

WEB 2.0 - PŘÍKLADY, TECHNOLOGIE A PROBLÉMY

Tomáš Pitner

Masarykova univerzita, Fakulta informatiky, tomp@fi.muni.cz

ABSTRAKT:

Web 2.0 je souhrnné označení vysoce interaktivních, snadno použitelných služeb současného webu (Google, Yahoo, Flickr a další). Příspěvek ukazuje příklady Web 2.0 služeb, postupy jejich realizace a modely použití.

KLÍČOVÁ SLOVA:

Web 2.0, webové služby, integrace webových služeb, REST, AJAX

CO JE WEB 2.0

V posledních letech dosáhlo jak rozšíření a dostupnost samotných technologií, tak především otevřený způsob jejich použití natolik široké úrovně, že se začíná hovořit o nové generaci webu, o Webu 2.0, [O'Reilly, 2005].

Není snadné Web 2.0 přesně definovat – některé významné osobnosti dokonce mají za to, že tento pojem vlastně nic neznamená [Clarke]. Přesto intuice a poslední zkušenosti ukazují, že změny v přístupech k použití webu se opravdu natolik mění, že si nové označení pravděpodobně zaslouží. Pojďme uvést několik charakteristik služeb, jež podobu moderního webu utvářejí.

Tim O'Reilly [O'Reilly, 2006] dále uvádí seznam několika vlastností, charakterizujících služby pro *Web 2.0*:

- jde o *snadno rozšiřitelné* (a *integrovatelné*) služby
- spravují *unikátní*, obtížně obnovitelné *datové zdroje*, jejichž hodnota stoupá, pokud jsou využívány (tj. nelze je snadno imitovat, např. služby Google s jejich extrémně rozsáhlou datovou bází)
- důvěřují uživatelům jako *spolutvůrcům* (tzv. read/write web, např. veškeré wiki projekty vč. Wikipedie) a využívají „*kolektivní inteligenci*“ (systémy shromažďující odkazy, ohodnocující a značující je)
- pro své šíření využívají (a motivují) vlastní uživatele (mají tzv. síťový efekt – mj. veškeré peer-to-peer sítě, vč. sítí komunikačních – jako je Skype)
- jsou využitelné na *různých zařízeních* (projevuje stále větší role zařízení mobilních)
- mají *jednoduchá uživatelská i aplikační rozhraní* (na rozdíl od soudobých komerčních multimediálních webů)
- nejsou nikdy hotové, existují trvale jako „*betaverze*“ (příkladem budiž i řada služeb Google – např. Calendar)

Výše uvedené vlastnosti velmi dobře charakterizují projekty a produkty, které dnes na Internetu patří k nejuspěšnějším. Jmenujme například *Google* (s celou řadou svých služeb – *Gmail*¹, *Google Maps*², *Google Calendar*³ a další), stejně jako známou *Wikipedii*, jakož i další

¹ <http://www.gmail.com>

² <http://maps.google.com>

³ <http://www.google.com/calendar>

wiki. Specifickou skupinou služeb Web 2.0 jsou *weblogy* (blogy), které lze na mnoha serverech zdarma zřídít a používat jako internetové „deníčky“ na prezentaci vlastních názorů, postojů či na reflexi aktuálního dění.

Sítě *rovnocenných uživatelů* (peer-to-peer networks) jsou další charakteristickou skupinou služeb Web 2.0, uveďme např. *BitTorrent* nebo *DirectConnect*.

WEB 2.0 Z POHLEDU VÝVOJÁŘE – TECHNOLOGICKÁ BÁZE

Demokratizace uživatelského přístupu k webu se odráží i v možnostech využívat služby Web 2.0 *programově*. Tomu nahrává jednoduchá technologická báze služeb Web 2.0, které jsou typicky:

- postaveny na jen základních internetových (nejčastěji *webových*) *protokolech* (HTTP, případně protokolech nižší úrovně – UDP);
- využívají zavedených jednoduchých *výměnných datových formátů* (jazyky založené na XML, formát JSON⁴ – JavaScript Object Notation);
- nabízejí obvykle jen úzkou funkcionalitu – je ale možné dosáhnout komplexnější funkcionality *kombinací (integrací) služeb*.

Prostým až minimalistickým technologickým základem jde Web 2.0 poněkud proti proudu současného vývoje podnikových IT, kde jsou nasazovány stále složitější, mnohvrstvé architektury, postavené na komplikovaných protokolech (např. *webové služby*). Jak tedy k vývoji Web 2.0 aplikací přistupovat? Předně jde o volbu *stylu softwarové architektury* (*Architectural Style*).

ARCHITEKTURNÍ STYL REST

Styl architektury je představován sadou zvolených charakteristik či omezení (constraints), předepisujících, z jakých komponent bude *webových* systém složen, jaké role budou komponenty plnit a jak budou komunikovat.

Předmětem dnes již klasického díla – disertace Roy Fieldinga *Architectural Styles and the Design of Network-based Software Architectures*, [Fielding] – je klasifikace architekturních stylů a návrh takového moderního stylu, který by a) stavěl na původních zásadách budování webu a b) identifikoval pozitivní stránky jednotlivých přístupů a dále je optimalizoval. Výsledkem je návrh architekturního stylu REST (Representational State Transfer), který se k charakteru služeb Web 2.0 velmi dobře hodí. Základními pojmy a stavebními kameny REST jsou:

- **zdroje** (*Resources*) – z konceptuální úrovně považujeme za zdroje vše, co je dostupné na internetu a jednoznačně identifikovatelné svým URI (Uniform Resource Identifier). Zdrojem může být textový (HTML) soubor, obrázek, stejně jako abstraktním z databáze generovaný zdroj (např. popis uživatele).
- K obsah zdrojům přistupujeme pomocí jeho vhodné konkrétní **reprezentace** (*Representation*), jíž mohou být textová nebo XML data, stejně jako data v obrazovém formátu (PNG, JPEG).
- Komunikace mezi komponentami znamená čtení, vytváření, modifikaci nebo rušení zdroje, přičemž jsou využívány běžné HTTP metody (GET, POST, PUT, DELETE).

Pro vývojáře již existují kompletní rámce pro práci na službách typu REST, např. pro Javu je to mj. rámec *Restlet*⁵.

⁴ <http://www.json.org/>

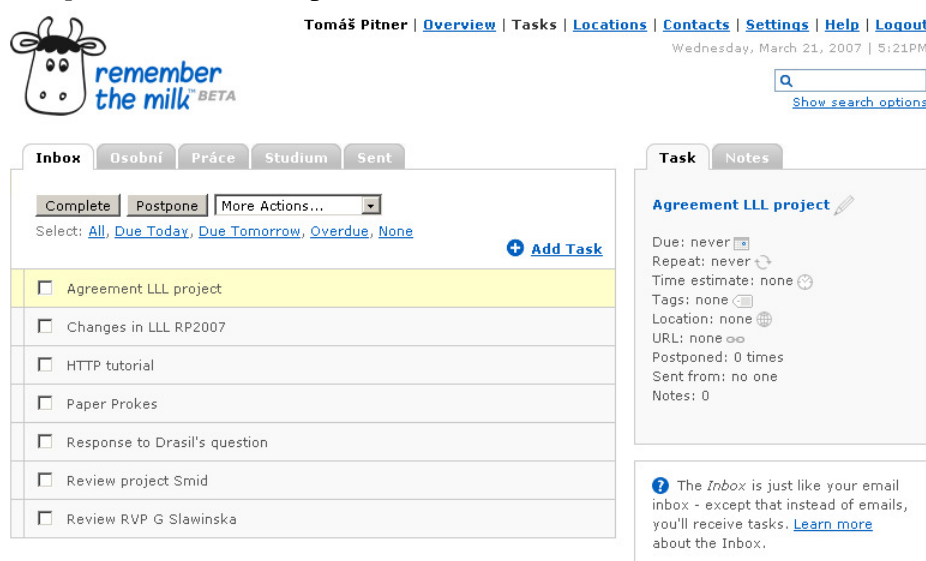
⁵ <http://www.restlet.org>

RICH INTERNET APPLICATIONS - FENOMÉN WEBU 2.0

Tím, jak webové (Web 2.0) aplikace pronikají do dalších oblastí IT, průvodním jevem je stírání rozdílů mezi aplikacemi instalovanými a běžícími na klientském počítači (desktopovými) a webovými. Hovoří se o tzv. *Rich Internet Applications* (RIA), jejichž hezký přehled lze najít v [Coach]. Téměř žádná plnohodnotná webová aplikace se neobejde bez silné serverové části, která realizuje většinu aplikační logiky a stará se o persistentní ukládání dat. Na straně klienta (prohlížeče) bývá situace různá.

I pokud jde o služby typu Web 2.0, některé si vystačí jen s prostým HTML, jiné naopak využívají specializovaných běhových prostředí, jakým je např. Java WebStart⁶, umožňující spouštění de facto plnohodnotných javových aplikací – ovšem bez nutnosti komplikované instalace, údržby a aktualizací.

Základní platformou vývoje klientské strany aplikací však zůstává webový prohlížeč a jeho možnosti, mezi něž patří hlavně skriptování *JavaScriptem*, dnes již plně standardizovaným jako *ECMA Script*, viz [ECMAScript].



Obr. 1 Remember The Milk - příklad vysoce interaktivní webové aplikace

Aby aplikace webová vykazovala stejnou funkcionalitu i stejný pocit z práce jako aplikace desktopová, musí nést podobné charakteristiky:

- být *vysoce interaktivní* (klikání tlačítek a čekání na přegenerování a znovunačtení stránky je neakceptovatelné);
- mít ovládací filozofii a prvky uživatelům důvěrně známé z aplikací desktopových (menu, lišty nástrojů, rozvržení s panely, komponenty WYSIWYG editorů; velmi podstatná je práce stylem drag&drop);
- interakce se serverem by měla probíhat asynchronně, transparentně, na pozadí.

AJAX

Poslední požadavek lze dobře splnit využitím schopností programu v JavaScriptu asynchronně, i ve více vláknech, reagovat na uživatelské události (klikání na odkazy a tlačítka, výběru ze seznamů, vyplnění vstupního pole, přetažení komponenty myši...). Reakcí na události bývá většinou nějaké jednoduché zpracování na straně klienta a odeslání dat serveru – obvykle v podobě XML nebo JSON protokolem HTTP/HTTPS. Po splnění požadavku serverem se klientovi vrátí výsledek ve stejné nebo jiné formální podobě – a skript

⁶ <http://java.sun.com/products/javawebstart/>

na přijetí výsledku může asynchronně zareagovat a například na pozadí překreslit část uživatelského rozhraní, přidat komponentu nebo aktualizovat obsah. Tyto asynchronní postupy s využitím JavaScriptu (a typicky též XML) se označují jako AJAX (Asynchronous JavaScript And XML, [AJAX]).

SPECIFIKA VÝVOJE PRO WEB 2.0

Web 2.0 neznamena automatickou nutnost poříditi si nové nástroje, totálně změnit metodiky vývoje apod. Technologicky se staví na ověřených principech, protokolech a nástrojích. Přesto jsou zde zřetelné posuny:

- Vývoj je rychlý, vše je permanentní betaverze – uplatní se tak především *agilní metodiky* – také proto, že se vyvíjí s předem nejasným zadáním, které se nadto časem vyvíjí, analýzu nelze provést jednorázově.
- Vývoj je dosud převážně rozštěpený na serverovou a klientskou část. Serverové vrstvy jsou dnes již dobře pokryté nejrůznějšími webovými rámci, vývoj pro ně se děje pohodlně z integrovaných prostředí.
- Značnou část vývoje u typických Web 2.0 aplikací bohužel tvoří vývoj pro klientskou stranu představovanou prohlížečem. Využívají se knihovny překrývající platformové odlišnosti, ale nové verze prohlížečů stejně znamenají nutnost postupovat ad hoc podle situace.
- Na udržovatelnost kódu má velmi negativní vliv nutnost odděleného vývoje klientské a serverové části – změny je třeba propagovat ručně.

VÝVOJOVÉ NÁSTROJE

V poslední době jsme však svědky počínů, které situaci mění. Google zdarma nabízí *Google Web Toolkit* [GWT], což je soubor běhových knihoven a vývojových nástrojů umožňujících z jednoho zdroje vygenerovat a zkompileovat jak serverovou tak klientskou část aplikace, přičemž oboje programujeme v jazyce podobném Javě. Blíže ke GWT viz také [Pichlík].

Přesto se v některých prostředích nevyhneme přímému „nízkoúrovňovému“ použití AJAX, přičemž se neobejdeme se bez hotových *knihoven* (např. *Prototype*⁷) poskytujících základní zázemí. Předně sem patří knihovny zapouzdřující HTTP volání spojené s výměnou XML dat. Dnes již existují standardní postupy, jak překrýt platformové rozdíly a vytvořit objekt typu XMLHttpRequest⁸, který se postará o zaslání požadavku HTTP serveru a příjem XML dat zpět.

Bohužel jazyk těchto knihoven, JavaScript, nedisponuje na rozdíl od většiny soudobých pg. jazyků možností členit kód do *jmenných prostorů* (balíků) – rozsáhlejší program je pak nepřehledný a díky dynamické povaze jazyka (*slabé typovanosti*) i špatně laditelný.

Další důležitou knihovní funkcionalitou jsou nejrůznější *grafické ovládací prvky* (*widgets*) – menu, plovoucí komponenty s možností drag&drop, panely (tabs), vylepšené HTML ovládací prvky (odkazy, seznamy), jakož i celé mocné komponenty typu WYSIWYG editoru. Dobrý přehled dostupných RIA/AJAX komponent je k nahlédnutí v [Domenig].

Obecným problémem vývoje v JavaScriptu zůstává ladění a testování aplikací. Základní pomůckou vývojáře nadále zůstávají poměrně nízkoúrovňové prostředky jakými jsou *Firefox Web Developer Extension*, nejrůznější debuggery JavaScriptu, např. *Venkman*⁹ a samozřejmě obvyklé ladicí a testovací nástroje pro stranu serveru – viz [Pitner, 2006].

⁷ <http://www.prototypejs.org/>

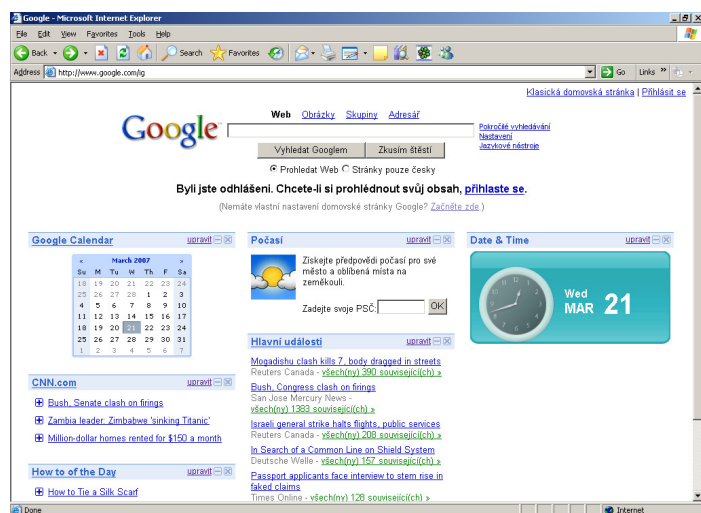
⁸ Specifikace konsorcia W3C, <http://www.w3.org/TR/XMLHttpRequest/>

⁹ Venkman JavaScript Debugger, <http://www.mozilla.org/projects/venkman/>

Nezbytnou dovedností vývojáře pro Web 2.0 je kromě znalosti JavaScriptu také zvládnutí celého procesu zpracování XML dat, byť je tento formát mnohde vytlačován intuitivnější a snadněji zpracovatelnou notací JSON umožňující snazší práci s daty na straně klienta s JavaScriptem, tj. typicky webového prohlížeče.

INTEGRACE SLUŽEB WEB 2.0

Nekomplikované rozhraní, používání jednoduchých výměnných datových formátů založených převážně na XML nebo JSON, přímo vybízí ke znovupoužívání služeb, mnohdy



Obr. 2 Google personalized homepage

v nečekaných kontextech. Znovupoužitím je míněno nejen využití jinou než předpokládanou cílovou skupinou koncových uživatelů, ale také použití služeb aplikacemi, dalšími službami.

Hezkou ukázkou integračního počínu je *Google Personalized Homepage*¹⁰, viz obr. 2, na niž si po zřízení účtu můžeme umístit obsah, jenž nás zajímá, ve formě více tzv. *Google Gadgets*, jakýchsi „okének“ k reálným aplikacím běžícím i mimo servery Google. Podívejme se nyní na nejčastěji integrované typy služeb Web 2.0.

Single Sign-On

Typickou znovupoužitelnou službou (i před Web 2.0) je *autentizace*, případně i *autorizace* přístupu k určitému zdroji prostřednictvím služby třetí strany, obr. 3:



Obr. 3 Proces přihlášení s použitím Google Account Authentication

1. Klient (uživatel) potřebuje přistoupit ke zdroji.
2. Zdroj ale není vybaven vlastní správou uživatelů a přístupových práv, proto deleguje požadavek další službě – např. *Google Account Authentication*, [GAA].
3. Ta zajistí ověření klientovy identity a v případně úspěšného přihlášení je klient přeměrován na požadovaný zdroj. Celý proces je znázorněn na následujícím diagramu.

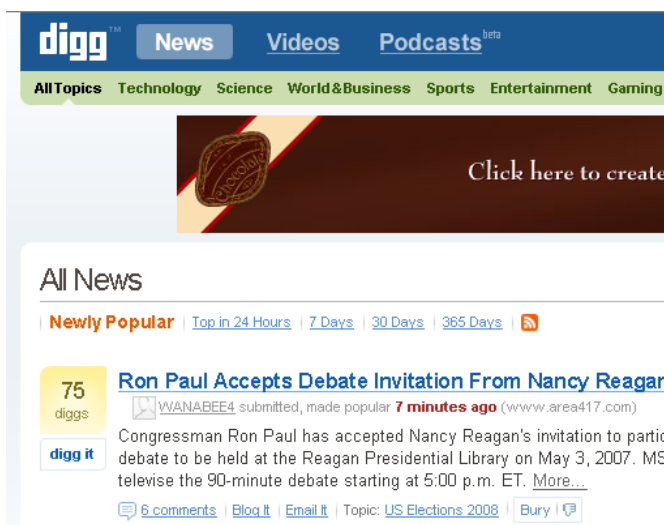
Při optimistickém scénáři tak bude do budoucna globálně existovat několik univerzálních specializovaných autentizačních služeb poskytující servis službám ostatním. Jak to bude s poplatky za používání a s dalšími omezeními při masovém využívání těchto služeb třetími stranami není jasné – tato záležitost se ale týká všech služeb Web 2.0 integrovaných jinam...

¹⁰ Google Personalized Homepage, <http://www.google.com/ig>

Mapové služby

Další, často „ochutnávanou lahůdkou“ pro integraci jsou mapové služby. Mnoho Web 2.0 aplikací přiřazuje zdrojům jejich lokaci – ať už jsou to události v kalendáři Google sdíleném více uživateli nebo záznam ve specializované službě, jakou je např. upcoming.org¹¹.

Využívání služeb je historicky jednou z nejstarších integračních oblastí – např. mapový servis mapy.cz poskytuje možnost identifikovat na mapě polohu určitého objektu (např. sídla určité organizace) a z webu dané organizace se na ně odkazovat.



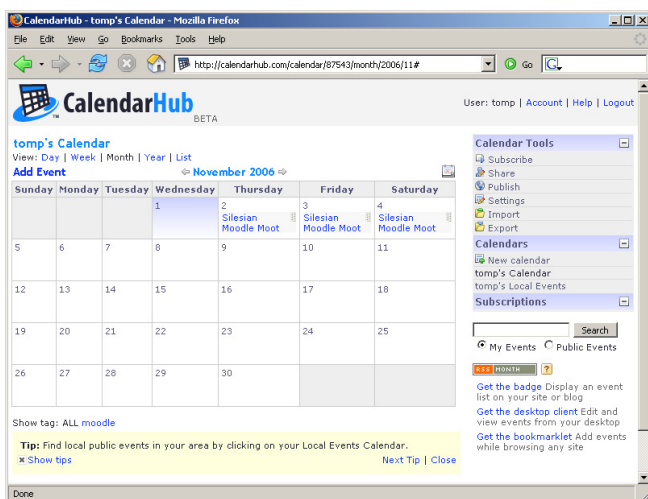
Obr. 4 Odkaz vložený do systému digg.com

Správa odkazů a dokumentů

Mnoho dnes již klasických, prototypových služeb Web 2.0 je určených k ukládání, sdílení a hodnocení odkazů na webové zdroje.

Mezi nejznámější patří del.icio.us¹², méně známý je digg.com (viz obr. 4) nebo furl.net, který ovšem kromě uchování odkazu a jeho sdílení dokáže také zálohovat samotný obsah odkazované stránky v podobě osobní kopie, která by – díky přístupnosti jen pro toho, kdo ji vyrobil, neměla porušovat práva autora/poskytovatele primárního obsahu. Podstatnou přidanou hodnotou je možnost linkované zdroje hodnotit uživatelskou komunitou.

Sdílení multimediálního obsahu



Obr. 5 Kalendář vystavený na CalendarHub.com

¹¹ <http://upcoming.org>

¹² <http://del.icio.us>

¹³ *Direct Connect*, [http://en.wikipedia.org/wiki/Direct_Connect_\(file_sharing\)](http://en.wikipedia.org/wiki/Direct_Connect_(file_sharing))

¹⁴ *Kazza*, <http://www.kazza.com/>

¹⁵ *Gnutella*, <http://www.gnutella.com/>

¹⁶ *BitTorrent*, <http://www.bittorrent.com/>

¹⁷ *Flickr*, <http://flickr.com>

¹⁸ *YouTube*, <http://www.youtube.com>

Kromě notoricky známých – a rozporupně vnímaných – služeb pro peer-to-peer sdílení digitálního obsahu jako jsou systémy *Direct Connect*¹³, *Kazza*¹⁴, *Gnutella*¹⁵ nebo *BitTorrent*¹⁶, existují systémy s úložištěm na straně serveru.

K ukládání statických obrázků (fotografií) slouží *Flickr*¹⁷, pro vystavení videa se používá *YouTube*¹⁸. Vedle těchto specializovaných serverů existuje celá řada univerzálních úložišť. V nejjednodušším případě se tváří jako velmi prostá webová stránka s možností odeslat (upload) soubor na server. Jako odpověď se vrátí poněkud

kryptické URL, pod nímž je soubor nadále persistentně uložen – dokonce není třeba žádná registrace a prostor nemusí být nijak omezen – což většinou není případ služeb jako Flickr, které významnější kapacitu (úložiště i datového přenosu) nabízejí už jen za příplatek.

I tyto služby jsou v současnosti cílem zájmu organizací kolektivního zastupování ve věcech ochrany autorských práv (v současnosti hýbe světem žaloba Viacom proti Google [IHT], do jehož koncernu YouTube patří) a je tedy možné, že extrémně liberální módy použití nebudou mít delší perspektivu – a možná ani samotný bezplatný provoz služeb.

Další spory se vedou v etické rovině. Již byly zaznamenány případy, kdy zveřejnění amatérského videa zachycujícího osobu, která neuklidila veřejné prostranství po vyvenčení psa, vedlo ke zničujícímu útoku na onoho „hříšníka“.

Kalendářové služby

Vděčné a dobře integrovatelné jsou služby kalendářového charakteru, např. CalendarHub, obr. 5. Mnohé – nejen kalendářové služby – nabízí programátorské rozhraní (API) pro přístup z aplikací psaných v běžných programovacích jazycích (např. Java) a zcela běžně je k dispozici přístup prostřednictvím volání webových služeb, ať už klasického, tzn. SOAP-typu, nebo „lehčí“ typu REST.

Služby syndikace obsahu

Syndikace, neboli soustředění a šíření (obvykle nového) obsahu pomocí specializovaných popisovačů – tzv. kanálů postavených na výměnných formátech jako jsou RSS a Atom. Existují webové aplikace i desktopové aplikace, případně rozšiřující moduly do prohlížečů, které dokážou soustředit více kanálů a tak uživateli zpřístupnit na jednom místě nový obsah z mnoha zdrojů.

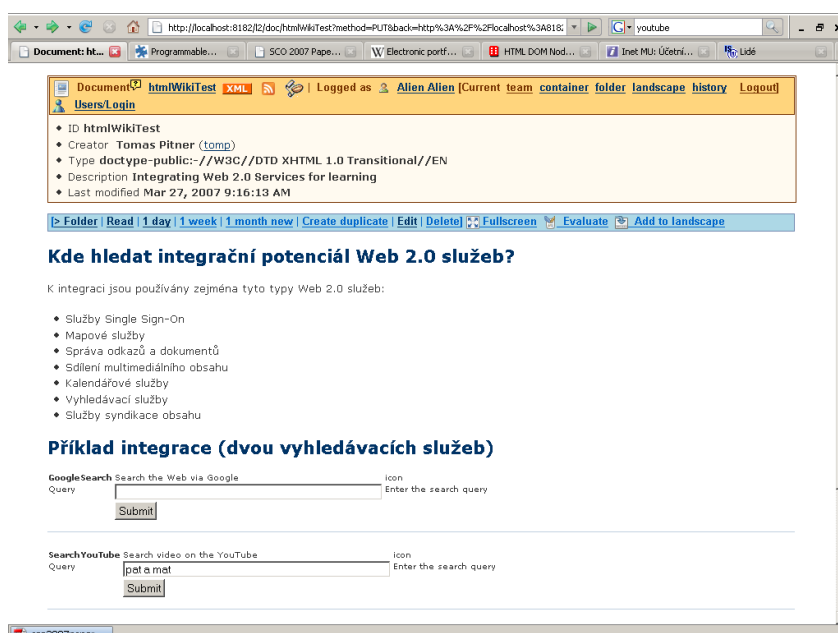
Mashups

Na portálu *ProgrammableWeb*¹⁹ najdeme nesčetnou řadu integrujících Web 2.0 aplikací, které jsou též označovány jako „Mashups“ a především odkazy na popisy API jednotlivých – i primárních – Web 2.0 služeb.

Podobně integračním směrem jdou i některé aktivity v oblasti elektronické podpory výuky, kde je důležité mít možnost rekombinovat dílčí výukové objekty (Learning Objects) a prezentovat je za novými účely a v nových formách.

PŘÍKLAD WEB 2.0 PLATFORMY PRO E-LEARNING

Platforma L2 (viz obr. 6) představuje možnost, jak na jednom místě integrovat nezávislé služby Web 2.0 v kontextu určeném pro výuku. Typickým případem užití takové platformy jsou



Obr. 6 Platforma L2 integrující Web 2.0 služby

¹⁹ <http://www.programmableweb.com/>

malé a větší pracovní (výukové) skupiny, ale i sebevzdělávání a údržba vlastního „e-portfolio“.

ZÁVĚR

Web 2.0 představuje novou generaci „odlehčených“ a dobře integrovatelných služeb, které „rostou“ se svými uživateli. Vývoj pro Web 2.0 však znamená nutnost zvládnout řady nových technologií a postupů, často v dosud nekomfortním prostředí. Nechť je tento příspěvek nápomocen aspoň pro první orientaci!

Tento příspěvek vznikl za podpory projektu Národního programu Informační společnost „E-learning v kontextu sémantického webu“, č. 1ET208050401.

LITERATURA

[O'Reilly, 2005] O'REILLY, Tim. *What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software*. 30 Sep 2005. Dostupné na World Wide Web:

<http://www.oreillynet.com/lpt/a/6228>

[O'Reilly, 2006] O'REILLY, Tim. *Levels of the Game: The Hierarchy of Web 2.0 Applications*. 17. července 2006. Dostupné na World Wide Web:

http://radar.oreilly.com/archives/2006/07/levels_of_the_game.html

[Fielding] Fielding, Roy Thomas. *Representational State Transfer (REST)*, kapitola z disertační práce, 2000. Dostupné na World Wide Web:

http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[Domenig] Domenig, Marc. *Rich Internet Applications and AJAX - Selecting the best product*

<http://www.javalobby.org/articles/ajax-ria-overview/>

[Coach] Wei, Coach. *AJAX, Java, Flash, and .NET*, <http://java.sys-con.com/read/232046.htm>

[ECMAScript] ECMA. *Standard ECMA-262 – ECMAScript Language Specification*,

<http://www.ecma-international.org/publications/standards/Ecma-262.htm>

[Pichlík] Pichlík, R. *Google Web Toolkit*, INTERVAL.CZ, <http://interval.cz/clanky/google-web-toolkit/>

[Pitner, 2006] Pitner, T. *Testování softwaru na platformě Java EE*, sborník konference Tvorba software, ČSSI a VŠB-TU, Ostrava, 2006

[IHT] Helft, M., Fabrikant, G. *Viacom sues Google over video clips on its YouTube site*, International Herald Tribune, March 14, 2007, <http://www.iht.com/articles/2007/03/14/news/web-0314viacom.php>

[GAA] Google Inc. *Google Account Authentication*, <http://code.google.com/apis/accounts/AuthForWebApps.html>

[AJAX] Wikipedia. *AJAX*, <http://cs.wikipedia.org/wiki/AJAX>

[Clarke] Clarke, G. *Berners-Lee calls for Web 2.0 calm*, The Register, 30.8.2006, http://www.theregister.co.uk/2006/08/30/web_20_berners_lee/