Google Wave
The Tech

Operational Transformation (OT)

The basic idea of OT can be illustrated by using a simple text editing scenario as follows. Given a text document with a string "abc" replicated at two collaborating sites; and two concurrent operations:

1. O1 = Insert[0, "x"] (to insert character "x" at position "0")

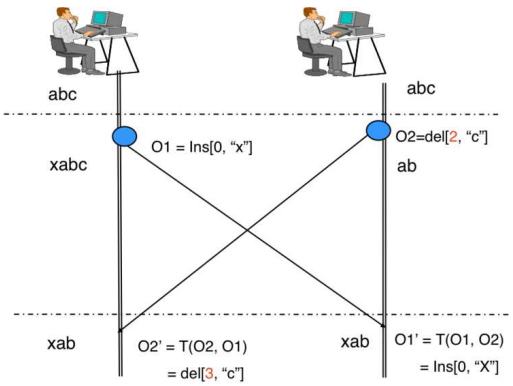
2. O2 = Delete[2, "c"] (to delete the character "c" at position "2")

generated by two users at collaborating sites 1 and 2, respectively.

Suppose the two operations are executed in the order of O1 and O2 (at site 1). After executing O1, the document becomes "xabc". To execute O2 after O1, O2 must be transformed against O1 to become: O2' = Delete[3, "c"], whose positional parameter is incremented by one due to the insertion of one character "x" by O1.

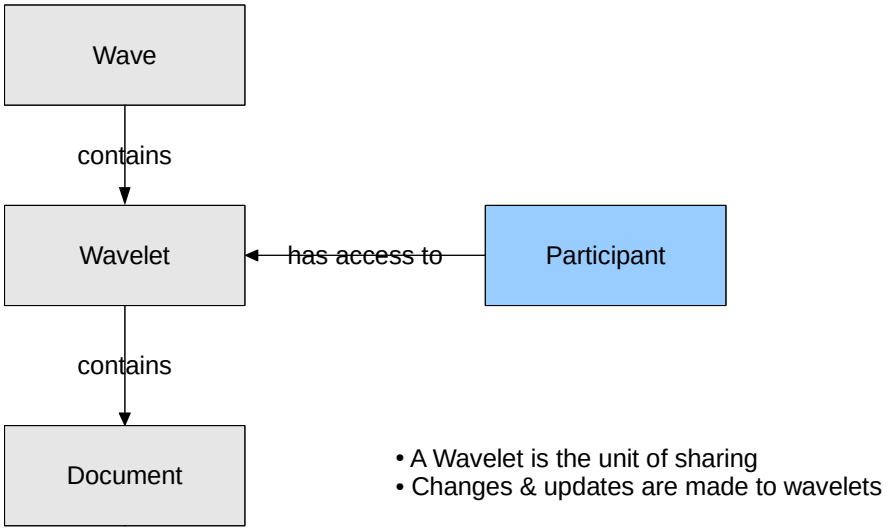
Executing O2' on "xabc" shall delete the correct character "c" and the document becomes "xab". However, if O2 is executed without transformation, then it shall incorrectly delete character "b" rather than "c".

The basic idea of OT is to transform (or adjust) the parameters of an editing operation according to the effects of previously executed concurrent operations so that the transformed operation can achieve the correct effect and maintain document consistency.



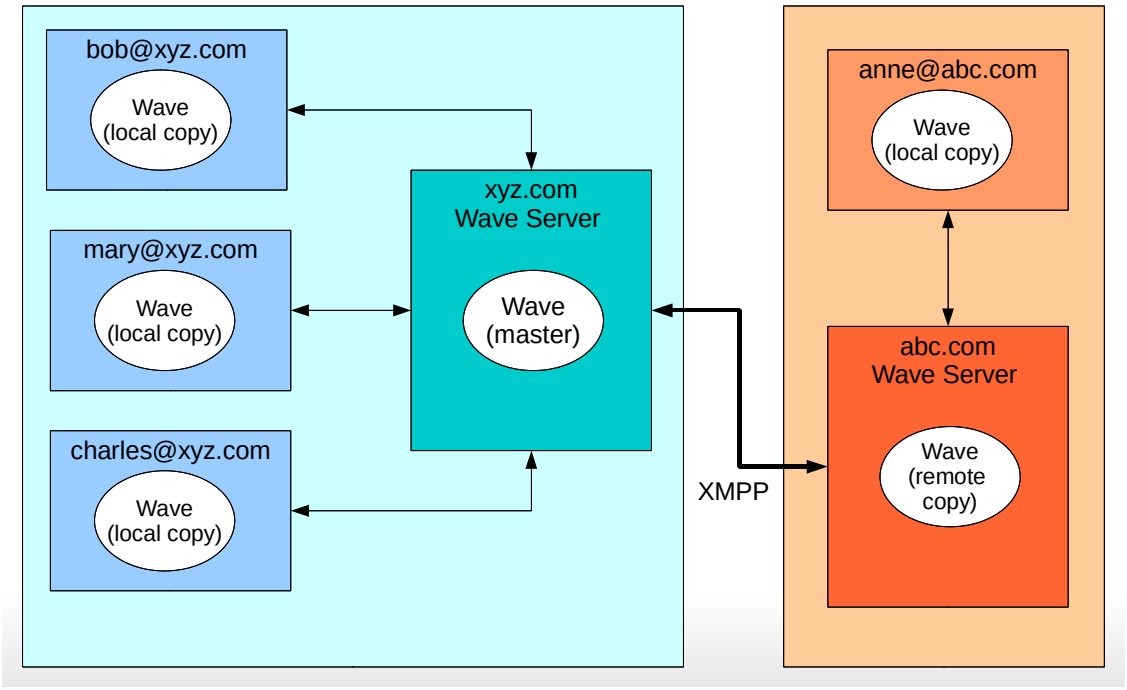
See http://en.wikipedia.org/wiki/Operational_transformation and http://www.waveprotocol.org/whitepapers/operational-transform

Wave Data Model



See
http://www.waveprotocol.org/whitepapers/internal-client-server-protocol

Wave Communications



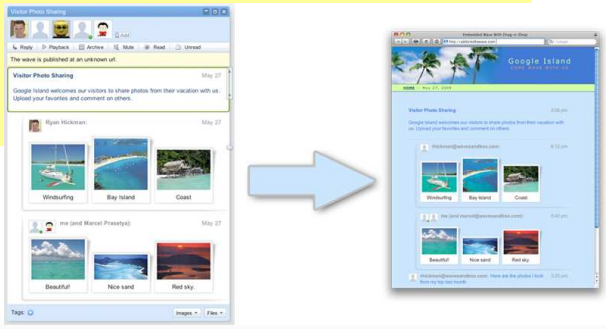
See
http://www.waveprotocol.org/whitepapers/internal-client-server-protocol
and
http://www.waveprotocol.org/draft-protocol-specs/draft-protocol-spec

The Wave API

- Google provides a complete API for Wave
- Two key parts:
  - **Embed API** – Allows you to embed a Wave into your own system or site
  - **Extensions API** – Allows you to create “mini-apps” that can be added to Waves

The Embed API

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Google Wave Embed API Example: Simple Wave</title>
    <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>
  </head>
  <body onload="initialize()">
    <div id="waveframe" style="width: 500px;
  </body>
</html>
```



See http://code.google.com/apis/wave/embed/guide.html

The Extensions API

- Extensions come in two flavours:
  - **Gadgets** - provide a shared program which runs within the wave, and to which all participants have access.
  - **Robots** - are applications which can be added to waves as automated wave participants. Robot extensions commonly automate tasks, but can also participate in the wave as a participant, interacting with the conversation based on their capabilities.

See http://code.google.com/apis/wave/extensions/

Writing Gadgets

- Gadget are created using the Google Gadgets API
- They are published as XML documents on the web
- You add a gadget to a Wave by linking to its URL

See http://code.google.com/apis/wave/extensions/gadgets/guide.html and http://code.google.com/apis/gadgets/docs/dev_guide.html

TickTock Gadget

```
<?xml version="1.0" encoding="UTF-8"?>
<Module>
<ModulePrefs title="ticktock - universal time" height="41" scrolling="false" author="Jonathan
Ackerman" author_email="dont_spam_me_rabidgremlin@gmail.com"
description="Ticktock displays the time in tocks. The time in tocks is the same for everyone
regardless of where they are on the planet and 1000 tocks make up a day.Great for organizing IM
chats and meetings with people in different timezones.">
<Require feature="flash" />
</ModulePrefs>
<Content type="html"><![CDATA[
<div id="flashcontainer" style="text-align: center;margin-top:5px;"></div>
<script type="text/javascript">
_IG_EmbedFlash("http://www.rabidgremlin.com/ticktock/tocker.swf", "flashcontainer", {
swf_version: 6,
id: "flashid",
width: 171,
height: 41
});
</script>
]]>
</Content>
</Module>
```

See http://www.rabidgremlin.com/ticktock/google_ticktock.xml

Writing Robots

- You can use a robot to perform actions such as :
  - modify information in a wave
  - interact with participants in a wave
  - communicate and synchronize information in a wave to the outside world or to other waves
  - access or modify state in a third-party (such as a database)
- Robots can be written in Java or in Python
- Currently have to host on Google App Engine

See http://code.google.com/apis/wave/extensions/robots/index.html

Writing Robots - continued

```
package com.google.wave.api.samples;
import com.google.wave.api.*;
public class ParrotyServlet extends AbstractRobotServlet {
  @Override
  public void processEvents(RobotMessageBundle bundle) {
    Wavelet wavelet = bundle.getWavelet();
    if (bundle.wasSelfAdded()) {
      Blip blip = wavelet.appendBlip();
      TextView textView = blip.getDocument();
      textView.append("I'm alive!");
    }
    for (Event e: bundle.getEvents()) {
      if (e.getType() == EventType.WAVELET_PARTICIPANTS_CHANGED){
        Blip blip = wavelet.appendBlip();
        TextView textView = blip.getDocument();
        textView.append("Hi, everybody!");
      }
    }
  }
}
```

See
http://code.google.com/apis/wave/extensions/robots/java-tutorial.html
and http://wave-samples-gallery.appspot.com/

fin