One criterion for my tree was to ensure that it was well organized. I'd say this accounted for 40% of my overall concern for the tree. A tree that is well organized leads to a language that is easier to read and understand. The nature of the language I'm implementing itself aided in ensuring that the flow of symbol definitions occurred in an intuitive manner. I tried to make sure the descriptions of nonterminal symbols followed shortly after their first occurrence in the tree definition.

Another primary concern of mine was to avoid any redundancy. This consumed about 30% of my effort. Within a file written in this distfile format, there are a few places where descriptions of things are the same. By structuring my AST in such a way that I was able to reuse symbols in these places, I've ensured some consistency in the language itself. This way, for example, any place where you might set attributes for a package, you can use the exact same methods.

My last concern for developing my AST was that the actual language would be easy to deduce by reading its specification. I'm often annoyed when BNF descriptions of languages seem cluttered and unorganized. I tried to write mine in such a way that portions of it resemble the actual language itself. This makes it not only easier for a user to learn the language, but also for someone to extend the language if the need arises.
RULE: Root ::= Document END;

RULE: Document LISTOF Package | Group | Host END;

RULE: Package ::= 'package' '(' PkgName ')' '(' PkgAttrList ')' END;

RULE: Group ::= 'group' '(' GrpName ')' '(' PkgEntList ')' END;

RULE: Host ::= 'host' '(' HostNameList ')' '(' HostArchList ClientsList ')' END;

RULE: HostArchList LISTOF HostArch END;
RULE: ClientsList LISTOF Clients END;
RULE: HostNameList LISTOF HostNameCon END;

RULE: HostNameCon ::= HostName END;

RULE: HostArch ::= 'arch' '(' ArchName ')' '(' PkgEntList ')' END;

RULE: Clients ::= 'clients' '(' HostNameList ')' '(' HostArchList ')' END;

RULE: PkgEntList LISTOF PkgEnt END;

RULE: PkgEnt ::= PkgName END;
RULE: PkgEnt ::= PkgName '(' PkgAttrList ')' END;

RULE: PkgAttrList LISTOF PkgSource | PkgDest | PkgSpecial END;

RULE: PkgSource ::= 'source' '(' Path ')' END;
RULE: PkgDest ::= 'dest' '(' Path ')' END;
RULE: PkgSpecial ::= 'special' '(' ShellScript ')' END;