Some tools for building web-based groupware

Michel Beaudouin-Lafon
mbl@lri.fr

Javascript

- More powerful than you think
- Functional + Object-oriented
  - function map(a, f) {
    var sum = 0;
    for (var i = 0; i < a.length; i++)
    sum += f(a[i]);
    return sum;
    }
    map([1, 2, 3], function(x) { return x*x }); // 14

Javascript

- var p = {
  x: 10, y: 25,
  moveBy: function(dx, dy) {
  this.x += dx; this.y += dy
  }
  }
- p.moveBy(5, 10); // p = { x: 15, y: 35, moveBy: …}

Javascript

- function PointConstructor() {
  this.x = 0; this.y = 0
  }
  PointConstructor.prototype.moveBy =
  function(dx, dy) {
  this.x += dx; this.y += dy;
  }
- var p = new PointConstructor();
  p.moveBy(5, 10);

Node.js

- Use javascript to program web app servers
- Asynchronous event handling
  - var Emitter = require('events').EventEmitter;
    emitter = new Emitter();
    emitter.on('hello', function(name) {
    console.log('Hello ', name);
    });
    emitter.emit('hello', 'Alice');
- Streams: general event-driven streams

Node.js – Express HTTP server

- Basis of many web applications
  - var express = require('express'),
    app = express.createServer();
    app.get('/', function(req, res) {
    res.send('Hello World');
    });
    app.listen(8080);
  - // serve static files
    app.use(express.static(__dirname + '/public'));

Node.js – socket.io

- Communication between web page and server
- Server :
  - var io = require('socket.io').listen(80);
    io.sockets.on('connection', function (client) {
    client.on('msg', function (data) {
    // send to all other clients
    client.broadcast.emit('msg', data);
    });
    });

Node.js – socket.io

- Client (web page)
  - <script>
    var server = io.connect();
    server.on('connect', function () {
    server.emit('msg', 'Hello');
    });
    server.on('msg', function(data) {
    $('#chat').append(data);
    });
    </script>

Node.js - sharejs

- Shared text strings and objects synchronized through operational transformation
  - sharejs.open('blag', 'text', function(error, doc) {
    var elem = document.getElementById('pad');
    doc.attach_textarea(elem);
    });

Node-webkit

- Combines a node.js server and a web browser
- Access to desktop: files, menus, …
- Desktop-based web applications
- Avoids web protocol issues (same-origin, …)
- Global namespace shared between node server and every web window

dnode

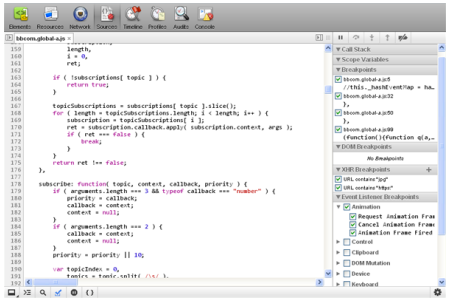
- Remote function call for node.js and browser
- Server:
  - **var** dnode **=** require('dnode');
    var server **=** dnode({
    transform **: function** (s, cb) {
    cb(s.replace(/[aeiou]{2,}/,'oo').toUpperCase())
    }
    });
    server.listen(5004);

dnode

- Client
  - **var** dnode **=** require('dnode');
    var d **=** dnode.connect(5004);
    d.on('remote', **function** (remote) {
    remote.transform('beep', **function** (s) {
    console.log('beep => ' **+** s);
    d.end();
    });
    });
  - $ node server.js &
    $ node client.js
    beep => BOOP

Debugging

- In the browser: web inspector



Debugging

- Node.js
  - Built-in debugger:
    - % node debug myscript.js
  - Remote debugging with web inspector
    - Node-inspector package
    - % node –debug-brk myscript.js
    - % node-inspector &
    - Browse to http://localhost:8080/debug?port=5858

Project ideas

- Shared drawing tool (doodling)
- Shared browsing
- Shared annotation of any web page
- Multi-room chat
- …