Overview of Google Wave
Google Wave Client

- search
- authentication
- access
- control
- playback
- waves
- attachments
- gadgets

- abuse detection
- saved searches
- folders
- contacts
- presence
Product vs Technology

Google Wave (product)

vs

Wave (technology)

wave is to Google Wave as email is to Gmail
Federation
What is Federation?
What is Federation?

Open

Iterating

http://waveprotocol.org/
Why Federation?

Avoid Fragmentation

Encourage Adoption
- Open APIs and Standard Protocols
- User Choice
Novell Pulse

Coming in mid 2010, Novell Pulse will first be delivered as a service and then later available for on-premise deployment. It will also work seamlessly with Google Wave so you and anyone you want to work with can have your choice and get down to business.

http://www.novell.com/products/pulse/
Technical Background
Wave Data Model
Documents

<body><line t='h3'>1</line> H3e!lo!
</body>
Annotations

<body>
    <line t='h3'>Hello!</line>
</body>

style/fontSize=bold [3,7]
Wave Data Model

Wavelet

• unit of concurrency
• unit of federation
Federation: Sharing Wavelets

Wave Servers

• Run OT
• Share updates to wavelets

Wavelet id determines owner
• domain: “initech-corp.com”
• id: “conv+3sG7”
Operational Transforms
Operational Transformation

Client

"ABCDE"

Del 4

"ABCE"

Del 2’ = Del 2

"ACE"

Server

"ABCDE"

Del 2

"ACDE"

Del 4’ = Del 3

"ACE"
Operational Transformation

Changes to a shared object is an operation
  - e.g. insert character "a" at position x

As long as there is a function `transform()` with the following behavior, it is always possible to have all the clients come to an convergent state.
  - $S = \text{Server Operation}$
  - $C = \text{Client Operation}$
  - $S' = \text{Transformed Server Operation}$
  - $C' = \text{Transformed Client Operation}$

$\left(S', C'\right) = \text{transform}(S, C)$

where

$C' \cdot S = S' \cdot C$

Client and server must have the same `transform()` function
Operations Compose
Efficiency

Requires $nm$ transformations, where $n$ is the number of client operations and $m$ is the number of server operations.
First Compose, Then Transform

We can design composition so that transformation running time is $O(n \log n + m \log m)$
Document Interface

A streaming interface.

Similar to an XMLStreamWriter or a SAX handler.

An example operation could perform the following sequence:

| Skip 8 | ElementStart tagName: "li" | Insert "hello" | ElementEnd | Skip 8 | Delete 3 |
The operation composer works by "zipping" two streaming operations into a single streaming operation.
The operation transformer works by "zipping" two streaming operations into two streaming operations.
Triggering Federation

AddParticipant Operation

- david@googlewave.com
- hannon@acmewave.com

A participant’s domain determines where to send ops
Example – Initial Synchronzation

bob@acmewave.com adds milton@initech-corp.com

- Bob's client sends AddParticipant("milton@initech-corp.com")
- acmewave.com looks up initech-corp.com's waveserver
- acmewave.com pushes the AddParticipant operation
- initech-corp.com requests the wavelet history
Federation Delta Flow

AcmeWave.com

Federation Host

Wave Server

Wave Client

Federation Remote

XMPP

Federation Remote

Wave Server

Federation Host

Initech-corp.com

send updates on local waves

receive updates on remote waves
Data stays on your network

- On-premise solution
- If you don't add anyone from the outside the data stays in your network
- Scenario: Chat between colleagues in a company
Progress to date

Published draft specifications

- Google Wave Federation Protocol
- Google Wave Conversation Model
Progress to date

Open source

- http://code.google.com/p/wave-protocol/
- ~40K lines of code, Java, Apache 2.0

Two Components:
- Wave Model + Operational Transformation
- FedOne: Basic prototype client/cerver and early crypto library
FedOne

Executable spec for the wire protocol

Not the beginning of a reference impl
FedOne
Work in progress

Opening up a federation port on WaveSandbox.com
  ● Still highly experimental

Updating the FedOne client/server
  ● Client does a better job displaying OT
  ● Server understands the new conversation manifest

Published a Google Wave Contributors Licensing Agreement
Going Forward

- Iterating towards stable specifications
- Gain experience running an open federation port
- Open a federation port on wave.google.com
- Open source the lion's share of Google's client & server
- Develop a production quality reference implementation
Participate

Google can't do this alone!
We’d like to work with you.

http://www.waveprotocol.org/
http://groups.google.com/group/wave-protocol/
Questions?
Agents and Robots