We have implemented two code generators, one in html + JavaScript and one in Java. Our HTML consist of a dropdown for each attribute, containing all possible values. After selecting a value for each attribute, the JavaScript checks if it is a valid assignment. The Java code guides the user through each attribute, shows the possible values and then checks the assignment in the end. Both saves a valid assignment to a txt file.

The architecture of both solutions is similar; in each solution, constraints is converted into a set of language specific valid if statements. With help from a couple of generated helper methods, invalid assignments is removed from the possible values, the assignment is then validated by going through each attribute to check if the is still any values left and if the selected value is still available.

The actual conversion of constraints from our domain specific language to either JavaScript or Java is done with a recursive function “generateIfConstraintString”, the function traverses the binary tree and builds a “if statement”.   
SMDP2015\configproject\xtext\org.xtext.example.smdpdsl\src\org\xtext\example\mydsl\generator\ JavaCodeGenerator.xtend (252-272)

This function uses two helper methods; the first, “ConvertAttributeName”, retrieves the value the user has selected for a given attribute, the other, ”convertOperand”, converts our operands from our DSL specific operand type (can, has i.e.) to the language specific equivalent.

For JavaScript, ConvertAttributeName looks like this  
SMDP2015\configproject\xtext\org.xtext.example.smdpdsl\src\org\xtext\example\mydsl\generator\ JavaScriptCodeGenerator.xtend (289-294)

It’s using a HTML5 dom selector to get the selected value and convert it to a double if it’s expected.   
The equivalent in the Java, selects the value from a HashMap.  
 SMDP2015\configproject\xtext\org.xtext.example.smdpdsl\src\org\xtext\example\mydsl\generator\ JavaCodeGenerator.xtend (344-349)

The then part of the if-statement is converted into code that removes invalid values in the recursive method “generateThenConstraintString”. This is again using ”convertOperand” to convert the operand, the generated code is using a hardcoded function to remove the values at runtime.

SMDP2015\configproject\xtext\org.xtext.example.smdpdsl\src\org\xtext\example\mydsl\generator\ JavaCodeGenerator.xtend (313-334)