User Driven Programming

PhD Poster

Peter Hale
Home page http://www.cems.uwe.ac.uk/~phale/

Introduction

- SEEDS (Systems Engineering Estimation and Decision Support) team - involved in cost modelling solutions.

End User Programming

- Why do we need to make it easier for end users to program?

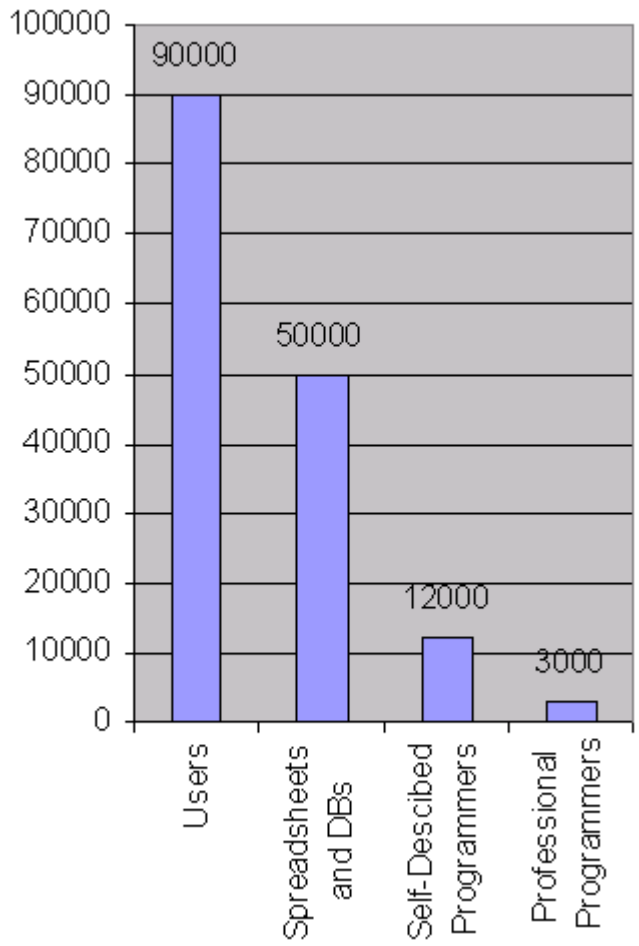

Figure 1 - End User Programming

Based on data from US bureau of Labour Statistics. Sources - http://www.cs.cmu.edu/~bam/papers/EUPchi2006overviewColor.pdf - Myers et al. Scaffidi, C., Shaw, M., Myers, B. (2005). Estimating the Numbers of End Users and End User Programmers, IEEE Symposium on Visual Languages and Human-Centric Computing, (VL/HCC'05): 207-214 Dallas, Texas.

User Driven Programming

- Software development is time consuming and error prone because of the need to learn computer languages.
- Mitigating this allows users to devote full effort to the problem to be solved.
- User Driven Programming creates software that enables people to program with visual representation of a tree diagram.
- Applied to aerospace engineering but should be applicable to any subject.

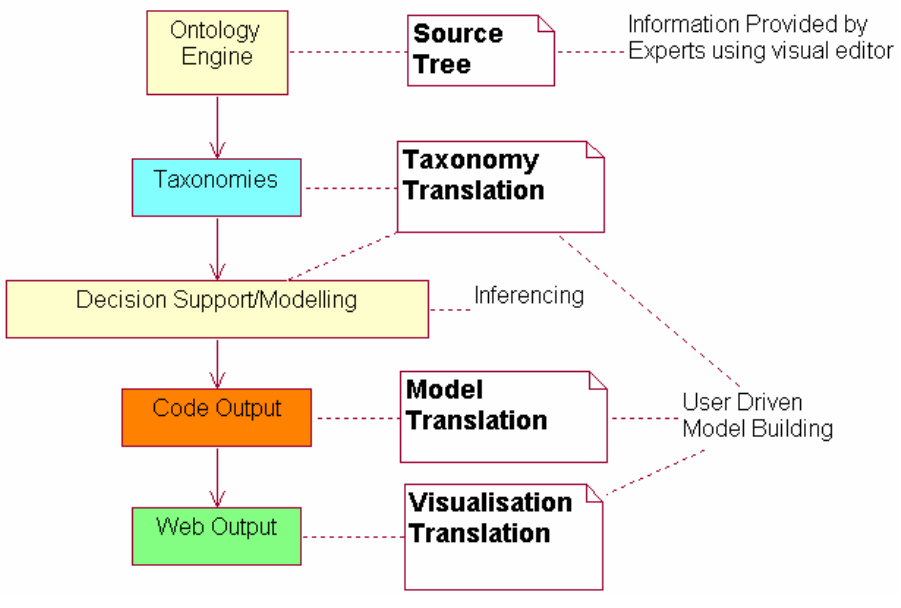

Figure 2 – Source to Result Tree Translation

Translation

- The Ontology representation is translated into a computer model.
- An Ontology defines relationships between things.
- Relationships can be conveyed to a software model that evaluates them.

To achieve this the translator requires -

1 Search trigger(s) resulting from user actions.
2 Knowledge of the relationships between nodes in the tree.
3 Ability to read a equations held in a standardised mathematical form.
4 Rules of syntax for the language of the code to be output.

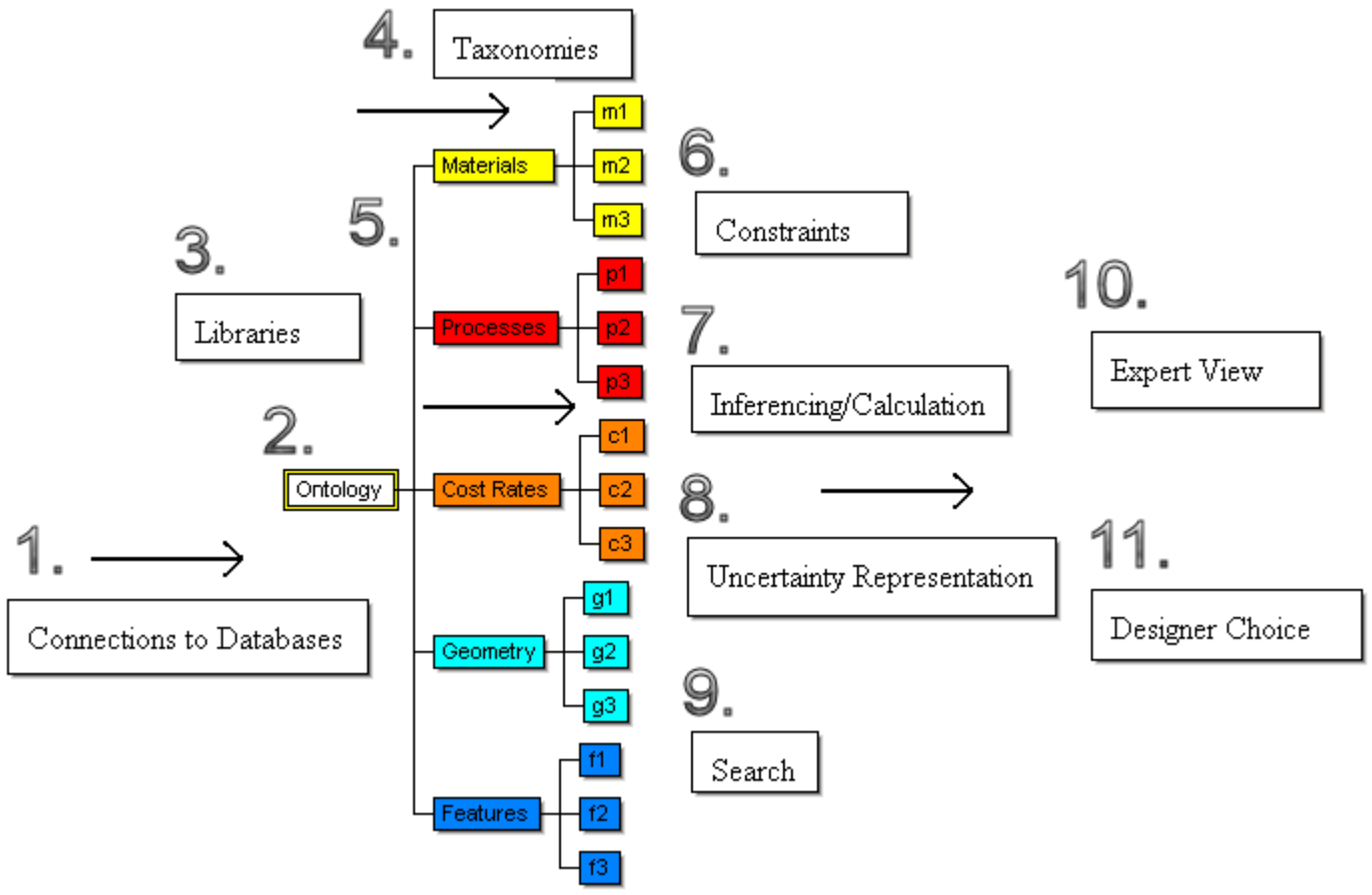

Figure 3 - Visualisation and Interaction Mechanism

Semantic Web Modelling system

1. Connections are established between the ontology system and any databases, spreadsheets, or other systems that hold relevant information for that modelling problem.

2. The ontology is created using RDF/OWL [2], and an interface built to allow domain experts to edit the ontology.

3. Libraries are created in a partnership between ourselves and domain experts.

4. Taxonomies are populated by model builders who want to use them for their modelling problem. These are based on the libraries created in step 3.

5. Taxonomies are colour coded for ease of understanding, this part of the diagram was built with Vanguard system (explained below). We have created a link between the ontology tool and this decision support and calculation tool. Vanguard system reads information from the ontology tool.

6. There are 2 sorts of constraints that can be used in order to make it easier for users to build and adapt models. These are constraints on the way the ontology, and models are built, and user interface constraints to reduce the scope for error.

7. The colour coding makes calculation clearer because all taxonomies can be used in any calculation, this results in a multicoloured result tree that represents the entire calculation history. User choices affect how items are related for the calculation; choices could be made manually or via a search. Colour can also be used to represent cost, time, or uncertainty.

8. Each node can also represent uncertainty, and we have prototyped including uncertainty expressions in the calculations.

9. The result tree can be represented on the web and in other programs, this allows for further searching, processing and evaluation of results. Visualisation techniques and the use of searchable languages such as XML, and SVG can assist in this.

10. and **11.** Experts such as designers can interact with the ontology, the model, and results, it's intended that there will be a two way feedback mechanism where the expert can make changes at any stage, and this filter into changed results. This can then support a cycle of results and rework.

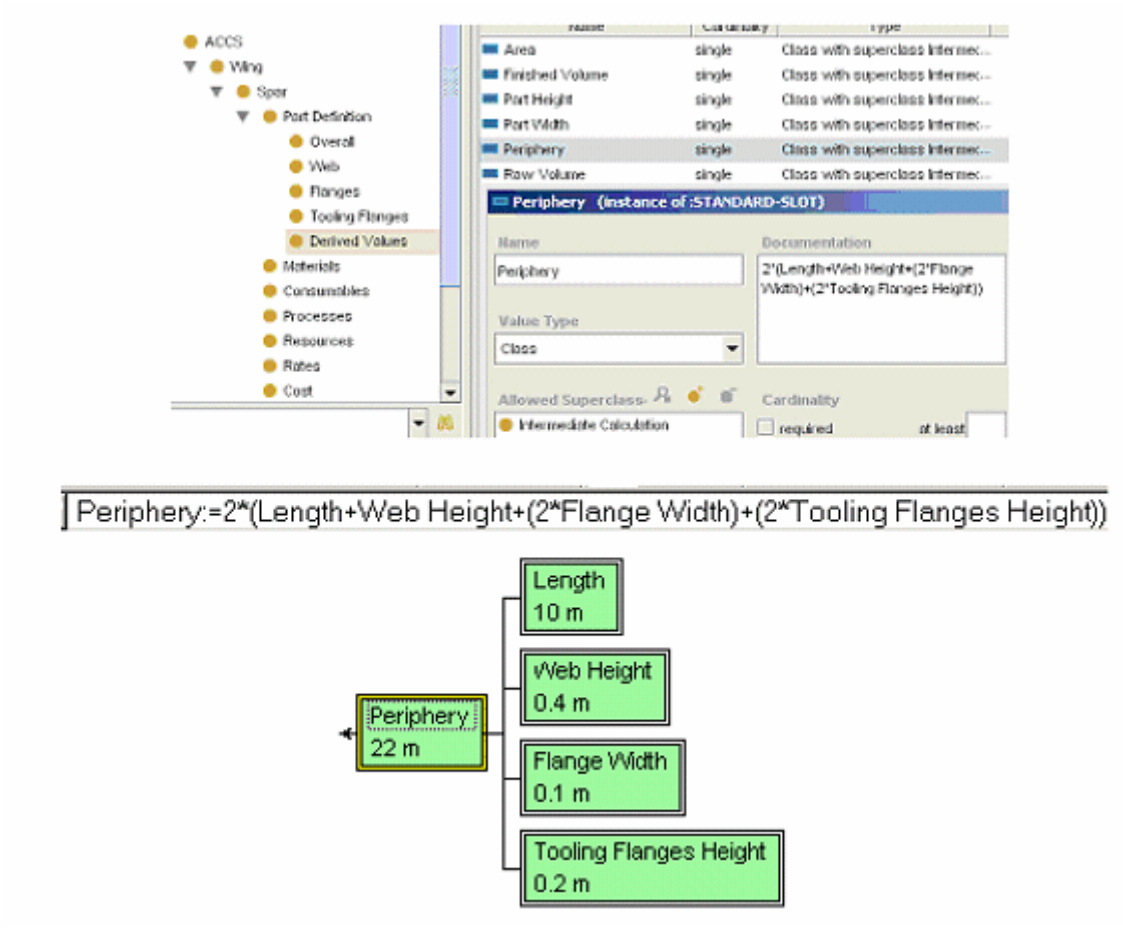

Figure 4 – Vanguard Studio Representation and Calculation

Visualisation

Figure 5 shows how the program visualises information for the spar and its' part definition, material, manufacturing processes etc.

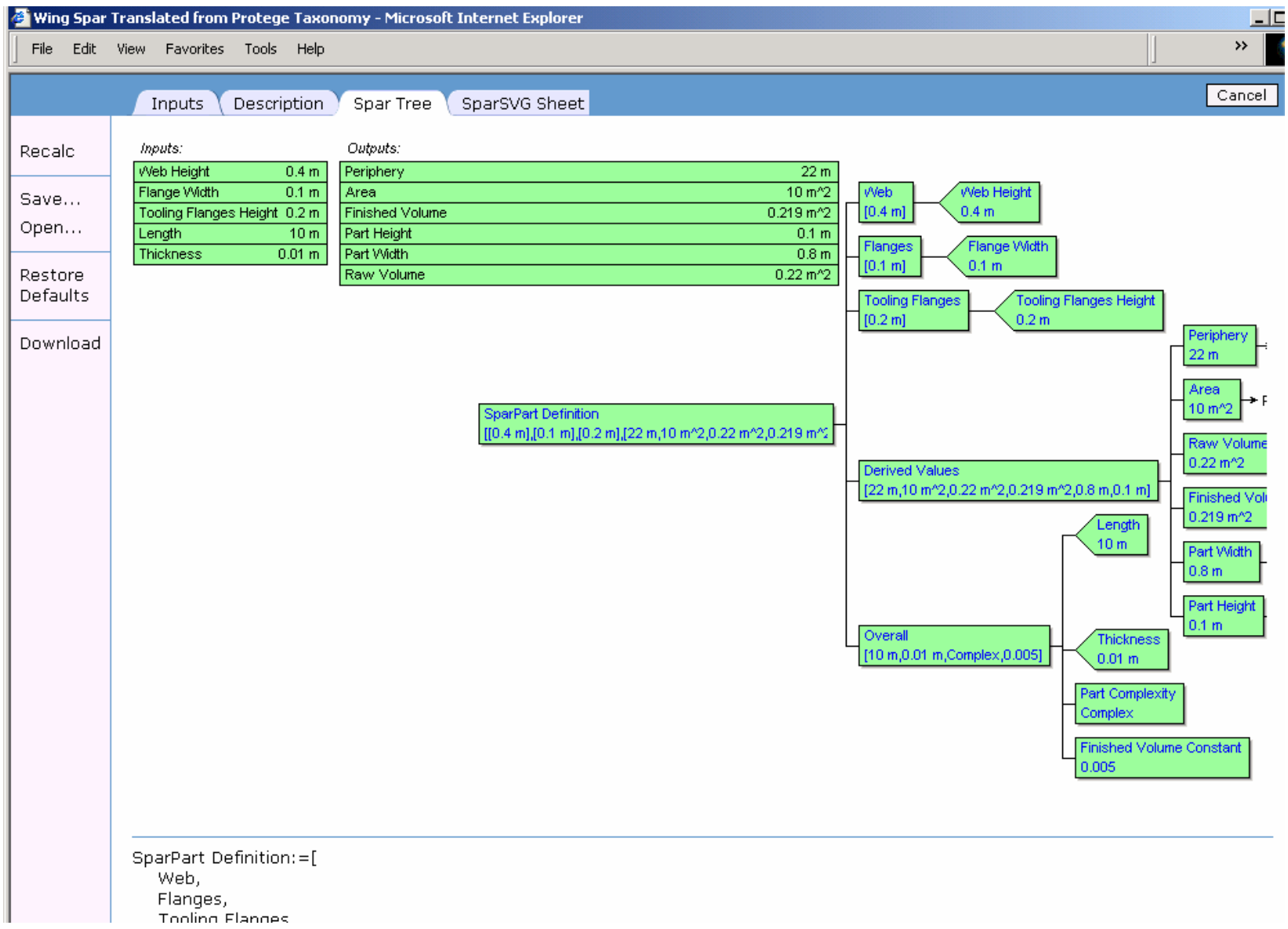

Figure 5 – Vanguard Studio Visualisation

- This is an interactive view of the Vanguard Studio model that was read from Protégé.
- Changing the figures and pressing recalc results in a new calculation.
- This tree is translated into SVG (Scalable Vector Graphics) and JavaScript for an interactive CAD style view.

Translations from Tree Based View to Component Diagram

Figure 6 shows the spar translated to XML and visualised using Flash multimedia. The spar is then translated and visualised in an SVG based interactive diagram.

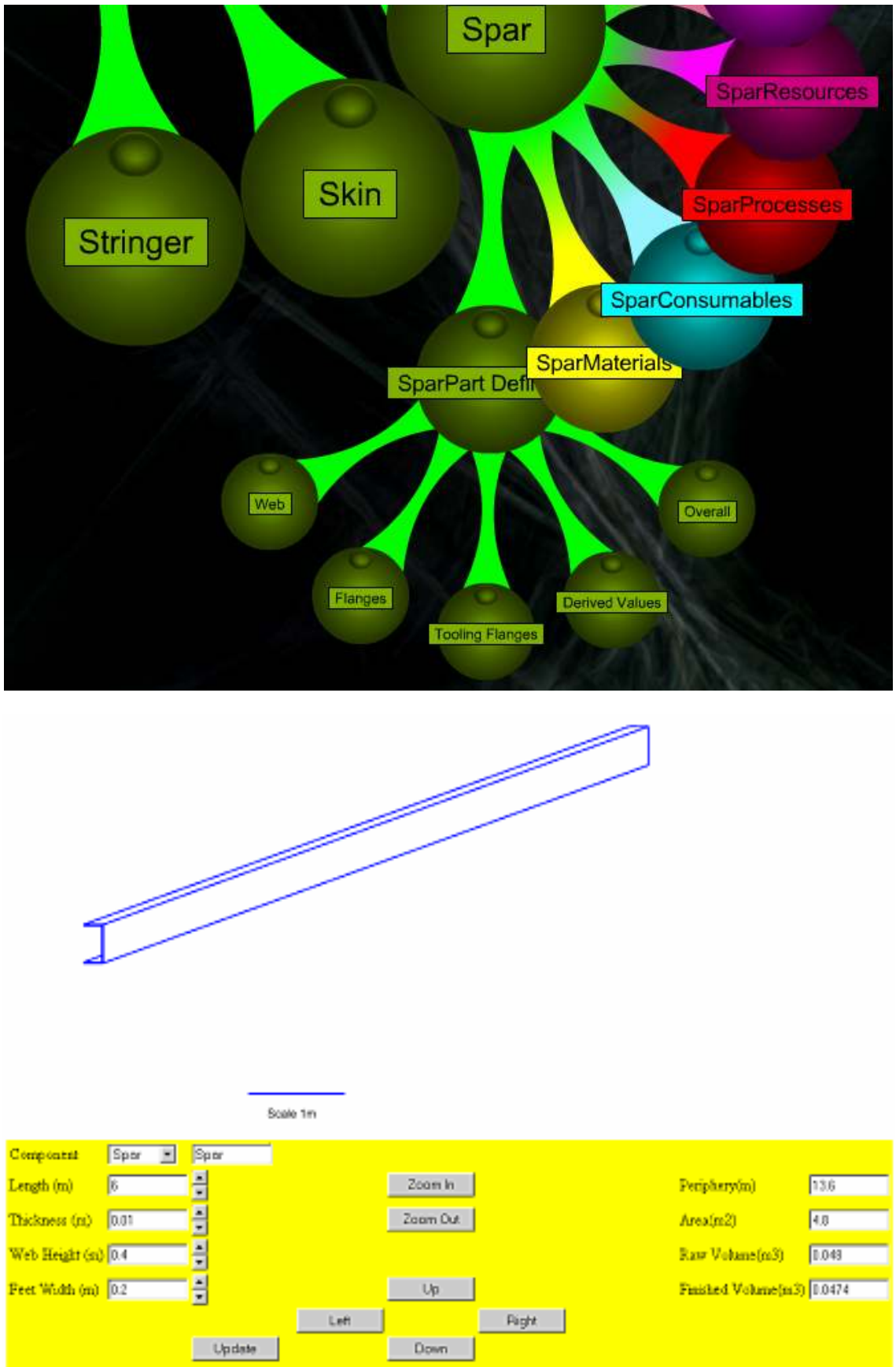

Figure 6 –Tree View - XML - Flash - SVG Representation

Others in this kind of research

Kurt Cagle -Understanding XML - http://www.understandingxml.com/

General Electric - ACUITy enterprise modelling tool -Jena Conference Paper - http://jena.hpl.hp.com/juc2006/proceedings/crapo/paper.pdf - An Ontology-Based Architecture for Adaptive Work-Centered User Interface Technology - A Aragones, J Bruno, A Crapo, M Garbias.

Jena Conference Proceedings - http://jena.hpl.hp.com/juc2006/proceedings.html

Orbeon - http://www.orbeon.com/ - Orbeon XForms Presentation Server.

Protégé - Conference and Project Information - http://protege.stanford.edu/community/conferences.html

UWE - Christophe Bru - http://www.cems.uwe.ac.uk/~cbru/

Vanguard Global Knowledge Portal - http://wiki.vanguardsw.com/

Chris Wallace - http://www.cems.uwe.ac.uk/~cjwallac/