elementów (znaczników, tagów), służących do opisu struktury informacji zawartych na stronie internetowej. Są to zdefiniowane i rozpoznawane przez przeglądarki polecenia w nawiasach ostrokątnych, tworzące strukturę dokumentu, modyfikujące jego wygląd i nadające znaczenie fragmentom tekstu.

Pierwszą dostępną wersją był HTML 2.0, opublikowany przez IETF (Internet Engineering Task Force). Wiele cech tej specyfikacji było następstwem istniejących już implementacji. Później rolę IEFT przejęło konsorcjum W3C, w skład którego obecnie wchodzą przedstawiciele m.in. Adobe, Apple, Microsoft, Google, IBM i Oracle. W 1999 roku został opublikowany HTML 4.1 i wraz z ISO/IEC 15445:2000 obowiązuje do dziś jako rekomendowana wersja standardu HTML.

Pierwszy szkic HTML5 powstał w 2008 roku z połączenia dwóch specyfikacji - Web Forms 2.0 oraz Wep Apps 1.0, stworzonych przez grupę WHATWG. Mark Pilgrim następująco definiuje HTML5: jest to nowa generacja HTML'a, zastępująca HTML 4.01, XHTML 1.0 oraz XHTML 1.1 Dodatkowo ma zastąpić DOM2HTML i częściowo JavaScript. Tym co szczególnie odróżnia go od poprzednich wersji, to nastawienie nie tylko na dokumenty, ale także na aplikacje sieciowe (ang. web applications, API). Twórcy HTML5 przewidują, że specyfikacja zostanie ukończona w 2014 roku, jednak będzie dalej udoskonalana jako „HTML Living Standard".

W HTML5 wprowadzono liczne zmiany syntaktyczne, takie jak uproszczenie pisowni niektórych elementów (m.in. Deklaracji Typu Dokumentu i znacznika meta), a także zawarto serię nowych **znaczników semantycznych**, takich jak m.in. header, footer, navigation, które umożliwią przeglądarkom łatwiejsze parsowanie semantycznej struktury dokumentu Korzyści z semantycznych elementów będą czerpać również wyszukiwarki przy indeksowaniu stron oraz inteligentne czytniki dla niepełnosprawnych poprzez wyselekcjonowanie i odczytywanie tylko istotnych treści, pomijając fragmenty takie, jak menu nawigacyjne, reklamy, paski boczne itd.

Dużą zmianą w stosunku do poprzedniej wersji jest wprowadzenie wielu nowych znaczników, wśród których można wyróżnić: video, audio, device, mark, progress, meter, time, ruby, canvas czy command.

HTML5 to technologia której ramy obejmują również nowe mechanizmy i API, takie jak m.in. Web Storage API, Offline Web

Applications, Web Workers, GeoLocation, MathML, Web Sockets, Microdata, Web Sockets i Messaging API.

3 Technologia Flash

Począwszy od swojego skromnego debiutu jako FutureSplash w 1996 roku, oprogramowanie oraz platforma Flash rozwinęły się w kierunku zaawansowanej technologii umożliwiającej wdrażanie różnego rodzaju materiałów multimedialnych, aplikacji RIA (Rich Internet Applications), sieciowych, *enterprise*, SaaS, multimedialnych kampanii marketingowych, aplikacji socjalnych, konsumenckich, aplikacji wideo, interfejsów użytkownika, aplikacji dla urządzeń mobilnych i telewizyjnych. Narzędzia Adobe, które tworzą ekosystem Flash, to oprogramowanie Flash Professional, Flash Builder, Flash Access, Flash Media Server, Flash Platform Services, Flash Media Playback i Flash Catalyst oraz framework Flex.

Program Flash jest aplikacją hybrydową, ponieważ oprócz narzędzi do tworzenia grafiki wektorowej oraz możliwości przetwarzania grafiki rastrowej, umożliwia tworzenie zaawansowanych animacji, edytowanie treści multimedialnych (np. dźwięku) oraz projektowanie interaktywnych aplikacji i interfejsów dzięki elastycznemu językowi programowania ActionSctipt. Ponadto płynnie współdziała z wieloma innymi technologiami, w tym Adobe ColdFusion, PHP, XML oraz Microsoft .NET.

Adobe Flash Player jest środowiskiem uruchomieniowym dla technologii Flash w przeglądarkach internetowych, służącym do wyświetlania aplikacji i filmów Flash (SWF). Innym środowiskiem wykonawczym technologii Flash jest AIR, które umożliwia tworzenie w językach HTML i JavaScript samodzielnych aplikacji, działających poza przeglądarką. AIR to spójne i elastyczne środowisko projektowe, dostarczające aplikacje na różne urządzenia i platformy, w tym urządzenia przenośne i systemy telewizyjne.

4 Porównanie HTML5 i Flash'a

Technologia HTML5 zapewnia natywne wsparcie dla wielu technik, które dotychczas były realizowane z pomocą pluginów takich jak Adobe Flash. Techniki te to, w szczególności, odtwarzanie plików wideo i audio, wyświetlanie grafiki wektorowej (SVG), a także, przy współpracy skryptów JavaSctipt, rysowanie grafiki 2D, tworzenie animacji, interakcji, aplikacji oraz gier przeglądarkowych. Oczywiście istnieje wiele różnic pomiędzy definicjami oraz funkcjami HTML5 i Flash, jednak uwidacznia

się pomiędzy zastosowaniem tych dwóch technologii pewna część wspólna. Czy HTML5 zastąpi Flash'a w jakimś obszarze? Która technologia oferuje bogatsze możliwości? Z punktu widzenia programisty czy webmastera są to ważne pytania, a odpowiedź na nie może mieć decydujący wpływ na wybór technologii, w jakiej będzie on tworzył.

4.1 Odtwarzanie multimediów

Wyświetlanie plików multimedialnych wideo i audio w Internecie było do tej pory głównie domeną technologii Flash, jednak HTML5 również oferuje taką możliwość poprzez znaczniki audio i video. Ich składnia wygląda następująco:

```
<audio src="muzyka.mp3" type="audio/mpeg" controls>
     Twoja przeglądarka nie obsługuje znacznika
audio.
</audio>
<video src="film.ogg" controls>
     Twoja przeglądarka nie obsługuje znacznika
video.
</video>
```

Porównując właściwości oraz proces wstawiania pliku wideo na stronę internetową za pomocą HTML5 oraz Flash'a, nasuwają się następujące wnioski:

- HTML5 wymaga mniej linii kodu niż Flash oraz kod ten jest łatwiejszy do zapamiętania i mniej skomplikowany. W HTML5 używamy znacznika <video>, a przy osadzaniu pliku SWF w dokumencie HTML znaczników <object> i <embed> lub skryptu JavaScript generującego kod.
- Filmy w HTML5 wyświetlają się bez wsparcia ze strony dodatkowego pluginu, jak ma to miejsce w przypadku plików we Flash'u.
- Domyślny odtwarzacz audio i wideo HTML5 jest wbudowany w przeglądarkę, podczas gdy odtwarzanie audio i wideo na stronie WWW przez Flash Player wymaga stworzenia własnego odtwarzacza w technologii Flash lub pozyskania go z Internetu.
- Aby film był odtwarzany we wszystkich przeglądarkach i urządzeniach kompatybilnych z HTML5 należy przekonwertować go do trzech kontenerów: MP4, Ogg i WebM, natomiast Flash Player wymaga konwersji do jednego formatu spośród MP4, F4V i FLV, by film był odtwarzany we wszystkich przeglądarkach i urządzeniach z zainstalowaną odpowiednią wersją wtyczki Flash Player.

- Obydwie technologie oferują możliwość stylizacji wyglądu odtwarzacza. W przypadku HTML5 można się w tym celu posłużyć stylami CSS oraz JavaScript'em.
- W obu technologiach możliwe jest nałożenie filtrów i efektów na film wideo. Element canvas można stylizować przez CSS, osadzić w elemencie canvas, a także połączyć ze znacznikiem SVG, co zostało ukazane w przykładach.
- Flash Player obsługuje progresywne i strumieniowe (w tym dynamiczne i DVR) przesyłanie danych przez Adobe Flash Media Server, natomiast HTML5 video wykorzystuje metodę progresywnego przesyłania danych za pośrednictwem protokołu HTTP. Firma Apple wprowadziła w swojej przeglądarce Safari mechanizm strumieniowania HTTP, więc istnieje szansa, że jej śladami podążą inni twórcy przeglądarek, lecz dopóki to się nie stanie, znacznik video nie jest równie funkcjonalny pod względem przesyłania danych jak Flash Player.
- Flash Player posiada funkcję pełnego ekranu, natomiast w specyfikacji HTML5 nie zdefiniowano takiej możliwości. Jednak twórcy przeglądarek zaczęli implementować tą opcję, co oznacza, że HTML5 w przyszłości prawdopodobnie nie będzie ustępował odtwarzaczowi Flash Player w tym aspekcie.
- Ochrona zawartości wideo we Flashu jest możliwa za pomocą Adobe Flash Media Server i np. metody Flash Access DRM, natomiast HTML5 nie zawiera ustandaryzowanego sposobu ochrony treści. Oznacza to, że HTML5 nie zabezpiecza filmu przed pobraniem na dysk przez polecenie „zapisz jako" czy zrzutem ekranu przez „print screen" i moim zdaniem konieczne jest wprowadzenie metody autentyfikacji użytkownika lub szyfrowania treści.
- Napisy, dubbing i adnotacje mogą być nałożone na filmy wyświetlane we Flashu jak i w HTML5. Program Adobe Flash oferuje do obsługi napisów komponenty z których mogą korzystać programiści ActionScript, natomiast HTML5 umożliwia dodanie napisów poprzez znacznik track.
- W przeciwieństwie do HTML5, Flash oferuje bogaty ekosystem aplikacji umożliwiających dołączanie reklam do filmów wideo wraz z mechanizmami analitycznymi dla reklamodawców. Jest to istotne z punktu widzenia komercyjnych witryn, które korzystają z reklam wyświetlanych przez Flash Player w celach zarobkowych.
- W obu technologiach możliwe jest przyspieszenie/zwolnienie odtwarzania - we Flash'u za pomocą ActionScript'u, natomiast w HTML'u za pośrednictwem JavaScript'u (poprzez atrybuty PlaybackRate lub defaultPlaybackRate).

Mimo, że element video jest znaczną innowacją w specyfikacji HTML5, to brak mu na razie pewnych niezbędnych właściwości, jakie posiada Flash, szczególnie w obszarze transmisji mediów typu *Premium*. Z drugiej strony znacznik video można zintegrować z innymi elementami strony oraz stylizować przez CSS. Serwisy takie jak YouTube udostępniają możliwość korzystania z HTML5 video, jednak zapewniają odtwarzanie we Flash'u jako alternatywę i najprawdopodobniej to podejście będzie najbardziej popularne w ciągu najbliższych lat. Pewnym jest, że wideo w sieci jest wciąż ewoluującym standardem i wiele z jego nowych funkcji jest w fazie rozwoju lub dopiero się pojawi.

4.2 Grafika i animacje

Jednym z nowych elementów, jakie zawiera specyfikacja HTML5, jest canvas (z ang. „płótno"), dzięki któremu w ramach strony WWW możliwe jest rysowanie dynamicznej grafiki. Element canvas może znaleźć zastosowanie w obszarach takich jak wizualizacja danych (np. wykresy), animowana grafika, aplikacje sieciowe czy gry internetowe. Zaletą elementu canvas, jest możliwość zintegrowania go z innymi elementami strony. Znacznik canvas ma prostą składnię i może posiadać atrybuty takie, jak width, height czy ID:

```
<canvas id="moj_canvas" width="360" height="240">
Twoja przeglądarka nie obsługuje elementu canvas.
</canvas>
```

Canvas oferuje większość narzędzi, jakie można znaleźć w programach do rysowania, takich jak Flash: linie, wypełnienia, gradienty, cienie, kształty, krzywe Beziera. Różnica polega na tym, że do rysowania wykorzystywany jest JavaScript, a nie graficzny interfejs użytkownika. Proste operacje rysujące można przeprowadzić za pomocą zdefiniowanych w JavaScript'cie metod (m.in. fillRect(), strokeRect(), beginPath(), moveTo(), arc(), drawImage()).

Oprócz tego element canvas umożliwia rysowanie i dodawanie m.in.: tekstu, cieni i efektu rozmycia, stylów linii (grubość, zakończenie itd.), transformacji (translacja, rotacja, skalowanie), przezroczystości, gradientów i masek obcinających (ang. clipping paths). API Context 2D umożliwia również manipulację pojedynczymi pikselami poprzez metody createImageData, getImageData i putImageData. Obiekt ImageData oprócz rysowania metodą „piksel po pikselu" może służyć na przykład do tworzenia filtrów graficznych lub matematycznych wizualizacji (np. fraktali).

W mojej opinii każdy projekt, stworzony za pomocą JavaScript'u i canvas, można z powodzeniem wykonać w programie Flash. Bardzo dużą część projektów we Flash'u można również uzyskać wykorzystując canvas, ponieważ technicznie ich możliwości są na chwilę obecną podobne. Obie technologie różnią się jednak pewnymi szczegółami, które należy wziąć pod uwagę przy wyborze platformy, w jakiej chcemy stworzyć projekt.

- Jedna z największych różnic między canvas, a Flash'em istnieje na ich poziomie abstrakcji. Canvas jest jedynie niskopoziomowym API dla grafiki 2D, nie posiadającym obiektów, warstw czy grup. Natomiast Flash posiada różnego rodzaju narzędzia, gotowe komponenty i interfejs ułatwiający tworzenie graficznych elementów aplikacji. W przyszłości zapewne powstaną framework'i rozwiązujące tą słabość canvas.
- JavaScript jak i ActionScript mają wspólne korzenie w ECMAScripcie, dlatego ich składnia wygląda podobnie, co ukazałam w przykładach. W obydwu językach do rysowania można wykorzystać ścieżki oraz metody moveTo i lineTo. Różnica między ActionScript, a JavaScript istnieje m.in. w deklaracji typu zmiennej.
- Większość najnowszych przeglądarek posiada wbudowany inspektor/debugger kodu HTML/JavaScript, z którego można korzystać przy tworzeniu projektu w canvas oraz tworzeniu plików .log dla skryptów JavaScript przez console.log API. Narzędzia takie jak Firebug w Firefoxie, Chrome Developer Tools, Dragonfly w Operze i Internet Explorer Developer Tools dostarczają bogatych informacji i możliwości analizy kodu, niedostępnych dla plików Flash.
- Przy skrypcie JavaScript pomijany jest krok kompilacji, który pojawia się przy tworzeniu projektów we Flash'u. Oznacza to, że po modyfikacji kodu w edytorze tekstu, po przeładowaniu strony w przeglądarce od razu można zobaczyć zmiany.
- Można mieć wiele zastrzeżeń do sposobu, w jaki Flash radzi sobie z kursorami, ponieważ jego zmiana często skutkuje opóźniającym się (ang. lagging) kursorem, co negatywnie wpływa na doznania użytkownika. Używając JavaScript'u do zmiany kursora można korzystać ze standardowego zestawu kursorów i wielu zróżnicowanych opcji.
- Z uwagi na fakt, że canvas jest zwyczajnym elementem HTML, może bardzo łatwo reagować na zdarzenia klawiatury natychmiast po załadowaniu strony. W przypadku Flash'a istnieje konieczność wykorzystania dość skomplikowanego ActionScript'u oraz kliknięcia komponentu „flaszowego", aby aktywować zdarzenia klawiatury.
- Wykorzystywanie różnych czcionek jest znacznie efektywniejsze we Flash'u. Osadzenie czcionki w pliku SWF powoduje, że będzie ona poprawnie wyświetlana na każdym komputerze, natomiast w

przypadku canvas wyświetlanie czcionki zależy od wbudowanych czcionek w systemie.

- Specyfikacja canvas nie zawiera żadnego modułu umożliwiającego automatyczne zawijanie tekstu w ramach pola tekstowego, w przeciwieństwie do Flash'a, co narzuca konieczność pisania dodatkowych skryptów JavaScript.
- Renderowanie fragmentów dokumentu HTML na ekran możliwe jest wyłącznie we Flash'u, aczkolwiek prawdopodobnie w przyszłości 2D Contex również umożliwi renderowanie fragmentu dokumentu prosto na canvas.

Brak GUI sprawia, że canvas nie jest przyjaznym narzędziem dla projektanta, który, mając do wyboru Flash'a i canvas, prawdopodobnie wybierze do rysowania to pierwsze narzędzie. Jednak canvas oferuje znacznie więcej niż rysowanie grafiki rastrowej. Umożliwia rysowanie dynamiczne i kontrolowane skryptem, a więc tworzenie animacji, aplikacji, interaktywnych elementów czy gier, które do tej pory kojarzone były głównie z produktami stworzonymi we Flash'u. Być może w przyszłości pojawią się aplikacje pośredniczące przy tworzeniu zawartości canvas przez graficzny interfejs użytkownika. Natomiast istnieje już kilka framework'ów przyspieszających tworzenie aplikacji w canvas, np. jcotton, CAKE, Primer czy jCanvasScript.

4.3 Animacje we Flash'u i HTML5

Canvas, w przeciwieństwie do Flash'a, nie został zaprojektowany do tworzenia animacji, z czego wynikają pewne jego ograniczenia. Prawdopodobnie największym jest to, że po narysowaniu kształt nie może zostać zmodyfikowany. Aby go przesunąć trzeba narysować go od nowa wraz ze wszystkimi innymi elementami rysunku. Rysowanie od nowa skomplikowanych klatek może zabrać dużo czasu, a ponadto wydajność odtwarzania takiej animacji zależy głównie od prędkości komputera, na którym jest wyświetlana. Natomiast Flash daje możliwość łatwego modyfikowania obiektu w kolejnych klatkach, co m.in. przyczyniło się do jego sukcesu na rynku.

Tworzenie animacji na stronach internetowych za pomocą JavaScript'u było rozwiązaniem szeroko stosowanym również przed pojawieniem się HTML5 i elementu canvas. Opiera się głównie na programowaniu zorientowanym obiektowo, osadzonym w technologiach JavaScript/DHTML. Idea stojąca za animacjami generowanymi poprzez JavaScript jest stosunkowo prosta: elementy drzewa DOM (<img>, lub inne) są przemieszczane w ramach obszaru strony internetowej według jakiegoś wzorca zdeterminowanego równaniem logicznym lub funkcją. Dla ułatwienia tworzenia animowanych efektów na stronach napisano wiele framework'ów i bibliotek JavaScript, takich

jak jsAnim, scripty2, GX, Animator.js, Scriptio, z których część jest oparta o biblioteki jQuery czy MooTools.

Inną metodą na tworzenie prostych animacji na stronach HTML jest wykorzystanie kaskadowych arkuszy stylów CSS3, które dają możliwość przypisywania każdemu elementowi na stronie efektów wizualnych takich, jak animowane przejścia (ang. transitions), przekształcenia 2D i 3D (ang. transforms) i animacje (ang. animations). Starsze przeglądarki, które nie obsługują nowych efektów CSS3 mimo, że nie będą wyświetlać animacji, będą poprawnie wyświetlać elementy strony w sposób statyczny.

Animacje w CSS3 mogą posłużyć przede wszystkim do tworzenia animowanych elementów nawigacyjnych, bannerów reklamowych itd. Nie oznacza to, że za pomocą CSS3 nie można wykonać bardziej skomplikowanych animacji, tym bardziej, że w CSS3 znajdują się właściwości, takie jak gradienty, przezroczystość, odbicia, maski, cienie, własne czcionki czy zaokrąglone krawędzie, a dzięki parametrom typu odtwarzanie z opóźnieniem i długość animacji można uzyskać większą kontrolę nad wyświetlaną animacją.

Adobe Flash oferuje kilka sposobów na tworzenie animacji, w tym klatka po klatce, poprzez interpolację klatek pośrednich na osi czasu (classic, motion i shape tween), ułożenia kinematyki odwrotnej oraz za pośrednictwem ActionScript'u.

Zarówno Flash jak i CSS3 posiadają wbudowane techniki do modyfikacji matematycznej krzywej, na podstawie której obliczane są właściwości obiektu między klatkami kluczowymi. Do technik tych należą funkcje ease-in, ease-out i ease-in-out. *Easing* pozwala na dopasowanie np. prędkości poruszania się obiektu, tak aby jego ruch nie był jednostajny, lecz wydawał się bardziej naturalny i zgodny z prawami fizyki, np. zwalniał, przyspieszał lub zawierał kombinację tych efektów.

Po przeanalizowaniu cech jednej jak i drugiej technologii, można wysunąć wniosek, że CSS3 i canvas/JavaScript są użyteczne wszędzie tam, gdzie chcemy, by animacja dotyczyła elementu strony (np. tła) czy była z nim zintegrowana. Natomiast przy tworzeniu animacji pozaprzeglądarkowej lub odtwarzanej na pełen ekran, jak np. film animowany, lepiej sprawdzi się Flash, ze względu na bogactwo narzędzi oraz graficzny interfejs użytkownika, które ułatwiają pracę projektanta.

4.4 Interakcje

Głównym pośrednikiem w interakcjach na stronach HTML był dotychczas i jest nadal JavaScript, odpowiedzialny za warstwę zachowań elementów strony. Nowe moduły znajdujące się w specyfikacji HTML5, dalej angażują JavaScript, tak jak opisany w poprzednim rozdziale canvas. Najnowsze wersje popularnych

przeglądarek nie tylko obsługują nowe znaczniki HTML5 i API, lecz zawierają udoskonalane przez lata silniki JavaScript/ECMAScript.

W specyfikacji HTML5 wprowadzono serię API umożliwiających większą interakcję między stroną internetową, a użytkownikiem. Moduły te to w szczególności drag and drop, Undo History Manager, atrybut hidden, contenteditable i spellcheck. Techniki te były osiągalne już wcześniej przez zastosowanie innych technologii, takich jak Flash, Microsoft .NET czy Ajax, jednak wprowadzenie ich do specyfikacji redukuje lub ogranicza złożoność skryptów, które należy wykorzystać i umożliwia wykonywanie zadań wymagających wcześniej dodatkowego pluginu.

Interaktywne zdarzenia **drag and drop** („przeciągnij i upuść") od dawna obecne są we Flash'u czy bibliotekach JavaScript, np. JQuery, a w następstwie ich popularności i funkcjonalności wbudowano je również w specyfikację HTML5. Model obsługi drag and drop w HTML5 jest skomplikowany w porównaniu do innych rozwiązań, jednak jego zaletą jest fakt, że korzysta z natywnego API przeglądarki, czyli realizuje główny cel HTML5 standaryzacji pewnego zestawu API. Odpowiednikiem operacji drag jest w ActionScript zdarzenie startDrag, a operacji drop zdarzenie stopDrag. W ActionScript 3 obsługa drag and drop została utrudniona, lecz wciąż jest mniej skomplikowana niż w HTML5. Tak jak w przypadku JavaScript'u, wykorzystuje się *słuchacze* zdarzeń (ang. event listeners).

Nowe API, które początkowo były rozwijane w specyfikacji Web Applications 1.0, przyczynią się do powstawania w HTML5 interaktywnych aplikacji. Wśród najważniejszych można wymienić GeoLocation, Web Sockets, Web Storage, File API, File API: Writer, Clipboard API and Events oraz Media Capture API.

Coraz szerszy dostęp do Internetu w telefonach komórkowych i tabletach przyczynił się do wzbogacenia HTML5 o obsługę ekranów dotykowych. Nowe zdarzenia to m.in. touchstart, touchmove oraz touchend. Również Flash podąża za tym trendem i wyposażony został w klasę Multitouch, która zawiera właściwości umożliwiające obsługę dotyku i gestów.

Rozwój technologii takich jak Flash i Java umożliwił projektowanie bardziej skomplikowanych i współczesnych gier, w tym multiplayer. Dzięki elementowi canvas zaczęły się również pojawiać od niedawna gry w technologii HTML5. O ile gier flashowych nie sposób dziś policzyć, to gier stworzonych w HTML5 jest obecnie stosunkowo niedużo. Z pewnością w przyszłości ich liczba będzie rosnąć wraz ze wzrostem popularności tej technologii i poszerzaniem się wiedzy na jej temat.

Zarówno HTML5, jak i Flash umożliwią tworzenie grafiki trójwymiarowej, a co za tym idzie, gier przeglądarkowych 3D. Technologią, która jest odpowiedzialna za renderowanie grafiki 3D na

canvas jest WebGL – silnik 3D dla JavaScript'u natywnie dostępny w najnowszych wersjach przeglądarek, natomiast we Flash'u jest to Stage3D API, w które zostaną wyposażone kolejne wersje Flash Player. Obydwa rozwiązania oferują akceleracje sprzętową GPU oraz shadery i w kategoriach projektowania cechuje ich wiele podobieństw. WebGL i Stage3D rysują ciekawą perspektywę na przyszłość oraz mogą określić na długi czas standardy grafiki trójwymiarowej dla WWW, sprawiając, że projektanci Flash odejdą od bibliotek Away 3D, Alternativa 3D czy Papervision 3D na rzecz kodowania grafiki przestrzennej w którejś z nowych technologii. Ponieważ HTML5 nie posiada spójnego środowiska deweloperskiego tak jak Flash, tworzenie gier w HTML5/WebGL, które dorównają grom „flaszowym" na pewno będzie trudnym zadaniem, aczkolwiek nie niemożliwym. WebGL, w porównaniu do innych technologii grafiki 3D, oferuje łatwiejsze tworzenie wyświetlaczy przeziernych HUD oraz interfejsów zawierających 2D i 3D dzięki możliwości łączenia znacznika canvas z innymi elementami dokumentu. Istniejąca architektura sieciowa uzupełnia działanie WebGL, np. umożliwiając pobieranie tekstur i modeli z adresów URL. Nowe API HTML5, Web Sockets, umożliwiające asynchroniczną, dwukierunkową komunikację między przeglądarką, a serwerem, pozwoli także na tworzenie gier multiplayer.

Kolejnym krokiem w dziedzinie interakcji może być przeniesienie interaktywnych mediów z przeglądarek w kierunku środowiska pozaprzeglądarkowego. Flash posiada ku temu warunki ze względu na istniejącą już platformę AIR, natomiast HTML jest słabo zintegrowanym środowiskiem do programowania aplikacji.

Istotnym czynnikiem przy wyborze technologii do tworzenia interaktywnych witryn może być aspekt ochrony własności intelektualnej. To co odróżnia HTML5 od Flash'a to jawność kodu. Skrypt JavaScript, CSS oraz struktura dokumentu HTML są łatwo dostępne dla każdego gościa strony internetowej przez narzędzie Inspektor Kodu, ponieważ są to języki wykonywane po stronie klienta (w odróżnieniu do np. PHP czy Perl), a pliki nie są kompilowane. Algorytmy podlegające prawu ochrony własności intelektualnej narażone są na kopiowanie i modyfikowanie do własnych potrzeb przez osoby trzecie. Dla aplikacji o „otwartym źródle" (ang. open source) napisanych w HTML'u nie będzie to stanowić problemu, natomiast może stanowić przeszkodę dla aplikacji komercyjnych. W odróżnieniu od HTML'a, ekstrakcja kodu ActionScript z pliku SWF jest trudna i możliwa tylko za pomocą specjalistycznych programów (w ograniczonym zakresie). Jest to skuteczna forma ochrony kodu źródłowego przed kradzieżą. Zagadnienie to wyłoniło się już dawno i podejmowano próby zablokowania dostępu do źródła czy zaszyfrowania JavaScript'u przez inne skrypty i programy, niektóre dość skuteczne. Jednak mimo to, umieszczając w Internecie stronę czy

aplikację zbudowaną w technologii HTML/CSS/JavaScript, powinniśmy mieć świadomość, że kod może zostać skopiowany i wykorzystany przez osoby trzecie.

4.5 Użyteczność i dostępność

Podczas porównywania HTML5 i Flash'a wyłania się bardzo obszerny problem użyteczności i dostępności tych technologii. W przypadku serwisów internetowych użyteczność (ang. web usability) mierzy się za pomocą intuicyjności, ergonomii, ułatwieniach w korzystaniu z zasobów serwisu i zachowaniach typowych dla konwencji Internetu.

Dostępność (ang. web accessibility) to dziedzina koncentrująca się na zapewnieniu dostępu do strony internetowej dla jak najszerszej grupy odbiorców, w tym osób niepełnosprawnych czy osób korzystających z urządzeń mobilnych.

Z bazującej na pluginie natury Flash'a wyłania się kilka aspektów użyteczności. Ponieważ Flash wymaga instalacji pluginu w przeglądarce, filmy Flash nie są w stanie skorzystać z pewnych wbudowanych w przeglądarkę możliwości, takich jak przycisk wstecz, przeszukiwanie dokumentu czy sygnalizowanie zmianą koloru hiperłącza o tym, że zostało już odwiedzone w przeszłości. Pojawiają się również trudności w integracji treści wyświetlanej przez Flash z treścią strony, np. przy zachodzeniu filmu Flash na obszar rozwijanego menu.

Z punktu widzenia architektury informacji i SEO (ang. Search Engine Optimization) nowe znaczniki semantyczne w HTML5, takie jak nav, header czy aside wpływają dodatnio na dostępność strony i jej optymalizację pod kątem wyszukiwarek. Natomiast wadą serwisów zaprojektowanych we Flash'u jest to, że podczas ich przeszukiwania typowe roboty napotykają na problemy, co m.in. jest przyczyną tego, że Flash nie jest rekomendowanym narzędziem do prezentacji treści w Internecie. Od 2008 roku wyszukiwarka Google może indeksować tekst umieszczony w pliku SWF oraz wykrywać adresy URL w pliku SWF, jednak istnieje wiele doniesień o tym, że indeksowanie to nie jest wystarczająco efektywne. Przy tworzeniu projektów we Flash'u rekomendowane jest utworzenie alternatywnej treści HTML, która może być indeksowana przez wyszukiwarki.

Istotna przy projektowaniu dostępnej witryny jest jej kompatybilność z różnymi przeglądarkami. Kompatybilność serwisu stworzonego we Flash'u jest bardzo wysoka: w każdej przeglądarce z zainstalowaną wtyczką Flash Player serwis wyświetlany jest tak samo, co oszczędza czas i pracę programisty. W przypadku HTML webmasterzy od dawna mierzą się z problemem dostosowania strony internetowej do każdej przeglądarki, tak by była wyświetlana identycznie i bez błędów, co zmusza ich np. do tworzenia kilku alternatywnych stylów CSS, osobnych

dla każdej przeglądarki czy opracowywania tzw. *hack'ów* CSS. Wraz ze wprowadzeniem nowych elementów w HTML5 mogą pojawić się nowe problemy przy ich stosowaniu, ponieważ ich implementacje mogą się różnić między przeglądarkami.

Wyniki badań nad obecnością Adobe Flash Player na rynku pokazują, że technologia Flash jest obecnie znacznie bardziej dostępna niż HTML5. Plugin Flash Player 10, według badań przeprowadzonych w grudniu 2010 r. dla firmy Adobe, zainstalowany był na ok. 99% komputerów podłączonych do Internetu. Na podstawie statystyk firmy NetMarketShare z marca 2011 roku dostęp do technologii HTML5 w marcu 2011 miało ok. 46% użytkowników Internetu. Badanie to dotyczyło dziesięciu najpopularniejszych przeglądarek. Trzy z czterech wiodących pod względem popularności przeglądarek były starszymi wersjami Internet Explorera, łącznie dając sumę 54% zainstalowanych na komputerach przeglądarek, które nie są kompatybilne z HTML5.

Podsumowanie

HTML5 to język hipertekstu, natomiast Flash jest kompletnym środowiskiem, nierozerwalnie związanym z językiem ActionScript i posiadającym graficzny interfejs użytkownika. Tytuł pracy, „porównanie technologii HTML5 i Flash" może nasuwać skojarzenia, że sam język HTML oferuje funkcje porównywalne z Flash'em. Nie jest to jednak prawdą. Faktycznie, w przypadku elementu video i audio nie jest wymagany dodatkowy kod JavaScript czy CSS, ponadto HTML umożliwia rysowanie grafiki wektorowej za pomocą SVG, więc pod tymi względami istnieje podobieństwo do Flasha'. Natomiast przy wykorzystaniu np. elementu canvas do rysowania niezbędny jest język JavaScript. Flash bez języka ActionScript również byłby ubogi w możliwości - bez niego nie powstałyby gry, aplikacje czy nawet bardziej zaawansowane animacje. CSS oraz JavaScript mogą być potraktowane jako część *rodziny* HTML5. Znamiennym jest, że warstwa prezentacji celowo została wydzielona ze specyfikacji HTML przez jej twórców do osobnej specyfikacji CSS.

Z analizy dokonanej w poprzednich podrozdziałach wynika, że HTML5 z pomocą JavaScript i CSS może konkurować z Flash'em w wielu dziedzinach. Warto podkreślić, że możliwości, które oferuje HTML5 są dostępne natywnie w każdej przeglądarce, która ma zaimplementowane wsparcie dla danej części specyfikacji. Nie jest wymagana instalacja pluginu, jak ma to miejsce w przypadku Flash'a.

Obydwie technologie oferują wyjątkowe cechy i funkcjonalność oraz obydwie są właściwe w zależności od celów przedsiębiorstwa lub czynności które chcemy udostępnić użytkownikowi [66]. Przykładowo,

jeśli któreś z poniższych stwierdzeń są prawdziwe, zaleca się korzystanie z HTML5:

- chcemy, by wideo było wyświetlane w systemach *low-endowych*,
- chcemy zmniejszyć koszty (nie kupować licencji Flash),
- chcemy, by wideo/aplikacja działała na iPadzie, iPhonie i innych platformach Apple,
- chcemy pracować w środowisku „open development".
- Natomiast Flash powinien być wykorzystywany, jeśli:
- produkt musi być wyświetlany na wszystkich przeglądarkach, w tym starych modelach Internet Explorer,
- potrzebne są zaawansowane opcje wideo, np. streaming czy relacje na żywo,
- chcemy ochronić zawartość dokumentu,
- chcemy dynamicznie wmontować reklamy do filmu wideo.

Według Dominique Jodoin, dyrektora Bluestreak Technology, nie ulega wątpliwości, że rozwój HTML5 jako adoptowanego standardu przebiega bardzo dynamicznie, jednak nie oznacza to bynajmniej wymierania technologii Flash. HTML5 nie powinien być traktowany jako standard, który zastąpi Flash'a, a raczej jako jego konkurencja [66]. Flash jest wielofunkcyjnym narzędziem o bardzo wysokiej dostępności, z możliwościami daleko wykraczającymi poza sieć WWW i na tym polu jego zastosowanie jest nieocenione. Przedstawia dużą wartość jako platforma do tworzenia aplikacji stanowiących liczną bazę zainstalowaną w środowiskach korporacyjnych, gdzie nie istnieją istotne przesłanki do przepisywania działających aplikacji do HTML5.

Wnioski sformułowane w pracy mogą pomóc projektantom i webmasterom w wyborze technologii odpowiadającej ich potrzebom i celom tworzonego projektu. Szczegółowa analiza HTML5 i Flash'a ukazała ich zalety, jak również słabe punkty i luki. Szczególnie w przypadku HTML5, który jest otwartym standardem, silnie współpracującym z JavaScript'em, luki te mogą być z powodzeniem wypełnione przez stworzenie przez programistów odpowiedniego oprogramowania.

Przez kilkanaście lat Flash wypełniał zadania, którym HTML nie mógł sprostać i przyczynił się do przemiany pierwotnej sieci WWW w tzw. Web 2.0. Teraz, gdy HTML5 jest wyposażony w funkcje, które do tej pory oferowały wyłącznie technologie oparte na pluginie, sieć WWW jest gotowa na dokonanie kolejnego zwrotu. Faktem jest, że HTML5 jest technologią konkurencyjną dla Flash'a, z której można korzystać już dziś. Dzięki temu projektanci mają do wyboru większą liczbę technologii, a w wyborze tym mogą kierować się własnymi potrzebami i możliwościami.

Literatura

[1] Keith J., HTML5 For Web Designers, 2010, A Book Apart

[2] Lubbers P., Albers B., Salim F., Pro HTML5 Programming. Powerful APIs for Richer Internet Application Development, 2010, Apress

[3] David M., HTML5, Designing Rich Internet Applications, 2010, Elsevier

[4] Hawkes R., Foundation HTML5 Canvas: For Games and Entertainment, 2011, Apress

[5] Pilgrim M., HTML5 Up and Running, 2010, O'Reilly Media

[6] Dowd S., Reinhardt R., Adobe Flash CS4 Professional Bible, 2009, Wiley Publishing Inc.

[7] http://www.w3.org

[8] http://www.whatwg.org

[9] http://wikipedia.pl

[10] http://www.adobe.com

[11] http://my.opera.com/core/blog/2010/03/03/everything-you-need-to-know-about-html5-video-and-audio-2

[12] http://www.netmarketshare.com

[13] http://dev.opera.com/articles/view/custom-html5-video-player-with-css3-and-jquery

[14] http://www.google.com/support/webmasters/bin/answer.py?answer=72746#1

[15] http://www.ninebyblue.com/blog/search-friendly-flash

[16] https://developer.mozilla.org/en/Canvas_tutorial

[17] http://www.schillmania.com/content/projects/javascript-animation-1

[18] http://css3.bradshawenterprises.com

[19] http://www.jasob.com

[20] http://venturebeat.com/2010/12/17/html5-vs-flash-how-will-the-battle-play-out-in-2011

[21] http://www.waxpraxis.org/blog/2010/10/html5-impedance-mismatch

COMPARISON HTML5 AND FLASH TECHNOLOGIES

Summary – Over the last decade have seen flourishing on the Web, which have contributed ideas, such as Web 2.0, Rich Internet Applications (RIA) and the Semantic Internet. It is believed that Web 2.0 has changed the paradigm of interaction between users and owners of the site, placing content creation in the hands of users. The article refers to this important subject.