Automated provision of a generic modelling system

P Hale

Introduction

The SEEDS team is involved in cost modelling solutions, particularly for aerospace. We have been working with Rolls-Royce aerospace and Airbus. This presentation demonstrates how a cost modelling Ontology can be used to automatically produce cost models.

User Driven Programming

Software development is time consuming and error prone because of the need to learn computer languages. If people could instruct a computer without this requirement they could concentrate all their effort on the problem to be solved. This is User Driven Programming. This research aims to create software that enables people to program in natural language. Users enter information in a tree diagram. The program translates this human readable representation into computer languages. I am applying this technique to aerospace engineering but it should be applicable to any subject.

The approach is illustrated in Figure 1. A Semantic editor, Protégé, is used to create the Ontology for the cost models.

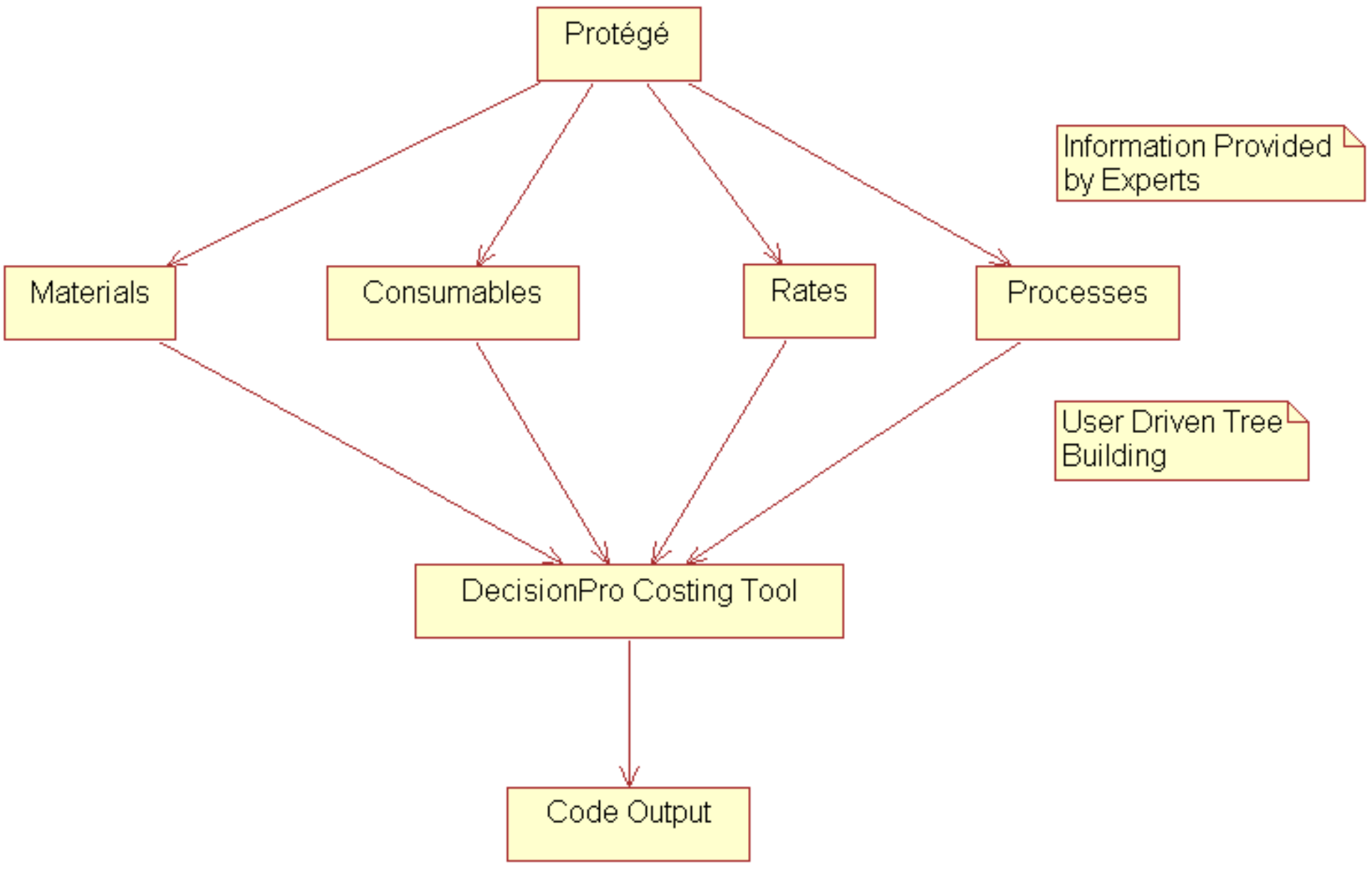

Figure 1

TRANSLATION

The Ontology representation can be translated into a computer model. An Ontology defines relationships between things. These relationships can be conveyed to a software model that evaluates the relationships.

To achieve this the translator requires

1 Search trigger(s) to be sent to it as a result of user actions, so users can convey their requirements.

2 Knowledge of the relationship between an item and its' immediate siblings e.g. parents, children, attributes

3 Ability to read a list of functions held in a standardised mathematical form. These can be held as a structured mathematical description within the Ontology.

4 Rules of syntax for the language of the code to be output.

Figure 2 shows the Protégé user interface used to define a decision support model

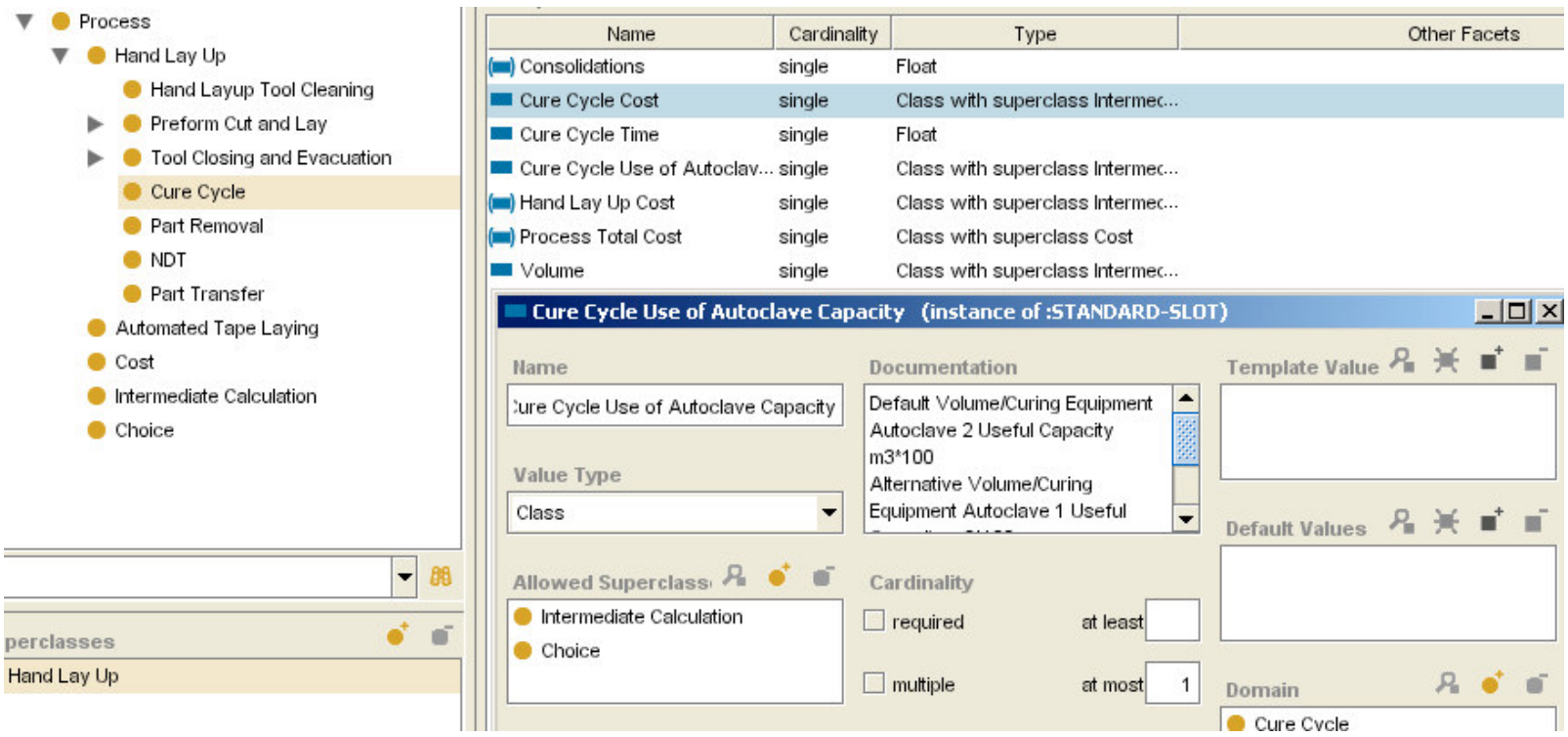

Figure 2

I use a structured syntax to allow users to make choices, Protégé and DecisionPro interprets this to produce a button that allows users to make the decision. The syntax used is

Default choice 1
Alternative choice 2
.
.
Alternative choice n

Figure 3 Shows how this can be translated, evaluated and visualised in a Decision Support system. It also shows how the system carries out calculations as defined in the Protégé user interface. Volume of the spar is calculated by referring to its' properties.

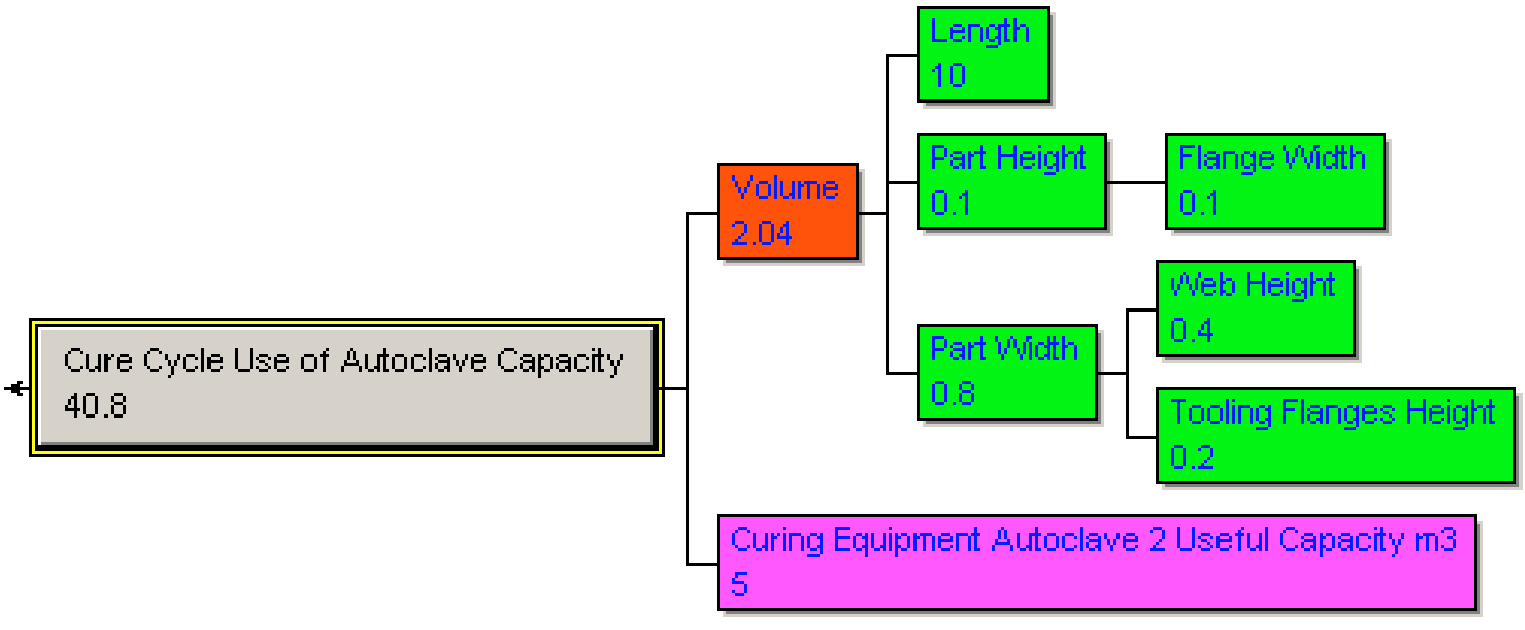

Figure 3

VISUALISATION

Figure 4 shows how the program visualises information for the spar and its' part definition, material, manufacturing processes etc.

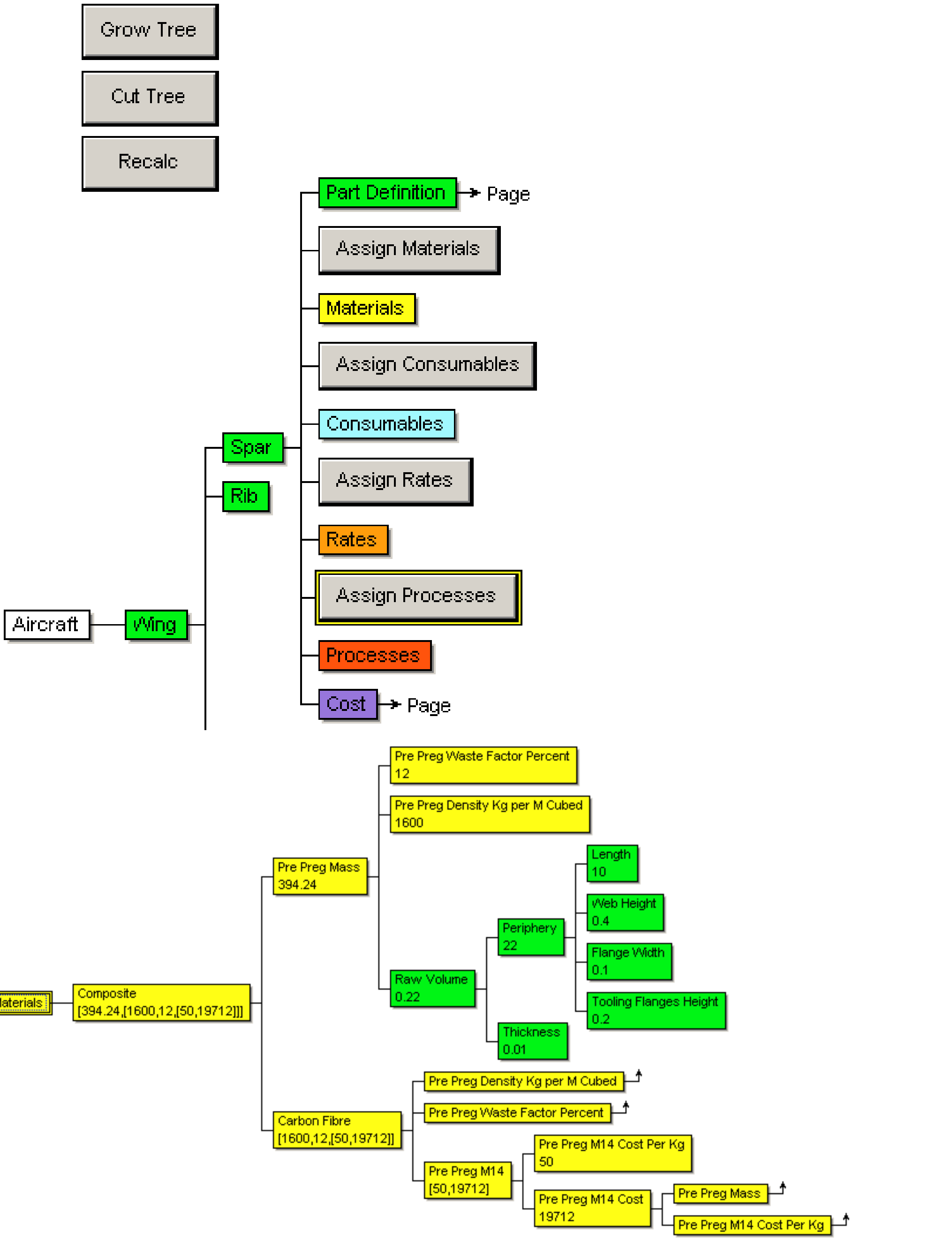

Figure 4

FURTHER OUTPUT

Figure 5 shows the tree exported into XML format and visualised on the Web

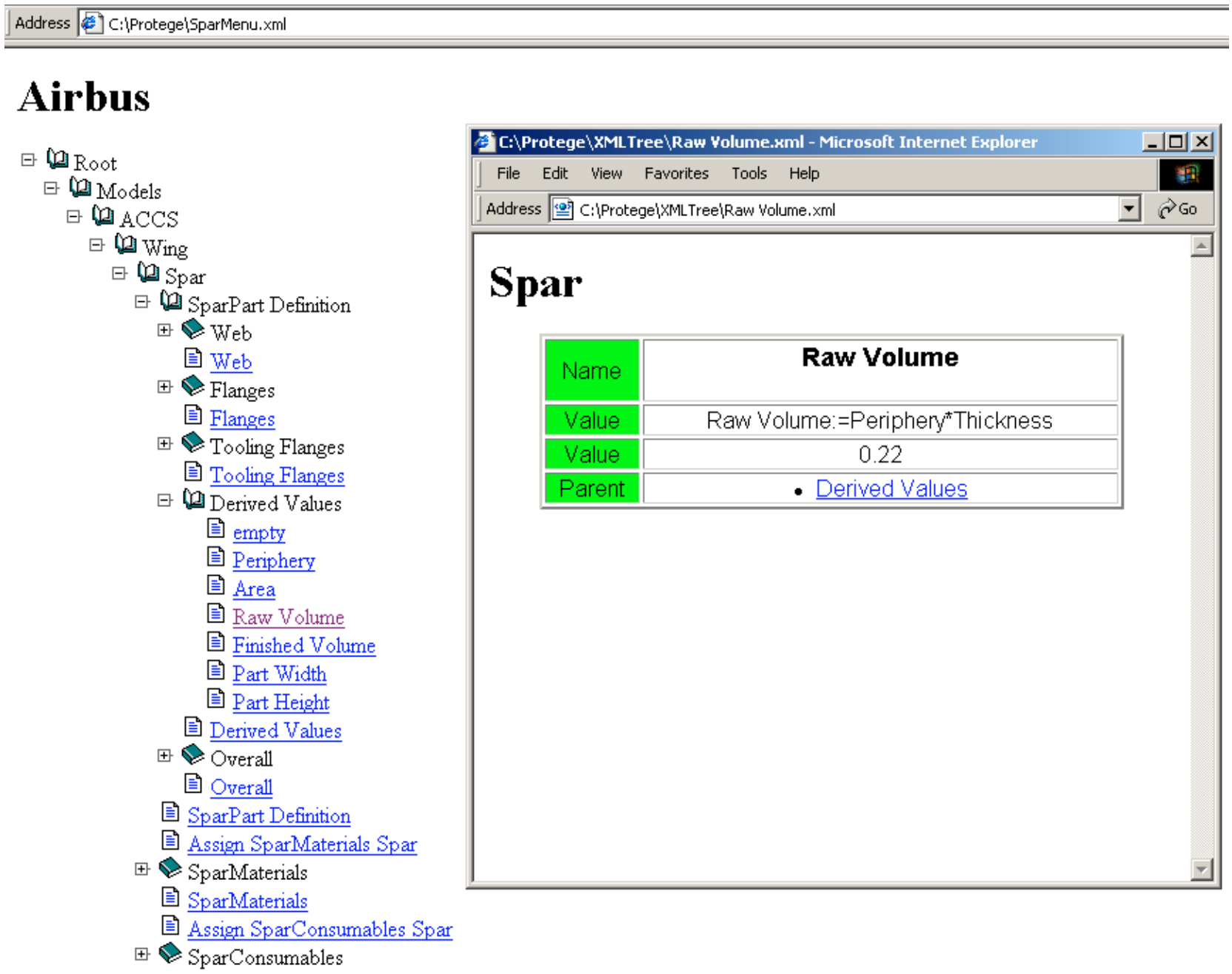

Figure 5

Figure 6 shows the spar DecisionPro tree exported into Java code

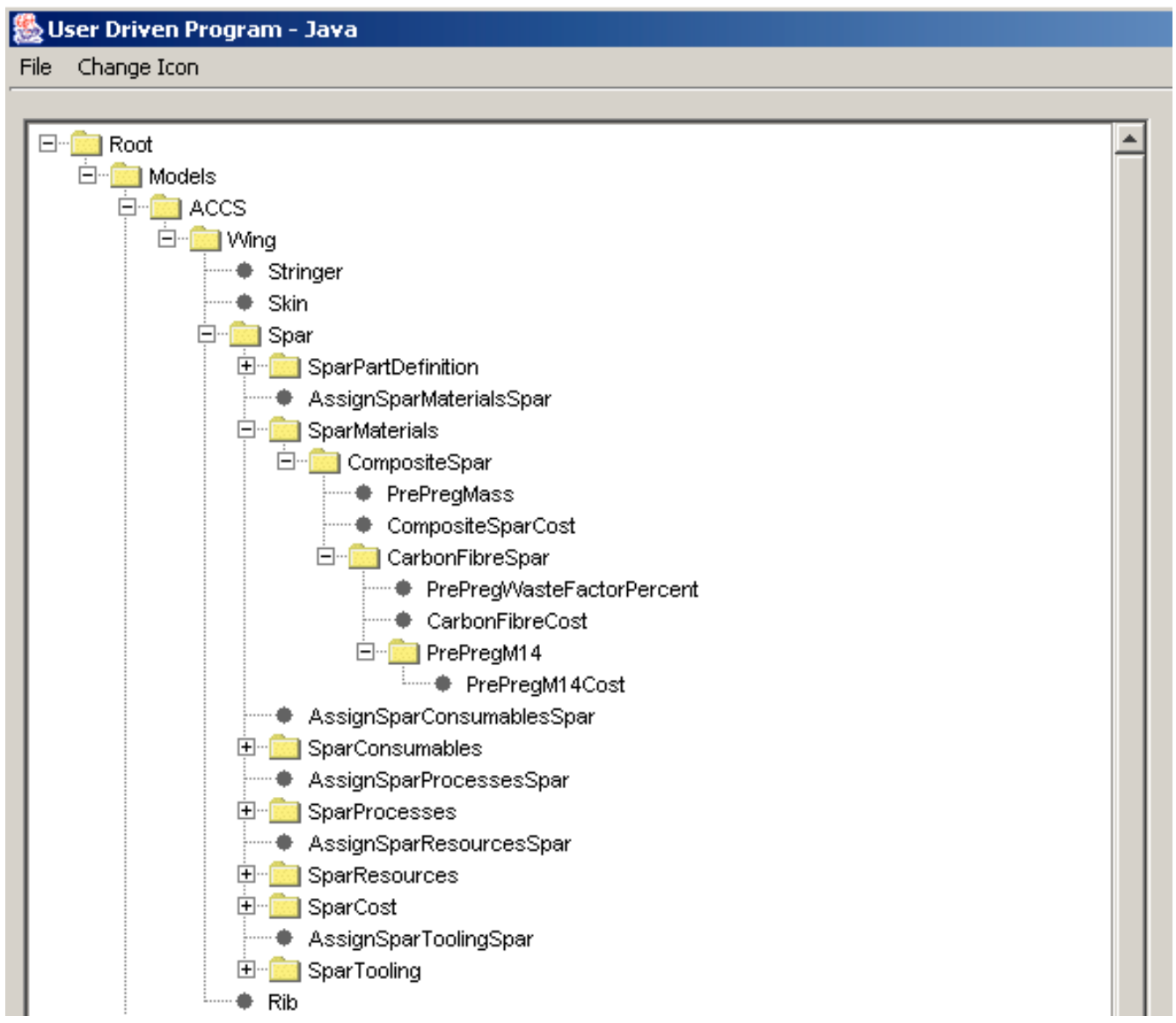

Figure 6