

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, particularly for aerospace.
- Working with Rolls-Royce aerospace and Airbus.
- This presentation demonstrates how a modelling Ontology can be used to automatically produce representations.

End User Programming

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

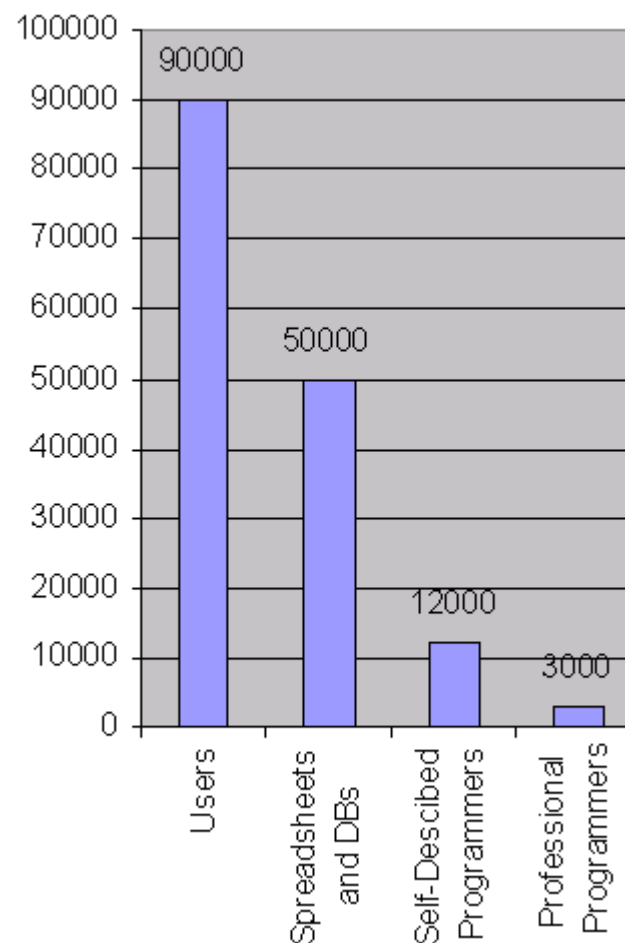


Figure 1 - End User Programming

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

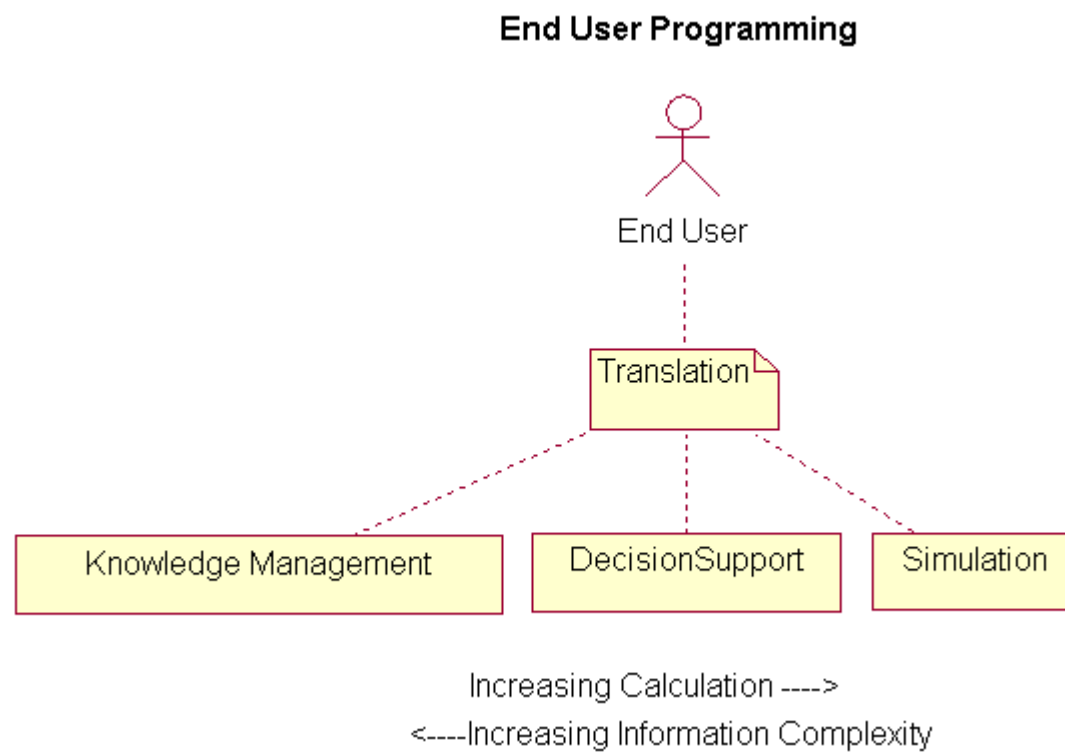


Figure 2 - User Driven Programming

- 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.
- The tree representation is translated into computer languages.
- Applied to aerospace engineering but should be applicable to any subject.

A Semantic editor, Protégé, is used to create the Ontology for the cost models.

The approach is illustrated in Figure 3.

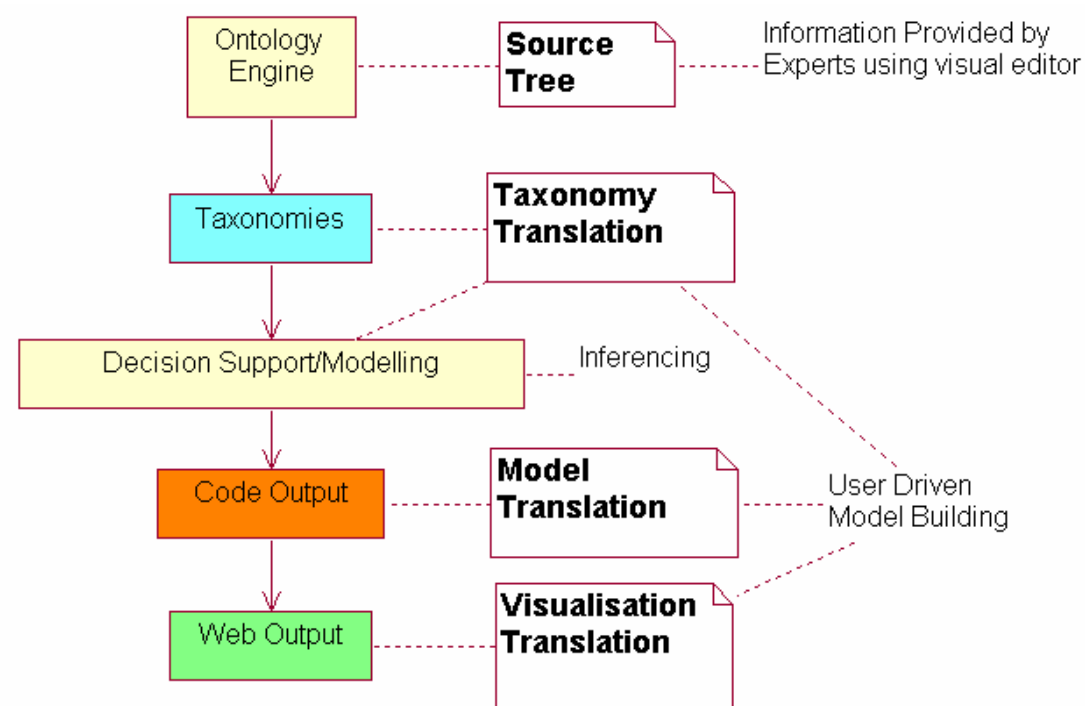


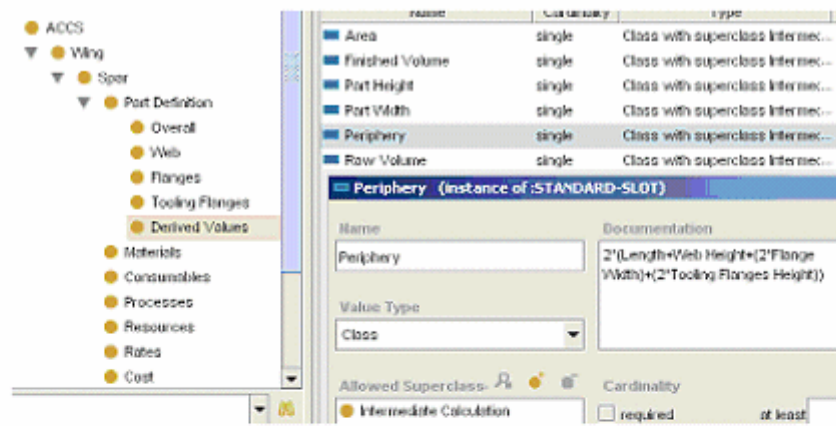
Figure 3 – Source to Result Tree Translation

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.



$$Periphery = 2 * (Length + Web Height + (2 * Flange Width) + (2 * Tooling Flanges Height))$$

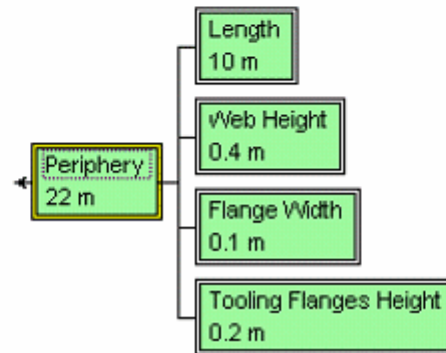


Figure 4 – Vanguard Studio Representation and Calculation

Visualisation

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

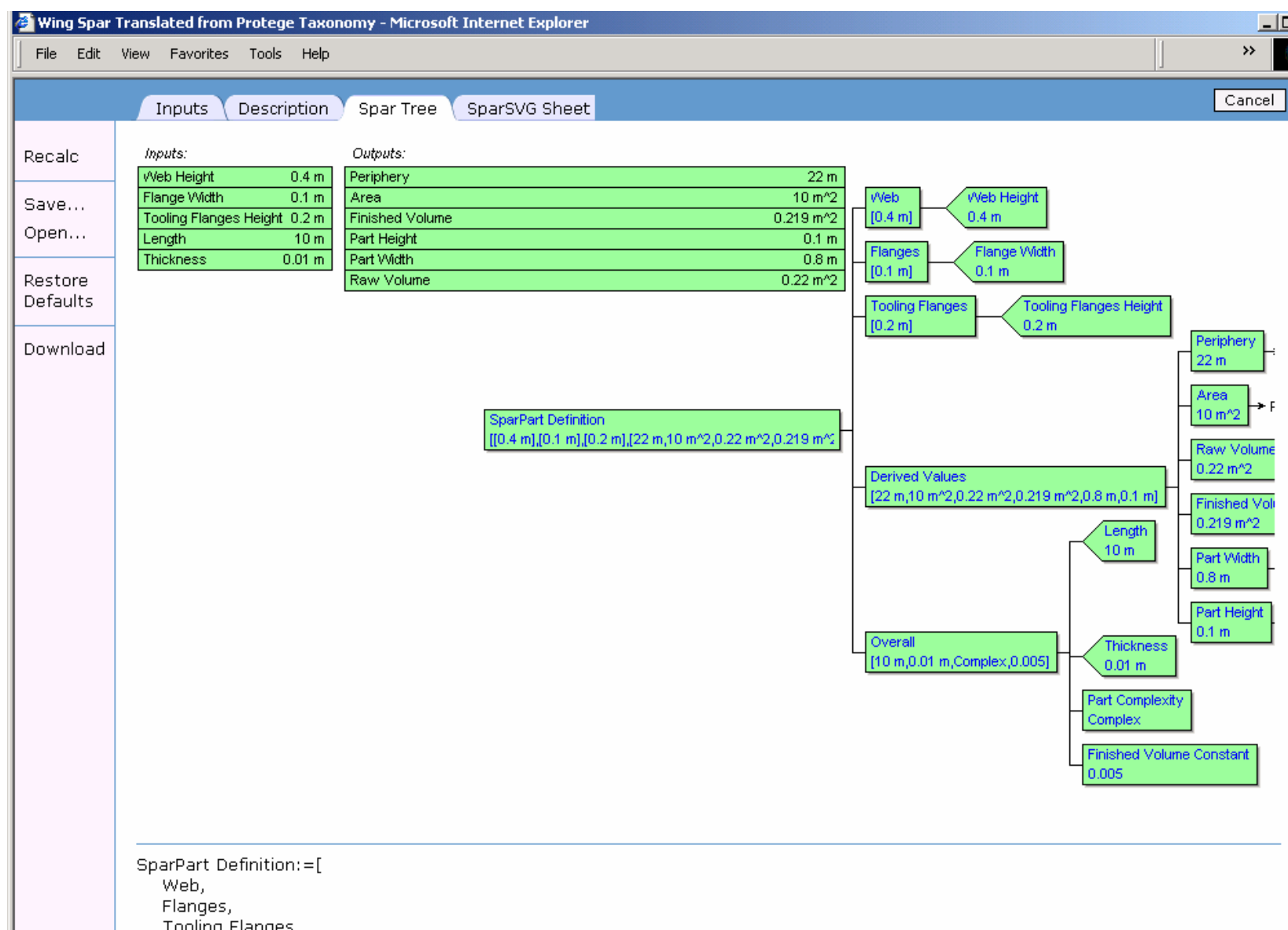


Figure 5 – Vanguard Studio Visualisation

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

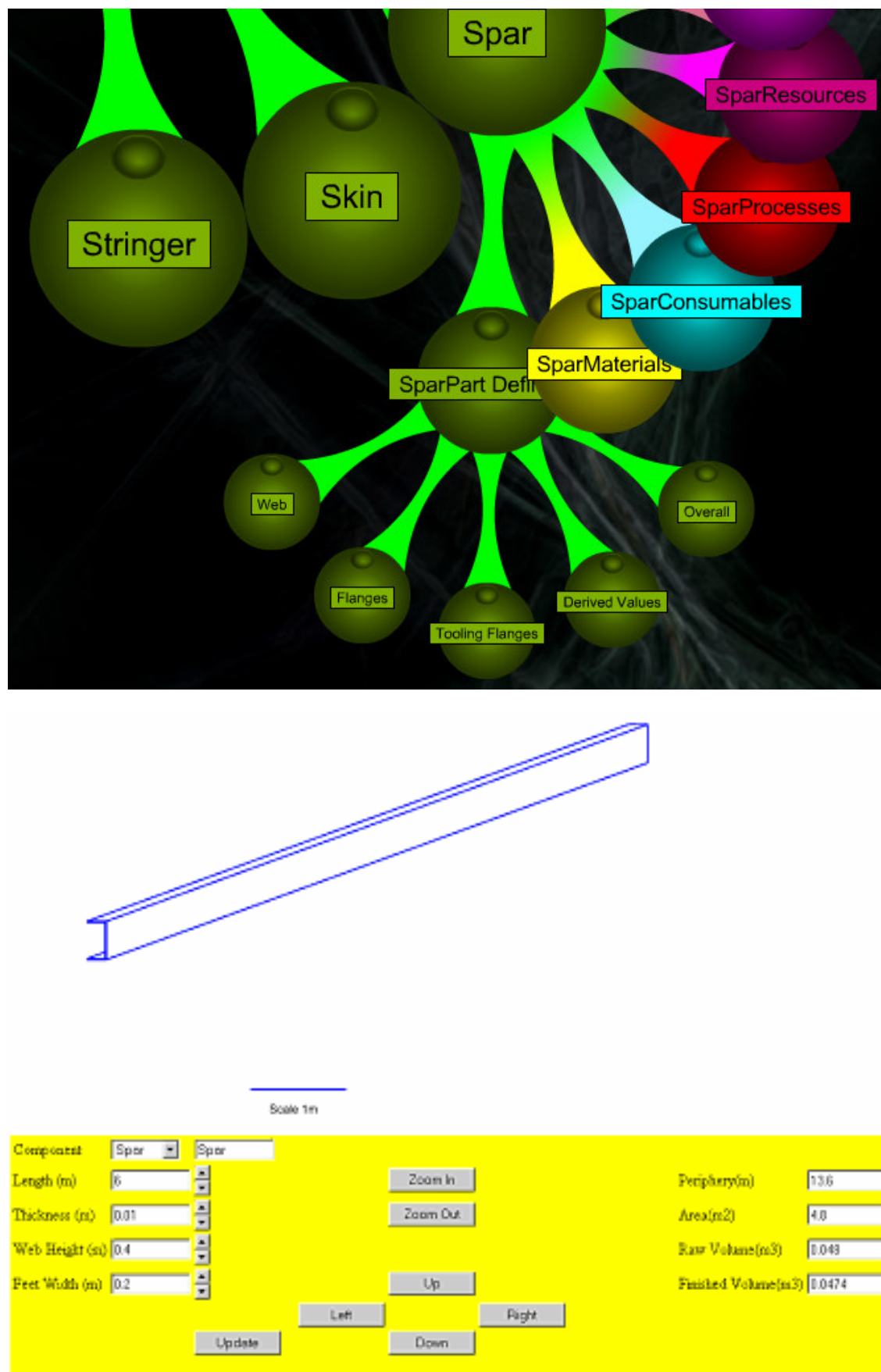


Figure 5 –Tree View - XML - Flash - SVG Representation

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 Aragonés, 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/>