User Driven Modelling and Systematic Interaction for End-User Programming

Modelling for Engineering Processes

Peter Hale UWE

University of the West of England, Bristol

Abstract

Problem -
Enable translation of human problems/representation to computer models and code.

To what extent can diagrammatic representations of problems be used in order to provide modelling solutions.

Application Area -
Engineering Modelling, for manufacturing processes and cost, so far applied to Aerospace Composite Wing Box cost, and Aircraft Engine Design and Cost.

Introduction

Purpose -

To test this problem -

- C.S. Peirce (1906) -
- 'Prolegomena to an Apology for Pragmaticism'
- "Come on, my Reader, and let us construct a diagram to illustrate the general course of thought; I mean a system of diagrammatization by means of which any course of thought can be represented with exactitude"

To limit the Scope –

- Research restricted mainly to engineers (who often use diagrams)
- To domain of modelling (which often requires diagrams)

Introduction Continued

Benefits -

- Enables engineers to visualise problems such as representation of a product data structure in a familiar way
- Gives a visual and colour coded representation of equations
- Visualisation is easier to navigate and understand than that in spreadsheets, and more maintainable

Wider Implications -

- This research could also be used for business modelling, all kinds of process modelling, and workflow

Research Approach

- This approach involves building a systematic infrastructure and capability, and solving problems which could hamper this
- And is based on creation of systems that can be customised to produce other systems and models, and translation from abstract diagrammatic representations to computer representations

Methodology

Engineering Modelling Design

- Making the structure of a model be the same as the structure of the engineering component modelled turns 2 problems into one
- This speeds up co-operation in prototyping of both the software model and the component

Systems Engineering Involvement

- Systems Engineering involved in analysis of the relating of interdisciplinary research requirements, in both engineering and computing
- Systems engineering important as application area of modelling, for aerospace (Airbus and Rolls-Royce) involves complex engineering and a systematic approach
- Required systematic production of systems that must be usable by wide range of users to produce and share customised engineering models

Engineering Modelling Requirements

- Both rapid prototyping and rapid application design/development involve iterative fast development with prototypes communicated

- Requirements emerge gradually as part of this process, so early stage design can begin, in co-operation with life-cycle management.

Engineering Requirements Soft Systems

- To get full benefit from this all staff who are part of this must be enabled to be involved
- Design, manufacturing, management, and life-cycle management people need to be able to access the models

Engineering Modelling User Participation

- Longer term aim is to enable direct modelling/prototyping by customers of the modelling tool e.g. engineers/end-user programmers
- Such a system documents itself as the structure of the engineering product and software model are displayed/visualised

Engineering Modelling Design Tools

- Integration of information representation UML (Unified Modeling Language) is progress towards this user participation

- Also a user interface is required that makes it easier for engineers to model using such a combined UML solution

Engineering Modelling Issues

- Despite object-oriented programming techniques being heavily influenced by the approach used by engineers for Bill of Materials/Product Data Structure modelling this link has become difficult
- Much of object-oriented programming was developed before graphical user interfaces became practical and common
- So objects/classes are often represented mainly by text with visualisation/representation being added as an afterthought

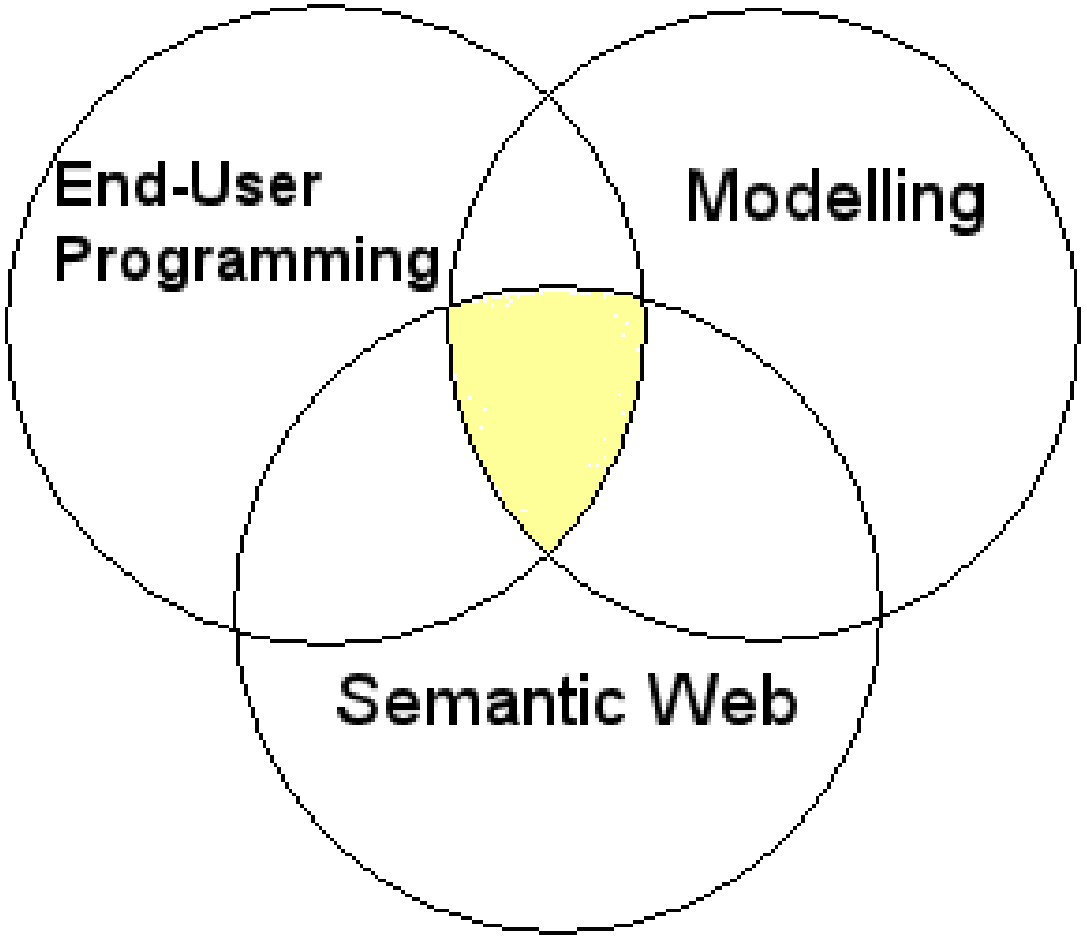
Engineering Modelling Issues 2

- This is not useful for engineers who are used to objects being physical things, or at least diagrams
- A further problem has been an over-emphasis on encapsulation (hiding an objects' details, while creating an interface for its use)
- This can lead to errors due to re-use of objects that are not fully understood
- So the classes/objects need to be visualised, even if user does not intend to change the contents, so user has sufficient understanding

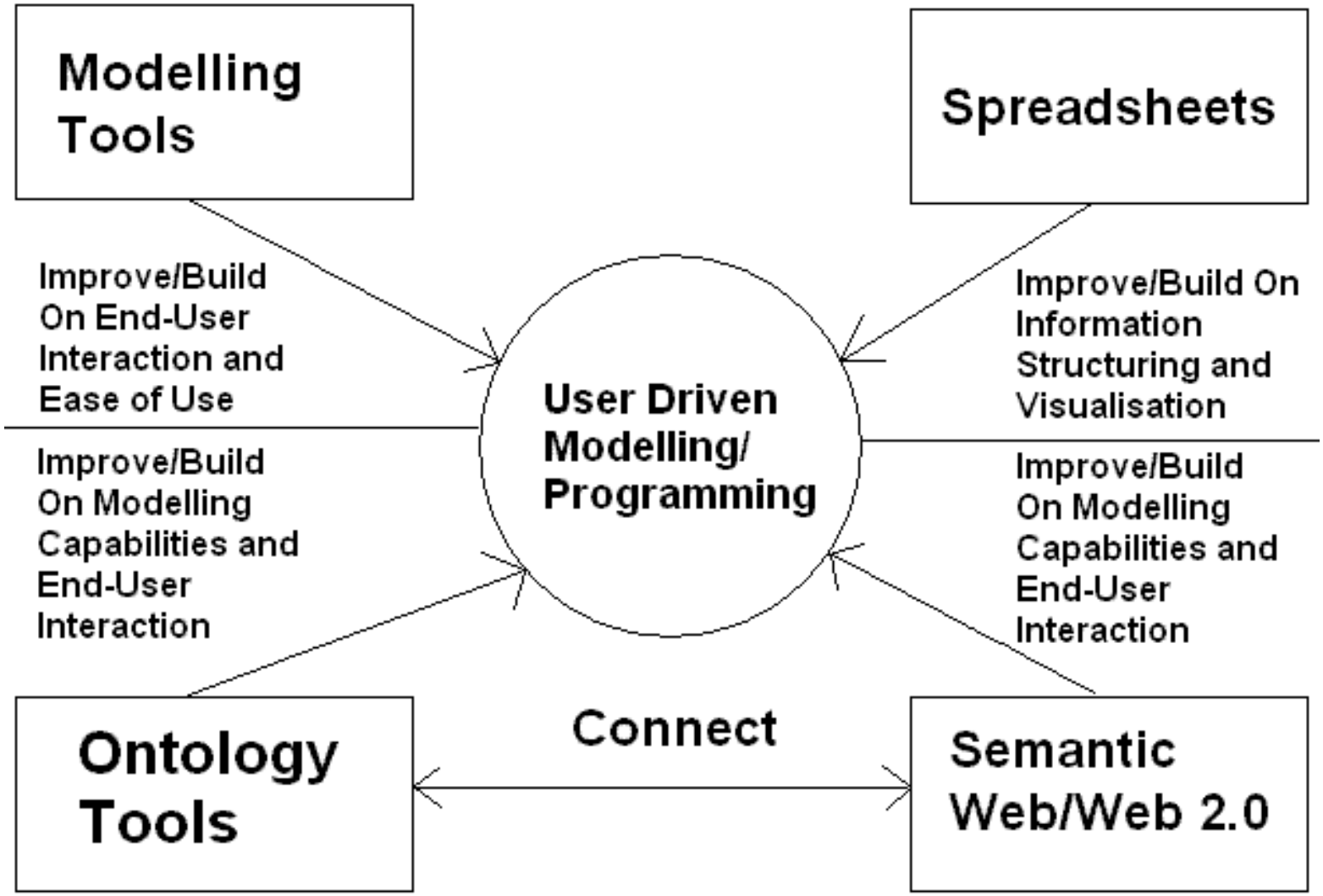
Engineering Modelling Conclusions

- It has been assumed that because engineers often deal with hard systems, a hard systems object-oriented approach would enable them to program
- But engineers spend much of the time involved with soft systems and interactions with others
- Needs to be more emphasis on allowing engineers to specify the problem at high level and this translated to code, rather than expecting engineers to code all the objects

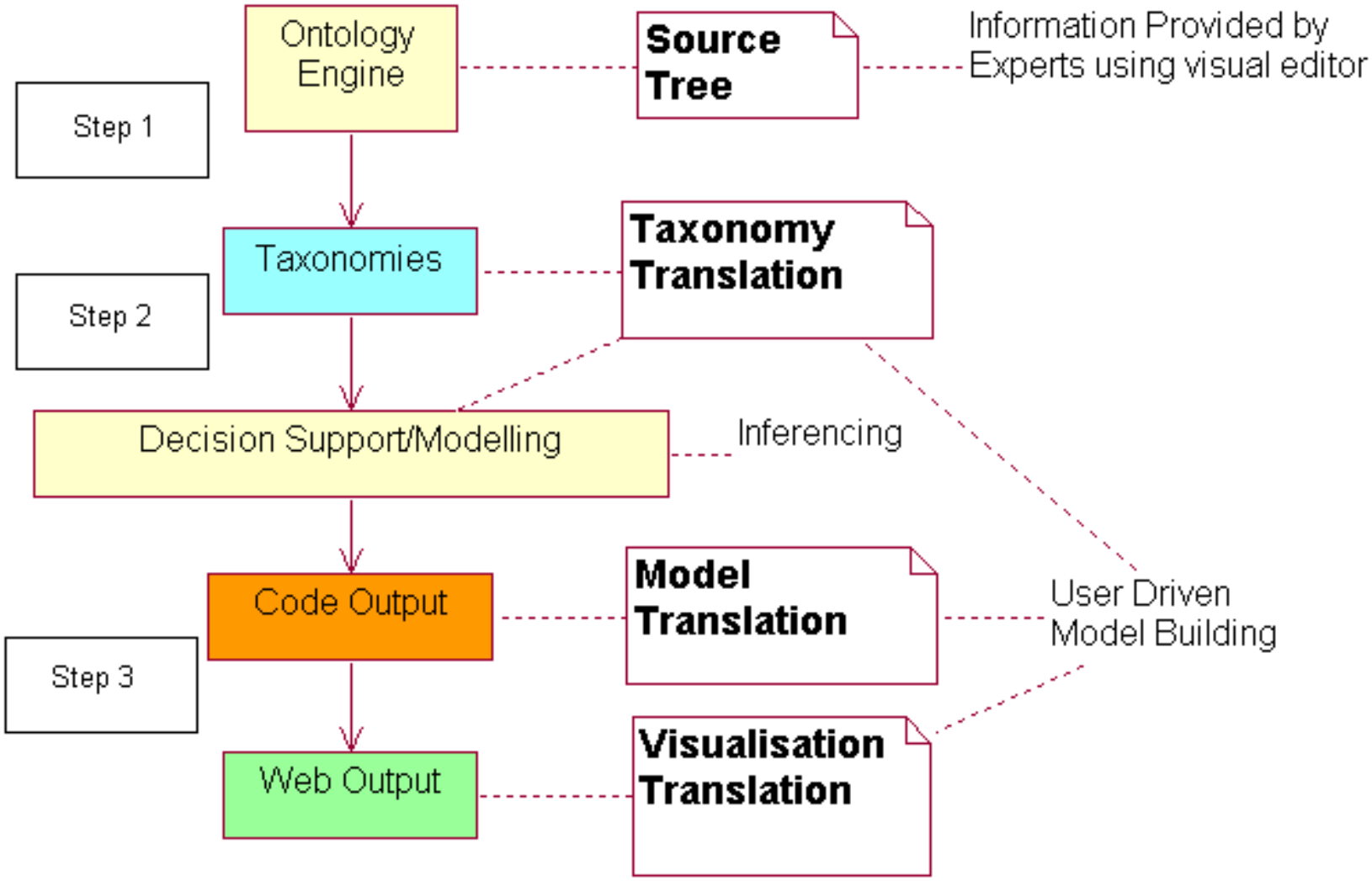
Application of Software



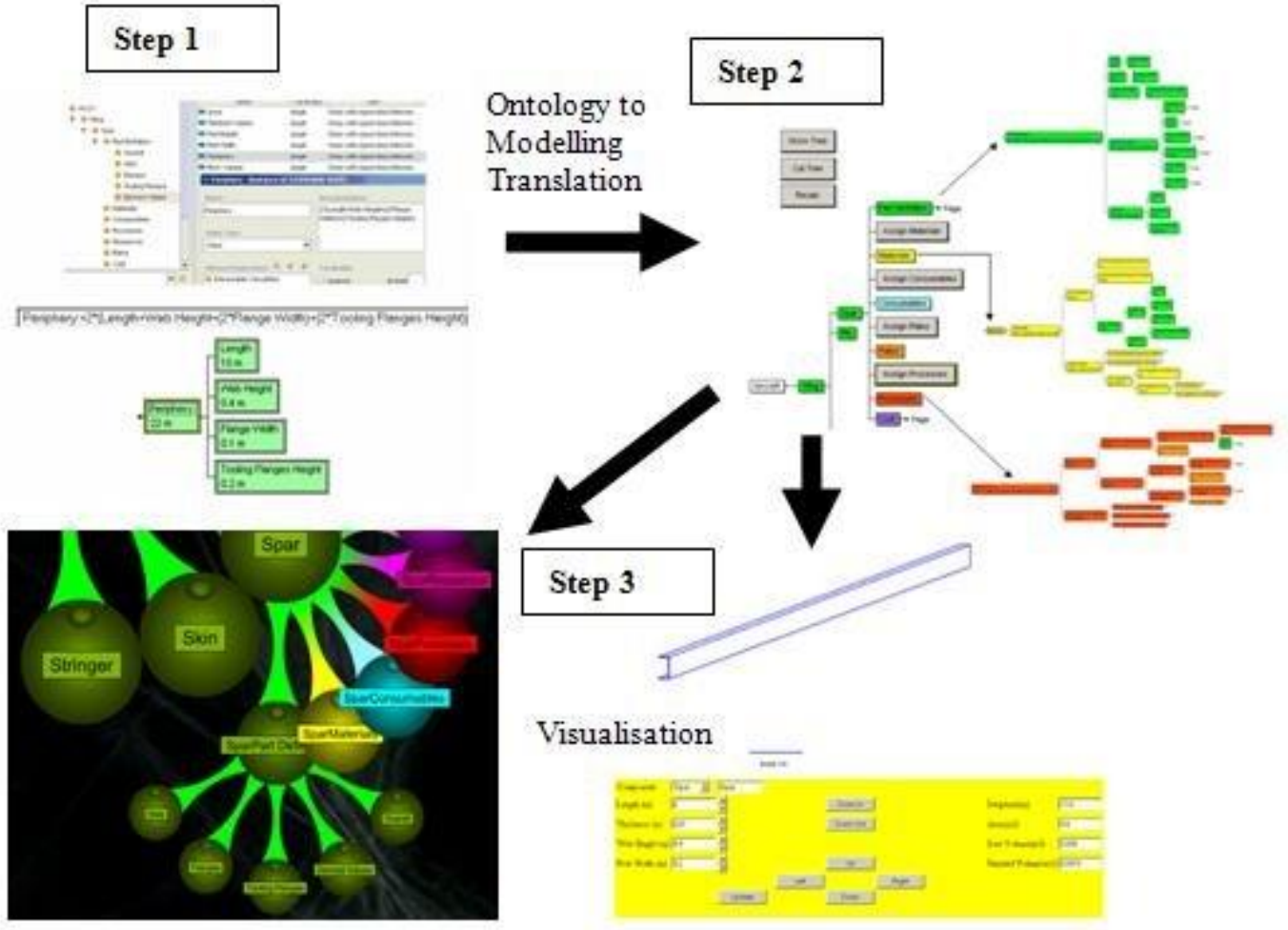
Tools and Technologies 2



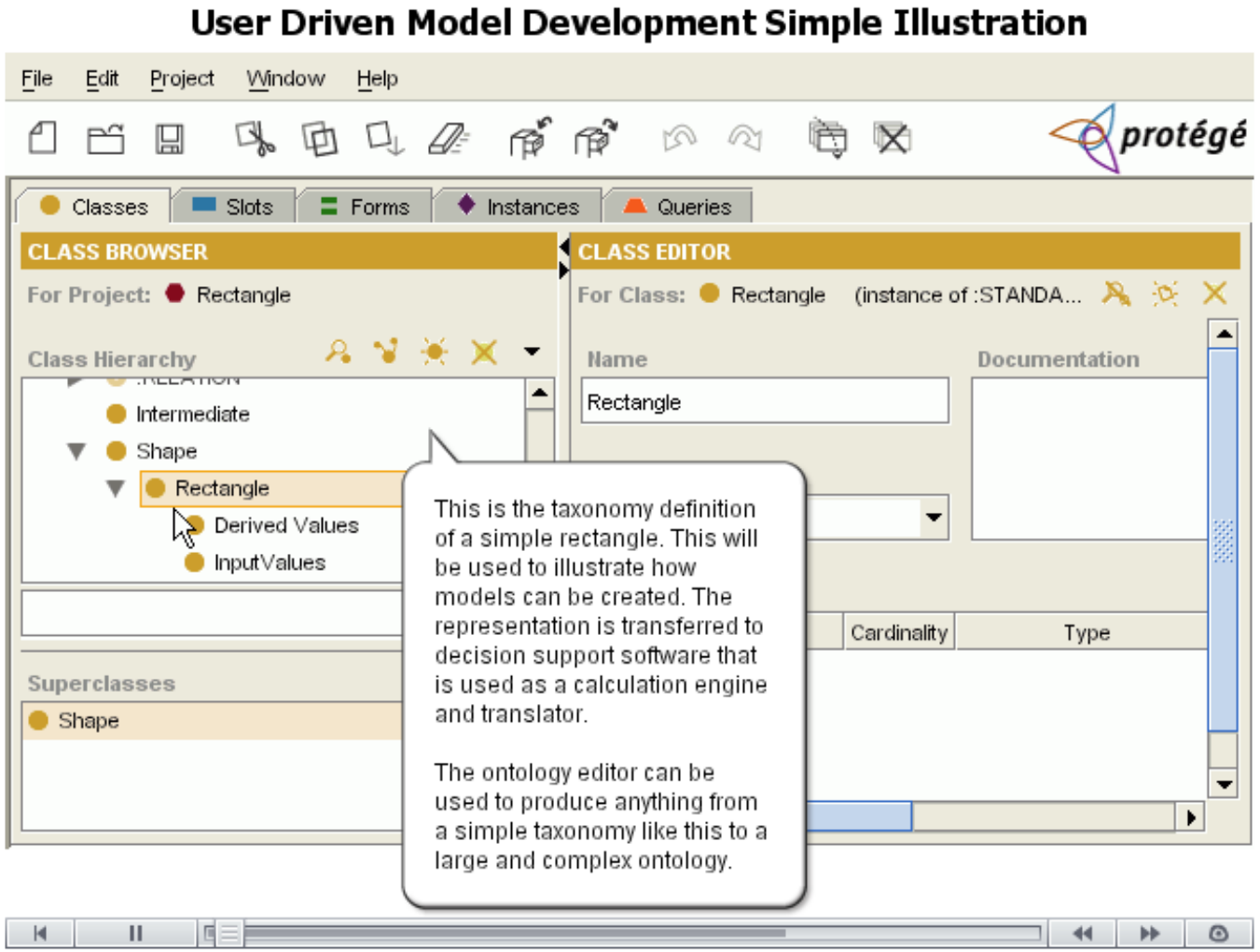
Translation Process



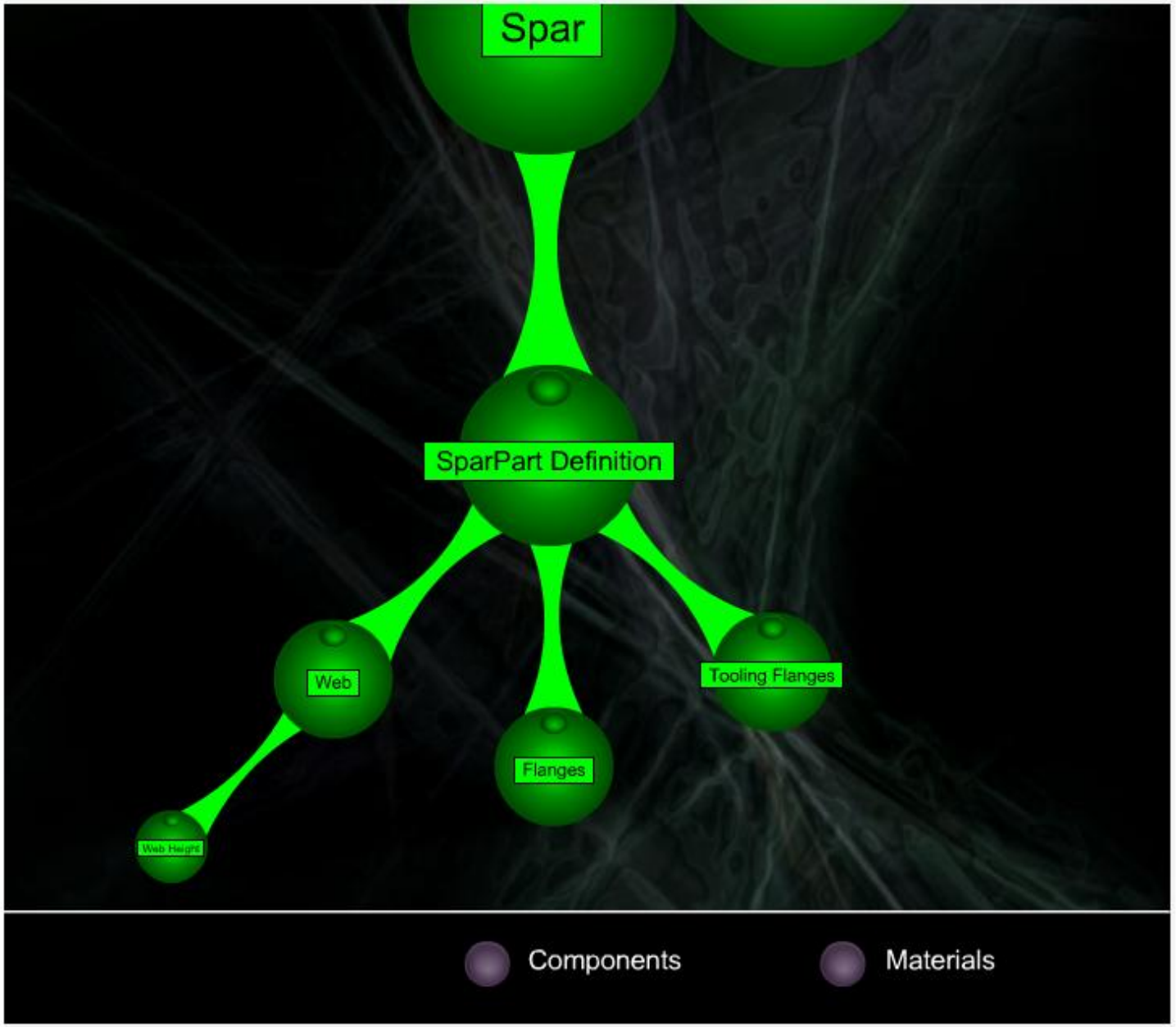
Translation Stages



Example Illustration



Web Tree Representation



Rhodes et al. 2002

Summary and Findings

- Closes the gap between those producing modelling systems, and those who require them
- Makes it easier to iterate through solutions and solve problems more quickly and collaboratively
- Experienced programmers can build a modelling environment that can then be used by non programmers to create process models
- Enables collaboration, simulation and modelling by translation from a model based representation of software to the actual software
- Gives users greater involvement
- Partially automates the process of software creation via a collaborative structure that maps the problem, and user interface creation by diagrammatic and/or tree based representation

Conclusion

- This approach to modelling and end-user programming enables interoperability, and collaboration
- This assists with Maintenance, Extensibility, Ease of Use, and Sharing of Information.

References

- Peirce, C.S. (1906) Prolegomena to an Apology for Pragmaticism [online]. Available from: http://www.existentialgraphs.com/peirceoneg/prolegomena.htm [Accessed 9 March 2010].
- Rhodes, G., Macdonald, J., Jokol, K., Prudence, P., Aylward, P., Shepherd, R., Yard, T., 2002. A Flash Family Tree, In: Flash MX Application and Interface Design Flash MX Application and Interface Design. ISBN:1590591585. [online]. Available from: http://www.friendsofed.com/book.html?isbn=1590591585 [Accessed 9 March 2010].