Requirements for Phylogenetic Tree Visualisation - A User Driven Approach

Peter Hale, Tony Solomonides, Ian Beeson, Neil Willey, Karen Bultitude, Darren Reynolds University of the West of England

Research Question

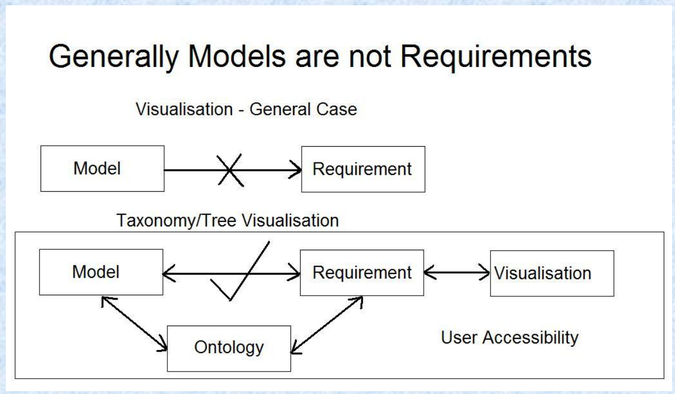
'To what extent is it possible to improve user-driven collaborative software development through interaction with diagrams and without requiring people to learn computer languages?'

Introduction

- Many computer literate people are experts in a particular domain, such as biology
- Enable computer literate scientists to model and visualise problems using software by minimising code writing
- User Driven Programming/Modelling

Problem Statement

- Software development is difficult for users - time, experience, access to programming tools
- Modelling with relationship tree and visualisation - possible to construct visualisation software for non programmers
- Methodology needed for creation of systems to enable collaborative end-user modelling/visualisation
- Methodology would use visualisation to allow scientists to model , visualise and debate taxonomies/phylogenies



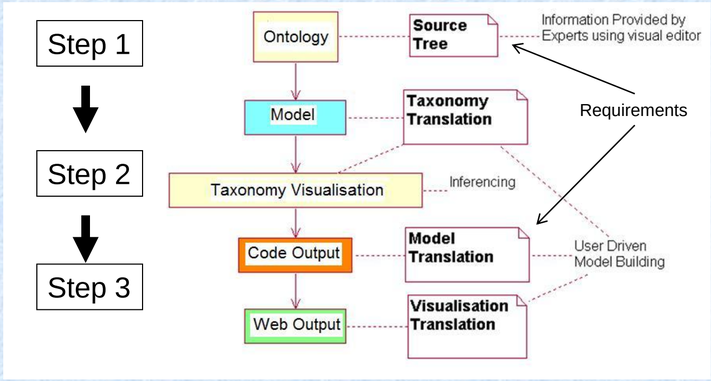
Step 1 → Step 2 → Step 3

Visualisation and Interaction Mechanism

Source to Result Translation

Theory

- Requirements can then be integrated into the visualisation development processes
- Visualised Ontology drives visualised interaction
- Step by Step Translation
- This Research arises from work with engineers on their product process taxonomies



Methodology

Translation

- Ontology representation is translated into a computer model
- Ontology defines relationships between things, apply this to Semantic Web
- Relationships conveyed to a software model that evaluates them
- Taxonomy visualised and output to web

Implementation

Stepped translation - Ontology to Modelling and Visualisations

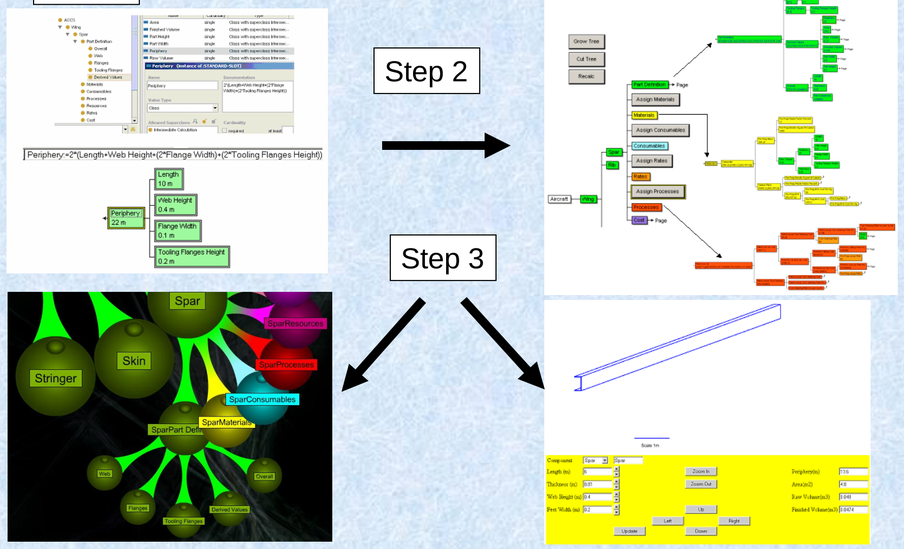


Figure 3 - Stepped Translation and Visualisation

Results

Improvement	Achieved By
Maintenance	Structuring and Translation
Extensibility	Structuring and Visualisation
Ease of Use	Visualisation, Interaction, and Translation
Sharing of Information	Shared Ontology and Interoperability

Recommendations

- Enable people to create software visually
- Create design abstractions familiar to domain experts e.g. diagrams for engineers
- Ensure interoperability using open standards
- Automate user to computer translation process

Conclusion

Can Widen programming participation by including computer literate non-programmers -

Enable interactive modelling/visualisation of problem