

Google™ wave

Google Wave

Marcin Koziński

Seminarium „Systemy rozproszone” na Wydziale MIM UW

8 października 2009

Spis treści

- 1 Wprowadzenie
- 2 Google Wave API
- 3 Federation Protocol

Spis treści

- 1 Wprowadzenie
 - Historia
 - Co to jest Google Wave?
 - Jak wygląda?
- 2 Google Wave API
- 3 Federation Protocol

Parę lat temu...

Google Maps

[Google](#)
[Web](#)
[Images](#)
[Video](#)
[News](#)
[Maps](#)
[more »](#)

Stanford, California, United States KTH-hallen, Stockholm, Sweden

Maps

29	Take exit 24 A-B-C on the left toward I-93 N/Concord NH/S Station/I-93 S/Quincy	0.4 mi
30	Merge onto Atlantic Ave	0.8 mi
31	Turn right at Central St	0.1 mi
32	Turn right at Long Wharf	0.1 mi
33	Swim across the Atlantic Ocean	3,462 mi
...		
34	Slight right at E05	0.5 mi
35	At the roundabout, take the 2nd exit onto E05/Pont Vauban	0.1 mi
36	Turn right at E05	5.7 mi
37	Take the exit onto A29/E44 toward Amiens Toll road	27.8 mi
...		
38	Take the exit toward Dieppe/Amiens/Calais/A151/Rouen	1.1 mi

1000 mi / 2000 km

©2007 Google - Map data ©2007 European Commission, Mapbox, etc.

Parę lat temu...

Lars i Jens Rasmussen



Bracia Lars i Jens, to główni pomysłodawcy i twórcy Google Maps. Po dużym sukcesie postanawiają rozpocząć kolejny ambitny projekt. Ich pomysłem jest narzędzie do komunikacji i współpracy.

Komunikacja i współpraca dzisiaj

- Najpopularniejsze formy komunikacji w Internecie dzisiaj: e-mail i komunikatory (IM). Zaprojektowane ponad 40 lat temu. Naśladują tradycyjną komunikację: pocztę i rozmowy telefoniczne.
- Dlaczego w ogóle musimy wybierać pomiędzy nimi, skoro się dopełniają?
- Współpraca jest kolejnym oddzielnym kanałem. Jeśli jest potrzeba dyskusji na temat współtworzonego dokumentu, trzeba użyć konferencji na czacie lub grupy dyskusyjnej.
- Narzędzia do współpracy stawiają przed wyborem: bogate narzędzia edycji dokumentu, czy natychmiastowa widoczność zmian? Google Docs vs EtherPad.

Wszystkie funkcje w jednym narzędziu

- Oryginał wave'a znajduje się na jednym serwerze. Klienci i inne serwery oczywiście cache'ują sobie jego kopie, ale zawsze jest jeden serwer źródłowy.
- Wave jest w takim samym stopniu rozmową i dokumentem. Umożliwia komunikację i pracę przy pomocy bogato formatowanego tekstu, zdjęć, filmów, map, itp.
- Wave jest współdzielony. Każdy uczestnik może odpowiedzieć w dowolnym miejscu wiadomości, edytować zawartość i dodawać kolejnych uczestników w dowolnym momencie.
- *Playback* pozwala przewinąć wave na początek i odtworzyć ponownie, żeby zobaczyć kto powiedział co i kiedy.
- Przesyłanie danych na żywo podczas pisania (prawie znak po znaku) umożliwia szybsze wymiany zdań, obserwowanie zmian i interakcję z rozszerzeniami w czasie rzeczywistym.

Konwersacja w Google Wave

The screenshot shows a Google Wave window titled "Google Wave - Anna-Christina...". The address bar shows "http://wave.google.com/". The interface is divided into several sections:

- Navigation:** Includes links for Inbox, Active, All, By Me, History, Spam, Settings, and Trash. It also shows search and folder options.
- Contacts:** Lists participants in the wave, including Anna-Christina, Gregory, Jens, Lars, Narelle, Steph, Alec, Andrew, Ben, and Casey.
- Inbox 1 - 17 of 27:** A list of messages with timestamps and participant avatars. The selected message is "Snapshots! - I'm really impressed with all the nature shots we've put together!" from Noel, dated 3:39 pm.
- Snapshots!:** A detailed view of the selected message, showing the text "I'm really impressed with all the nature shots we've put together!" and a gallery of 12 nature photographs. The gallery includes images of a Desert Antelope, Toucan Too, Dawn Trees, Waterfall, Winter Leaf, Autumn Arvo, Old Creek, and Deserted.
- Tags:** A section at the bottom of the message view with buttons for "Images" and "Files".

Równoczesna edycja

Pics from a walk in Sydney...
- □ ×

Reply Playback Archive Mute Read Unread Folder Spam

Pics from a walk in Sydney... May 26

Just from walking around in Kirribilli / Milsons Point, under the bridge where Luna Park ↗

Dan P: What is Luna Park? Dan

4:28 pm

Gregory (and Casey): It is an amusement park on the water. It has a long Gregory history, in Sydney and in Melbourne. In Sydney, it's burned down a couple of times, and has been shut down intermittently for ever. It no longer has any big rollercoasters (it used to have The Big Dipper, now you have to put up with The Wild Mouse, which is quite the disappointment).

If you go, the best thing is the Casey

↻ ✎

is. It was a lovely day!

clouds

sun in the grass

ferris wheel

IMG_0713

Shane Stephens (and me): Those are some great photos! Where were the ones of grass taken?

4:31 pm

☰ ▶ ☰ ↶ 🔍 ↷

Wave z mapą

The screenshot shows the Google Wave web interface. On the left is a navigation sidebar with options like Inbox, Active, All, My Mail, History, Spam, Settings, and Trash. Below that are search and filter sections. The main area is divided into two panes. The left pane shows an inbox with a list of messages, including a highlighted message from 'Lawrence' at 11:30 am. The right pane shows a conversation with 'Lawrence' where a map is displayed. The map shows a red path connecting several cities, with a red dot indicating a location. The conversation continues with text messages and another map view.

Google Wave - Daynah

Navigation

- Inbox
- Active
- All
- My Mail
- History
- Spam
- Settings
- Trash

SEARCHES [ADD]

FILTERS [ADD]

Contacts

Daynah Main Account

Search

Lawrence wq on tshahBox daynah-test

Inbox 1 - 13 of 13

New Wave in: inbox

Archive Mail Read Unread Favor Spam

Let's see-- adding daynah. hehe. lol! oh, I keep getting refresh... oh. 11:30 am 30 msgs

Testing Google Wave - Hi Lawrence, how are you? Did you enjoy Google POT? I had a 11:02 am 13 msgs

Welcome to Wave! - We wanted to welcome you into our new product! We're excited to 10:55 am 1 msg

Known Issues with Wave - This Wave will list out the current issues. NOTE: This wave is 10:55 am 1 msg

Keyboard Shortcut Cheat Sheet - This is a quick guide to the current keyboard shortcuts 10:55 am 1 msg

What's Just Changed in Wave - This Wave will be updated by the Wave team for each 10:55 am 1 msg

Getting Started with Wave - This wave should help you get started in using Wave. NOTE: 10:55 am 1 msg

Gears for Wave - Gears 0.5.21.0 was released at the same time as the Wave 10:54 am 1 msg

Search Cheat Sheet - This is a quick guide to the operators and redirects supported by wave 10:54 am 1 msg

I wonder how you can delete Google Wave records. -- I see the trash can, but no way of 10:21 am 1 msg

Hi - How are you? I am typing from my test account. This is fun! I'm logged in as test too 10:11 am 4 msgs

Google Wave Notes --UI looks very nice 9:19 am 1 msg

- 9:18 am 1 msg

Let's see--

Reply Feedback Archive Mail Read Unread

Lawrence: Let's try something else.

Map Satellite Hybrid

now: now that is super cool. 10:56 am

Lawrence: hehehe! can you see that? 11:00 am

me (and Lawrence): lol I see a heat in the map! 11:04 am

yes you do! hey I can edit your message - How interesting! Does it highlight my change on yellow?

Tags: testid datab

Images Files

Planowanie wspólnego wyjścia

The screenshot shows a Google Wave interface with a central message stream and a poll on the right. The messages are as follows:

- 3:43 pm**: Dinner and board games - Who's up for dinner and board games next week? The thinking...
- 3:39 pm**: Sceptical! - The really impressed with all the... (replying to "Dinner and board games...")
- 3:22 pm**: BBQ on Sunday! - Our house. Let us know if you can come! What can I bring? Just...
- 3:19 pm**: Sucks! - Hey Dan and Sam, I think its time to get ready... (replying to "BBQ on Sunday!")
- 3:18 pm**: Hey Wagh - Where get coffee tomorrow? We can meet by the gate then.
- 1:12 pm**: Weekend movie player? - Looking to relax May 1, you got movie player around or left on my desk.
- 1:05 pm**: See what turned up on my front lawn yesterday! - It's fairly late in the season to be...
- 12:22 pm**: It's Movie Time!! - Whoa... the DVD are cool again what about "Confessions of a Shopaholic"?
- 8:03 am**: Picn from a walk in Sydney... - Just from walking around in Kirribilli - Miesse Park.
- May 20 7:59 pm**: Gallery opening this Thursday - Hey everyone, my sister is finally going up and I would love if you...
- May 20 7:59 pm**: Any tips for combating jet lag? - Please share! For the first time, they are on long air journeys. Got...
- May 20 7:59 pm**: The Castle (Mandatory Fan!) - Speaking of movies, who hasn't seen "The Castle"? Such...
- May 20 7:59 pm**: Hey, here are some photos from the Kirribilli track - Check out the border walk! At the...
- May 20 7:59 pm**: Favorite Place to Read - Po! - Whoa do you like to go when you have time off? Should like...
- May 20 7:59 pm**: Who wants to see "Transformers"? -
- May 20 7:59 pm**: Official post - Check out the photos from our photo walk! Gladly shared the best photos! -

The poll on the right is titled "Let us know if you can come!" and has three columns: Yes, No, and Maybe?. The "Yes" column includes names like Dan Koberling, Anna-Christina Douglas, Dan Peterson, and Andrew McLean. The "No" column includes Mike Valdesano, Gina Chang, and Tobias Oster. The "Maybe?" column includes Neil Gordon, Mike Standback, John Lippman, Stephen Wu, and Sam Thompson.

Partia szachów

The screenshot displays a Google Wave window with a blue header and a white background. The interface is divided into several sections:

- Navigation:** A vertical sidebar on the left contains icons for 'Index', 'Active', 'All', 'Go to file', 'History', 'Spaces', 'Settings', and 'Trash'. Below this are sections for 'SEARCHES' and 'FOLDERS'.
- Contacts:** A section below navigation lists contacts such as 'Anna-Christina', 'Gregory', 'Steph', 'Alex FULL', 'Andrew M', 'Dae', 'Casey', and 'Daniel'.
- Wave Content:** The main area shows a list of messages. The top message is 'Chess times = fun times - It's off!' with a timestamp of 3:54 pm. Other messages include 'Dinner and board games', 'Snaghotel', 'BBQ on Sunday!', 'Sushi time?', 'Hey Steph...', 'Wayward music player?', 'See what turned up on my front lawn yesterday!', 'It's Movie Time!!!', 'Pics from a walk in Sydney...', 'Gallery opening this Thursday', 'Any tips for combating jet lag?', 'Say, here are some photos from the Kokoda track', and 'Favorite Place to Travel - Fall'.
- Chess Game:** A window titled 'Chess times = fun times' is open on the right. It shows a chessboard with pieces in their starting positions. To the right of the board is a list of moves:
 1. d2-d3 e7-e6
 2. g2-g3 g6-g7
 3. h2-h3 h6-h7
 4. h3-h4 h7-h8

Użyte technologie

- Cały klient jest napisany w Google Web Toolkit. GWT w najnowszej wersji korzysta z wielu udogodnień HTML 5 i bez tego Wave nie mógłby wyglądać jakby był normalną aplikacją (nie w przeglądarce) i osiągnąć takiej wydajności.
- Wszystkie roboty działają na Google AppEngine. To nie jest docelowe rozwiązanie, w finalnej wersji będzie można postawić robota na dowolnym serwerze.

Kalendarz projektu

- **28 V 2009** – Pierwsza demonstracja Wave'a podczas Google I/O. Pierwsze konta deweloperskie dla programistów spoza Google, zostały założone tego dnia dla ludzi obecnych na prezentacji.
- **30 IX 2009** – Dostęp do przepremierowej wersji Wave zostaje rozszerzony na nie-deweloperów. Konta ma otrzymać 100 000 użytkowników. Do tego momentu utworzono już 20 000 kont dla deweloperów.
- **???** – Uruchomienie pełnej, stabilnej i otwartej wersji.

Spis treści

- 1 Wprowadzenie
- 2 Google Wave API
 - Gadżety
 - Roboty
 - Osadzanie
- 3 Federation Protocol

Platforma

- Google Wave może być wykorzystany na podobnej zasadzie jak Maps. Prosty sposób osadzania, pozwala wkleić uproszczoną wersję interfejsu Wave do własnego serwisu.
- Można rozszerzać klienta Wave na dwa sposoby:
 - Gadżety – podobne do tych wklejanych na stronę iGoogle (mini-mapa, lista zakupów, itp.). Podstawowa różnica jest taka, że teraz stan gadżetu może być w czasie rzeczywistym widoczny dla wszystkich uczestników wave'a i edytowany przez nich.
 - Roboty – programy, które mogą zostać zaproszone jako uczestnik wave'a, żeby wykonać dla nas jakąś pożyteczną pracę. Mają wgląd w całą zawartość wave'a, dostają powiadomienia o zmianie stanu i mogą dowolnie dodawać i edytować treść.

Przykład z gadżetami

BBQ on Sunday!

Marcel edited this message on May 25 51 of 73

Dan K (and Noel, Dan P., ...): May 25
BBQ on Sunday!
 Our house, 6pm

Let us know if you can come!

Here's a map to our casa:

Map data ©2009 TopoGIS

Yes	No	Maybe?
Dan Kettering	Alexis Vuillemin	Noel Gordon Could catch a plane I guess :)
Dan Peterson What a great neighborhood!	Grace Chung	
	Vadim Gerasimov	

Przykład 1: najprostszy gadżet

```
<?xml version='1.0' encoding='UTF-8' ?>
<Module>
  <ModulePrefs title='Hello Wave'>
    <Require feature='wave' />
  </ModulePrefs>
  <Content type='html'>
    <![CDATA[
      Hello, Wave!
    ]]>
  </Content>
</Module>
```

Przykład 2: liczenie kliknięć

```
<script type='text/javascript'>
function buttonClicked() {
    var value = parseInt(wave.getState().get('count', '0'));
    wave.getState().submitDelta('count': value + 1);
}

...

// Reset value of 'count' to 0
function resetCounter(){
    wave.getState().submitDelta('count': '0');
}

</script>
<input type=button value='Click Me!' id='butCount'
    onClick='buttonClicked()'>
<input type=button value='Reset' id='butReset'
    onClick='resetCounter()'>
```

Przykład 2: liczenie kliknięć

```
function stateUpdated() {
  if(!wave.getState().get('count')) {
    div.innerHTML = 'The count is 0.'
  }
  else {
    div.innerHTML = 'The count is ' + wave.getState().get('count');
  }
}

function init() {
  if (wave && wave.isInWaveContainer()) {
    wave.setStateCallback(stateUpdated);
  }
}

gadgets.util.registerOnLoadHandler(init);
```

Przykład 3: aukcja

```
function buttonClicked() {
  var viewerId = wave.getViewer().getId();
  var state = wave.getState();
  var bid = parseInt(document.getElementById('yourBid').value);
  var currentBid = parseInt(state.get(viewerId, '0'));
  if (bid > currentBid) {
    delta = {};
    delta[viewerId] = bid;
    state.submitDelta(delta);
  }
}

function init() {
  if (wave && wave.isInWaveContainer()) {
    wave.setStateCallback(renderInfo);
    wave.setParticipantCallback(renderInfo);
  }
}
gadgets.util.registerOnLoadHandler(init);
```

Metody wave'owego gadżetu

- `setStateCallback(callback, opt_context)`
- `setParticipantCallback(callback, opt_context)`
- `wave.getState()`
- `submitDelta(delta)`
- `wave.Participant.getId()`,
`wave.Participant.getDisplayName()`,
`wave.Participant.getThumbnailUrl()`

- Roboty komunikują się z wave serwerem przy pomocy Wave Robot Protocol. Jest on póki co jeszcze niestabilny, dlatego nie jest udostępniony.
- Udostępnione są dwie biblioteki: w Pythonie i w Javie, stanowiące interfejs do protokołu.
- Roboty mogą: modyfikować wave'y, podejmować interakcję z innymi uczestnikami, tworzyć nowe wave'y. Poza tym wszystko inne, co jest możliwe na serwerze, na którym działają.

- Roboty deklarują, jakie zdarzenia je interesują i są powiadamiane o nich.
- Dodatkowo mogą zarządzać, żeby serwer kontaktował się z nimi w ustalonych odstępach czasu. Są to tak zwane *cron events*.
- Robot nie może skontaktować się z wave serwerem, kiedy chce. Może tylko odpowiadać na wysłane mu zdarzenia (także te cronowe).
- Roboty nie są traktowane specjalnie. Ich akcje mają taką samą wagę, jak akcje ludzkich uczestników. Nie ma żadnej gwarancji, że kiedy wysłana przez robota akcja jest wykonywana, wave jest wciąż w stanie, na który jest ona odpowiedzią.
- Roboty powinny mieć swoje profile, wraz ze zdjęciem :)

Przykłady robotów

- **Polly the Pollster** – pozwala na przeprowadzenie ankiety i wyświetla jej wyniki.
- **Bloggy** – publikuje wave'a na blogu.
- **Buggy** – podłącza się pod bug trackera i importuje zgłoszenia do wave'a.
- **Tweety** – loguje się na Twoje konto w Twitterze i pozwala czytać i wysyłać twity.
- **Rosy** – w czasie rzeczywistym (znak po znaku), tłumaczy dyskusję pomiędzy 40-toma językami.

Yet another smiley robot

```
from waveapi import events
from waveapi import robot

def OnBlipSubmitted(properties, context):
    blip = context.GetBlipById(properties['blipId'])
    contents = blip.GetDocument().GetText()
    contents = contents.replace(':-(', unichr(0x2639))
                    .replace(':-) ', unichr(0x263A))
    blip.GetDocument().SetText(contents)

if __name__ == '__main__':
    yasr = robot.Robot('Yasr',
                      image_url='http://wave-api-dmo.appspot.com/public/smiley.png',
                      profile_url='http://code.google.com')

    yasr.RegisterHandler(events.BLIP_SUBMITTED, OnBlipSubmitted)
    yasr.Run(debug=True)
```

Visitor Photo Sharing

Replay Playback Archive Mute Read Unread

The wave is published at an unknown url.

Visitor Photo Sharing May 27

Google Island welcomes our visitors to share photos from their vacation with us. Upload your favorites and comment on others.

Ryan Hickman: May 27

Windsurfing Bay Island Coast

me (and Marcel Prasetya): May 27

Beautiful! Nice sand Red sky.

Tags: Images Files



Embedded Wave With Drag-n-Drop

http://code.google.com/wave/

Google Island
COAST WAVE WITH US

WELCOME -- May 27, 2009

Visitor Photo Sharing 3:06 pm

Google Island welcomes our visitors to share photos from their vacation with us. Upload your favorites and comment on others.

rhickman@wevesandbox.com: 6:12 pm

Windsurfing Bay Island Coast

me (and marcel@wevesandbox.com): 5:45 pm

Beautiful! Nice sand Red sky.

rhickman@wevesandbox.com: Here are the photos I took from my trip last month. 3:20 pm

- W sekcji head:

```
<script src=''http://wave-api.appspot.com/public/embed.js''
  type=''text/javascript''></script>
<script type=''text/javascript''>
  function initialize() {
    var wavePanel = new WavePanel
      ('http://wave.google.com/a/wavesandbox.com/');
    wavePanel.loadWave('wavesandbox.com!w+waveID');
    wavePanel.init(document.getElementById('waveframe'));
  }
</script>
```

- Sekcja body:

```
<body onload=''initialize()''>
  <div id=''waveframe'' style=''width: 500px; height: 100%''></div>
</body>
```

Spis treści

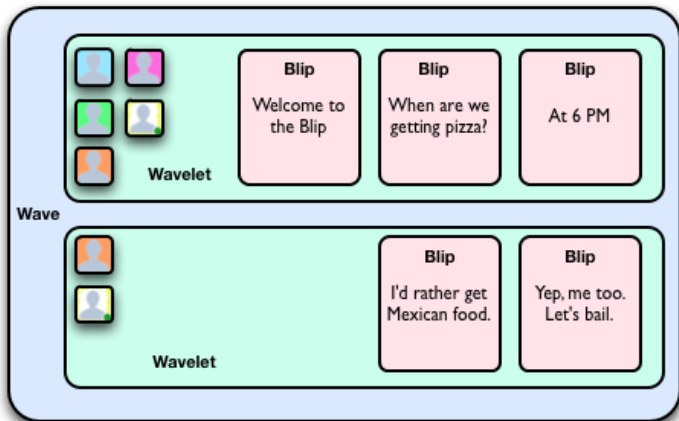
- 1 Wprowadzenie
- 2 Google Wave API
- 3 Federation Protocol**
 - Model danych
 - Kontrola współbieżności

Google Wave Federation Protocol

- Google chciałby, żeby Wave stał się ogólnointernetowym standardem i wyparł e-mail, komunikatory i wszystkie formy współpracy.
- Oczywiście to nie byłoby możliwe, gdyby to była tylko jego własna technologia, dostarczana tylko na jego serwerach, itd.
- Od samego początku projekt zakładał, że każdy musi być w stanie postawić swój własny serwer i stać się dostawcą wave'a.
- Dlatego powstał protokół, który umożliwia komunikację między serwerami, w przypadku, gdy uczestnicy jednego wave'a mają konta u różnych dostawców.

Google Wave Federation Protocol

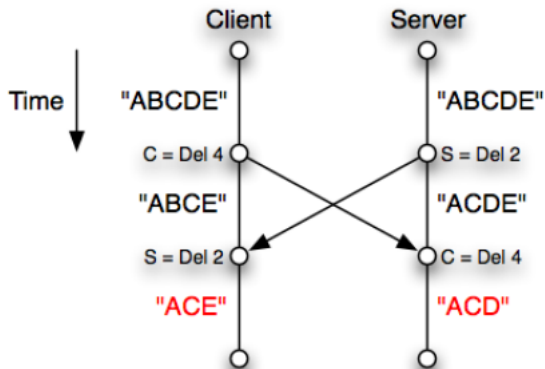
- Każdy wavelet posiada swój serwer centralny i jest to serwer uczestnika, który go rozpoczął. Kiedy zostają dodani uczestnicy z innej domeny, serwer źródłowy kontaktuje się z ich serwerem i powiadamia o dodaniu uczestnika.
- Serwer źródłowy wysyła update'y do wszystkich innych. Pozostałe z kolei przysyłają operacje wykonywane przez ich uczestników.
- Wszystkie zapytania o historię wave'a (playback) są również przesyłane do serwera centralnego, który w odpowiedzi przesyła odpowiedni stan wave'a.



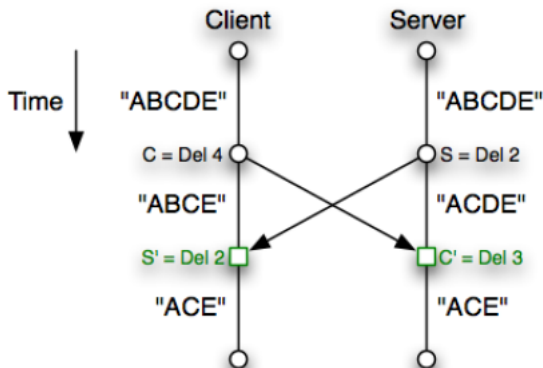
Podstawowe elementy

- **wave** – cała pojedyncza dyskusja, posiadająca stan aktualny i historię oraz jednego lub więcej uczestników;
- **wavelet** – podstawowa jednostka kontroli dostępu; w zasadzie podobnie jak wave jest to dyskusja, z tym że cały wavelet ma jeden zbiór uczestników, którzy mają pełny dostęp do czytania i edytowania całej zawartości.
- **blip** – najmniejsza jednostka dyskusji, pojedyncza wiadomość.
- **dokument** – jest to zawartość blipa, zapisywana w formacie XML.

Współbieżność: wersja naiwna



Współbieżność: wersja poprawna

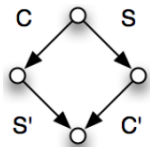


Operational Transform

- Zmiany są przesyłane pomiędzy stronami w postaci operacji.
- Stany obiektu u obu stron będą zbiegać do oczekiwanego stanu, jeśli istnieje funkcja **transform()** o następujących właściwościach:

Niech:

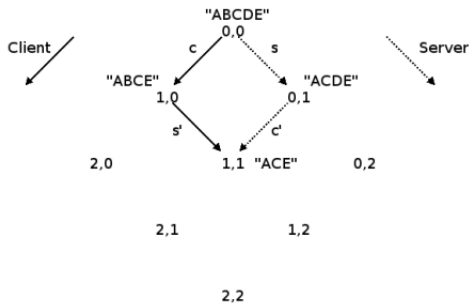
- S – operacja serwera,
 C – operacja klienta,
 S' – przetransformowana operacja serwera,
 C' – przetransformowana operacja klienta,
- $(S', C') = \text{transform}(S, C)$.



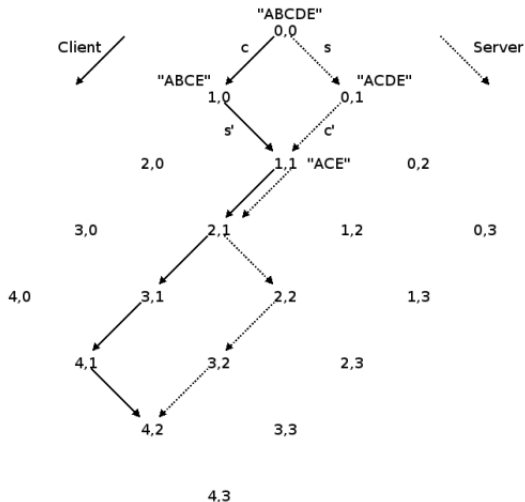
Wówczas $S' \cdot C = C' \cdot S$

Przestrzeń stanów

Operacje wykonane przez klienta powodują schodzenie w lewo, u serwera – w prawo.

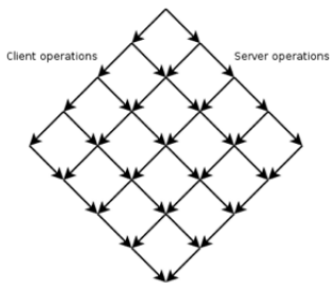


Poruszanie się w przestrzeni stanów



Duże transformacje

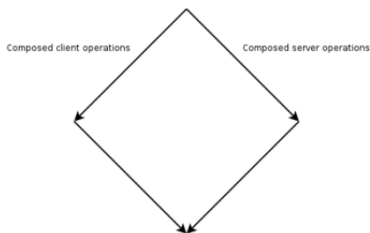
Jeżeli stany stron odbiegną od siebie bardzo, transformowanie wszystkich operacji po obu stronach może być bardzo kosztowne. Jeżeli klient wykonał n operacji a serwer m , potrzeba nm transformacji.



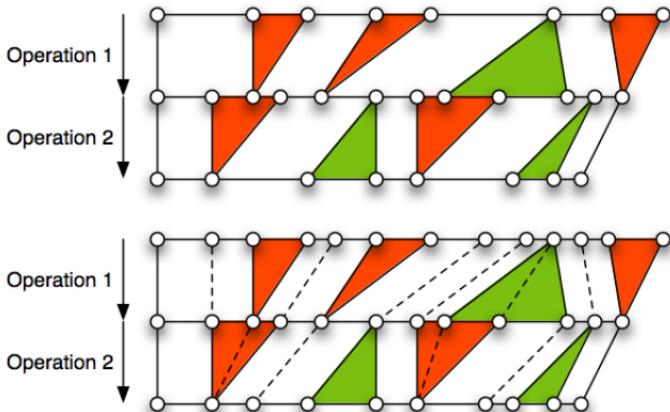
Kompozycja zamiast dużych transformacji

Operacje w Federation Protocol są tak dobrane, żeby złożenie dwóch operacji wciąż było operacją. Jeżeli kompozycja byłaby efektywna, można by najpierw skleić wszystkie operacje klienta, potem operacje serwera i na koniec wykonać tylko dwie transformacje.

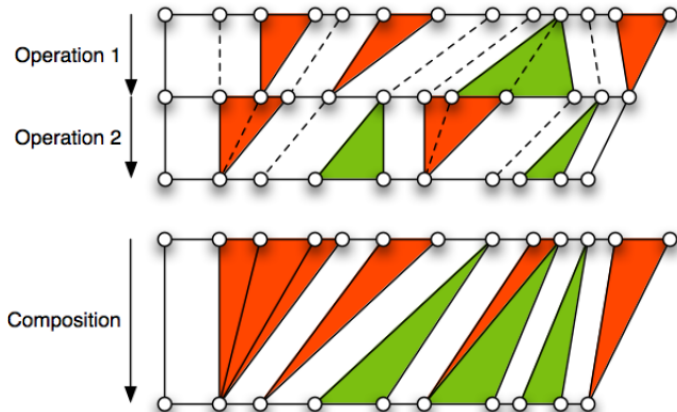
Okazuje się, że da się zaimplementować kompozycję tak, żeby koszt transformacji spadł do $O(n \log n + m \log m)$.



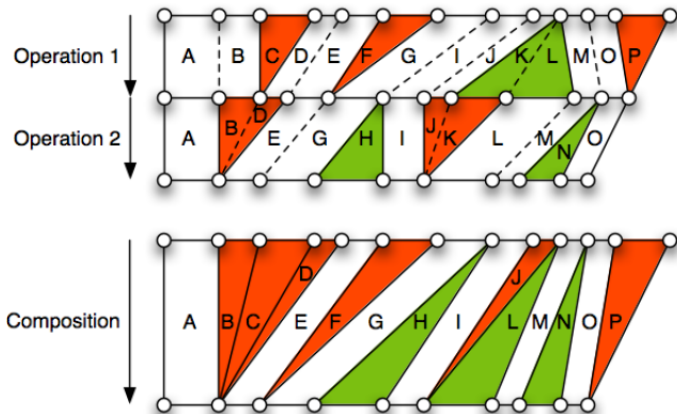
Przykład kompozycji



Przykład kompozycji

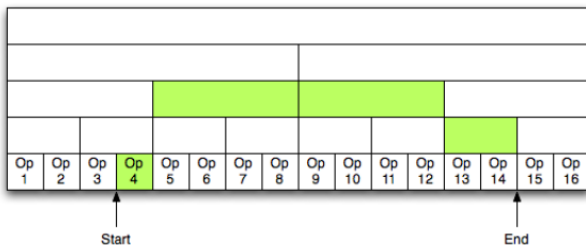


Przykład kompozycji



Drzewo kompozycji

Dodatkową korzyścią jest to, że można pomiędzy stanami w historii przechodzić w czasie logarytmicznym, przy pomocy takiej struktury danych.



Strony projektu

- <http://wave.google.com>
- <http://code.google.com/apis/wave>
- <http://waveprotocol.org>

Filmy z Google I/O

- [Demonstracja](#)
- [Google Wave: Under the hood](#)
- [Programming with Google Wave](#)
- [Google Wave: Powered by GWT](#)