Web Compiler Service

Axel T. Schreiner
http://www.cs.rit.edu/~ats/talks/wcs/wcs.mov
partially supported by a Provost’s Learning Innovation Grant 2005

mailto:ats@cs.rit.edu
Outline

- Origins
- User Experience
- Architecture & Implementation
- Next?
Origins

http://www.cs.rit.edu/~ats/talks/lp/lp.mov

- **pj** — integrate *jLex*, *jay*, and parse trees
- **jag** — tree visitor generator
- **lp** — *interactive* tutorial on language processing

- (deprecated) Services:
  - http://murphy.cs.rit.edu:8080/js/js javac  

User Experience

http://wcs.cs.rit.edu:8080/wcs/wcs

HTML form

Applet


Applications

- Language Processing tutorial
- Java tutorials (R. L. Canosa & D. E. Mulcahey)
- Homeworks and solutions
- ... (almost) anything that can be compiled to execute in a JVM, subject to the applet sandbox.

http://www.cs.rit.edu/~rlc/Red/
Poor Man’s IDE

- HTML form
- Javascript to manage text areas
- select to control compilation

Request

```
processor.0=javac
source.0=public class Hello ...
sink.0=applet
```

- global key/value pairs, e.g., for help text
- job step key/value pairs, grouped by suffix
- sources inline or URL
- form=true creates initialized HTML form

Backend


- Process-based Task framework
- wraps standard and native tools
- common Java compilation step (1.4)
- jar to package applet archive
- self-cleaning file system management
Execution

- applet simulates standard i/o and java launch
- fake `System` inserted in each package, each delegates to applet
- user program is *not* adapted to applet

Summary

- \( \wedge \)cs supports interactive experiments with Java and compiler tools (for now).

- No need for client-side installation.

- No need to adapt programs to browser.

- Sandbox security — client’s programs are executed client-side.

- Used in Java and compiler construction tutorials.

http://www.cs.rit.edu/~rlc/Red/
DOM, at least for (indirect) form creation

```
<jobs action='http://wcs.cs.rit.edu:8080/wcs/wcs'
css='http://wcs.cs.rit.edu:8080/wcs/wcs.css'
dynamic='false' javascript='http://wcs.cs.rit.edu:8080/wcs/wcs.js'
target='_blank'>
  <job processor='pj' format='text' sink='applet' file='gl/gl.pj'
    scanner='static' trace='false' tree='static' flatten='false' />
  <job processor='javac' format='text' sink='applet' file='gl/Eval.java' />
  <job processor='javac' format='text' sink='applet' file='gl/Env.java' />
  <job processor='javac' format='text' sink='false'
    file='gl/doc-files/euclid.txt' />
</jobs>
```
Next?

- **oops** — LL(1) parser generator with serialized scanner and parser
- Other parser generators, e.g., CUP, JFlex, ANTLR
- Java version selection
- C# — but how to spoof `System`?