Unified FLEET Assembly Proposal

Greg Gibeling
Thursday October 26th, 2006
GDG05 – Unified FLEET Assembly Proposal

Okay, I've looked over the various versions of the language (ArchSim, AM05 and RDLC). Most of them are quite similar. Clearly the things we need are: codebag declarations, move instructions and import/include. I believe that the specification of the FLEET itself (the aliases) should be relegated to a separate file, since the various simulators will need different formats for this information. I propose the following syntax:

```
// Import/Include all of the declarations from the specified file
// as if there where in this file.  Similar to a C/C++ #include
Include AnotherFile.fleet

// Declare an initial codebag named CodeBagName
// There may be more than one initial codebag in a program
// A codebag may be declared non-initial by removing the word "initial"
initial codebag CodeBagName {
      // Move (once) literal 0 to input 0 (of the input array named
      // "Input") on the SHIP named "Barrier".
      move (0) -> Barrier.Input[0];
      move (0) -[1]> Barrier.Input[0];

      // Standing move from output 0 of the ship named "Barrier"
      // to the bit bucket.
      move Barrier.Output[0] -[*]> BitBucket;

      // Move 10 data items from port #0 of the output port array
      // "Data" on the SHIP named "Memory" to the port "Input" of the
      // third (#2) FIFO ship.
      move Memory.Data[0] -[10]> FIFO[2].Input;

      // Token move from the output "Output" of the third (#2) SHIP in
      // the "FIFO" SHIP array to the bit bucket.
      move FIFO[2].Output -[0]> BitBucket;
};

      Lexemes: "initial", "codebag", "move", "->", "-[*]>", "-[", "]>",
"BitBucket", ";", ".", "[", "]", "(", ")"
```

The choices here are simple: the use of ";" to denote block endings means the type of whitespace is immaterial, a nice feature with linux/dos linebreak differences. I find that putting the count inside the move array is slightly more easier to read, since it clearly separates this from the source and destination. The use of parentheses to mark literals (which includes numbers, codebag names and possible "true" and "false") is pretty well established, though not really necessary.

The use of the [] syntax for both arrays of SHIPs and arrays of ports on SHIPs makes it easy to write matching FLEET code and RDL. Of course it also hints at the

very common C/C++/Java syntax for class membership and arrays, making it relatively widely understood.

I believe that these choices match NONE of our existing languages, and so probably represent a good compromise. I will not offer a complete lexical spec. and grammar here, as there is little point until the debate has ended.