



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

MOŽNOSTI TECHNOLOGIE GOOGLE WAVE

GOOGLE WAVE TECHNOLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PAVEL MICHLÍK

VEDOUcí PRÁCE
SUPERVISOR

Ing. RADEK POLIŠČUK, Ph.D.

BRNO 2010

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2009/2010

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Pavel Michlík

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Možnosti technologie Google Wave

v anglickém jazyce:

Google Wave Technology

Stručná charakteristika problematiky úkolu:

Studium možností a aplikačního rozhraní interaktivní komunikační platformy Google Wave, potřebných pro tvorbu ukázek použitelných v oborovém studiu na ÚAI FSI

Cíle diplomové práce:

Příprava ukázkových aplikací (robotů, gadgetů) v jazycích Python a Java, demonstrujících možnosti programového rozšíření komunikační platformy Google Wave

Seznam odborné literatury:

Trapani, G., Pash A. The Complete Guide to Google Wave [online].
<<http://completewaveguide.com/>>

Švec, J. Učebnice jazyka Python [online]. <<http://macek.sandbox.cz/texty/python-tutorial-cz/tut/>>

Vedoucí diplomové práce: Ing. Radek Poliščuk, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2009/2010.

V Brně, dne 20.11.2009

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

Abstrakt

Diplomová práce pojednává o nové technologii Google Wave určené pro spolupráci více účastníků v reálném čase. Předmětem práce bylo popsat základní stavební prvky Google Wave a vytvořit ukázkové aplikace jako možnosti rozšíření této technologie. Výsledkem práce jsou dvě ukázkové aplikace. První aplikace (robot) je postavená pomocí klientských knihoven určené pro vývoj tohoto typu rozšíření a je napsána v jazyce Java. Druhá ukázková aplikace (gadget) využívá jiný způsob rozšíření Google Wave a je programována pomocí technologií XML, HTML, CSS a JavaScriptu.

Abstract

The Master's thesis explains API and features of Google Wave technology, which was designed for flexible real-time collaboration of defined participants. The project aim was description of Google Wave basic elements and creation of sample Wave extensions, demonstrating capabilities of the communication protocol. The first extension (robot) was developed using client libraries in Java. The second extension (gadget) shows another approach of Google Wave extensions, based on XML, HTML, CSS and JavaScript.

Klíčová slova

Google Wave Federation Protocol, FedOne, Google Wave, Google Wave Preview, Google Wave API, Google Wave Robot Wire, Gadgets API, Wave, Web 2.0, HTML5, Gadget, Robot

Keywords

Google Wave Federation Protocol, FedOne, Google Wave, Google Wave Preview, Google Wave API, Google Wave Robot Wire, Gadgets API, Wave, Web 2.0, HTML5, Gadget, Robot

Poděkování

Touto cestou bych rád poděkoval vedoucímu diplomové práce Ing. Radku Poliščukovi, Ph.D. za jeho věcné vedení a cenné rady při zpracování této práce.

Obsah

Abstrakt	5
Poděkování	7
1 Úvod	11
2 Google Wave	13
2.1 Wave koncept	14
2.2 Google Wave Federation Protocol	15
2.3 FedOne	16
2.4 HTML5	17
3 Google Wave Preview	19
3.1 Uživatelské rozhraní	19
3.2 První vlna	20
3.3 Vlna jako komunikační nástroj	22
3.4 Tvorba obsahu	24
4 Google Wave API	31
4.1 Google Wave Sandbox	31
4.2 Robot	31
4.3 Gadget	37
4.4 Wave Embed API	39
5 Ukázkové aplikace	41
5.1 RawWaveRobot	41
5.2 Gadget	47
6 Závěr	53
Literatura	55

1 Úvod

V dnešní době, kdy používání internetu neodmyslitelně patří k běžným lidských činnostem, se objevuje nová kategorie webových aplikací označovaných jako Web 2.0. Termín byl použit v roce 1999 Italem Darcy DiNucci, jenž se o něm zmiňuje v článku *Fragmented Future* [1]. Termín Web 2.0 jde velice těžko exaktně definovat. Označení Web 2.0 spíše představuje vývoj webových aplikací, kdy statický obsah webových stránek bývá nahrazen prostorem, ve kterém dochází k interakcím mezi uživateli a k následné tvorbě obsahu webu za účelem sdílení informací.

Termín je úzce spojen s Timem O'Reillym, který jej představil na konferenci O'Reilly Media Web 2.0 v roce 2004 a kde popsal nové trendy ve vývoji webových aplikací. John Batelle a Tim O'Reilly nastínili v úvodním projevu nové pojetí používání internetu ve smyslu „*Web jako platforma*“, kde jsou lokální aplikace z desktopového prostředí převáděny do webového prostředí [2]. Navrhli označení Web 2.0 jako novou verzi World Wide Webu. Neodkazují však na vývoj nových technologií, ale spíše na způsob, jakým vývojáři programují své webové aplikace a nový způsob chování uživatelů využívajících tyto aplikace.

Popularita označení webu jako Web 2.0 nabývá na důležitosti s rostoucím počtem využívaných blogovacích aplikací, wikipedie a sociálních sítí. Národním příkladem webové aplikace splňující požadavky Web 2.0 je Wikipedia [3]. Wikipedia vznikla za účelem vytvořit internetovou encyklopedii, která není založena na odborných znalostech jednotlivých expertů v daném oboru, jak je tomu u klasických encyklopedií, ale opírá se o důvěru anonymních uživatelů, kteří vytvářejí obsah a upravují jednotlivé články. S heslem „*given enough eyeballs, all bugs are shallow*“ dochází neustále k opravám, případně k aktualizacím veškerých informací [2].

Zejména v dnešní době jsou velmi populární webové aplikace, které zprostředkovávají prezentaci osoby pomocí jejích profilů, obsahující osobní informace a zároveň nabízejí možnost komunikovat se svými přáteli. Tyto aplikace jsou nazývány souhrnně sociální sítě. Zvláště pak aplikace typu Facebook, Twitter nebo MySpace lze považovat za fenomén dnešní doby a také za nejvýznamější představitele sociálních sítí.

Za velké hráče na trhu, kteří se snaží produkovat webové aplikace nesoucí označení Web 2.0, jsou považovány především společnosti Google a Yahoo. Společnost Google je jedna nejprogresivnějších společností ve vývoji webových aplikací. V této oblasti získala významný podíl na trhu. Je nutno podotknout, že hlavní příjmy společnosti pocházejí ze služby poskytující textové reklamy, což jí umožnilo získat některé kvalitní projekty a implementovat je do svých služeb. Příkladem takové strategie může být například koupě společnosti Pyra Labs, která provozovala nejpopulárnější blogovací službu <http://www.blogger.com> nebo společnost Applied Semantics stojící za projektem sémantického vyhledávání s využitím on-line reklamní kampaně. Tento projekt, který nese název AdSense, tvoří dnes většinu příjmů společnosti Google.

Zejména u společnosti Google pozorujeme rozvoj a nabídku internetových aplikací, které jsou využívány při nejrůznějších lidských činnostech na internetu. V poslední době se Google snaží vyvíjet aplikace, jenž jsou založené na nových technologiích a snaží se tak posunout webové aplikace opět kupředu.

V roce 2009 během konference Google I/O v San Franciscu společnost představila aplikaci založenou na novém protokolu. Byla zde prezentována nová platforma s názvem

Google Wave umožňující vytvářet obsah v reálném čase ve webovém prostředí. Google Wave je on-line aplikace poskytující prostor pro komunikaci mezi uživateli a umožňuje kooperaci uživatelů za účelem tvorby obsahu. Je navržena tak, aby sjednotila jednotlivé způsoby komunikace na internetu (nejvíce využívané služby jakými jsou: e-mail, instant messaging, sociální sítě a wiki) do jednoho webového komunikačního prostředí. Wave (vl-na) je chápána jako část konverzace nebo dokumentu, kde spolu mohou lidé komunikovat a spolupracovat, editovat text nebo vkládat obrázky, videa a mapy.

První část diplomové práce se zaměřuje na představení produktu Google Wave jako takového. V této části práce budou popsány jednotlivé technologie, které představují základní stavební kameny aplikace. Zpočátku bude popsána technologie založená na síťové komunikaci, poté na technologie reprezentující logickou část aplikace a způsob prezentace webové aplikace v prohlížečích.

Dále bude převažovat popisná část aplikace Google Wave (konkrétní webová aplikace založená na technologii Google Wave), především popis uživatelského rozhraní, které se od doposud známých aplikací odlišují. Obsahem této kapitoly bude popis jednotlivých funkcí a možností umožňujících pracovat s komunikačním prostorem a tvořit tak obsah.

Po představení uživatelského rozhraní budou nastíněny možnosti rozšíření této platformy o další užitečné funkce a vlastnosti, a to za pomoci externích softwarových aplikací. Nejprve bude seznámení se způsobem propojení aplikace s cizími částmi kódu pomocí Google Wave rozhraní a popis této komunikace. Následně bude objasněn způsob práce ve vývojovém prostředí, které umožňuje podílet se na vývoji Google Wave aplikace, případně testovat vlastní aplikace a opravovat chyby v aplikacích.

V předposlední kapitole budou představeny jednotlivé ukázky rozšíření pro aplikaci Google Wave, stručný popis jejich funkcí a vlastností umožňujících doplnit do aplikace potřebné chování. V rámci této kapitoly bude představen funkční robot (aplikace), který bude využívat většinu možností technologie Google Wave a demonstrovat získávání informací z různých zdrojů. Úkolem robota bude zpracovávat jednoduché i složitější matematické vzorce, převádět matematické nebo chemické vzorce do čitelného formátu a spouštět jiné funkce. Součástí této kapitoly bude také i prezentace jiného typu rozšíření (gadget) pro platformu Google Wave, které lze v rámci technologie Google Wave vyvíjet.

Po uvedení způsobů vývoje platformy Google Wave, představení ukázkových aplikací spojujících toto komunikační prostředí s jinými systémy (případně s jinými webovými aplikacemi), možností rozšíření aplikace (o nové vlastnosti a funkce) bude závěr zaměřen na přínos nových technologií ve vývoji webových aplikací a možnosti jejich využití v jiných webových projektech.

V diplomové práci jsou použity anglické termíny, protože oficiální české názvy zatím nejsou k dispozici. Doslovný překlad těchto termínů v rámci práce by mohl být zavádějící.

2 Google Wave

Google Wave představuje zcela novou platformu pro komunikaci a interakci uživatelů, založenou na poskytování dokumentů prostřednictvím tzv. Waves podporující současné úpravy ze strany všech uživatelů a zajišťující nízké latence při aktualizaci změny obsahu. Google Wave umožňuje lidem komunikovat a podílet se na tvorbě nejrozličnějších dokumentů téměř v reálném čase. Je snadno rozšiřitelný hlavně díky tomu, že existuje několik možností, jak implementovat nové vlastnosti a funkce.

Za touto myšlenkou stojí vývojáři, bratři Lars a Jens Rasmussen z Google Sydney, kteří předtím pracovali na projektu Google Maps. Kolem roku 2004 začali přemýšlet, že z projektu Google Maps odejdou a zvažovali možnosti nového projektu. Jens přišel s návrhem pracovat v oboru komunikace. Vznikl projekt s kódovým označením „Walkabout“ a začali si postupně pokládat jednotlivé otázky:

„Proč musíme žít s rozdíly mezi různými typy komunikace- e-mail versus chat, nebo konverzace versus dokumenty?“

„Mohl by jediný komunikační model obsáhnout všechny, nebo většinu systémů používaných dnes na webu, v jedno plynulé kontinuum? Jak bychom to mohli jednoduše udělat?“

„Co když jsme se snažili navrhnout komunikační systém, který využíval stávající schopnosti počítačů, než aby napodoboval neelektornické formuláře?“ [4]

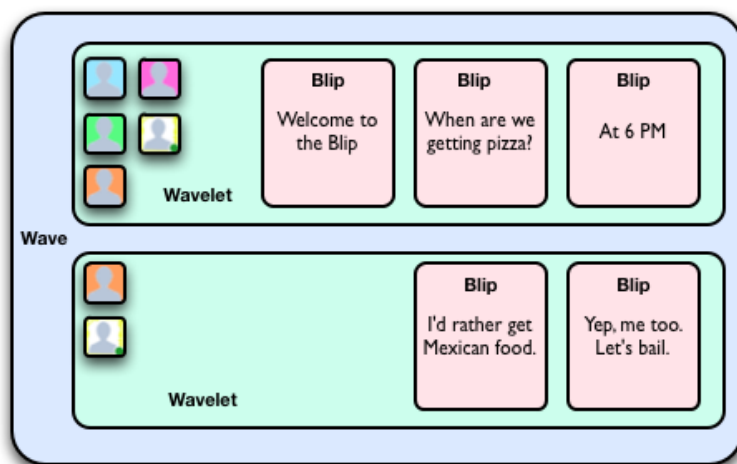
Po několika měsících vývoje v kancelářích Sydney dokázal pětičlenný tým vytvořit prototyp. O pár let později se o vývoji této technologie/službě veřejně zmiňují na konferenci Google I/O v San Franciscu konané 27. května 2009. Podle svých zakladatelů můžeme technologii Google Wave rozdělit na tři vrstvy:

- Google Wave Product (dnes označována jako Preview verze) je určena pro veřejnost, aplikace využívající návrhu Wave. Google Wave Preview je označována také jako HTML5 aplikace díky integrování a využití některých konceptů této specifikace.
- Google Wave může být považována také za platformu s otevřeným API rozhraním, které umožňuje dalším vývojářům propojit aplikaci s jinými webovými službami a vytvářet tak nová rozšíření, která jsou součástí vlny.
- Google Wave protokol je základní formát pro ukládání a sdílení dat prostřednictvím vln podporujících real-time komunikaci, která se projevuje okamžitým provedením jakýchkoliv změn. Protokol umožňuje spolupracovat s jinou službou (založenou na stejné technologii Wave) nebo přímo s Google Wave službou.[4]

Google Wave je napsána v jazyce Java využívající OpenJDK od společnosti Sun Microsystems. Webové (uživatelské) rozhraní je napsáno v Google Web Toolkit. Než přistoupíme k popisu jednotlivých technologií - stavebních prvků Google Wave, je potřeba zmínit základní koncept této technologie.

2.1 Wave koncept

- **Wave (Vlna)** - je chápána jako strukturová konverzace, která obsahuje jednoho nebo více účastníků podílejících se na tvorbě obsahu vlny nebo také jako entitu, v níž uchováváme veškerý obsah, přičemž je zaznamenávána historie změn obsahu. Lze si ji také představit jako prostor, kde dochází k interakci mezi účastníky v reálném čase. Vlna je chápána jako kontejner obsahující jednu nebo více tzv. Wavelet (Vlnek), tzn. vlnu je možné definovat jako množinu jednotlivých vlnek.
- **Wavelet (Vlnka)** - vlnku lze chápat jako konverzaci, která pochází z některé vlny. Vlnky jsou kontejnery, které obsahují jeden nebo více dokumentů. Jednoduché dokumenty, které obsahují pouze jednoduché zprávy, jsou označovány jako Blip (zpráva/příspěvek). Vedle toho jsou zde složitější dokumenty (Data Document). Vlnka je jednoduchý prvek, který má přístup k datům ve vlně. Všichni účastníci na vlně mají plné právo číst a editovat obsah v rámci jedné vlnky. Všechny události objevující se v Google Wave API jsou spouštěny na úrovni vlnky, nebo na nižší vrstvě. Pokud uživatel vytvoří vlnku v rámci určité vlny, nebude mít přístup k mateřské vlnce.
- **Document (Dokument)** - je základní prvek, který je podmnožinou vlnek. Existují dva základní typy dokumentů:
 - **Blip** - jeden ze základních prvků konverzace představující jednoduchou zprávu, která patří do vlnky. Blipy mohou obsahovat další blipy jako své potomky, přičemž každá vlnka obsahuje alespoň jeden kořenový blip.
 - **Data Document** - jsou dokumenty obsahující data, která jsou součástí vlnek, ale nejsou zobrazeny uživateli. Data mohou být uložena v jakémkoliv formátu. Typické je využití klíče a odpovídající hodnoty. Dokumenty jsou využívány jako lokální datová úložiště pro vlnky. [5]



Obr. 1 Wave koncept

Vztahy mezi jednotlivými entitami konceptu Google Wave vystihuje uvedený **obr. 1**. Google Wave pracuje na podobném principu jako předchozí systémy- e-mail nebo Usenet, ale místo posílání jednotlivých zpráv uživateli se zasílají dokumenty „Waves“ (Vlny), které

obsahují ucelenou sadu jednotlivých multimediálních zpráv (Blipy) uložených na serveru. Tyto vlny jsou sdíleny mezi spolupracovníky, kteří mohou být kdykoliv během existence vlny přidáni nebo odebráni.

2.2 Google Wave Federation Protocol

Google Wave je od samého počátku prezentována jako otevřená platforma, která umožňuje „federate“ (sdružovat) vlny/dokumenty s jinými poskytovateli wave služeb a komunikovat mezi sebou nebo s `wave.google.com`. Těžištěm této technologie je Google Wave Federation Protocol, který představuje server-server protokol. Z označení protokolu server-server vyplývá, že cílem protokolu není komunikace mezi serverem a klientem, nýbrž mezi poskytovateli wave služeb (komunikace mezi servery). Cílem protokolu je sdílet aktuální stav vlny mezi uživateli, jenž mají různé Wave providery. Protokol umožňuje sdružovat vlny mezi poskytovateli Wave služeb a internetu. Kdokoliv se může stát poskytovatelem Wave služeb za účelem vytvoření prostředí pro své osobní účely, nebo také využít jako službu v rámci firemního systému. Rovněž lze poskytovat tuto službu na internetu pro širokou veřejnost.

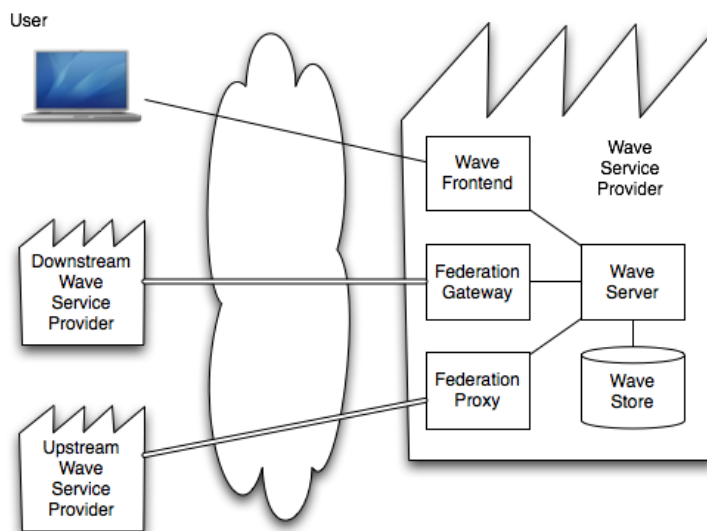
Každý poskytovatel Wave služeb je identifikován svým doménovým jménem, přičemž uživatelé obdrží wave adresu, která je velmi podobná syntaxi e-mailu. Adresa každého uživatele je složena z jeho jména a domény poskytovatele služeb `<username>@<domain>`. Uživatelé přistupují ke všem svým vlnám prostřednictvím svého providera. V případě, kdy vlna obsahuje účastníky z různých wave serverů, poskytovatelé udržují všechny kopie vlny (tyto kopie jsou jejím uživatelům k dispozici). Provideři Wave služeb produkují změny na vlně prostřednictvím federation protokolu, nesou odpovědnost za ověřování uživatele pomocí cookies a hesel a provádí lokální kontrolu přístupu.

V kapitole 2.1 byl nastíněn základní koncept Wave technologie. Každá vlna může obsahovat jednu nebo několik vlnek, které zahrnují blipy nebo datové dokumenty tvořící obsah každé vlnky. Každá z těchto vlnek si nese seznam uživatelů, kteří mají do vlnky přístup a jsou označováni za účastníky vlnky. Všichni provideři mezi sebou sdílejí kopie vlnek, které mají alespoň jednoho společného účastníka. Mezi těmito poskytovateli je určen základní poskytovatel, který vlastní konečnou kopii vlnky. Pokud dojde k otevření vlny uživatelem, dostane seznam vlnek, ve kterých je uživatel účastníkem (má přístup k vlnce přímo nebo nepřímo prostřednictvím členství ve skupině). Příkladem může být případ, kdy jsou data vlny zobrazena pouze uživateli, jenž je účastníkem dané vlny. Jiným příkladem může být soukromá vlnka zobrazující se pouze v případě, pokud do ní máme přístup.

2.2.1 Princip protokolu

Centrální částí protokolu je datové uložisko (wave store) ukládající operace, dále pak server, který zpracovává operace pomocí Operational Transformation, zapisuje a čte data z wave store. Wave služba poskytuje vlny uživatelům, kdy poskytovatelé služeb umožňují připojení do služby pomocí Google Wave Data Model and Client-Server Protocol. Z hlediska protokolu je důležité, že nám umožňuje sdílet vlny s účastníky pocházejících z různých wave serverů a udržovat stejný stav vlny mezi těmito servery. Tuto činnost nám poskytují dvě základní části protokolu, federation gateway a federation proxy.

- **Federation gateway (brána)** - má za úkol komunikovat s lokálními operacemi na vlnce.
- **Federation proxy** - umožňuje komunikovat se vzdálenými operacemi na vlnce. Poskytovatel potřebuje tuto část ke komunikaci federation brány se vzdálenými servery.



Obr. 2 Komunikace

Poskytovatel služby vyšle požadavek prostřednictvím federation brány, která se připojí k federation proxy jiného providera. Na základě požadavku dojde ke vložení operace v rámci vlnky zasláné poskytovatelem s požadavkem. Federation Protocol má několik mechanismů, jak spolehlivě doručit operace z brány do proxy. Federation brána udržuje dotazy odchozích operací pro každý vzdálený server (doménu). Tyto dotazy jsou udržovány ve frontě dokud, jejich přijetí nepotvrdí federation proxy. Federation brána se bude neustále pokoušet připojovat k proxy vždy, když dojde k selhání připojení. Po návázání spojení federation brána doručí frontu s operacemi. Po přijetí fronty proxy zašle oznámení o této skutečnosti zpět federation bráně. Jestliže potvrzení neobdrží, odesílatel nezařadí operace jako doručené. [6]

2.3 FedOne

Během představení technologie Google Wave se společnost zavázala k otevření kódů a protokolů, které tvoří stavební prvky aplikace. První velká část kódu, která byla uvolněna je FedOne Wave Reference Server. Projekt obsahuje návrh specifikace Google Wave Federation Protocol a zdrojový kód v jazyce Java. Prototyp serveru FedOne obsahuje několik částí: Operational Transform (OT)¹, základní model Wave, Federation Protokol a klient-server, který využívá wave protokol. OT je srdcem celé kolaborace technologie Google wave.

¹ Operational Transform (OT) - je technologie, která byla vynalezena pro kontrolu a údržbu shodnosti editace jednoduchých dokumentů během spolupráce více uživatelů. Podporuje mnoho funkcí určené pro groupware systémy. [7]

Tento projekt neobsahuje některé části kódu, a to především klientské knihovny v jazyce Java a Python. Je také potřeba si uvědomit, že uvolněný kód není samostatný XMPP server, ale externí aplikace implementující XEP-0114 (Jabber Component Protocol), který představuje podporu rozšíření pro XMPP server. [8]

2.4 HTML5

HTML5 představuje hlavní revizi značkovacího jazyka HTML určeného k tvorbě webových stránek. WHATWG² začala pracovat v červnu 2004 na specifikaci pod názvem Web Applications 1.0 [9]. V říjnu 2009 sdružení WHATWG vydalo zprávu o poslední verzi specifikace, která nese název „*Last Call*“ [10].

Základní specifikaci použila v roce 2007 skupina W3C³ jako výchozí bod pro svou budoucí specifikaci. Tato skupina předložila svůj první veřejný pracovní návrh 22. ledna 2009 [11]. Na uveřejněné specifikaci stále probíhají práce a očekává se, že zůstane po dobu několika let respektována. Některé součásti specifikace HTML5 jsou již implementovány v prohlížečích dříve, než bude vydána konečná doporučená specifikace HTML5. Odhaduje se, že tato specifikace bude kandidátem na doporučenou verzi v roce 2012. Již dnes je možné najít jisté pokroky u jednotlivých částí specifikace, které jsou stabilní a existují implementace v prohlížečích nebo jsou blízko k dokončení. Klasickým příkladem může být prvek `<CANVAS>`. Naproti tomu jsou některé oblasti, na kterých se neustále intenzivně pracuje, mění se a k některým dosud dokonce neexistuje dokumentace [12].

2.4.1 Vlastnosti specifikace

HTML5 definuje nové elementy a atributy reflektující potřebu vytvářet webové stránky. Některé elementy nahrazují prvky, které se objevují v každém moderním webu pro tvorbu layoutů pomocí bloků `<div>` nebo řádkových elementů `<span>`. Takovým příkladem může být vykreslení základního layoutu obsahující základní bloky. Místo obvyklých předpisů `<div id="header">` pro určení hlavičky stránky (nebo jiných částí, patička, navigace, atd.) jsou implementovány nové atributy `<header>` (`<footer>`, `<navigation>`) nahrazující tyto předpisy.

Syntaxe HTML5 už není založena na SGML i přes podobnost značky. Specifikace je nicméně navržena tak, aby byla zpětně kompatibilní se starší verzí. Dodává se nový řádek do dokumentu, který vypadá jako SGML deklarace `<!doctype html>`, umožňující zpětně kompatibilní zobrazení ve všech prohlížečích.

Příklad 1 HTML5

```
1  <!doctype html>
2  <html>
3    <head></head>
4    <body>
```

² Web Hypertext Application Technology Working Group (WHATWG) je skupina zabývající se vývojem HTML a souvisejícími technologiemi. Tato skupina byla založena jednotlivci z firem Apple, Opera Software nebo také z komunity Mozilla Foundation.

³ World Wide Web Consortium (W3C) je mezinárodně uznávaná organizace zabývající se World Wide Web (WWW).

```
5   <header>...</header>
6   <nav>...</nav>
7   <article>
8     <section>
9     ...
10    </section>
11  </article>
12  <aside>...</aside>
13  <footer>...</footer>
14  </body>
15 </html>
```

Uvedený **př. 1** syntaxe HTML5 ukazuje jenom několik základních tagů, které jsou obsaženy v této specifikaci. Kromě změn některých tagů, HTML5 zavádí několik API rozhraní pro programování webových aplikací. Stávající DOM⁴ je rozšířen o některé nové funkce a především jsou standardizovány a zdokumentovány některé návrhy.

- Rozhraní pro přehrávání videa a zvuku. Zavádí se nové atributy pro vložení těchto médií: `<video>` a `<audio>`.
- Možnost vytváření offline webových aplikací díky rozhraní pro tvorbu offline databází.
- Drag & drop API, které umožňuje jednoduše pomocí myši přemísťovat jednotlivé objekty na stránce.
- Nové rozhraní, které nabízí editování dokumentu.
- Specifikace implementuje také rozhraní pro 2D kreslení a tvorbu bitmapových obrázků. Zavádí se nový atribut `<canvas>`.

Kompletní seznam rozdílů HTML5 proti starší specifikaci HTML je uveřejněn na stránkách skupiny W3C. [14]

⁴ Document Object Model (DOM) představuje objektovou reprezentaci dokumentu HTML a XML. Tedy DOM je rozhraní, které umožňuje přistupovat k dokumentu, modifikovat jeho obsah, strukturu nebo jednotlivé části dokumentu. [13]

3 Google Wave Preview

Původně byla vydána verze pouze pro vývojáře, avšak 29. září 2009 byla uvolněna webová aplikace Google Wave Preview pro prvních 100.000 zájemců, kteří si podali žádost prostřednictvím internetové stránky u společnosti Google.

Označení webové aplikace Preview napovídá, že se jedná o stále se vyvíjející software, který byl spuštěn pro veřejnost z důvodu obdržení zpětné vazby od uživatelů. Google Wave Preview je produkt, konkrétní webová aplikace umožňující široké veřejnosti vyzkoušet si technologii Google Wave. Uživatelé se mohou prostřednictvím této webové aplikace podílet na tvorbě obsahu se svými přáteli, spolupracovat, editovat jednotlivé příspěvky, vkládat data různého formátu, vytvářet fotogalerie, využívat rozšíření umožňující propojit aplikaci s dalšími webovými službami, získávat data třetích stran nebo vkládat mapy a videa. Všechny tyto úkony jsou prováděny v reálném čase.

Google Wave Preview je webová aplikace využívající specifikace jazyka HTML5. K bezpečnému chodu aplikace je nutné zvolit vhodný prohlížeč, který implementuje některé prvky specifikace HTML5. Je-li použit prohlížeč, který nemá implementovanou specifikaci, Google Wave Preview upozorní uživatele na tuto skutečnost před vstupem do aplikace.

Pokud uživatel nemá přístup do Google Wave Preview, existují dvě možnosti získání přístupu do aplikace. První možností, jak získat oprávnění ke vstupu, je tzv. nominace od uživatele, jenž už má přístup do aplikace Google Wave. Při první registraci do aplikace uživatel obdrží 8 pozvánek (nominací). Někteří z uživatelů obdrželi nominace při prvním přihlášení, jiní museli několik týdnů na nominace vyčkat. Společnost Google všechny nominace schvaluje a potom rozesílá pozvánky do aplikace. Druhou možností je přímé podání žádosti u společnosti Google na adrese <https://services.google.com/fb/forms/wavesignup/>.

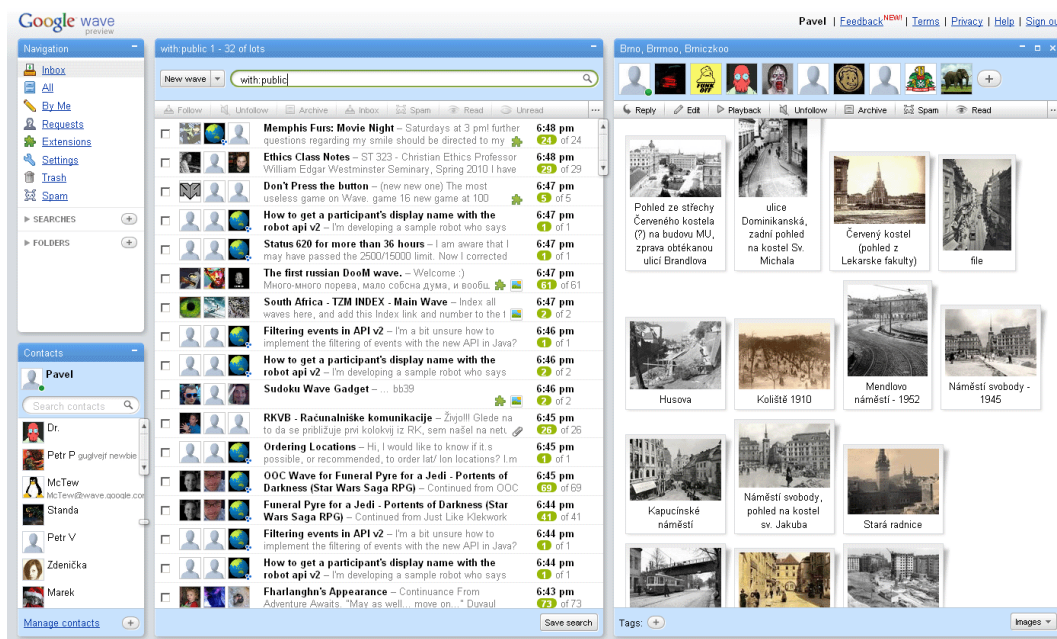
Dne 18. května 2010 byla na oficiálním blogu Google Wave zveřejněna zpráva, že přístup do aplikace Google Wave Preview je od tohoto data otevřen všem, kteří mají účet u společnosti Google. Uživatelé, kteří účet nemají, si jej mohou zdarma založit.

3.1 Uživatelské rozhraní

Při prvním přihlášení do aplikace Google Wave Preview překvapí neobvyklým vzhledem uživatelského rozhraní v porovnání s jinými webovými aplikacemi viz **obr. 3**. Layout s jednotlivými panely na stránce je obdobný e-mailovému klientovi. Asi největším rozdílem ve vzhledu aplikace oproti jiným stránkám je skutečnost, že aplikace nemá klasický scrollbar, který by umožnil rolovat s celou stránkou. Celá stránka je rozložena na velikost prohlížečského okna a je rozdělena do několika nezávislých panelů.

Základní vzhled aplikace Google Wave je rozdělen do tří sloupců, jenž obsahuje čtyři základní panely **obr. 3**. Levá část aplikace nabízí dva panely v jednom sloupci. První panel obsahuje základní navigační menu, obdobné e-mailovým klientům. Druhý panel, umístěný pod prvním panelem, obsahuje kontakty z Googlu účtu. V panelu kontaktů jsou uvedeni pouze uživatelé, kteří mají přístup do aplikace Google Wave Preview.

V prostřední části aplikace se nachází třetí panel obsahující v horní části vyhledávací lištu, pod ní je umístěna na šedém podkladě nástrojová lišta. Obsahem celého panelu je seznam vln. V inboxu je seznam aktivních vln, které obsahují změny od posledního prohlížení vlny. Poslední sloupec obsahuje panel umožňující prohlížet obsah vlny (pokud není žádná vlna otevřená, v tomto panelu je k naleznutí tlačítko pro vytvoření nové vlny).



Obr. 3 Google Wave Preview

Jestliže v tomto panelu obsah vlny překročí viditelnou obsahovou část, objeví se na pravé straně tohoto panelu scrollbar, který je odlišný od scrollbaru známého z běžných webových stránek. Pokud je v tomto panelu otevřená vlna, nachází se na horní části panelu seznam spolupracovníků ve vlně. Pod tímto seznamem účastníků se nachází nástrojová lišta jako v prostředním panelu obsahující seznam vln. Tato nástrojová lišta slouží k provádění definovaných operací s vlnami.

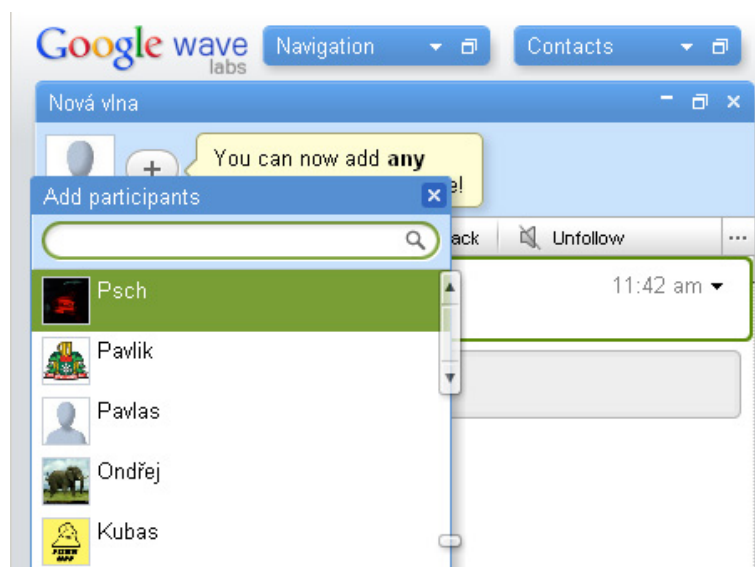
3.2 První vlna

V případě, že uživatel obdrží přístup do aplikace pomocí nominace od jiného uživatele, objeví se jeho kontakt v seznamu v panelu kontaktů. Pokud se tedy jakýkoliv kontakt z vašeho seznamu kontaktů z Google účtu stane novým uživatelem Google Wave, projeví se tato změna zobrazením kontaktu v seznamu kontaktů v Google Wave Preview. Pokud je seznam kontaktů prázdný, je možné pokračovat vyhledáváním veřejných vln v **kap. 3.4.1**.

Novou vlnu lze vytvořit tlačítkem umístěným v prostředním panelu vedle vyhledávací lišty. Kliknutím na tlačítko se v pravém panelu zobrazí prázdná vlna, kde je možné ihned začít psát konkrétní text. Stisknutím tlačítka **Done** dojde k ukončení editaci, v tomto okamžiku se nám zobrazí pop-up okno se seznamem kontaktů viz **obr. 4**. Kliknutím na jednotlivé kontakty jsou ti přidáváni jako další účastníci do vlny.

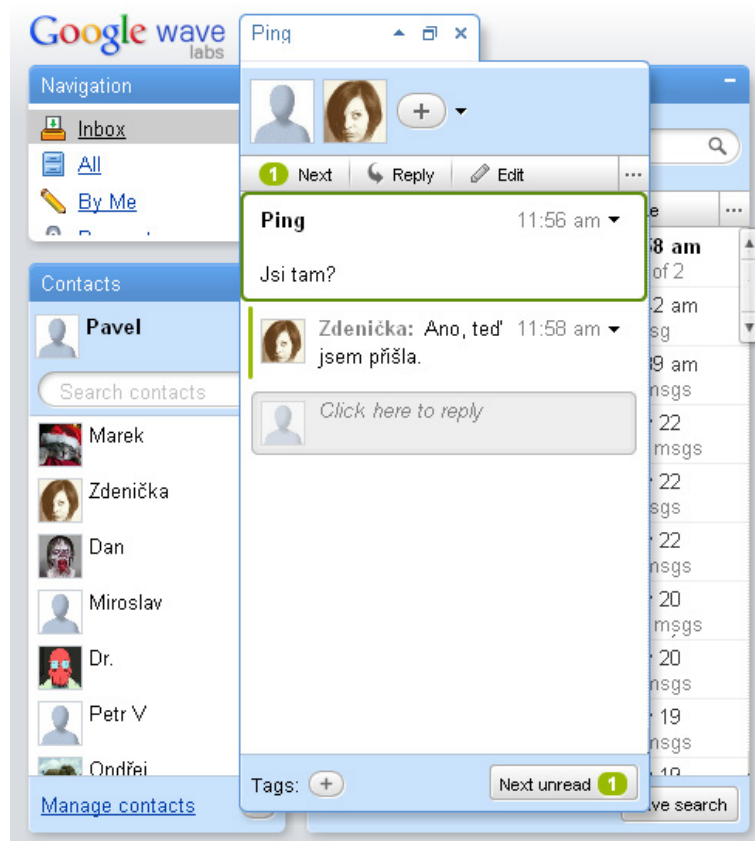
Další možností jak vytvořit vlnu, je kliknutí na ikonu kontaktu v seznamu. Zobrazí se pop-up okno s informacemi o daném uživateli a v pravém dolním rohu okna se objeví tři tlačítka: **New wave**, **Ping**, **Recent waves**. **New Wave** - tlačítko nám zajistí vytvoření nové vlny s tímto kontaktem. Tlačítko z této nabídky je **Recent Waves**, v prostředním panelu zobrazí všechny vlny, jejichž spolupracovníky je uživatel a vybraný kontakt.

Posledním tlačítkem je tzv. **Ping**. Slouží pro rychlé zahájení konverzace pro okamžitou reakci vybrané osoby. Klasickým příkladem způsobu této komunikace je instant messaging. V této chvíli uživatel nechce vytvářet plnohodnotnou vlnu, ale jenom rychle



Obr. 4 Vytvoření nové vlny

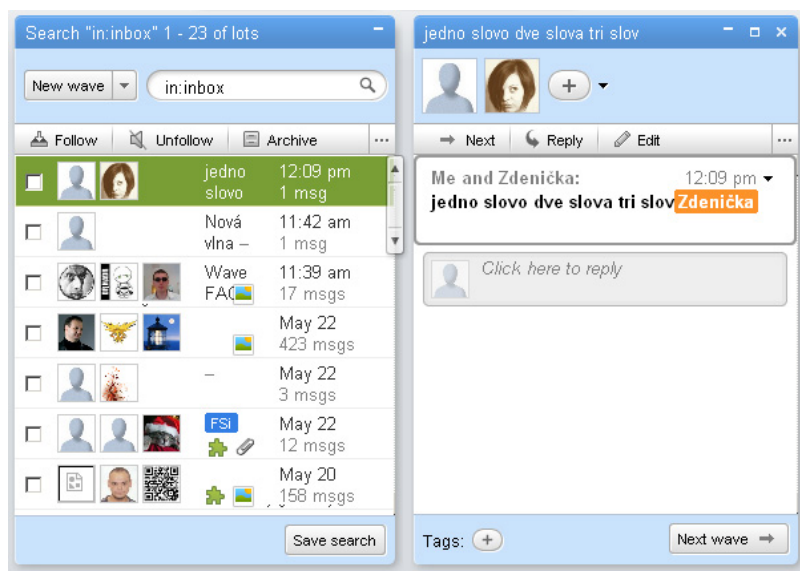
zahájit komunikaci s druhou stranou. Právě k tomuto účelu slouží tlačítko **Ping**, které vytvoří vlnu k zahájení komunikace s kontaktem. Tlačítkem **Ping** se v horní části okna zobrazí malé okno představující vlnu, kde je vyhrazen prostor pro psaní kratších zpráv. Ping panel je možné minimalizovat. Pokud dojde k obsahové změně vlny, bude lišta okna obarvena zelenou barvou. [15]



Obr. 5 Tlačítko Ping

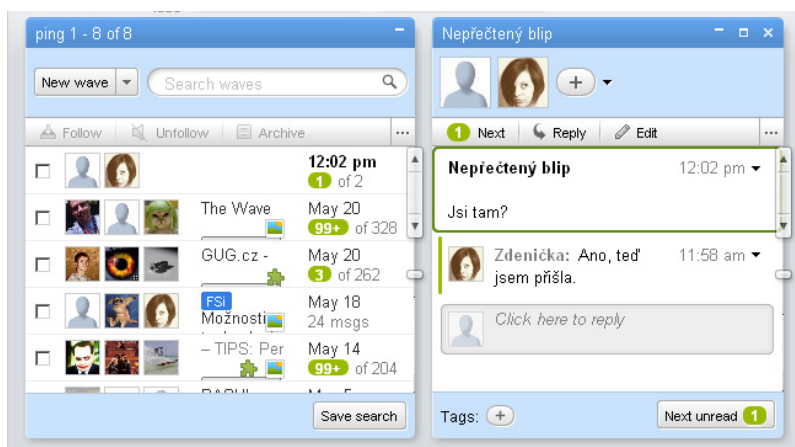
3.3 Vlna jako komunikační nástroj

Vlna představuje nástroj pro práci s dokumenty a spolupráci více účastníků na jedinou, pracujících v reálném čase. Všechny změny, které jsou provedeny na vlně, jsou okamžitě aktualizovány tak, aby každý z účastníků měl právě vždy poslední verzi dokumentu.



Obr. 6 Wave kurzor

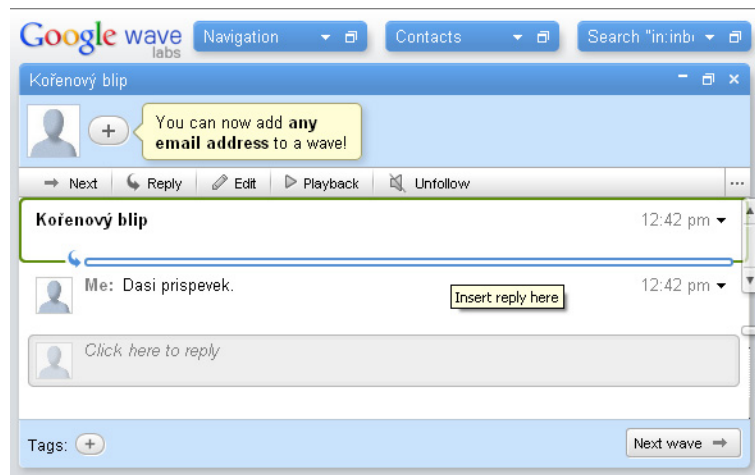
Jestliže je vkládán, vytvářen, editován obsah na vlně zároveň s dalšími účastníky, aplikace Preview zobrazuje barevné kurzory se jmény uživatelů, kteří provádějí aktuální editaci obsahu. Kurzor zobrazuje znak po znaku tak, jak jej uživatel píše do vlny. Vše je zobrazováno na **obr. 6**. Tento způsob editování představuje nový přístup vnímání spolupráce mezi účastníky, kdy jim je zajištěn přístup k dokumentu během úprav jiných spolupracovníků.



Obr. 7 Nepřečtené vlny a blipy

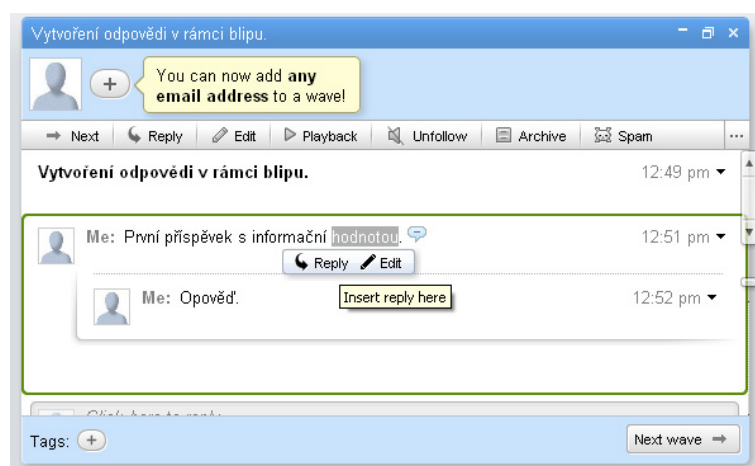
Všechny změny obsahu uskutečněné na vlně se projevují grafickým upozorněním. V prostředním panelu, kde se nachází seznam vln, se veškeré změny projevují tučným písmem v předmětu vlny a vpravo bude zobrazena zelená ikonka s počtem nepřečtených

blipů od posledního čtení. Pokud dojde k otevření aktualizované vlny, budou veškeré změny provedené v blichech zobrazeny zeleným pruhem umístěným vlevo. Takto označený blip představuje nepřečtenou zprávu ve vlně viz **obr. 7**.



Obr. 8 Vytvoření odpovědi

Aplikace nabízí několik způsobů, jak lze editovat, případně vkládat nový obsah do vlny. První možností je odpovědět vložením dalšího blipu pod již existující blip najetím kurzoru myši na spodní hranu blipu. Vytvoří se blip, který představuje v hierarchické struktuře dokumentu stejný stupeň vnoření. Vytvořením nového blipu ve vlně, která již obsahuje několik příspěvků, můžeme provést posunem kurzoru myši mezi blipy, zobrazí se opět modrý rámeček, kliknutím na něj pak vložíme nový blip, jenž se zobrazí odsazením od rodičovského blipu. Tento případ je demonstrován na **obr. 8**.



Obr. 9 Inline odpověď

Další a velice elegantní možností je vložení blipu v rámci jiného blipu, tzn. možnost umístit libovolně v textu nový blip, reagující pouze na určitou větu nebo část textového řetězce v blipu.

Dvojitým kliknutím levého tlačítka kurzoru myši v místě textu, kde je vhodné umístit nový blip, dojde k vyvolání nabídky obsahující položky **Reply** a **Edit**. Stisknutím tlačítka **Reply** se zobrazí nový blip, přičemž začátek vloženého blipu v textu je indikován grafickou ikonkou v podobě bubliny se symbolem plus nebo mínus. Pomocí této funkce je možné

sbalit, případně rozbalit blip v místě vložení. Celý proces je zobrazen na **obr. 9**, kde lze vidět nabídku pro vytvoření inline odpovědi i výsledek zobrazení odpovědi v rámci jiného blipu.

Poslední možností, vytvoření obsahu ve vlně, je editace již vytvořených blipů. Editovat stávající blipy je možné v editačním režimu, který lze vyvolat několika způsoby. Prvním způsobem vyvolání editace z roletkové nabídky, kterou lze zobrazit pomocí ikonky ve tvaru trojúhelníku umístěného v pravém horním rohu každého blipu. V této kontextové nabídce je k shlédnutí seznam veškerých úkonů, které lze s blipem provádět. Druhá možnost je pomocí tlačítka **Edit** na nástrojové liště umístěné pod seznamem účastníků vlny se přepnout do editačního režimu. Na této nástrojové liště jsou umístěny tlačítka pro práci s vlnou.

3.4 Tvorba obsahu

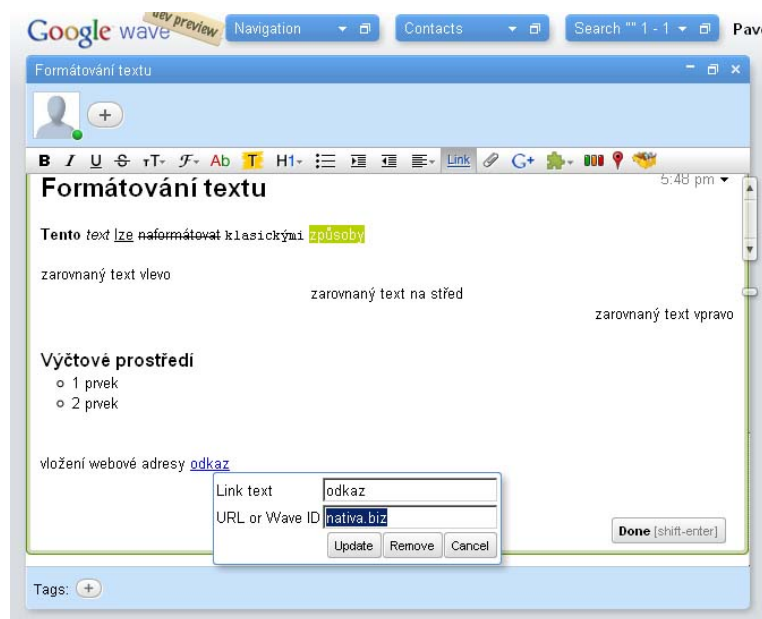
Vnímání tvorby obsahu ve vlnách je jednou z nejsilnějších stránek technologie Wave. Aplikace nám nabízí pohodlně a velmi flexibilně vkládat různé formáty obsahu. Představme si tvorbu obsahu např. v e-mailu. Mail byl vytvořen proto, aby si lidé mohli prostřednictvím počítačové sítě zasílat jednoduché zprávy. Pokud chceme zasílat jiná data než textového charakteru, musíme je v e-mailu poslat jako přílohu, tzn. tato data nejsou součástí těla zprávy. Naproti tomu je v rámci vln možno vkládat různé formáty dat a informací (např. texty, fotografie, videa, mapy atd.), které se stávají obsahem celé zprávy a vytváří komplexní prostor obsahující rozdílné formáty dat.

Vytvářet obsah v prostředí vlny je velmi intuitivní. Při vytvoření blipu je uživatel automaticky přepnut do editačního režimu a ihned může psát jednoduchý neformátovaný text. V editačním režimu se místo navigační lišty ve vlně zobrazí editační lišta s tlačítky k formátování textu tak, jak je známe z běžných textových editorů. Lišta nabízí spoustu možností formátování textu, ať už od klasického zvýrazňování textu, nebo vytváření složitějších textových zápisů. Za klasickými tlačítky umožňující základní formátování textu se nachází tlačítka pro rozšíření obsahu vlny, tedy tlačítka pro vkládání jiného typu informací (přílohy, fotografie, externí aplikace atd.).

Formátování textu

Jestliže uživatel začne vkládat nebo upravovat text, který je potřeba formátovat, tento text si označí myší jako výběr a k úpravě použije tlačítka zobrazená na editační liště. K některým základním formátovacím funkcím existují klávesové zkratky, které mohou urychlit formátování textových řetězců. Na **obr. 10** vidíme editační lištu a příklady formátovaného textu v rámci vlny. Editací lišta nám umožňuje:

- tučný, přeškrtnutý, podtržený, kurzívou formátovaný text
- výběr ze 14 různých fontů
- vytváření odstavců, zarovnávání na střed, vpravo a vlevo
- vytváření výčtových prostředí
- výběr z několika úrovní písma



Obr. 10 Formátování textu

Pro vložení odkazu na webovou stránku stačí vybrat text a kliknout na tlačítko **Link** na editační liště, nastavit výchozí adresu a název odkazu. Druhou možností vložení odkazu do vlny lze provést pomocí tlačítka v podobě **G+**, které zprostředkuje pop-up okno s Google vyhledávačem pomocí kterého lze vyhledávat webové stránky, obrázky, videa a následné vložení do prostředí vlny.

Přílohy

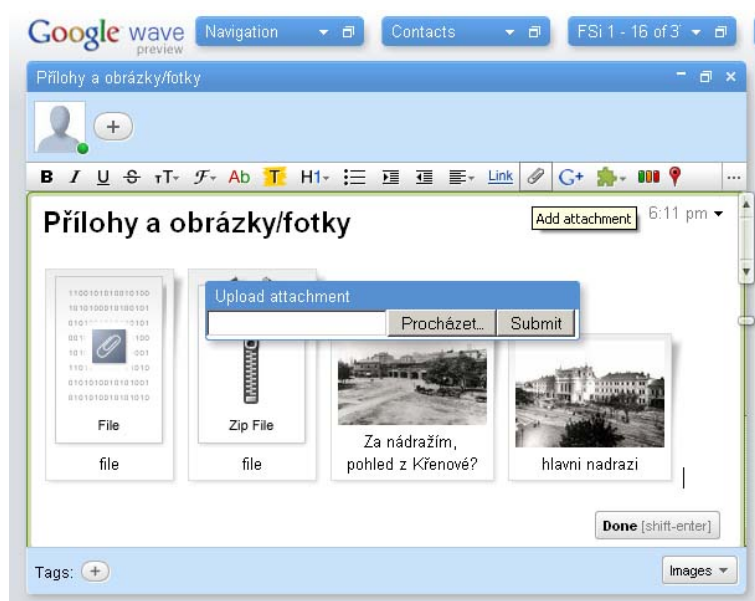
Existují dvě možnosti uložení přílohy do prostředí vln:

- První možností je použití tlačítka v podobě spony umístěné na editační liště. Zobrazí se dialogové okno, kde je nutné vybrat cestu k danému souboru, a vložit.
- Druhou možností je využít drag & drop funkci. K této funkci je nutné mít nainstalovaný plugin Google Gears (je uvolněn pro všechny dnes nejpoužívanější webové prohlížeče) ve vašem webovém prohlížeči.

Vložené přílohy (ne obrázky) jsou zastoupeny ikonou indikující formát přílohy. Ve výchozím nastavení jsou přílohy v podobě obrázků a fotografií prezentovány jako malé náhledy. Ať už se jedná o běžnou přílohu nebo obrázek, každý takto přiložený soubor obsahuje popisný text (titulek). Při nahrávání přílohy je popisný text nastaven na jméno souboru bez přípony. Při úpravě titulku se název souboru nemění, pouze popisný text v rámci vlny.

Maximální velikost jedné přílohy je omezena na 20 MB. Všechny snímky nahrané přes Google Wave budou převzorkovány kvůli snížení velikosti souboru. Nedoporučuje se používat Google Wave pro přenos fotografií ve vysokém rozlišení. Naproti tomu nabízí velmi pohodlný způsob on-line prezentace kvalitních obrázků a fotografií.

Pokud je více fotografií nebo obrázků ve vlně, je možné si soubor snímků prohlédnout jako „slide-show“. V pravém dolním rohu panelu zobrazující obsah vlny se nachází tlačítko



Obr. 11 Přílohy



Obr. 12 Prohlížení obrázků

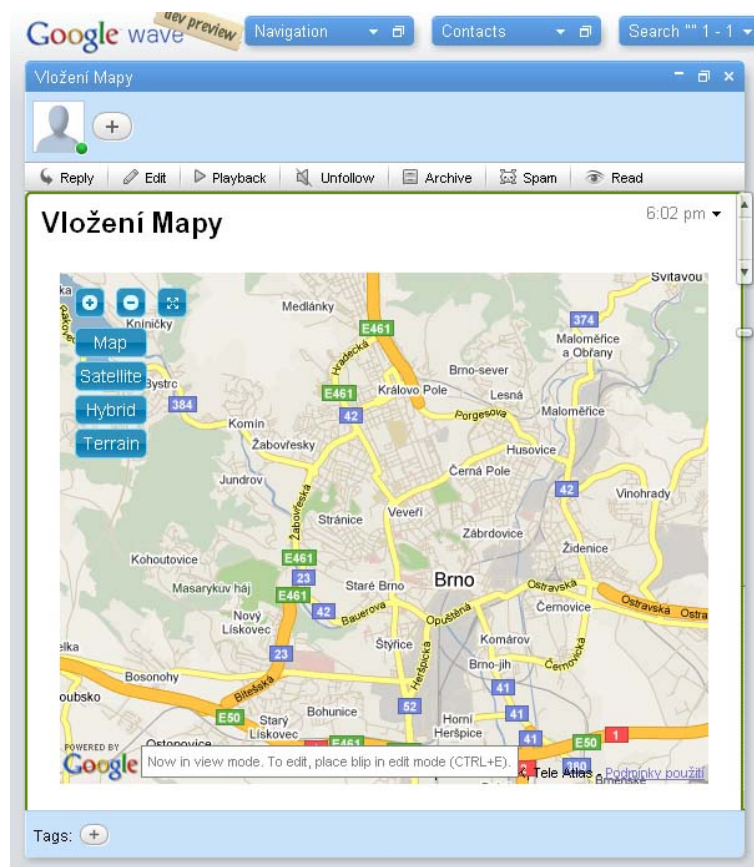
Images obsahující menu s funkcí **View as slide show**. Vyvoláním této funkce se zobrazí seznam fotografií s náhledy, v nichž lze listovat, jak je znázorněno na **obr. 12**.

Gadgets

Ve výchozím nastavení se na editační liště nachází tlačítka pro vložení aplikací, která přidávají interaktivní ovládání do prostředí vln a jsou zakomponována do Google Wave Preview jako výchozí gadgety: gadget Map a Yes/No/Maybe gadget.

Gadget Map je aplikace umožňující vložit interaktivní mapu do vlny. Google Wave byla vytvořena stejnými vývojáři osvědčené technologie Google Maps, a proto lze očekávat bezpečné fungování aplikace/gadgetu v prostředí vln. V této aplikaci je možné prohlížet mapu, vyhledávat různá místa, přidávat výchozí body na mapu, kreslení čar mezi jednotlivými body. Lze také přepínat mezi jednotlivými druhy map: hybridní, satelitní, terénní.

Gadget Map nabízí obdobné funkce, které jsou známé z webové aplikace Google Maps. Všechny změny provedené v tomto gadgetu vidí všichni účastníci vlny v reálném čase. Jestliže tedy dochází k přiblížení části mapy, nebo vložení bodů na mapu, účastníci na vlně vidí všechny úkony prováděné v aplikaci.

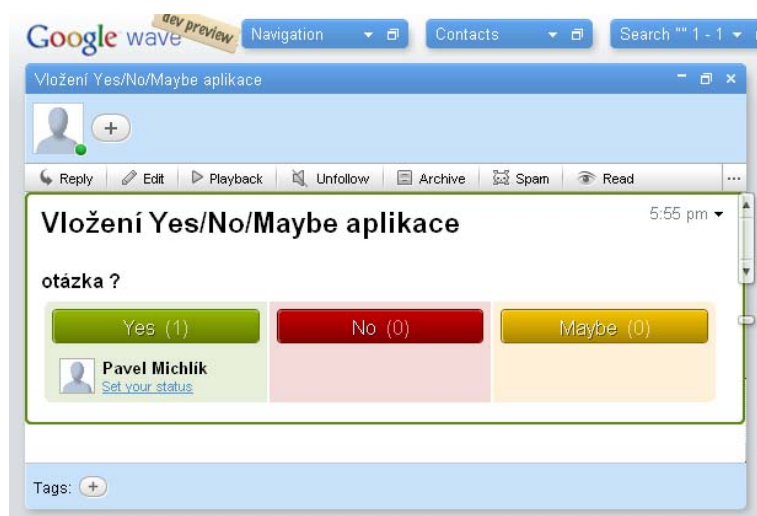


Obr. 13 Gadget Mapa

Yes/No/Maybe gadget představuje aplikaci umožňující uchovávat odpovědi na definovanou otázku. Ikona indikující tuto aplikaci na editační liště je v podobě tří barevných obdélníků, umístěná nalevo od tlačítka k vložení mapy. Do blipu, uživatel napíše otázku a poté vloží pod otázku tento gadget. Zobrazí se tabulka o třech sloupcích s atributy, které představují odpověď na danou otázku. Chce-li jiný účastník reagovat na otázku, klikne na jednu ze tří možností (atributů tabulky) v horní části aplikace. Poté se zobrazí ikona v příslušném sloupci viz **obr. 14**. Je možné přidat poznámku pomocí tlačítka **Set my status** umístěného pod ikonou účastníka. Tato poznámka bude zobrazena pod ikonou. Reakci na otázku lze změnit kliknutím na jinou odpověď (atribut tabulky).

3.4.1 Vyhledávání

Vyhledávání je funkce, kterou nejvíce ocení uživatelé při hledání svých vln nebo prohledávání různých informací ve vlnách. V horní části prostředního panelu je umístěno formulářové políčko k zadávání různých dotazů. Vyhledávání v rámci Google Wave je možné rozdělit na dva způsoby. Základní vyhledávání umožňuje vyhledat vlny, ve kterých je obsaženo definované slovo. Druhým způsobem je hledání pomocí specifických atributů.



Obr. 14 Yes/No/Maybe gadget

Pro základní vyhledávání našich vln stačí napsat specifické slovo obsažené ve vlně do formulářového pole a stisknout enter. Dojde k prohledání všech vln, kde jsme účastníky a zobrazí se seznam vln obsahující zadané slovo. Lze samozřejmě kombinovat různá slova a vkládat různé parametry pro splnění různých podmínek. Příkladem může být parametr AND, který zajistí vyhledání vln obsahující obě slova, mezi kterými je tento operátor umístěn. Kompletní seznam jednotlivých oprátorů lze najít v příručce pro Google Wave.



Obr. 15 Vyhledávání veřejných vln

Pokročilé vyhledávání v Google Wave je založené na zadávání operátoru a hodnoty ve tvaru `operator:hodnota`, podle kterých se bude prohledávat. Jednoduchý příklad použití můžeme demonstrovat na zobrazení všech veřejných vln v rámci Google Wave. Je nutné se zmínit, že aplikace Google Wave nabízí možnost vytvářet veřejné vlny, do kterých se může každý uživatel aplikace Google Wave připojit. Veřejná vlna se vytvoří přidáním uživatele `public@appspot.com` do seznamu kontaktů a poté přidáním do vlny, kterou chceme prezentovat jako veřejnou. Takové vlny lze vyhledávat zadáním operátoru a hodnoty do vyhledávacího pole v této podobě: `with:public`. Tento příkaz zajišťuje zobrazení všech veřejných vln v rámci Google Wave. Kompletní seznam operátorů a hodnot lze najít opět v příručce pro Google Wave.

V praxi je možná kombinace obou způsobů. Příkladem může být vyhledání všech veřejných vln zabývajících se například operačním systémem Android pro mobilní telefony. Příkaz pro vyhledání vln dle zadaných kritérií může vypadat takto: `with:public Android` viz **obr. 15**.

V této kapitole nebyl představen kompletní seznam funkcí a možností, které lze ve webovém prostředí Google Wave Preview vykonávat. Byly zde ukázány a popsány základní vlastnosti aplikace představující stručný přehled možností technologie. Vývojáři Google Wave neustále zdokonalují a aktualizují funkce aplikace Preview, proto některé zde uvedené popisy už nemusí odpovídat skutečnosti. Více informací o dalších vlastnostech a funkcích jsou prezentovány v příručce Google Wave umístěné na webové adrese <http://completewaveguide.com>.

4 Google Wave API

Google Wave jako open-source „platforma“ nabízí vývojářům několik API rozhraní, které umožňují implementovat další části kódu do Google Wave. Je možné tyto API rozhraní kombinovat mezi sebou. K dispozici jsou tři způsoby rozšíření webového klienta Google Wave:

- **Roboti** - Představují účastníky na vlně, kteří mohou číst, editovat, reagovat na různé události uskutečněné v rámci vlny.
- **Gadgets** - Webové aplikace s uživatelským rozhraním implementující nové vlastnosti a funkce do Wave dokumentu.
- **Embedded Waves** - Tento způsob nepřidává nové možnosti do aplikace, ale umožňuje jednoduše implementovat prostředí vlny do webových stránek.

Vývojáři vyvíjející nové rozšíření pro technologii Google Wave si mohou vybrat dvě webová rozhraní pro testování svých aplikací. Buď využít již zmíněný produkt uvolněný pro veřejnost Google Wave Preview, nebo využít vývojářské prostředí nesoucí označení Google Wave Sandbox.

4.1 Google Wave Sandbox

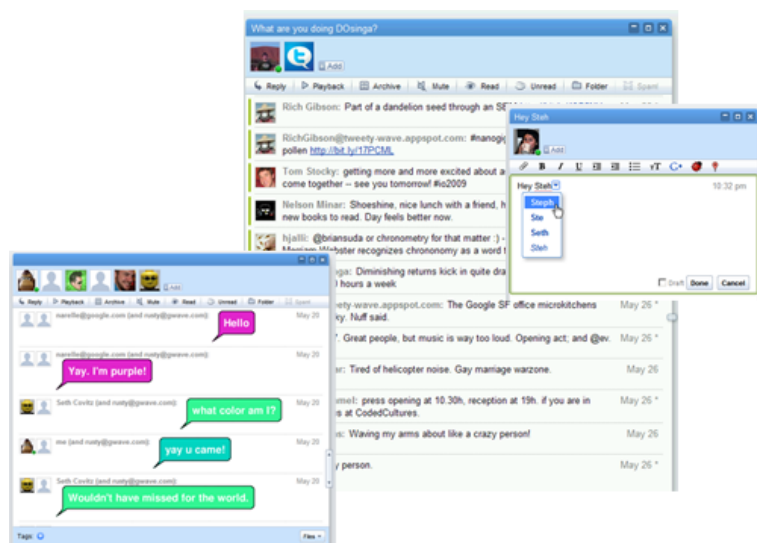
Vývojářská verze webového prostředí Google Wave Sandbox umožňuje především ladit své aplikace s jinými vývojáři, případně testovat své aplikace před nasazením do veřejného prostředí Preview. Ačkoliv jsou si prostředí velice podobná, verze pro vývojáře obsahuje několik rozdílů:

- Umožňuje testovat nové experimentální API funkce, které jsou implementovány do rozhraní Sandbox před ostrým nasazením do Preview (pro zpětnou vazbu od uživatelů).
- Verze Sandbox nabízí **Debug menu**, které poskytuje další funkce pro vývojáře rozšíření debugovat svoje aplikace pro Google Wave. Přidáním parametru `&ll=debug` do URL adresy vlny v prostředí aplikace Sandbox se zobrazí nová položka **Show log** v **Debug menu**.

S debugováním aplikací a způsobem práce ve vývojářské verzi Sandbox budeme seznámeni v kapitole popisující vlastní řešení ukázkových aplikací pro Google Wave.

4.2 Robot

Roboti představují účastníky vlny, kteří se mohou podílet na tvorbě obsahu, číst a editovat příspěvky a vkládat nejrůznější obsah. Mohou odstraňovat a přidávat účastníky, vytvářet nové blipy a nové wave dokumenty. Fyzicky roboti představují skripty, které běží na wave serveru. Tyto skripty komunikují s klientem Google Wave prostřednictvím protokolu Google Wave Robot Wire. Protokol je založen na posílání zpráv mezi robotem a serverem Google Wave. Robot inicializuje vyvolané události na serveru, dojde ke zpracování programu uvnitř robota a odešle operace, které se mají provést pomocí protokolu.



Obr. 16 Ukázka robotů

Pro vývoj robotů je k dispozici Google Wave Robot API složená z knihoven pro různé programovací jazyky a protokolu Robot Wire. V současné době jsou k dispozici dvě klientské knihovny pro programovací jazyky Python a Java. Právě tyto klientské knihovny mají za úkol zpracovávat a předávat příslušné zprávy, které jsou zasílány prostřednictvím Google Wave Robot Wire protokolu.

Dne 2. března 2010 byla vydána nová verze Google Wave Robot API nesoucí přídavné označení v2. Přinesla řadu vylepšení, zejména pak Google Wave Protocolu Wire a souvisejících klientských knihoven pro vývoj robotů. Dále aktualizovaná verze přináší řadu změn, především přidání nových funkcí, odstranění mnoha jazykových závislostí, optimalizace výkonu, přičemž logika funkcí zůstala zachována a je zpětně kompatibilní s předešlou verzí. Nejvíce změn bylo provedeno v klientské knihovně v jazyce Java. Výsledkem jsou dvě klientské knihovny. Jsou si velmi podobné jak do rozsahu, tak i do způsobu použití [17]. Google na svých stránkách informuje vývojáře o migraci kódu na verzi v2, protože zpětná kompatibilita bude zachována pouze do 30. června 2010. [18]

4.2.1 Google Wave Robot Wire

Robot protokol se skládá ze zpráv, které jsou zasílány mezi klientem a serverem prostřednictvím protokolu HTTP obsahující následující části:

- JSON⁵ Messages Bundles - obsahují události v podobě zpráv, jsou zasílány serverem robotovi.
- JSON-RPC⁶ - obsahem zpráv jsou operace, které jsou zasílány Robotem na Wave v reakci na událost.

⁵ JavaScript Object Notation (Javascriptový objektový zápis) - představuje formát hierarchického zápisu dat (datový formát), data jsou organizována v polích nebo agregátových objektech. JSON je zcela obecný datový formát, který lze využít v libovolném programovacím jazyce a je většinou využíván k přenosu dat v internetovém prostředí. [19]

⁶ JSON-RPC - je jednoduchý protokol zakódovaný v JSON, který definuje datové struktury a příkazy. Umožňuje obousměrnou komunikaci mezi serverem a klientem. [20]

Robot posílá zprávy obsahující operace do klienta Wave v reakci na nějakou událost. Robot tedy vykoná definované činnosti v reakci na tyto události zasláním operací na server Wave v podobě JSON-RPC zpráv. [16]

Messages Bundles

Jak je vidět z **př. 2** (tento kód je pouze ilustrační, původní zpráva obsahuje mnohem více atributů) JSON Message Bundles představuje datový formát obsahující potřebné informace pro komunikaci serveru a robota/aplikace. Zdrojový kód prezentuje zprávu, která reaguje na událost BLIP_SUBMITTED (Událost, která nastane při dokončení nového blipu ve vlně). Tyto zprávy obsahují většinou několik základních částí:

- **events** (události) - obsahují informace týkající se dané události. V této části jsou obsaženy informace o typu vyvolané události.
- **blipy/wavelet** - prvky obsahují informace o každé entitě (Vlnka, Blip), na které došlo k vyvolání události, přičemž některé mohou být navzájem propojeny v závislosti na typu události.
- **MetaData** - další informace, které mohou obsahovat prvek **events** např. **robotAddress**

Technologie Wave umožňuje vložit zprávy z několika událostí do jednoho elementu a posílat je jako jeden balíček. Informace uložené v JSON zprávách reflektují stav klienta, kdy došlo k události, čili neodrážejí stav, který se během té doby mohl změnit.

Příklad 2 JSON Message Bundle

```
1  {
2    "events": [{
3      "modifiedBy": "user@domain.com",
4      "timestamp": 1255935016481,
5      "type": "BLIP_SUBMITTED",
6      "properties": {
7        "blipId": "b+ja8F_Hw4J"
8      }
9    }],
10   "wavelet": {
11     "creationTime": 1255934856713,
12     "creator": "user@domain.com",
13     "lastModifiedTime": 1255935016481,
14     "participants": [ "user@domain.com", "user2@domain.com" ],
15     "rootBlipId": "b+ja8F_Hw4J",
16   },
17   "blips": {
18     "b+ja8F_Hw4J": {
19       "annotations": [{
20         "name": "conv/title",
```

```

23         "value": ""
24     },
25     "elements": {},
26     "blipId": "b+ja8F_Hw4J",
27 }
28 },
29 "robotAddress": "myrobot@domain.com"
30 }

```

Vložený zdrojový kód **př. 2** prezentuje událost BLIP_SUBMITTED vyvolanou při dokončení blipu. Obsahem této JSON zprávy jsou informace: jaký typ události nastal, v jakých částech dokumentu, tzn. informace v jakém blipu byla volána a informace o vlnce, která obsahuje tento blip. Součástí těchto hodnot jsou i informace, který z účastníku ji vyvolal, čas atd.

Operace

Při vyvolání události dojde ke zpracování definovaných instrukcí v metodě inicializující vyvolanou událost a odešle JSON-RPC zprávu na Wave server obsahující operace, jenž mají být provedeny v rámci vyvolané události. Následující zdrojový kód ukazuje příklad operace, který nastaví titulek vlny, vytvoří nový blip a vypíše do něj textový řetězec „Hello World“.

Příklad 3 JSON-RPC

```

1  [{
2      "params": {
3          "capabilitiesHash": "b31821e"
4      },
5      "id": "0",
6      "method": "robot.notifyCapabilitiesHash"
7  },
8  {
9      "params": {
10         "waveletId": "google.com!conv+root",
11         "waveId": "google.com!w+ja8F_Hw4g",
12         "waveletTitle": "A wavelet title"
13     },
14     "id": "op1",
15     "method": "wavelet.setTitle"
16 },
17 {
18     "params": {
19         "blipId": "b+ja8F_Hw4h",
20         "waveletId": "google.com!conv+root",
21         "waveId": "google.com!w+ja8F_Hw4g",
22         "blipData": {
23             "blipId": "TBD_google.com!conv+root_1",

```

```

24     "waveletId": "google.com!conv+root",
25     "waveId": "google.com!w+ja8F_Hw4g"
26   },
27 },
28 "id": "op2",
29 "method": "blip.createChild"
30 },
31 {
32   "params": {
33     "blipId": "TBD_google.com!conv+root_1",
34     "how": 0,
35     "waveId": "google.com!w+ja8F_Hw4g",
36     "text": "Hello World",
37     "waveletId": "google.com!conv+root"
38   },
39   "id": "op3",
40   "method": "document.modify"
41 }]
```

JSON zpráva se skládá z několika částí `params`. Tyto části obsahují metodu, kde každá z těchto metod obsahuje další pole `params` a odpovídající hodnoty, přičemž jednotlivé metody jsou seřazeny podle `id` viz **př. 3.** [16]

4.2.2 Google Wave Robot API

Robota lze vyvíjet pouze pomocí Google App Engine prostředí pro běh a správu aplikací [21]. Jestliže vytvoříme Robota pomocí Google App Engine, toto prostředí implicitně identifikuje aplikaci pomocí url adresy ve tvaru `aplikace.appspot.com`. V současné době musí být všichni roboti zavedeni pomocí App Engine, ale společnost Google dodává, že v budoucnosti bude možnost zavést aplikaci na jakýkoliv server a doménu. [22]

Události

V předchozí kapitole byla nastíněna komunikace robotů s prostředím skrze HTTP požadavků, kde jsou zasílány datové zprávy ve formátu JSON pomocí protokolu Google Wave Robot Wire. V této sekci bude popsán model událostí, které může robot registrovat, a data, která lze získat.

Google Wave Robot API je primárně řízen událostmi (*event-driven*). Události jsou registrovány roboty, na něž mohou prostřednictvím svých vlastních operací reagovat. Události představují činnosti vyskytující se v rámci vlny, u nichž je nutná registrace pro reakci na událost provedením definované akce. Je nutné vyčkat na vyvolání události, na kterou lze zpětně reagovat. API poskytuje několik částí pomáhající implementovat akce reagující na dané události:

- Každá API klientská knihovna definuje několik základních metod, pomocí kterých můžeme zachytit výskyt události a poskytnout reakci jako odpověď na výskyt této události.

- Každá vyvolaná událost je reprezentována jako objekt obsahující informace vlny, kde byla událost vyvolána. Lze využít výchozí obsah objektu reprezentující vyvolanou událost a obsah vlny, nebo toto implicitní chování zcela přepsat.

Události se vyskytují pouze na různých vrstvách Wave modelu, které vymezují prostor pro běh vestavěných funkcí.

- *Wavelet* - události jsou vyvolány na úrovni vlnky a obvykle mají omezený rozsah.
- *Blip* - události nastávají na úrovni blipu a vyskytují se mnohem častěji.

Speciálním typem události je událost dokumentu. Tato událost nastává vždy, pokud dojde ke změně informací v blipech nebo datových dokumentech ve vlně. Jedná se o nejobecnější událost v rámci vlnky. Příkladem může být událost `DOCUMENT_CHANGED`: nastane při každé změně textů, dat ve vlně. V tomto případě je třeba zvážit náročnost na přenos dat (JSON zprávy) při každé změně, proto je vhodné využít tzv. filtrování událostí (představuje novou funkci uveřejněnou až s Robot API v2). V **tab. 1** jsou uvedeny obslužné metody událostí použité v rámci kapitoly ukázkových aplikací. Kompletní seznam těchto metod jsou zveřejněny na stránkách Googlu. [23]

Událost	Metoda v klientské knihovně - Java	Popis
DOCUMENT_CHANGED	onDocumentChanged	nastane při každé změně dokumentu
BLIP_SUBMITTED	onBlipSubmitted	vyvolána při dokončení nového/editovaného blipu
WAVELET_SELF_ADDED	onWaveletSelfAdded	při vložení robota do vlny
GADGET_STATE_CHANGED	onGadgetStateChanged	při změně state objektu vloženého gadgetu ve vlně

Tab. 1 Přehled obslužných metod událostí

Klientská API definuje několik obslužných metod událostí (event handlers), které zachytí vyvolané události v rámci vlny. V jazyce Java lze zachytit událost přepsáním (override) obslužné metody události. Přidáním robota do vlny bude v klientu Google Wave vyvolána událost `WAVELET_SELF_ADDED`, která je zprostředkována pomocí Robot Wire protokolu a klientská knihovna zavolá obslužnou metodu události v robotovi.

Příklad 4

```

1  @Override
2  public void onWaveletSelfAdded(WaveletSelfAddedEvent event) {
3      event.getWavelet().reply("\nRobot byl pridán do vlny");
4      domain = event.getWavelet().getDomain();
5      log.info("DOMENA: " + domain);
6  }
```

Vložený zdrojový kód na **př. 4** demonstruje přepsání obslužné metody události `WAVELET_SELF_ADDED` vyvolané přidáním robota do vlny a vytvořením nového blipu s obsahem zprávy „Robot byl přidán do vlny“. Každá událost předává data v podobě objektu (objekt event). Pro každou úroveň (Wavelet, Blip), ve které je událost vyvolána obsahuje objekt různá data. Volání události na úrovni blipu obsahuje objekt události, identifikátor (ID) blipu, kde událost nastala. Na úrovni wavelet (vlnky) je součástí objektu odkaz na ID kořenového blipu viz obsah JSON zprávy v **př. 2**.

Operace

Operace v rámci Google Wave návrhu představují vzdálené volání procedur umožňující robotům inicializovat své akce v rámci vlnek. Operace v rámci Wave API mohou být buď *event-driven* (operace jsou spojené s podáním události v rámci systému Google Wave), anebo *aktivní* (robot provádí své vlastní akce). Aktivní roboti mohou využívat Active Robot API s uveřejněním nové verze Robot API.

V režimu řízeným událostmi (*event-driven*) obsahuje webový požadavek odeslaný prostřednictvím Google Wave informace o jedné nebo více událostí. Jestliže dojde k provedení události, robot může uskutečnit definované operace k provedení změn na vlně za podmínky, že je robot účastníkem vlnky, na které se změny provádí. Operace uskutečněné robotem na vlně nevyvolají další událost, nejsou zpět hlášeny do robota, tzn. jestliže robot aktualizuje informace v některém z blipu, nebude robot zpětně dostávat událost o změně dokumentu. Příkladem může být operace, při níž robot změní část textu v blipu. V tomto případě nebude webový klient zpětně odesílat událost `DOCUMENT_CHANGED`.

Operace jsou zpracovávány v pořadí v jakém byly doručeny na vlnu a provádí se asynchronně, tzn. neexistuje zde záruka, že další účastníci na vlně nemohou provádět své vlastní operace v uplynulém čase. Je tedy nutné počítat s tím, že operace, které budeme provádět na vlně, nemusejí být prováděny okamžitě. Vlna představuje historickou entitu uchováající jak obsah, tak i historii operací, které byly provedeny v rámci vlny.

Pro manipulaci s událostmi a následným prováděním operací jsou poskytnuty klient-ské knihovny pro programovací jazyky Java a Python. Například Java využívá obslužnou metodu události (*event handler*) reagující na požadavky Wave, převádí získaná data do objektů a potom vyvolá metodu na serveru při každé události. Operace jsou poskytovány metodou a manipulují s objekty reprezentujícími datový model Wave. [24]

4.3 Gadget

Gadgets představují externí aplikace třetích stran s vlastním uživatelským rozhraním. Umožňují rozšířit Google Wave klienta o nové vlastnosti a funkce. Tyto aplikace běží na straně klienta a jsou umísťovány do wave dokumentu (vln).

Hlavním charakteristickým rysem gadgetů je schopnost udržovat „stav“ aplikace, případně pracovat se změnami stavu. Lze si jej představit jako objekt s hodnotami sdílenými všemi účastníky na vlně. Stav v gadgetu je nazýván „společným stavem“, jelikož tento stav aplikace sdílejí všichni účastníci vlny. Nejenomže všichni účastníci sdílejí stejný stav aplikace, ale také každý z účastníků může kdykoliv změnit tento stav aplikace. Každý Wave Gadget udržuje objekt o stavu „state objekt“ aplikace, kterému lze přiřadit libovolný klíč a odpovídající hodnotu. Pomocí těchto klíčů a hodnot je možné se dotazovat



Obr. 17 Ukázka gadgetů

na aktuální stav gadgetu, provádět změny stavu nebo aktualizovat stav aplikace. Narozdíl od robotů není dovoleno Gadgetu zasahovat do prostředí vlny, ale pouze měnit svůj stav.

Správné fungování gadgetu je závislé na stavu a objektech účastníků. Tyto informace musí být k dispozici tehdy, když je k nim přístup. Pro správný provoz aplikace je důležité se při návrhu struktury gadgetu držet několika zásad.

- Většina aplikací pro Google Wave obsahuje inicializační funkci `init()`, je aktivována při prvním spuštění aplikace. Pro správné fungování je vhodné nepřistupovat ke stavu aplikace nebo k objektům účastníků pomocí této funkce. Hodnoty stavu nebo objektu nebudou mít vypovídající hodnotu, dokud se neprovedou volání (`setParticipantCallback()` a `setStateCallback()`).
- Při vytváření interaktivního gadgetu je vhodné vyvolat události pomocí UI prvků. Pro zachycení vstupu uživatele se využívá prvků, jako je např. tlačítko. Je doporučováno manipulovat se stavem aplikací právě při vyvolání událostí pomocí těchto UI prvků.
- Jestliže dojde ke změně stavu aplikace, dostanou účastníci oznámení o provedených změnách. Uvnitř volání `setParticipantCallback()` a `setStateCallback()` je vhodné umístit logickou část programu, která zajistí aktualizaci vlny. Tato volání jsou vyvolána, pokud dojde ke změně stavu, nebo ke změně účastníků na vlně.

V současnosti je poskytována JavaScript knihovna pro vývoj gadgetů. Gadget představují XML soubory. **Př. 5** prezentuje jednoduchou aplikaci (gadget), která vypíše „Hello World!“ do vlny a ilustruje základní syntaxi této specifikace pro tvorbu gadgetů. Prvním elementem je `<Module>`. Indikuje, že se jedná o gadget aplikaci. Dalším důležitým prvkem je `<Require feature="wave" />` inicializující gadget, který má přístup do aplikace Google Wave. `<![CDATA [...]]>` mezi tyto značky lze umístit jakýkoliv HTML, CSS nebo JavaScript kód. [25]

Příklad 5 Gadget

- 1 `<Module>`
- 2 `<ModulePrefs title="Gadget" height="120">`

```
3   <Require feature="wave" />
4 </ModulePrefs>
5 <Content type="html">
6   <![CDATA[

8   ]]>
9   </Content>
10 </Module>
```

4.4 Wave Embed API

Toto rozhraní umožňuje jednoduše a pohodlně vložit obsah vlny i se všemi jejími interaktivními vlastnostmi do jakékoliv webové stránky pro usnadnění komunikace a spolupráci uživatelů. Jelikož celá konverzace je uložena na jednom místě, budou veškeré změny, které budou provedeny na stránkách s nově vloženou vlnou, ihned reprezentovány na vlně sdílenou prostřednictvím Google Wave Preview klienta. Vložená vlna se chová stejně jako v klientu Google Wave Preview. Dokonce podporuje drag & drop funkci pro vkládání příloh a obrázků. Google Wave Embed API lze využívat přidáním JavaScriptové knihovny do našich webových stránek. [26]

5 Ukázkové aplikace

V rámci diplomové práce byly vytvořeny dvě rozšíření pro Google Wave demonstrující možnosti této nové platformy. První aplikací je rozšíření označované jako robot nesoucí název RawWaveRobot, který funguje na principu zadávání příkazů s hodnotami a slouží především pro výpočet jednoduchých i složitějších matematických výrazů, nebo pro zobrazení technických informací v prostředí vlny. Druhou aplikací pro Google Wave je gadget s názvem TaskList pro sdílení úkolů mezi účastníky na vlně, na které se lze přihlašovat, a umožňuje rozdělovat jednotlivé úkoly pro menší skupinu účastníků.

5.1 RawWaveRobot

Pro vývoj ukázkového robota byl využit software preferovaný společností Google a jako programové prostředí byla zvolena klientská knihovna pro jazyk Java. Pro tuto možnost vývoje společnost Google doporučuje využít vývojové prostředí Eclipse, pro něž byl uvolněn Google plugin. Google plugin obsahuje App Engine SDK a několik funkcí (testování a nahrávání aplikace na Google server) k tvorbě robota v Eclipse. Dále je nutné nainstalovat Java Development Kit verze 6 (JDK).

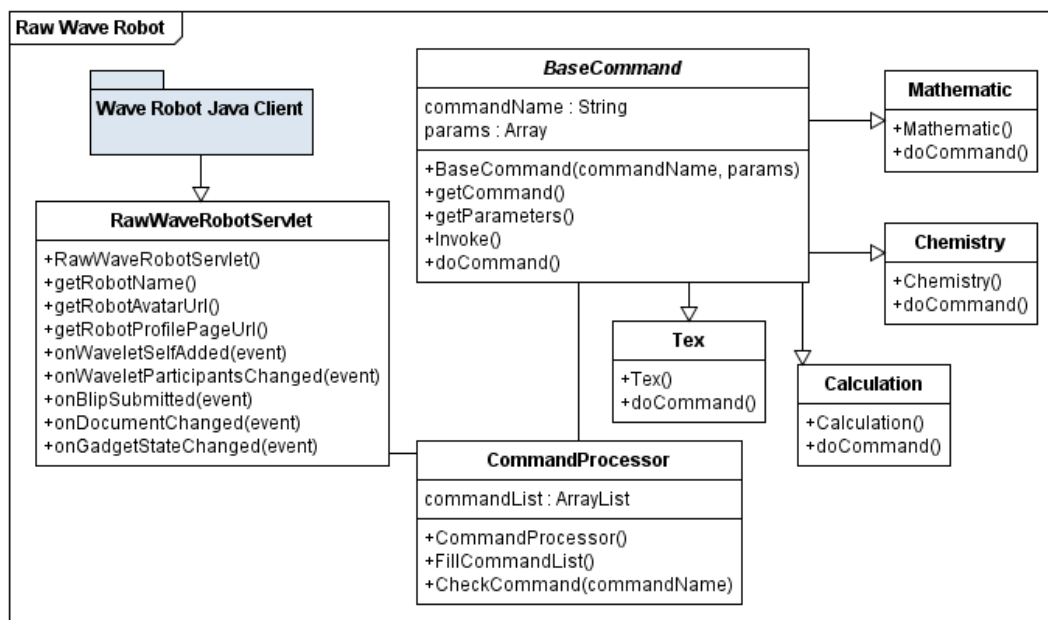
K tvorbě robota v aplikaci Eclipse, je nutná registrace aplikace na webové adrese <https://appengine.google.com> v App Engine, kde na základě registrace je získána ID aplikace. Toto ID je velmi nutné pro identifikaci aplikace v rámci Google Wave. V App Engine lze vytvořit maximálně 10 aplikací, po jejich registraci není umožněno měnit název, upravovat nebo smazat aplikaci. Při vytvoření nového robota v Eclipse se vygeneruje několik složek se soubory a skripty. Tento balík zatím neobsahuje klientskou knihovnu Java potřebnou pro vývoj robota v rámci Google Wave. Tuto knihovnu (je složena z několika souborů) je nutné ručně nakopírovat do složky obsahující knihovny robota a vytvořit odkazy v Eclipse. Celý tento proces je uveřejněn na stránkách Googlu s návodem na vytvoření primitivního robota [27].

RawWaveRobot demonstruje možnosti tvorby robotů, které lze implementovat do aplikace Google Wave s využitím klientské knihovny a několika jiných externích knihoven uvolněných pod open-source licenci. Robot dále implementuje propojení s jinými webovými stránkami pomocí jejich API rozhraní a zprostředkovává získané informace. Robot pracuje na principu zadávání příkazů, kde každý příkaz představuje nezávislou část, která může a nemusí být propojena s jinou částí nebo funkcí aplikace (robota). Každá část bude obsahovat identifikační příkaz a několik definovaných parametrů, kterými lze ovlivnit výstup informací do prostředí vlny. Úkolem aplikace je především umožnit převádění informací technického charakteru do prostředí vlny v čitelném formátu, nebo provádění jednoduchých i složitějších matematických výpočtů.

5.1.1 Návrh

Při návrhu robota bylo nutné zohlednit základní požadavek. Robot má fungovat na principu zadávání příkazů a několika volitelných parametrů. Očekává na začátku každého blipu definovaný specifický znak. V případě, že je zadáno jako první písmeno specifický znak, bude následující text ve vlně prezentován jako příkaz. Pokud tento znak není uveden, robot nebude dále nic zpracovávat. Všechny příkazy musí být aktivovány pomocí

prefixu „!“ kromě příkazu pro výpočet matematických výrazů, který bude určen pro zápis textového bloku obsahující jednotlivé matematické zápisy. Při této koncepci byla navržena struktura pomocných tříd, které lze vidět na **obr. 18**.



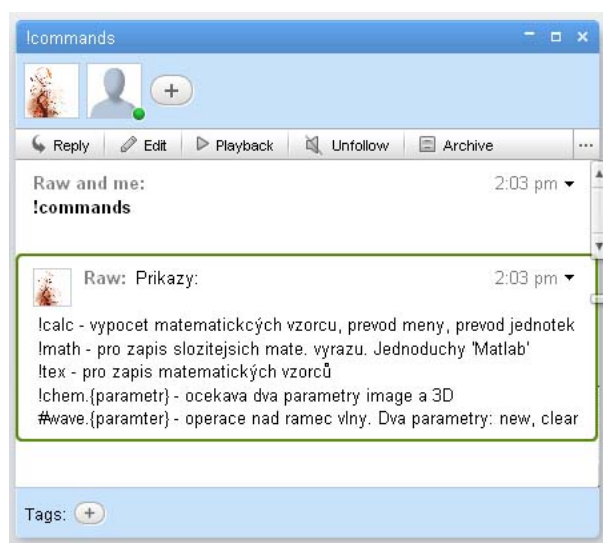
Obr. 18 UML schéma

Třída `RawWaveRobotServlet` představuje hlavní třídu, která dědí vlastnosti z abstraktní třídy `Robots` (součást klientské knihovny Java pro tvorbu robotů). V této třídě se definuje základní profil robota, kde lze přepisovat (Override) jednotlivé metody a nastavujeme základní informace aplikace (jméno robota v rámci vlny, avatar, případně odkaz na webovou stránku obsahující dokumentaci, nebo návod k použití). V této třídě se přepisují jednotlivé metody představující události (kapitola 4.2.2).

Další pomocnou třídou aplikace je třída `CommandProcessor`, která slouží k rozpoznání jednotlivých příkazů zadaných v blicích, a dle této hodnoty rozhodne, který příkaz bude vyvolán. Při inicializaci robota se v konstruktoru vytvoří seznam objektů (příkazů), které reprezentují jednotlivé příkazy. Pro tuto koncepci bylo nutné vytvořit abstraktní třídu `BaseCommand` reprezentující každý příkaz jako objekt. Třída obsahuje několik metod. Tou nejdůležitější metodou je abstraktní (protected) metoda `doCommand`, která je vždy přepsána metodou z objektu daného příkazu. Každý příkaz bude tedy reprezentován vlastní třídou, která bude dědit z abstraktní třídy `BaseCommand` a bude obsahovat metodu `doCommand`, která se provede v závislosti na daném příkazu.

5.1.2 Implementace

Ukázkový robot je přístupný na webové adrese <http://rawaverobot.appspot.com/> a jeho adresa v rámci Google Wave je ve tvaru `rawaverobot@appspot.com` (adresa představuje ID robota, které se vytvoří při registraci aplikace). Nejprve je nutné robota přidat do seznamu kontaktů uživatele v Google Wave a poté je možné ho přidat do vlny jako ostatní účastníky. Po přidání robota do vlny se vyvolá událost `WAVELET_SELF_ADDED`, kterou robot zpracuje a následně vypíše do vlny informace o této skutečnosti.



Obr. 19 Příkaz !commands

Aplikace funguje na principu zadávání příkazů. Na začátek každého blipu je napsán specifický znak „!“, který zaručí, že robot bude zpracovávat následující text za tímto znakem. Každý příkaz je očekáván v samostatném blipu, není tedy možné různě kombinovat příkazy v jednom blipu. Pro každý příkaz je nutné vytvořit nový blip, kde na začátku napíšeme specifický znak a jméno příkazu (případně další parametry nebo hodnoty pro daný příkaz). Příkaz spustíme ukončením editace blipu. V závislosti na typu příkazu dostaneme výstup buď v dalším blipu jako odpověď, nebo se výstup stane součástí blipu, kde byl zadán příkaz. Kompletní seznam příkazů, které robot umožňuje, lze vypsat jednoduchým příkazem `!commands` bez parametrů viz **obr. 19**.

V následující části jsou popsány jednotlivé příkazy a parametry, které lze zadávat v rámci robota. Každá hodnota, která má být zpracována pomocí příkazů je napsaná za název příkazu. Příkaz a hodnota musí být oddělené pevnou mezerou, aby došlo ke správné interpretaci příkazu. Všechny příkazy jsou zadávány v obecném tvaru: `!{prikaz}.{parametr} {hodnota}`.

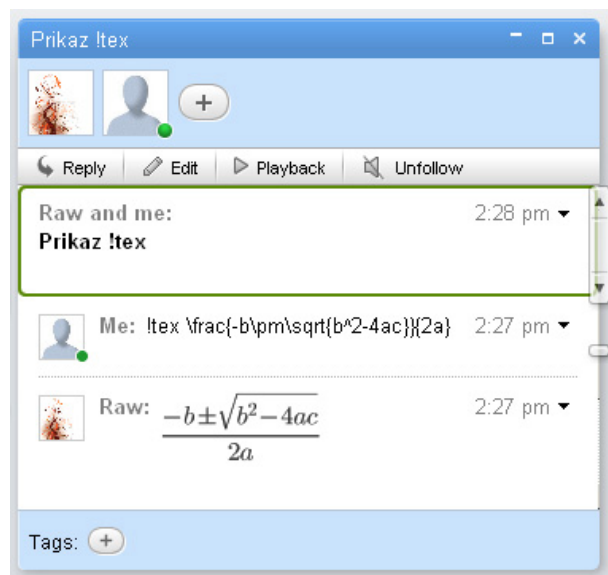
!chem

Tento příkaz slouží především pro získání informací z oboru chemie. Hlavní činností je zadání názvu chemické sloučeniny nebo chemického vzorce, který se v rámci vlny zobrazí jako chemický diagram nebo 3D model sloučeniny. Příkaz očekává jeden ze dvou možných parametrů, kterými lze ovlivnit výstup:

- `!chem.image` - tento parametr zajišťuje vložení obrázku s diagramem chemické sloučeniny
- `!chem.3D` - parametr vloží 3D model zadané sloučeniny. Funkce je přístupná díky propojení dvou různých typů rozšíření platformy Google Wave. Příkazem je zabezpečeno načtení externí JavaScriptové knihovny, která reprezentuje 3D model sloučeniny. Tato knihovna je poté vložena do gadgetu, jenž je schopen tuto knihovnu prezentovat v prostředí vlny a zobrazit 3D model sločeniny. Knihovna umožňuje různě natáčet tento model do různých úhlů, přibližovat model atd.

!tex

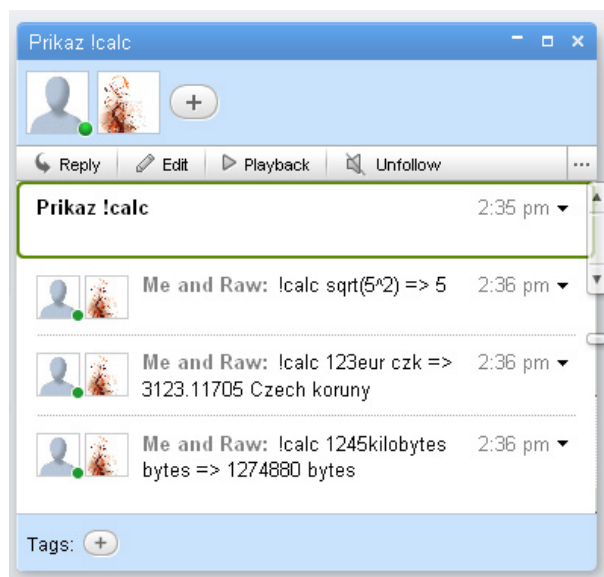
Příkaz je určen k zobrazení složitých i jednoduchých matematických vzorců, které se převedou pomocí sázecího systému $\text{\TeX}$ do čitelné podoby ve formě obrázku. Tento obrázek zadaného vzorce je vložen do prostředí vlny a lze jej kamkoliv přemísťovat. Vzorec, který převádíme v prostředí vlny, je zadáván syntaxí $\text{\TeX}$ u pro sazbu matematických výrazů. Jednoduchou ukázkou příkazu může být zobrazení vzorce pro výpočet kořenů kvadratických rovnic.

Obr. 20 Příkaz **!tex****!calc**

Pro výpočet jednoduchých matematických vzorců, převádění základních fyzikálních jednotek, nebo přepočet kurzu měny v prostředí vlny nám slouží příkaz **!calc**. Příkaz neočekává žádné parametry, ale pouze hodnoty. Výstup (spočtená hodnota) příkazu je zobrazen v blipu, kde byl zadán příkaz, výsledek se stává součástí blipu. Příkaz umožňuje 3 druhy výpočtů:

- **výpočet jednoduchých matematických výrazů** - při tomto způsobu výpočtu se očekává za jménem příkazu jednoduchý vzorec. Tento způsob výpočtu může představovat velmi jednoduchou kalkulačku v prostředí vlny. Příkaz pro výpočet matematického vzorce může vypadat následovně: **!calc sqrt(25) + 5 ^ 2**. Příkaz nám vypočítá hodnotu ze zadaného vzorce (sqrt zápis představuje odmocninu) a zobrazí v blipu.
- **převod fyzikálních jednotek** - příkaz je ve tvaru: **!calc 3443kilobytes bytes**. První hodnota je složena ze dvou částí. První část představuje číslo reprezentující počet daných jednotek a za tímto číslem jsou specifikovány jednotky (tyto části nesmí být odděleny mezerou). Druhá hodnota očekává zápis jednotky, ve které má být daný výpočet proveden.

- **výpočet kurzu měny** - na stejném principu zápisu jako u převodu jednotek, funguje převod měny. Příkaz je opět složen ze dvou hodnot, přičemž první obsahuje číslo (jednotky) a měnu, druhá hodnota reprezentuje vypočtenou měnu. Příklad výpočtu kurzu měny: `!calc 100eur czk`.

Obr. 21 Příkaz `!calc`

!math

Příkaz `!math` a jeho syntaxe zápisu se odlišuje od výše popsáných příkazů. Slouží také jako příkaz `!calc` pro výpočet matematických vzorců, nicméně umožňuje jednotlivé výpočty ukládat do proměnných pro pozdější výpočty, nebo zapisovat posloupnost příkazů. Součástí příkazu `!math` je možnost vytvářet textový blok obsahující posloupnost příkazů, který může představovat jednoduchý skript v prostředí Google Wave. Pomocí příkazu `!math` lze v prostředí vln počítat: základní matematické operace, matice, vektory, komplexní čísla, goniometrické funkce.

Zápis syntaxe je založen na určení textového bloku, ve kterém se očekávají různé matematické příkazy. Každý takový blok musí na začátku každého blipu obsahovat následující posloupnost znaků: `##!math`. Na konci je nutné uvést uzavírací znaky `##` určující konec bloku. Pro správné interpretování zápisu je nutné, aby na konci každého blipu byly obsaženy uzavírací znaky (za těmito znaky se už nesmí vyskytovat další znak). Každý příkaz zapsaný v bloku musí být ukončen středníkem. Následující **př. 6** prezentuje jednoduchý skript pro výpočet kořenů kvadratické rovnice.

Příklad 6

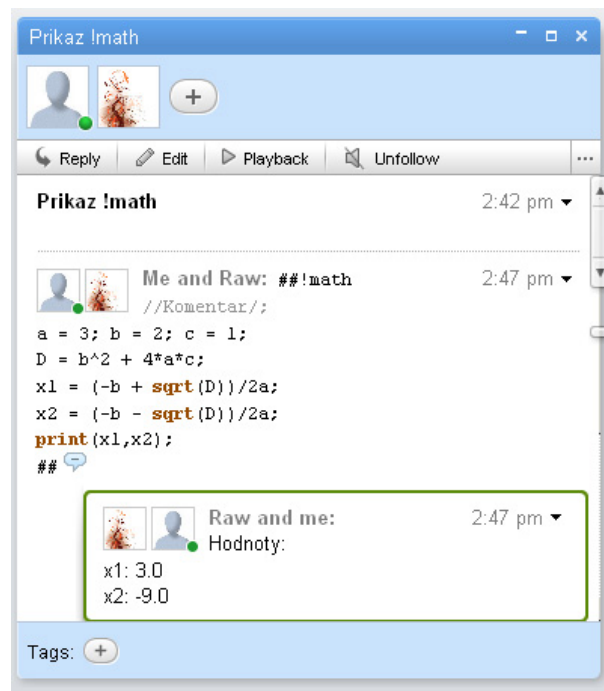
```

1  ##!math
2  //Koeficienty /;
3  a = 3; b = 2; c = 1;
4  // Vzorecky pro vypocet kvadraticke rovnice /;
5  D = b^2 + 4*a*c;
6  x1 = (-b + sqrt(D))\\2a;
```

```

7  x2 = (-b - sqrt(D))\\2a;
8  // Vystup. Pokud není nastaven, zobrazí se všechny proměnné. /;
9  print(x1,x2);
10 ##

```



Obr. 22 Příkaz !math

Součástí této funkce je zobrazování barevné syntaxe během zadávání matematických výrazů. Jakmile uživatel umístí základní posloupnost znaků `##!math##` určující blok příkazů do nového blipu, robot rozpozná tyto znaky a nastaví strojové písmo vhodné pro zápis skriptu (toto chování probíhá v editačním režimu). Syntaxe podporuje obarvení komentářů a základních matematických funkcí viz **př. 6** a **obr. 22**.

V bloku, který obsahuje několik zadaných matematických výrazů, je definována (volitelná) funkce, která má za úkol zobrazit požadované hodnoty, které jsou umístěny jako argumenty. Příkladem je zapsání příkazu `print(promenna1, promenna2, ...)` na konec skriptu, který zajistí zobrazení definovaných hodnot v argumentu funkce. Pokud tato funkce nebude definována ve skriptu, výstupem skriptu bude kompletní slovník uložených hodnot (při každé inicializaci skriptu jsou do slovníku uloženy některé hodnoty: Ludolfovo číslo π , Eulerovo číslo, komplexní číslo).

!wave

Všechny zmíněné příkazy demonstrovaly operace robota provedené v rámci vlny. Příkaz **!wave** demonstruje možnost přistupovat k funkcím nad rámec operací ve vlně. Tato možnost Active Robot API je přístupná až s uveřejněním Google Wave Robot API v2. Takovým příkladem využití může být vytvoření nové vlny zadáním příkazu v blipu. Pro využití Active Robot API je nutné ověřit robota u společnosti Google. Ověřovací proces vytvoří OAuth token - soukromý zašifrovaný klíč, který slouží k ověření pravosti robota a jeho požadavků. Příkaz **!wave** obsahuje dva parametry:

- `#wave.new` - tímto parametrem lze vytvořit novou vlnu s předdefinovanými spolupracovníky. Nová vlna bude obsahovat seznam účastníků, kteří byli obsaženi ve staré vlně, kde došlo k provedení příkazu.
- `#wave.clear` - tento parametr zajistí, že všechny blipy (kromě kořenového blipu) obsažené ve vlně budou smazány.

5.1.3 Testování

Je potřeba zmínit, že funkčnost robotů aplikace Google Wave nelze testovat lokálně na počítači, nýbrž je zapotřebí každý napsaný kód nebo pozměněnou část programu nahrát pomocí např. Eclipse do App Engine a poté spustit v některém z webových prostředí Preview nebo Sandbox. Při registraci robota pomocí App Engine je získáno jedinečné ID k identifikaci robota v rámci Google Wave. Toto prostředí nabízí užitečné funkce, především pro ladění kódu.

Pokud je vyvolána ve webovém prostředí Google Wave nějaká událost, kterou robot inicializuje, tak komunikace mezi serverem a robotem bude zaznamenána v logu v App Engine. Logy obsahují komunikaci v podobě JSON zpráv (4.2.1) mezi robotem a klientem Google Wave. S využitím nové funkce v Robot API v2 je možné také logovat námi definované operace a hodnoty některých proměnných. V tomto výpisu komunikace mezi serverem a robotem jsou vypsány i nezachycené výjimky v programu.

Prostředí App Engine nabízí kompletní správu aplikace v rámci celé sítě Google. Kromě výpisů jednotlivých logů při výskytu události, které robot inicializuje v Google Wave, nabízí toto prostředí další velice užitečné funkce pro správu aplikace, např. historie nahrávání jednotlivých verzí aplikace, náročnost aplikace na vytížení procesoru a paměti. Další funkcí, která stojí za zmínku je vedení statistických dat komunikace mezi serverem a robotem; jsou zapisovány hodnoty jako velikost přenesených dat, počet požadavků, využití rozhraní pro databáze atd. Společnost stanovila denní limity jednotlivých položek pro komunikaci mezi serverem a robotem. Pokud bude překročen některý ze stanovených limitů, dojde k pozastavení aplikace. Všechny limity jsou resetovány každých 24 hodin.

5.2 Gadget

Ukázkový gadget představuje externí aplikaci vybudovanou pomocí značkovacích jazyků HTML, CSS, XML a skriptovacího jazyka JavaScript. Aplikace je primárně určena k zadávání úkolů pomocí jednoduchého formuláře a následného zobrazení seznamu všech úkolů v prostředí vlny. Všichni účastníci sdílejí stejný aktuální seznam úkolů. Prostřednictvím aplikace se lze na tyto úkoly přihlašovat nebo odhlašovat (zobrazit seznam přihlášených osob), zadávat jim různou prioritu, nebo označit úkol jako dokončený.

Další užitečnou funkcí gadgetu je jednoduché a parametrické vyhledávání podle jména zadavatele úkolu a priority. Při vyhledávání lze tyto dva způsoby kombinovat mezi sebou. Jednoduché vyhledávání porovnává pouze jednoduché textové řetězce v attributech: nadpis úkolu, popis úkolu. Parametrické vyhledávání je založeno na dvou parametrech, kterými lze zobrazit užší výběr úkolu splňující některá kritéria.

5.2.1 Návrh

Gadget lze psát v kterémkoliv editoru pro psaní zdrojových kódů, aplikace obsahuje základní soubor XML, který má předem definované atributy a mezi speciální znaky lze umístit HTML, CSS a JavaScript kód. Logická část aplikace je napsána v jazyce JavaScript komunikující s Gadgets API. XML soubor představující aplikaci musí být veřejně dostupný z internetového prostředí pomocí URL. Aplikaci lze hostovat v App Engine nebo pomocí jiných hostingových služeb.

Gadgets API v prostředí vln umožňuje uchovávat tzv. state (více **kap. 4.3**). Pro tuto koncepci lze do tzv. „state objektu“ ukládat různé kombinace klíčů a hodnot. Hodnoty mohou představovat různé datové typy: textové řetězce, čísla nebo dokonce objekty. Gadgets API umožňuje pracovat s tímto state objektem, dotazovat se na různé klíče a hodnoty, vkládat, editovat data a implementovat vlastní volání, která se provedou při změně stavu objektu.

Gadget bude pracovat na principu vkládání úkolů s různými informacemi. Proto je nutné navrhnout datovou strukturu tak, aby všechny potřebné informace byly vždy dostupné ve state objektu. Dalším požadavkem na uchovávání informací o seznamu úkolů je zajistit kontinuitu dat a správnost přiřazených atributů k jednotlivým úkolům.

Na základě požadavků byla navržena struktura dat v podobě JSON objektu obsahující jednotlivé informace v hierarchické struktuře. Díky tomuto návrhu lze velice jednoduše uchovávat větší množství informací k jednotlivým úkolům a řadit je podle identifikačního čísla v JSON objektu, kde si každý úkol nese stejné atributy, ale s jinými hodnotami. Ukázka state objektu z gadgetu se seznamem dvou úkolů, viz **př. 7**.

Příklad 7 State objekt

```
1  {"tasklist":
2    "{
3      1271866494495: {
4        name: "Ukol cislo 1",
5        creator: "michlik@wavesandbox.com",
6        participants: ["user@domain.com", "user2@domain.com"],
7        description: "Popis ukolu",
8        priority: "1",
9        deadline: "za 3min"
10     },
11     1271866502371: {
12       name: "Ukol cislo 2",
13       creator: "michlik@wavesandbox.com",
14       participants: ["user3@domain.com"],
15       description: "Popis ukolu",
16       priority: "1",
17       deadline: "za 3min"
18     }
19   }"
20 }
```

State objekt obsahuje jednu definovanou proměnnou pojmenovanou tasklist a její hodnotu v podobě JSON objektu. Dle vloženého kódu lze vidět jednoduchý datový formát celého state objektu. Hodnotou této proměnné bude reprezentovat Objekt v JSON obsahující seznam úkolů (každý úkol je identifikován jedinečným číslem) a seznamem jejich atributů. Pro tuto koncepci návrhu je možné zajistit efektivní uchovávání dat a správné prezentování informací v gadgetu.

5.2.2 Implementace

Gadget lze vložit do vlny pomocí tlačítka „puzzle“, které je přístupné v editačním režimu na editační liště. Po kliknutí na tlačítko se zobrazí pop-up okno s formulářovým políčkem pro zadání úplné webové adresy aplikace. Ukázkový gadget je hostován na webové adrese: <http://michlik.cz/wave/tasklist/app.xml>. Druhou možností, jak vložit gadget do vlny, je pomocí ukázkového robota RawWaveRobot, který skrývá další příkaz **!gadget** (bez parametrů a hodnot). Zadáním příkazu do blipu bude automaticky tento gadget vložen do vlny a připraven k použití.

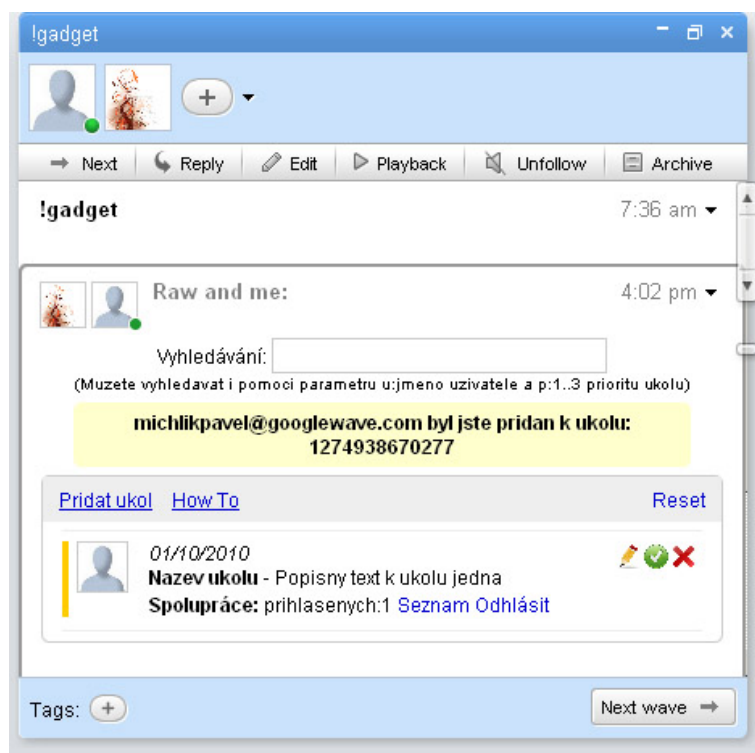


Obr. 23 Ukázkový gadget TaskList

Po vložení gadgetu do vlny se zobrazí rámeček s horní šedou lištou obsahující několik příkazů. Na liště je umístěn odkaz k vytvoření úkolu. Po kliknutí na něj se zobrazí formulář obsahující několik atributů pro zadání úkolu. Atributy úkolu jsou: název, popis, priorita a čas, kdy má daný úkol skončit. Atribut priorita při zadávání nového úkolu umožňuje nastavit rozsah (nastavujeme číslo od jedné do tří) důležitosti daného úkolu. Čím vyšší číslo je nastaveno, tím je daný úkol důležitější. Po vyplnění jednotlivých atributů je možné vložit úkol do state objektu. Pokud některé položky nejsou vyplněny (jsou prázdné), aplikace na to upozorní účastníka zobrazením chybové zprávy nad hlavní lištou. Všechny chybové nebo informující zprávy v rámci aplikace jsou zobrazovány nad hlavní lištou gadgetu, které se zobrazí po dobu 5 sekund. Po uplynutí této časové doby se informující zpráva přestane zobrazovat.

V případě, že je několik vytvořených úkolů v aplikaci a účastník je autorem alespoň jednoho z nich, máme přístup k základním metodám operujících s daným úkolem. Tyto metody jsou přístupné pouze autorovi úkolu, ostatní účastníci na vlně nemají přístup k

těmto metodám. Vpravo nahoře každého úkolu se nachází tři obrázky reprezentující tlačítka jednotlivých metod. První metodou je editace příspěvku, kdy lze změnit některé atributy úkolu jako např. nadpis úkolu nebo popis úkolu. Druhým tlačítkem je nastavení úkolu jako „dokončený“, kdy tento stav se projeví nastavením šedé barvy písma a přeškrtnutým textem v úkolu. Pokud je úkol označen jako „vyřešený/dokončený“, nelze se na něj přihlašovat nebo provádět jakékoliv operace, dokud autor úkolu označí úkol jako nedokončený. Poslední možností je smazat úkol se všemi dosavadními informacemi.

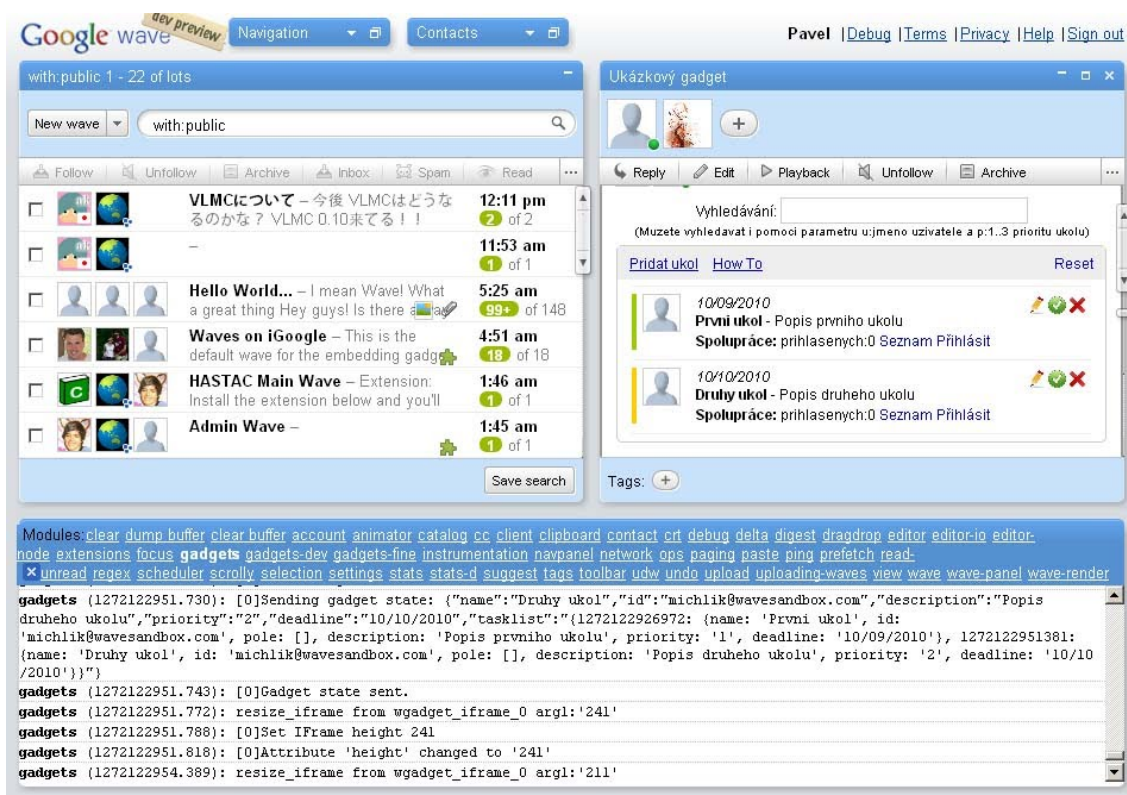


Obr. 24 Zobrazení informující/chybové zprávy

Každý úkol obsahuje sekci pro spolupráci na daném úkolu. Součástí této sekce je informace o počtu přihlášených spolupracovníků v úkolu, možnost přihlásit se na daný úkol; přístupná je i opačná funkce - možnost odhlásit se z úkolu (tato možnost je přístupná pouze v případě, že jste se na daný úkol přihlásili). Dále se nabízí zobrazit seznam přihlášených uživatelů na daný úkol, pokud je tento seznam prázdný, opět budeme informováni chybovou zprávou nad hlavní lištou aplikace (na **obr. 24** lze vidět informující zprávu po přihlášení na úkol). Všechny zmíněné funkce v rámci sekce spolupráce jsou přístupné pro všechny účastníky na vlně.

Nad rámečkem se nachází vyhledávací pole pro zadávání klíčových slov. Vyhledávat lze dvěma způsoby. Prvním možností je zadání klíčového slova nebo textového řetězce pro získání úkolů obsahující tato slova v nadpisu nebo popisu úkolu. V případě, že bude zadána posloupnost znaků do vyhledávacího pole, budou všechny znaky ihned reprezentovány pro zobrazení úkolů obsahující daný řetězec. Druhým způsobem je definovat parametry, podle kterých je zobrazen užší výběr úkolů. Prvním parametrem u parametrického vyhledávání je u (jména uživatelů) a druhým z nich je p (priorita úkolu). Zadávání probíhá napsáním parametru, znaku dvojtečky za parametrem a hodnoty, která je hledaná. Příkladem může být následující zápis: `u:uživatel p:1 text`. Příklad ukazuje propojení obou způsobů

vyhledávání. Tímto zápisem jsou vyhledávány úkoly, které vytvořil definovaný uživatel s prioritou úkolu 1 obsahující zadaný text.



Obr. 25 Google Wave Sandbox - Debug log

5.2.3 Testování

Pro testování gadgetu je vhodné využít webové rozhraní Google Wave Sandbox určené především pro vývojáře aplikací Google Wave. Sandbox obsahuje rozšiřující funkci pro ladění externích aplikací. Pomocí této aplikace lze sledovat kombinací klíčů a hodnot uložených ve state objektu, případně jenom definované sledované proměnné.

Funkce je zpřístupněna přidáním parametru `?ll=debug` do URL adresy vlny. Při otevření vlny se zobrazí na adresním řádku prohlížeče adresa vlny:

```
../#restored:wave:wavesandbox.com!w%252BKxhtJaIkK
```

parametr pro zpřístupnění výpisu state objektu umístěný do url před začátek textového řetězce `#restored`:

```
../?ll=debug#restored:wave:wavesandbox.com!w%252BKxhtJaIkK
```

Po úpravě adresy je nutné obnovit webovou stránku, která zpřístupní nové možnosti v nabídce **Debug** umístěného v pravém horním rohu aplikace. V této nabídce se objeví nová položka **Show debug log**. Po kliknutí se objeví zcela nový nezávislý panel ve spodní

části aplikace Google Wave viz **obr. 25**. Díky tomuto panelu lze sledovat výpis klíčů a hodnot uložených ve state objektu.

Během vývoje a testování ukázkových aplikací v prostředí Google Wave Preview a Sandbox někdy docházelo ke zpomalení odezvy provedených akcí. Během používání aplikace se také stávalo, že došlo k nadměrnému vytížení procesoru, což mohlo mít za následek ne zrovna dobré implementaci objektu `<canvas>` v návrhu specifikace HTML5. Problém mohl být také způsoben výkoností JavaScriptového kódu.

Jako další nevýhodu Google Wave lze považovat debugování aplikací (robotů). Dosud nebyla uveřejněna knihovna pro testování těchto typů aplikací na lokálním počítači. Je nutné každou změnu kódu aktualizovat pomocí App Engine, což představuje nemalé zdržení při vývoji a především ladění aplikací.

Opět není možné druhý typ rozšíření (gadget) debugovat lokálně, ale až vložím do prostředí vlny. Při testování gadgetů, kde logická část aplikace je napsaná pomocí JavaScriptu, může špatně napsaný kód vést ke zpomalení celé aplikace nebo až k pádu webového prohlížeče. Proto je dobré se držet zásad pro vývoj tohoto typu rozšíření.

Na druhou stranu jako výhodu technologie Google Wave lze považovat podrobný popis API rozhraní a dokumentaci vývojářských knihoven určených pro vývoj aplikací. Zejména pak ochotu vývojařů odpovídat na jednotlivé otázky týkající vývoje aplikací a provádět v krátké době opravy klientských knihoven při nahlášení chyby.

6 Závěr

Předmětem této diplomové práce bylo popsat zcela novou technologii Google Wave od společnosti Google, která představuje evoluci ve vývoji webových aplikací a hlavně změnu ve vnímání tvorby obsahu v internetovém prostředí. V této práci jsem se zaměřil na popis jednotlivých technologií tvořící základní stavební kameny technologie Google Wave. Zabýval jsem se popisem technologií Google Wave, jak na síťové vrstvě, tak i na prezentační vrstvě. Z hlediska síťové vrstvy především popisem zcela nového konceptu sdílení dat a nového protokolu Google Wave Federation Protocol, který je určen pro práci s vlnami zaručující okamžitou aktualizaci vlny při provedení změny na vlně. Součástí práce je také popis webové aplikace Google Wave Preview, která demonstruje využití této technologie v praxi a umožňuje širší veřejnosti vyzkoušet si nové vlastnosti a možnosti této technologie.

V předposlední části diplomové jsem se zaměřil na seznámení s platformou Google Wave. Uveřejněním této technologie byla představena Google Wave API rozhraní, která umožňují vývojářům rozšiřovat webovou aplikaci Google Wave Preview o další užitečné vlastnosti a funkce. Popsal jsem jednotlivá rozšíření komunikační platformy a jejich následná API rozhraní potřebná pro vývoj těchto aplikací. Součástí je i popis vývojového prostředí Google Wave Sandbox, kde lze tyto aplikace testovat a debugovat před nasazením do aplikace s označením Preview.

V závěrečné části diplomové práce jsem vytvořil dvě ukázkové aplikace využívající Google Wave API rozhraní, jejímž úkolem bylo představení možnosti využití této komunikační platformy a vysvětlení základních principů chování rozšiřujících aplikací.

V první ukázkové aplikaci jsem vyvinul robota určeného pro psaní technických dokumentů v prostředí vlny a umožňujícího zadávat matematické vzorce pro výpočet hodnot. Tento robot je především určen pro převádění technických dat nebo řešení problémů v rámci prostředí vln. Robot funguje na principu zadávání příkazů, kde každý příkaz představuje dílčí problém. Pro získávání dat využívá jiná webová API rozhraní, díky kterým lze velmi pohodlně převádět a zobrazovat data do prostředí vln v Google Wave. Aplikace implementuje některé open-source knihovny v jazyce JAVA pro výpočet složitějších matematických výpočtů, pomocí kterých bylo vytvořeno jednoduché prostředí pro běh jednoduchých matematických skriptů v rámci vlny. Další vlastností je propojení robota s gadgetem. Díky tomuto propojení lze vkládat 3D diagramy chemických sloučenin do prostředí vlny. Tento ukázkový robot demonstruje převážnou část Google Wave Robot API v2 uvolněnou pro tvorbu robotů.

V rámci práce jsem vytvořil další aplikaci (gadget) s názvem TaskList, která demonstruje jinou možnost rozšíření komunikační platformy Google Wave. Aplikaci jsem navrhnul s ohledem na předepsané postupy pro realizaci gadgetu v prostředí wave. Cílem gadgetu je umožnit účastníkům na vlně rozdělení dílčích definovaných úkolů, které je potřeba v rámci skupiny účastníků řešit. Aplikace nabízí jednoduché rozhraní pro vkládání úkolů s několika atributy (jméno úkolu, popis úkolu, priorita, čas, kdy se má úkol vyřešit) pro zobrazení všech úkolů v rámci aplikace. U každého úkolu je zobrazena fotografie účastníka, který daný úkol zadal. Součástí gadgetu jsou také funkce pro editaci a mazání jednotlivých úkolů. Tyto funkce jsou zpřístupněny pouze autorovi úkolu. Jako další možnosti aplikace jsem navrhl funkci pro spolupráci. Tato aplikace poskytuje možnost přihlášení na úkol jako spolupracovník. Užitečnou funkcí aplikace je parametrické a

jednoduché vyhledávání, které umožňuje jednoduše filtrovat úkoly dle definovaných parametrů a to především podle autora úkolu nebo priority úkolu. Pořadí parametrů lze mezi sebou kombinovat.

Při úvodním představení širší veřejnosti vzbudila technologie Google Wave obrovský zájem mezi uživateli. Z prvních prezentací této technologie nebylo možné určit, k čemu daná webová aplikace má sloužit či dokonce jaké vlastnosti a funkce bude obsahovat. Až po uvolnění aplikace Google Wave Preview se naskytla možnost testování nové technologie. Google Wave byla s nadsázkou prezentována jako technologie, která nahradí dnešní běžné komunikační možnosti, a to především e-mail a instant messaging.

V současné době se tato technologie stále nachází ve fázi vývoje, některé vlastnosti a funkce stále obsahují chyby, nebo dokonce schází. Přes nedostatek některých vlastností nabízí tato technologie obrovské možnosti využití webových aplikací, zejména při vzájemné spolupráci uživatelů a tvorbě společného obsahu. Především pro menší počet účastníků na vlně nabízí tato technologie obrovský potenciál.

Ukázkové aplikace, které jsem vytvořil, lze využít v rámci studia na fakultě strojního inženýrství, oboru automatizace a informatiky, aplikace Google Wave pro flexibilní řízení projektů, oraganizování výuky se studenty nebo pohodlné získávání dat z prostředí vlny. Implementace navržených poznatků v této diplomové práci mohou sloužit pro řešení úkolů v rámci řízení projektů, organizaci rozsáhlých akcí nebo zkvalitnění vlastního studijního procesu. Dalším příkladem využití této aplikace v rámci výuky je použití prostředí pro vývoj a běh aplikací. Může sloužit pro studium objektově orientovaného programování (OOP) s využitím připravených klientských knihoven za účelem seznámení studentů s vyvíjením nových vlastností a funkcí v koncepčně složitějších aplikacích. Díky uvolnění API rozhraní je možné vyvíjet do prostředí speciální vlastnosti a funkce potřebné při řešení speciálních úkolů. Z hlediska sdílení a výměny informací mezi uživateli aplikace Google Wave představuje nenahraditelného pomocníka.

Literatura

- [1] DINUCCI, Darci. *Fragmented Future* [online]. 1999, [cit. 2010-03-16]. Dostupné z <http://www.cdinucci.com/Darcy2/articles/Print/Printarticle7.html>.
- [2] WIKIPEDIA. *Web 2.0* [online]. 2010, [cit. 2010-03-16]. Dostupné z http://en.wikipedia.org/wiki/Web_2.0.
- [3] O'REILLY, Tim. *What Is Web 2.0?* 2005, [cit. 2010-03-16]. Dostupné z <http://www.oreilly.de/artikel/web20.html>.
- [4] RASMUSSEN, Lars. *Went Walkabout. Brought back Google Wave* [online]. 2009, [cit. 2010-03-16]. Dostupné z <http://googleblog.blogspot.com/2009/05/went-walkabout-brought-back-google-wave.html>.
- [5] GOOGLE. *Google Wave API* [online]. 2009, [cit. 2010-03-16]. Dostupné z <http://code.google.com/intl/cs-CZ/apis/wave/guide.html>.
- [6] LASSEN, Soren. THOROGOOD, Sam. *Google Wave Federation Architecture* [online]. 2010, [cit. 2010-04-22]. Dostupné z <http://www.waveprotocol.org/whitepapers/google-wave-architecture>.
- [7] WRANG, David. MAH Alex. *Google Wave Operational Transformation* [online]. 2010, [cit. 2010-04-22]. Dostupné z <http://www.waveprotocol.org/whitepapers/operational-transform>.
- [8] GOOGLE. *Wave Protocol* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://code.google.com/p/wave-protocol/>.
- [9] WHATWG. *Web Applications* [online]. 2004, 4th of June 2004 [cit. 2010-03-16]. Dostupné z <http://lists.whatwg.org/htdig.cgi/whatwg-whatwg.org/2004-June/000005.html>.
- [10] WHATWG. *WHATWG Mailing List* [online]. 2009, [cit. 2010-03-16]. Dostupné z <http://lists.whatwg.org/pipermail/whatwg-whatwg.org/2009-October/023849.html>.
- [11] W3C. *W3C HTML5* [online]. 2009, [cit. 2010-03-16]. Dostupné z <http://www.w3.org/TR/html5/>.
- [12] WHATWG. *WHATWG FAQ* [online]. 2009, [cit. 2010-03-16]. Dostupné z http://wiki.whatwg.org/wiki/FAQ\#When__will__HTML5__be__finished.3F.
- [13] WIKIPEDIA. *Document Object Model* [online]. 2007, [cit. 2010-04-22]. Dostupné z http://en.wikipedia.org/wiki/Document_Object_Model.

-
- [14] W3C. *HTML5 differences from HTML4* [online]. 2010, 10th of March 2010 [cit. 2010-04-22]. Dostupné z <http://dev.w3.org/html5/html4-differences/>.
 - [15] TRAPANI, Gina. *The Complete Guide to Google Wave* [online]. 2009, [cit. 2010-03-16]. Dostupné z <http://completewaveguide.com/>.
 - [16] GOOGLE. *Google Wave Robot Wire* [online]. 2009, [cit. 2010-03-26]. Dostupné z <http://code.google.com/intl/cs-CZ/apis/wave/extensions/robots/protocol.html>.
 - [17] GOOGLE. *Introducing Robots API v2: The Rise of Active Robots* [online]. 2009, [cit. 2010-03-26]. Dostupné z <http://googlewavedev.blogspot.com/2010/03/introducing-robots-api-v2-rise-of.html>.
 - [18] GOOGLE. *Google Wave Robots API: Overview* [online]. 2009, [cit. 2010-03-26]. Dostupné z <http://code.google.com/intl/cs-CZ/apis/wave/extensions/robots/index.html>.
 - [19] WIKIPEDIA. *JavaScript Object Notation* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://cs.wikipedia.org/wiki/JSON>.
 - [20] WIKIPEDIA. *JavaScript Object Notation* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://en.wikipedia.org/wiki/JSON-RPC>.
 - [21] GOOGLE. *Google App Engine* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://code.google.com/intl/cs/appengine/>.
 - [22] GOOGLE. *Google App Engine* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://code.google.com/intl/cs/apis/wave/extensions/robots/basics.html>.
 - [23] GOOGLE. *Google Wave Robots API: Events* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://code.google.com/intl/cs/apis/wave/extensions/robots/events.html>.
 - [24] GOOGLE. *Google Wave Robots API: Operations* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://code.google.com/intl/cs/apis/wave/extensions/robots/operations.html>.
 - [25] GOOGLE. *Wave Gadgets Tutorial* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://code.google.com/intl/cs/apis/wave/extensions/gadgets/guide.html>.
 - [26] GOOGLE. *Wave Embed Developer's Guide* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://code.google.com/intl/cs-CZ/apis/wave/embed/index.html>.
 - [27] GOOGLE. *Google Wave Robots API: Java Tutorial* [online]. 2010, [cit. 2010-03-26]. Dostupné z <http://code.google.com/intl/cs/apis/wave/extensions/robots/java-tutorial.html>.