FLOW Meeting 4/22/11

Mark's Implementation of the Solver parser

Yacc could not allow 2 production of expressions

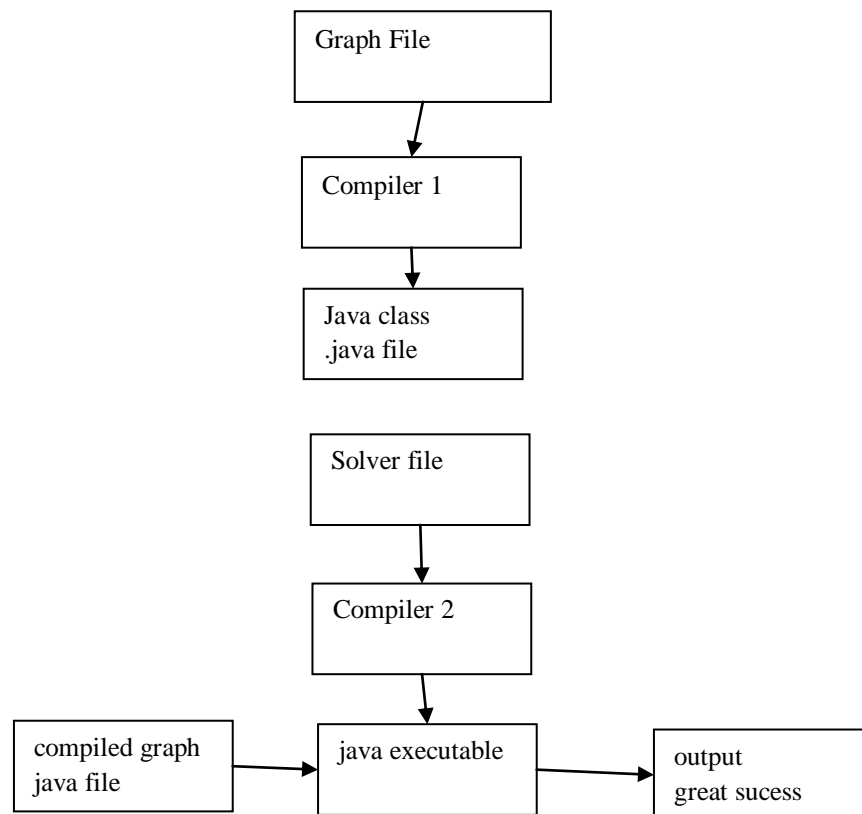Solution: Use two separate files for each (Graph and Solver)

Discussion of Compliation

Commands (tenative):

1) flow graphfilename
2) flow solverfilename CompiledGraphFile
3) java flowOutputfile

Diagrams

```
                        ┌─────────────────┐
                        │ Graph File      │
                        └─────────────────┘
                                │
                                ▼
                        ┌─────────────────┐
                        │ Compiler 1      │
                        └─────────────────┘
                                │
                                ▼
                        ┌─────────────────┐
                        │ Java class      │
                        │ .java file      │
                        └─────────────────┘


                        ┌─────────────────┐
                        │ Solver file     │
                        └─────────────────┘
                                │
                                ▼
                        ┌─────────────────┐
                        │ Compiler 2      │
                        └─────────────────┘
                                │
                                ▼
┌─────────────────┐     ┌─────────────────┐     ┌─────────────────┐
│ compiled graph  │ ──► │ java executable │ ──► │ output          │
│ java file       │     │                 │     │ great sucess    │
└─────────────────┘     └─────────────────┘     └─────────────────┘
```

We also discussed the output of the graph compilation. William sugested that rather than outputting a java class that creates the graph when you instatiate an object of that class, we can output a serialized object already created.

What do we have done?

Translation: yacc and SDT

Lexer

code generation for Graph

What needs to be done?

Split the lexer and parser for graph and solver

Solver code generation (Mark)

yacc code generation (explicitly implement printing the root)

Test cases:

Good working test cases that do things

Error cases: 1) syntax errors 2) logical errors? i.e. undeclared variables (we haven't really implemented a symbol table yet)

Document what errors have occurred

Goal: fewest test files as possible

Extension: write a script to automate testing