The MRRoboto Language

Jason Kopylec
Hema Krishnan
Adam Marczyk

Programming Languages
Spring 2004

The Inspiration

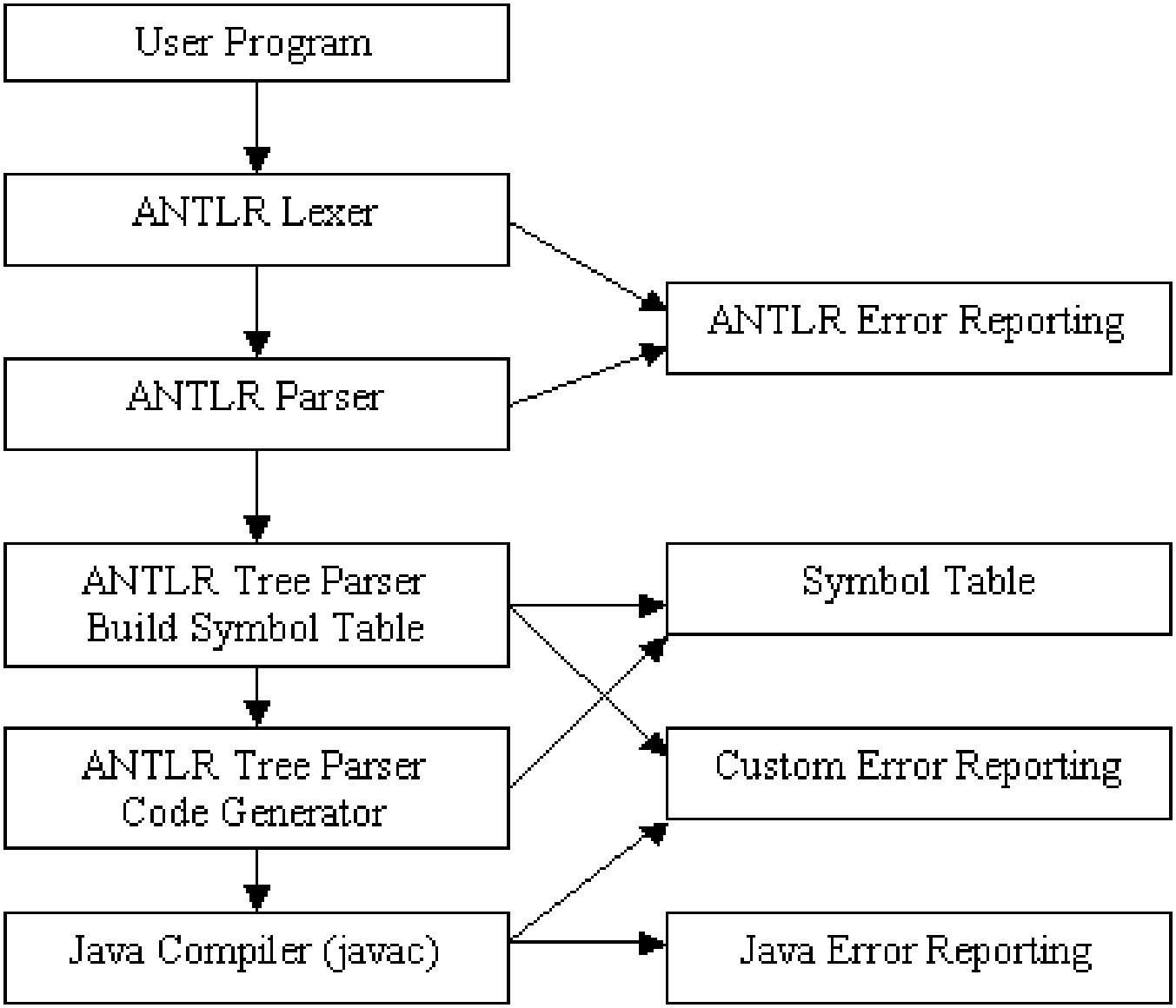
- In most operating systems, writing programs that interact with other programs is difficult.
- Graphical user interfaces are designed for humans, not programs.
- There are many conceivable situations for which it would be nice to be able to control other programs programmatically.
- Solution: Control the keyboard and the mouse through software.

The Concept

- The Macro Record Robot Language (MRRoboto for short) allows programmers to automate control of the keyboard and the mouse through simple batch programs.
- The syntax is Java-like, but simplified so that scripts can be written quickly.
- A good way to teach beginning programmers basic concepts such as flow control and modularization.

The Execution

- The ANTLR parser generator was used to create a grammar file that translates MRRoboto scripts into human-readable Java source code which can then be compiled and run.
- Translation consists of three steps: lexical analysis, syntax tree generation, and a tree-walking algorithm that performs semantic analysis and generates the code.


User Program
ANTLR Lexer
ANTLR Parser
ANTLR Tree Parser
Build Symbol Table
ANTLR Tree Parser
Code Generator
Java Compiler (javac)
ANTLR Error Reporting
Symbol Table
Custom Error Reporting
Java Error Reporting

Some Examples

```
** a simple hello world program
mouseMove(10, 10)
click()
mouseMove(50, 10)
click()
type("notepad")
press("|ENTER|")
type("Hello World")
```

More Examples

```
** delays and flow control
int a
a = 3
wait(60000) **60 seconds
if(a < 10)
    moveAndDoubleClick(20,20)
else
    while(a < 5)
        moveAndClick(50,50)
    end
end
```

The Main Block and Procedures

```
string b
b = "foo"
doType(b)


procedure doType(string msg)
    type(msg)
end
```

Error Detection and Recovery

- Code that violates the grammar rules (bad token, incorrect syntax, etc.) will be detected by ANTLR during initial lexical analysis and parsing.
- Code that contains semantic errors (undefined symbol, type mismatch, missing return statement, etc.) will be detected by the semantic analyzer during the final stage of translation.

Future Directions...

- We had relatively little time to implement computer-vision functionality, other than some simple pixel color detection, so the ability of MRRoboto programs to actually interact with the screen is fairly limited.
- More sophisticated means of interaction, perhaps including automatic detection of system-native windowing components, seems to be a likely place for future expansion.