

# **Interactive Visualisation Tools for Supporting Taxonomists' Working Practice**

**Alan J. Cannon**

A thesis submitted in partial fulfilment of the requirements of  
Napier University for the degree of Doctor of Philosophy

**March 2006**

## Abstract

The necessity for scientists and others to use consistent terminology has recently been regarded as fundamental to advancing scientific research, particularly where data from disparate sources must be shared, compared or integrated. One area where there are significant difficulties with the quality of collected data is the field of taxonomic description. Taxonomic description lies at the heart of the classification of organisms and communication of ideas of biodiversity. As part of their working practice, taxonomists need to gather descriptive data about a number of specimens on a consistent basis for individual projects. Collecting semantically well-defined structured data could improve the clarity and comparability of such data. No tools however currently exist to allow taxonomists to do so within their working practice.

Ontologies are increasingly used to describe and define complex domain data. As a part of related research an ontology of descriptive terminology for controlling the storage and use of flowering plant description data was developed.

This work has applied and extended model-based user interface development environments to utilise such an ontology for the automatic generation of appropriate data entry interfaces that support semantically well defined and structured descriptive data. The approach taken maps the ontology to a system domain model, which a taxonomist can then specialise using their domain expertise, for their data entry needs as required for individual projects. Based on this specialised domain knowledge, the system automatically generates appropriate data entry interfaces that capture data consistent with the original ontology. Compared with traditional model-based user automatic interface development environments, this approach also has the potential to reduce the labour requirements for the expert developer.

The approach has also been successfully tested to generate data entry interfaces based on an XML schema for the exchange of biodiversity datasets.

## Contents

<b>Chapter 1 Introduction</b>	<b>1</b>
<b>1.1 Taxonomic working practice</b>	<b>1</b>
<b>1.2 Aim of Research</b>	<b>2</b>
<b>1.3 Organisation of Thesis</b>	<b>3</b>
<b>Chapter 2 Taxonomy</b>	<b>5</b>
<b>2.1 Introduction</b>	<b>5</b>
<b>2.2 Introduction to Taxonomy</b>	<b>5</b>
2.2.1 Classification	6
2.2.2 Character Concepts	7
2.2.3 Linnaean taxonomy	8
<b>2.3 Analysis Of The Taxonomic Process</b>	<b>9</b>
2.3.1 Introduction	9
2.3.2 Initial Preparations for a Taxonomic Project	13
2.3.3 The Crude Similarity Sort	13
2.3.4 The Detailed Sort	13
2.3.5 Completing the Project for Publication	17
2.3.6 Differentiation in the process due to the nature of the intended publication	18
2.3.7 Collaborative Working	19
<b>2.4 Taxonomic Description Issues</b>	<b>19</b>
2.4.1. Definition of Character	20
2.4.2 Character selection and definition	21
2.4.3 Unavailable data elements	23
2.4.4 Consequences	24
2.4.5 Summary	25
<b>2.5 Existing Computerised Tools for Taxonomy</b>	<b>25</b>
<b>2.6 Opportunities For Computerised Tools To Support Taxonomists' Working Practice</b>	<b>29</b>
2.6.1 General Requirements	29
2.6.2 Supported Tasks	31
<b>2.7 Conclusion</b>	<b>37</b>
<b>Chapter 3 Supporting Information Systems Literature Research</b>	<b>39</b>
<b>3.1 Introduction</b>	<b>39</b>
<b>3.2 Tailored Data Entry Interface Generation Tools</b>	<b>40</b>
3.2.1 Presentation Aids & Toolkits	41
3.2.2 Automatic Interface Generation	42
3.2.3 Ontologies	48
<b>3.3 Browsing databases</b>	<b>54</b>
<b>3.4 Visualising Structured Data</b>	<b>56</b>
3.4.1 General State of Visualisation Research	56
3.4.2 Interaction Techniques	58
3.4.3 Display Methods	59
3.4.4 Visualisation Summary	69
<b>3.5 Conclusion</b>	<b>70</b>
<b>Chapter 4 Capturing Description Data for Taxonomic Projects</b>	<b>71</b>
<b>4.1 Introduction</b>	<b>71</b>

## Interactive Tools for Supporting Taxonomists Working Practice

<b>4.2 System Concept</b>	<b>73</b>
4.2.1 Introducing the System	73
4.2.2 Description Data	75
4.2.3 Presenting the interfaces	76
<b>4.3 Proforma Building</b>	<b>77</b>
4.3.1 Process of usage	78
4.3.2 Refinements to the process of usage during storyboard development	81
<b>4.4 Entering Instance Description Data</b>	<b>82</b>
<b>4.5 Evaluation</b>	<b>83</b>
4.5.1 Presenting choices for scoring	83
4.5.2 Overview Visualisation	84
4.5.3 Presenting Definitions with Multimedia Aspects	87
4.5.4 Level of Guidance	87
4.5.5 Glossary Based Problems	88
<b>4.6 Conclusions</b>	<b>91</b>
<b><i>Chapter 5 Ontology-Based Generation of Data Entry Interfaces</i></b>	<b>92</b>
<b>5.1 Introduction</b>	<b>92</b>
<b>5.2 Approach and Tasks</b>	<b>94</b>
<b>5.3 Ontology Based Domain Model</b>	<b>96</b>
5.3.1 Abstract Domain Model	97
5.3.2 Domain Ontology.	98
5.3.3 Mapping to the Concrete Domain Model	100
5.3.4 Specialised Domain Model	103
<b>5.4 Presentation Models</b>	<b>107</b>
<b>5.5 Development Methodology</b>	<b>108</b>
5.5.1 Methodology Principles	108
5.5.2 Evaluation of Interactive Tools	108
5.5.3 Interactive tool development phases	110
<b>5.6 Conclusion</b>	<b>111</b>
<b><i>Chapter 6 Specialisation Process</i></b>	<b>112</b>
<b>6.1 Introduction</b>	<b>112</b>
<b>6.2 Domain Model</b>	<b>113</b>
6.2.1 Names	113
6.2.2 Fixed score	114
<b>6.3 Task Model</b>	<b>115</b>
6.3.1 Primary specialisation task	116
6.3.2 Other supporting tasks	116
6.3.3 Specialisation Task Restrictions	118
<b>6.4 Ontology presentation model</b>	<b>118</b>
6.4.1 Interface Designs	119
6.4.2 Presenting the specialised domain model	120
6.4.3 Mapping the domain and task model to the interface	121
6.4.4 Mapping tasks to the interface	121
6.4.5 Domain terms for domain model terms	139
<b>6.5 Specialisation Issues</b>	<b>139</b>
6.5.1 Results consistent with ontology	140
6.5.2 Users able to specify their concepts	143
6.5.3 Informed choices	144
6.5.4 Other Effectiveness and Usability issues	151
6.5.5 Time	158
<b>6.6 Conclusion</b>	<b>163</b>

<b>Chapter 7 Data Entry Process</b>	<b>166</b>
<b>7.1 Introduction</b>	<b>166</b>
<b>7.2 Domain Models</b>	<b>166</b>
7.2.1 Specimens	166
7.2.2 Multiple values	167
7.2.3 Concrete description objects	169
7.2.4 Modifiers	169
7.2.4 Not scored statement	170
7.2.5 Description object presence attribute	170
<b>7.3 Data Entry Task Model</b>	<b>171</b>
7.3.1 Task Order	171
7.3.2 Attribute instantiation	172
<b>7.4 Presentation Models</b>	<b>172</b>
7.4.1 Two presentation models	172
7.4.2 High-level concept	173
7.4.3 Description object hierarchy	174
7.4.4 Description object	175
7.4.5 Navigation task	176
7.4.6 Attribute	177
7.4.7 Supporting task: review instantiation	184
7.4.8 Specimen details	184
7.4.9 Changing the data entry presentation model	185
7.4.10 Domain terminology	185
<b>7.5 Evaluation</b>	<b>185</b>
7.5.1 Expressing data entry concepts	186
7.5.2 Informed Decisions	191
7.5.3 Efficient and effective usage	193
<b>7.6 Conclusion</b>	<b>200</b>
<b>Chapter 8 Application in other domains</b>	<b>203</b>
<b>8.1 Introduction</b>	<b>203</b>
<b>8.2 Importing domain ontologies</b>	<b>204</b>
<b>8.3 Other domain ontologies</b>	<b>205</b>
<b>8.4 TDWG Taxonomic Transfer Concept Schema</b>	<b>206</b>
8.4.1 Mapping the ontology	207
8.4.2 Effectiveness of presentation	216
<b>8.5 Conclusion</b>	<b>219</b>
<b>Chapter 9 Conclusion</b>	<b>220</b>
<b>9.1 Introduction</b>	<b>220</b>
<b>9.2 Discussion</b>	<b>221</b>
<b>9.3 Main Contributions</b>	<b>225</b>
<b>9.4 Future Work</b>	<b>227</b>
<b>References</b>	<b>230</b>
<b>Appendix A: Use Cases</b>	<b>244</b>
<b>Appendix B: Storyboard Examples</b>	<b>267</b>
<b>Appendix C: Prometheus 2 Data Model</b>	<b>271</b>
<b>Appendix D: Third Phase Development</b>	<b>277</b>
<b>Appendix E: Wide User Tests</b>	<b>288</b>

## Figures

<i>Figure 2.1: Simplified Taxonomic Hierarchy Example of Western Cutlery</i>	6
<i>Figure 2.2: The Taxonomic Process</i>	11
<i>Figure 2.3: Data Flow Diagram – Top Level</i>	12
<i>Figure 2.4: Data Flow Diagram – Detailed Sort</i>	15
<i>Figure 2.5: Identify New Character Process</i>	16
<i>Figure 2.6: Selection of computerised tools to illustrate support for taxonomists' working practice</i>	26
<i>Figure 2.7: Potential areas for Computerised Tools</i>	31
<i>Figure 3.1: Protégé Frames instances tab for knowledge acquisition</i>	50
<i>Figure 3.2: Protégé Frames forms tab for tailoring instance editing forms</i>	51
<i>Figure 3.3: OWLviz visualisation of an ontology, showing the subsumption class relationships</i>	52
<i>Figure 3.4: Gene Expression Information Resource Project - MOUSE ATLAS</i>	53
<i>Figure 3.5: Gene Expression Information Resource Project - MOUSE ATLAS (with overview navigation pane)</i>	54
<i>Figure 3.6: Treemap of stock holdings</i>	64
<i>Figure 3.7: Information Pyramid of file directory</i>	64
<i>Figure 3.8: Basic cone tree model</i>	66
<i>Figure 3.9: Hyperbolic Disc</i>	68
<i>Figure 4.1: System to capture specimen description data in taxonomy.</i>	74
<i>Figure 4.2. Example of Proforma Builder interface storyboard.</i>	76
<i>Figure 4.3 Example of data entry interface storyboard.</i>	77
<i>Figure 4.4: Creation of Electronic Proforma</i>	79
<i>Figure 4.5: Creating an Electronic Proforma Prototype. Example from initial storyboarding.</i>	80
<i>Figure 4.6: Specifying a character to add to a proforma. Example from storyboarding.</i>	82
<i>Figure 4.7: File tree overview. The Leaflet of the Leaf is focussed on in this example.</i>	86
<i>Figure 4.8: Storyboard proforma builder interface with pop-up definition box</i>	87
<i>Figure 4.9: Alternative orientation structure editor</i>	90
<i>Figure 5.1. Ontology Driven Automated Generation of Data Entry Interfaces utilising a Model-Based Approach.</i>	95
<i>Figure 5.2: Abstract Domain Model</i>	98
<i>Figure 5.3 Major terms and relationships represented in the angiosperm domain ontology conceptual model</i>	99
<i>Figure 5.4: Mapping from Angiosperm Ontology Conceptual Model to Abstract Domain Model</i>	101
<i>Figure 5.5: Mapping structure terms to description objects.</i>	102
<i>Figure 5.6 Cloning description objects.</i>	106
<i>Figure 6.1: Specialisation Interface Screenshot</i>	119
<i>Figure 6.2: Specialisation Interface: definitions Explorer Tab Screenshot:</i>	120
<i>Figure 6.3: Description Object Hierarchy Tree</i>	122
<i>Figure 6.4: Including a universally applicable description object in the specialised domain model</i>	126
<i>Figure 6.5: Result of including 'lobe'</i>	127
<i>Figure 6.6: Attribute and value object hierarchy</i>	128
<i>Figure 6.7: Attribute-value tree example for selected 'Lobe' description object</i>	129

## Interactive Tools for Supporting Taxonomists Working Practice

<i>Figure 6.8: Attribute Details Panel for leaf length:width ratio attribute</i>	131
<i>Figure 6.9. Sub-attribute nodes on attribute-value tree</i>	132
<i>Figure 6.10: Mouse-over definition example</i>	133
<i>Figure 6.11: Attribute Specialisation Pop-up Window</i>	137
<i>Figure 7.1: Data entry Interface example</i>	173
<i>Figure 7.2: Data Entry Interface components</i>	174
<i>Figure 7.3: Description object informative data display for a description object with clones</i>	176
<i>Figure 7.4: Attribute Instantiation Complex Interaction Object</i>	178
<i>Figure 7.5: Alternate instance-scores example</i>	179
<i>Figure 7.6: Attribute instantiation IO example for an abstract numerical attribute</i>	180
<i>Figure 7.7: Attribute instantiation IO example for a concrete numerical entry attribute</i>	180
<i>Figure 7.8: Attribute instantiation IO example with value objects represented by checkboxes</i>	181
<i>Figure 7.9: Attribute instantiation IO example with value objects represented by pictorial selection</i>	181
<i>Figure 7.10: AIO selection strategy</i>	183
<i>Figure 8.1: Transfer XML format (.xsd file). Top level elements</i>	204
<i>Figure 8.2: Transfer XML format (.xsd file). Terminology elements</i>	205
<i>Figure 8.3: Excerpt from TCS schema (version 0.88), showing top-level xml elements</i>	207
<i>Figure 8.4: TCS alternative types of names. (XML Spy view)</i>	209
<i>Figure 8.5: Excerpt from TCS xsd containing PublicationDetailed data</i>	211
<i>Figure 8.6: Description object 'PublicationDetailed'</i>	212
<i>Figure 8.7: Specialisation interface example for TCS ontology</i>	216
<i>Figure 8.8: Data entry interface example for TCS ontology</i>	217
<i>Figure 8.9: XML spy representation of TCS schema</i>	218

## Acknowledgements

I would like to thank Professor Jessie Kennedy, for her supervision, guidance and support throughout the project. Thanks also go to other colleagues at Napier University, particularly Trevor Paterson and David Benyon. I would also like to thank Mark Watson, Kate Armstrong, Sarah MacDonald, Martin Pullan and the other taxonomists at RBGE who gave so generously of their time to test the approach and answer my questions. Finally I would like to thank my family, especially my wife, Nicky and daughter Amelia for their patience, love and support throughout.



## Contributing Papers

Cannon, A., Kennedy, J., Paterson, T., Watson, M. (2004) Ontology-Driven Automated Generation of Data Entry Interfaces. In Williams, H., Mackinnon, L. (Eds.), Key Technologies for Data Management (Proc. BNCOD21): Lecture Notes in Computer Science 3112 pp.150-164. : Springer Verlag.

Paterson, T., Kennedy, J., Pullan, M. R., Cannon A. J., Armstrong, K., Watson M. F., Raguenaud C., McDonald S. M., Russell G. (2004) A Universal Character Model and Ontology of Defined Terms for Taxonomic Description, DILS 2004, LNBI 2994, pp. 63-78

Pullan, M., Armstrong, K., Paterson, T., Cannon, A., Kennedy, J. (2005). The Prometheus Description Model: An examination of the taxonomic description building process and its representation. *Taxon*, 543, pp. 751-765. ISSN 0040-0262

# Chapter 1

## Introduction

### 1.1 Taxonomic working practice

Taxonomic description lies at the heart of the classification of organisms and communication of ideas of biodiversity. Taxonomists are under pressure to complete work to accurately classify organisms before they become extinct. They are however proceeding with their work with only limited support from the computing advances of the last decades. That support is generally limited to areas of major research such as genome sequencing and to general desktop utilities such as word processing and spreadsheets. Whilst bioinformatics research has also focussed on the sharing of data repositories between experts in related disciplines in the biological sciences, comparatively little research has been done in supporting the capturing of the complex descriptive data that lies at the heart of biological classification.

There are a number of perceived problems with taxonomic description data where a lack of standards and the complex nature of the data give rise to issues of clarity, re-use and comparability. With communication of important concepts thus compromised, there have been proposals to improve the situation by using a structured data model, based on the use of defined terms, to capture descriptive data in a rigorous database [Diedrich 1997, Prometheus 2001]. There is however no support for taxonomists to use such a model to consistently capture their descriptive data.

Traditional methods of generating data entry interfaces to databases tend to involve simplistic forms-based interfaces generated by DBMSs to conform to the table structure (or views thereof). These however fail to capture domain semantics, constraining entry to basic data types at best. Other general automatic user interface generation tools have so far failed to gain widespread acceptance or address the needs of high-quality complex data entry. Higher quality user interfaces are made possible by the intervention of expert developers who develop interfaces for specific domain needs. In taxonomy however the domain of descriptive terminology is very large and a data entry interface based on all descriptive possibilities would be impractical. In addition each taxonomic project needs to collect data on a consistent set of features for all specimens described in

## **Chapter 1 - Introduction**

it; this set however varies depending on the project. It is equally infeasible to involve an IT expert for each individual project.

The necessity for scientists and others to use consistent terminology has recently been regarded as fundamental to advancing scientific research, particularly where data from disparate sources must be shared, compared or integrated. As a result, ontologies are increasingly used to describe and define complex domain data. Using an ontology to control the data entry for a database has the potential to ensure that better quality data is captured and that data from differing data providers will be compatible. It may also allow a data entry interface to be created that allows domain users to enter data using terms with which they are familiar but which are clearly defined semantically. Existing ontology based approaches for data entry are, however, still limited to using automatically generated forms-based data entry interfaces, unless manual editing is used. These systems are designed to populate a knowledge base describing relationships between described instance items of interest, rather than regulating the capture of the description of a complex concept.

### **1.2 Aim of Research**

The primary aim of this research is to support taxonomists in capturing descriptive data capture which is unambiguous, in particular by supporting the use of a structured data model that captures the semantics of the domain data in a clear and unambiguous manner. This involves supporting their working practice of collecting specimen description data using a proforma template with a theoretically consistent basis. This proforma and the descriptive data concepts that are of interest, change for different taxonomic projects.

This research tackles this problem through the use of domain ontologies to generate and control data entry interfaces that support high quality data capture. Investigating the use of ontologies in this context became a secondary aim of this research.

The approach developed to meet the primary aim adopts that of model-based user interface development environments. We hypothesise that using the ontology as the basis for a domain model will improve the user interfaces that can be generated for the task of data entry. By editing an ontology that captures the range of descriptive

## Chapter 1 - Introduction

possibilities for a domain (taxonomic description), taxonomists can generate data entry interfaces specific to the needs of their projects. Use of such an automatically generated project specific interface will support high-quality data entry.

### 1.3 Organisation of Thesis

The main contribution of this thesis is in the use of domain ontologies to support the capture of high-quality descriptive data in the instance field of taxonomy. This involved the ontology-based automatic generation of suitable interfaces, the effective presentation of ontologies to domain users for constrained editing and the use of ontologies to support high-quality data entry. The generalised application of the approach is also discussed. The thesis is organised as follows.

Chapter 2 examines the field of taxonomy; investigating and modelling the working practice. Qualitative research was conducted at the Royal Botanical Garden, Edinburgh (RBGE) in order to gain insight into the taxonomic process. Together with existing literature, the tasks and issues concerned with the taxonomic working process were analysed in order to identify areas where computerised support could be of value; in particular the issues of taxonomic descriptive data collection are noted.

Chapter 3 examines the literature in information visualisation, data entry interface generation and ontology fields relevant to addressing the identified descriptive data collection problems identified in chapter 2. Potentially useful techniques are identified.

Chapter 4 investigates an initial solution to the difficulties of capturing high quality descriptive data for taxonomy. Storyboard and use case walkthroughs are evaluated to identify the parameters of a system that can address the problem.

Chapter 5 describes a model-based user interface development environment (MB-UIDE) approach to providing data entry interfaces to databases for taxonomists. The tool is based on the storyboards and evaluation reported in chapter 4 and uses a domain ontology to provide domain knowledge to the system. The main models that effect the approach are introduced, including the system models and the conceptual model of the utilised ontology. The development and evaluation methodology for a designed interactive software tool is described.

## **Chapter 1 - Introduction**

Chapter 6 describes the specialisation process by which domain experts can specify their descriptive data concepts on the basis of the domain ontology. The system models and developed specialisation interface are described. The evaluation of the process, models and interface based on user testing and expert evaluation is discussed. Changes in the system resulting from the evaluation are highlighted.

Chapter 7 covers the other main user process, the data entry process itself. The models and automatically generated interface are described. Again the evaluation and resulting changes are discussed in a similar manner to chapter 6.

Although the primary focus for user tests and development is taxonomic description, the needs of a generalised approach that can use different domain ontologies were considered. Accordingly chapter 8 discusses the application of the approach to another instance domain, that of TCS, in full.

Finally chapter 9 discusses where this work fits into related research, what the main perceived contributions to research are and what future work might be done to take the approach further.

## Chapter 2

# Taxonomy

### 2.1 Introduction

This chapter discusses the problem area within taxonomic working practice addressed in this research.

As part of this research, qualitative research was conducted at the Royal Botanical Garden, Edinburgh (RBGE) in order to gain insight into the taxonomic process. Together with existing literature, the tasks and issues concerned with the taxonomic working process were analysed in order to identify areas where computerised support could be of value. A model of taxonomic working practice is developed from the undertaken qualitative research and within that framework those problems of taxonomic description that this research addresses are identified.

The rest of this chapter is organised as follows. The general subject of taxonomy and relevant concepts important to taxonomic description are first briefly introduced to give appropriate context to the following work. Next, undertaken qualitative research into taxonomic working practice is described and the working practice is modelled. Problems with the central practice of taxonomic description within this working practice are identified and discussed in greater depth. Other computerised tools are examined to see where aid and support is already available to taxonomists in their working practice. Areas where extra computerised support to taxonomists could be provided are identified and the specific area where this research will focus is determined.

### 2.2 Introduction to Taxonomy

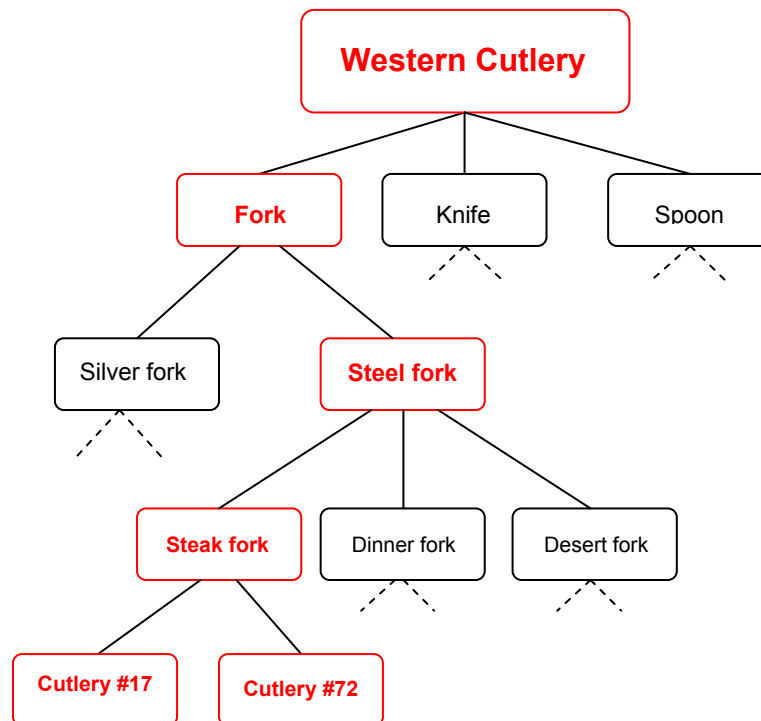
Biological taxonomy is the branch of biology concerned with the classification of organisms into an ordered hierarchical system of groups reflecting their natural relationships and similarities. In botanical science, this involves naming, describing, identifying and classifying groups of plants. The purpose of taxonomy is to

## Chapter 2: Taxonomy

communicate classification concepts, providing biologists and others the ability to identify, categorise and refer to organisms in a meaningful way. Taxonomic output is fundamental to all fields of biology that refer to organisms, and taxonomists in turn use the data derived from these sources when refining past classifications [Jacobs 1969].

### 2.2.1 Classification

In general terms, taxonomy involves the process of classifying objects based on a series of descriptive concepts they have in common. In botanical science, classification of plants is determined by concepts such as morphological characteristics (e.g. petiole length, number of petals), genetic make-up, and ecological characteristics. The classification process produces a hierarchical data set, in which each grouping is called a taxon (pl. taxa). In classical Linnaean plant taxonomy the lowest possible level of the data set is the plant specimen. Taxonomic classifications (taxonomies) do not all represent the same number of hierarchy levels, and some do not go down to the specimen level. The classification data set purely represents the taxonomic groupings, and not the concepts used to make classification decisions.



**Figure 2.1: Simplified Taxonomic Hierarchy Example of Western Cutlery**

Figure 2.1 gives a simplistic example of a taxonomic classification of cutlery. Western cutlery is classified into basic shapes {knife, fork, spoon}, forks are then classified by

## Chapter 2: Taxonomy

material {silver and steel forks}, steel forks are then classified again {steak fork, dinner fork, desert fork}, the steak fork grouping can be seen to include specific cutlery specimens {#17, #72}.

The same organism may be classified according to different taxonomic opinions, resulting in different classifications with different names, groups and underlying concepts. Using the above example, cutlery could alternately be split by materials first and then by design style and price. This would produce a different classification using the same pieces of cutlery. In taxonomy all previous published classifications are considered valid, leading to the problem of using conflicting hierarchies, where the same basic building blocks of the classification are grouped in different ways.

### 2.2.2 Character Concepts

The cutlery example above shows the results of the classification, but, as can be seen, the concepts upon which the classification is made can only be guessed at from this data. Concepts (such as basic function, material, number of prongs) are being used to delimit the groups in the classification process. Objects are grouped with other objects that share similar concepts (for example all the 2-pronged, steel forks could be grouped by similarity into the steak fork group).

Descriptions of taxa (and possibly specimens) found in taxonomic publications contain these concepts, although they do not necessarily explicitly declare which concepts have been used to delimit the taxonomic groups (taxa) at any particular level. A taxonomic description is a record of what an organism (an actual specimen or a taxon of any rank) looks like, consisting of statements on features of the organism. These statements are usually referred to as characters. It is these characters that form the essential concepts underlying taxonomic classifications, describing and differentiating taxa.

The following is an example of a typical plant taxon description:

***Torilis* genus description** (Taken from the Chinese Umbels descriptions) [Sheh 2004]



## Chapter 2: Taxonomy

*Annual or perennial, herbaceous, bristly, hispid or pubescent throughout. Stem erect and branching. Leaf blade 1-2-pinnate or pinnately decomposed. Loose compound or capitate umbels, lateral or terminal and lateral; involucre bracts few or wanting; rays 2-12, spreading-ascending, or obsolete; bractlets 2-8, linear or subulate. Flowers white or purplish-red. Calyx-teeth deltoid, acute. Petals obovate, with a narrower inflexed apex, abaxially appressed-strigose. Stylopodium thick, conic; styles short. Fruit round-ovoid or oblong, flattened laterally, tuberculate or prickly; primary ribs filiform, setulose, the lateral ribs displaced onto the commissural surface, the secondary hidden by the numerous glochidiate prickles or tubercles which occupy the entire interval; vittae 1 under the secondary ribs, 2 on commissure. Carpophore bifid at the apex or cleft one-third or one-half of its length.*

The characters used in a taxon description can be any of a number of different types. In plant taxonomy, organisms are grouped by similarity (phenetic relationships) of their chemical, morphological, anatomical, physiological, and ecological characters (particularly the anatomical-morphological characters). Evolutionary characters are used in the related field of systematics (a term often loosely used as synonymous with taxonomy). Whilst encompassing the areas covered by taxonomy, systematics, also covers evolutionary studies (the processes of evolution and evolutionary relationships between groups) to attempt by classification to convey the evolutionary history of plant groups (phylogenetic relationships).

### 2.2.3 Linnaean taxonomy

This research refers primarily to classical Linnaean taxonomy using phenetic relationships of anatomical-morphological characters to create a specimen-based taxonomic revision or new classification, although it is generally applicable to other types of taxonomy.

## 2.3 Analysis Of The Taxonomic Process

### 2.3.1 Introduction

Taxonomists work on projects concerning a group of plants based upon a geographical area (floristic works called floras) or subject group (either revisions based upon previous taxonomists' taxonomic classification concepts or monographs representing a new subject group). These projects aim to produce a publication which contains the results of their work, including relevant classifications and supporting descriptions.

There is no agreed consensus upon a formalised methodology for creating the various published works that are the end-product of taxonomic projects. The taxonomic process generally involves sorting specimens into groups that reflect their natural variation to create a hierarchical classification. During this process, the character concepts that underlie the classification are developed. The results of the classification process are published to communicate the taxon concepts to interested parties.

As a first step in supporting taxonomists' working practice, it is necessary to understand their working process in order to identify where and how computerised tools can be of help. Accordingly, as there is no agreed methodology, the taxonomic working process was identified and modelled based on the results of qualitative research undertaken at the RBGE.

#### 2.3.1.1 Qualitative Research Study Methodology

The qualitative research took the form of an ethnographic study using three main methods: interviews of taxonomists; collected documentation; and observations of taxonomists during their work. The stated aim of the study was to investigate the current taxonomic process within RBGE, to understand taxonomists' views on the term 'character' and explore user opinions on possible future roles of information technology to aid the taxonomic process.

#### Interviews

Semi-structured interviews were conducted with eight taxonomists from the RBGE in June-July 2001. Interviews were taped, and later transcribed. Complementary notes were taken during and immediately following interviews. All interviewees were asked to

## **Chapter 2: Taxonomy**

outline what they do and how. They were all asked what their understanding of character was and how they chose the characters they described. They were also asked to outline any difficulties they had with reading past descriptions and with their work in general.

Further preliminary unstructured and follow-up semi-structured interviews with three RBGE taxonomists were also conducted to clarify various related issues.

### Documentation

Limited documentation was collected from interview participants of documents relating to their work. This was limited due to the nature of the taxonomic process, particularly the lack of collaborative working during the actual process, which consequently meant there were no inter-personal communication records as well as poor filing and destruction of paper working notes. Finished published documents (monographs, floras, etc) were available and consulted.

### Observation

Direct observation of an initial sort of a group of specimens, using dried pressed material, into approximate taxa by taxonomists at RBGE was conducted. Observation of a detailed sorting was also conducted, in which the repeated task elements were observed. Other informal demonstrations of parts of working practice by RBGE taxonomists were requested and observed.

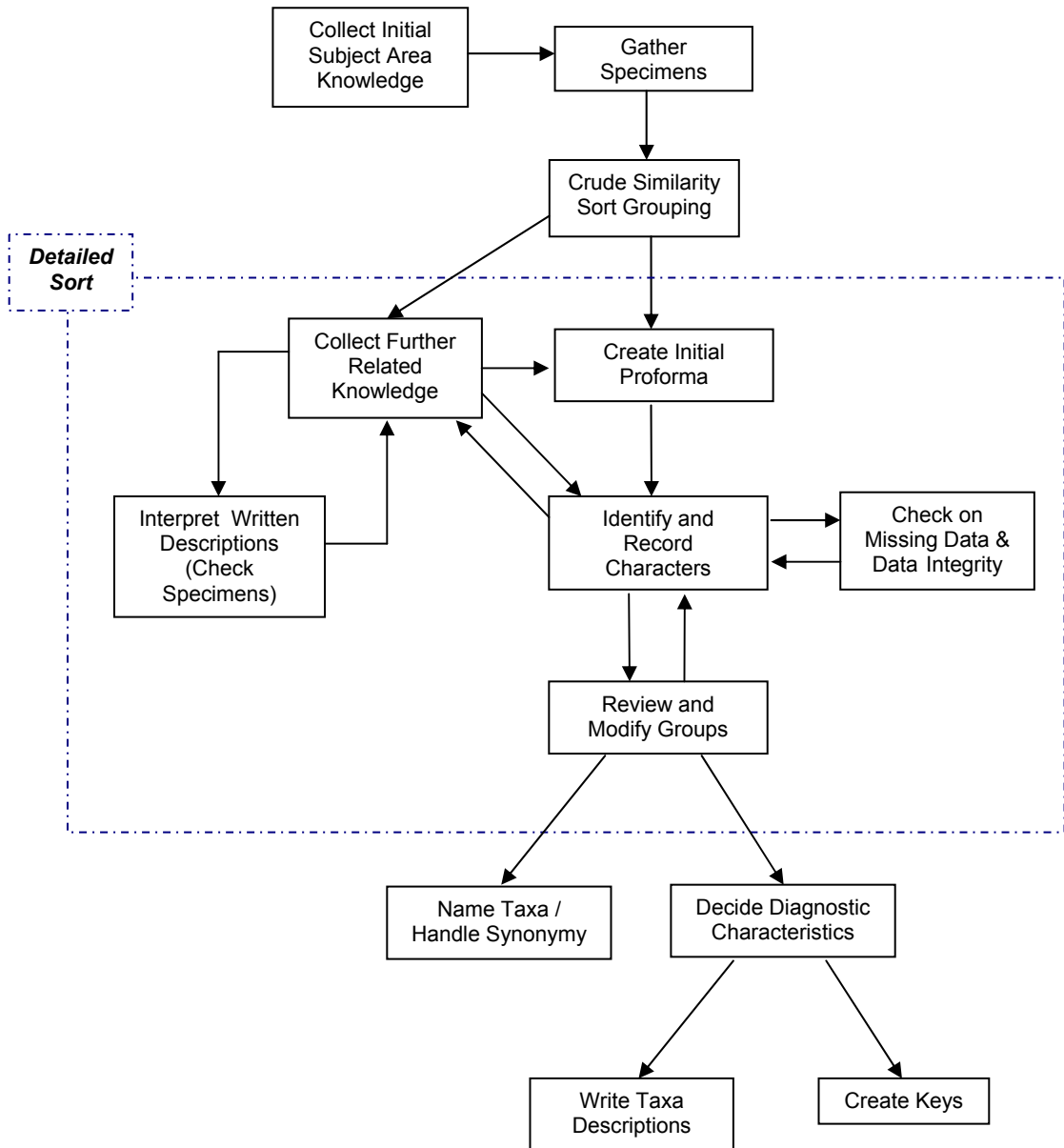
Whilst a full ethnographic study was not conducted as it was felt the results would not justify the time required, some cultural impressions were drawn and recorded based upon informal contact with RBGE taxonomists.

#### 2.3.1.2 Identified Process

The qualitative research project at the RBGE found that there was a general taxonomic process, which could be identified, even if it was not formalised. This process holds true in general terms for the various types of publication. The emphasis and level of detail however, depend upon the type of publication and its intended audience. The process identifies the taxonomic groups and the observed plant characteristics that make up a description. Figure 2.2 models this process. The constituents of this process are detailed

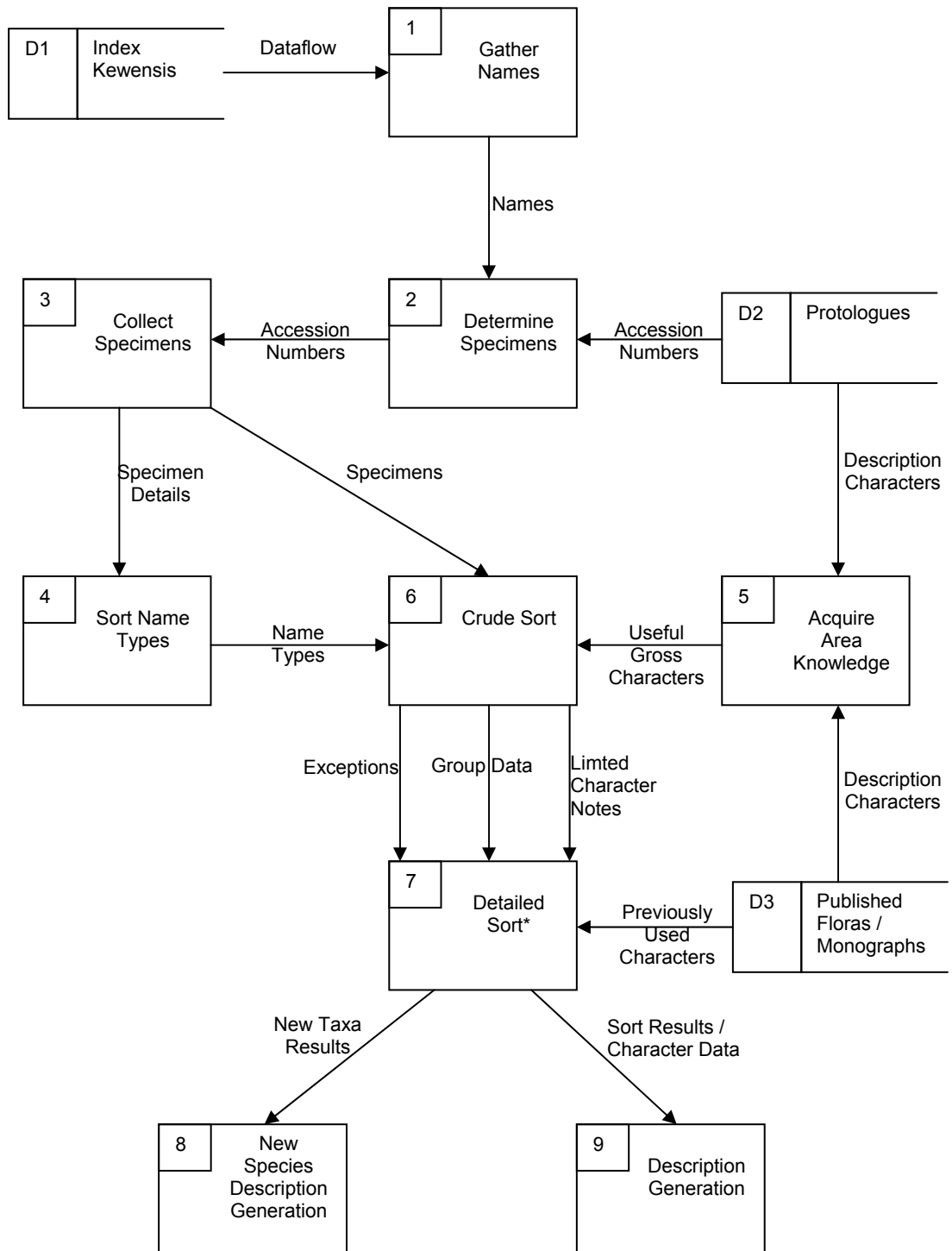
## Chapter 2: Taxonomy

below. Data flows within the system were problematic to capture, as due to the generally single person nature of the taxonomic process, many of the data flows passed between the same individual in an informal manner. Identified data flows are detailed in figures 2.3 and 2.4.



**Figure 2.2: The Taxonomic Process**

## Chapter 2: Taxonomy



**Figure 2.3: Data Flow Diagram – Top Level (\*The Detailed Sort process is expanded in figure 2.4)**

## **Chapter 2: Taxonomy**

### **2.3.2 Initial Preparations for a Taxonomic Project**

The first task in the taxonomic process is to establish a list of names used within the study group or geographical area, usually using a taxonomic index such as Index Kewensis [2004] (an on-line database of seed plants) or Index Fillicum [2004] (a database of ferns). Using this as a base, the protologues (the first publication of each name) are gathered in order to verify the validity of the name, check priority, and possibly to read the first description. If the work being undertaken is a flora, floras from the surrounding area may be consulted. Based on this data, sufficient physical specimens from the subject group or geographical area are gathered to support the study.

### **2.3.3 The Crude Similarity Sort**

Using the collected physical specimens, an initial crude similarity sort into groups is done. This process tends to be done physically, by making piles of the physical specimens. Often multiple taxonomists may work together at this stage, with colleagues helping the primary investigator. This process can be quite quick, especially for revision work. For floristic works, this stage tends to be more important. Often the work is done in the collecting room where access to computers is more limited.

Firstly in the crude sort, the specimens which are ‘types’ of previously used taxa names (each taxa in a classification has a type specimen from which its name is partly derived) are sorted out, as the basis for the sort. The remaining specimens are then rapidly divided up into crude taxa, without looking at past classifications, using gross macroscopic characters (such as leaf shape) and general overall appearance. Some taxonomists may use past descriptions to give an idea of what characters to look for. Few actual notes are usually made at this stage, other than the specimen composition of the crude taxa themselves.

### **2.3.4 The Detailed Sort**

The detailed sort is an iterative process utilising detailed characters to refine the crude taxa groups. The specimens are examined in more detail to determine how much variation between specimens can be considered simply within-taxon variation and how much is sufficient to warrant separating the specimens into different groups. It is during

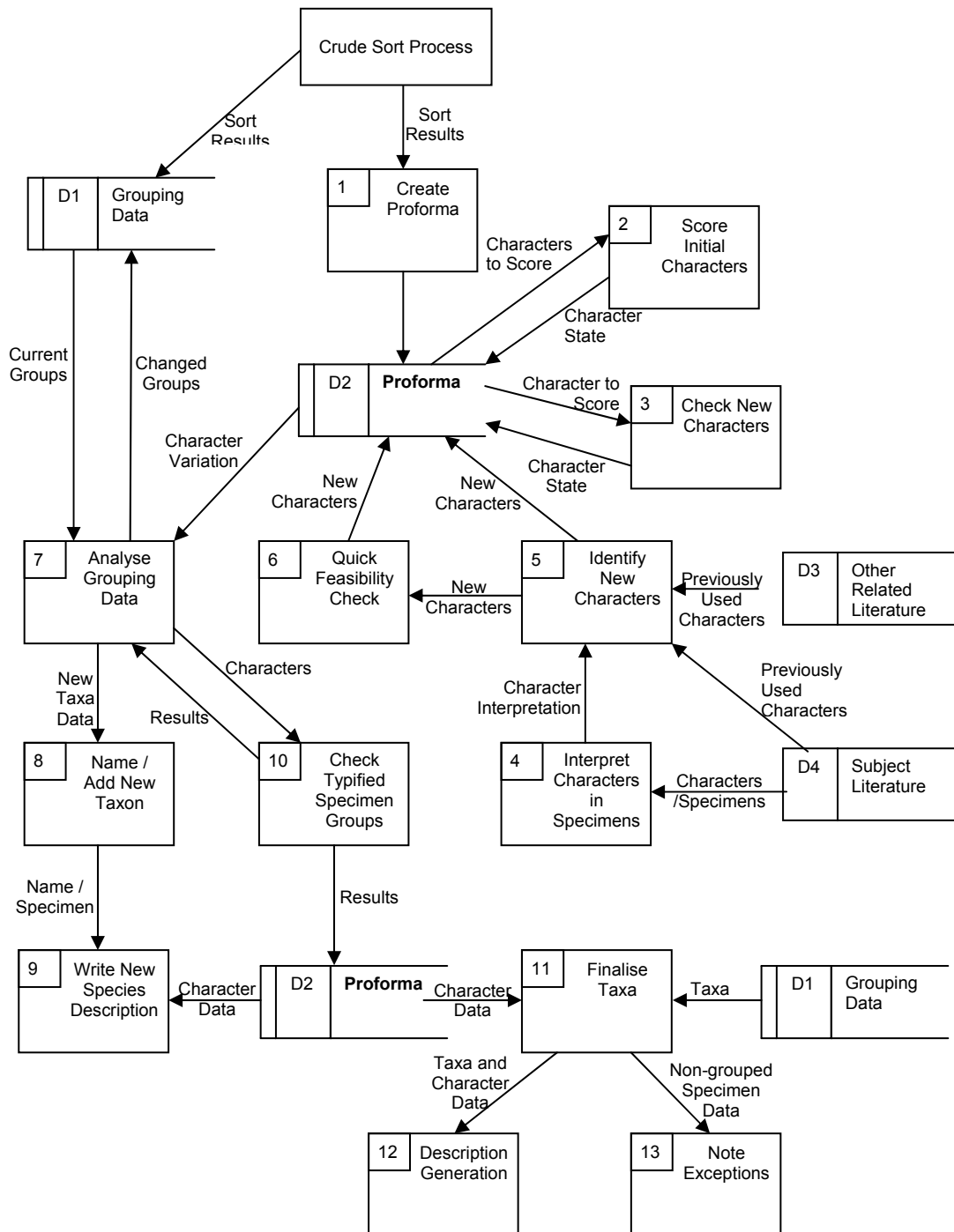
## Chapter 2: Taxonomy

this stage that recognisable character concepts are developed and articulated. Taxonomists look for useful comparable characters and the variation between those characters to develop and support their classification concept. The end result of this detailed sort is a set of taxa and a record of the characteristics of the taxa or the specimens that make up the taxa. Some of the specimen material may be too poor to assign it to any taxa, in which case it remains a taxonomic problem.

Notes are more commonly kept during the detailed sort than in earlier stages. Usually records are kept in the form of a proforma, which records the characters for each specimen. Initially the proforma is empty, the taxonomist populates it with characters and then records the specific scores for individual specimens. The same proforma is generally used for all the specimens in a particular study. The central importance of the role of the proforma can be seen in the data flow diagram (Figure 2.4). Generally, at present, the proforma tends to be in the form of a basic spreadsheet or a paper table. Additional unformatted notes are also made concerning ideas, concept drawings, etc. Occasionally some taxonomists will record the taxa and their descriptions directly without a proforma during quick floristic work.

The detailed sort can be further split up into sub-processes to help analyse the process.

## Chapter 2: Taxonomy



**Figure 2.4: Data Flow Diagram – Detailed Sort**

### 2.3.4.1 Proforma Creation

The character proforma is first created with characters based upon the taxonomist's knowledge gained from their general work, from ideas developed in the crude sort and from reading other related descriptions after the crude sort is completed. In addition, by

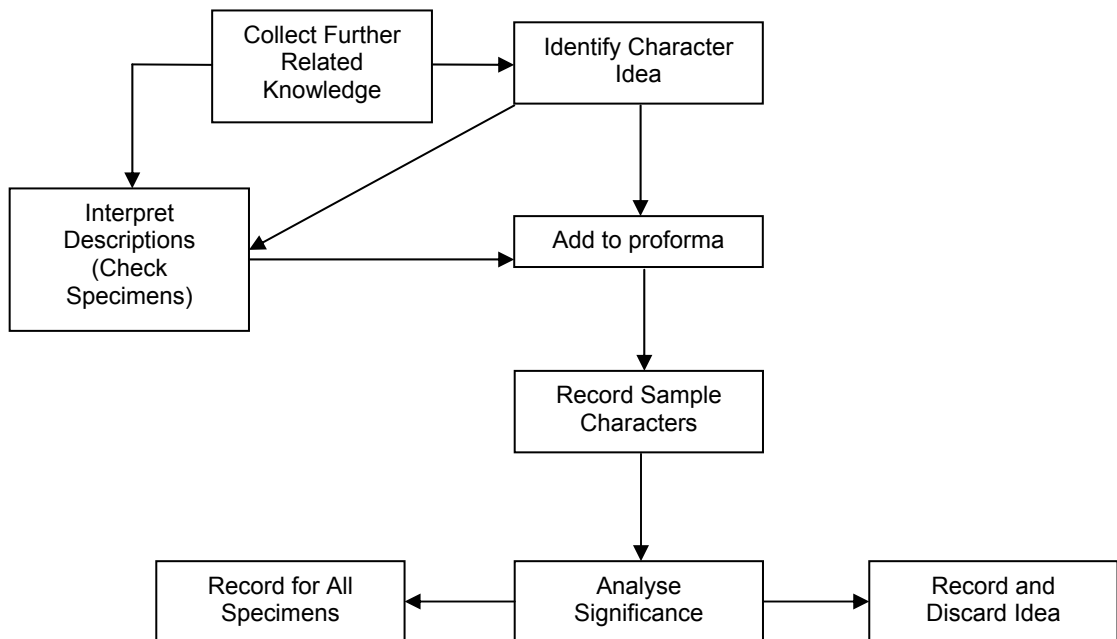


## Chapter 2: Taxonomy

looking at the specimens the taxonomist identifies the various major structures and features of the plant that might be used as characters. Some taxonomists determine the domain of possible values (states) for each character at this point (taxonomists creating data to be compatible with electronic description formats (see 2.5) definitely do this). No definite overt decision about definitions of terms is generally determined at this stage, although the taxonomist may have implicit assumptions regarding the definitions of the terms used.

### 2.3.4.2 Identify New Characters

Observable characters are also added to the proforma after it has been initially created and specimens are recorded. Identifying new characters (see Figure 2.5) is based on research and ideas found during the detailed sort process after the initial proforma has been set-up.



**Figure 2.5: Identify New Character Process**

A taxonomist gains ideas about possible characters of interest during the process as they study the specimens in question and become more familiar with the subject area. By having an overall understanding of the subject area and the characters already recorded, the taxonomist may be able to identify areas where few characters are currently

## **Chapter 2: Taxonomy**

recorded. Taxonomists also read around the subject area, and may gain ideas of characters from this related literature. Often due to a lack of definition for terms, taxonomists referencing literature must return to referenced samples to check the exact meaning of terms used to describe characters. The taxonomist's personal working experience and wider biological knowledge also suggests new characters, which may be utilised. In general, the process of developing character concepts is imprecise and relies heavily on intuition based upon the taxonomists' professional expertise.

Once a new character is identified, its significance is reviewed, to determine if it is a useful character for describing and splitting taxonomic groups. To do so, the taxonomist may record a representative sample from the existing taxonomic groupings. If these representative samples show the character is likely to be useful (in terms of ability to be effectively recorded and to potentially differentiate groups), the taxonomist will then record all the specimens using the character, if not, then they will record what has been done and discard the idea.

### 2.3.4.3 Review Groups

A repeated task during the detailed sort, is the constant updating both mentally and on paper, of the grouping data, based on the recorded characters. Initially the grouping information is held in physical piles of specimens. This grouping data is then later recorded. With small data sets (both of specimens and number of recorded characters), the taxonomist is able to do much of this reviewing of groups mentally. With larger data sets, the amount of data and number of variables becomes too great for the taxonomists to maintain a mental picture of their entire data set, forcing them to rely on ad-hoc notes and manipulated spreadsheets.

### **2.3.5 Completing the Project for Publication**

After the detailed sort is completed, the work is finalised for publication. Taxa are named according to the normal taxonomic rules and any naming issues (synonymy) are resolved. Reviewing individual recorded proformas, final groupings and the spread of characters makes the final determination of which characters are included in the published descriptions to identify and delimit taxonomic groups. The creation of keys and actual writing up of the descriptions then occurs as required before the work is

## Chapter 2: Taxonomy

published. Writing descriptions is generally completed at the end of the process by compiling notes. The proforma, other records and notes made during the sort process are not usually published and are often discarded.

### 2.3.6 Differentiation in the process due to the nature of the intended publication

There is some difference in the emphasis and level of character detail between the taxonomic process for creating monographic works (monographs, revisions, etc.) and floristic works (floras, field guides, etc.) due to the nature of the publication and intended audience.

In the case of more detailed monographic works in which more time is likely to be available, the detailed sort can be a long process, with more detailed records, particularly of specimens, being made. Monographic work is more likely to look for new characters to base classification decisions upon and is more rigorous in determining what constitutes a recorded character. New classifications are generally put forward in monographs and revisions. Consequently the final published descriptions tend to be more comprehensive, and they often possess introductory chapters which justify and explain the character concepts, sometimes relating these to others in the literature.

In the case of floristic works, the detailed sort tends to be shorter, more based on accepting previously published taxa and descriptions, where available. Taxonomists working on floristic projects are less likely to record detailed characters of each specimen, often preferring to construct the characteristics of the taxa as they go along rather than recording every specimen individually. In contrast with monographic works, floristic works tend to be more concise. They may use and comment on existing taxonomic publications, but generally do not develop new classifications (although they may aggregate and merge pre-existing works to form a single integrated taxonomy for plants of a region). Descriptions in floristic works may still convey taxon concepts, however this role is secondary to the primary purpose of differentiating between taxa found in a geographical area. Thus the descriptions are often short and only give the diagnostic characters essential for identification. For example, the description of the *Sanicula* genus in the Umbelliferae of India [Mukherjee 1993] flora is less than half the length of the description from Shan & Constance's monograph [Shan 1951]. Completed floristic works do not usually have detailed information on the characters used. Some

## **Chapter 2: Taxonomy**

works may have glossaries explaining the meaning of terms, however these are essentially general guides, and may not accurately convey the character concepts used in a particular taxa.

### **2.3.7 Collaborative Working**

Collaborative working practice in the taxonomic process is very limited. One taxonomist usually undertakes the identified process (figure 2.2), alone. On occasion there are collaborative projects, for example with a large flora, however in these, the individual taxonomist is assigned a sub-set of the subject area on which they will work, producing descriptions in an agreed format, but not actually collaborating on the details of the taxonomy. This solo working method combined with the lack of accepted formalised procedures, gives rise to individualistic working practices.

## **2.4 Taxonomic Description Issues**

A number of issues with working practice in the taxonomic description process were identified in the course of the qualitative research undertaken and related literature search.

A taxonomic description is the main way in which a taxon concept is communicated. Character data are used within a description to delimit taxon concepts, however, there are problems with the way in which these character data are handled and used for communication. There are many instances where this communication is ineffective. For example it may not be possible to see from published descriptions whether two species are closely related because the same criteria are not used in the construction of the different descriptions [Sivarajan 1991]. In another example, taxonomists in a study were asked to identify images of diatoms. The ability to correctly identify each specimen ranged from 33.8% to 86.5% with a mean of 63.3%. One cause of such low accuracy could be deficiencies in the descriptions of species in floras [Kelly 2002]. As the purpose of taxonomy is to communicate taxon concepts in order to provide frameworks within which other disciplines can work, it is important to improve the communication of these concepts. Doing so involves improving the communication of the underlying character data.

### 2.4.1. Definition of Character

A principal difficulty with the use of characters as the focus for the communication of taxon concepts is that the term ‘character’ itself can be misleading as there is little consensus amongst taxonomists on what exactly constitutes a character. The term is used in several different ways and has several definitions. One study [Colless 1985] for example found 19 different explicitly stated, or clearly implied, definitions of ‘character’ from a survey of 50 publications. Most commonly a character is seen either as some kind of feature, aspect or attribute of a plant or as a statement on a feature of the organism [e.g. Blackwelder 1967, Wiley 1981, Stuessy 1990, Frisrup 1992, Bailey 1999].

Most of the ways in which ‘character’ is utilised in descriptions can actually be identified as really belonging to one of two main conceptual entities [Diederich 1997]. One is where the character is a general concept (e.g. *leaf shape*) which is separated from the value (e.g. *ovate*). In this situation the character is a combination of a structure (e.g. *leaf*) and an abstract concept or property describing that structure (e.g. *shape*). The score is often referred to as the character state. The other use is where the structure and the score are combined (e.g. *leaves ovate*) in which case the property (e.g. *shape*) is implicit. In the qualitative research study, taxonomists generally supported one of these two general approaches to character in theory, although some respondents believed that additionally a character concept only truly existed where it also existed to support variation between taxa. A small minority of respondents preferred character concepts to refer to a more complex entity, comprising multiple individual concepts. Further investigation also showed that approximately 75% of respondents experienced at least some degree of difficulty in manually breaking down complex descriptive statements from existing natural language taxonomic descriptions (e.g. ‘*mericarps with slender, almost glabrous ridges*’) into atomic characters of the structure, property, state format.

Due to the loose use of character, the amount of data placed in one character by some taxonomists is quite variable, and may include more than one feature, depending on how important the feature is in delimiting taxa. For example, the ‘*mericarps with slender, almost glabrous ridges*’ example from above could be seen as one character or as multiple characters such as: ‘*mericap ridges, present*’; ‘*mericap ridges, absent*’;

## Chapter 2: Taxonomy

'*mericarp ridges, slender*'; and '*mericarp ridges, almost glabrous*', depending on whether there were other taxa with thick, almost glabrous ridges or slender, hairy ridges.

In order to promote clarity and comparability of character data, a common approach to the conceptualisation of character data would need to be developed and adapted.

### 2.4.2 Character selection and definition

A more significant problem with the use of character data in descriptions is that character selection and definition is inconsistently applied.

#### 2.4.2.1 Character Selection

Character selection has been recognised as a "*weak link*" [Davis 1963] for some time. One problem is that taxonomists use a mental picture of the similarities of their specimen groups and often do not consciously use defined characters to delimit those groups. They only break down their group descriptive concept into its component parts to communicate their ideas. Even when characters are selected at early stages, the selection is done on an ad hoc basis, based largely on experience and favoured publications. Poorly selected characters can mean that taxonomists' cannot effectively communicate their taxon concepts. For example, a monograph on the *Biscutella* genus [Machatschke-Laurich 1926], provides descriptions and key characters which are not sufficiently different for related taxa to be distinguished. An examination of the cited specimen material however, showed that the taxa are recognisable on the basis of characters not mentioned by the author [Davis 1963].

Another selection problem arises with a lack of consistency in the characters selected for inclusion in published works. Within one publication, one taxon may be described using characters that are not recorded for the next. Additionally the order characters are placed in a description may vary. Usually, descriptions are generally ordered acropetally (from the bottom of the plant up and from the outside in), but there is no formalised order. The lack of consistency in format between descriptions, particularly the failure to use the same characters in the same order for multiple taxa in the same publication, may cause misinterpretation and general difficulty in parsing descriptions, as well as making effective comparisons impossible.

## Chapter 2: Taxonomy

### 2.4.2.2 Character Definition

The other major problem with character data in descriptions is that there is no accepted universal defined terminology and glossaries of terms used are not often provided, giving rise to uncertainty, as well as rendering meaningful comparisons impossible. Taxonomic description terms can have a number of definitions. For example, Lawrence [Lawrence 1951] defines the term ‘tomentose’ as ‘*densely woolly or pubescent; with matted soft wool-like hairiness*’ whereas Stearne’s definition is ‘*thickly and evenly covered with short more or less appressed curled or curved matted hairs*’ [Stearne 1983]. A term used without a definition will mean that a reader’s interpretation of the term will depend on which definitions they have previously encountered and if they have their own personal preferences for the term. Individual taxonomists tend to have a set of terms that they always use and have fixed ideas about their meaning, which may be different from other taxonomists’ understanding. One taxonomist’s understanding of a term may consequently be significantly different from another’s understanding of that same term and this difference in opinion may not be apparent.

The use of simple ambiguous natural language terms as part of a description also make interpretation difficult. For example, a key to the species of *Sanicula* differentiates *S. lamelligera* from *S. petagnioides* on the basis of the number of spines found on the fruit. The states for this character are ‘densely spined’ and ‘sparsely spined’ [Huang 1993]. The author had a sound taxon concept for each species but any user of this key may find it difficult to distinguish between ‘densely’ and ‘sparsely’ as it is undefined what constitutes ‘densely’ as opposed to ‘sparsely’ in this instance [McDonald 2002].

Due to the age of some descriptions there can also be difficulties with language as the meanings of terms over time changes. Taxonomy is a discipline with a long history and many older works are still heavily utilised. This is particularly a problem in reference to descriptions written in Old French or German where the meaning of words has changed over time (Botanical Latin by contrast is more stable). In addition old descriptions are often incomplete, leaving obvious difficulties in interpretation.

## **Chapter 2: Taxonomy**

In these all these situations where definitions are lacking, taxonomists can only resolve exactly what is meant by a term, by the time consuming method of returning to the original specimens upon which the author based their description.

The lack of explicit definitions of terms can also have an effect within a project, due to definition creep. In this situation, a taxonomist's internal conceptualisation of a character or term, may gradually change during the life of a project without the change being apparent to the taxonomist or later interpreters. This situation most commonly arises in the case of terms to sub-divide a range of concepts into a set of atomic alternatives. (e.g. a particular leaf shape term may be used for an increasingly wider range of actual leaves during the life of a project.)

### 2.4.2.3 Inconsistent similar terminology

Related to the problem of different possible definitions of the same term, there is a general lack of consistency in terminology used. Different people do not describe things in the same way and it is often not clear whether they mean the same thing as no definitions are available. (e.g. one author might describe inflorescences as diachasial cymes and another simply as cymes. They may actually mean the same thing, but without definitions it is impossible to be sure.) Once again the taxonomist must return to the original specimens to resolve the question.

### **2.4.3 Unavailable data elements**

Lack of definition data is only one element of the thought processes underlying the summarisation of the data, which is not available for evaluation. A considerable amount of time and effort is spent gathering actual specimen data, much of which is not included in the summary account. Once the work has been published, this data is often discarded and not made available for re-use or verification [Diederich 2000], [Cannon 2001]. Participants generally regard this as an unfortunate side effect of the current process, where paper proformas holding descriptive observations made during the course of the project are discarded. A lack of rigor in note keeping and required speed of some of the undertaken work were also causes of this loss.



## Chapter 2: Taxonomy

Equally, when features (e.g. fruit) are not included in the descriptions without the author explaining why, it is impossible to distinguish between the feature not being present on the described specimen, the author accidentally missing the feature or the author considering the feature to be of insufficient interest to warrant describing it [Watson 1971]. Again the interpreter must return to the original specimens.

### 2.4.4 Consequences

All of these difficulties in using character data from descriptions not only make effective communication of concepts between taxonomists difficult, but they also have had subsequent consequences for the type of work that can be effectively undertaken. For example, the description difficulties mean that to complete a taxonomic revision involves returning to the specimens, using the descriptions as a guide rather than a source of data. This is possible for situations where there are only a few specimens or species, but working on large families (500+ species) is very time consuming [Jacobs 1969]. Engler's *Das Pflanzenreich* [Engler 1900-1953] was the latest world monograph on many plant families [Stace 1989]. However, many smaller families have been revised several times since then. Revision of large families is not attempted because the revision of one large family is likely to take one taxonomist their entire working career [Jacobs 1969]. If descriptions were recorded in a way that enabled interpretation with less need for re-examining thousands of specimens, the taxonomic process would be much quicker and tackling larger taxonomic works with a team approach would be more feasible.

Specimen descriptions represent a huge potential data resource, not just for future taxonomic revisions, analyses and the creation of identification keys, but for other biological disciplines such as biodiversity and ecological studies. However, these uses require the meaningful integration of data from different description sets, which, in the absence of both an agreed character model and particularly a shared descriptive terminology, is currently not possible.

These problems have led a number of taxonomists to suggest that there should be a standard approach to taxonomic descriptions [e.g. Diederich 1997, TDWG 2000] with terminology used consistently throughout all descriptions and the criteria used to describe one organism used in all other descriptions [Allkin 1984]. However, so far it

## **Chapter 2: Taxonomy**

has not been possible to agree a standard descriptive terminology or structure for taxonomic descriptions.

### **2.4.5 Summary**

As a consequence of the way in which character description data have historically been collected, recorded and made available, the interpretation, integration and re-use of character data is problematic. Typically neither the character concepts, nor the terms used to describe these characters have been rigorously or consistently selected and defined. The original thought processes and description data underlying the summary description are not available for evaluation as common practice has been to record only the 'characters' of interest for a given study. More detailed original descriptions, typically recorded onto paper proformas are not composed in a format suitable for reuse, and are often discarded, causing a significant loss of potentially useful information.

Subsequent interpretation of descriptions may therefore be ambiguous and descriptions from disparate sources cannot be compared or reused with any confidence of accuracy. Published descriptions are therefore of limited value to a modern taxonomist, who must often re-examine a specimen in order to interpret the original description or re-describe the specimen themselves.

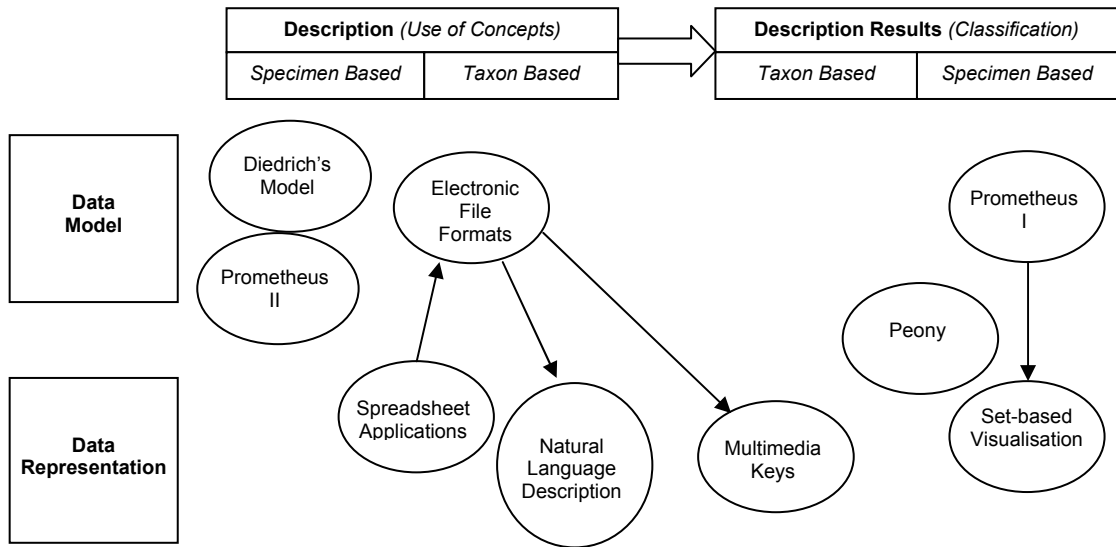
In the following sections of this chapter, this research identifies possible opportunities for using computerised tools to address some of these description issues, in order to improve the collection, recording, communication and use of character data within the taxonomic process.

### **2.5 Existing Computerised Tools for Taxonomy**

Currently, varying degrees of computerised techniques are utilised in the creation of plant descriptions and in the wider taxonomic descriptive process. Newly qualified taxonomists tend to utilise computerised techniques to a fuller extent than other, more experienced taxonomists, although there is also significant variation by individual [Cannon 2001]. There is, however, no consensus among taxonomists on which of the available computerised techniques to utilise. The provision of more advanced electronic

## Chapter 2: Taxonomy

support for taxonomists' working practice is limited to certain areas, as can be seen in Figure 2.6



**Figure 2.6: Selection of computerised tools to illustrate support for taxonomists' working practice**

There is support available for taxonomists in utilising the single and multiple classifications they develop (e.g. Peony [Hyam 2002], Prometheus I [Pullan 2000], Graham's set-based visualisation [Graham 2001]) but more limited support is available for the description process utilising the character concepts underlying the eventual classifications.

Some plant specimens also exist in electronic form but these are generally at a very basic level of detail and are not all in the same format. Mainly photographs and accession data are likely to be available. Some character notes may be included but this is unusual.

Spreadsheets of varying degrees of sophistication are utilised to record character observations during the detailed sort process, however, taxonomists must construct these spreadsheet proformas themselves on an ad hoc basis. A majority of record keeping is however still done in paper format, including drawings.

Supporting the use of characters in the description process are various electronic systems that attempt to capture the description data. Electronic descriptions have been

## Chapter 2: Taxonomy

seen as a way of making descriptions more uniform and informative. In general a common basic conceptual model is used, consisting of a number of characters (defined using name and data type) plus a series of possible states (also defined by name) declared for each character. Within a data set, the use of these character names is consistent, however there is not necessarily consistent use between data sets. The various electronic description systems fundamentally differ only in terms of data storage methods.

The primary example of these systems is DELTA (Description Language for Taxonomy), a data format for representing and manipulating taxonomic descriptions [Dallwitz 1980]. DELTA is designed primarily for holding generalised descriptions of taxa and is the most extensive electronic description system in terms of range of data types and inter-relationship of characters. A DELTA data set can include a glossary of terms used within it.

The following example shows samples of DELTA Character Lists. For describing a specific taxa/specimen, one of the listed states would be recorded.

***Flora of the Canadian Arctic Archipelago. Volume 1.  
Pteridophytes and Monocotyledons [Aiken 2001].***

***Petiole vestiture***

#78. *Petioles <general hairiness>/*

- 1. glabrous/*
- 2. hairy/*
- 3. glabrescent/*
- 4. scaly/*

#80. *Petioles <surface hairs length compared to petiole diameter>/*

- 1. hairs less than the diameter of the petiole/*
- 2. hairs more than the diameter of the petiole/*

*PETIOLE: the stalk that attaches the leaf blade to the stem.*

*GLABROUS: without hairs.*

*GLABRESCENT: initially hairy but becoming glabrous.*

*SCALES: reduced leaf-like structures, several cells wide.*

DELTA has a number of main drawbacks, which limit its usefulness. The user can use any terminology within the 'character' and its 'states', thus allowing the combination of more than one observable characteristic within one character and a general lack of clarity of the character concept. Whilst definitions of terms may optionally be included

## Chapter 2: Taxonomy

by the user, these are not comparable across data sets as they are expressed in natural language using undefined terminology. These limitations have arisen in DELTA because the focus is on maintaining consistency within data sets, without regard to either the consistent re-use of terms across data sets, or for the comparison of data sets from disparate sources. This situation is compounded by a lack of support for utilising the system, such as in determining appropriate definitions or character hierarchies. In practice this means that the defining facility is rarely used for the same reasons taxonomists rarely define their terms in current non-computerised practice. There is no visual representation of the proforma or description beyond spreadsheets and natural language descriptions. To provide data for electronic description data formats, some spreadsheets are however specifically designed to be able to export to data formats such as DELTA.

DELTA does have some additional limits on its use such as restrictions on representation of dependencies between characters [Newman 2001], inability to share lists of character states between different characters, and limitations on modifying character states when scoring specimens (for example to indicate a measurement is approximate).

Of the other electronic formats, LucID was developed by Cooperative Research Centre for Tropical Pest Management, in Australia, as an identification tool [CBIT 2003]. LucID software allows user to create identification/diagnostic systems. NEXUS [Maddison 1997] is the other major electronic format. It is an extensible file format for systematic information, and is designed to include diverse kinds of information, which can include taxonomic description data. These both suffer from similar problems to DELTA, and are less suited to general taxonomic description data, being designed primarily for other purposes.

DELTA and LucID can be used to create multimedia keys, which must be individually created for each classification. A taxonomic key is a particular use of descriptive data, which aims to help someone identify a particular unknown plant. The intended audience is usually not expert taxonomists. Keys use character data, but the data is tailored to the objective of distinguishing one type of plant from another. One drawback with keys when working with DELTA is that the same data set may not produce both good keys

## Chapter 2: Taxonomy

and descriptions because they place differing weights on the usefulness of different characters.

The limitation of current electronic description formats to consistency within one data set, and lack of rigour in the use of character concepts, is a barrier to communication, where taxonomists wish to effectively communicate their concepts. Such communication requires a more rigorous approach that allows description data to be collected which is comparable across data sets, uses defined atomic character concepts and allows taxonomists to express their concepts and the relationships between them. One attempt to create such a conceptual model for a description database is Prometheus II. The Prometheus II project has devised a data model for capturing descriptive character information, along with a supporting lightweight ontology for a group of plants (angiosperms) [Paterson 2004]. Again though, like the electronic description systems, there are no supporting tools to aid the user in creating descriptions using the data model and ontology.

Generally there is a lack of major supporting tools for the collection, representation and formulation of consistent, high-quality specimen description data.

### **2.6 Opportunities For Computerised Tools To Support Taxonomists' Working Practice**

Based upon the preceding analysis of taxonomy, this section identifies some of the areas where computerised tools could be utilised to support taxonomists' working practice.

#### **2.6.1 General Requirements**

In creating computerised tools to support taxonomists working practice, there are a number of user considerations which must be addressed. The user profile is an important aspect of any proposed system. Implicitly this discussion assumes the taxonomist is the user any tool is designed to support. To be explicit, the user profile is an expert taxonomist creating a flora or revision/monograph.

## **Chapter 2: Taxonomy**

### 2.6.1.1 Visual Cognitive Process

A high degree of reliance on visual elements by taxonomists was noted and observed during qualitative research. Taxonomic thought processes involve a substantial degree of maintaining a number of disparate elements within the individual taxonomist's mind. In order to do so, they use a highly visual thought process which aids recall within the visual modality. Furthering that visual element in any tool would be of value in its potential effectiveness.

### 2.6.1.2 Time Pressure

Taxonomists feel themselves under pressure to complete work quickly, due to environmental pressures. There is a need to complete taxonomic work speedily and progress to publication, so that the work can be used before the plants go extinct. This time pressure is a particular problem in areas of rapidly disappearing ecosystems.

An additional aspect of time issues is that taxonomists work at different levels of detail in creating descriptions and recording character information, depending on the intended audience for their publication. Taxonomists working at lesser levels of detail often do not have time to observe or record more detailed information. Any tool designed to support all levels of detail, must take this into account, with flexibility in the extent and detail of any descriptive data used.

### 2.6.1.3 Individualistic Working Practice & Cultural Considerations

Adoption of any proposed new computer tool is influenced by cultural considerations. Due to taxonomists' individualistic training, there is not a general consensus on the exact working practices adopted. In order to lessen resistance to changing these established individualistic working practices, any new system that involves changes to working practice should have some level of flexibility to increase likelihood of wider adoption.

Generally the opinions of taxonomists about computerised techniques are also individualistic, although some generalisations can be made. More recently qualified taxonomists are already using computer methods (such as DELTA) extensively, and during qualitative research were relatively enthusiastic about adopting new

## Chapter 2: Taxonomy

computerised techniques and rigorous description methods. Generally taxonomists are not anti-computerisation, but established taxonomists are likely to display resistance to changing their individualistic working practices.

### 2.6.2 Supported Tasks

Within the identified taxonomic working practice, specific areas and tasks, where electronic support for taxonomists is lacking but which could be supported by computerised tools, were determined. Figure 2.7 shows the main identified areas which are explored further below.

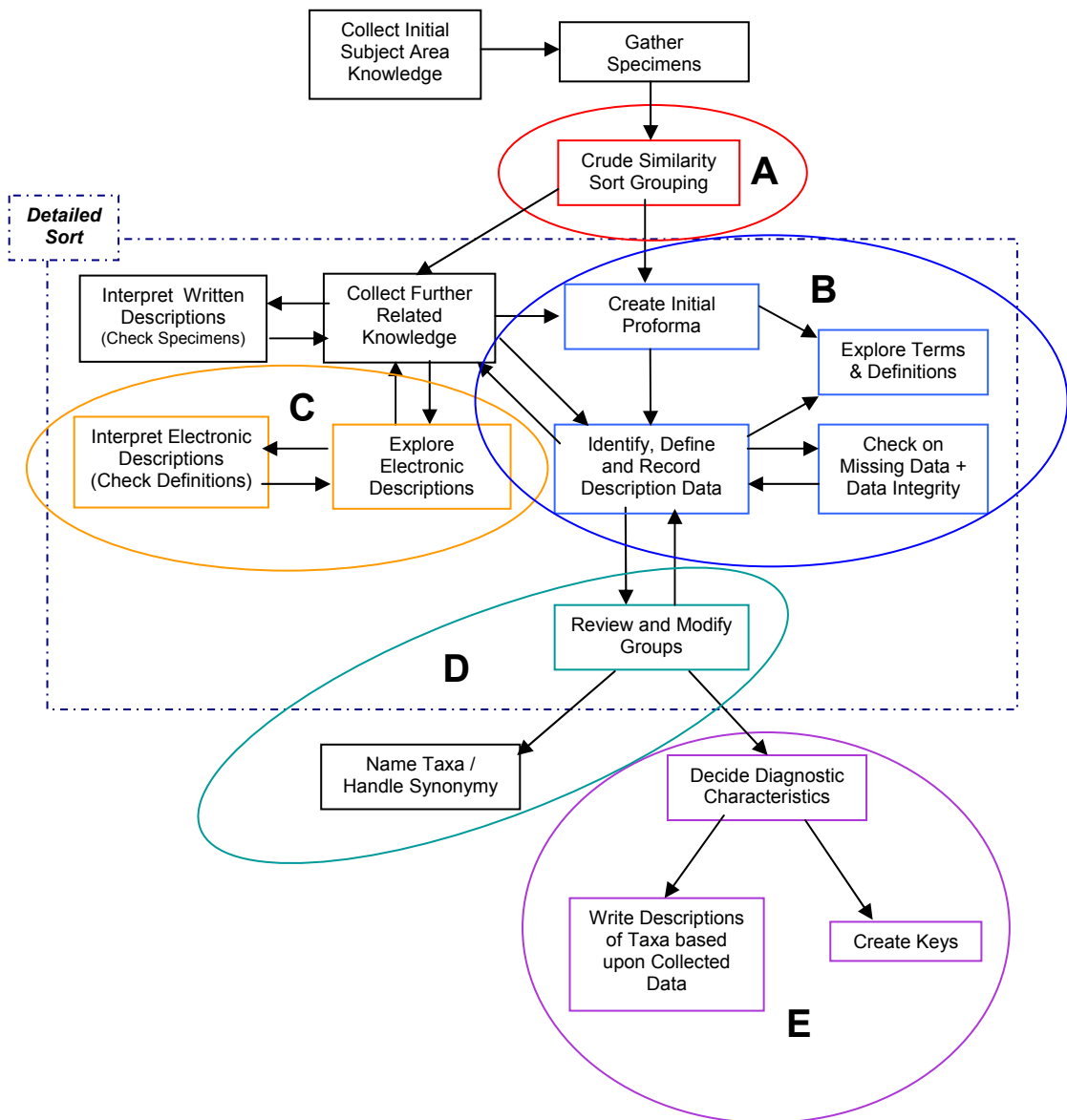


Figure 2.7: Potential areas for Computerised Tools



## Chapter 2: Taxonomy

### 2.6.2.1 Crude Sort

The crude sort (area A in figure 2.7) is one of the few areas where a number of taxonomists might currently work on a project together. Information systems can often make a contribution in aiding communication between co-workers. Further analysis however, shows substantial difficulties in supporting working practice in this area.

The crude sort utilises crude concepts of general appearance to determine the rough classification, rather than detailed character concepts. These crude concepts are less atomically defined, being more based upon an overall impression backed by experience and previous taxonomic works.

The crude sort does not lend itself well to recording data, as time is usually limited, with more limited physical access to computing resources. Access to computers in the collecting rooms where this process often takes place is improving however.

As little recording of character concepts is done at this stage, and there seems little likelihood of doing so, there are probably very limited opportunities for computer-aided work with character descriptive information. Uses of computer visualisation in this stage could potentially be informative visualizations of other data sets, for use in flora work, where more emphasis is placed on the crude sort and previous classifications.

If there were recorded electronic specimens available it might be possible to use the character concepts captured in these descriptions in place of actual specimens, to aid a crude sort. There is however, limited taxonomic work done on plant subjects, which have been previously covered in the detail required to provide the electronic specimen data necessary to work with. Time for projects in taxonomy is generally limited and so little work is done which would simply repeat earlier work.

### 2.6.2.2 Capture Description Data in Detailed Sort

In the detailed sort stage taxonomists conceive and work with the detailed character concepts. Serious issues were identified with the usage of character concepts during the taxonomic process. In particular the accuracy of the recording methodology and subsequent effectiveness for peer communication are compromised by current methods.

## Chapter 2: Taxonomy

The problems of definitions and description (see 2.4) are the major contributor to this. A tool supporting the capture of consistent, high-quality descriptive data during the detailed sort could aid taxonomists to address these quality issues.

In addition to the problems of the quality of descriptive data, a lot of potentially valuable data captured in the detailed sort (proforma description data and informal notes) are simply discarded following completion of projects (see 2.4.3). Providing an effective, yet relatively simple method of storing and accessing character data during the detailed sort process could alleviate this problem. As an additional benefit, storing such data in a database could allow the re-use of the descriptive data in other projects (such as later revisions, creation of more abstract descriptions of higher level taxa, creation of related floristic works, creation of taxonomic keys).

The proforma is the primary data store for character concepts during the identified taxonomic process. This central role can be seen in the detailed data flow diagram (data store D2 in figure 2.3). The proforma is not static in taxonomy - it changes for each taxonomic project as the characters of interest change, the proforma also evolves during a taxonomic project. It is transformed into a description when scores have been assigned for a specific entity. An interface tool which acted as proforma, recording and representing the character description data, could aid taxonomists in recording their data in a more consistent manner. Such a proforma tool would cover area B in figure 2.7, although due to the central role of proformas, it could also partly overlap with areas C and D.

Support for a consistent use of defined atomic character concepts in such a proforma tool would address some of the issues of clarity of descriptions for communication with other taxonomists and generally improve the quality of the data. Whilst data models such as Prometheus II [Paterson 2004] and Diedrich's character model [Diedrich 2000] partially address the question of an effective data model for storing character data in a database system, these systems do not include supporting methods to aid taxonomists in capturing or working with their data. The existing electronic description formats (see section 2.5) have some limited support for data capture, but generally lack support for informed and consistent description building.

## Chapter 2: Taxonomy

By incorporating definitions in a proforma tool, taxonomists could be encouraged to utilise definitions for their own recording of data, and allow them to access other taxonomists' definitions to clearly comprehend their concepts. However, during the research it became apparent that there was no agreed vocabulary used by taxonomists when describing their specimens. To date it has not been possible in botanical taxonomy to achieve universal definitions for characters or the terminology used to describe 'characters' although there have been some failed attempts (e.g. TDWG attempted to standardize the terminology for botanical descriptions [TDWG 2005]). Some taxonomists favour one specific type of term (e.g. Botanical Latin) although there is no agreement as to which type to use. There is however agreement that, at the least, they wanted to know which definition of a term an author was actually using and that thus all used descriptive terms should be defined.

To support the informed defining of terms, taxonomists may need guidance in determining an appropriate definition set, which would represent an additional task to the process of determining characters. Simply having definition text (or associated pictures) for terms is only part of understanding a term. Terms are not necessarily independent entities, in order to comprehend their full implications, it may be necessary to understand the context in which they are written (for example by seeing the alternate elements in a range of states or the wider structural context of a structure term). These terms and their relationships form a definition space which users could explore to determine appropriate definitions. In addition to understanding the relationships of terms, characters can themselves have relationships to other characters, which must also be understood in order to fully grasp a concept. The need to manage all these relationships requires defined terms to be backed by an appropriate set of relationship rules from a suitable data model. An ontology can be used to define and control terminology data, including its internal relationships and may be of value for this purpose. Taxonomists however, whilst admitting to the need to define terms, initially expressed concerns that using an ontology to define and constrain term usage in specimen descriptions might restrict the flexibility and expressiveness of current natural language descriptions.

Representing characters to users would be necessary for the functioning of such a proforma tool. It would need to represent defined character concepts, in an un-scored method in order to build an electronic proforma and enter data. It would also require to

## Chapter 2: Taxonomy

represent the concepts as scored characters to explore descriptions (e.g. for checking entered data or working with descriptions later in a project). Single and multiple character concepts would need to be represented within specimens, and depending on the extent of any tool, multiple characters across specimens might be represented. This aspect overlaps with the needs of a system to explore existing data sets (see 2.6.2.3).

The need for minimizing any time requirements on taxonomists could be supported by integrating the proforma creation, results recording and database entry requirements within one interface, which the taxonomist can also use for data exploration.

In summary, in area B of figure 2.7 users require a quick accurate way to record character information and observational data, along with associated definitions. If there is no easy way of doing this then taxonomists will likely continue to discard any data they do not consider to be important when writing a description, fail to define their terms, and will not take up use of rigorous descriptive data models that could improve the comparability of their data.

### 2.6.2.3 Exploring data sets

Exploration of description data sets is a procedure carried out at many levels, and lies at the heart of the communication between taxonomists. Exploration of existing data sets (area C in figure 2.7) allows taxonomists to seek out description characters which will be of use to their work and examine them in the context which gives them meaning; as well as to compare relevant descriptions. Exploration of the data set currently being worked upon (part of area B in figure 2.7) would help ensure taxonomists have recorded what they wish to record and help determine if there were structural areas where few characters had been examined. It would also inform the process of deciding upon diagnostic characters after the detailed sort was completed (part of area E in figure 2.7).

Exploration would require to be done at a number of levels, from overview to details. The provision of definition data would simplify and greatly speed the task of determining exactly what was meant by an author, obviating the requirement to go back to the original specimens.

## Chapter 2: Taxonomy

At the lowest level, individual characters and their constituent components must be represented. Whilst this is likely to be fairly simple where only one character is concerned, where a large number must be displayed (such as for a complete specimen description) the screen space requirements and interrelationships may present more difficulties. It is however at this level, that help is most required. When data sets become very large, the taxonomist has difficulty maintaining a clear mental picture of the description set upon which he is working. Visualisations of descriptions and description sets, could aid the taxonomist in maintaining a mental picture of the working data set by providing cognitive visualisation assistance.

Taxon descriptions may consist of the amalgamation of multiple specimen descriptions. These should be visualised without unnecessary repetition of data the multiple specimens hold in common, in order to most accurately and clearly represent the taxa. Other multiple specimen representations may also be required for comparisons across a data set and between data sets. Within one data set, the same empty proforma is likely to be used for all specimens. Representing the scores for the same character concept in the different specimens may be of help in determining the score in another specimen. Likewise such a representation could show the spread of scores across specimens, such as required for manipulating grouping data (see 2.6.2.4).

A necessary pre-requisite of exploring existing data sets is the existence of captured, usable data. There are attempts to interpret existing natural language descriptions (e.g. Multiflora [Lydon 2003, Multiflora 2003]) for computerised use, although these remain at an early stage. The electronic description formats lack consistency across data sets, making them more difficult for this usage. In order to explore effectively with electronic proforma data such as envisioned in 2.6.2.2, the data would however first need to be collected as it does not currently exist.

### 2.6.2.4 Manipulating Groups

Reviewing and manipulating grouping data (area D in figure 2.7) would seem to be an area of the detailed sort where taxonomists could benefit from visualisation technology.

Currently this task is done manually with grouping data being maintained by paper notes and/or actual piles of specimens. Tools to allow taxonomists to manipulate the

## Chapter 2: Taxonomy

grouping data in an easy form to allow them to visualise the effects on potential classifications of changes in the grouping data would potentially improve productivity and empower decision making. If additionally, a database of descriptions was available, a variable granularity visualisation of the complete data set of specimens under examination could be used by the taxonomist to inform and build a picture of the interrelationships of the data set. This could then be used to determine whether a character had a reasonable spread across the specimens.

There are currently no tools to do these tasks specifically, although there was another ongoing investigation covering this area at the time of the initial study, however no results have been subsequently forthcoming.

### 2.6.2.5 Key Creation & Writing Descriptions

Area E in figure 2.7 covers the final elements of the taxonomic process. Although these tasks are often done manually, there are some tools which attempt to help. Lucid [40] for example is already designed to help with the creation of keys. These tools are however limited by the lack of wide acceptance of their description data formats, upon which they rely. The advantages of another tool to help key creation would be of value if high quality data in a more acceptable format could be captured in the first place. Similarly a tool to automatically write pseudo-natural language descriptions of taxa, based on captured data, would only be possible if high quality data could be first captured.

## 2.7 Conclusion

From this initial investigation it was felt that the areas of most potential for computerised support lay in supporting more effective use of character concepts for description building. The accuracy of the recording methodology and subsequent effectiveness for peer communication are compromised by current methods. The main identified problems of inconsistent character selection, lack of character/terminology definitions and general data loss cause the difficulties in the clarity, comparability and re-use of descriptive character data for both taxonomy and the wider biological sciences. It was decided to investigate further the possibilities of a tool which supported

## **Chapter 2: Taxonomy**

the capture of good quality description data for taxonomists which would begin to address these problems, which have not been effectively addressed previously in this field. Such a tool would also have to address the possible visualisation of description characters and their definitions. Further investigations using storyboards and use cases were developed to assess and develop such an approach, this is detailed in chapter 4.

A literature review was undertaken to further assess the identified opportunities for computerised support, supporting the initial prototype tool developed to address and further investigate the character concept issues examined in this chapter. The review assesses the current state of data entry interface generation, suitable data models including the use of ontologies and other visualisation technologies.

## Chapter 3

# Supporting Information Systems Literature Research

### 3.1 Introduction

From the research detailed in chapter 2, it was proposed that a tool that captured good quality specimen description data for a database, supporting the use of a rigorous structured data model and a defined terminology to compose descriptive characters could help taxonomists address their description problems (see section 2.6.2.2 and figure 2.7B).

To put the discussion of techniques in context, a brief note is made here of the nature of the data being dealt with. Structured taxonomic descriptive data can be broken down into atomic characters of a structure, property, state format (see section 2.4.1). Users thus need to work with data that has structures, properties and states with defined relationships between them based on domain semantics (definition space) as well as rules on how to combine them with appropriate descriptive relationships to form character equivalents (description space). It was believed that there would be at least hundreds, if not thousands of structures, properties and states in any terminology, giving rise to a very large number of possible combinations. In addition, to clearly identifying the location of the physical plant structure to which a character refers is likely to involve more than one domain term for structures.

In addressing the issues of capturing description data, interfaces to databases are required for two primary purposes: data entry and data exploration. An interface supporting data entry is required to allow users to enter complex description data at a high standard of quality. An interface supporting data exploration is required to support the use of a defined terminology in building descriptions and to a lesser extent to explore existing descriptions. Tasks of data editing and detailed data analysis are likely to be less relevant.

This chapter looks at existing information systems techniques that would aid this



### **Chapter 3 - Supporting Information Systems Literature Research**

process. First the generation of suitable data entry interfaces for taxonomic working practice are discussed. Then the visual browsing of databases is briefly considered for exploring the definition space. Lastly, techniques for presenting, exploring and manipulating structured descriptive data are investigated.

#### **3.2 Tailored Data Entry Interface Generation Tools**

Creating appropriate, good quality data entry interfaces for databases is traditionally a difficult and time-consuming process for an IT expert. This mirrors the situation in GUI development in general. The most common tools used by designers are Rapid Application Development tools, Integrated Development Environments and authoring tools [Molina 2004]. These support skilled designers to create UIs, usually with some form of graphical control of the UI implementation. Whilst this speeds development [Myers 2000] and can empower designers, it does not support inexperienced designers in avoiding bad designs [Molina 2004].

Given the aim of supporting taxonomists' working practice it is important that descriptive data is gathered in a consistent fashion for all specimens in a given project. Using a project-specific proforma template for descriptions currently ensures this consistency requirement and an equivalent control mechanism is necessary for any supporting data entry tool. A data entry interface based on all description possibilities would not ensure this consistency, even if it were feasible given the number of descriptive possibilities. Building a new interface for each project is however equally infeasible given the number of such projects and the need for developer intervention. Consequently any approach to address the problem of capturing description data, would potentially include some degree of interface generation or tailoring by taxonomists.

Strands of research address the problem of supporting the generation of UIs. Of particular relevance are those strands which support automatic or semi-automatic generation of interfaces. Automatic UI generation is often based upon some form of model-based solution or abstract design. There is also research which aims to support interface generation by investigating guidelines and presentation aids to help the developer to develop effective interfaces by reducing the burden of design work. Relevant research is discussed below.

### 3.2.1 Presentation Aids & Toolkits

There are various techniques and toolkits which attempt to support the generation of suitable interfaces.

#### 3.2.1.1 Toolkits

Presentation toolkits (or user interface toolkits) generally consist of a set of customisable widgets, and possibly a design tool. Java's Swing toolkit is a good example of this, with a large collection of interaction objects (widgets), and utilising the standard Java language and development tools to create a database visualisation interface. Such kits are limited however. They tend to focus on the concrete interaction level, specifying the exact widget that will be used (e.g. a password box rather than perhaps a general abstract text input widget). For simple tasks this is a valuable tool, but the pressure on the designer builds up quickly with more complex applications, as there is little support for them. Whilst these could support the general design of an interface they would be too low level for end-users.

#### 3.2.1.2 Guidelines

Design criteria is an area where benefits might be gained from classification and standardisation. Codification of such criteria would aid in the creation of application independent presentation tools [MacKinlay, 1999]. Vanderdonck's Corpus Ergonomicus [Vanderdonck 1996] is a classification of a wide number (3700+) of ergonomic guidelines for human-computer interaction. It was hoped this would help in creating computer-aided presentation tools, however when they were used (in TRIDENT [Vanderdonck 1995]) to aid the automatic generation of interfaces, many hindrances were found because of their complexity and the difficulty to translate them for use by an automatic system. Weaker design principles were thus incorporated because they were easier to incorporate, however the load on the designer did fall [Bodart 1994]. Other work on the computer aided design of user interfaces has tended to support the designer by providing presentation guidance.

Work has also been done on visual placement; categorising different types and strategies of placement in order to help automate this aspect of a presentation tool. This was attempted based on GRIDS [Feiner 1988, 1990] and in an experiment by Gillo

[Gillo 1994]. To do so does however involve answering many questions to allow the application to determine the optimal placement strategy to suggest. [Vanderdonckt 1994, 1995]. A simple automatic layout strategy may be required to avoid the need for end-users to tackle such difficulties if the layout of an interface to capture taxonomic descriptions is affected by any tailoring for project needs.

#### **3.2.2 Automatic Interface Generation**

An automatic system that created an appropriate high-quality data entry interface would obviate the need for an expert UI developer to manually generate a new interface for varying project requirements. If appropriate tailoring could be completed by end-users, then specialist developers would not be needed at all to generate project specific data entry interfaces.

##### 3.2.2.1 DBMS and web based form and report generators

It is common for DBMSs (e.g. MS Access, Oracle, Paradox) and web generators (e.g. Macromedia's ColdFusion, Adobe's Dreamweaver) to automatically or semi-automatically present forms-based user interfaces to allow data entry to a relational database. These tools tend to be tied to the respective proprietary DBMS. The generated forms are generally simplistic, being designed to conform to the structure of database tables or views. The tools generally operate by using wizards and visual representations to explore and select the table views to be automatically represented in the data entry forms. They can, through such simple representations, enable non-UI designers to specialise what data entry fields to represent. Whilst this would be a useful metaphor for specialising a data entry interface for a project, there are significant limitations to these tools, beyond any proprietary links.

These tools can only constrain data entered to conform to the system data type associated with the table attributes, (for example by dragging the table attributes to XHTML form controls [Raggett 1999]). In most databases many attributes are stored as character strings, for which it is difficult to ensure consistent use or data quality, especially in terms of their semantics related to the domain of the attribute. To address many of the issues raised in chapter 2 concerning the quality and comparability of taxonomic description data, it is important that the semantics of the data are captured along with the actual data.

Whilst adequate for designing the simple interfaces (for which they are designed), the tools lack the flexibility and power required to build a more complex or non-standard interface. The structured description data envisioned in section 3.1 does not lend itself well to a simple relational database management system, where a user simply picks appropriate relational table views to generate 1 record per screen type forms. To display and query semi-structured data requires substantial programming [Petropoulos 2005].

### 3.2.2.2 Development Environments

User Interface Development Environments (UIDEs) as the name suggest provide a complete development workspace in which a dedicated user interface application can be developed. Presentation, task/dialogue and data components are usually included to some degree.

The DBMS and web based form generators mentioned above are examples of simple UIDEs. Another example of a simple and basic UIDE is IBM's Data Explorer system, which aims to visualise data by a 3 stage process: 1. Describe and import data; 2. Use a visualisation program to process data and build the final interface/visualisation; 3. Present the resulting image. The visual program thus created can be saved in a scripting language [IBM 2000]. Data Explorer is a limited program, but it does have the advantage of being able to connect to a variety of data sources. There are numerous different data sources for non-description data in taxonomy and the related life sciences, indeed to address this problem there is ongoing research efforts into data integration across the life sciences. It is likely that even if more description data was captured in databases, there is no guarantee of agreement as to the type of system. Being platform independent would thus be a worthwhile goal in a tool to support specimen description data entry.

### 3.2.2.3 Model-based tools

Automatic UI generation tools are commonly based upon some form of high level specification such as a model based or other abstract design, where a presentation model controls the selection and layout of UIs, based on the modelled tasks and/or domain. Model-Based UIDEs (MB-UIDEs) attempt to combine abstract modelling with a more systematic approach to interface development. Generally the main components of a

### Chapter 3 - Supporting Information Systems Literature Research

MB-UIDE are the modelling tools which the designer uses to build the models; the models themselves; automated design tools that cover those aspects of development not completed by the designer; and the implementation tools that convert the design into working application code.

The modelling tools used tend to be graphical interfaces for creating and editing modelling entities. These tools are usually forms based and are designed for expert UI designers to use. ITS [Wiecha 1990] is an exception, requiring text editing with a modelling language. Using these tools, the UI developer investigates and models their understanding of the domain and/or task (and possibly other aspects).

The models can cover domain (or data), task, dialogue, user and/or presentation aspects. Not all MB-UIDEs include all these models, but at least one of them must be explicitly modelled. Usually they focus on either the task or the domain model, with some approaches supporting automatic interface generation (e.g. MECANO [Puerta 1994], JANUS [Balzert 1996]) and others providing support and guidance to designers to finalise an interface (e.g. ITS [Wiecha 1990], Humanoid [Szekely 1992], MASTERMIND [Szekely et al 1996b], MOBI-D [Puerta 1997]). There is no standard format for the various models used in different systems (some example formats for domain models include C++ classes (UIDE [Foley 1989]), ER Models (TRIDENT [Vanderdonckt 1995], GENIUS [Janssen 1993]), OO-Object models (JANUS [Balzert 1996], TADEUS [Elwert 1995]), algebraic specifications (FUSE [Lonzewski 1996]).

Abstraction in itself does not free the UI developer of the need to select appropriate interaction objects (although they may only be selecting abstract versions, with the details of the concrete coding being done automatically [Zloof 1998]. To arrive at an interface design, there must be a clear computer supported relation from the declared abstract model(s) to the generated interface. Automatic generation systems use some sort of mapping between elements of the models to interface design elements in order to generate the interface. This mapping may itself be modelled as some form of presentation model as part of the modelling process.

Early model-based attempts were quite limited, concentrating on the application and data model. These were capable of automatically producing simple, generally static interfaces based on simple data models. Examples include JANUS [Balzert 1996] which

### Chapter 3 - Supporting Information Systems Literature Research

used hard-coded algorithms and Humanoid [Szekely 1992] which used design templates to generate an interface.

Later attempts such as ADEPT [Markopoulos et al 1992], MASTERMIND [Szekely 1996b], MECANO [Puerta 1994], TRIDENT [Vanderdonck 1995], TADEUS [Elwert 1995], and Teallach [Barclay 2003, Griffiths 1999, 2001] all improve upon this, with wider range of more sophisticated interacting models. These systems attempt to capture a richer understanding of the domain, to produce more dynamic, better quality interfaces. MECANO for example used a richer domain model than the simple data models used previously, that not only modelled the data, but also some of the domain relationships.

Whilst most MB-UIDEs have a presentation model to some extent, TADEUS and Teallach both utilise a dual level presentation model, with both concrete and abstract layers [Gray 1998]. This gives flexibility to the designer and to the user, as the user can choose to change abstract widgets later on, and the designer can make an abstract design, but has the capability to more rigidly determine a vital widget when required. Whether end-users should be concerned with changing the presentation is uncertain but it would allow the customisation of a general design.

Many MB-UIDE only allow a fixed set of interaction objects thus limiting their expandability to new visualisation techniques or application specific widgets. Some like Teallach, based on the Java Swing set, are expandable however [Barclay 1999]. Abstractions such as this, illustrate one of the bonuses of a MB-UIDE over traditional UI development, in that the quantity of low level coding required is not only minimised through the use of an appropriate development environment, but that low-level interaction code can be auto-generated from an abstract idea of the interface.

The model based approach aims to provide a more declarative system for designing an interface. Most of them are quite proscriptive in determining the design cycle, requiring one model to be completed before starting the next. Teallach is an exception to this, as it supports a more open development style, letting the developer work on the models in any order or simultaneously [Barclay 2003]. This claims to support the creative process, and overcomes one problem of the other MB-UIDEs, in that they are quite resistant to change, requiring full cycles to return to an earlier model, although it could be argued

### Chapter 3 - Supporting Information Systems Literature Research

that this enforced methodology was an advantage as it places structure on the development process. Some specific work has been done on determining the correct methodology to utilise for MB-UIDE, fitting it into the development lifecycle [Bodart 1995a]. However, there are still concerns about the requirements in MB-UIDE because of their rigid top-down methodology which leaves prototype construction to near the end, after domain and task models are fully in place. [Griffiths 1999].

Although a number of MB-UIDEs have been developed, their claim to functional status and the claim of some to be capable of providing links to a large category of databases (e.g. Teallach claims to theoretically connect to any Object Orientated DB), have not been fully tested. These approaches still require substantial investment by a UI developer, particularly if they are to be successful in creating a useful domain specific interface, and as Novak has observed '*Nobody will create applications using specifications (models), if they can do it faster directly editing*' [Novak 2003]. This is doubtless one of the reasons that model based approaches have so far failed to achieve widespread commercial adoption, despite a strong research base [Traeteberg 2004]. One reason for this failure may be the moving target problem [Szekely 1996a]. This refers to the situation where the GUI interaction metaphors have standardised in recent years, reducing the diversity of user interfaces and thus the need for an abstracted approach that can address different interaction styles. This situation may be changing however with the advent of more ubiquitous computing, where a variety of new interface challenges must be met on different types of displays such as PDAs, phones, and wall screens. These challenges could make model-based approaches more viable.

Within model-based approaches automated design has become less popular, as some researchers believe that good design requires human knowledge of tasks and domain requirements [Szekely 1996a]. Puerta's work shows this trend, moving from automatic generation based on a domain model in MECANO [Puerta 1994], to a task based design-support approach in MOBI-D [Puerta 1997]. Consequently, approaches which attempt to support the designer have received more significant attention, such as through generating multiple user interfaces for designers to assess (e.g. KnowiXML [Furtado 2004]). These approaches are less relevant to this research however, as we do not wish to involve designers for every changing project requirement, nor do we wish to turn our end-users into designers. It seems that the move away from domain model based automatic generation can be traced to the inability to model sufficient information

### Chapter 3 - Supporting Information Systems Literature Research

about the domain in a model, to generate an appropriate interface without human intervention. Is this however because so wide a net is being cast regarding the dialogs and interactions that could be required? In other words are the approaches too general and non-specific?

Recently a number of domain specific model-based automation approaches have shown promise e.g. SUPPLE [Gajos 2004], DIGBE for building control systems [Penner 2002], PUC for remote controls [Nichols 2004]. These do not attempt to solve the general automation of interface generation problem at once, rather concentrating on specific domains where a model based approach can overcome the problems of being too generic. Researchers have recently identified finding specific domains where automatic generation can be applied successfully as a challenge for user interface research. Addressing specific domains rather than pursuing purely general solutions, may lead to important new techniques for that domain, which could be applied to similar domains [Nichols 2005]. This research does address that challenge.

The model-based approaches to automatic generation have parallels with the taxonomy description capture problem, however in order to build a proforma requires an end-user to work with the model of the description domain data, instead of the developer that the standard MB-UIDE approaches envisage. Interface tailoring at both concrete and abstract levels is performed by a number of MB-UIDEs, for example to match user profiles, but again this is aimed at the designer. (e.g. DIGBE [Penner 2002] tailors to user profiles, FUSE [Lonczewski 1996] to user expertise). Szekely [1996a] identifies using the auto generation capabilities of MB-UIDEs to support end-user interface tailoring as a challenge for model based approaches. Tailoring to the proforma needs of projects would be valuable in taxonomy as well as tackling this research challenge.

Simple data models were found to provide insufficient information on the domain task to be used as the sole basis for a useful interface, leading to the use of domain models. One concept in current research which bears resemblance to a domain model is an ontology. Ontologies were thus considered for their ability to act as a domain model and to see how ontologies could be edited and presented, as this could have parallels with using the taxonomy glossary and data model to generate a proforma.



### 3.2.3 Ontologies

Using an ontology to control the data entry for a database has the potential to ensure that better quality data is captured and that data from differing data providers will be compatible. It may also allow a data entry interface to be created that allows domain users to enter data using terms with which they are familiar but which are clearly defined semantically. However at the start of this research there was no agreed ontology for taxonomic description, although a structured data model and associated glossary can be considered to be similar to a weak form of ontology.

#### 3.2.3.1 Defining Ontology

There is a great deal of literature concerning ontology, much of which while valid in its own field, is not relevant to this project. Originally ontology referred to a discipline of philosophy concerning the nature of reality, asking the question 'what is?', of the kinds and structures of objects, properties, events, processes and relations in every area of reality' [Smith 2003]. In the past decades, however, the concept of ontology has gained new meanings in other disciplines, particularly information sciences.

Ontology has become a loosely utilised term for a number of different approaches. Gruber's definition "*An ontology is a formal explicit specification of a shared conceptualisation*" [Gruber 1993a] is commonly cited in computer science literature, but is itself a fairly wide definition, open to interpretation. Another definition [Guarino1995] defines ontology as '*an engineering artefact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words...In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships; in more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation.*'

Generally ontological systems in non-philosophical disciplines aim to improve communications between entities in a field by ensuring that, within the scope of the ontological system, the entities precisely understand what each other means when referring to concepts and relationships. This is similar to the aims of the tool to support taxonomists by promoting better communication of ideas through improved description.

### Chapter 3 - Supporting Information Systems Literature Research

There are top-level ontologies, which are concerned with general categories (e.g. time, space, identity, quantity, etc) as opposed to domain specific ontologies (e.g. of geography, medicine, ecology). The former attempts (e.g. OIL[Fensel 2000]) are criticised for employing exclusively set-based ontology construction, which would make them unsuitable in many actual real-world applications. In taxonomy, a domain specific ontology could be envisioned that would simply define the terms for use in description and regulate how those terms could be related. Whilst this would be a fairly simple and limited ontology which did not attempt to model the entirety of the domain, such relationships would likely go further than just subsumption.

#### 3.2.3.2 Use of ontologies

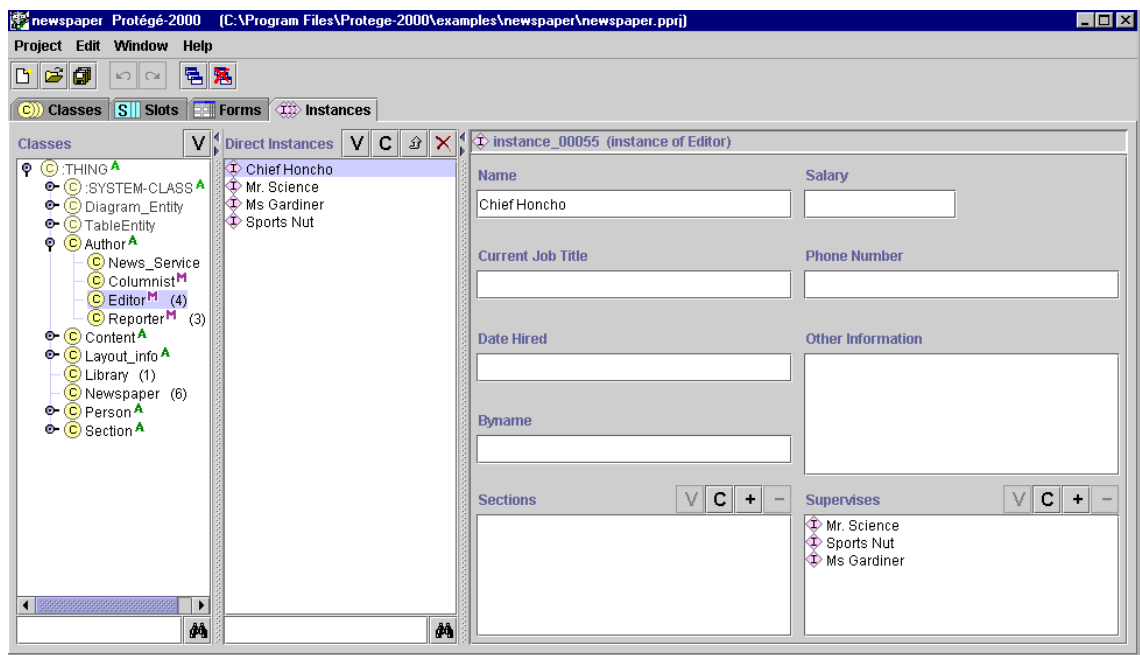
As mentioned, ontologies are widely used in a variety of contexts. A summary classification of ontology usages can be made, such as in the following table.

Uses of Ontology
For communication
Between implemented computational systems.
Between humans.
Between humans and implemented computational systems.
For computational inference
For internally representing and manipulating plans and planning information.
For analyzing the internal structures, algorithms, inputs and outputs of implemented systems in theoretical and conceptual terms.
For reuse (and organization) of knowledge
For structuring or organizing libraries or repositories of plans and planning and domain information.

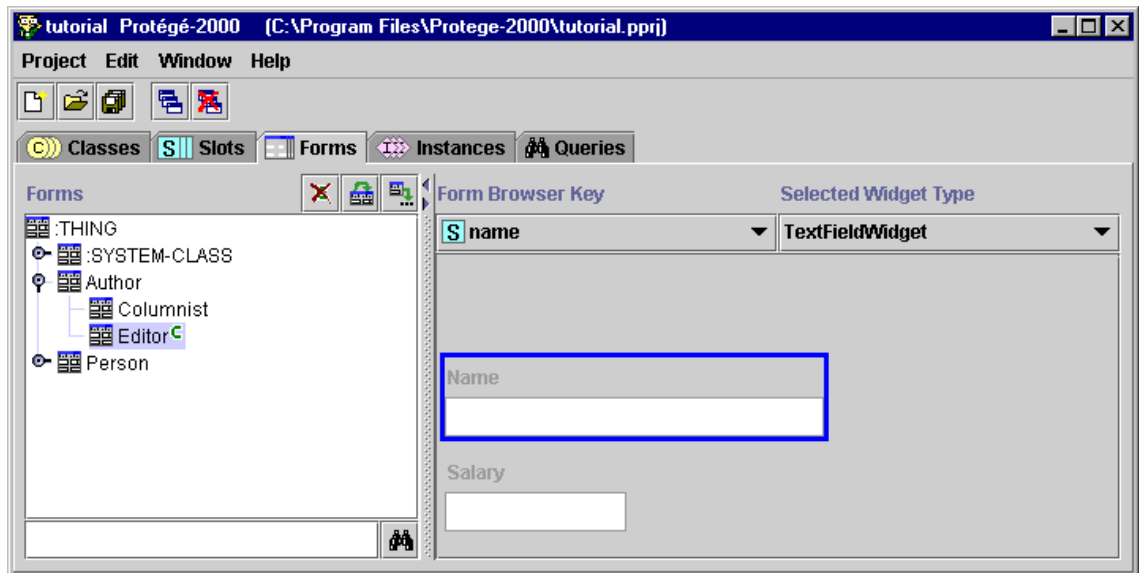
**Table 3.1: Uses of Ontologies [Gruninger 2002]**

In these terms, an ontology primarily for reuse and organisation of knowledge, and for communication between humans, as well as between humans and implemented computational systems would be of possible use in controlling terminology. There are knowledge base systems in existence which attempt to garner knowledge about a domain for these purposes.

Existing ontology based approaches for data entry are, however, generally still limited to using automatically generated forms-based data entry interfaces unless manual editing is used (e.g. [Gennari 2002]). A form tends to be generated for each class instance with IO widgets derived from the ‘slot’ data types. Links between forms are based on class subsumption relationships. See figure 3.1 for an example of a data entry form and figure 3.2 for an example of an editor for such forms that allows the representative concrete widget to be changed. These systems are designed to populate a knowledge base describing relationships between described instance items of interest, rather than regulating the capture of the description of a complex concept.



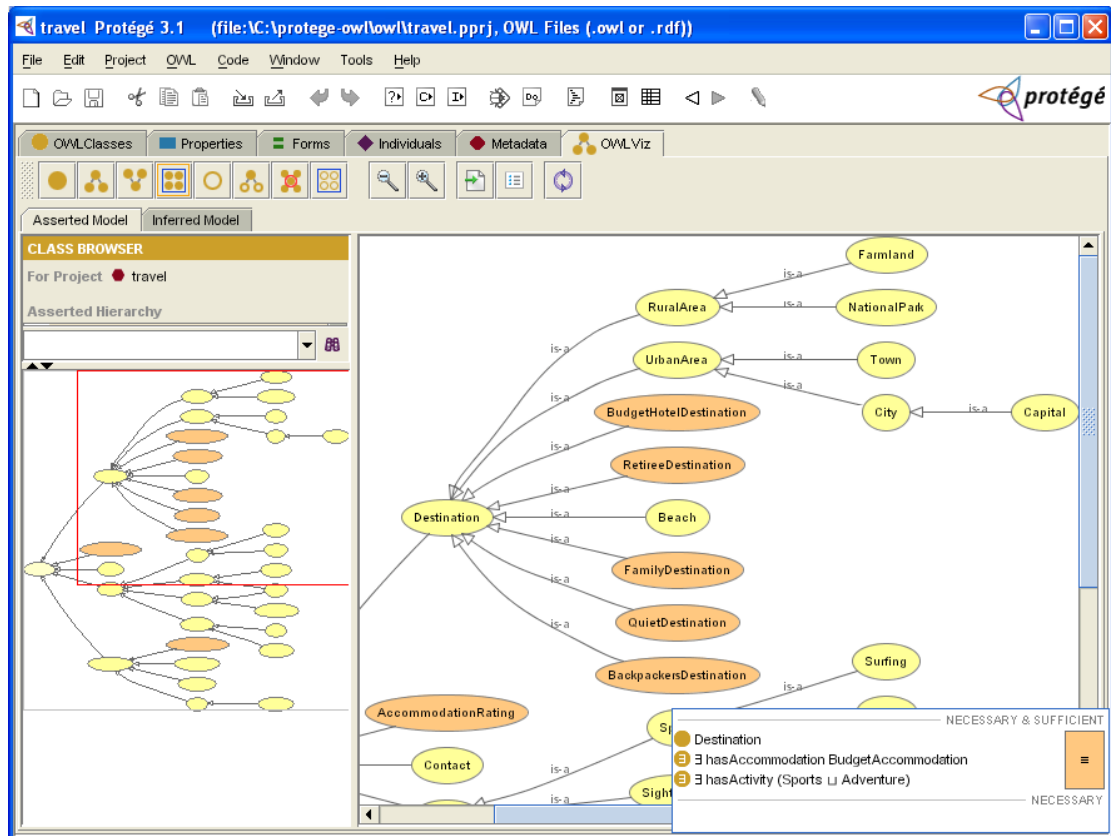
**Figure 3.1: Protégé Frames instances tab for knowledge acquisition [Stanford Medical Informatics 2005a]**



**Figure 3.2: Protégé Frames forms tab for tailoring instance editing forms [Stanford Medical Informatics 2005b]**

### 3.2.3.3 Ontology Representations

Editing an ontology has analogies with building a structured proforma using a defined description terminology. However, ontology editor tools are designed with IT specialists in mind. Again tending to the forms based interface, they conform to the structure of the ontology as might be expected, using ontology modelling language. Generally ontology editors represent the class hierarchy with some form of file tree visualisation, with the linked columns or separate forms, for associated instances, slots or other elements. Some of these editors do however use some graphical views of class relationships such as in figure 3.3, however even using all the screen space, only a small number of nodes are clearly displayed in these visualisations.



**Figure 3.3: OWLviz visualisation of an ontology, showing the subsumption class relationships. [Stanford Medical Informatics 2006]**

There has also been substantial work in the medical informatics field by ontologists with displaying ontologies visually, in this case mainly for exploration. Many ontologies in the scientific and medical fields concentrate on logical relationships between concepts. They seek to use one consensual nomenclature (standard definitions for terminology). In biological ontologies wide agreement on basic anatomical details/terms and detailed composition hierarchies differentiate these ontologies from the taxonomic case. Nevertheless sufficient parallels can be drawn in underlying subject matter, to warrant investigating some representative visualisation techniques used in medical ontologies.

The Gene Expression Information Resource Project [Davidson 1997, Baldock 2002] for example includes an ontology of genes and anatomy of mice with a 3D atlas. The 3D Atlas uses a high-resolution digital representation of mouse anatomy, using serial sections of embryos at various developmental stages taken from the ontology (see figures 3.4, 3.5). The ontology allows users to visually explore the tissue representations and relate gene expressions or anatomical terms to parts of the visualisation.

### Chapter 3 - Supporting Information Systems Literature Research

The visually impressive multi-pane biological visualisations, with alternate views of physical representations are not however possible in the proposed taxonomic description interface. A multimedia based approach to description representation and exploration may be possible for individual plant descriptions, but each one would require special work, unless suitable mapping abstractions could be found. They are more analogous to multimedia character keys, than to direct representations of character description and definition space. The use of linked display panes and details on demand are however shown to be a useful technique with such structured data.

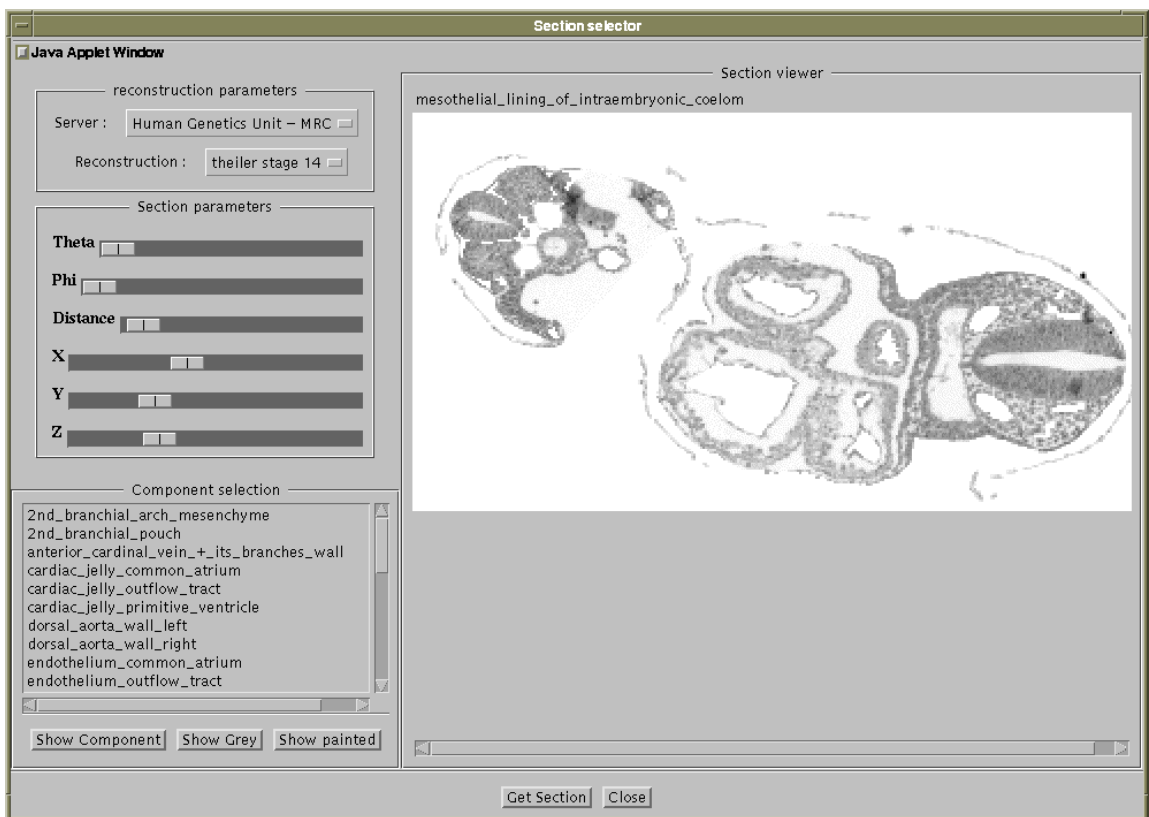
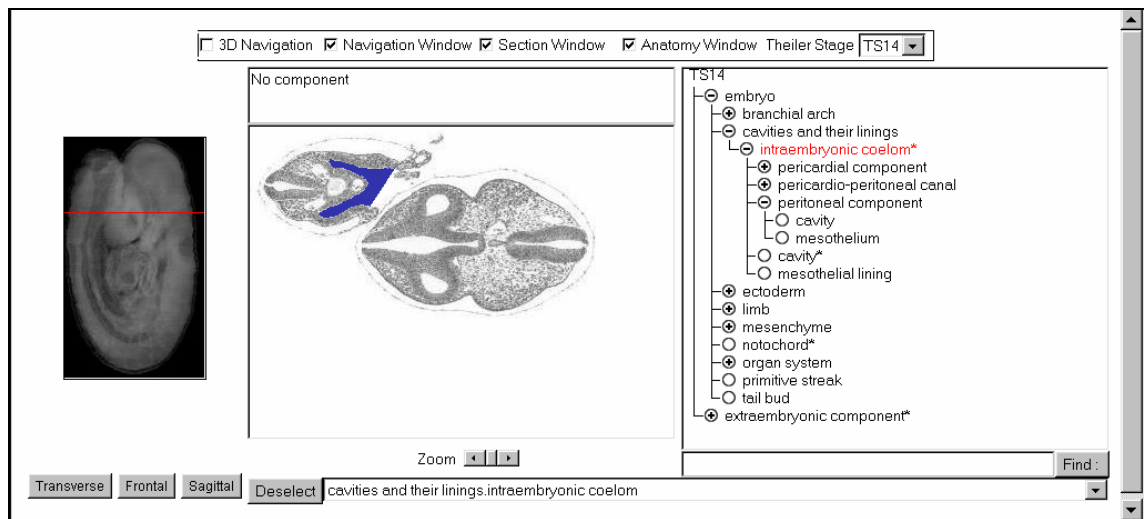


Figure 3.4: Gene Expression Information Resource Project - MOUSE ATLAS



**Figure 3.5: Gene Expression Information Resource Project - MOUSE ATLAS (with overview navigation pane)**

### 3.3 Browsing databases

Complex and scientific data sets often require the user to be able to browse the data looking for relationships, data inferences, and other analysis results. Exploring taxonomic description data in the type of supporting system envisioned in chapter 2 is an example of this type of task with this type of complex data. Such data exploration requires dynamic non pre-determined querying, which if using a traditional command line interface can be a cumbersome and involved process, requiring knowledge of the underlying data structure and of a data query language which an end-user would not necessarily possess. Additionally, command line interfaces are not a technique which makes best use of the advanced visualisation techniques which are well suited to aid this type of cognitive task.

Dynamic querying is an alternative to command line data languages such as SQL, for querying [Ahlberg 1994]. It is an interactive technique, which allows users to manipulate control widgets such as sliders, to control the data displayed. This dynamic approach involves constant and rapid updating of displays. Research in this area supports the use of the technique particularly for data exploration applications [Goldstein 1994, Schneiderman 1999]. Schneiderman [1999] claims to have found user enthusiasm for interfaces involving dynamic querying, based on the user's feeling of feedback and control over the database. He also found it most useful for users who were not experts in a query language (such as taxonomists). However, it was also found that

### Chapter 3 - Supporting Information Systems Literature Research

the dynamic query approach did not match well with other current computing technology. DBMS tools and graphical display systems are stretched to cope with the high speed of queries. To take advantage of dynamic querying, application specific programming is required. More problematically, dynamic queries rely on mapping meaningful alphanumeric data aspects to the control widgets. In the taxonomic definition space, there are no meaningful numeric aspects to the data and the alphabetical order of terms does not match with any natural ordering based on domain semantics.

Generic visual interfaces, capable of supporting such dynamic querying, and which can be used for a wide range of data sources have been proposed [e.g. Carey 1996]. Such an interface would not have the benefits of being tailored for a specific series of tasks. It would however, be useful for data exploration tasks. Various object orientated database systems have browsers, which allow users to browse using hypertext navigation (the standard form of navigation in OODB browsers). To be truly useful for data exploration, such a navigation system, needs to allow querying as an extension of browsing [Carey 1996], thus enabling a user to quickly begin to explore a database without the time-consuming complex task of creating an interface.

Visual database browsers have been researched from relatively early compared to general database visualisation research, e.g. KIVIEW [Motro 1988], Databrowse [Rogers 1988], OdeView [Agrawal 1990]. Their visual aspect was desirable for browsing rich data models to make the relationships between data objects explicit. Support for graphical querying from OODB interfaces was developed in applications such as Pasta-3 [Kuntz 1989], SNAP [Bryce 1986]. One follower of such early research, PESTO (Portable Explorer of Structured Objects) is a generic hypertext interface which supports both browsing and querying of OODBs [Carey 1996]. The querying aspect is designed as a natural extension of such browsing and not a separate aspect. Part of the GARLIC project [Carey 1995], PESTO is designed to be portable to any object orientated database. It uses a tool called PASTA [Carey 1996] – a set of Tcl procedures – to interact with the underlying OODB system. This system however, does not support the many advances in the presentation aspect of database visualisation. More recently, similar approaches have been developed for relational databases such as Microsoft's Query builder and for XML databases (e.g. XML-GL [Comai 2001], Xquery [Boag 2005]).



Generally visual querying helps IT experts form queries visually, however the user has to at least understand the database schema itself, including its underlying object model in order to formulate a query [Petropoulos 2005]. In this regard, they are thus unsuitable for taxonomists to use to explore their definition space.

### 3.4 Visualising Structured Data

As no existing tool was found that could adequately address the problem of capturing taxonomic description data, techniques were investigated that might contribute to an effective visualisation of a defined terminology and the related proforma equivalent. The structured descriptions of specimens form a conceptual description space and the supporting defined descriptive terminology form a conceptual definition space. Both of these need to be effectively represented in order to empower users to make informed decisions in their work. The field of information visualisation was investigated for suitable presentation and interaction techniques.

#### 3.4.1 General State of Visualisation Research

There is a common phrase "A picture is worth ten thousand words", the essence of which is that graphics can be used as common shorthand for communicating an idea. Graphics are useful both to present an idea, solve logical problems [Bertin 1981] or find patterns. Advances in computing technology have much improved the potential for effective use of the graphical medium, with improved rendering and real-time interaction available on standard PC computers [Card 1999]. Visualisation research has taken advantage of this situation – first in the scientific field and most lately to more general fields such as education, administration, military and business. The more general application of the graphical medium in computing is known as information visualisation, which Card defines as "*The use of computer-supported, interactive, visual representations of abstract data to amplify cognition*" [Card 1999, p7]. Visualisation research primarily aims to allow users to interact with data sets and perceive data relationships utilising more efficient perceptual and motor operations as opposed to a greater number of more demanding cognitive operations [Casner 1991, Roth 1997].

The origins of visualisation can be traced back to Playfair in 1786, however in the

### Chapter 3 - Supporting Information Systems Literature Research

modern era, Bertin and Tufte developed their respective influential theories of graphics in the 1960s-early 1980s [Bertin 1967, Tufte 1983]. These theories laid the groundwork in communications for what became the discipline of information visualisation. At first research concentrated on scientific visualisation, where physical data was being represented [see work of Tukey 1977, Cleveland & McGill 1988]. This particularly required work on visualising data sets with a large number of variables [e.g. Inselberg 1990, Mihalisin 1991]. This work in turn led to wider application in other non-scientific fields with non-physical abstract data. The early work on scientific visualisation (and later info visualisation generally) was quickly applied in the fields of computer graphics and AI to develop the ability to automate data presentation, in particular to automate matching data types and generating the graphical representation of the data. (e.g. McKinlay's early work on APT [MacKinlay 1999]; [Roth & Mattis 1990]; [Casner 1991]).

Visualisation research has investigated a range of techniques for achieving its goals as well as the general effectiveness of visualisation. Generally, interaction with graphical visualisations should be appropriate to the type of information, as well as the user's preferences, skill/experience level and the type of task involved [Mitchell 1997]. Generally matching a user's mental picture of the data can support the value of a visualisation [Graham 2001].

Making use of our range of perception to aid cognition is one of the driving forces behind visualisation, hence much research concentrates on attempting to utilise more of our perceptive abilities. Mapping data to multi-dimensional graphical representations is one popular area of visualisation research. Work on the effects of utilising different combinations of colours and shapes, aims to enhance the clarity of such representations. Other graphical visual data representations, such as starfield displays [Schneiderman 1999a], geographic representation [Roth 1990, 1997], and multi-dimensional query results with pixels representing data items [Keim 1999] attempt to visualise data which does not lend itself well to traditional representations. Other techniques such as multiple screens and the use of multi-media presentations combining sound, vision and animation all attempt to improve the presentation side of the visualisation equation. VR technology may in future yield further improved 3D representations allowing a fuller use of our perceptive capabilities with directional and in depth sound and vision [Rheingold 1991].

Explorations of defined terminologies have the potential to be problematic due to their sheer size causing information overload. User interaction with visualisations has been found to improve their effectiveness for example by dynamic filtering to avoid information overload [Schneiderman 1983]. Other HCI research has fed into this area by providing a basis of ongoing work on various forms of interaction such as voice. Some studies have shown multi-modal interfaces to be potentially valuable in improving effectiveness of interactions with graphic visualisations when used appropriately [Oviatt 1996, Roth 1997].

### 3.4.2 Interaction Techniques

It is by user interaction with visualisations that interactive information visualisations are potentially so powerful. The primary user interactions which can be applied to the information visualisation view (as opposed to operations on the underlying data) can be classified as focus, filter and link [Gershon 1998]. All of these techniques are of potential relevance to this project, particularly in rendering explorations of the description and definition space at a useful level of detail and relevance.

In visualising the description space, the number of components in a description and the limitations on screen space, mean that once the general area of interest is identified, relevant components must be brought to prominence, in order for the concept details to be determined. Focussing covers this interaction, which involves variable magnification of the graphical display to bring some elements to prominence.

Focus techniques include simple uniform zooming and non-distortion moving of point of interest. More details on the advantages and disadvantages of the various focussing methods can be found in the section on display techniques below. An important sub-set of this type of interaction is known as focus+context, in which, while areas of interest are focussed upon, the surrounding context is maintained in the graphical representation (albeit at lesser detail). Focus+context can be achieved by utilising a 3d perspective placing focussed objects in the foreground (such as in Perspective Wall [Mackinlay 1991]) or by a fisheye geometric distortion lens [Furnas 1986].

Filtering can remove or highlight the representation of information that does or does not

## Chapter 3 - Supporting Information Systems Literature Research

match certain conditions determined by the user. It can be applied to structured or unstructured attribute data representations. In the case of hierarchies, complete sub-trees are filtered out instead of groups of attribute data. Filtering can also be said to include manipulating colour or contrast of representations. Filtering differs from focussing by directly manipulating the displayed representation, whereas focussing affects the display space.

Filtering techniques were anticipated to be required for exploring the definition space, which otherwise could be too large and complex. Filtering techniques could also be relevant to exploring description space, although focussing techniques may be sufficient in themselves for this purpose.

In this research, text and multi-media definition data is required as a detail on demand in any visualisation of description or definition space. Further this research aims at interlinking the visualisation of description and definition space. Both of these features can be implemented using linking techniques. Linking means that an action carried out on one view of a data set will be carried out on other views containing that data. This mirroring of view interaction is very useful where more than one type of view is required to perform a task. This technique can be used for details on demand, whereby the user can select an element and gain a detailed view of that element in another view (such as a pop-up window or another pane of a multi-pane display). Linking is the key to integrating information across multiple applications. The VISAGE system [Roth 1997] is one example of this, where a number of basic analysis and reporting tools are combined into one integrated information workspace, creating its visualisations dynamically by dynamic scripting.

### 3.4.3 Display Methods

#### 3.4.3.1 Linear Visualisation

A definition glossary can always be represented as a flat linear structure such as a list, ordered alphabetically or in some other manner. Although a linear structure can also be imposed on plant structures, it is an arbitrary decision based on the loose concept of acropetally (from the bottom of the plant up) and from the outside in, and is thus subject to individual interpretation. Such an ordered list could thus be made of descriptive data, although it would not make use of the hierarchical nature of plant structure data. The

other elements of the description are non-linear.

Document and software visualisation are common although not exclusive subjects of linear data information visualisations (such as Seesoft [Eick 1992, Ball 1996] and table lens [Rao 1994] for source code, perspective wall [Mackinlay 1991], Tile bars [Hearst 1995]). These visualisations are ordered lists which can usefully relate order to some other feature of the data. However we do not need to see patterns in large data collections. Document visualization is not, restricted to linear data visualisations, as some visualisations also address the networked relationships between documents in a document collection – this aspect is addressed in 3.4.3.2 below.

#### 3.4.3.2 Network Visualisation

Definition space (see 3.1) in this research could be represented as a network. The size of definition data to be represented is relatively large with many potential links (e.g. authorship, synonymy, as well as domain semantic links such as type-of, is-a, part-of).

Node and link diagrams are the traditional way to visualize networks, using nodes for the data items and links (edges) for the relationships between them. The links can be directed or undirected, nodes may be unstructured (unlabeled), nominal (labelled), ordinal or quantitative. Compared to hierarchies, it is generally hard to form good information visualizations of networks. Apart from simply aesthetic considerations, crossing edges can be confused for nodes as well as lead to confusion over the end points of actual links. There are a number of mathematical algorithms designed to give a desirable layout, minimizing edge crossings, although these are generally restricted to fewer than 100 nodes [Card 1999]. A 3D approach can reduce the crossing problem, although it requires the user viewpoint to change as they are viewing the 3D graph in 2D. In addition 3D node-link graphs also suffer from occlusion and producing effective depth cueing. As with hierarchical representations non-spatial cues such as colour, shape and transparency can provide extra dimensions to a view.

High data volume, such as in a fully populated definition space, also causes challenges in producing effective node and link visualizations, as the screen becomes very cluttered and the crossing edge problem is magnified very quickly. There are three ways usually used to reduce network data to manageable size: aggregation (for large numbers of links/nodes); averaging (for large numbers of time periods); threshold and exception

### Chapter 3 - Supporting Information Systems Literature Research

reporting (for detecting change) [Becker 1995]. Aggregation is a technique, which is of possible utility in this research, to cope with the large number of nodes in definition space. Alternately filtering interaction techniques will be required to reduce the definition space to manageable size.

Document collections and more lately, World Wide Web structures are a common subject of network information visualisations (or transformed hierarchical ones). SemNet [Fairchild 1988] is one of the earliest 3D network systems. Of particular relevance, SemNet is designed to handle semantic networks (between prolog modules in the original research). Being a knowledge base viewer, it has some similarities with representing a defined terminology for descriptions. SemNet employs a 3D node and link diagram as its primary graphical encoding. Developed for a large knowledge base of Prolog rules, the links show the connectivity (link colour representing the kind of relationship) between modules (sets of prolog rules). Fisheye distortion is utilised for focusing on areas of the representation.

SemNet does not seek to represent any relationship between the prolog modules and their usage. In SemNet details of the knowledge base are de-emphasised and the structure is emphasised. The definition space visualisation in the proposed taxonomic character interface however needs to work with descriptions and the details of the definitions emphasised, not the structure of the data.

As mentioned above, there are a number of mathematical algorithms to determine node positioning. One type of algorithm is used to place nodes based on the degree of similarity of a chosen data criteria. These Spring-Mass algorithms can be used to organise a graph structure by calculating suitable links, giving 'strengths' to them, and giving 'weights' to the nodes. Typically these 'strengths' and 'weights' are generated from user interaction, such as in the Hyperspace [Wood, 1995] system of visualising World Wide Web hypermedia structure, where the user chooses areas or keywords, and related pages concerning the topic move close together and others are repelled. The visual proximity leads the user to perceive the nodes share relevant aspects in common. The model uses algorithms based on the 'springs' to move the nodes in a series of moves to minimise the 'conflicting attraction/repelling forces'. This oscillating motion can cause some difficulty in viewing the representation. The representation can only be used when suitable weights can be calculated.

### Chapter 3 - Supporting Information Systems Literature Research

The metaphor has, however, been quite widely used (e.g. Chen 1997, Hendley 1995, Donath 1995, Xiong 1998). One such use has been in the primary display of a semantic network of World Wide Web documents [Fowler 1996]. This application, called Document Explorer is one of a number of visualisations of document collections based on semantic content [e.g Fowler 1992, Benford 1995]. Galaxies and Themescapes [Wise 1995] are another way of visualising document collections based on their contents, in these cases, the subjects of the documents. Galaxies render the 3D relationship of documents down into a 2D scatterplot, clustered by similarity. Themescapes utilises a 3D landscape with height denoting theme strength and valleys, cliffs etc denoting relationships between document and their component themes. All these semantic networks differ from the envisioned defined terminology, as they form semantic networks based on keywords attached (or derived from) the documents. In taxonomic descriptive terms such content-based relationships do not exist, as no presumptions as to the meaning of a definition are made (nor is there an agreed vocabulary to define subject areas). Links based on semantic relationships in an ontology of descriptive terms could possibly perform a similar function, however this type of visualisation is probably more complex than is warranted by the semantic links likely to be agreed upon.

Visualising multiple taxonomic descriptions for comparison was considered of possible utility for informing data entry, although rigorous comparison of specimen descriptions was not part of the identified focus of the taxonomic research issue. Approaches to visualising multiple structures have traditionally utilised animation (such as Huang's animated huge graphs [Huang 1998]) to show change, or numerous snapshots (e.g. time tube [Chi 1998], Turo and Johnson's treemap based technique [Turo 1992]) displayed together. The approaches mentioned above though, are aimed at visualising change in hierarchies, rather than comparison. The limitations of screen space in the numerous snapshot representations approach would be equally found in any display incorporating multiple full hierarchical description representations. Other multiple structure approaches include Multi-Treemaps [Furnas 1994] which re-organise existing hierarchies to give different viewpoints (with different roots) on it, which is less applicable to this project. Graham's [2001] set based taxonomic visualisation can show multiple hierarchies, but is concerned with change across taxonomic hierarchies, which re-use the same nodes. Given the limited value of such comparisons for the purposes of

### Chapter 3 - Supporting Information Systems Literature Research

informing data entry and the research investment required, simultaneously visualising multiple descriptions was not taken further.

#### 3.4.3.3 Hierarchy visualisation

In this research, description data can be seen as having a hierarchical structure, based upon a compositional hierarchy of descriptive structures, which can each have a number of related properties, which in turn can have a number of related states as a value domain.

Hierarchical data structures are a popular subject of information visualisations. The structure lends itself well to storing, classifying and manipulating data. It is widely used as file systems, taxonomies, military and business organisational structures, etc. Indeed the advantages of a hierarchy for visualisation has led some people to transform their network data into hierarchies, conforming to a greater or lesser extent to the primary limitation of tree – only one path going up from a node.

A number of advances on simple indented lists and tree diagrams have been developed. Two main methods of indicating structure in hierarchies exist: enclosure and connection, both have been used in information visualisations.

One early popular enclosure technique is treemaps [Johnson 1991] such as seen in figure 3.6. This is a space efficient 2D representation which makes maximum usage of available screen space, using enclosure to indicate the hierarchy. The top-level objects in the tree form large rectangles, the next lowest layer in the tree form rectangles within the first, and so on (like a venn diagram). The size of the rectangles and the colour are used to denote elements of the data. Individual elements can be focused upon and details viewed. Levels of the hierarchy can be filtered out from lowest to top.

The metaphor gives prominence to the lowest-level objects at the cost of visualising the internal structure. It is effective for trees where there is a quantifiable variable, especially if large values are important; however it is not so useful in non-quantifiable cases. Treemaps also have problems with unbalanced trees, where the number of levels of hierarchy are not at all homogenous, thus losing the efficiency of the space saving metaphor. Botanical descriptions cannot quantify many of the elements, making this analogy difficult to apply to this research. In addition the structure of the non-leaf



### Chapter 3 - Supporting Information Systems Literature Research

elements can be difficult to identify in the visualisation, which would cause problems in if used to explore the description space.

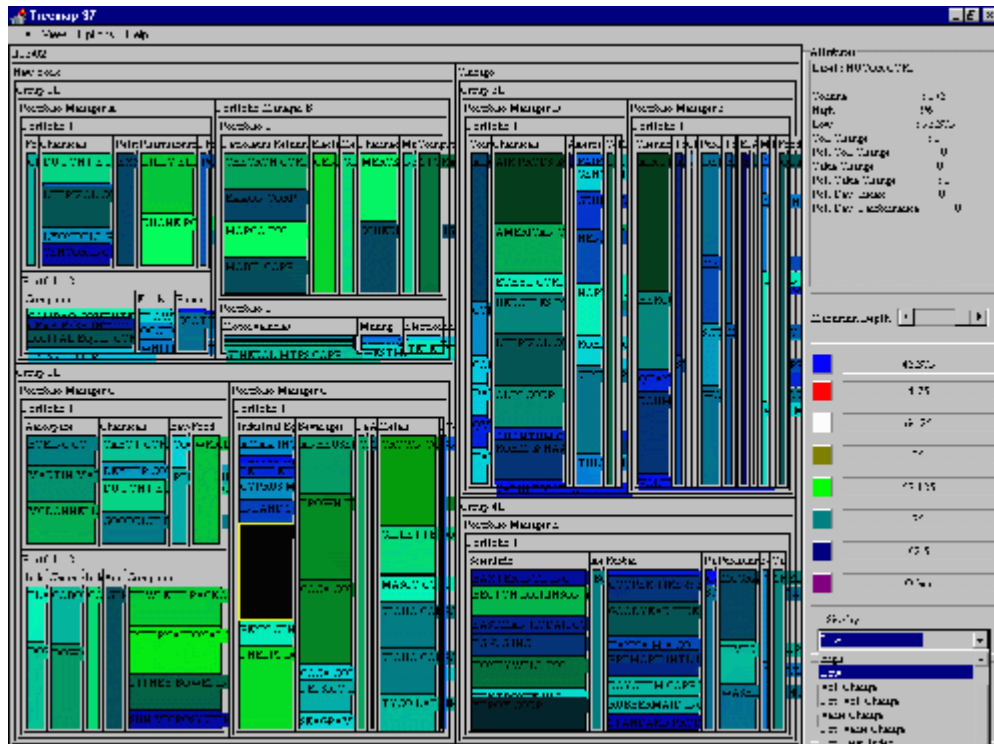


Figure 3.6: Treemap of stock holdings

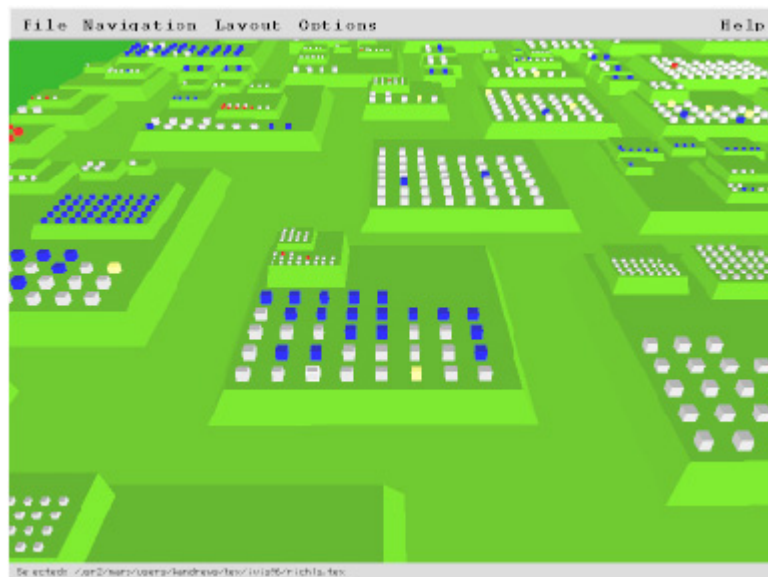


Figure 3.7: Information Pyramid of file directory [Wolte 1997]

There are other enclosure techniques which can offset these difficulties by using a 3D effect such as Andrews's Information Pyramids [Andrews 1997] seen in figure 3.7, which is designed for a file system, though the technique can be applied to other

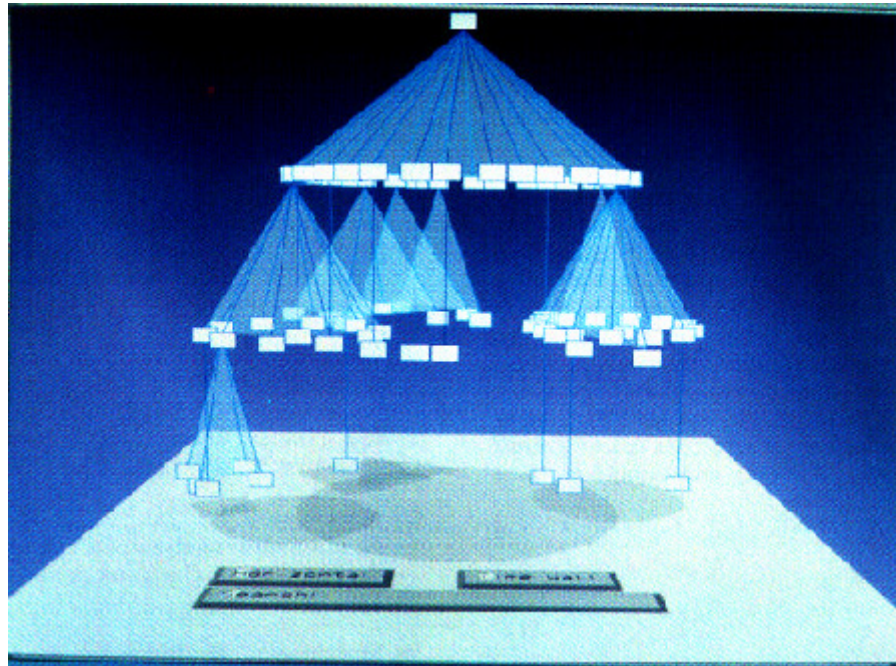
### Chapter 3 - Supporting Information Systems Literature Research

hierarchical data. Basically, these use a similar idea to tree maps but using a 3D effect, to give height and perspective, giving a clearer picture of the underlying structure, whilst still making the lowest-level objects clear on top, and allowing some elements of the tree to be given prominence by perceived closeness to the viewer. A plateau represents the root level and smaller plateaus on top represent the sub-trees, separate icons at the top represent the leaf level objects. Like treemaps the size of the plateau areas represent different attributes of the data. In addition to the 3D focus, a basic zoom technique is supported, although when using it, the wider view of the data structure is lost. The clearer view of the underlying structures would be of value in browsing description space, but the problems of quantifying elements remain, making the use of enclosure techniques not as appropriate for our case.

Using connection in a traditional node-link diagram with a tree like structure is a common technique for representing hierarchical data, but one that quickly can become cluttered and difficult to fit on screen as the size of the tree becomes large in any dimension. The most common hierarchy visualisation is the file navigator, such as windows explorer. Most presentation toolkits have a variation on this construct, such as JTree in Java. Whilst they can be subject to problems of screen space in large unbalanced hierarchies, these visualisations usually allow users to expand and contract branches of the tree to focus on interesting areas. This representation of a hierarchy is commonly used and familiar to most users. Example can be seen in the displays of the ontology instance entry forms such as figure 3.1, where a file tree browser is in one column of the display. The file tree metaphor would appear to be a standard basis to compare other visualisations to.

### Chapter 3 - Supporting Information Systems Literature Research

Cone trees [Robertson 1991] are an early way to overcome the limitations of screen size. Developed by Card et al in the early 1990s, they use a 3D representation, laying out the hierarchy uniformly in 3 dimensions. Nodes are drawn as small rectangles, which can have text on them (cone tree versions only place text on selected nodes for readability).



**Figure 3.8: Basic cone tree model [Robertson 1991]**

Cam trees is an alternative horizontal layout which can display text for each node (due to different dimensionality). When a node is clicked upon the cone tree rotates to bring the selected node and its ancestors to the foreground and highlighted. This rotation is animated to allow the user to maintain their mental picture of the hierarchy structure. This animation can consume significant graphical computing resources in large hierarchies.

The primary problem associated with cone trees is occlusion, in that nodes in the foreground, occlude those at the back of the display. The system is good for displaying uneven tree shapes, as it is quickly apparent using perception rather than detailed cognition, as to where the balance of the data lies. On the other hand, Robertson [1991] believe that the effectiveness of cone trees in large balanced hierarchies was low as it is difficult to distinguish between virtually identical substructures from a distance.

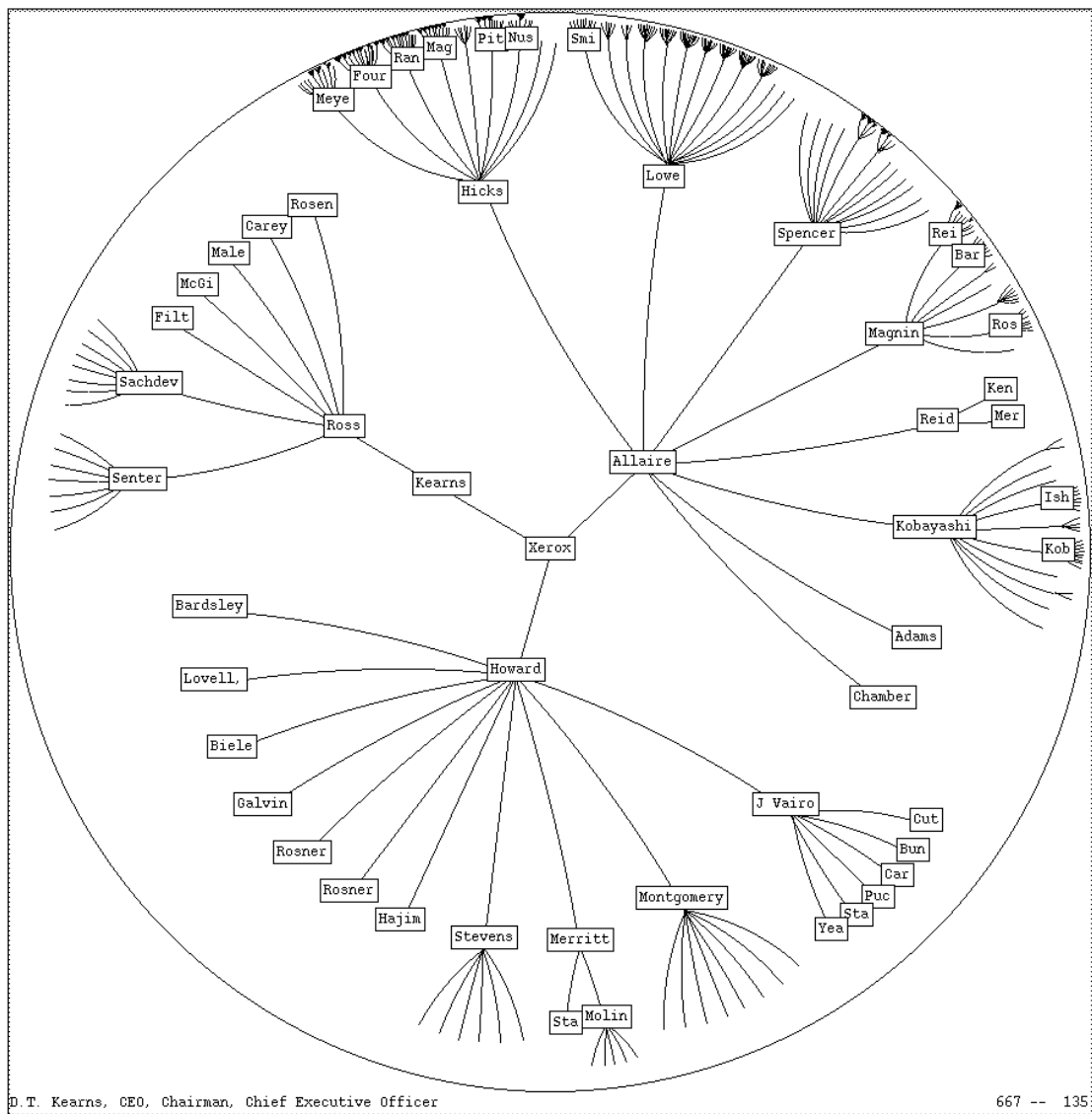
Other work has improved upon the basic model. Jeong and Pang [1998] tackled the

### Chapter 3 - Supporting Information Systems Literature Research

occlusion problem using their Reconfigurable Disc Tree, reducing the visual overlap of nodes, although labelling nodes was still difficult. Carriere and Kazman [1995] developed a system called fsviz which attempts to improve matters by a number of additions, including removing visual clutter, making better use of colour and node shape, providing fish-eye viewing, providing user controlled interactive filtering (via dynamic querying). Fsviz and cone trees generally are best suited for structural and trend related queries, as opposed to seeking a particular file. Taxonomic practice requires both general structural browsing as well as searching for particular description element types, so a visualisation which is more generally useful is required, although some adaptation of cam tree ideas may be valuable.

The Cheops Pyramid [Beaudoin 1996] is another hierarchical system, designed for very large hierarchical data sets, such as the dewey decimal system, with millions of nodes. It uses overlapping sub-trees (stacked overlaid triangle representation) and selected nodes can be focused upon. Due to its dense format, nodes other than the selected sub-tree are generally occluded. The botanical taxonomic description system is not of a size in that order of magnitude, that such a system would be necessary.

Focus+context interaction techniques were identified as being of use in representing description space, as they allowed exploration of detailed descriptions, whilst maintaining an overview perspective of the specimen description. One display metaphor that is based on such interaction is Hyperbolic Discs, which were proposed by Lamping and Rao [1994] as a method of visualising large trees. A fisheye focus and context lens is used to view a hierarchy, which is uniformly laid out on a 2D hyperbolic plane and then mapped to a circular display area. Nodes are labelled and when selected (can also drag any point to any other point), an animated sequence moves the selected node to the prominent area at the centre of the screen. Areas near the centre become magnified, areas further away shrink, allowing the user to maintain the visual context, whilst browsing and maintaining a picture of the hierarchy structure.



**Figure 3.9: Hyperbolic Disc [Lamping 1995]**

With a potentially relatively high number of descriptive structure nodes believed likely to be found near the root, displaying all of these in limited standard space was a potential problem in the proposed interface. Hyperbolic lenses allow larger hierarchies to be displayed in less screen space than a traditional layout. Conventional 2-d layouts of large trees are problematic, because of exponential growth – if leaf nodes are to be given adequate spacing, then nodes near the root must be placed very far apart, obscuring the high level tree structure, and leaving no space to display the context of the entire tree. Lamping claims up to 10 times as many nodes can be displayed with a 2D hyperbolic disc over a traditional 2D tree [Lamping 1995]. However, taxonomic description node growth is not generally so exponential beyond the initial level obviating some of the need for hyperbolic space to expand the effective space available for representing a specimen description's structure hierarchy.

### **Chapter 3 - Supporting Information Systems Literature Research**

A study [Pirolli 2001] into the effectiveness of hyperbolic viewers found they can be most effective when the text attached to the nodes has a high degree of information scent which can be followed to a distant goal, using the greater structure context available during browsing (allowing users to navigate more nodes, more efficiently). When this useful labelling was not present, the study found there was little advantage over a standard browser. Although not specifically using hyperbolic views, Schaffer's user studies [Schaffer 1996] also found advantages of general fisheye distortion lenses, as opposed to zoom lenses, in navigating hierarchically clustered networks. The problem with hyperbolic views is that they, and focus+context distortions in general, can be disorientating to the user, because of the level of visual distortion.

Hyperbolic approaches were considered to be a possible alternative to the file navigator approach for representing an overview of description space in the interface, although with reservations about user disorientation and the screen space requirements of a circular display.

Core Trees [Yang 1999] adopt a superficially similar 2D view to the hyperbolic disc, placing a hierarchy within a circle. It does not however use hyperbolic distortion for changing focus, instead as focus changes and selected nodes are moved to the centre, descendants of this selected node are pulled into the circle and other nodes are pushed out of the circle display.

#### **3.4.4 Visualisation Summary**

Some visualisation techniques can certainly be of potential use for some of the tasks needed, but no single visualisation can directly present the hierarchical description and associated definition model. Integrated workspaces embracing multiple linked means of visualisation are a good concept, but need to be tailored. Likewise the value of focussing and filtering techniques can be seen for exploring description space in a potential interface. Specimen description space can be represented by some hierarchical systems, although none are without some difficulties in adapting to this use. The basic file tree visualisation is probably the most useful, as users are likely to be familiar with the metaphor and it is relatively compact. Hyperbolic visualisations are another possibility for use as an overview of description space. Filtering techniques would offer one method of controlling the definition space, reducing it to manageable size.

### 3.5 Conclusion

No single tool exists which can solve the problem of capturing structured high-quality description data; however a number of techniques could be utilised or extended as part of a solution. Automatic generation of a data entry interface based on tailoring of the defined terminology forms the basis of an approach. Existing approaches mapping a domain model to an interface have parallels to this but do not support end-user editing or the needs of high quality data entry specifically. The concept of an ontology overlaps with that of a domain model and can also be used as a basis for data entry. Ontology based knowledge acquisition tools have similar drawbacks and are also aimed at a wider form of ontology than the structured data model/defined terminology model. Building an electronic proforma equivalent can use linked views of the elements of the description space to allow the interface to focus on the details of definitions or descriptive concepts whilst maintaining a view of the description context. Hierarchical views of description space will be needed with the file tree metaphor being the simplest yet probably most effective. A hyperbolic view is worth pursuing as a possible alternative. Filtering techniques based on user-controlled widgets may be able to give users a reasonable view of definition space that is sufficient for their purposes without a complex network visualisation.

## Chapter 4

# Capturing Description Data for Taxonomic Projects

### 4.1 Introduction

In chapter two it was seen that a system to capture high quality specimen description data during the detailed sort process (see 2.6.2.2 and figures 2.2 & 2.7) could be of benefit in addressing issues of clarity, consistency and comparability of taxonomic descriptions in an area with limited IT support at present.

From the research described in chapter two there emerged certain key attributes of any such envisioned system. The system must capture description data from taxonomists during their projects for storage in a database. This data should be structured to allow other taxonomists to interpret and compare, in an unambiguous manner, data from disparate providers who used the same data structure at a later stage. One of the primary elements of this data format should be the use of defined terms. While supporting such an appropriate data format, the system must still allow freedom of expression to taxonomists and be able to capture all nuances of data currently captured. The description data must be collected on a consistent basis for all specimens in a given project. However, users require to customise the data collection for individual projects, without the need for outside IT expert help, in a relatively simple and timely manner that minimises any extra burden on working practice. Additionally, matching the user's mental picture of the data would support the high degree of visual cognitive thinking involved by taxonomists during their projects.

Current tools do not meet all these attributes as seen in chapter 3. However, related research did suggest that ideas from model-based automatic interface generation could be adapted to allow end-users to model a proforma using a structured data model and defined terminology. This proforma would contain a model of the descriptive data to be collected on each specimen for a given project. Using some form of mapping from the data model, the system could then display an appropriate data entry interface. Information visualisation and other research discussed in chapter 3, also suggested ideas



#### **Chapter 4 - Capturing Description Data for Taxonomic Projects**

for the presentation of a structured proforma and defined terminology that could be adopted (e.g. hierarchical file tree and hyperbolic displays, filtering techniques and linked views).

The feasibility and parameters of a computerised system designed to capture specimen descriptions, containing the key attributes identified above, were investigated using a series of Use Cases (see Appendix A) and storyboards (see figures 4.2, 4.3, 4.5-4.9 and Appendix B for examples of these storyboards).

Use Case scenarios [Cockburn 2001] were developed to investigate the basic system feasibility and requirements. To this end, scenarios for current usage and a general computerised system were developed. Two further Use Cases scenarios were developed to compare the level of guidance to be offered to users. Storyboards of interface designs for the conceptualised system were developed. These focused on the approach described in 4.2-4.4. Some alternative storyboard elements were designed and these are described in the appropriate evaluation sections later in this chapter.

In an extension of the qualitative research detailed in chapter 2, four RBGE taxonomists provided feedback using storyboarded walkthroughs and further interviews. Peer review by three information scientists (from Napier University's School of Computing) of the storyboard walkthroughs and Use Cases was also utilised for analysis. The Use Case based process of usage and initial rough storyboards were used to gain initial feedback and based on this, the process of usage was refined for the final developed storyboard walkthrough evaluations.

The approaches and interface paradigms that were investigated are discussed in this chapter. First the proposed approach and its constituent interfaces are introduced. Then the process of usage and indicate changes following initial evaluations that result in the refined process represented in the final storyboard walkthroughs are discussed. Next elements of the presentation are discussed and finally general conclusions are drawn. The lessons from this initial concept work inform the development of an interactive tool (see chapter 5).

## 4.2 System Concept

### 4.2.1 Introducing the System

The basic system concept under investigation is one in which a taxonomist specifies the descriptive specimen data to be collected, in terms of a structured data model and defined terminology. This generates a structured proforma, which is used to control the data entry of specimen descriptions in a consistent manner for a given taxonomic project.

The high level task of capturing specimen description data can be broken down into two main user tasks:

- Specifying what description data is to be collected for the taxonomic project. (see Use Case Scenario 2, Level 2: ‘create proforma’ in Appendix A).
- Entering the instance description data for each specimen in the taxonomic project. (see Use Case Scenario 2, Level 2: ‘scoring taxonomic description for a specimen’ in Appendix A).

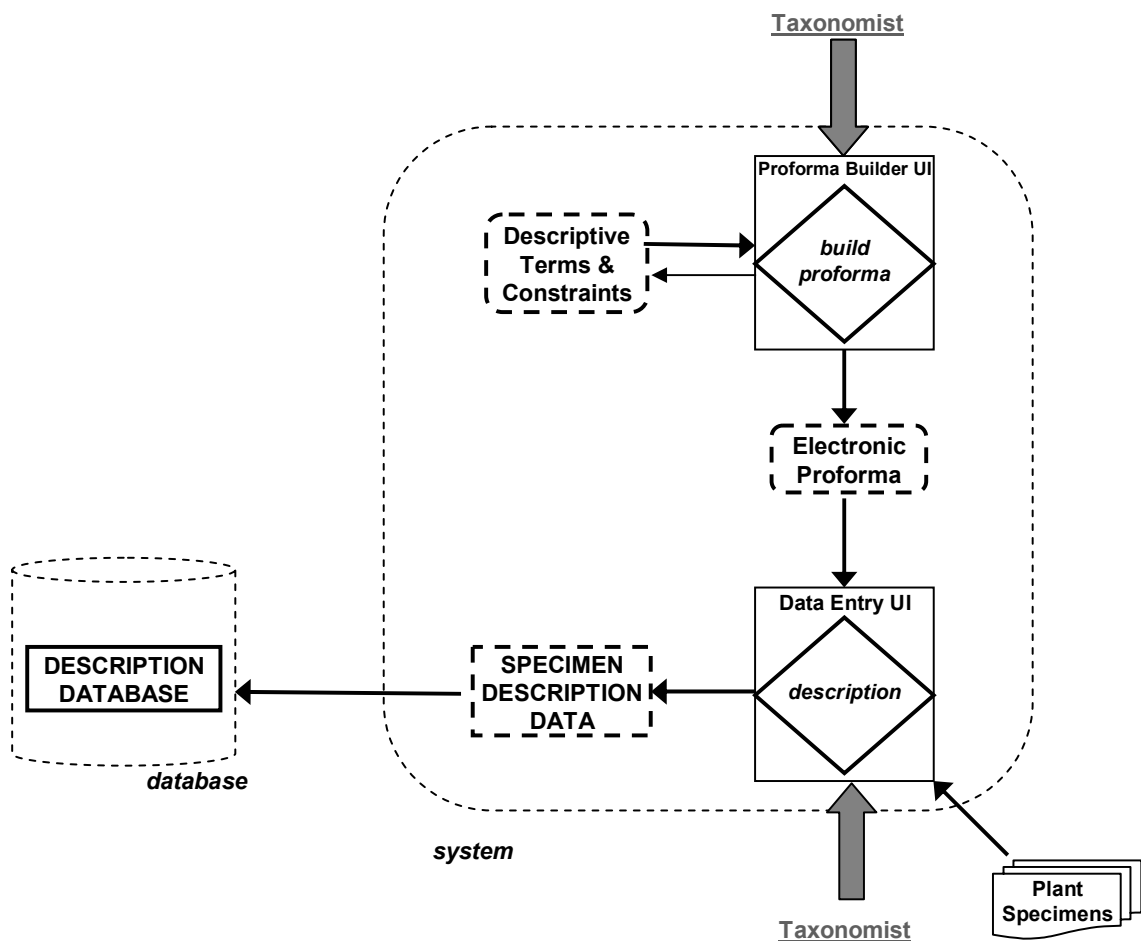
As the process of the detailed taxonomic sort is an iterative process, the user also requires the ability to return to the proforma creation stage to add extra data requirements to the proforma as needed.

Altering the descriptive data specified in the proforma when data has only been collected on a proportion of the specimens incurs a potential weakness in the quality of data recorded, as for example data entry decisions could be made on different options for different specimens. However removing this possibility altogether is not possible within the stricture of taxonomic working practice. Taxonomists confirmed that they could not necessarily know all *character* concepts that would prove to be of interest during a project before the process of collecting descriptive *character* data had begun. As time pressures on taxonomists work was intense there was certainly no time to start the whole process of collecting description data again when *character* concepts to be recorded were added or extended. The approach thus must allow for at least a limited degree of iterative working between the two main user tasks.

## Chapter 4 - Capturing Description Data for Taxonomic Projects

Each main user task has its own system-generated interface upon which the user operates (see Proforma Builder UI and Data Entry UI in figure 4.1). Linking the two interfaces in an integrated system will support altering the proforma during a project.

Figure 4.1 illustrates this two-interface system for addressing taxonomists' needs as initially conceptualised. Two UIs are shown in the figure, one for building an electronic proforma and one for data entry based on that proforma. The system generates these interfaces as required to display the details of defined descriptive terms and constraints on how to combine those terms. Presentation and dialogs were based on known user tasks and an implicit mapping from the data model.



**Figure 4.1: System to capture specimen description data in taxonomy. Two user interfaces are represented: the Proforma Builder interface and the Data Entry interface.**

## Chapter 4 - Capturing Description Data for Taxonomic Projects

The decision was made to keep the system independent of the description database to make it platform independent. In addition to general adaptability this decision would support the possible future use of the system on more mobile devices to capture data in the field. This would also avoid ties to a server, which would be advantageous as networks may not be available in some locations where taxonomists work with specimens such as herbariums. It was decided that the Prometheus II database [Paterson 2004] would be suitable to be used for this purpose during development and it was assumed specimen data could be exported in that format. This database was being developed to hold taxonomic descriptive data in a suitably rigorous format.

### 4.2.2 Description Data

Description data must be applied and utilised by both users and system to describe specimens in a controlled fashion.

To address the needs of ensuring clarity and comparability of the collected description data, it was decided to use only descriptive terms with clear definitions and controlling relationships between them. Taxonomic description data consists of *characters* of interest, which as seen in chapter 2, can each be broken down into a physical plant structure, an aspect or property thereof and the actual state or value. Using this breakdown, an electronic proforma thus consists basically of the structure terms in which the user is interested, the aspects or properties of those structures they wish to comment upon and the possible values or states that an individual specimen could possess. A specimen description instance differs from a proforma primarily in that it contains the actual value/state that applies for that specimen from the domain of possible values/states.

The terms from which the proforma is built and the constraints on how these terms are used are integral to the system (see figure 4.1 'Descriptive Terms & Constraints'). The defined terms were organised as a glossary of defined descriptive structure, property and state terms to which the taxonomist could add (within certain limits) where necessary. The provided interface would control how these defined terms could be combined through a series of in-built rules. It was conceived that these rules would be loosely based on the Prometheus II data format rules [Paterson 2004]. The Prometheus II data format being developed in parallel with this project aimed to develop a database

## Chapter 4 - Capturing Description Data for Taxonomic Projects

model for taxonomic descriptive data. Unlike other available electronic description formats, it was intended to support comparability and was designed for use with a defined terminology. The details of the Prometheus II data model used during this stage of research can be found in Appendix C.

### 4.2.3 Presenting the interfaces

Figure 4.2 illustrates an example of the envisioned proforma builder UI. The panel on the left side is an overview of the current state of the proforma, representing the plant structures with the specified characters. On the right side is a definition explorer panel, where defined terms from the glossary are displayed, with filter, grouping and order buttons to control the view. A dialog pane for adding a character can be seen in the centre of the screen. The panels are co-ordinated with linking techniques.

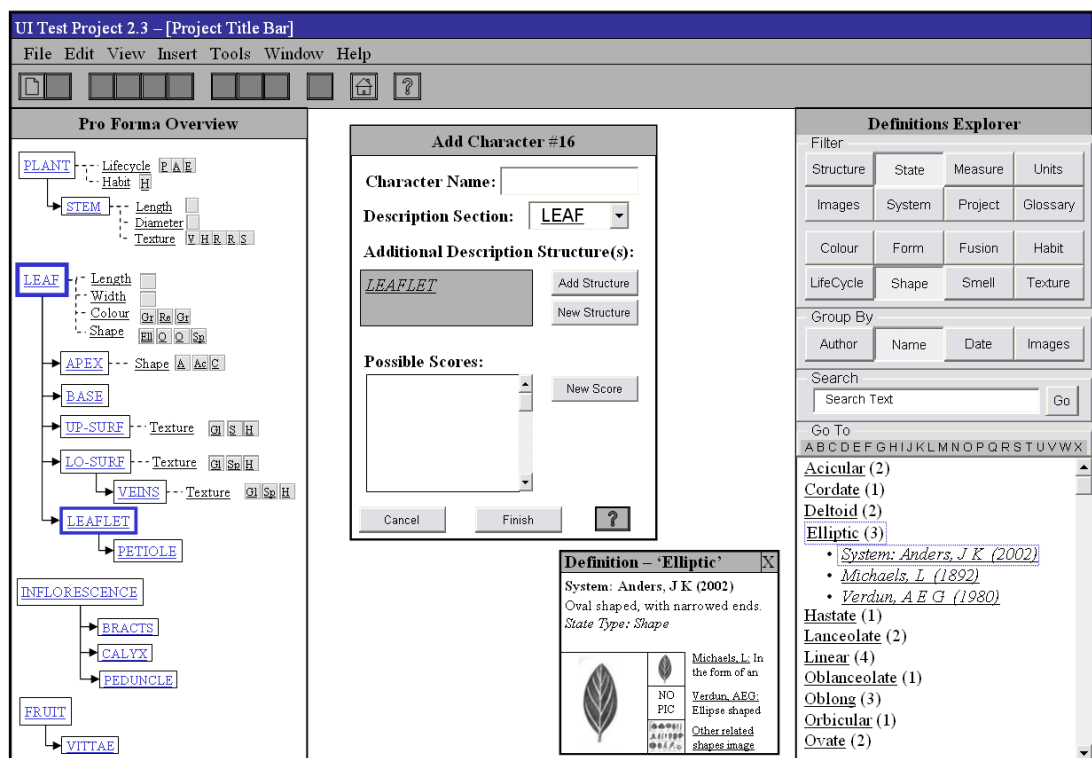


Figure 4.2. Example of Proforma Builder interface storyboard.

Figure 4.3 illustrates an example of the data entry interface. The overview of the proforma is again seen on the left side of the screen, with a character presented for data entry on the right. Variations on interface presentation were investigated and are discussed at 4.5.

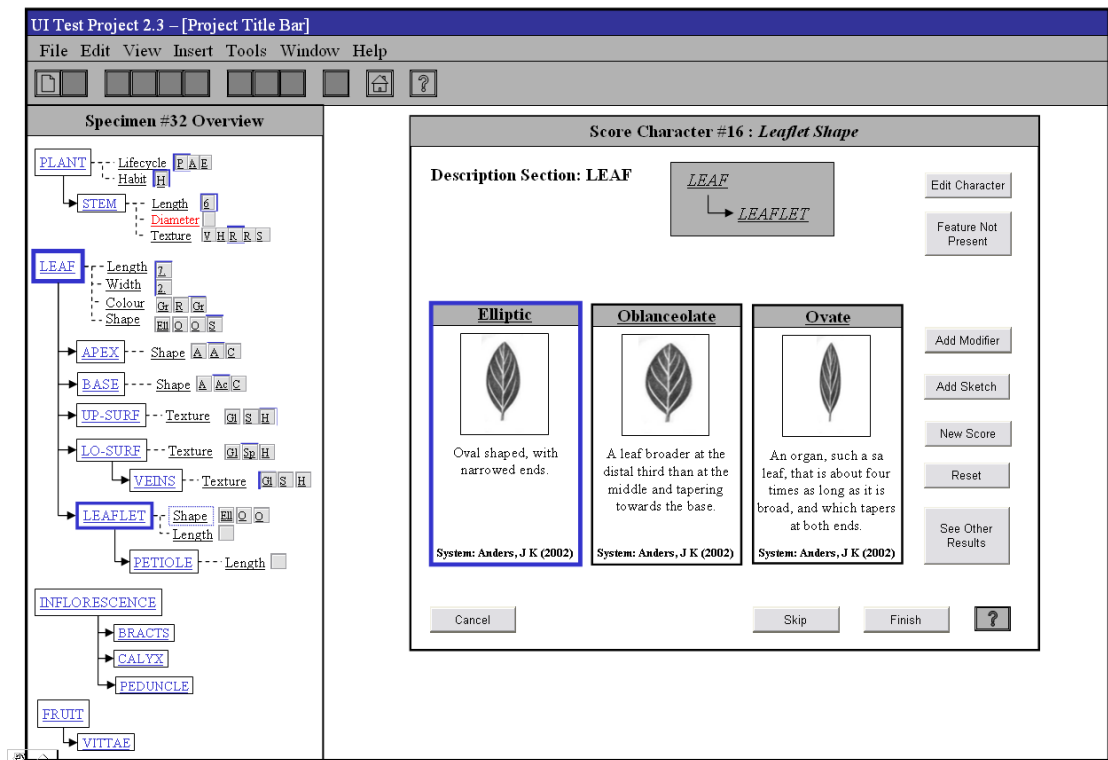


Figure 4.3 Example of data entry interface storyboard.

### 4.3 Proforma Building

Specifying the description data to be collected for each specimen in a project is one of the main user tasks. The *character* concepts of interest that a taxonomist wishes to capture data about vary from project to project. These *character* concepts of interest are the data requirements that must be specified in order to collect consistent data for each specimen in the study. The specification applies to all specimens in the project.

The specification task is different than the cognitive process of initially conceiving and discovering the *character* concepts that are relevant for a project. Deciding what taxonomic *character* concepts a user wishes to utilise within their project, uses the taxonomist's expertise, knowledge and reviews of appropriate literature. It is not intended to replace this aspect of conceiving initial *character* concepts with the specification task, although it may provide an extra source of support for it. The specification task at its narrowest includes placing the initial concepts into terms of the adopted data model. *Character* concepts as described in chapter two are often loosely applied in current practice and require to be specified consistently for our purposes. Additionally it should be noted that the initial cognitive concepts, being general

## Chapter 4 - Capturing Description Data for Taxonomic Projects

amorphous concepts, are often difficult for taxonomists to articulate even in the loose sense of *character* concepts,. It is sometimes only when they think of placing them in a proforma, that the initial general concepts evolve into the individual, recognisable *character* concepts as discussed in section 2.4.1.

### 4.3.1 Process of usage

Figure 4.4 shows a breakdown of the specification task. This involves repeatedly adding new *character* data requirements. The primary sub-tasks include:

- Determine the structure hierarchy and identify the relevant structure(s) for a *character* concept.
- Determine the type of *character*. Characters were divided into quantitative, qualitative and relative types.
- Specify the property and state.
- Explore definition space. Users required to find appropriate defined terms to use in specifying characters.
- Explore description space. Users required to explore the existing specified data and existing descriptions to find useful elements to re-use in creating further '*character*' data requirements.

This specification task and the implications of the nature of the data used to perform operations upon are discussed in more detail below.

## Chapter 4 - Capturing Description Data for Taxonomic Projects

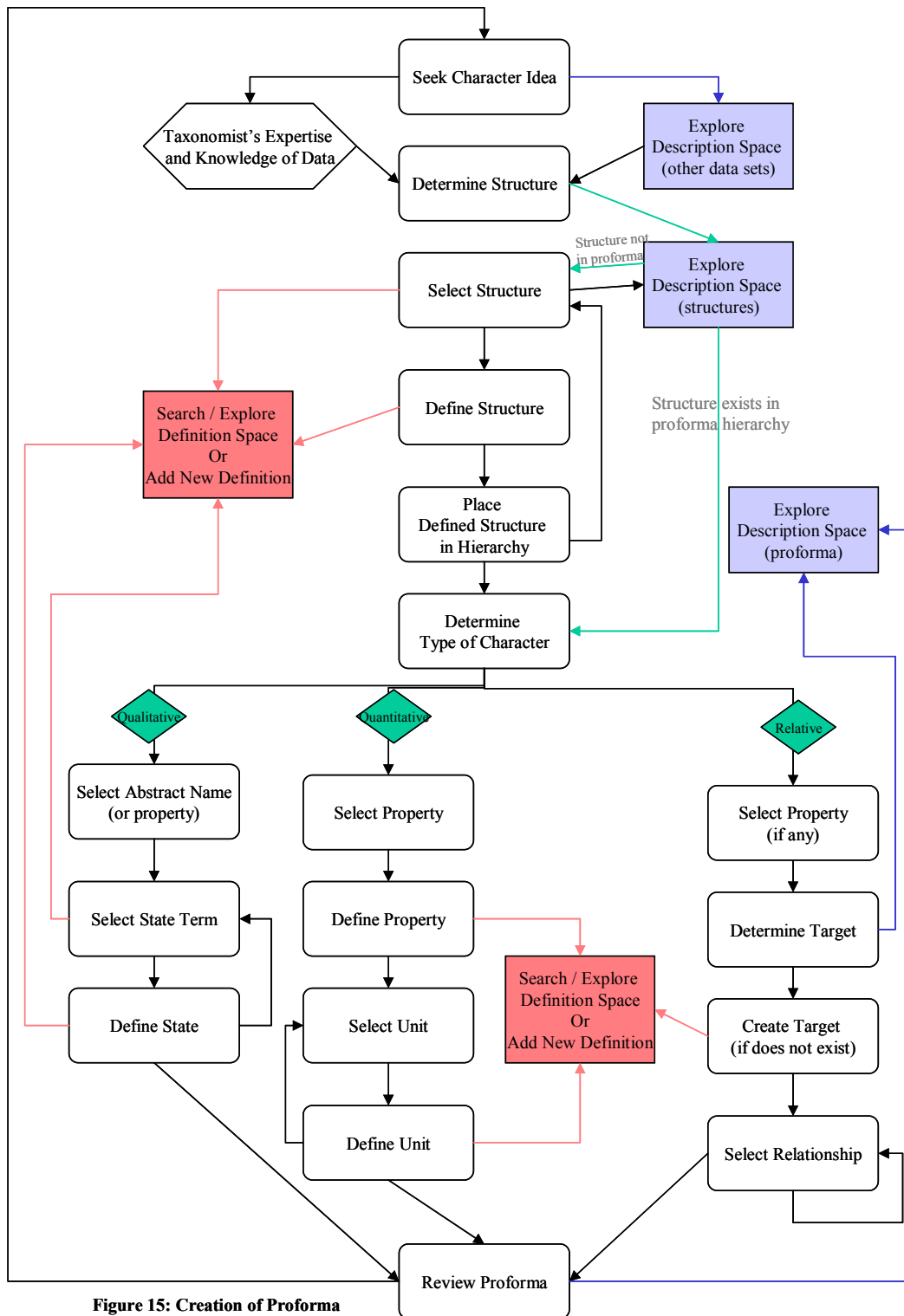


Figure 15: Creation of Proforma

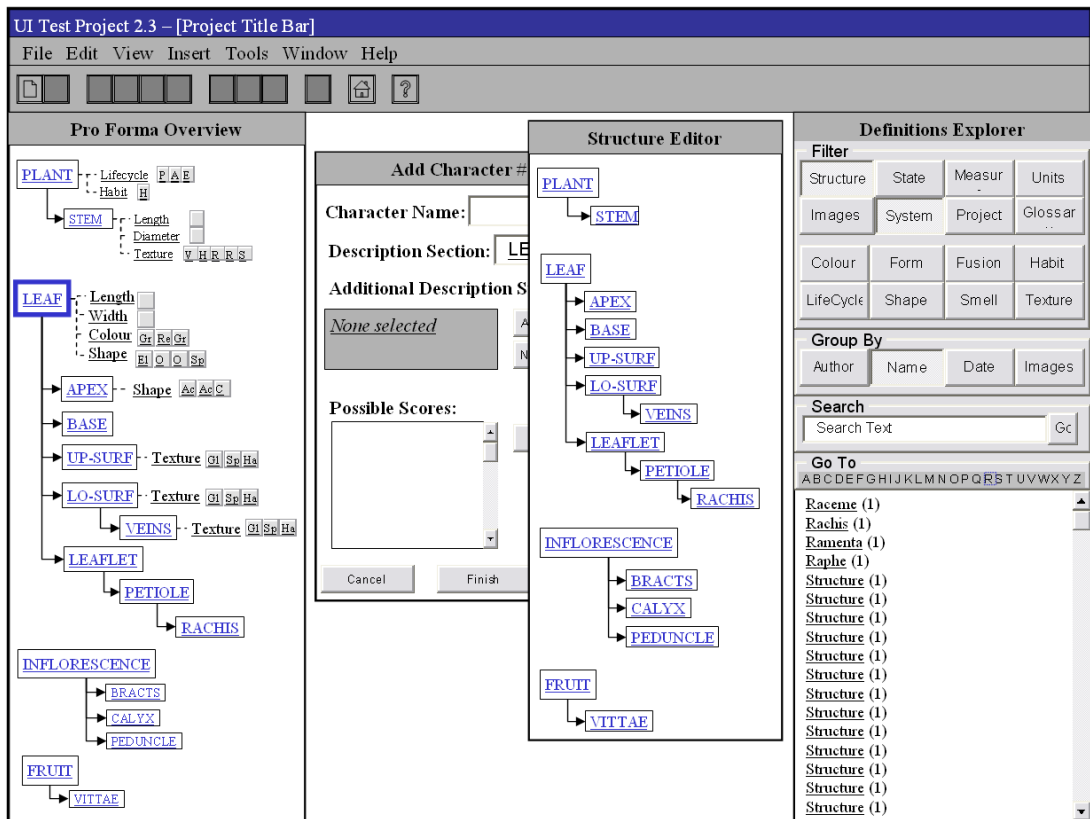
**Figure 4.4: Creation of Electronic Proforma. *Character* concepts are successively added into the system's understanding of the data requirements for a project by the taxonomist.**

To specify a *character* once the taxonomist has a descriptive concept in mind, the taxonomist must first describe what plant structure is being referenced. In the Prometheus II data model, a plant structure is described using either single defined



## Chapter 4 - Capturing Description Data for Taxonomic Projects

structure terms, or multiple defined structure terms linked by part-of relationships, which creates in the plant description, hierarchies of structures. The system adopts a similar view, which allows flexibility to identify any plant part and is meaningful in domain terms to taxonomists. The taxonomist thus determines a structure for a *character*, by either creating a new term instance or utilising part or all of an existing structure hierarchy, already created within the proforma. Creating a new instance of a defined structure involves selecting a structure term, defining that term if necessary, and placing it within the simple structure hierarchy. This is repeated if necessary until the structure is sufficiently described. In figure 4.5, a structure hierarchy example can be seen in the ‘Structure Editor’ pane.



**Figure 4.5: Creating an Electronic Proforma Prototype. Example from initial storyboarding.**

Once the structure of the *character* has been determined, the aspect or property of that structure which is being described must be determined. This aspect is the abstract quality of the structure being measured, which is chosen from the fixed list of system properties (e.g. shape, colour, length). The final step in describing a *character* for the proforma, is determining the domain of possible scores which the *character* can take

## Chapter 4 - Capturing Description Data for Taxonomic Projects

(e.g. elliptic, ovate). For quantitative *characters*, any number can be accepted, consequently a domain need not be determined.

Selection of terms involves exploring description space or definition space. Using description space, selection could be achieved by finding existing terms in the proforma and using them, along with their definitions. The taxonomist may wish to either re-use a structure term as is, in the same place in the structure hierarchy (as explained above). Alternately, they may wish to copy the structure, creating a second instance of it in a different structure hierarchy context. Alternately using definition space, selection could be achieved by searching through the list of defined terms held within the system's domain model. The taxonomist can then select and utilise terms of interest. Various filters and search aids are provided by the interface, based upon relationships between terms and other terms or system understood macro terms (such as 'Defined Term' *Leaf* is a kind of 'Structure'). Aids for searching definition space are important due to its potentially very large size. Figure 4.2 shows an example of the locating of appropriate 'State' terms for a *character*, using filters to restrict the viewed terms to 'state' terms and 'shapes'. The resulting terms are grouped by their name, with 3 specific defined terms being displayed for the selected name 'Elliptic'.

### 4.3.2 Refinements to the process of usage during storyboard development

Based on initial investigations, this process was refined before final storyboard walkthroughs were done. The need for users to indicate the property was removed, with the system inferring the property from the possible scores. Each state would have to include the data on which system property it applied to for this inference to be made. The system would also have to enforce that all scores referred to the same property. Defined measurements based on system quantitative properties (e.g. length, width) would also be represented in the glossary and one could be added as a possible score to a character, with appropriate inferences made.

An abridged example is given below of adding a qualitative character to the proforma, based upon the storyboarded walkthroughs (e.g. figs. 4.3, 4.4., further storyboards in Appendix A). The user decides what part of the plant is being described in this character. Then the user explores the pro forma overview to determine if the required structure(s) is already present and selects the elements '*Leaf - Leaflet*'. Next they explore

## Chapter 4 - Capturing Description Data for Taxonomic Projects

definition space for relevant possible defined states, selecting defined terms for 'elliptic', 'oblong' and 'oblanceolate'. The system infers the property 'shape' from the first selected state and generates a default name for the character of 'Leaflet Shape'. The interface now looks like figure 4.6. Lastly the user reviews the character and indicates the character is complete.

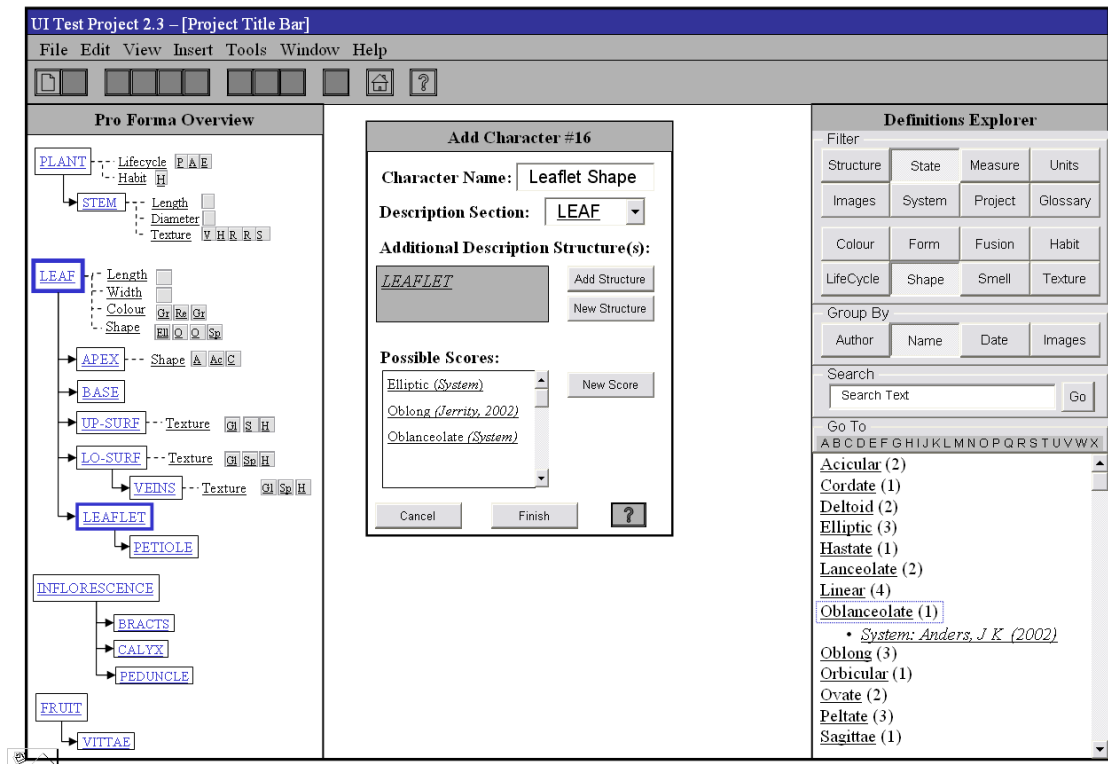


Figure 4.6: Specifying a character to add to a proforma. Example from storyboarding.

### 4.4 Entering Instance Description Data

In the other main user task, the user enters the data for each specimen in their project. The physical aspect of this task dictates that generally one specimen is fully recorded before moving on to the next one, although as knowledge is accrued during the project, additional data requirements may be identified, causing previously recorded specimens to be revisited to enter additional data.

The sub-tasks for the data entry task primarily included the repeated sub-task of entering/selecting data for each specified *character*. Data entry revolves around selection from presented options and numerical entry. As the aim of the system involved

## Chapter 4 - Capturing Description Data for Taxonomic Projects

the use of defined terms to improve clarity and data quality, text entry was avoided. Secondary tasks such as exploring definitions, and checking previously entered description data are supported on demand.

Scoring views used a *character* by *character* within one specimen process, with one *character* presented per window. This view emphasises the use of graphics to allow rapid informed choice of scores. An example can be seen in figure 4.3. An alternate data entry view was investigated during the final storyboard evaluation. This used a multi-specimen spreadsheet metaphor. This presented a familiar spreadsheet metaphor with one specimen per row and one character per column. This is discussed further in 4.5.1.

### 4.5 Evaluation

A number of issues relating to the presentation of the system were investigated and are discussed below, where some conclusions regarding their value for supporting taxonomists' working practice could be reached. Additional evidence on other presentation issues was also garnered during the storyboarding process which is discussed under the relevant sections in later chapters.

#### 4.5.1 Presenting choices for scoring

Two basic types of data entry could be provided depending on the type of *character*: numerical data entry; and selection from a domain of defined term options. Selection from options required relevant details of the options to be provided. The storyboard tests investigated providing full details of each option (as in figure 4.3). One concern about adopting this presentation is that it could potentially lead to a screen space problem if the number of options was large.

The tests showed that the multi-media aspect of the definition was thought by users to be important and useful at this data entry stage. The reinforcing of structural context by both the focus of the overview and by explicit presentation next to the data entry options proved worthwhile as different users noticed the structural context in different places. The ability to add extra details such as notes was considered generally valuable, but little detail was evaluated concerning this aspect.

## Chapter 4 - Capturing Description Data for Taxonomic Projects

The evaluated approach presented one character per window, which users felt was generally a reasonable match for their real world observation of specimens. However users did suggest that as they became familiar with a proforma, they might wish to speed up data entry avoiding lots of clicking to get to the next character. One user suggestion was to utilise the overview as a data entry screen (as all options were represented upon it). Negative factors to that suggestion were that multimedia aspects would not be presented and there was no room to indicate any details. An approach which grouped more than one character per window should be considered. The suggestion that the structure was used as a grouping mechanism was considered to be a very good match for working practice by users.

The main single character in a single specimen view was compared with the spreadsheet scoring view (4.4). The single specimen view was considered most appropriate for data entry as it matched working practice of focusing on one real world specimen at a time. The spreadsheet type of scoring view was considered mostly useful for analysing the project's data for classification purposes, however this aspect of the taxonomic process lies outside the area identified for capturing description data, hence the decision was taken to concentrate upon the single specimen scoring view.

The ability to reference other previously scored specimens was seen as being useful in maintaining consistency of scoring and avoiding what taxonomists' referred to as 'scoring drift'. This drift involves the situation where the user's concept of an actual *character* score gradually and unconsciously changes through time, so that the values for *characters* of a specimen scored at the beginning of a project are no longer comparable with those scored for one at the end. Users believed that use of defined terms and repeated multimedia reference during scoring would also help alleviate this issue.

### 4.5.2 Overview Visualisation

Having a visual overview representation of the entire proforma and where appropriate the specimen currently being scored, was considered important to help support the users mental picture of the data. Users expressed support for this concept. To maintain consistency between the data specification and data entry interfaces, tying both together, this overview should ideally be common to both. By using the same visualisation

#### Chapter 4 - Capturing Description Data for Taxonomic Projects

paradigm for the overview in both interfaces, the learning curve for new users would be reduced. Users also felt a closer familiarity with the data at data entry time when they could see the data requirements they had specified in a familiar form.

Three basic paradigms were considered based primarily on the research in chapter 3.

A simple flat listing of *characters* was one simple paradigm that was considered due to its simplicity (see section 3.4.3.1). Such a linear visualisation however required excessive space to effectively display the plant structure context for each *character*. Additionally the ordered nature of such a list was of less value for matching the user's mental picture of the data than a hierarchical visualisation. Although a linear structure can be imposed on the structures of the description, it is an arbitrary decision based on the loose concept of acropetally (from the bottom of the plant up) and from the outside in, and is thus subject to individual interpretation. The other elements of the description are non-linear. This view was not developed.

Hierarchical visualisations matched the user's mental picture of the data. This was evident from early interviews and from later storyboards. Variations on a file tree analogy proved simple for users to grasp. Users were mostly familiar with file trees from using them in their PCs. During storyboard tests they quickly grasped how they worked and were enthusiastic about their ability to navigate such trees to help control their use of the data specification and data entry interfaces.

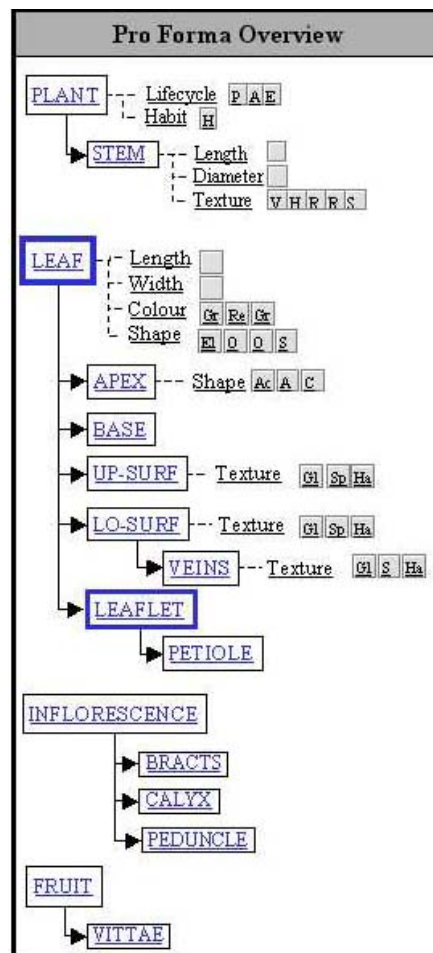
An alternative hierarchical view using a hyperbolic disc [Lamping 1996] was also shown to users. This had been considered in chapter 3 (3.4.3.3, figure 3.9). They found this view to be more confusing and less intuitive. Initial difficulties might be overcome, especially as only a static view of description data for this interactive visualisation was evaluated with users. However, the hyperbolic disc also failed to as closely match the user's mental picture of the data as the file tree. The tree metaphor, narrow at the top and getting wider along branches as the user travelled downwards made intuitive sense to users who thought of their specimens as plants but they did not see this in the hyperbolic disc's centre outward in all directions view.

Some variation of the file tree was therefore decided to be the best analogy to use for the overview and central navigation control for the interfaces.

## Chapter 4 - Capturing Description Data for Taxonomic Projects

The same basic overview could be used for both interface tasks, with the actual scores for a specimen also being represented on the overview during the data entry task. Focus and filtering visualisation techniques such as expansion and contraction of the file tree and/or some form of focus+context fisheye view [Furnas 1986] would ensure the relevant portions of the data could be sufficiently focussed upon on the screen in the case of large data sets.

Linking techniques (see section 3.4.2) involving the co-ordinated interaction of the overview with the other elements of the interfaces gives users a simple navigation tool, and provides a sense of integration of the whole application. Additionally, text and multi-media definition data is required as a detail on demand when working on either of the two main tasks, which can be made available from the overview.



**Figure 4.7: File tree overview. The Leaflet of the Leaf is focussed on in this example.**

### 4.5.3 Presenting Definitions with Multimedia Aspects

Since encouraging the use of defined terms is seen as a major contributor to improving the quality of data collected, it is important to present definitions in a useful and easily accessible manner. Pop-up definition boxes (see figure 4.8) were used to allow multiple definitions to be displayed contemporaneously, without dedicating screen space to them.

Storyboarding and interviews suggested users would find both text and multi-media definition aspects complementary to each other. Having both aspects in one definition box would thus be of use. Users found the provision of definitions useful at all stages of both proforma building and data entry. Placing links to alternative definitions of the same term and related terms (e.g. same author) within a definition box was found to be potentially useful when searching for terms to use, but not at other times.

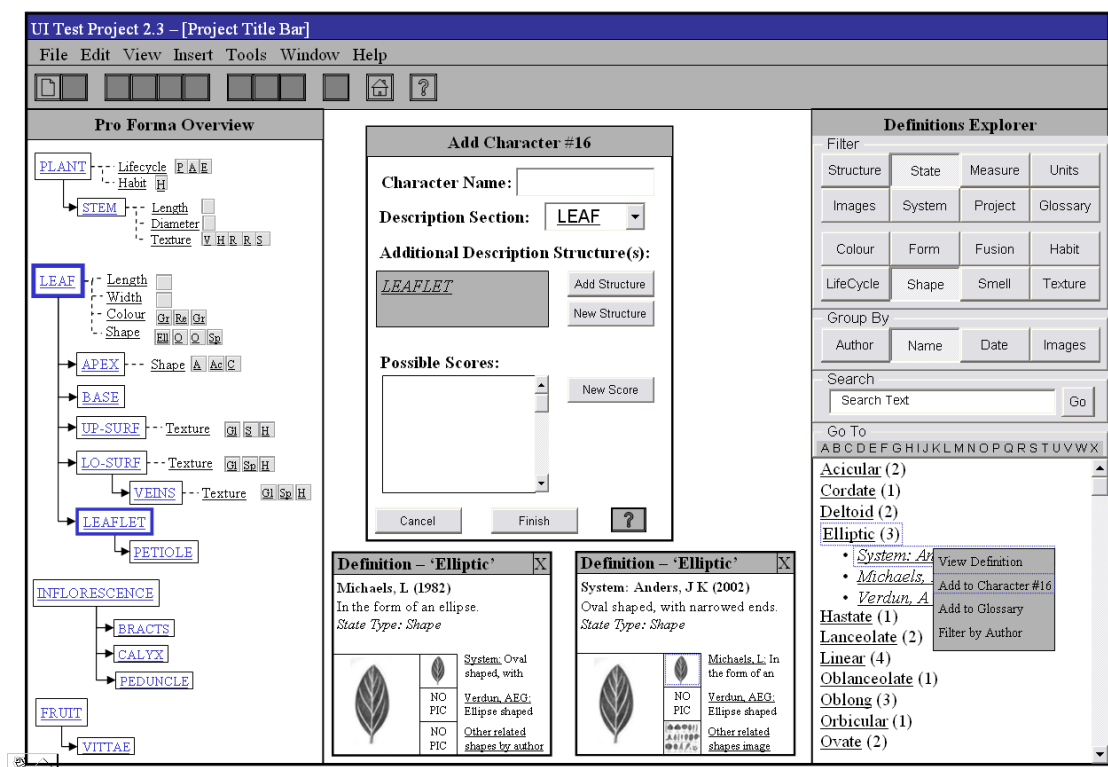


Figure 4.8: Storyboard proforma builder interface with pop-up definition box

### 4.5.4 Level of Guidance

The main investigated approach allowed users to add structure and character details to their proforma in an order of their own choosing, using a free-form multi-pane



## Chapter 4 - Capturing Description Data for Taxonomic Projects

constructor approach as described above. An alternative process, using strong guidance and a 'wizard' metaphor, was developed in the Use Cases and described to users for evaluation along with the main storyboarded approach.

The user in this process is guided by a structured set of system generated step-by-step dialogs to add characters to the proforma, working through adding a structure, then adding all the characters for those structures, then adding another structure, etc. It was intended to offer only common choices of defined terms to users at the basic level with extra options for advanced users. For data entry users would be presented with data entry questions one by one in default order till the whole proforma had been completed for a specimen. Further details of this approach can be seen in the Use Case Scenario 4 in Appendix A.

The wizard metaphor approach was not viewed as favorably as the multi-pane constructor approach, as users felt the constraints would interfere with their freedom to move rapidly from one area of the process to another – something which was seen as a major strength of the multi-pane constructor approach. A limited implementation of a wizard idea, in conjunction with the main interface approach, was however felt to be of possible potential value for novice users. From questioning in follow-up interviews, this concern with novice users appeared to stem from continuing concern over how easily users, not familiar with the data model, might find the general structured data approach. Streamlining the process and improving the inferences the system could make from the underlying data provided other possible means of tackling the issue of users unfamiliar with the underlying data format. Subsequently this heavy guidance approach was not pursued further.

### 4.5.5 Glossary Based Problems

Storyboard walkthrough testing with taxonomists showed that there was an issue with the number of potential relationships between different terms that could be specified. Whilst allowing freedom of expression, the potential number of relationships made it equally difficult to specify common utility *character* concepts as it was to determine specialised or unusual *characters*. Further support for automatically linking common terms was thought to be of value (for example by automatically linking states with the structures and properties they were commonly used with).

## Chapter 4 - Capturing Description Data for Taxonomic Projects

Users displayed a particularly strong resistance to the concept of determining a hierarchy of structures without added guidance. This task was seen as an extra artificial and tedious task. In their present non-computer-aided practice, the details of structure hierarchies are not recorded. In most cases users interpret the structural content of *character* descriptions based on inherent domain knowledge of botanical biology that they already possess. This interpretation is not however without the potential for error or confusion when interpreting another taxonomist's work. To encourage the use of a more rigorous descriptive practice, the task of determining the context of use of the structure terms required to be simpler for users than building a structure hierarchy from scratch.

An additional problem during evaluation with the specifying of structure hierarchies was that users became distracted by the possibility of placing structures within the hierarchy in places where they could not exist in real life physical plant biology. Potential relationships between some structures, attributes and states could, like some structure hierarchy relationships, also be meaningless in real life terms. Presenting these meaningless possibilities to users was distracting as well as making it difficult to identify terms of interest to particular contexts.

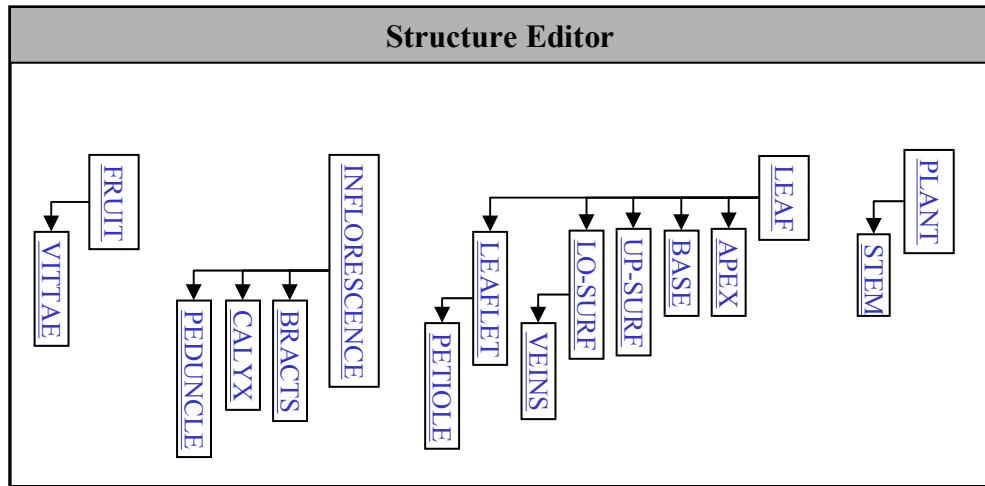
Two different presentation techniques were used to present the hierarchy for creating and editing. Utilising the overview to show the created hierarchy was a basic method, complicated by the use of the overview to also show attribute data of the structures. The use of a separate pane (see figure 4.5) with just the structure hierarchy simplified this issue, clarifying the view of the structure hierarchy but still failed to overcome the basic resistance to the task. Changing the orientation of the editor as in figure 4.9 made no difference in storyboard tests.

Varying the level of guidance in the process from free-form construction to a wizard-type guided process (see 4.5.4) did not change the basic resistance to the hierarchy creation task, as it still needed done.

A solution involving the fundamental nature of the data and rules used for the structure hierarchy would need to be used to overcome these issues rather than a purely presentation based solution. Having a display of the structure hierarchy without property

## Chapter 4 - Capturing Description Data for Taxonomic Projects

and state data is however likely to ease any editing of the structure hierarchy by clarifying the view.



**Figure 4.9: Alternative orientation structure editor**

In answering these problems there is however no universal plant structure hierarchy that could be utilised to avoid hierarchy creation. Any hierarchy of all plant structure options was too extensive to be practical. There were also no domain semantic restrictions on which structures could be combined. Taxonomists felt any such restrictions would be too constraining to apply universally to all plants.

One solution to the glossary problems that was considered involved using proforma templates. These templates would be created for small groups of related types of plants, which could then be customised by users, using the standard approach. The templates would be built using the normal rules and would include basic structure hierarchy and common *characters* for the plant group. This solution however did not address the problem that it was possible to create structure hierarchies which could not exist in the real world. The template proformas were simply a baseline, which could be freely added to. Templates also might still involve a lot of user defining of hierarchies if the group of plants covered by the template was large. In this case the hierarchy would have minimal detail, as the larger the group of plants, the smaller the structure hierarchy they have in common. On the other hand if the groups covered by the template were small enough to allow a more detailed common hierarchy, many different templates would be required due to the number of groups. Irrespective of the size of the group, the templates would be based on one taxonomic opinion of what structures and related *characters* would be

## **Chapter 4 - Capturing Description Data for Taxonomic Projects**

of interest for a specimen. As the user might not share this opinion, each user would potentially have to create their own set of templates. This potential approach was discarded and not developed beyond the theoretical stage in favour of another more promising approach based on using an ontology for a limited group of plants as a domain model.

### **4.6 Conclusions**

The hypothesised conceptual approach where a taxonomist specifies a structured proforma, which is used to control the data entry of specimen descriptions in a consistent manner for a given taxonomic project, was found to be feasible. The proposed two interface approach could support the use of a structured data model and defined terms to help address the data quality issues of taxonomic description data. Additionally, making the two interfaces part of one combined application allows some freedom of working order to move between the operations of both main user tasks as required to support working practice.

The basic approach was found to match well with the needs and expectations of users, with evaluation respondents very quickly grasping the basics of the system concept. However the level of effective support for users in creating the electronic proforma was limited by the simple glossary based description data and rules. Users felt they had to complete too many steps themselves in the process of usage, challenging the aim of completing tasks in a simple and timely manner. More relationships between terms based on domain semantics were required to support a streamlined process of usage. An approach utilising a simple ontology, to address this issue is discussed in the following chapters.

The following chapters discuss an interactive application based upon this approach. Some early conclusions about the issues regarding such a system have been drawn, such as the general nature of an overview visualisation and the value of multimedia definitions. Insight and evidence about other aspects of this system were made during this investigation which aid in drawing conclusions during the later interactive investigation. These aspects are discussed in the relevant sections of later chapters.

## Chapter 5

# Ontology-Based Generation of Data Entry Interfaces

### 5.1 Introduction

In chapter 4 we introduced an approach which allowed taxonomists to specify a structured proforma that was then used to control the data entry of specimen descriptions in a consistent manner for a given taxonomic project. A structured data model and defined terms were used to specify the proforma in order to address the issues of taxonomic description (see 2.4). However during storyboarded walkthrough evaluations, issues were found with the envisioned process due to the lack of user guidance for description building, especially regarding structure hierarchies. Refining the process of usage required more relationships to be included in the glossary, in order to avoid unwanted additional tasks that challenged the aim of working in a timely manner (2.6.1.2).

The Prometheus II project [Prometheus II 2005], which had developed the descriptive data model upon which the rules for the early approach were taken, was also looking at constraining the usage of defined terms to improve the comparability of their terminology. As developing an ontology of all plant descriptive terms was thought to be infeasible (2.4.4), it was decided to reduce the scope of an ontology to ensure sufficient common ground could be found. The scope had to be sufficiently small that an agreed vocabulary and composition hierarchy of structures could be built in a reasonable time, yet not so small that it would have too limited an application both in terms of user groups and in making comparisons for classification decisions. An ontology defining descriptive terms for the domain of flowering plants (restricted to the inclusion of macroscopic anatomical-morphological features found in traditional specimen descriptions) was selected as an appropriate compromise [Pullan 2005].

Although it was felt that agreement on the definition of structure terms could be found within the group of flowering plants, there existed a wide range of possible combinations for the composition of the structures. Rather than define a single

## Chapter 5 - Ontology-Based Generation of Data Entry Interfaces

composition hierarchy for flowering plants, it was necessary to define a ‘super plant’ composition hierarchy, with duplication of structures in different possible contexts. This hierarchy could never be found on any single real world plant, any particular plant would have a sub-set of this ontology compositional structure hierarchy.

It was hypothesized that having an ontology that included all possible relationships, which constrain how defined terms can be combined to form descriptions, based on domain semantics, could help solve the problems of insufficient support and unwanted additional tasks identified in chapter 4. In the previous approach, a structured proforma was specified to serve as a basis for automatically generating an interface that collected data on a consistent basis for a project. An equivalent process could be achieved by editing the description ontology to specify a project proforma. This editing would need to make a coherent plant compositional hierarchy from the ontology’s ‘super-plant’ and would be constrained to ensure no ontologically unsupported relationships were used. A data entry interface could then be automatically generated for this edited ontology.

This process fits well with the MB-UIDE techniques where the ontology could be used as a basis for a domain model that was specialized for projects. A project specific data entry interface could then be automatically generated. MB-UIDE approaches have historically attempted to generate more appropriate, effective interfaces by moving from a data model to a richer domain model basis or by improving the modeling of the task model (see 3.2.2.3). We have a fixed task, that of data entry, so no editing or soliciting of tasks would be necessary. Some type of presentation model could then map the edited domain model with these known tasks to an effective data entry interface that supported the collection of ontologically defined data.

Taking a MB-UIDE approach would have the advantages of clarifying the approach and supporting abstract modeling of the domain independent of issues of presentation. If a suitable means of deriving the domain model from the ontology can be derived, the MB-UIDE approach would also provide a framework where it could be envisioned that other descriptive ontologies could be utilised including one for other biological groups such as gymnosperms, fungi or for domains outside taxonomic description (chapter 8 discusses the application of the approach to another domain). The requirements of an underlying ontology thus need to be considered in evaluating the approach. It would also be useful to have the approach independent of the ontology, so that simple

## Chapter 5 - Ontology-Based Generation of Data Entry Interfaces

alterations in the developed ontology content would not require re-programming of the tool.

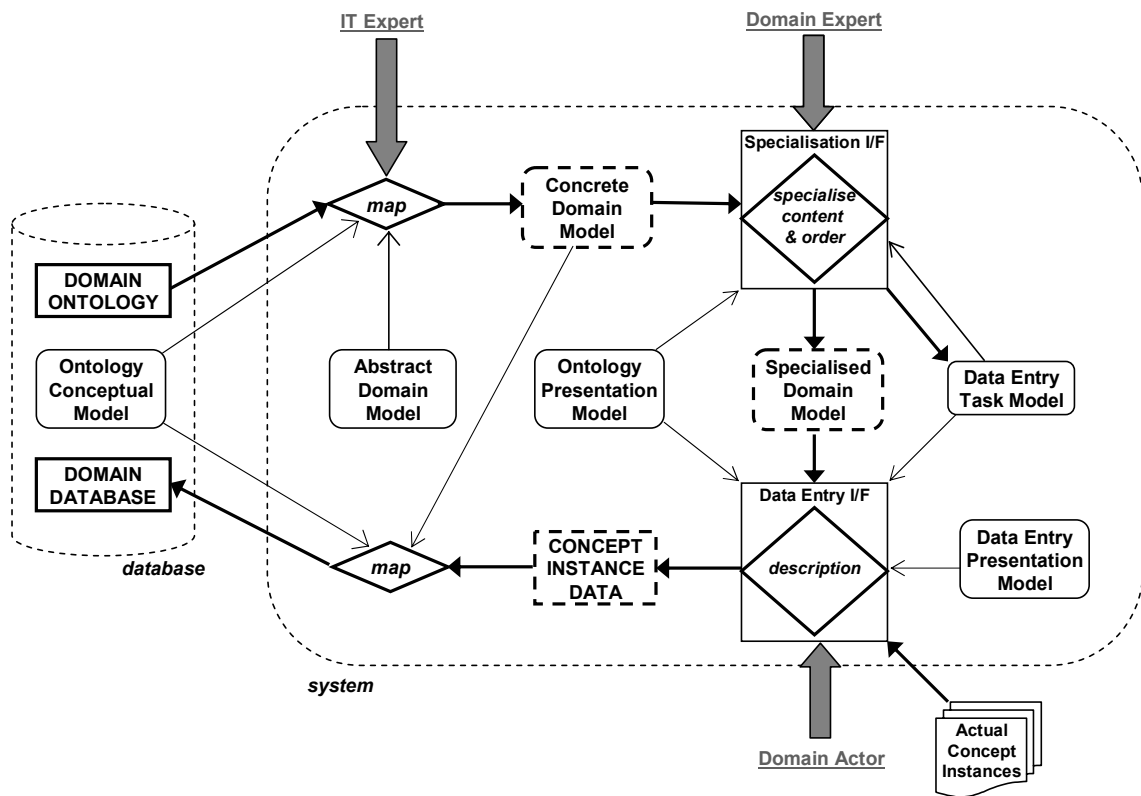
Despite calls for using the capabilities of model-based approaches to support end-user interface tailoring [Szekely 1996a], MB-UIDE approaches still support developers rather than end-users to specify models and tailor interfaces. Whilst taxonomists would not have to specify the model from scratch in our approach, the approach would need to support taxonomists to appropriately edit the domain models in non-modeling terms, taking advantage of the ontology to address concerns of insufficient support raised in the previous approach.

In comparison to chapter four, this approach uses domain ontologies to provide domain knowledge to the system. This chapter introduces the ontology-based approach with its known tasks first. Then we discuss the explicit domain model used to capture initial domain knowledge and specific project description needs. The utilised ontology and the process of how the underlying ontology is mapped to the system are examined. Presentation models used to generate the system interfaces are briefly introduced. This approach was developed iteratively through cycles of tool development and evaluation. The development methodology is introduced before turning in subsequent chapters to the details of the implementation and evaluation of the approach.

### 5.2 Approach and Tasks

The high level task of capturing description data is still broken down into two main user tasks as described in chapter 4. The task model is pre-determined to be the general task of data entry for a database, with the same essential breakdown into specifying data requirements and actual data entry. Each of these tasks utilises a user interface. In figure 5.1 these are labelled as the **Specialisation I/F** and the **Data Entry I/F**.

## Chapter 5 - Ontology-Based Generation of Data Entry Interfaces



**Figure 5.1. Ontology Driven Automated Generation of Data Entry Interfaces utilising a Model-Based Approach.**

The user may be the same individual or a different one for these two tasks. In the taxonomy field, both tasks would traditionally be completed by the same taxonomist, but for larger surveys a taxonomist may specify the proforma and have other biologists describe specimens using the proforma. The user for the specialisation task must be a domain expert, as this task requires a higher degree of domain knowledge. The data entry task user can be any user who has sufficient domain knowledge to interpret the actual concepts to be described during the data entry process.

For our system, the task model for the system is fixed (as data entry to a database) and thus is encapsulated within the system. The only user modifiable aspect of the task model is the default order in which data entry is presented to users. This is shown in figure 5.1 by the interaction of the user interfaces with the **Data Entry Task Model** (see chapter 6 for further details).

Generalising the approach outside of taxonomy, the basic system architecture remains valid in scenarios where there is the same basic task model of specifying the project



data entry requirements and entering descriptive data about concepts of interest based upon these requirements.

### 5.3 Ontology Based Domain Model

Domain models within the system control the application and use of description data by both users and the system to describe specimens in a controlled fashion. In chapter 4 the approach primarily utilised a terminology glossary for this purpose, but issues were found with the lack of user guidance for description building that could be provided by the system, leading to additional undesired user tasks. A more elegant solution considered was to expand the terminology glossary to an ontology by including additional domain specific relationship data. For example including all the possible structure to structure relationships in a plant would allow the system to present users with a hierarchy including all the possible **structure terms** in context.

An ontology based approach has advantages in generalising the approach, as it allows more details of domain specific relationships to be captured as part of the imported domain model and not held by in-built rules. Whilst the details of the domain specific terms and relationships vary for different domains, the basic format of taxonomic description data is fairly simple and can be easily generalised for describing other objects of interest. That basic format serves as the basis for the **abstract domain model** used by the system as shown in figure 5.2.

Based upon this abstract model, a series of domain models are utilised to represent domain knowledge within the system. In figure 5.1, we can see that an existing domain ontology is mapped to the **abstract domain model** and thereby transformed into a **concrete domain model**. A domain expert then specialises this **concrete domain model** to create a **specialised domain model** for a given project of work, which contains all the possible data options for entering data on the concepts of interest (equivalent to the structured proforma discussed in chapter 4).

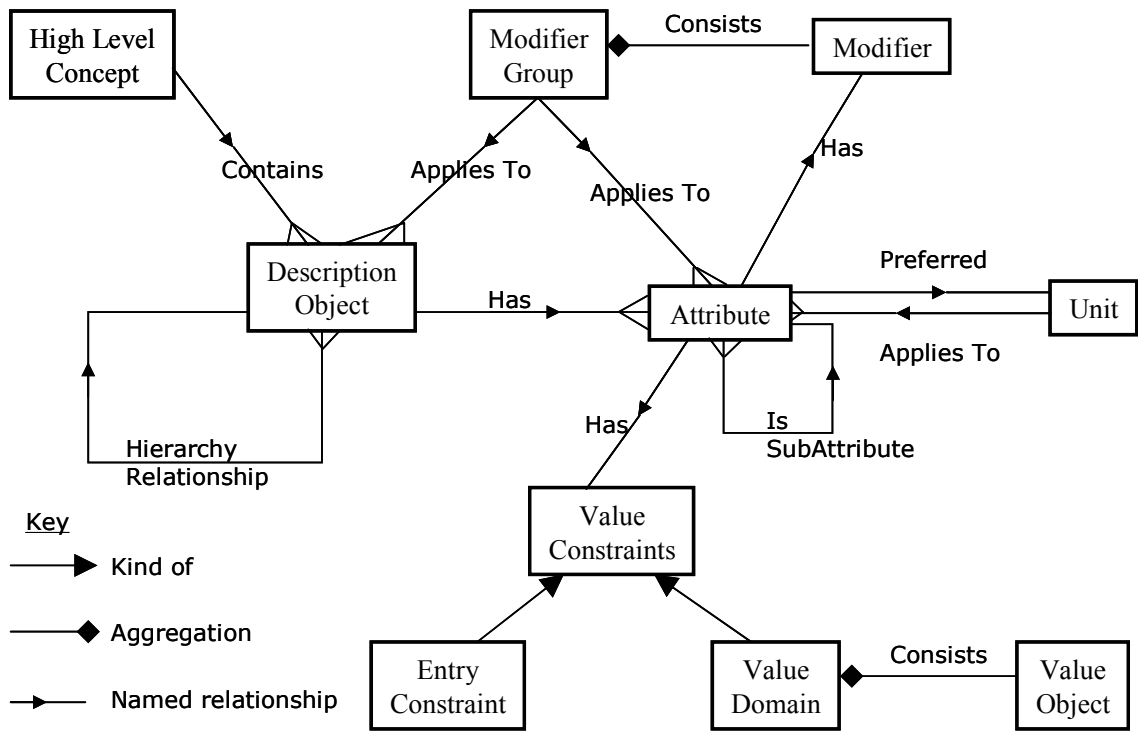
The various domain models are described further in the remainder of this section.

### 5.3.1 Abstract Domain Model

The **abstract domain model** represents the system's understanding of any domain ontology that can be imported to the system. Figure 5.2 shows the main descriptive elements of the **abstract domain model**.

The system is designed to capture data concerning any high level concept (e.g. specimen) that may be sub-divided into a hierarchy of defined constituent sub-concepts ('**description objects**') that are themselves described by instantiating **attributes** that they possess. Each **attribute** of a **description object** can be instantiated within the limits of its **value constraints**. These **value constraints** might restrict entered data (such as the data type or numerical range of entered data), or define selection from a limited set of **value objects**. A **value object** represents a defined concept that can be used to instantiate an **attribute**. Additional entities (**modifiers** and **units** of measurement) allow more detailed description of **attributes** and their value constraints. All the main descriptive elements (**description objects**, **attributes**, **value objects** and **units**) have definitions.

This data format could be relatively widely applicable, representing both physical and abstract concept domains (e.g. a control system process or academic department) where there are entities that users wish to describe.



**Figure 5.2: Abstract Domain Model: the conceptual model for controlling domain knowledge in the system. All the main constituent descriptive elements used to control the description of a high-level concept are shown. The hierarchy of description objects is formed using the Hierarchy Relationship.**

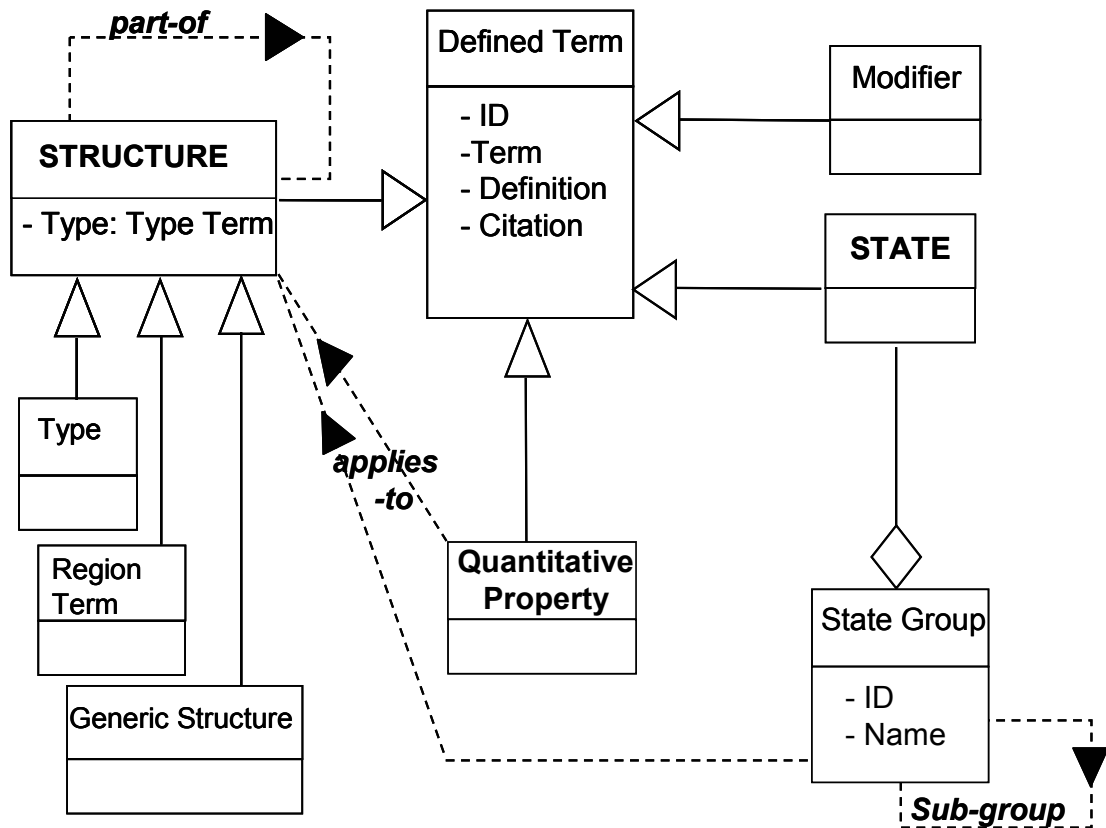
### 5.3.2 Domain Ontology

In order to instantiate the **abstract domain model** with actual domain knowledge, an appropriate domain ontology is mapped to it, to create a **concrete domain model** (see figure 5.1).

The approach assumes the existence of an appropriate domain ontology, which does not necessarily have to be created solely for this system. Ontology is a widely used term, with a variety of meanings [Guarino 1995]. The commonly quoted definition ‘*a specification of a conceptualisation*’ [Gruber 1993a] is generally appropriate for this usage. Specifically, the domain ontology for our approach is a semi-formal, constrained and structured form of natural description language, with defined terms and possible relationships between them. Even so defined, ontologies can contain many different objects and relationships with various semantics.

In the taxonomy domain, an ontology to define the possible descriptions of angiosperm specimens (flowering plants) is being used as a domain ontology. The ontology is

composed of 'Defined Terms' (terms with associated definitions and citations) and relationships between these terms.



UML based notation:

- diamond ended lines represent basic aggregation;
- unfilled arrows and solid lines represent type-of relations
- solid arrows and dotted lines represent other named relationships

**Figure 5.3 Major terms and relationships represented in the angiosperm domain ontology conceptual model.**

As shown in figure 5.3, there are three major subclasses of defined terms used to create descriptions of biological specimens: 'Structure terms' representing all the possible anatomical structures of a given specimen (e.g. *petal*, *stamen*); 'Quantitative Property terms', represent aspects of a 'structure' that might be described quantitatively (e.g. *length*). In descriptions 'quantitative properties' are scored by numerical values. 'State terms' represent the actual values for an observation of a given structure (e.g. *round*, *yellow*). 'State Group' relationships in the ontology capture permitted relationships between groups of 'States' and the set of 'Structures' that they may be used to describe. 'Is-part-of' forms the central organising relationship for the ontology, and allows representation of a compositional hierarchy of all the possible structural relationships

found on any given specimen (e.g. a blade is part of a leaf, or part of a leaflet, which itself is part of a leaf). Full details of this ontology can be found in [Paterson 2004].

### 5.3.3 Mapping to the Concrete Domain Model

The **concrete domain model** contains the system's understanding of the domain ontology, including all the possible descriptive data.

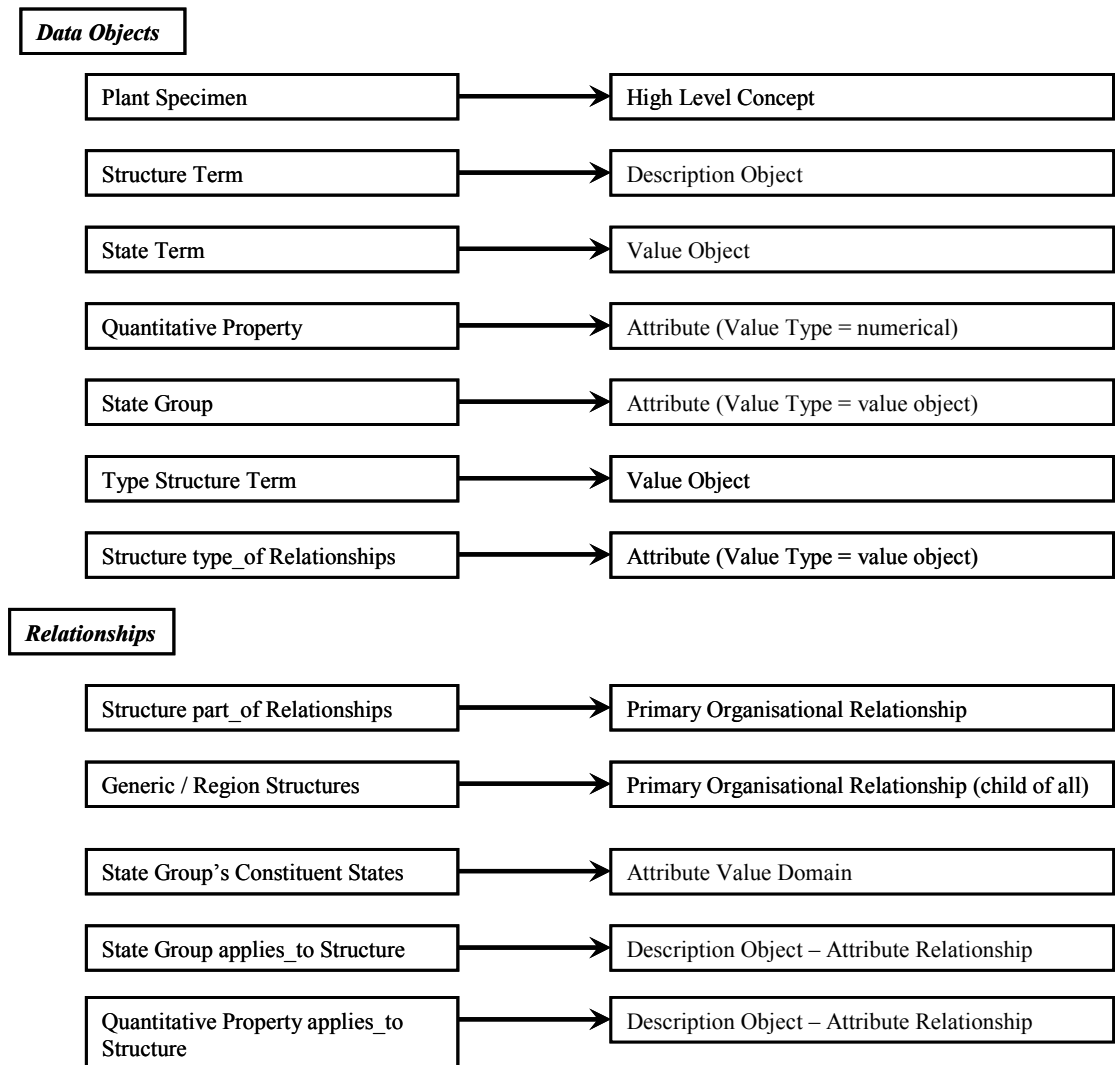
The potential variation in composition of domain ontologies makes their automatic adaptation for a domain model a nontrivial task [Wang 2002] that defies automatic mapping of the domain ontology, thus requiring the intervention of an IT expert actor. The IT expert makes a mapping between the **abstract domain model** and the particular domain ontology's conceptual model. This allows the system to derive the **concrete domain model** from an imported domain ontology as shown in figure 5.1.

It is only necessary to perform this mapping once for a given domain conceptual model. Where the database schema and ontology conceptual model are based on the same domain conceptual model, this mapping also allows the system to format the entered data for transfer back to the database application. Where this is not the case, a second expert mapping would be required for each database schema.

Initially the mapping was captured within a programmatic java class. Later a transfer XML format was developed into which the ontology was transformed either programmatically or manually (see chapter 8).

In order, to perform the domain ontology mapping, a number of key objects and relationships (from figure 5.2) need to be identified or derived. At the fundamental level, **description objects** need to be identified along with a primary description object hierarchy inter-relationship (to form a **description object hierarchy**). **Attribute** objects must be identified or derived from ontology terms and/or relationships between **description objects** and possible **value objects**. In addition, the applicability of **attributes** to **description objects** is identified. **Value objects** must be identified from the descriptive terms that could form possible values of a **description object** via an **attribute** relationship (**value objects** can also be **description objects** themselves or instances of **description objects**). Beyond these basic terms and relationships, the

**abstract domain model** can have modifiers, units and various other aspects mapped to it.



**Figure 5.4: Mapping from Angiosperm Ontology Conceptual Model to Abstract Domain Model**

Figure 5.4 shows the major elements in mapping the angiosperm ontology conceptual model (figure 5.3) to the **abstract domain model** (figure 5.2). ‘Plant Specimens’ represent the **high-level concepts** that are described. Their constituent ‘Structures’ map to **description objects**, and their ‘is-part-of relationships’ form the **description object hierarchy** relationship. Figure 5.5 shows a simple example of mapping to description objects, where the permitted ‘structure term’ ‘part-of’ relationships are displayed as a directed acyclic graph on the left. The materialised **description objects** are highlighted in the tree view of the **description object hierarchy** on the right.

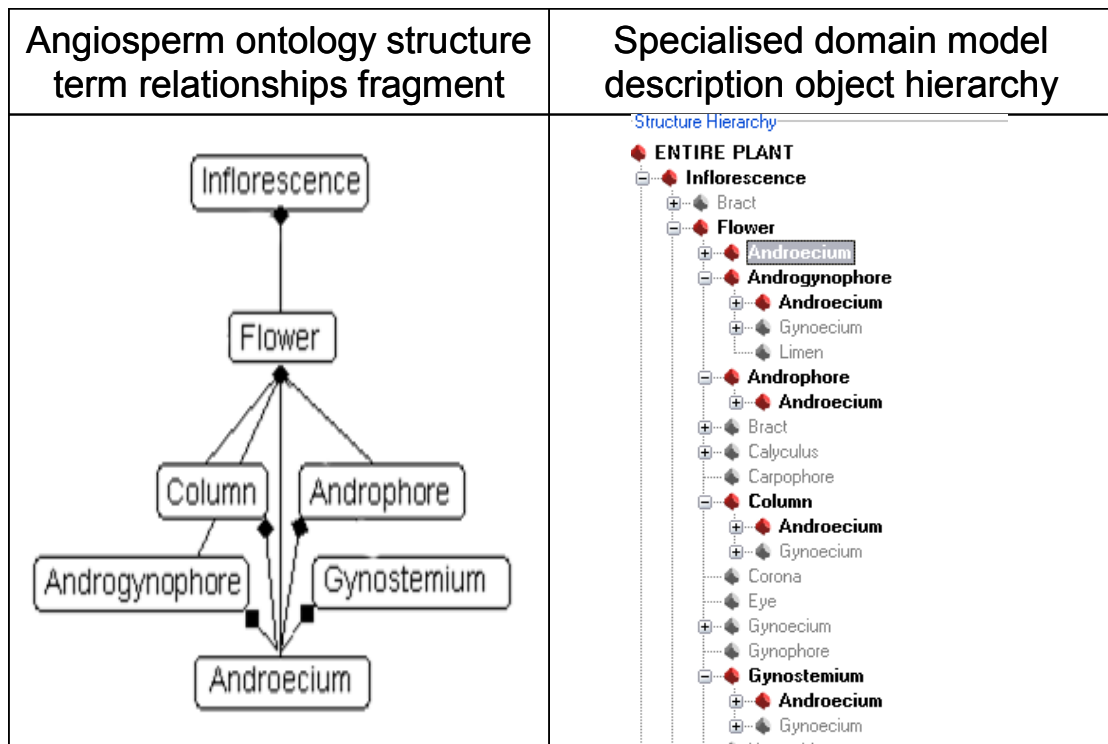


Figure 5.5: Mapping structure terms to description objects.

‘Properties’ and ‘State group’ relationships form the **attributes** of a **description object**. ‘States’ form **value objects**, which belong to an **attribute** and which are constrained over a **value domain** defined by the permitted grouping relationships between ‘Structures’ and ‘States’. The ontology had some ‘Type Structure’ ‘type-of’ ‘Structure’ relationships (see figure 5.3) e.g. ‘*umbel*’ is a type-of ‘*inflorescence*’. There was no further information in the ontology other than the definition of the ‘Type Structure’ (i.e. ‘*umbel*’ was not a described instance of ‘*inflorescence*’, just a defined term). A short analysis showed taxonomists wanted to be able to choose between different ‘Type Structures’ at data entry. Accordingly the ‘Type Structures’ were mapped to **value objects** of an **attribute** named “Type of <Structure name>” for the **description object** mapped from the ‘Structure’.

The **concrete domain model** is thus the representation of the domain ontology in terms the system understands and hence the initial domain knowledge for the system. It contains all the descriptive terms and permissible relationships between the terms that can be used to form descriptions within the system. It also contains some macro data to aid presentation, such as domain specific terminology for the types of descriptive terms. For example in angiosperm taxonomy that **high-level concepts** are ‘Plant Specimens’ or that **description objects** are ‘Structures’.

### 5.3.4 Specialised Domain Model

#### 5.3.4.1 Introduction

The **specialised domain model** is a subset of the **concrete domain model**. It cannot contain elements which contradict the **concrete domain model** and thus the underlying domain ontology. Domain expert users create a **specialised domain model** for individual projects utilising the **Specialisation I/F** to effectively edit the **concrete domain model** as seen in figure 5.1. The **specialised domain model** is thus simply the transformed **concrete domain model** containing only the descriptive elements of relevance to a particular project.

A data entry interface based on the whole angiosperm domain ontology would be too large in terms of its usability and would cover a much larger number of ‘Structures’ and characteristics than a taxonomist would utilise in any one taxonomic project. Individual projects are typically restricted to only a small subset of the angiosperm group of plants. As was seen in chapter 2, taxonomists are interested in different sets of specimen characteristics dependent on the focus of their work. The exact data requirements of a given taxonomic project must therefore be established. Normally, taxonomists do this by creating paper-based proforma templates for each project, which have entries for the major describable characteristics of the specimens that they wish to record. In the approach in chapter 4, taxonomists specified an electronic proforma to this end. The ontology-based system provides an equivalent to this process by allowing taxonomists to edit the **concrete domain model**. Interface constraints on users ensure the edited model does not contradict the underlying ontology.

These **specialised domain models** enable the system to present data entry forms based solely on the data and semantics relevant for the users’ particular project. Below some nuances of the specialised domain model that can be specified by users are discussed

#### 5.3.4.2 Identified descriptive data elements of interest

In order to generate a data entry interface based only upon the descriptive elements of interest to a particular project, the relevant data elements must be identified by taxonomists.



**Description objects** to be included in the **specialised domain model** must be identified by users. The logic of the **description object hierarchy** means that there must be a path of included **description objects** from any included **description object** up to the root. For each included **description object**, any **attributes** of interest also require to be identified. Likewise where those **attributes** have a value domain composed of **value objects**, the **value objects** to be included in the value domain require to be identified.

### 5.3.4.3 Concrete description instances

By default the **description objects** being instantiated are considered to be abstract concepts (for example the **description object** ‘leaf’ refers to leaves in general on a plant). **Description objects** can also be declared to be concrete instances as opposed to being abstract. In this case the eventual entered data refers to an individual specific instance of the **description object**, and all its instantiated **attributes** are grouped for that one instance. For example the concrete **description object** ‘leaf’ refers to a specific leaf on a plant specimen. At the data entry stage, multiple instances of the one concrete **description object** can be captured. The exact number of concrete instances to be captured is determined at data entry and not at the specialisation stage.

The data entered for the **attributes** of the concrete instance are grouped together. This allows users to record a volume of related numerical data suitable for later statistical analysis. Recording concrete data also avoids the loss of data through the amalgamation of results which can occur in the more abstract case. For example a user could record that leaf #1 was rough and 16mm in length; leaf #2 was smooth and 5mm in length; leaf #3 was rough and 21mm in length; leaf #4 was rough and 17mm in length; leaf #5 was rough and 19mm in length; leaf #6 was smooth and 3mm in length. The abstract data recorded would simply be that the leaves were rough or smooth with a length in the range of 3-21mm.

Concrete status does not however affect other **description objects** by dint of their relationships to the concrete one. So that the data for **description objects** that are children of a concrete **description object** is not related to the individual concrete instances. Where there is a desire to relate the data from different **description objects** in such a manner, another mechanism, cloning, exists.

### 5.3.4.4 Clones

Normally a **description object** is instantiated once. There are however occasions where two or more **description objects** with the same ontology based composition (i.e. path to root) and description (same possible related **attributes**, **values** and related child **description objects**) are useful.

The primary example is where the user knows in advance of data entry that there is more than one distinctive type of the **description object**. For example, the user, specialising the angiosperm ontology for a project looking at the *Alyxia* group of plants, knew there were two distinct types of 'flower' in the plant specimens that they were interested in. These plants, the user knew, often had small 'terminal' (situated at the apex) flowers in addition to the main ones. In addition to many characteristics of the main flowers, the user was interested in the number of these terminal flowers and in the presence of 'bracteoles' (a sub-structure of 'flower', i.e. a child **description object** of the 'flower' **description object**). In order for clarity, the user thus required two separate **description objects** for 'flower'.

The domain model accommodates this by allowing the use of clones. A cloned **description object** has all the possible **attributes**, **values** and child **description objects** of the original, but can be independently specialised, for example by including different description elements.

Cloning **description objects** differs from using a concrete concept. Clones can have different description possibilities defined for each of the clones at the specialisation stage, whereas concrete instances all have the same description possibilities. Additionally, the description of the child **description objects** is linked to the relevant clone. In the concrete concept case, the child **description objects** of the original abstract concept are not affected and do not have data recorded for them separately for each concrete instance.

Figure 5.6 shows an example of cloning. Each box represents one description object with its own unique path to the root (those with the same identifying letter are based on the same defined term, but are different description objects because of their different composition). The user clones the description object F with the compositional path I – F. This results in a clone F2, which has the compositional path I – F2. The child

description objects of F are the same as those of F2, except that their compositional path includes F2 rather than F, e.g. I – F2- B rather than I – F – B. The included status of attributes and child description objects of F2 is inherited from F at the point of creation, but thereafter F2 is independent of F.

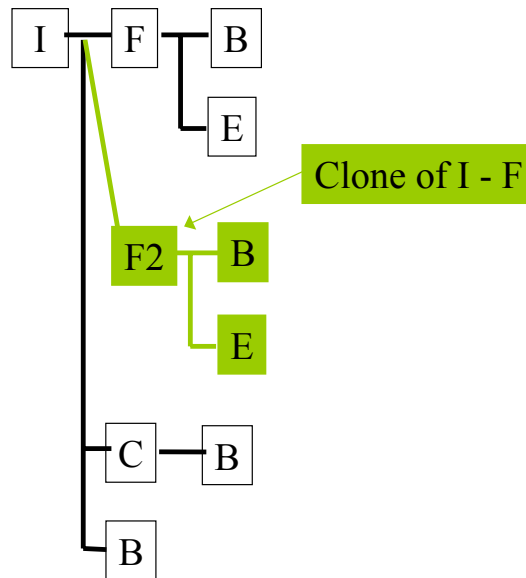


Figure 5.6 Cloning description objects.

#### 5.3.4.5 Ratios and Relative Characteristics

The specialisation process allows users to build some complex descriptive concepts that involve more than one **attribute** or **description object**. The primary reason for these more complex descriptive concepts is to allow the recording of ratios and other concepts that relate two **attributes** of the same or different **description objects**. There are a number of common uses for these descriptive concepts. For example, in a system test with the ‘*Ranunculus*’ group of plants, the expert user wished to record the relative colour hue of the upper surface of the leaf as opposed to the lower surface. An example of the use of ratios appears in the ‘*Alyxia*’ group of plants where the user wished to record the ratio of leaf length to leaf width.

The **specialised domain model** allows **attributes** to be related to other **attributes**, using a set of **relative modifiers** that are included in the ontology. These modifiers can alter the **attribute’s** value domain, for example to change numerical entry to a domain of relative values such as greater-than, less-than, etc. The value domains of **relative modifier** altered **attributes** are determined by the imported ontology. Not all ontologies will have such terms depending upon their nature.

A related complex descriptive alteration of **attributes** is spatial modified **attributes**. In this case the user uses a **spatial modifier** to form a link to another **description object**, in order to clarify the **attribute**'s position in relation to the **description object**. An example of this would be to note where a measurement should be taken on a plant structure e.g. in the '*Alyxia*' group of plants the user wished to record the presence of '*bracts*' at the '*base*' of the '*pedicel*' and their presence '*below*' the '*calyx*'. This involves 2 specialised attributes, each with a spatial modifier (e.g. description object '*flower – bracts*', **attribute** '*presence*', **spatial modifier** '*at*' with target **description object** '*flower - pedicel – base*'). The allowed **spatial modifiers** are also taken from the imported ontology.

### 5.3.4.6 Preferred Units

**Attributes** with numerical value domains can have a preferred unit term attached to them (see figure 5.2). Users are able to access and alter this aspect of the **specialised domain model**. The underlying ontology may have default preferred units of measurement for **attribute** terms that would be reflected in the **specialised domain model**, although the angiosperm ontology does not. The units of measurement are also taken from the imported ontology. If the base underlying ontology does not have unit terms, the IT expert performing the initial mapping of the ontology for the system, would require to add units of measurement either themselves or from a secondary source.

## 5.4 Presentation Models

There are two presentation models in the system one for ontology presentation and one for data entry (see figure 5.1).

In order to allow the expert taxonomist to create a **specialised domain model**, the system uses a modelling tool (the **Specialisation I/F**) which presents the entire angiosperm domain model for exploration and editing. The **ontology presentation model** is used in this tool to provide a general layout presentation for displaying ontologies based on the **abstract domain model**. This presentation is designed to present a relatively simple editing interface for a non-IT user. This presentation model is also utilised to display aspects of the **specialised domain model** in the final data entry

## **Chapter 5 - Ontology-Based Generation of Data Entry Interfaces**

interface. The **ontology presentation model** is captured within the system and is based on the presentation of the storyboard approach in chapter 4, with changes to reflect the requirements of the ontology based approach. The model is refined through the iterative development and testing (see chapter 6).

The **data entry presentation model** determines the layout and selection of interaction objects for the data entry interface. Again this is based on the work done in chapter 4, refined through development and testing (see chapter 7).

The presentation models will be discussed in further detail in the following chapters.

### **5.5 Development Methodology**

#### **5.5.1 Methodology Principles**

Developing tools for taxonomists was conducted in a user centred approach, involving an early focus on developing user requirements and understanding, then iterative analysis, design, development and testing. These design principles have been articulated and demonstrated in studies for the design of effective computer systems for some time (e.g. [Gould 1985]).

After the problem domain had been investigated and analysed in the first development stage as described in chapters 2-3, the design and development of prototype solutions progressed through a number of iterative cycles. The second development stage, aimed at identifying a workable approach to the identified problem, involved the Use Cases and storyboards described in chapter 4. The remaining development stages involved the development of interactive tools. The interactive prototype tools were iteratively developed to explore, evaluate and refine the approach.

#### **5.5.2 Evaluation of Interactive Tools**

During each interactive development stage, RBGE taxonomists participated in the evaluation of prototypes. Evaluation concentrated on qualitative feedback as there was no comparable system with which to measure the effectiveness of the approach. The current paper-based system does not collect defined, structured data of the sort

## Chapter 5 - Ontology-Based Generation of Data Entry Interfaces

envisioned and neither does any other taxonomic tool. While some parallels can be tentatively drawn with current practice in evaluating the specialisation process, too great an emphasis cannot be placed upon any such comparisons.

In the main test, during each phase, users were asked to perform sets of representative tasks during which they were encouraged to talk about their actions, observations and problems. This ‘think-aloud’ methodology [Lewis 1982] is designed to elicit qualitative feedback from a relatively small number of users. Interviews were utilised at the conclusion of the set tasks to follow-up issues. Since the user group was generally small, interviews were used rather than questionnaires since they could be more flexible, both in ability to explain difficult questions in depth, avoiding misunderstanding of questions and in the ability to ask follow-up questions on areas of interest that arise in the evaluation. In-depth understanding of any issues was considered more important than analysing the hard numbers that could be collected from questionnaires. The users were professional taxonomists who represented the real end-users of the system and as such were the ideal test subjects to select [Nielsen 1993]. As is common in professional and corporate environments the tests had to be kept relatively short (e.g. [Weiss-Lijn 2001]). On average 1 hour was given to each user in each test, which was long enough for representative tasks to be completed.

Observation of the individual users using the system to perform full actual tasks was used where possible to back-up the other evaluations. Additionally users were interviewed to gain insight into their experiences with the system. The results of these full tasks (specialised domain model files and specimen descriptions) were also analysed. These full task tests helped ensure that we were not overlooking issues due to the nature of tasks we set in the main tests. Small focussed tests were also utilised to gain feedback about specific features from a taxonomist working on the Prometheus II project.

Peer review by computer scientists on the Prometheus II project, was used to evaluate interim designs using informal heuristic evaluation techniques. These evaluators had both some experience of the domain and usability, which Nielsen [1992] showed significantly improved the number of usability problems they could find.

### 5.5.3 Interactive tool development phases

The third development phase involved testing the articulated approach for specialising the domain ontology for a project. At this stage, the full angiosperm ontology had not been developed. The description data from a proforma developed for the *codnopsis* group of plants was transformed for use as an ontology to test the functionality of the specialisation interface hierarchical description view. Initial informal feedback on prototypes was received from two RBGE taxonomists and from computer scientists on the Prometheus II project staff. A formal user test with three RBGE taxonomists was undertaken.

The fourth development phase utilised the angiosperm ontology and extended the interactive tool to include data entry interface. It also revised the specialisation interface. As the interface reached a degree of full functionality we were able to undertake three evaluations of users undertaking the task of specialising the ontology and entering specimen data for basic cases with few descriptive complications (*Codnopsis* and *Begonia* datasets – one user, *Cyclocodon* dataset – one user, *Umberlifferae* dataset – two users). A full user test involving six RBGE taxonomists was conducted.

The fifth development phase further revised the system and its associated interfaces. Some extra functionality was added. It aimed to test the effect of the whole system with users in depth by having full tasks completed and evaluated, with potentially more complicated and extensive data (Middleton's *Alyxia* dataset – two users, *Prunus (Bhutan)* dataset – one user, *Ranunculus* dataset – one user). It also aimed to test a wider user population, to which end, a user test with thirteen representative taxonomists was completed. A short follow-up evaluation to this last test was made to answer some outstanding issues regarding the editing of some complex aspects of the domain model.

A final prototype development was also tested in another domain, that of the TCS bio-informatics XML transfer schema [TDWG 2005] as discussed in chapter 8.

Fuller details of individual tests are given in relevant evaluation section. Test scenarios and interview guides are shown in Appendix D-E.

## **5.6 Conclusion**

A system to capture high-quality descriptive data, relevant to individual projects of work, utilising two UIs for specialising data requirements and entering data has been introduced. The main models (task, domain and presentation) used by the system have also been introduced. These basic elements can be found to an extent in various model-based user interface development environments, however these do not provide for end-users to determine project data requirements, as the interfaces are designed for use by IT experts rather than general end-users. Nor where data entry interfaces are automatically generated from data requirements, do they specifically address supporting the needs of high quality data entry. The described system uses an imported ontology to represent domain knowledge, defining terms and relationships to attempt to improve clarity, comparability and give appropriate guidance to users.

The following two chapters examine the two main user tasks, specialising data requirements and data entry, in detail. They will examine the interfaces and models that support them, along with results of the evaluation process.



## Chapter 6

# Specialisation Process

### 6.1 Introduction

This chapter discusses the specialisation process and its associated user interface. As described previously, in the specialisation task the user specifies the descriptive data to be collected on each specimen for a given project of work, based upon the underlying domain ontology. That domain ontology is presented to domain experts for editing and specialising in a **specialisation interface**. This interface is system generated using an **ontology presentation model** which acts upon the **concrete domain model** (the system's understanding of the domain ontology). The end result of this process is a **specialised domain model** based upon and consistent with the underlying ontology, which only contains the data relevant for the given project of work.

The specialisation interface was iteratively developed in the 3<sup>rd</sup> to 5<sup>th</sup> development phases as described in 5.5. During the 3<sup>rd</sup> phase our approach was developed with interactive prototypes whose presentation was loosely based on the storyboards discussed in chapter 4. This approach made use of a simple ontology for the description of *codnopsis* data which it presented to the users for editing (see 5.5.3). The basic approach was supported by the evaluation except that users found it easier and more positive to select presented descriptive element relationships for inclusion, than to begin with most of them included and edit out those they did not require. A fuller description of the 3<sup>rd</sup> phase development and other evaluation results is available in Appendix D.

Following the 3<sup>rd</sup> stage evaluation, the interface was substantially refined for the 4<sup>th</sup> development phase when the angiosperm ontology was used. The fundamentals of the process and interface did not change during the 4<sup>th</sup> and 5<sup>th</sup> phases of development. Refinements and added functionality for the interface did take place due to the 4<sup>th</sup> stage evaluation and this is noted in the relevant sections below. The 5<sup>th</sup> stage also made minor refinements to the interface to address usability issues and to add further functionality. A final wide user test of the approach was undertaken towards the end of the 5<sup>th</sup> stage of development. In addition to developing this approach, the evaluations of

## Chapter 6 - Specialisation Process

our approach gave rise to a number of issues that also helped evaluate and refine the angiosperm ontology, as weaknesses in the ontology were made apparent through usage.

This chapter discusses the domain, presentation and task models as they were used in the specialisation process in the 4<sup>th</sup> and 5<sup>th</sup> development phases leading up to the final user test. The remainder of the chapter discusses the various issues that arose during the evaluation and testing of the process.

### 6.2 Domain Model

The **concrete domain model** represents the system's baseline understanding of the imported domain ontology. The **specialised domain model** is the transformed **concrete domain model** identifying the descriptive data elements of relevance to a particular project of work. It cannot contain elements which contradict the **concrete domain model** and thus the underlying domain ontology. In contrast to the 3<sup>rd</sup> phase, no elements of the ontology are initially included by default in the **specialised domain model**. The basic format of the **specialised domain model** has already been discussed in the previously (5.3.4). This section discusses extra elements of the domain model that were added for the needs of our approach in the 4<sup>th</sup>-5<sup>th</sup> stages. These elements are defined by users, not derived from the ontology.

#### 6.2.1 Names

One issue that quickly arose with clones (5.3.4.4) was that the names used for the **description object** clones needed to be differentiated from each other. Whilst the system could generate differentiated names with numbers or more meaningfully by using any fixed **attribute** scores (see below), the most meaningful names were those determined by the expert user who knows the concept underlying the cloned **description object**. The domain model was thus expanded to allow for a nametag to be attached to instantiated **description objects** within the **description object hierarchy**. This nametag does not alter the underlying ontology definition.

The name used for the specialised **attribute** in the interface can also be altered to better match the user's conception and thus act as a better aide memoir (such changes have no

## Chapter 6 - Specialisation Process

ontological weight). The default names from the ontology for **attributes** were not found to always be clear to different users. This could represent a weakness in the angiosperm ontology which was created in a field where there is little agreement on terminology.

Allowing expert users to incorporate nametags that are more in tune with their concepts, whilst not allowing them to create new defined terms, allows them to continue to work with the underlying ontology. The danger of users using nametags to circumvent the ontology constraints and inappropriately use a defined term is, however, a continuing concern.

### 6.2.2 Fixed score

The **specialised domain model** allows instantiated **description objects** to be determined as having certain characteristics that hold true whenever the **description object** is determined as present for a specimen during data entry. This allowance is in response to two scenarios that arose during development.

The first is to do with cloning, where one clone may be differentiated from the other due to always having some characteristic. In the ‘*Alyxia*’ example in 5.3.4.4, the ‘*flower*’ clone had the **attribute** ‘*arrangement:position:general*’ fixed with the **value object** ‘*terminal*’ to represent the terminal flowers. If a specimen has any data recorded about the ‘*flower*’ clone (or its descendents), it will also have this fixed score linked to the clone to aid correct interpretation of the data.

A variation on the basic cloning scenario exists where users use fixed scores to effectively pre-define through description a number of alternative clones. The clones would then function as summary types of a **description object** and its descendents, allowing the user to select the summary(ies) that was present on a specimen at data entry. A user using the system to enter legacy description data (also about the ‘*Alyxia*’ group of plants) attempted to use the fixed score facility in such a manner to attempt to match the original gross summary characteristics, with 6 clones of *infructescence seed embryos*. Each clone had different substructures with different fixed scores. Whilst the user was able to record the legacy data, this is not an ideal use of the provision during normal data entry. The quality of the data entry would be theoretically better, if it was

## Chapter 6 - Specialisation Process

recorded based on observation of the actual characteristics, rather than upon pre-conceived amalgamations of possibly related characteristics.

The second related case is one where the user wishes to record some descriptive characteristics that they know holds true for all the specimens they wish to record in their project. They may wish to record this detail to allow for later comparison with other data sets and for added clarity. Many users will not wish to note characteristics common to all their specimens, as they are really only interested in recording data for purposes of differentiating their specimens from each other. If there is no benefit from the extra data for the immediate user, then despite the longer-term benefits to others of increased clarity and comparability of the final descriptive data set, then they may not invest the extra time required. By allowing users to mark-up the **specialised domain model** with fixed scores at the specialisation stage, the extent of that extra time is less than if the user was required to enter the data for each specimen during data entry.

### 6.3 Task Model

As discussed in the previous chapter, the task model is encapsulated within the system and is not modifiable, except for the default order of the data entry task. The specialisation process gives a wide degree of freedom to users as to the order in which they specify the elements of the ontology for inclusion in the **specialised domain model**, allowing users to operate in a way that matches their individualistic working practice and cognitive process.

Whilst ideally the specialisation task should be entirely completed before specimen data is captured, users do not always have time to restart everything over again when alterations to the data to be collected are discovered after data entry has begun in a project. The specialisation task can thus be mixed with the data entry task to allow some level of iterative working practice, revisiting specialising the ontology after entering data for some specimens and discovering additional potentially useful features to capture data about.

### 6.3.1 Primary specialisation task

The primary specialisation task is the repeated task of adding a specified **attribute** to the **specialised domain model**. This is analogous to the adding of ‘characters’ to the proforma as discussed in chapter 4. There is no required order in which **attributes** are specified, nor is there any requirement that an **attribute** be fully specified before beginning the specification of another **attribute**. The user is free to return to previously specified **attributes** and edit their specialisation.

Specifying an attribute includes finding and identifying the **description object** and **attribute**, as well as specialising the **attribute** by identifying value domains, preferred units, fixed scores, relational and spatial modifiers. The name used for the specialised **attribute** in the interface can also be altered.

### 6.3.2 Other supporting tasks

There are a number of other supporting tasks in the specialisation process. Some of these are briefly discussed below. A summary of tasks is given in table 6.2.

#### 6.3.2.1 Check definitions

During the specialisation task at almost any stage, users may seek to check the definitions of the descriptive terms they are using to determine the ontology definition. All **description objects**, **attributes**, and **value objects** have definitions.

Based on the 2<sup>nd</sup> and 3<sup>rd</sup> development phases, mouse-over definition access is used to give a quick check or confirmation of a definition for terms the user is currently using or considering using. Invoking a pop-up definition box is more commonly used when either multimedia definition aspects are very important to comprehension (e.g. leaf outline shapes) or where detailed comparison of multiple descriptive terms is valuable such as when a number of similar ontology terms exist.

## Chapter 6 - Specialisation Process

### 6.3.2.2 Check for existence of desired terms

When determining how to specify a descriptive concept as one or more specified descriptive **attributes**, users may desire to check if descriptive terms they know from their own domain knowledge are represented in the ontology.

### 6.3.2.3 Review specialised domain model

To insure the **specialised domain model** reflects the user's intentions, requires reviewing the model as specialised so far. This can be undertaken at any point and should ideally be undertaken before collecting data based upon the final **specialised domain model**. There are three main alternatives for completing this task.

The first alternative is by using the main specialisation interface (see table 6.1). The second alternative for reviewing is to access a preview of the data entry interface. A tab provides access to data entry screens based upon the current state of the **specialised domain model**. Although users cannot edit the presentation of the data entry interface directly, this can provide a useful preview of the effects of the data choices the user has made. It also provides a different visualisation of the **specialised domain model**. Lastly a third alternative review can be made by viewing a XML output of the **specialised domain model**. As the user must go outside the system, this would tend to be done only near the end, whilst the other review methods would be more likely to be used at any time.

### 6.3.2.4 Alter default task order

The **data entry task model** is encapsulated within the system, but one part of it is user modifiable and is presented to users: the default order of data entry tasks. During the data entry stage, each included **description object** in the **specialised domain model** is presented to users for data entry. The default order in which the **description objects** are presented is by a top down, depth first enumeration of the **description object hierarchy**.

In the taxonomy example this functionality is provided to reflect the working practice of taxonomists, who, within the general task of describing specimens, may want to specify the order in which they describe the particular characteristics of the specimen, to fit with

## Chapter 6 - Specialisation Process

traditional biological description methodologies (acryptic order). Evaluation during the 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> evaluation phases consistency show this, unfortunately it was never represented in the angiosperm ontology to give this domain knowledge to the system. Allowing users to alter the order was thus instituted in the 5<sup>th</sup> development phase.

### 6.3.2.5 Save and load specialised domain models

Users can save the state of the edited **specialised domain model** at any time. This saves an XML file to the user's computer file system. Previously edited specialised domain files can be loaded after the ontology has been loaded to the system. Loading a **specialised domain model** file can only be done to a specialisation interface based on the same version of the ontology.

### 6.3.3 Specialisation Task Restrictions

The domain expert cannot directly alter the **data entry presentation model** (for example by choosing the actual data entry abstract interaction objects, although they can alter the data in the **specialised domain model** upon which determinations are made by the **data entry presentation model**). This ensures a modelling split between data determination and presentation, thus avoiding confusion between the two different processes. It also emphasises that the taxonomists do not have to perform the job of designers.

## 6.4 Ontology presentation model

The specialisation interface is automatically generated by the system based upon an **ontology presentation model** that presents the underlying ontology to users for controlled editing. The system interprets the **task** and **concrete domain models** using the **ontology presentation model** to determine the layout and interface interaction objects. This section discusses the **ontology presentation model** that was developed during the iterative testing of the system with the co-operation of RBGE taxonomists.

### 6.4.1 Interface Designs

The adopted approach presents the ontology for users to select those elements they are interested in as the primary paradigm. Only where there were no explicit ontological relationships, are users required to specify their own relationships based on ontology and abstract domain model rules (e.g. clones, universally applicable description objects, relational modifiers).

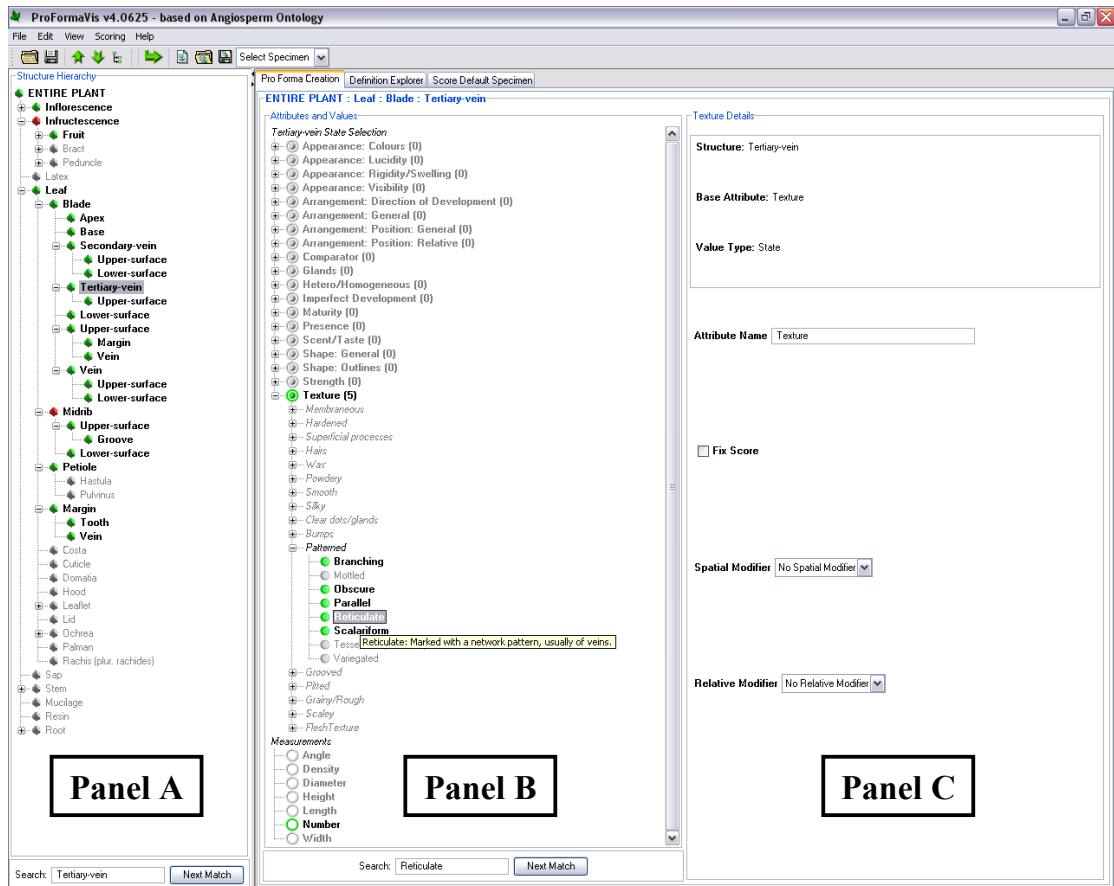
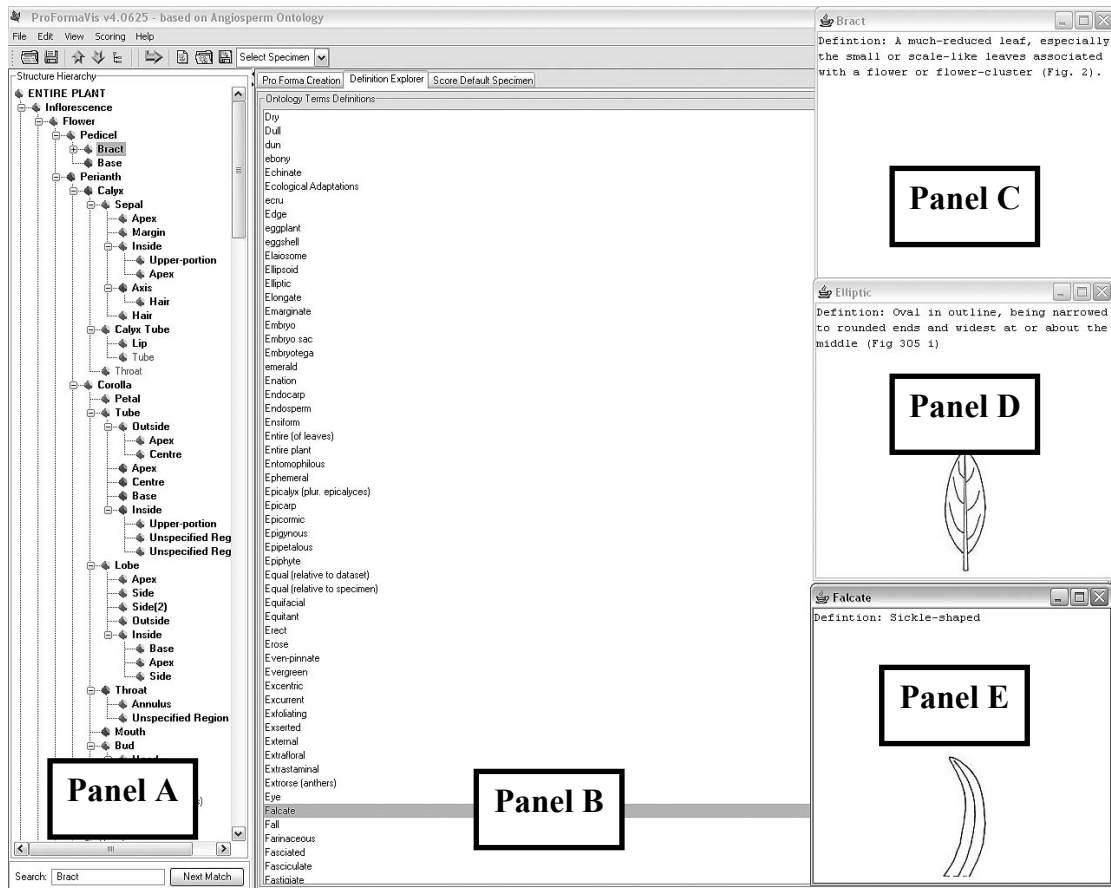


Figure 6.1: Specialisation Interface Screenshot

Figure 6.1 shows the main screen of the specialisation interface with its three main components. Panel 6.1A represents the **description object hierarchy**. Panel 6.1B represents the potential **attributes** and their related possible **value objects** for a selected **description object**. Panel 6.1C provides further interaction with the selected **attribute**.



## Chapter 6 - Specialisation Process



**Figure 6.2: Specialisation Interface: definitions Explorer Tab Screenshot:**

Figure 6.2 shows the supplementary definitions explorer screen which provides direct access to the defined ontology terms and their related definitions for informational purposes. Panel 6.2A still represents the **description object hierarchy** overview. Panel 6.2B represents an alphabetical listing of all defined terms from the ontology. Panel 6.2C, D & E are definition boxes showing any text and multimedia definitions for selected ontology terms.

### 6.4.2 Presenting the specialised domain model

The **ontology presentation model** maps elements of the ontology (as understood by the system in its domain model) and the implicit task model to the interface. Both interactive and informative elements are included in the **ontology presentation model**.

The description template (proforma) of the plant specimen is represented by the combination of all the included descriptive elements of the **specialised domain model**. The **ontology presentation model** differentiates such elements' inclusion by boldness

## Chapter 6 - Specialisation Process

of the typeface, with non-included elements being 'greyed-out' (or not represented if a filtered view is requested by users).

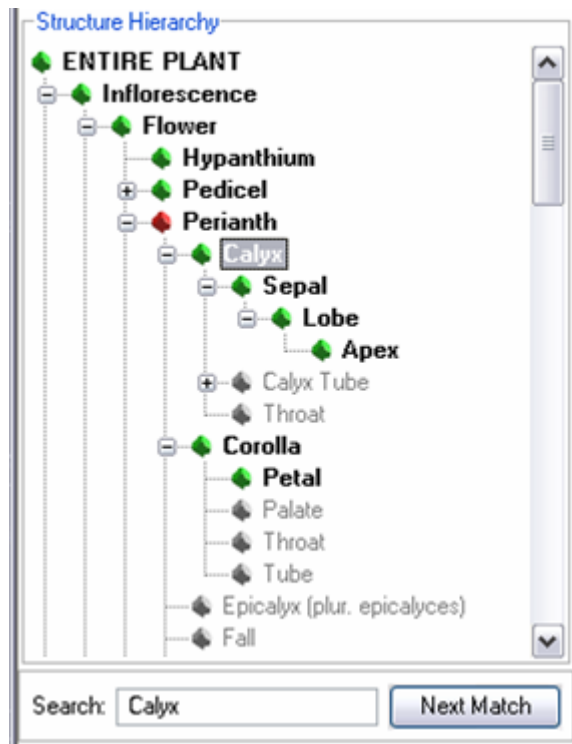
An alternative paradigm for representing the description template that was considered, but rejected, during development was a second hierarchical tree view that only contained the **specialised domain model** elements, with the first view being a static representation of the **concrete domain model**. This paradigm however was wasteful of limited screen space and would involve redundant repeated information for users. The second view would also confuse the navigation of the interface. Clarity of the view of the **specialised domain model** could be achieved by filtering the view to exclude non-included elements, when appropriate. User testing indicated this clarity of the overview was only required on occasions of reviewing their work at the end of the process.

### 6.4.3 Mapping the domain and task model to the interface

#### 6.4.3.1 Description object hierarchy

The **description object hierarchy** is mapped to a collapsible tree visualisation, as seen in figures 6.1A, 6.1A, 6.3. This visualisation shows the hierarchy relationships in a file tree format. Like most file tree type visualisations, the user can expand and collapse sections of the tree to filter the view. Clone **description objects** can only be shown once created. This **description object hierarchy** tree element is shared with the data entry interface, where it is still controlled by the **ontology presentation model**.

## Chapter 6 - Specialisation Process



**Figure 6.3: Description Object Hierarchy Tree**

In the 3<sup>rd</sup> phase prototypes (see appendix D), **attributes** were combined with the **description objects** in a main view. Concerns were raised at that stage about confusion if there were lots of attributes represented. During development of the system prototypes using the angiosperm ontology, representing the **attributes** along with the **description objects** had the consequence of making this visualisation cluttered and difficult to navigate. This was in part due to the number of **attributes** that were involved.

The angiosperm ontology restricts relationships; however it still allows a large number of **attribute** relationships. The exact number of **attribute** relationships varied depending upon the version of the ontology being used, but even in the final version that was utilised, there were 23 different **attributes** that applied to every **description object**. Additionally of the 70 **attributes** that are restricted to specific **description objects**, up to 27 can apply to any one **description object** (*flower* in this case), although usually only 0 to 5 extra **attributes** apply. Adding these 23-50 **attribute** child nodes to each **description object**, made the task of identifying and navigating by the **description object hierarchy** very difficult.

## Chapter 6 - Specialisation Process

Early in the 4<sup>th</sup> development phase, the option of including **attribute** nodes (represented with different icons and colour from **description object** nodes) in the **description object hierarchy** view was demonstrated to two taxonomists for comment. Users' responses to the visualisation included “*confusing*” and “*harder to find things*”, with the general view that they found the ontology easiest to follow when the **attributes** were represented in a separate linked view. Expert peer informal usability heuristic review confirmed these findings.

### 6.4.3.2 Description objects

**Description objects** are mapped to nodes on the **description object hierarchy** tree (see figures 6.1A, 6.2A, 6.3). These nodes have a degree of interactivity to control the focus of the interface's view and edit the specialised domain model. By selecting the node, the **description object's** descriptive **attributes** and related **value objects** are displayed in other panels of the interface (see figure 6.1B/C). Interaction on the node allows the user to alter the **description object's** inclusion status, concrete status and nametag as well as to add clones. Informational access to definitions is also linked to the nodes. The status of the **description object** is shown by various indicators of the node's presentation. Each **description object** node has an icon and a nametag.








The following representations (table 6.1) are used consistently by the presentation model in all of the collapsible tree visualisations for nodes representing **description objects**, **attributes** and **value objects**:

Specialised Domain Model	Ontology Presentation Model
<i>Inclusion status</i>	Icon and text is 'greyed out' for <b>description objects</b> not currently included in the <b>specialised domain model</b> . (An alternate view command from the menu bar toggles this representation to no representation at all for all non-included <b>description objects</b> .)
<i>Type of node (description object, attribute or value object)</i>	Icon shape
<i>Ontology based nametag</i>	Text content
<i>User edited nametag</i>	Mouse-over tool-tip
<i>Text definition</i>	Mouse-over tool-tip
<i>Currently selected node*</i>	Text highlighting box

**Table 6.1: Standard tree node indicators of underlying descriptive element status.**

(\* not from domain model)

The following indicators are specific to **description object** nodes:

- Concrete status: Icon shape indicates:
  - Abstract description only status: , , , .
  - Concrete description status: , , .
- Attributes: Icon colour indicates whether the **description object** has any descriptive **attributes** enabled for data entry.
  - Green for has one or more descriptive **attributes**.
  - Red for has no descriptive **attributes**.
- Clone status: Bracketed number in text content

Early interfaces used a bracketed number after the text content to indicate the number of specified descriptive attributes, but feedback indicated this was distracting to users and included more information than users required, as users only used summary indicators to determine if any attributes had been specialised for a particular description object. The bracketed effect was also found to be more useful for distinguishing clones, which 3<sup>rd</sup> phase testing had identified as an issue (Appendix D).

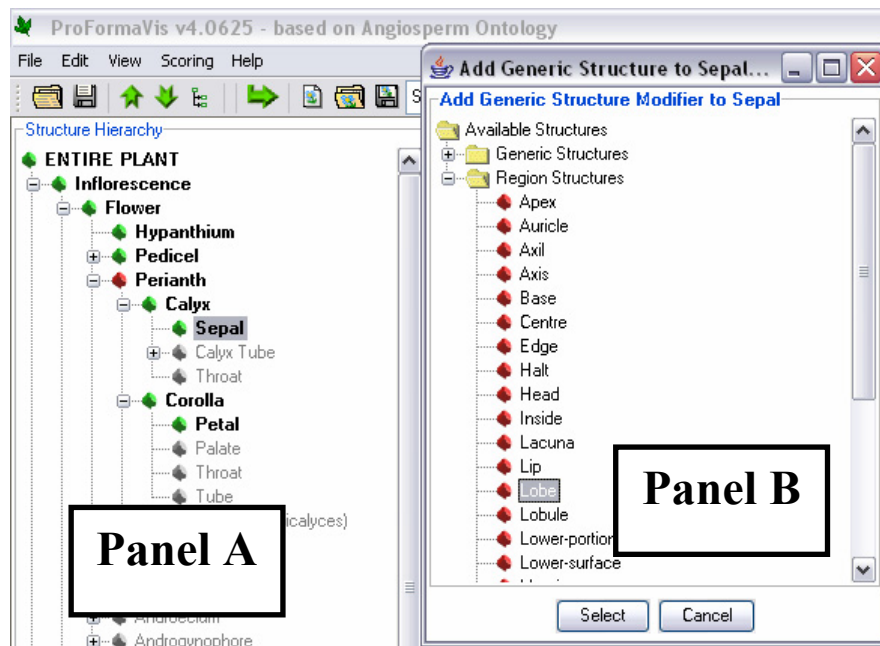
### 6.4.3.3 Universally applicable **description objects**

Some **description objects** can be instantiated within the **description object hierarchy** at any point. An example of this in the angiosperm ontology would be a generic region such as '*base*' or a generic plant structure such as a '*gland*'. In some ontologies such as the angiosperm ontology, to include all these possibilities in the displayed hierarchy would be unfeasible due to their sheer number which would swamp users' perception of the other **description objects**, particularly as only a very small percentage would ever apply to any particular project. As a result these **description objects** are only visualised in the hierarchy file tree upon a user including them in the **specialised domain model**. Essentially a user must define the part-of relationship

Users can utilise a command to access a view of all universally applicable **description objects** that could be included as child **description objects** of a selected **description object**. This presents users with a pop-up secondary file tree visualisation of the possible child **description objects** for inclusion (see figure 6.4B).

The organising hierarchy of this secondary tree is any *type\_of* universal **description object** relationship that is included in the ontology. In the angiosperm ontology example there are generic structures and region structures that are universally applicable. If there are no *type\_of* relationships for this case, then the visualisation is essentially a linear one. The organisation of the universal **description objects** within the *type\_of* demarcations is simply alphabetical, as there is no other domain knowledge from the ontology that would present a better organisation.

## Chapter 6 - Specialisation Process



**Figure 6.4: Including a universally applicable description object in the specialised domain model.**

This example from specialisation of the angiosperm ontology for the '*Prunus*' group of plants, adds a '*Lobe*' universally applicable **description object** to the selected '*sepal*' **description object**. Figure 6.4A shows the displayed main **description object hierarchy** with '*sepal*' selected. Figure 6.4B shows the pop-up tree for universal **description objects** where the user has selected '*lobe*'. Figure 6.5 shows the result of this operation upon the main **description object hierarchy** file tree visualisation with '*lobe*' included as a child of '*sepal*'. It is possible to add further children to '*lobe*' or '*sepal*' and in the '*Prunus*' example the user also included the '*apex*' universal **description object** as a child of '*lobe*'.

## Chapter 6 - Specialisation Process

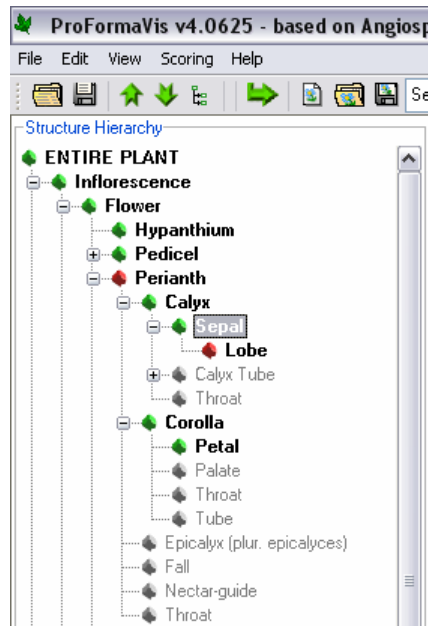


Figure 6.5: Result of including ‘lobe’

### 6.4.3.4 Description object’s attributes and value objects

When a node is selected in the **description object hierarchy** file tree, the **description object’s attributes** and their value domains are mapped to a collapsible tree visualisation – the **attribute-value tree** (see figures 6.1B, 6.7). This standard hierarchical visualisation can be expanded and contracted as required. It also automatically expands an **attribute** node to show the value domain (where applicable) when an **attribute** is included as ongoing evaluation showed that users will wish to select the value domain next.

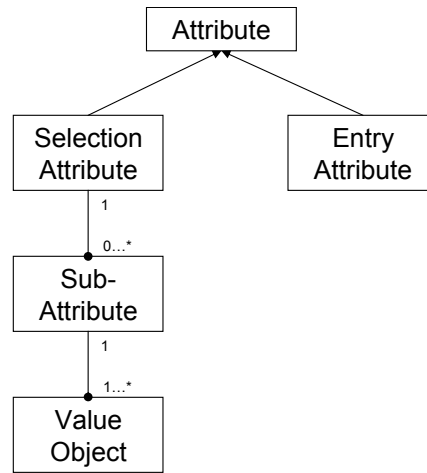
The invisible root node effectively represents the **description object**. The highest level visible nodes are the basic value constraint types of **attributes**: selection and entry. In the angiosperm ontology the text content for these nodes is taken from the ontology, which supplies domain metaphors for the concepts (namely ‘*state selection*’ and ‘*measurements*’). There is no interaction on these high level nodes.

The child nodes of the high level value constraint type nodes represent the **attributes** allowed for the selected **description object**. Figure 6.6 shows the components being represented in the hierarchical visualisation and figure 6.7 shows an example of this. Nodes representing **attributes** that can be instantiated with user-entered data such as free text or numerical data, do not have any sub-nodes. Nodes representing **attributes**



## Chapter 6 - Specialisation Process

that can be instantiated by selecting from a value domain do have sub-nodes representing related **sub-attributes** and the **value objects** that form the value domain.



**Figure 6.6: Attribute and value object hierarchy**

## Chapter 6 - Specialisation Process








Figure 6.7: Attribute-value tree example for selected ‘Lobe’ description object (*‘Inflorescence:Flower:Gynoecium:Pistil:Stigma:Lobe’*).

### 6.4.3.5 Attributes

**Attributes** are mapped to nodes on the **attribute-value tree** (see figures 6.1B, 6.7) for the linked selected **description object** node in the linked **description object hierarchy** visualisation (see figure 6.1A). These nodes have a degree of interactivity primarily allowing the user to change to the **attribute**’s inclusion status in the **specialised domain model** and by selecting the node, further details and specialisation options for the descriptive **attribute** are displayed in another panel of the interface (see figure 6.1C). Informational access to definitions is also linked to the nodes.

The status of the **attribute** is shown by various indicators of the node’s presentation including the consistent ones shown in table 6.1. Each **attribute** node has an icon and a nametag.

## Chapter 6 - Specialisation Process

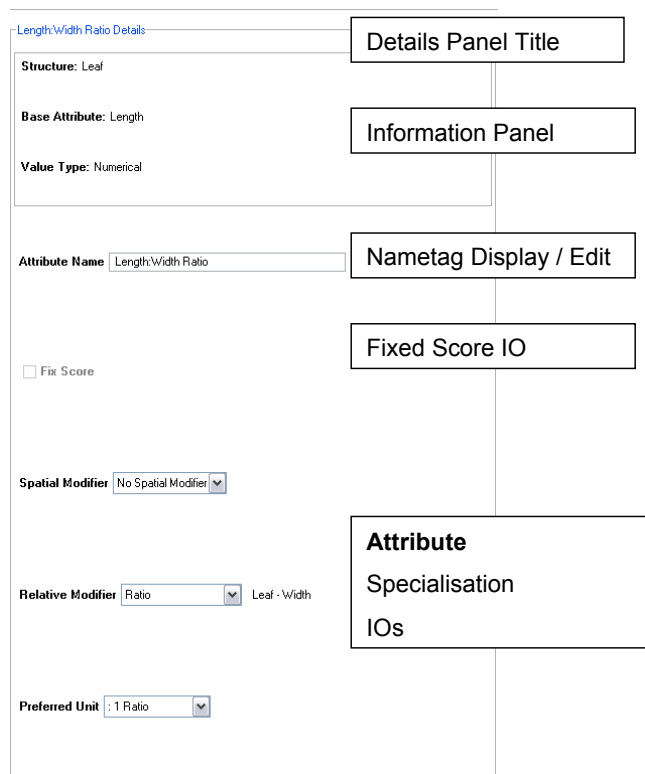
- User-edited nametag: Text content
- Value constraints: Icon type identifies the node as representing an **attribute** and represents whether the **attribute** can be instantiated with user entered data or with selection from defined **value objects**.
  - **Attribute** instantiated from value domain of **value objects**: , , .
  - **Attribute** instantiated by user entered data (with other possible value constraints): , .
- Value domain:

Icon colour and Text colour indicates whether the **attribute** has a viable value domain for data entry.

  - Green for has one or more **value objects** included or has a defined data entry value type.
  - Red text and icon indicates a warning that the **attribute** is included, but has no viable value domain.
  - Blue text indicates that the **attribute** is a fixed score.
  - Text content brackets indicate the number of included **value objects** in the value domain (where appropriate).

Figures 6.1C and 6.8 show the **attribute details panel**. This provides an informational display of the descriptive **attribute**'s details and interaction objects to specialise it. The details of the **attribute**'s value domain of **value objects** is dealt with within the **attribute**-value hierarchy, not in this details panel which deals with all other details.

## Chapter 6 - Specialisation Process



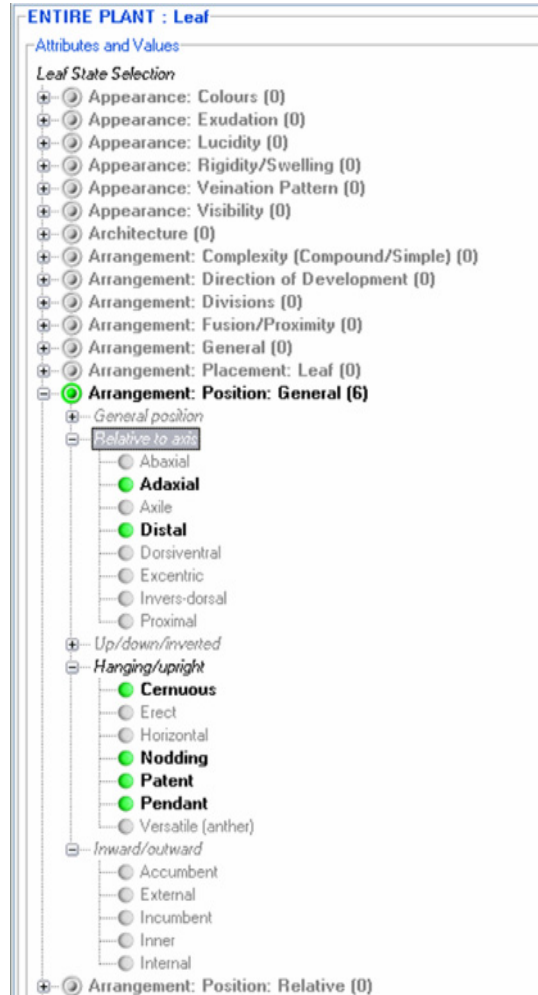
**Figure 6.8: Attribute Details Panel for *leaf length:width ratio* attribute.**

These interactive and informational elements in the panel are drawn from the current **specialised domain model**. Only relevant interactive elements are presented. This determination is drawn from data in the domain model. Current status of these elements is displayed in the interaction objects (IOs).

A text edit box for the **attribute** nametag is presented for all **attributes**. The current nametag is present in the text box as user derived names are often similar to original system names, making it usually easier to edit the current name than type from scratch [Nielsen 1993]. Fixed score IO (checkbox) is greyed-out for entry attributes as they are not supported for non-selection type attributes (although it would be logical to extend this IO for these attributes in future work). The preferred unit (pull down) IO is only presented for **attributes** with a numerical entry value domain. Two of the interactive and informational elements of the details panel bear more detailed mention: spatial and relative **modifier** IOs and are discussed in 6.4.4.5.

6.4.3.6 Sub-Attributes

**Sub-attributes** are represented as nodes in the **attribute-value tree**. These are always child nodes of the **attribute** they are related to. Not all **attribute** nodes have **sub-attribute** nodes. Figure 6.9 shows an example of **sub-attribute** nodes.



**Figure 6.9. Sub-attribute nodes on attribute-value tree. Sub-attributes ‘General position’, ‘Relative to axis’, ‘Up/down/inverted’, ‘Hanging/upright’, ‘Inward/outward’ are shown as child nodes of attribute ‘Arrangement: Position: General’.**

**Sub-attribute** nodes have text based on their name. The text format indicates whether there are any **value objects** related to the **sub-attribute** included in the **specialised domain model**. When there are no such **value objects**, the text is greyed-out (e.g. ‘Inward/outward’ in figure 6.9), when there are one or more, the text format is bold (e.g. ‘Hanging/upright’ in figure 6.9). There is no icon on the node as there is no interaction on the **sub-attribute** node. When selected, the node is highlighted, and the details panel (see figure 6.1C, 6.8) reflects the **sub-attribute’s** parent **attribute**.

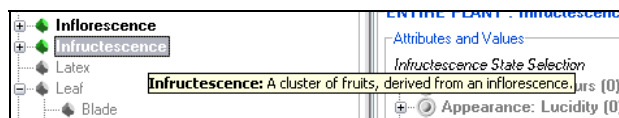
6.4.3.7 Value Objects

The **value objects** related to **attributes** by a value domain relationship are mapped to nodes on the **attribute-value** hierarchy file tree (see figure 6.1B, 6.7) for the relevant selected **description object** node in the linked **description object hierarchy** visualisation (see figure 6.1A). These nodes have a degree of interactivity primarily allowing the user access to the **value object**'s inclusion status for the **attribute**'s value domain in the **specialised domain model**. Selecting the node causes the **attribute** details panel to reflect the **value object**'s parent **attribute**. Informational access to definitions is also linked to the nodes.

The status of the **value object** is shown by various indicators of the node's presentation. Each **value object** node has an icon (○, ●) and a nametag with indicators as in table 6.1.

6.4.3.8 Definitions

Figure 6.10 shows an example of the type of text definitions that are mapped to tool-tips, available on mouse-over the nodes of the various tree visualisations.



**Figure 6.10: Mouse-over definition example.**

Full definitions including multi-media aspects are mapped to pop-up definition boxes (see figures 6.2C, D, E), available upon request from interaction with the representative tree nodes. The user can have as many definition boxes in existence as required, to allow for comparison where necessary.

The definitions explorer tab gives a view of all the ontology based defined terms by name as seen in figure 6.2. Access to the definitions of each term is the same as on tree nodes for consistency.

6.4.4 Mapping tasks to the interface

6.4.4.1 Task summary

Known task	Interaction with presentation element
Search for descriptive element	1. Search tree views 2. Use search bar
Check for existence of desired term in ontology	1. Search definitions explorer pane alphabetically 2. Search for an example descriptive element
Include descriptive element	Relevant tree node ( <i>double click or node menu</i> )
Remove descriptive element	Relevant tree node ( <i>double click or node menu</i> )
Change attribute nametag	Attribute details panel
Fix Score	Attribute details panel
Specify preferred units	Attribute details panel
Specify relational or spatial modifier	Attribute details panel <i>(select modifier from pull down list to access pop-up guided window e.g. figure 6.11)</i>
Define new attribute relationship	Description object node menu (then select <b>attribute</b> from presented menu list)
Check text definition	Mouse-over tool tip on relevant tree node
Check full definition	Definition box accessed from relevant node menu
Include universal description object	Parent description object node menu to access pop-up window (e.g. figure 6.4) and select description object
Clone description object	Parent description object node menu
Change description object nametag	Description object node menu (access pop-up interaction box)
Determine concrete status	Description object node menu
Alter task order	Main menu item and description object hierarchy
Review specialised domain model	1. Description object hierarchy tree 1b. Filtered Description object hierarchy tree 2. Data entry interface 3. XML export
Export/Load specialised domain model	Main menu

Table 6.2: Task Summary

Table 6.2 shows a summary of the major tasks in the specialisation process including elements of the primary task of specifying attributes. Some of these tasks are discussed in more detail below.

### 6.4.4.2 Primary user task

To specify an **attribute**, the user identifies and selects the relevant **description object** node on the **description object hierarchy** tree. This gives the user access to the relevant **attribute-value tree** on which the user identifies the desired **attribute**. Where appropriate, the user can then identify **value objects** to be included in the **attribute's** value domain in the **specialised domain model**. Users can include elements using a standard double-click selection technique that is intended to make the task of specifying quick and easy.

Special cases of identifying the **description object** exist where the node must first be added to the **description object hierarchy** view. These cases include cloned **description objects** and universally applicable **description objects**. These special cases essentially require the user to define a hierarchy relationship that is not explicit in the ontology but is permitted. A similar case occurs when the user has already included an **attribute** for the **description object** that utilises the same base term and now they desire to add another **attribute** with the same type of term (e.g. where a user wants two different measurements of *length*, one for normal measurement and one for a ratio with the *width*). Adding another **attribute** node involves selection from all possible **attributes** that are relevant for that **description object**.

At any point once a **description object**, **attribute** or **value object** has been selected, the user can include them in the **specialised domain model** by interacting with the selected node. Reversing the inclusion of descriptive elements is achieved by interaction with the appropriate nodes in a similar manner to that for including elements.

A number of other specialisation sub-tasks for the selected **attribute** can be carried out by simple interaction with the **attribute details panel** (see figure 6.1C). These include: editing the **attribute** nametag, determining a preferred unit, determining the **attribute** should be considered a fixed score. Two other interaction objects on the **attribute** specialisation panel (figure 6.1C, 6.8) are used for more complex specifying with



## Chapter 6 - Specialisation Process

relational or spatial modifiers. Figure 6.11 shows the pop-up panel that is used for this type of task.

### 6.4.4.3 Search tree

Users can either navigate the tree directly using their domain knowledge to seek objects of interest or they can utilise the search bar to find the **description object** they are interested in. The search bar was introduced following the 4<sup>th</sup> stage evaluation where users were experiencing difficulties finding some terms, particularly for value objects, where the associated attribute was not obvious from domain knowledge. This occurred due to weaknesses in that aspect of the ontology.

The search bar identifies and highlights the first node that matches the search text, other matches can be found by clicking the 'Next Match' button (see bottom portion of figure 1A). This obviously still requires some domain knowledge to have an idea of the term, though it can still help if users are unsure of the exact form of the spelling of the ontology term or where in the **description object hierarchy** it might appear. Whenever a **description object** node is selected, the represented **description object** is placed in the search box to allow users to search for other instances of the ontology term in the hierarchy.

The search bar at the bottom of the **attribute-value tree** panel (see figure 6.1B) can be used in a similar manner as the search bar in the **description object hierarchy** tree to find **attributes** in the **attribute** and value hierarchy tree (see above).

### 6.4.4.4 Searching for existence of terms

In testing, the definitions explorer was rarely used to search for the existence of terms in the ontology. Usually users just searched for the descriptive elements directly, only looking for terms if they could not find the element where they expected to. For example if it was not obvious to the user from their domain knowledge, which **attribute** the **value object** was related to in the ontology or where users assumed a domain term would exist as one type of descriptive element such as a **description object**, but in fact it was being used as another type of descriptive element, such as an **attribute**.

## Chapter 6 - Specialisation Process

In these rare cases, users could find out that the term they were seeking existed but was to be found in a different context. The definitions explorer's nodes included all mapped ontology terms and the definition includes the information on what type of descriptive element it is. The inclusion of a search function for the tree views reduced the definitions explorer usage to virtually non-existent.

### 6.4.4.5 Specify relative or spatial modifier

**Attributes** can be specialised by users using relative or spatial **modifiers** using the attribute details panel (see figure 6.8). The modifiers presented in the selection IO (a pull down list) are drawn from the ontology. Upon selection of one of the specialisation options, a pop-up window is generated, presenting users with further details and steps to complete the specialisation. Figure 6.11 shows an example of one of these pop-up windows for applying a **relative modifier** ('ratio') to a '*number*' **attribute** of the '*Inflorescence: flower*' **description object** to capture the concept of number of flowers per inflorescence. This example is taken from a RBGE taxonomist's specialisation for the '*Alyxia*' group of plants.

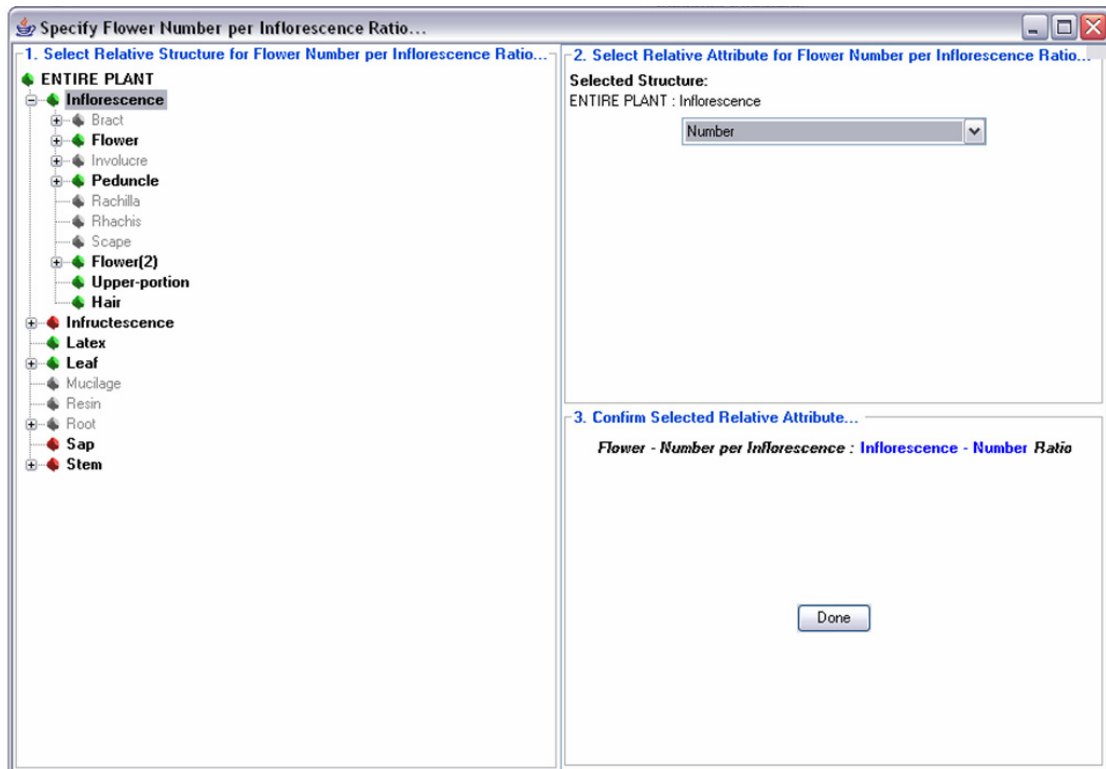


Figure 6.11: Attribute Specialisation Pop-up Window.

## Chapter 6 - Specialisation Process

In the pop-up window, seen in figure 6.11, the **description object hierarchy** is presented in a collapsible tree in the same way as in the **description object hierarchy** in the main window, except that the interactivity on the nodes is restricted to informational definition access and selection of the **description object** target for the **relative modifier** (e.g. in figure 6.11, the selected target is the '*inflorescence*' **description object**). The **attributes** of the selected **description object** are represented as entries in a selection pull down list. The task process of specifying the **relative modifier** is represented with headers giving basic instructions for each of the sub-tasks, with the name of the primary **attribute** and **description object** drawn from the selected elements in the linked main window. These guidance notes were added in response to user difficulties in completing this uncommon task during the 5<sup>th</sup> phase development full task tests. The process of defining the relationship was more complicated than other tasks both in number of steps and conceptually. This taken together with the infrequent occurrence of the task, meant users were becoming confused over how to proceed when they came across the need to define such a relationship.

The spatial modifier pop-up window is similar but only represents the **description object hierarchy** for selecting **description objects** as they are the only type of target applicable in this case.

### 6.4.4.6 Task Model Ordering

The data entry task order is represented in the **description object hierarchy** tree (figure 6.1A). Users can interact with the file tree to effect a limited alteration of the order of the **description objects**. These changes are then reflected in the file tree visualisation.

Users can alter the default data entry order at any time during the specialisation process by altering the position of sibling nodes on the **description object hierarchy** tree. This is achieved by using arrow icon buttons to move selected nodes up or down, till the required position is achieved. Drag and drop techniques are not supported in order to enforce the constraint that users cannot change the hierarchal parent-child structures, merely the order of sibling nodes, without having to issue usability unfriendly 'not permitted warnings' or have users wondering why their dragging worked sometimes but not others.

### 6.4.4.7 Reviewing the specialised domain model

Users can use the tree views with greyed-out and bold elements plus the other indicators of content (see table 6.2) to review the state of the model. Filtering techniques (3.4.2) can be used to get a clearer view of the **specialised domain model**. Users can choose to filter all non-included elements from the visualisations, including the **description object hierarchy tree** and the linked **attribute-value trees**.

### 6.4.5 Domain terms for domain model terms

Domain based terms are captured from the ontology that will substitute in interface for terms such as **description object**, **attribute**, **value object**, etc. In the angiosperm ontology for example '*structure*' is specified as the domain term for **description objects**, and '*measurement*' is the domain term for **attributes** with a numerical entry value domain. The ontology presentation model maps these domain terms to system terms in labels and headings to make the interface more intuitive for domain users.

## 6.5 Specialisation Issues

In evaluating the specialisation process, the analogy of building a taxonomic proforma was utilised to explain to users what needed to be done. This is however not an exact comparison as current working practice does not utilise an ontology based descriptive method. As described in chapter 3, users currently use any term they desire in whatever way they desire as free text, which they may define if they wish. There are no constraints on how to model a descriptive feature (taxonomic 'character'), so for example value domains may or may not be conceived or determined. When using one of the electronic descriptive formats such as NEXUS [Maddison 1997] or DELTA [Dallwitz 1993], the user does determine the domain of possible scores for a feature, but there is no ontology based description for the feature nor any other constraint on what constitutes a feature. Thus while some parallels can be tentatively drawn with current practice in evaluating the specialisation process, too great an emphasis cannot be placed upon any such comparisons.

Without a current system to directly compare with the specialisation process, a number of evaluation criteria were used during testing to develop the system. These criteria were based upon the needs of supporting the working practices of the taxonomist user

## Chapter 6 - Specialisation Process

test group as well as the general aims of the generalised system. These criteria were: having results consistent with the ontology; users being able to specify their descriptive concepts; users able to make informed choices; timeliness of the process; and other usability criteria to support effective use of the interface. These are discussed below.

### 6.5.1 Results consistent with ontology

Whilst it is not possible to quantifiably measure the quality of data captured by the system, one of the important theoretical benefits of the system is producing a set of captured data, that is consistent with a defined domain ontology, thus in principle supporting a high quality of said data. To do this the **specialised domain model** produced by the specialisation process must remain consistent with the domain ontology. The system has been successfully used to produce a number of **specialised domain models** that are consistent with the underlying ontology.

A number of constraints on user behaviour are used to ensure this consistency.

#### 6.5.1.1 Use of ontology relationships and terms

The descriptive elements that users must use for the **specialised domain model** are based on defined descriptive terms from the ontology. No new terms can be added to the **specialised domain model** by users and existing terms cannot have their underlying ontology-based term definitions altered. Primarily users select from elements based on terms and explicit relationships from the ontology. However users are able to add some new relationships to cover such areas as universal description objects which the ontology permits but does not have an explicit relationship for.

Users cannot create new relationships between terms that are not allowed for in the underlying ontology. This constrains users from moving **description objects** to different places on the **description object hierarchy** (other than sibling order changes) as this would involve creating new **description object to description object** primary organising relationships. These relationships cannot be changed as they may be fundamental to domain understanding of the **description object**. For example a '*branch*' as part of a '*stem*' is different to '*branch*' as part of an '*inflorescence*', both of

## Chapter 6 - Specialisation Process

these are supported by the ontology and could be in the **specialised domain model** but '*branch*' as part of a '*leaf*' is not.

Users are equally constrained from adding **attributes** for **description objects** that are not supported in the ontology. Adding such a new a **description object – attribute** relationship could lead to applying such an **attribute** in a context that the ontology designer never intended which could lead to misuse.

Value domain relationships (**attribute – value object** relationships) are used to constrain users from using **value objects** for purposes and in contexts for which they are not intended.

The constraints to stop using the various descriptive elements in contexts that the ontology builder did not intend is important, as the definitions attached to the terms can then be specific to their possible usage context. This allows them to be more accurate, without making the definitions themselves directly reliant on other definitions. It was important that the definitions should be as self-contained as possible, avoiding a highly elaborate definition space as that would more difficult to source than an ontology that did not require such a rich information structure.

### 6.5.1.2 Ontology-based logical consequences

Logical consequences of user actions upon the domain model data structures are enforced by the system to maintain consistency with the ontology. This constraint ensures that when an **attribute** is included in the **specialised domain model** that the parent **description object** is also added. Equally when a **description object** is included, that a path up to the root of the **description object** exists within the **specialised domain model**, including other **description objects** as necessary to ensure this. Including a **value object** of an **attribute** provokes similar behaviour to ensure the **attribute** is included in the **specialised domain model**.

Logical consequences also flow down the hierarchical data structures, ensuring that if a **description object** is removed from the **specialised domain model** that no **attributes** of it nor descendent **description objects** remain in the **specialised domain model**. Similar behaviour is enforced for removing **attributes**. This behaviour must be

## Chapter 6 - Specialisation Process

enforced as the **description objects**, **attributes** and **value objects** do not contain sufficient data in themselves to give their contextual relationships to other objects in the ontology.

Whilst users would be unlikely to wish to perform operations that would result in nonsense data, these constraints remain necessary. The system aims for domain users to edit the **specialised domain model**. Such users however may lack the expertise or knowledge to understand the details of the underlying data structures or the effects of their editing decisions upon the structures.

### 6.5.1.3 Users and consistency

During the user tests, users did exhibit behaviour that attempted to perform operations that would break consistency.

Users exhibited a wish to add new description elements based on new terms to the **specialised domain model**. Some users indicated such a desire to use their own favoured terms in early interviews, but the majority disagreed or accepted the value of only using those from an accepted common ontology (especially if it was their view over what terms to use, that was the accepted view). During later wide tests, a minority of users (15% during the final wide test) made some comment during speak-aloud observation, indicating a wish to add terms when they found they could not immediately find terms they wanted. This was sometimes due to failures of omission in the ontology (especially in earlier tests with an early version of the angiosperm ontology) and sometimes due to failure to find the desired term. In the latter case, being unable to add a new term, usually led the user to find existing equivalent terms that existed in the ontology. Enforcing this constraint does however, make the effectiveness of the system more reliant on the completeness of the underlying ontology being used. Enforcing it also ensures that spurious non-comparable terms are not used where equivalent terms are already available.

Users (7% in final wide test) also attempted to use the facility for changing the order of sibling **description object** nodes to actually move **description objects** to different places on the **description object hierarchy**. In doing so, they were attempting to place the **description objects** in a context, which better matched their own view of the data as

opposed to the ontology builders. Again this was most evident in early tests, with an early version of the ontology, where weaknesses of reliance on the ontology were being exposed.

### 6.5.2 Users able to specify their concepts

The users have concepts they wish to capture data about in their project. One of the basic evaluation criteria is the ability of the user to express those concepts using the specialisation interface and the system's domain model.

Published taxonomic description data was examined to investigate if the descriptive concepts found therein could be expressed in terms of the system's domain model. In this regard the developers of the angiosperm ontology contributed by using the system's specialisation interface as an interactive view of their ontology to ensure their ontology was comprehensive. This gave a degree of relatively thorough testing of the system's model with regard to its ability to capture concepts in the field of taxonomy.

Generally it was found that the vast majority of concepts could be simply expressed as one or more specialised **attributes** of **description objects** in the specialisation interface. There were however a substantial minority that were more difficult to express. These difficult concepts were still mostly capable of being expressed with the aid of cloned **description objects** plus relative or spatial modifiers, including adding extra universal or cloned **description object** nodes to the **description object hierarchy** tree for purposes of referencing them as part of relative or spatial modifiers. Users who were unfamiliar with the system did however find it difficult to specify these complex or non-standard concepts. For this reason, additional guidance was given in the interface for using relative and spatial modifiers. The pop-up windows for adding the modifiers had 3 step instructions for users to follow to specify these modifiers (see figure 6.11 example). A narrow test conducted at the end of development showed this clarified the procedure of specifying these modifiers, although some experience and cognitive thought was still required to translate some characters into specified **attributes**.

Users in the wide tests were also asked about their ability to express their concepts using the system. Although most of these users had only limited experience with the system, 92% of users expressed positive opinions on the ability of the system to express



## Chapter 6 - Specialisation Process

their concepts, believing for example they would be “*able to express one’s self*” and that it “*seems very flexible*”. Whilst the users did only have limited experience with the system, a number of them were very experienced taxonomists who had knowledge of a wide variety of descriptive concepts. None of these highly experienced taxonomists believed the descriptive system (i.e. the domain model) was incapable of expressing their descriptive concepts, though they might require some thought and familiarity with the interface for some of the extremely complicated ones.

### 6.5.3 Informed choices

In order to follow the principle of improving the quality of eventual data collected, it is also important that the user is able to make informed choices when specifying the descriptive data requirements for the project. The specialisation interface thus must support the making of informed choices by insuring the user is aware of the nature of the data elements they are working with and their relationships to each other. This criterion includes ensuring the visibility of system status, effective feedback and access to definitions.

#### 6.5.3.1 Visibility of System Status

Usability literature commonly cites the visibility of system status as a key usability heuristic [Nielsen 1994, Tognazzini 2003, Norman 1988]. Keeping users informed about the the current status of the edited **specialised domain model**, is equally a key enabler for users to make informed decisions.

One of the uses of the **description object hierarchy** tree (see figure 6.1A) is as an overview visualisation of the system status that is always visible. It is a view of the state of the **specialised domain model**, showing which **description objects** are included in the model, whether they have any descriptive **attributes** specified and which of them is currently selected.

The linked visualisation of the **attribute-value tree** of the selected **description object** (figure 6.1B) provides a visible status of the selected description object’s **attributes**. This is visible whilst that **description object** is being worked upon. Although this element of system status is only visible when the relevant **description object** is

## Chapter 6 - Specialisation Process

selected, this is usually the time when such detail is required. At other times, the **description object** node icon summary information of whether any **attributes** are specialised for a **description object** is sufficient for system status.

System status also needs to be up to date to have best impact on usability [Nielsen 1993, Tognazzini 2003], ensuring users are informed to make their decisions. The **description object hierarchy** tree (as well as all other elements of the interface) reflects the current status of the **specialised domain model**, matching this requirement.

Observation of users during the user tests, combined with speak-aloud feedback and post-test interview feedback showed users generally believed they knew the state of the **specialised domain model** and where they were working within it. Users also demonstrated an understanding of the node summary information, particularly commenting upon the distinctiveness of greyed-out elements and noticing that they had failed to add a viable value domain for an **attribute** they had included (based on red warning text/icon).

Visibility of the entire system status was however not always complete. When fully expanded, the overview scrolled off the screen, hiding elements of the model. Elements of the tree could also be occluded because they were contracted. In practice however this made little difference to users, as they only required to have a view on the detail of the immediate area of the **description object hierarchy** around which they were currently working. As the groups of **description objects** that users worked in matched well with the organising hierarchy structure (as observed during narrow user tests, when users were able to select their own working pattern), they did not usually require to view at a glance the details of the status of the remaining elements of the model. Their domain knowledge was sufficient to provide them with a wider context for these **description objects**. While a case for focussing techniques such as fisheyes [Furnas 1986] could be made here to improve the view of the immediate area whilst maintaining a wider context, and briefly tested in early storyboards, however they were not pursued in prototype development as the simpler to implement filter method of tree expansion/contraction was found to be sufficient to maintain a user's awareness of **description object** space, particularly the important sibling, parent and child nodes.

## Chapter 6 - Specialisation Process

This issue emphasises the importance of the primary organising relationship and the size of the ontology's **description object hierarchy**. In the angiosperm ontology, the relationship's match with working practice is good and consequently the occlusion of elements of the **description object hierarchy** is not a serious issue.

### 6.5.3.2 Description space awareness

As well as providing a general overview of the specialised domain model status, the **description object hierarchy** tree has a related system status visibility role in providing users with awareness of their location within the description space. Other cues are given in the **attribute-value tree** and in the various panel headings.

It was observed during observation of the 4<sup>th</sup> phase wide user test, that some users (with less domain experience) were very occasionally becoming confused as to which **description object** they were working with. This was noticed particularly where there were other **description objects** using the same basic defined term nearby in the **description object hierarchy**. These users were not observing the current **description object's** hierarchy context from the tree.

The heading to the **attribute-values tree** displayed details of the current selected **description object** node. Initially, these displayed details were kept to the basic nametag data for reasons of clarity and to minimise length. During development however, the heading was expanded to include the full path data of the **description object** (to show all its parents to the root). To evaluate these alternate headings, a short test with 3 taxonomists was conducted to compare the headings. This test showed an improved awareness with the longer path heading for the users who had shown most difficulty previously and no change, but no complications, for those who had previously shown no difficulty using the hierarchy view as a navigational cue.

### 6.5.3.3 Reviewing the specialised domain model

Linked to the visibility of system status, is the need for users to be able to review their specialisation decisions. Users need to be able to review their work in detail to see what effect their decisions have had, spot possible errors or omissions and determine what still requires to be done. The task options for reviewing the **specialised domain model**

## Chapter 6 - Specialisation Process

are 1. using the specialisation interface views (with filter); 2. previewing the data entry interface; or 3. exporting and viewing an XML file (see table 6.2).

During the final wide user test, users were asked to review the **specialised domain model** that they had specified using the filtered specialisation views and the XML file. Impressionistically, users were pleased with the filtered view, believing they were readily and effectively able to review their 'proforma'. Ignoring the effects of different interpretations of the same domain data, users made a small number of errors in their test tasks, mainly of omission. Of those errors that had not been corrected before the user reviewed the **specialised domain model** at the end of the specialisation process, 50% were picked up in the review with the filtered view (excluding errors due to misinterpretation of the test questions). Error catching is not the only objective of the reviewing task, but does have some indicative role in determining if users could effectively review their work.

Feedback from users, completing the full system task for real data sets of their own, stressed the usefulness of the filtered view for reviewing the content of their work, allowing them to determine relatively quickly and effectively if they had omitted anything of importance. These users were observed on a number of occasions returning to the specialisation process following review, to add or re-edit **attributes** to the **specialised domain model**.

Using the XML file, viewing it effectively as a text file in a web browser, users were not convinced they were able to review their work. No errors were discovered using this method in the final wide user test. Users were observed to be intimidated by the amount of detail and the technical format. In order for the file to be useful for this purpose, it would be necessary to improve the view of the XML, possibly through some form of xml transformation using a user-friendly display template. Users did retain a desire to be able to print out their **specialised domain model** for reviewing and the XML file could form such a basis. One notable contrary indication came from the experiences of an angiosperm ontology developer who was an IT expert as well as an expert biologist. This user used the XML file by preference for reviewing **specialised domain models** and gave feedback indicating they had no difficulty reading the format. They were not a representative end-user however.

## Chapter 6 - Specialisation Process

Although users were not specifically asked to review their work by previewing the data entry screens during the final wide test. Users did find the vast majority of any errors they had made during the data entry task for their first specimen. Additionally users that completed the full system tasks did occasionally use the preview to check another view of **description objects** they were working upon if they had a number of different **attributes**. As the data entry view concentrated on one **description object** per page, this effectively acted as a filtered view of the current **description object's attributes** and values, showing only those that were included in the **specialised domain model**.

### 6.5.3.4 Warning messages feedback

Warning messages are used to inform users of possibly unintended logical consequences to ensure consistency with the ontology. Ideally for usability, the system should ensure the system guides the user with constraints to remove the need for warnings [Nielsen 1993, Schneiderman 1998]. Whilst the system only presents choices to users that are supported by the ontology, removing a descriptive element may have a cascade action on other elements due to the logic of the ontology (see 6.6.1.2). In these cases, warning feedback in the form of confirmation dialog boxes is immediately provided. These messages attempt to explain in clear language the nature of the consequence, using domain specific terms where possible and identifying the descriptive elements involved. Error messages can generally help users understand systems better [Frese 1991] and this could be the case with these warning messages which safely explain consequences that may not be apparent to new or inexperienced users.

The system does not support full undo/redo facilities as usability guidelines would normally suggest [Nielsen 2004, Togazzini 2003], however the significance of this is offset by the ability of users to undo any unwanted inclusion or removal of descriptive elements by using the same procedure as originally used to do it. During the final wide test, 15% of users looked for such undo facilities. In all cases however users were able to recover from their errors by simply reversing the mistaken inclusion decision.

Even with undo facilities, the need for warning message feedback of possible intended consequences would remain, as users would need to be informed in cases where their editing decisions would have a larger impact on the **specialised domain model**, than

## Chapter 6 - Specialisation Process

was perhaps readily apparent in the immediate area of the description space in which they were working. There is however no reason in theory why undo and redo could not also be incorporated into such a system. It was not implemented during development, as it was a usability issue that never received a level of priority sufficient to justify the effort of implementation, especially considering that few research related insights were likely to be gained in a well known field.

### 6.5.3.5 Definitions access

A key concept of the approach for improving the comparability and general quality of collected data is the use of descriptive elements based upon defined ontology terms. This however, is not sufficient in itself. The user needs effective access to the definitions to make informed decisions on about the elements they choose to use.

Users were observed to make substantial use of mouse-over definitions during initial exploration of the **description object hierarchy** tree. This use declined after initial exploration, but users still tended to check the definitions of **description objects** that they were using, usually via mouse-over. Invoking pop-up full definitions of **description objects** was rarely witnessed in testing, but the angiosperm ontology did not have multimedia definitions assigned to the **description objects**, which would reduce the reason to do so.

Users also tended to check mouse-over definitions of **value objects** of interest, to determine if they matched what is desired. Initial interest was gained from their node text (name) and from association with other sibling **value objects** of interest. More occasionally, users were observed to use pop-up definition boxes of **value objects** for comparison or to check multimedia definitions. Most commonly this behaviour was observed in cases where there was multiple similar domain terms (e.g. leaf and petal outline shapes) or where the domain term was open to very variable interpretation (so-called “*woolly terms*” e.g. textures). When comparing definitions, users were observed to use 2 to 4 definition boxes at a time. Their domain knowledge, along with use of the mouse-over definitions was utilised to determine which elements to compare.

**Attribute** definitions were not represented in the angiosperm ontology, with their definitions being just their name. Many of these names had no strong domain basis for

## Chapter 6 - Specialisation Process

them, leading users to be unclear as to what they were. Consequently, users used the value domain of **attributes** to infer a meaning, which was slow and imprecise. This is a result of the ontology model and will not necessarily hold true in other domains where **attribute** definition use could be different.

Use of definitions was generally observed to fall with increased familiarity with the ontology. Users with greatest domain experience also demonstrated a very rapid fall off in accessing definitions, particularly **description object** definitions. This probably represents their improved learning curve for the ontology. These more experienced users (in domain and ontology terms) primarily made use of mouse-over definitions to very rapidly check select descriptive elements, just essentially checking that the definitions were as expected.

Users were generally positive with the access to definitions in the specialisation interface. 77% of final wide test users made positive comments on their access to definitions and their awareness of the definitions. Users were particularly positive about quick mouse-over access to desired definitions.

Alternative facilities for definitions access were considered before and during development. Assigning screen space to an always-visible definition area could in theory increase awareness of the ontology definitions underlying the descriptive elements being used.

To represent full definitions for all three descriptive elements (**description object**, **attribute** and **value object**) being manipulated, would however require a significant investment of limited screen space. Representing only text definitions would reduce screen space requirements, but multiple text definitions would be less readily distinguishable from each other, requiring extra cognitive steps to distinguish between them, thus losing the benefit of having them always visible. Representing only one descriptive element at a time would also reduce screen space requirements, but could cause user confusion over which element was being represented.

Additionally, having a multimedia definition display that was constantly changing as the user navigated the interface could prove to be distracting to users. Finally, a

## Chapter 6 - Specialisation Process

dedicated fixed screen space might be insufficient to display the multiple definitions required for comparison.

Whilst no full comparison test was attempted, users were asked at a number of stages during development for their views on an always visible definition access, including mock-up interface views for three experienced users. Whilst users' opinions are generally not infallible with regards to their own needs, these users did not believe they required constant always visible reminders of the definitions during the specialisation stage. This contrasts with their desire for multimedia definitions during data entry as can be seen in chapter 7. It should be noted however that users familiar with the ontology accessed fewer definitions than unfamiliar users.

The use of mouse-over text definitions and user-requested multimedia boxes was thus determined to offer the best match for the needs of the specialisation interface for reasons balancing: limited screen space; task flexibility; and ease of access. User testing showed this approach to satisfactorily support users and their informed decision-making.

### 6.5.4 Other Effectiveness and Usability issues

There are other important issues of usability that do not necessarily directly impact users' ability to make informed editing decisions, but do affect their ability to effectively use the specialisation interface. The question of how to manage some of these usability heuristics in an ontology-based system is discussed below.

#### 6.5.4.1 Autonomy

A degree of autonomy for users is often seen as a good usability principle [Tognazzini 1992]. The specialisation interface is designed to give users' freedom of action to explore the ontology and complete their tasks in an order of their choosing. This flexibility supports variable working practices within taxonomy as well as between different domains.

In practice during all testing phases, users generally specified one entire **attribute** at a time and specified all the desired **attributes** of one **description object** before moving



## Chapter 6 - Specialisation Process

onto another **description object**, returning to previous **description objects** only if they discovered they had missed something or made a mistake. During the wide user tests, one user however, preferred to first add all the desired **description objects** to form their specialised **description object hierarchy**, and then add the desired **attributes**.

During the wide user test, it was observed when asked to add characteristics from their own knowledge that they might wish to score in their own work, that 31% of users navigated towards a **description object – value object** combination, with the system only adding the relevant **attribute** by inference when they included one of its **value objects**. They then looked for alternative **value objects** to add. This behaviour is likely due to the differing concepts of characters (see 2.4.1). The users in these cases were conceiving their ‘character concepts’ as *structure-state* combinations.

Imposing a rigid task structure on users would force users into working practice that they might find unnatural and at odds with their mental conception of the data. Providing no boundaries for users however does not equate to users feeling free either [Tognazzini 2003]. Our approach uses boundaries constraining users to selection or defining relationships supported by the ontology.

### 6.5.4.2 User expectations

It is widely agreed that users are able to make better use of visualisations and user interfaces when they match the users’ expectations [Tognazzini 2003]. The users’ expectations relate to the both their mental model of the data and their expected tasks. Where there is dissonance between the visualisation and the users’ expectations, users can become confused and disengaged from the process, leading to extra potential for errors and an unwillingness to adopt the approach.

The current practice of proforma creation is used as a metaphor for the specialisation task. This leads users to expect the repeated task of creating characters to add to their proforma. Specifying a character does not however always map 1:1 to the system task of specifying **attributes**, depending upon where conceptual character boundaries are drawn. It can take multiple **attributes** of the same or multiple **description objects** to specify the one character concept. Despite this, the metaphor was still valuable when

## Chapter 6 - Specialisation Process

explaining the system. Users in speak-aloud results repeatedly made this analogy. Users in other domains may not have a similar task analogy to draw upon however.

In order to match the user's mental model of the data, the presentation model can draw upon domain terms from the ontology to use in the interface for various system terms. In the angiosperm ontology, this allows the interface to use terms such as *structure* instead of **description object**, *state* instead of **value object**, *measurement* instead of **attributes** with quantitative value domains. Informal feedback showed that this compared well with interfaces that just used system terms.

The selection of a relationship from the ontology to use as the primary organising relationship that has real-world meaning is important to match users' mental model of the data. The **description object hierarchy** is the primary view of the ontology and it is important that the users can relate to it in a real world sense. The part-of relationship in the angiosperm ontology does this very well. Users were able to relate to this '*structure hierarchy*' as a real world abstract super-plant, with the specialised hierarchy being a representation of a fully complete project *specimen* (or abstract *specimen*, where two alternative *structures* could not exist at the same time in the real world such as '*inflorescence*' and '*infructescence*').

One reservation mentioned by experienced users was that whilst the view was generally "*intuitive*", the order was off-putting as it did not match the domain standard (acryptic ordering). This demonstrated another reliance of the interface on the underlying ontology for suitable domain knowledge to maintain its metaphors, as the angiosperm ontology did not contain such ordering information for mapping.

Ontology relationships that restrict combinations of descriptive elements that do not make domain sense aid in matching user's expectations of the data. Users reacted poorly to being presented with options in the specialisation interface which did not make real world sense. This was particularly noted in the **description object** hierarchy relationships and in **value objects** that were inappropriate for certain **description objects**. For example, a number of users spontaneously commented during the final wide test that some **value objects**, based upon domain terms for shapes normally used to describe flowers, were being presented in the value domains for '*general shape*' **attributes** of **description objects** other than '*flower*'. Users found this off-putting, for

## Chapter 6 - Specialisation Process

whilst in the real world they would be used along with other general shapes for flowers, they would never be used in other contexts. The **concrete domain model** however can only match these real world restrictions when there are suitable relationships to map from the ontology.

Consistency of user expectations is another aspect of matching users' expectations. Usability heuristics suggest that usability is improved where users can expect consistent behaviour from an interface. The specialisation interface has a number of different types of descriptive elements represented by various tree nodes. Interaction with these nodes has however a level of consistency with them. They react similarly for example with right-click menus for primary interaction, double-click to include / remove from the specialised domain model, mouse-over definitions, greying-out to indicate inclusion status in the **specialised domain model**. These heuristic lessons were reinforced during development when differing behaviours were linked to double-clicking nodes in the two main interactive tree visualisations. In one tree double-clicking controlled expansion/contraction of the tree and in another it also controlled inclusion within the **specialised domain model** for leaf nodes. Whilst these behaviours were considered most useful for individual nodes, it generally resulted in users being observed to become confused as to what would happen if they double-clicked a node, and thus reluctant and hesitant to explore the interface. Being consistent in the behaviour gave users more confidence to use the interface. In the final wide test all users responded positively to being positive about general interface interaction behaviour. Reservations on that confidence were made by 45% of users when asked about the behaviour for adding relative and spatial modifiers. Notably, that behaviour used a different paradigm for behaviour that was not consistent with other behaviour (an unavoidable consequence of the nature of the task).

Consistency in user expectations also includes consistency with platform standards [Nielsen 1993]. Users learn an interface better if it matches the behaviour they are used to in any other computer system they are used to. In the case of the RBGE taxonomist users, they were mostly familiar to a greater or lesser extent primarily with Windows type applications. Most users had some familiarity with file trees, which helped them quickly grasp the basic mechanisms for navigating the main tree visualisations of the specialisation interface. One exception to this was double-click behaviour, where a minority of users sometimes used double clicks to select nodes and were initially

## Chapter 6 - Specialisation Process

surprised to change the inclusion status of the descriptive element. Fortunately the very visible change from grey-out to black text and coloured icon alerted users to this effect. These users were able to learn to avoid this behaviour relatively quickly and it had no catastrophic effects due to suitable confirmation warning messages when there were consequences beyond the current node. The beneficial effects of speed and ease of use for experienced users were observed to be far more significant.

### 6.5.4.3 Navigability of description space

A basic requirement for users to be able to make efficient and effective use of the specialisation interface is their ability to navigate around the description space to find all the descriptive elements of interest to them. The primary instrument of such navigation is the **description object hierarchy** tree, along with the linked **attribute-value tree**.

Efficient view traversal (EVT) is a prerequisite of the navigability of a visualisation. This involves the efficiency of viewing, selecting and moving to nodes to form a path through the structure. Furnas [1997] identifies two aspects for this efficiency. Firstly, the diameter of the view should be low, requiring short paths, in terms of the number of links to navigate from one node to another node in the structure. Secondly, the out-degree of nodes should also be low.

Trees generally have good efficiency, particularly balanced trees. The balance of the **description object** tree depends upon the ontology and the choice of the primary organising relationship when the ontology is initially mapped to the **abstract domain model**. The angiosperm ontology with the `part_of` organising relationship does not result in a balanced tree, but does have reasonable diameter and average out-degree. As the user can add nodes, a definitive maximum diameter and out-degree cannot be given. Before such editing, the angiosperm **description object hierarchy** tree has a maximum diameter of 11 and maximum out-degree of 25. As the tree is not balanced the average out-degree is however much lower and more reasonable (in the 2-6 range). The linked **attribute-value trees** have a good diameter (maximum 4) but a high maximum out-degree (43, again with a much lower average) due to the relatively large number of **attributes**, some of which have a large number of child **value object** nodes. The child **value object** nodes are split up by **sub-attributes** to improve navigation by reducing

## Chapter 6 - Specialisation Process

out-degree, but the ontology's support of suitable sub-groups of values to act as sub-**attributes** is not consistent. This is another area where the initial mapping must look for suitable relationships to map to the **sub-attributes** to improve out-degree.

The expansion and contraction interaction of the tree visualisations improves EVT by improving the diameter, as users can effectively focus the view, including high level nodes from other parts of the tree in the viewing window.

EVT does not take into account ensuring the users' node selections are informed and reasonable to get to a desired target. To have effective view navigation users need also to be able to find the shortest path to a desired target, without error and in a history-less fashion. Systematically labelled rooted trees generally meet these requirements if their hierarchical structure makes logical semantic sense to users [Furnas 1997].

In the angiosperm case, the part\_of structure relationship is conceptually familiar and logical to the domain users. The outlink-info at each node can thus be relatively small, being the semantic labels of each child node. These semantic labels, connected to a domain meaningful hierarchy, ensure that each node has good residue at every other node, allowing domain users to navigate to a desired target from any other node by the shortest path, without using trial and error. Users can use domain knowledge to determine which link their desired target destination lies in the direction of: a child node or the up-link that is implicitly everywhere not in the direction of one of the child nodes. Without such meaningful domain hierarchy related semantics, the visualisation would need to enumerate all possible target nodes down each link, which would be impractical.

The **description object hierarchy** tree and the **attribute-value tree** are not totally history-less as the domain meaning of any node is dependent not just on its label info, but upon the context of the path to the root. This is not an uncommon situation in many classification hierarchies. The context information is not available to users on the node itself, but is available by the user's understanding of the context in which they are working, by looking up the tree and by reminder in the heading labels of the tree panels. This factor in theory may slow down navigation if users require to visually check the context info at each determination stage of their navigation. Speak-aloud observation

## Chapter 6 - Specialisation Process

testing however showed users rarely checked the context info, maintaining the context cognitively whilst actively navigating to desired target in the tree.

These factors in theory point to the **description object hierarchy** tree and **attribute-value tree** having effective view navigability. Observation and feedback of users backed up this theory. Speak-aloud testing clearly showed users to be using domain knowledge to navigate the trees, using residue scent at the nodes to determine whether their desired target was likely to lie below child nodes or to track back up the up-link.

User feedback during the two wide tests showed users to be positive as to their ability to find desired objects within the **description object hierarchy** by using the tree and their domain knowledge. The most enthusiastic users for utilising the tree believed that the tree structure was “*intuitive*” and “*liked the ability to move around the structure hierarchy*”. These users corresponded with the most experienced taxonomists, suggesting that domain knowledge was being effectively harnessed for navigation. Observation supported this correspondence with domain experience, in confidence in using the **description object hierarchy**. Likewise, the users who were least confident in their feedback were the users with the least angiosperm domain knowledge.

The **attribute-value tree** was not as enthusiastically received but still showed reasonable navigability. These trees were smaller than the **description object hierarchy** tree and its navigability less critical. Users with most experience of the angiosperm ontology were able to navigate using this tree without major difficulty. The majority of test respondents, however, encountered some difficulties as the hierarchy was seen as partially artificial in domain terms, particularly with regard to the assignment of **value objects** to the value domains of **attributes**. Users did not always find the information scent of the **attribute** names useful for navigating to some **value objects** they wished to use. During the final wide test, it usually took users 1-3 attempts to correctly identify the **attribute** that a desired **value object** target lay beneath. Following incorrect information scent at **attribute** nodes was a primary contributor of time costs or failure in seeking **value objects**. 15% of users in the 2<sup>nd</sup> wide test expressed that they found finding desired **value objects** difficult. Naming of **attributes** was based on the angiosperm ontology’s groupings of states, which did not always have a well-founded domain term for the concept of the group. Attempts at changing the ontology grouping names and hence the resultant **attribute** names to be more intuitively

## Chapter 6 - Specialisation Process

had some limited success. This highlights the reliance of the interface upon having accepted ontology hierarchy relationships for the descriptive elements and term names that have good information scent for navigating the various hierarchies. Where there is controversy over the domain relationships or terms represented in the ontology, it is reflected in possible difficulties for effective navigation of the interface's tree views.

Both major tree visualisations had search boxes attached as discussed in the presentation model section. These search boxes could circumvent and support the use of the trees for navigation. They were observed to be particularly useful when users were inexperienced in the ontology or in domain terms. Less experienced users who utilised the search boxes were notably faster at reaching a desired goal than those who did not. 31% of users in the final wide test generally used search boxes in preference to the trees for navigation. To use a search box does require that users know what their desired goal is beforehand however. Some users were able to use search boxes to let them find one desired **value object** and then use exploration to discover other **value objects** of interest due to their sibling relationship on the **attribute-value tree**. Likewise some users used the search box to find a **value object**, in order to discover which **attribute** was relevant to it. The search boxes are thus able to shortcut tree navigation to a desired target, particularly for inexperienced users and provide some contingency where ontology relationships weaken the navigability of the trees.

### 6.5.5 Time

#### 6.5.5.1 Impact of time taken

The time question is a particularly important issue for taxonomists' working practices (see chapter 2), with some users cutting corners in current proforma building to save time despite the fact that this could cause problems with their collected data. Whilst users may concede that a time cost may in theory be reasonable to improve the quality of their data, it must be remembered that the main direct beneficiaries of the improved data quality are often not the person who collected the data. By improving the clarity and comparability of collected data, other users are likely to directly save time in comprehending and using the data, rather than the originator of the data, who believes they know what their concepts are.

## Chapter 6 - Specialisation Process

The specialisation process is only completed once for a given project, compared to the multiple times the data entry for a specimen is likely to be completed (usually between twenty and several hundreds). The specialisation process is therefore unlikely to bear a high proportion of the time cost for data collection.

The impression of being time efficient is however perhaps as important as actual time taken to encourage adoption of a system aimed at improved data quality by users who have a working culture that believes it is under severe time constraints. Impressions of the time costs of the specialisation process are particularly relevant, as users will notice the burden of new tasks such as this structured data specification. It should be kept in mind that the current proforma creation is a procedure that can seem to take a small fraction of time, if the proforma is loosely conceived and task elements such as conceptualisation of character concepts involve 'off-duty' cognitive thought.

The time cost of the specialisation task was investigated both empirically and impressionistically during the user tests and expert reviews. Metrics were not highly emphasised in testing, as exact costs of time will vary depending upon the complexity of the ontology and extent of the specialisation required for particular projects. Users may also mix the specialisation task in practice with their project knowledge gathering, which does not lend itself well to empirical measurement. Some empirical time testing was however done as an indicative exercise to give an idea of the actual costs of time. As there is no direct equivalent system for comparison, no empirical timing comparisons were attempted. Paper-based proforma building in current practice was used as an approximate metaphor for the specialisation process, but the nature and details of the tasks were so different as to render comparison meaningless. Other user feedback and observation gave impressionistic evidence as to the real and perceived cost of time.

### 6.5.5.2 Empirical Tests

During the final wide user test, users were timed in specialising the angiosperm ontology for a project based on the *prunus* group of plants. (See Appendix E for details of methodology.)



## Chapter 6 - Specialisation Process

Specialising a representative part of the ontology (on the basis of a given list of commonly utilised terms for *prunus inflorescences*) resulted in 15-18 **attributes** being specified (variation due to differing interpretations of the data and errors by users).

The time cost was measured at 16 - 34 minutes with a mean of 22.8 minutes [ $\pm$  6.07 minutes SD] for 11 users of varying domain experience who were mostly unfamiliar with the system. (The same task given to 2 users who were very familiar with the system and ontology was measured at 9 - 12.5 minutes.)

The number of specialised **attributes** that are required for an average project is difficult to determine. This is due to wide variation in the amount and detail of data required for different types of project. To give an idea of the numbers likely to be involved however, a complete **specialised domain model** for the prunus data as prepared during development by the taxonomist who helped create the test data list contained 38 specialised **attributes**. Other complete **specialised domain models** created by taxonomists based on real world data, during development, had between 37 and 218 specialised **attributes**.

These results, whilst not a comprehensive empirical study, are certainly indicative that the specialisation process can be completed in a timely manner. The cost of time for specialising the domain model for the *prunus* group can be compared with the cost of time for the data entry task based on that **specialised domain model** for 1 specimen (mean cost of 16.8 minutes). It can be seen that the specialisation task's contribution to the cost of time of the project data collection is likely to be of low significance if the data collected on that specialisation is from a large sample of specimens.

### 6.5.5.3 Impressions of times

Users were asked about their impressions of the time taken for the specialisation process in the final wide user test and full task tests. The respondents all expressed neutral to highly positive views on this subject, which compares well with the respondents concerns during the qualitative research study (chapter 2).

One very experienced user who was very familiar with using the DELTA electronic file format (see chapter 2) said that this system was "*much faster than DELTA*". That user

## Chapter 6 - Specialisation Process

identified the following major reasons for the difference:- ease of use of the specialisation interface; DELTA's rigid circular tasks as opposed to our autonomous approach (6.6.4.1); and requiring to define all terms used (if definitions were desired) upon each use in DELTA.

### 6.5.5.4 Cost of time variables

The cost of time will vary depending upon a number of variables, most clearly the complexity and size of the ontology. During early tests with an early version of the ontology lower time costs were observed in navigating a much simpler and smaller **description object hierarchy** for the obvious reasons of lower diameter (i.e. less steps being required to navigate to targets).

From the observation of users in the second wide and other narrow tests it was clear that it took significantly longer to navigate in **description object hierarchy** in complex deep areas which had repeated sections of the tree. Users in these cases took extra time to ensure they were in the correct section of the tree.

Specialising value domains could also take a long time where it was not obvious from scent of **attribute** names as to where desired **value object** nodes were. There was a noticeable difference in this regard between users who made wide use of the search bars and those who did not. Those who did use the search bars were generally quicker at finding both problematic **description objects** and **value objects**. However, those who did not use them believed they would speed up as they became familiar with the ontology, especially as regards the **attribute to value object** relationships.

A weak correlation of the users' taxonomic experience and the cost of time were found on the basic costing data with the significantly longest time taken (30 mins) by the user who was least experienced with angiosperm plants (excluding users who were distracted from the time tasks by articulating their views on the semantics of the represented ontology).

Familiarity with the ontology and the system were seen to reduce the cost of time fairly significantly. This was seen in the time costs of such users in the final wide test and more generally from observations of users during development. Users also commented

## Chapter 6 - Specialisation Process

during the wide test that they felt they were starting to find it quicker to find ontology elements of interest due to familiarity even by end of the test (1 hour).

### 6.5.5.5 Time cost breakdown by task

Seeking the relevant **description object** node is a necessary first step, including checking definitions as required to find a **description object** that matches the user's concept in the correct location in the **description object hierarchy**. The time cost varies significantly depending primarily on the number and complexity of nodes that the user requires to traverse from their starting point. As users do not go back to root for each **attribute** to specialise, working semi-logically through the hierarchy, they do not have to traverse great distances on the tree as often. By supporting such working practice in our approach this aspect's cost is minimised.

Seeking the relevant **attribute** node only requires traversing one level depth of the **attribute-value tree**, as they are always found on the second highest level. The breadth must be searched however and the out-degree of the top level nodes can be very high. This is thus usually a simple sub-task. However where information scent of the nametag is weak, users may have to check definitions and possibly potential value domains, particularly where they are less familiar with the ontology. Note that this is effectively optional for selection type **attributes** as the user may solely rely on seeking the related **value objects**.

Seeking the relevant **value object** nodes includes traversing the **attribute-value tree** as well as checking and possibly comparing alternative definitions. Definitions are most likely to be investigated in detail during this stage, contributing to cost of time particularly for users who are less familiar with the ontology. In the angiosperm ontology the **attribute-value tree** is shallow and wide (or otherwise unbalanced), with leaf nodes representing the **value objects** (a high out-degree from a number of **attribute** and **sub-attribute** nodes), which decreases the speed of traversal, as a relatively large number of child nodes must be searched one by one. The burden would be lessened where the ontology provided sufficient **attribute** to **sub-attribute** to **value object** relationships with useful domain based information scent names, to form a balanced tree, reducing the out-degree of the nodes.

## Chapter 6 - Specialisation Process

Other specialising of an **attribute** can incur a significant cost of time. This cost of time can theoretically be off-set against reduced cost of time for data entry in cases such as changing nametags, specifying preferred units of measurement and specifying fixed values.

A notable cost of time is incurred when relationships must be defined instead of selected from the presented ontology. Complex **attributes** involving adding **description object** nodes and relational or spatial **attribute** modifiers can take up significant multiples of the cost of simple ones. This is a consequence of the ontology trying to cover all eventualities but not being able to be explicit about them. Ontologies with less universal **description objects** and less relational or spatial modifiers would theoretically have a lesser cost of time to specialise.

### 6.6 Conclusion

This chapter has described the process by which domain experts specialise an ontology to specify the **specialised domain model** and the default task ordering of the **data entry task model**. The specialisation tool consists of an interface that is system generated based upon an **ontology presentation model** which acts upon the **specialised domain model**.

Our approach to the use of domain ontologies is shown to rely upon the nature and quality of those ontologies. Ontologies such as the one for angiosperm description are unusual in that taken as a whole it does not capture knowledge of a concept, but controls how a concept can be described. The compositional description hierarchy derived from the ontology cannot describe any real world entity, as it includes all descriptive possibilities. To use such an ontology to improve data quality requires that a domain user specialise it in order to use it to control consistent data entry for a project. This fits well with the need of taxonomists to ensure consistent basis for specimen data collection in their projects, while ensuring the data is comparable across projects.

Given the nature of the ontology, it can be presented as a basis for selection of a suitable proforma ontology (**specialised domain model**). The use of selection for identifying desired elements of the domain model is a strength of the approach, with no user entered text problems, low time costs and ease of use through techniques such as

## Chapter 6 - Specialisation Process

double-click inclusion. The selection concept was also very easy for users to conceptualise. Where the specialisation process was difficult for users was in circumstances where there were weaknesses in the ontology or where users had to define description relationships themselves. Where users had to define relationships, their impressions of ease of use were reduced and more guidance was needed particularly in the more complex of these cases. The **ontology presentation model** consequently displays as much of the materialised mapped ontology as possible, including all the explicitly defined relationships. Implicit **attributes** are also displayed for selection to ensure consistency with the explicitly defined **description object – attribute** relationships. In order to maximise the display of the description hierarchy for selection, we adopted a two linked views approach with two collapsible trees that provide good visibility of the elements important for taxonomic working practice by only presenting **attributes** and value domains when the relevant **description object** is being worked upon. More complex paradigms such as fisheye views were not required as the collapsible tree generally provided adequate views of the immediate description hierarchy context in which users were working. Extending the collapsible tree with summary indicators of relevant attribute data is also sufficient for seeing system status at-a-glance. Other extensions in support of working practice such as automatically expanding attribute nodes to view the value domain when they are included to the specialised domain model contribute to the usability of the selection process.

The reliance of the specialisation process on a good ontology which matches the users' expectations of the data was noticeable in testing. The *structure part-of* relationships in angiosperm ontology was a good match as the primary organising relationship, permitting users to navigate the **description object hierarchy** tree using their domain knowledge. Users with more domain experience were consequently better at this task. By comparison, **attributes** were poorly grounded in accepted domain terms and consequently difficulties were experienced by users in identifying **attributes** and **value objects** for inclusion.

Simply producing a specialised domain model consistent with the ontology is not sufficient in itself to have a good basis for producing an appropriate and effective data entry interface. Definitions from the ontology provide users with an understanding of the descriptive elements they are using and of how the data eventually produced will be interpreted by others. To harness the benefits of definitions effectively, text definitions

## Chapter 6 - Specialisation Process

need to be short enough to appear as tool tip text on mouse-over to support the presentation model's consistent quick access method. On-demand multimedia definitions provided deeper understanding and clarity when required, particularly for comparison of similar terms

Finally, the specialisation process relies on the domain users who effect the constrained editing. Despite the ontology based constraints on their modelling, the approach has to trust the users to make sensible decisions, as the ontology constraints do not foresee every circumstance and do give users some freedom to express their descriptive concepts. The user requires a level of domain knowledge to effectively complete the specialisation process to know what descriptive concepts they wish to include. To this extent, users must be considered to be domain experts. Users of the specialisation process do not need to be IT specialists; they only require basic computer skills to use the interface. Users learned the basics of the interface very quickly and informal observation suggested they could recall the knowledge after a break of some weeks, which is useful for a process that is likely to be sporadically used.

The involved models that resulted from an iterative development process have been described and discussed. The developed interface and process was tested successfully at number of stages and at increasing levels of complexity. Full detailed tests were successfully completed with a narrow user group. More artificial sample tests were successfully completed with the wider user group during the 1<sup>st</sup> and 2<sup>nd</sup> wide user tests. **Specialised domain models** that were consistent with the imported ontologies were generated for real and artificial data sets. The following chapter looks at the data entry process.

## Chapter 7

# Data Entry Process

### 7.1 Introduction

This chapter discusses the data entry process and system generated data entry interfaces. In the data entry process, the user instantiates a number of plant specimens by entering descriptive data about them in a data entry interface. This interface is system generated using a **data entry presentation model**, which acts upon the **specialised domain model** created by a domain expert during the specialisation process (see chapter 6). The end result of this process is a series of output files, each containing the **specialised domain model** instantiated for a specimen. These files can subsequently be used to transfer the data to a database based upon a mapping similar to that made to input the original ontology.

This chapter will first look at the domain, presentation and task models used in the specialisation process as developed during the 4<sup>th</sup> and 5<sup>th</sup> development phases (see 5.5), and then discuss various issues concerning the process that arose during the evaluation and testing of the system.

### 7.2 Domain Models

The data entry interface operates on the **specialised domain model** described in previous chapters. This section will only cover extra elements of that model required to accurately capture the data entered by users.

#### 7.2.1 Specimens

Each specimen being described requires a unique identifier to allow the system to track the instantiated data values for each specified **attribute** in the **specialised domain model**. The system provides such an identifier when users indicate they desire to

## Chapter 7 - Data Entry Process

instantiate a new specimen, although users can override the identifier with one of their own choosing to match domain or other practical considerations.

One XML file is exported for each instantiated specimen containing the instantiated portions of the **specialised domain model**. This file can be exported to a database or used to reload the data for the instantiated specimen back into the system for further data entry.

### 7.2.2 Multiple values

The domain model uses a system object called an **instance-score** to record instantiated values for an **attribute** of a specimen. This is straightforward when a user enters a single value for a specialised **attribute**. The domain model must also be able to distinguish and record accurately cases where users enter multiple values for an **attribute**.

#### 7.2.2.1 Ranges

Multiple values can form a range for **attributes** with a numerical entry value domain, where the **description object** is being described abstractly.

However, ranges do not apply to **attributes** with value constraints restricting entry to selection from defined **value objects**. It was decided not to allow **value objects**, of a value domain, to form ranges, as this would require a continuum. Primarily this decision was made in order to protect the independence of the definitions and to retain clarity of entered data. The ordering of such a continuum would be a source of highly subjective opinion. Take for example various outline shape **value objects** from which some taxonomists initially believed they could form a range. It quickly became apparent from informal exercises that different users were unable to interpret accurately the meaning of each other's ranges as users were not forming the same mental 'natural' continuum. The difficulty in doing so can be seen from considering how one would order a series of normal shapes such as square, rectangular, circular, oval, triangular. One could start with the shapes using straight lines and move to the more circular, or one could order based on the number of points, their symmetry, etc. Published descriptions were



## Chapter 7 - Data Entry Process

observed to use such qualitative ranges in current descriptions, however taxonomists were unable to interpret those ranges unambiguously.

If the **value objects** in a value domain were part of a continuum, users would need to select the whole continuum for inclusion in the **specialised domain model** or comparability would not be possible. Take a very simple example with 6 **value objects** in a continuum: A, B, C, D, E, F. User #1 only includes A, D, F in their **specialised domain model** and then records a range D-F for a specimen. User #2 includes B, C, D, E and records D-E in data entry for a specimen. At first glance these describe different real-world values, however consider that user #2 does not have F as a possible point and hence may have recorded E as the closest point. In fact they may be describing the same real-world range. More subtle and complex examples can be readily conceived, introducing an area of uncertainty to the interpretation of recorded data.

Anyone interpreting the results would always need to see the whole set of **value objects** that had formed the value domain to understand the meaning of the range. Essentially a range of **value objects** would only be shorthand for an enumerated set of **value objects**. Given the subjective nature of any ordering and the need for every point along a continuum, it was considered to be better to constrain users to explicitly selecting all **value objects** that were applicable to a specimen. This also fits with the domain model assumption that each **value object** definition is self-contained and is not reliant on other **value objects** in order to interpret its meaning clearly.

Numeric values do have a well understood and unambiguous range mechanism and so are supported. An **instance-score** of a numeric **attribute** can thus contain two numeric values, which represents a range or one value, which represents a singular value.

When a **description object** is being recorded concretely (see chapter 6 for concrete description status details), ranges are not considered suitable, as individual real-world **description object** instances are being recorded.

### 7.2.2.2 AND/OR

Apart from numeric ranges, cases involving multiple values are referred to as **AND-ing** or **OR-ing**. Both cases can apply to **attributes** with a value domain of **value objects**.

It is necessary to distinguish between these circumstances. Namely, where multiple values are applicable to each of a number of real-world instances of the **description object** in the **high-level concept (AND-ing)** as opposed to the situation where the **attribute** has different values on different individual real-world instances of the **description object (OR-ing)**. For example where a specimen has petals that are white and purple as opposed to a specimen whose individual petals are either white petals or purple petals. The permutations of this situation can be quite complex. Multiple values for an **attribute** that are **AND-ing** are contained within one **instance-score**. Multiple values for an **attribute** that are **OR-ing** are contained in separate **instance-scores** that are linked to the same **high-level concept**.

### 7.2.3 Concrete description objects

**Instance-scores** are grouped together for each concrete **description object** instance (see 5.3.4.3), using a sequential numerical identifier. As they only refer to one real-world instance, **OR-ing** is not permitted for **attributes** of a concrete **description object** instance.

### 7.2.4 Modifiers

Various descriptive modifier terms can be mapped from the ontology to the domain model. These modifiers come in modifier groups (see figure 5.2) that restrict how they can be applied. Some of these modifiers have allowed relationships to multiple **description objects** and/or **attributes** (relative and spatial modifiers in the angiosperm ontology). These are applied to **attributes** in the specialisation process (see 5.3.4.4).

Other modifiers are applied to an **attribute** by users during data entry. These groups may be restricted as to which **attributes** they can apply to based upon concrete status. In the angiosperm ontology, modifier groups ‘locator *modifiers*’ (e.g. ‘*at/on apex*’); ‘frequency *modifiers*’ (e.g. ‘*rarely*’) and ‘qualifier *modifiers*’ (e.g. ‘*approximately*’) fall into this category. The cardinality of applying modifiers from modifier groups is derived from the mapped ontology. By default if not stated in the ontology, only one

## Chapter 7 - Data Entry Process

modifier from each group is permitted to apply to any **instance-score** of an instantiated **attribute** for reasons of simplicity and to avoid over-use of modifiers.

These modifiers aim to provide users with an ability to qualify their descriptive statements without resorting to free text entry. This helps improve accurate communication of a user's descriptive observation in a consistent manner and avoid loss of such data. However, the value of such modifiers for any form of automated comparison using a database is doubtful.

### 7.2.4 Not scored statement

The domain model has provision for each specialised **attribute** to be marked as **not-scored**. This allows the data entry user to make a positive statement that although an **attribute** has not been instantiated for the specimen, this has been done for a reason and not simply due to an error of omission. No other data is recorded for an **attribute** when the **not-scored** statement is recorded.

Taxonomic specimen descriptions are beset by inconsistent recording of characters, with characters recorded for some specimens but not others, leaving later interpreters uncertain in the omitted cases whether the character was not present, not the same as other explicitly recorded cases, not able to be measured due to specimen condition, not interesting enough to the recorder or simply omitted in error. By providing the facility for a positive statement, the data can be interpreted with greater clarity.

### 7.2.5 Description object presence attribute

**Presence** is a special **attribute** that was added to the model during development. This **attribute** is included in the **specialised domain model** for every **description object** that has any included specialised **attributes** and is treated in a special manner by the **data entry presentation model**. **Presence** has a value domain with **value objects**: 'present'; 'absent'; 'not scored'; 'no comment'. When an ontology is imported, the system attempts to map an ontology 'presence' **attribute** (and relevant **value objects**) to the system **presence attribute**. If an ontology-based term can be mapped, then the exported descriptions will utilise it. If no mapping can be formed, the system will use a default in-built **presence attribute**.

The **presence attribute** is recorded for all **description objects** with specialised **attributes**. When the **description object's presence attribute** is recorded with the **value object** 'not scored', no other data is recorded for its specialised **attributes**. When the **description object** is recorded as 'absent', no other data is recorded for its **attributes** and all **description objects** below it in the hierarchy are also recorded as absent.

Enforcing the recording of presence aims to improve the clarity of the data. Similar to the reasons behind the 'not scored' statement (see above), this eliminates interpretative uncertainty when no data is recorded for a **description object**. It also ensures that the logic of the **description object hierarchy** relationships is reflected in recorded data.

In addition to the automatically included presence **attribute**, users can specialise and include any ontology based presence **attributes** that would normally be supported by ontology relationships. These user specialised presence **attributes** do not have the same consequences, as their specialisation may have altered the concept so that the underlying logic no longer applies.

### 7.3 Data Entry Task Model

During the data entry process the user instantiates a number of plant specimens, using the general repeated task of entering descriptive data for a specialised **attribute** in a data entry interface.

#### 7.3.1 Task Order

The user can control the task order, but the default ordering is to enter all data for a specimen before entering data for the next specimen. The user modifiable **data entry task model** controls default task order within a specimen. The **data entry task model** uses a depth-first enumeration of the **description object hierarchy** for the default order. Users can specialise the order of siblings in the specialisation process (see chapter 6). Within each **description object**, **attributes** are ordered alphabetically.

### 7.3.2 Attribute instantiation

The repeated task of instantiating **attributes** involves selecting or entering data, adding any applicable modifiers and controlling alternative ‘**OR-ing**’ data. Where the **attribute**’s value constraints are selection from a value domain of **value objects**, the user selects from representations of those **value objects**. The user selects as many **value objects** as required to accurately instantiate the **attribute**. Otherwise the user uses text entry facilities to enter data on the state of the **attribute**, within the value constraints of value-type (text, numerical, etc). Modifiers are added as appropriate. When abstract **description object** instance data is being recorded and alternative values for the **attribute** instantiation are present on different real world **description object** instances, users need to indicate this.

A special presence case of **attribute** instantiation exists for every **description object**. All **description objects** are initially marked as ‘present’ by default, so users only interact if the situation requires this be altered.

## 7.4 Presentation Models

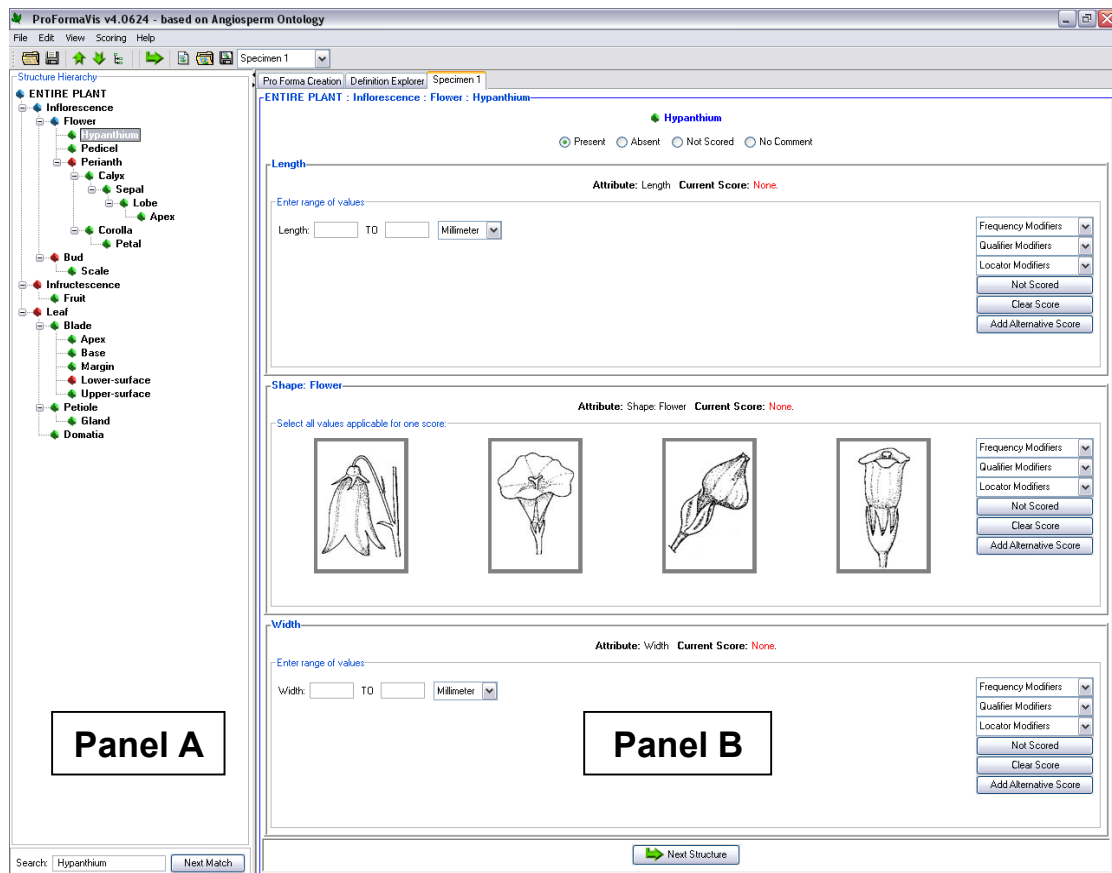
### 7.4.1 Two presentation models

Two presentation models acting upon the **specialised domain model** generate a data entry interface. Figure 7.1 shows an example of a data entry interface. The interface is composed of two main elements. The **ontology presentation model**’s **description object hierarchy** tree (such as figure 7.1A) acts as navigation control and overview of the current specimen. The **data entry presentation model** controls the second main element: grouped data entry panels for instantiating the **description objects** (such as found in figure 7.1B). A knowledge-based strategy of one **description object** per window is used to automatically identify this element. This type of strategy for the automatic identification of windows can be automated and is effective in cases such as this where the general task is known and embedded in the system [Bodart 1995a].

The **data entry presentation model** was developed through iterative development with RBGE taxonomists. It is conceptualised that the system could use different **data entry presentation models** for different domains, although the developed model is conceived

## Chapter 7 - Data Entry Process

as having a good degree of generic application (see chapter 8 for testing in another domain).



**Figure 7.1: Data entry Interface example based on the angiosperm ontology specialised for the ‘prunus’ group of plants. Panel A shows the description object hierarchy tree from the project’s specialised domain model and representing the instantiation state of specimen #1. Panel B shows the data entry IOs for instantiating the description object ‘Entire Plant : Inflorescence : Flower : Hypanthium’ for specimen #1.**

### 7.4.2 High-level concept

As shown in figure 7.2, the main data entry interface panels (**description object hierarchy tree** and data entry panels for instantiating **description objects**) represent a **high-level concept** instance (a plant specimen).

A **high-level concept** navigation IO exists on the main menu bar. The unique identifier for the **high-level concept** is displayed in this IO and repeated on the **description object** instantiation panel heading. This pull down list is populated by all **high-level concept** instantiations that are currently loaded on the system. Users can select these instantiations and the **high-level concept** instantiation in the main interface will reflect

## Chapter 7 - Data Entry Process

the selection. The main menu bar also allows users to add new **high-level concepts** to the system for instantiation. This generates a pop-up dialog with a suggested unique identifier, which can be overwritten by users. Other meta-data about the **high-level concept** can also be entered at this time. Users can load or export **high-level concept** instantiations using a main menu item or menu bar shortcut.

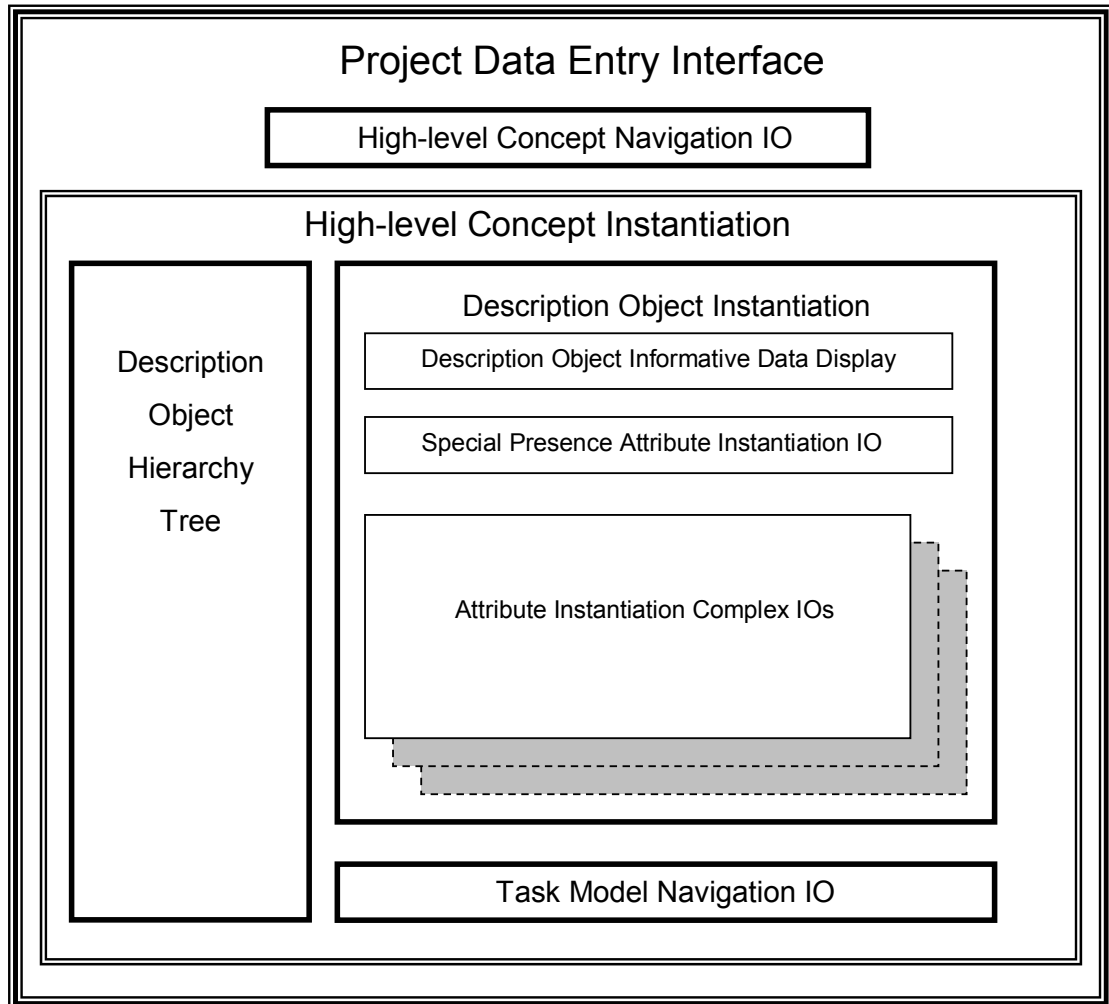


Figure 7.2: Data Entry Interface components

### 7.4.3 Description object hierarchy

The **ontology presentation model** uses the same tree view as is used in the specialisation interface to represent the **description object hierarchy**. Figure 7.1A shows an example of this file tree visualisation, figure 7.2 shows its relation to the other interface components. During the data entry process, the hierarchy is normally filtered to only show the **specialised domain model** rather than the whole ontology. The **data entry task model's** default task order is represented in the ordering of the hierarchy

view. This view remains the same for all **high-level concepts** using the same **specialised domain model**, except that its icons reflect the instantiation state of the current **high-level concept**.

### 7.4.4 Description object

**Description objects** are represented by nodes on the **description object hierarchy** tree and by the linked **description object** instantiation panels. The **description object** nodes are as described in chapter 6, although the node icon colour also reflects the instantiation state of the **description object** (blue when all **attributes** have been instantiated for current high-level concept). Whilst engaged in data entry, the node's editing interaction with the **specialised domain model** is disabled. Informational interaction (definition access) remains available on the node.

Each **description object** with **attributes** for instantiation is represented by a **description object** instantiation panel. Figure 7.2 shows these panels contain an informational element about the **description object**, a special **presence attribute** interaction element and a series of **attribute instantiation IOs**: one for each of its **attributes**. A coloured border with a **description object** identifier heading identifies the description object instantiation. Figure 7.1B shows an example of a **description object** instantiation panel for a '*hypathia*' **description object** of *specimen #1*.

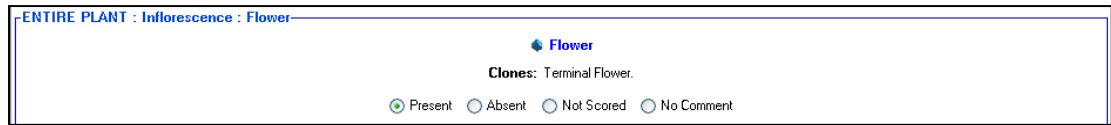
The informational element drawn from the **specialised domain model** (see figure 7.3) includes the **description object** nametag (the full path of **description object** nametags to the root is given in the panel border heading) and a summary icon (replicas of the node icons in the **description object hierarchy** tree).

If the **description object** has any clones, these are displayed by nametag as seen in figure 7.3. The clone information is provided to ensure users are aware of the existence of the clones, to ensure they do not begin data entry for the wrong **description object**. This was found to be especially necessary where only one of the clones was likely to be present on a specimen. The nametag for clones was most useful if it had been either specialised by the user or the system. By default the system simply includes a bracketed number after the original name for clone nametags. If the clone has a **fixed value** as many do, then the system will attach the name of the fixed **value object** to the original



## Chapter 7 - Data Entry Process

name to give a more meaningful name. An example of this is seen in figure 7.3, where the clone of flower has been named '*terminal flower*', reflecting the fact that the clone has a fixed value '*terminal*' for its '*arrangement: position*' **attribute**.



**Figure 7.3: Example of a description object informative data display for a description object with clones. This example is taken from a data entry interface for the 'Alyxia' plant group. The description object 'Entire Plant : Inflorescence : Flower' has a clone called 'Terminal Flower' which represents another distinct kind of flower description object to be found on 'Alyxia' plants.**

Where a **description object** is being instantiated as concrete **description object** instances, the **description object** instantiation panel representation refers to one concrete instance.

### 7.4.5 Navigation task

Users normally work within one specimen at a time, instantiating the **attributes** of one **description object** at a time using the default task order. To move from one **description object** to the next, the user clicks on the 'Next Description Object' button in the task model navigation IO (or on the main menu). The system then selects the next **description object** in the task order that has **attributes** to instantiate (the selected node on the linked **description object hierarchy** tree reflects this). There are however occasions when the user may wish to alter this default order.

Users can use the **description object hierarchy** tree to select **description objects** for instantiation. Selecting a node on this tree causes the linked **description object** instantiation panel for that **description object**, to be displayed.

Where **attributes** of concrete instances of **description objects** are being instantiated, users normally instantiate all **attributes** of one concrete instance before moving onto the next one. Once one concrete instance is completed, the user can select to instantiate a new concrete instance (using the 'new concrete instance' button in the task model navigation IO) if they desire or move to the next **description object** (see above). Users can however choose to work the **attributes** in any order if they desire. To navigate

## Chapter 7 - Data Entry Process

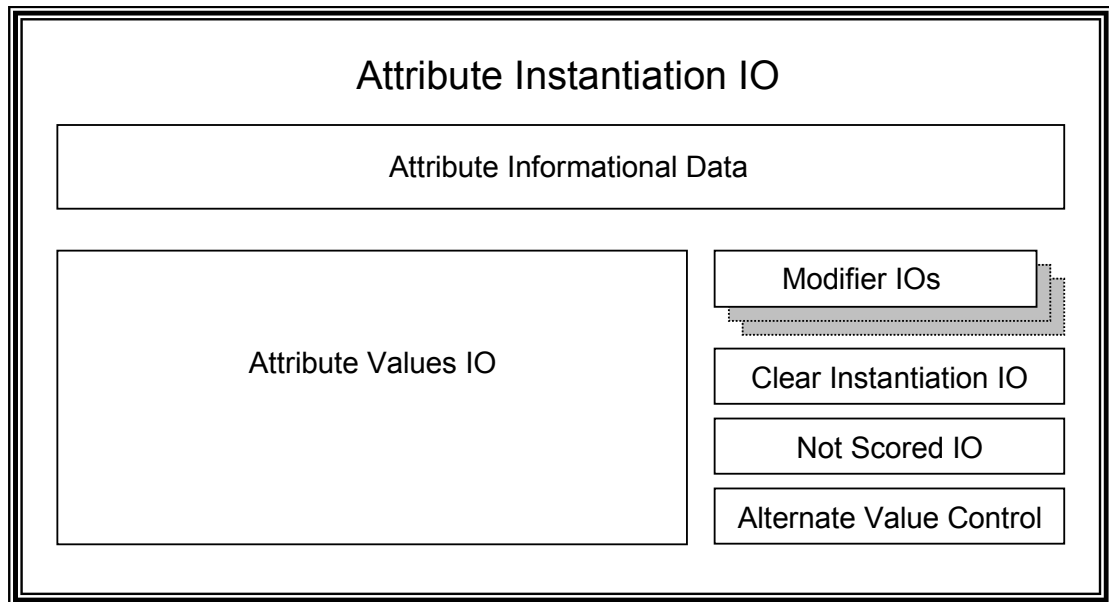
between concrete instances, users use a pull down list of all available concrete instances in the task model navigation IO. The choices include all concrete instances that have been generated for this **high-level concept's description object**. An extra choice is entering abstract data in addition to any concrete data. This allows users to enter both abstract and concrete data about a **description object** if desired. In some cases while users may want to record concrete data for some **attributes**, they might only want to record abstract data about other **attributes**.

### 7.4.6 Attribute

Each specialised **attribute** of a **description object** is presented for instantiation in a complex interaction object. This interaction object contains the data and interaction capability required to enter data for one **attribute**. The implementation of this interaction object varies. The abstract implementation is determined by the data entry task model, which selects abstract interaction objects (AIOs) from a system library. Based on the selected AIO, a concrete interaction object is generated by the system using the relevant **attribute** and related data from the **specialised domain model**.

#### 7.4.6.1 Abstract Interaction Objects

The AIOs are designed to group together the **attribute** specific data and all the interaction capability needed to instantiate the **attribute**. The grouping of these elements is identified by a 3D-effect border with bold coloured text heading. Relevant **description object** data is presented by the parent **description object** instantiation panel and the linked **description object hierarchy** view. Figure 7.4 shows the general template for the **attribute instantiation IO**, used as a basis for all the AIOs in the **data entry presentation model's** library.



**Figure 7.4: Attribute Instantiation Complex Interaction Object.**

The general layout, grouping border, **attribute** informational data display, ‘clear instantiation IO’ and ‘not scored IO’ elements are common to all AIOs in the library.

The display of **attribute** informational data represents the **attribute** name (along with any relative modifiers applied in the specialisation stage) and the current ‘score’. The term ‘score’ is used in the AIOs developed for taxonomy, as it is less IT-specific than ‘instantiation’ and is familiar to domain users. It was also assumed that the term would be comprehensible in general terms. The current ‘score’ always reflects the current instantiated data. The system includes ‘and’, ‘or’, **units** and **modifier** terms as well as the basic instantiated values, placing them so as to form a basic natural language interpretation of the current ‘score’. The **attribute instantiation IO** displays the details of its current instantiation with coloured text (red for none, bold blue for instantiated data) as an additional indicator of the instantiation state.

There is no separate ‘commit’ task to enter data on an **attribute**. The domain model reflects the current state of data entry at all times.

#### 7.4.6.2 Modifiers

Modifiers are given one selection IO (a pull-down list) per modifier group in all AIOs. The domain model applicability of modifier groups varies however, depending upon concrete status (e.g. in the angiosperm ontology, the ‘*frequency*’ modifier group is not

## Chapter 7 - Data Entry Process

applicable to concrete **description object** instances), affecting which modifier IOs are presented. The functioning of the modifier groups IOs, with regard to the selection of multiple modifiers, matches the cardinality of the modifier group in the domain model.

### 7.4.6.3 Alternate Values

An alternate value control (figure 7.4) is also presented in all AIOs except those concerned with concrete **description object** instances. An ‘Add Alternate Score’ button adds an alternate duplicate set of IOs for entering alternative instantiations of the **attribute**, as can be seen in figure 7.5. The alternate set is still contained within the original **attribute instantiation IO** and shares the same **attribute** informational data, including the current score display which reflects the whole instantiation including all entered alternatives. There is no limit to the number of alternative scores that can be added.

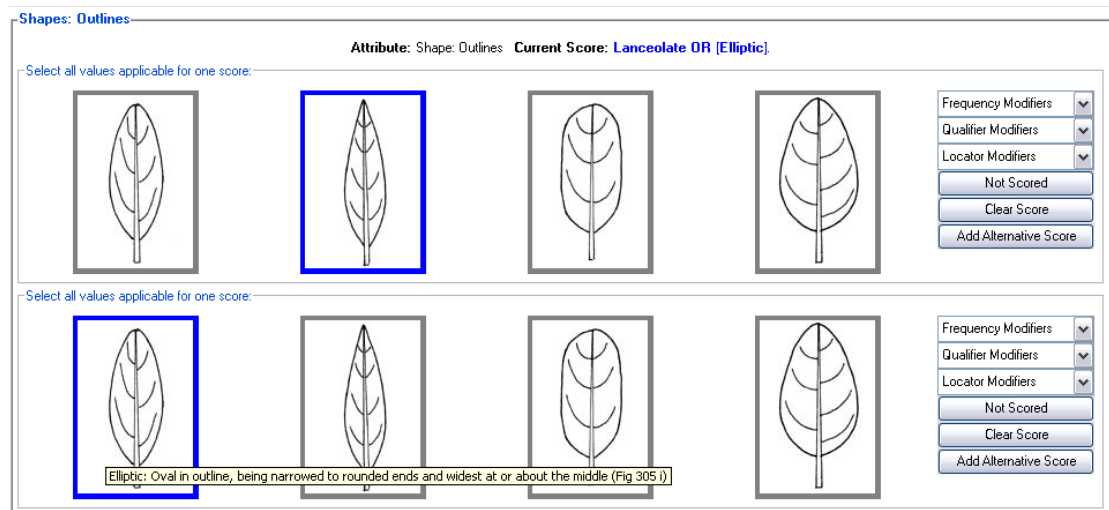


Figure 7.5: Alternate instance-scores example.

### 7.4.6.4 Attribute values IO

The **attribute** values IO (figure 7.4) implementation varies in the different AIOs. The AIOs include implementation controls such as internal layout management (e.g. the layout strategy for **value object** representations) and data entry instructions. **Attribute** values IOs have a set of short instructions for data entry in their headers (e.g. “Select all values applicable for one score” or “Enter range of values:” or “Enter value for concrete instance #2:”). These instructions vary for concrete instances as opposed to abstract

## Chapter 7 - Data Entry Process

ones. This backs up the design of the actual entry IOs (e.g. text boxes, checkboxes) in providing feedback to users on how to enter their data in the IO.

### 7.4.6.5 Numerical entry

In the AIOs designed for user entered data, the entered data is checked, to ensure that it meets any appropriate constraints, most notably only numerical data. Numerical AIOs also have a unit of measurement IO. Examples of concrete interaction objects based on these types of AIOs can be seen in figures 7.6 and 7.7.

The screenshot shows a window titled "Length". At the top, it displays "Attribute: Length" and "Current Score: 3.2 TO 6.4 Millimeter". Below this, there is a section labeled "Enter range of values". This section contains a text input field with "3.2", a "TO" label, another text input field with "6.4", and a dropdown menu currently set to "Millimeter". To the right of these inputs are three dropdown menus labeled "Frequency Modifiers", "Qualifier Modifiers", and "Locator Modifiers". Below these are three buttons: "Not Scored", "Clear Score", and "Add Alternative Score".

**Figure 7.6: Attribute instantiation IO example for an abstract numerical attribute. These IOs support ranges as discussed in the domain model section earlier.**

The screenshot shows a window titled "Width". At the top, it displays "Attribute: Width At Base" and "Current Score: None". Below this, there is a section labeled "Enter value for concrete instance 4:". This section contains a text input field and a dropdown menu currently set to "Millimeter". To the right of these inputs are two dropdown menus labeled "Qualifier Modifiers" and "Locator Modifiers". Below these are two buttons: "Not Scored" and "Clear Score".

**Figure 7.7: Attribute instantiation IO example for a concrete numerical entry attribute. These IOs support only single value entries and do not support ranges as discussed in the domain model section earlier.**

The use of an 'incrementer' widget was considered for the small number of cases where value constraints only allowed a relatively small range of integers. It was decided not to add this to the library and selection strategy as the basic numerical AIOs were

## Chapter 7 - Data Entry Process

acceptable and it was preferable to maintain consistency in order to avoid having users spend time determining how the IO worked and why it was different.

### 7.4.6.6 Value domains of **value objects**

In AIOs designed for selection from a value domain, the **value objects** are represented as selection choices. Users can select any number of these representations to instantiate the **attribute**, as required. Three basic **value object** representations are utilised: checkboxes with text labels, picture buttons and pull-down list menu items. Examples of the two most common representations can be seen in figures 7.8 and 7.9.

The screenshot shows a web-based interface for selecting value objects for the attribute 'Form: Inflorescence'. The current score is 'None'. The interface includes a header with the attribute name and score, a sub-header 'Select all values applicable for one score:', and a grid of checkboxes with labels: Acephalous, Corymbose, Fasciculate, Monocephalic, Racemose, Scapose, Solitary, and Umbellate. A mouse-over tooltip for 'Monocephalic' is visible, showing the definition: 'Monocephalic: One-headed or scapose, as in dandelion'. On the right side, there are three dropdown menus for 'Frequency Modifiers', 'Qualifier Modifiers', and 'Locator Modifiers', and three buttons: 'Not Scored', 'Clear Score', and 'Add Alternative Score'.

**Figure 7.8: Attribute instantiation IO example with value objects represented by checkboxes and text labels. Mouse-over definition of one of the value objects is visible.**

The screenshot shows a web-based interface for selecting value objects for the attribute 'Shape: Flower'. The current score is 'Tubular'. The interface includes a header with the attribute name and score, a sub-header 'Select all values applicable for one score:', and four pictorial selection buttons showing different flower shapes. The fourth button, representing a tubular flower, is highlighted with a blue border. A mouse-over tooltip for this button is visible, showing the definition: 'Tubular: With the petals partly united to form a tube.'. On the right side, there are three dropdown menus for 'Frequency Modifiers', 'Qualifier Modifiers', and 'Locator Modifiers', and three buttons: 'Not Scored', 'Clear Score', and 'Add Alternative Score'.

**Figure 7.9: Attribute instantiation IO example with value objects represented by pictorial selection buttons. The attribute has been instantiated with one of the value objects ('tubular'). The selected value object's nametag and definition are displayed via mouse-over.**

**Value objects** which have been selected are represented by a blue border for pictorial representations. Text labelled checkbox representations are noted as being selected by the condition of the checkbox.

## Chapter 7 - Data Entry Process

Feedback during development suggested that the order of **value objects** was not important to users if they did not represent a continuum. The order of the **value object** representations is thus arbitrary.

### 7.4.6.7 Definitions

The ontology based definitions of **value objects** are displayed upon mouse-over of the **value object** representations in the **attribute instantiation IOs**. Examples of this can be seen in figures 7.8 and 7.9. Additionally where **value objects** are represented pictorially, the pictures are drawn from the multimedia definitions, making the task of checking its definition part of the normal data entry process of looking at the available selection choices.

On rare occasions where the extra definition detail of the definition box is required, users can return to specialisation mode (by clicking on the appropriate main interface tab). There they will still be focussed on the same **description object**, and can check the definition through the **value object**'s node representation in that interface.

### 7.4.6.8 AIO selection strategy

In order to select an appropriate AIO, the **data entry presentation model** accesses various defined criteria of the underlying **attribute** data. The criteria include value-type, value domain size, availability of multimedia definitions for **value objects**, pictorial importance, **description object**'s concrete instance status. Figure 7.10 shows the selection criteria and logic used in the developed presentation model.

Chapter 7 - Data Entry Process

Value constraints: entry constraint	value-type:	<i>abstract</i>	AIO (text box)		
	<b>text</b>	<i>concrete</i>	AIO (text box, concrete)		
	value-type:	<i>abstract</i>	AIO (numeric, range)		
	<b>numerical</b>	<i>concrete</i>	AIO (numeric, concrete)		
Value constraints: value object domain	$v = 0$	<i>no representation for attribute</i>			
	$0 < v < 10$	$p = 0$	<i>abstract</i>	AIO (checkbox)	
			<i>concrete</i>	AIO (checkbox, concrete)	
		$p > (v-2)$	<i>abstract</i>	AIO (picture)	
			<i>concrete</i>	AIO (picture, concrete)	
		$0 < p < (v-1)$	$pi = true$	<i>abstract</i>	AIO (picture)
				<i>concrete</i>	AIO (picture, concrete)
			$pi = false$	<i>abstract</i>	AIO (checkbox)
				<i>concrete</i>	AIO (checkbox, concrete)
		$v > 9$	<i>abstract</i>	AIO (pull-down list)	
		<i>concrete</i>	AIO (pull-down list, concrete)		

Figure 7.10: AIO selection strategy

Notes:

'abstract' & 'concrete' refer to the concrete status of the **description object** instance.

$v$  = size of **attribute**'s value domain.

$p$  = number of **value objects** with multimedia definitions in the **attribute** value domain.

$pi$  = **attribute**'s pictorial importance (default=true).

7.4.6.9 Layout of **Attribute** IOs within **description object** instantiation.

The layout of the **attribute instantiation IOs** within the **description object** instantiation panel is controlled by a 'place one below the other strategy'. More complex layout strategies would be more difficult to automate without necessarily providing additional benefits. To contain all the appropriate interaction and data was likely to require the remaining screen space width, so other layout strategies would be limited if the **attribute** IOs were to remain so self-contained.



### 7.4.6.10 Special Presence **Attribute**

This **attribute** is represented differently from other **attributes**, using the special presence **attribute instantiation IO** (see figure 7.2). The **attribute** appears under the **description object** informative display and has no representation of the **attribute** itself other than the value domain. It uses radio buttons (with text labels) to represent the **value objects**, as the **attribute** has a cardinality of 1. Definition access on the **value objects** is as standard for the interface, by mouse-over.

### 7.4.6.11 Sub-attribute

**Sub-attributes** are not represented in the **data entry presentation model**. They are used to improve the navigability to **value objects** in very large value domains in the specialisation process. If very large value domains were common in another domain, and suitable **sub-attributes** existed, they could be represented in a new AIO.

## **7.4.7 Supporting task: review instantiation**

There are two tasks for reviewing the instantiation of a **high-level concept**. The first is a simple check to ensure all **attributes** have been instantiated (or marked as not scored). This involves scanning the **description object** node icons on the **description object hierarchy** tree, to see if there are any that indicate the **description object** has not been fully instantiated. A second check of the actual instantiation values can be made by paging through the various **description object** instantiation panels for the current **high-level concept**, checking the displayed values or by exporting the instantiation and viewing the separate XML file. The exported instantiation XML file was found to be too complex for domain users to use, being designed to capture the data for export to a database. Another simplified descriptive text file was thus designed and users could export that ‘viewable description’ from the main file menu to review their instantiation.

## **7.4.8 Specimen details**

In capturing taxonomic description data for a specimen, some details of the specimen are generally captured. The basic data entry interface allows for the capture of a unique identifier for a **high-level concept** but not other specimen meta-data. To record the other specimen meta-data, the unique identifier IO was expanded to capture a number of

## Chapter 7 - Data Entry Process

other specific fields. A more robust and logical method of capturing such meta-data would however, be to treat such data as another root **description object**, with **attributes** and other data elements. These could be captured from the ontology, mapped, then be specialised as normal and captured in the **specialised domain model**. The presentation model could then present them as with any other **description object**.

### 7.4.9 Changing the data entry presentation model

It is envisioned that the **data entry presentation model** could be altered by IT experts for different domains. Whilst the model developed during the project is considered to be useable for general domains, it was designed for taxonomy. It is possible in theory to design different AIOs to add to the library and to input a new AIO selection strategy. It would also be possible to change the layout strategy of having one **description object** instantiation panel per page. The general interface layout and architecture is currently fixed within the system and would require further programming to alter. This is discussed further in chapter 8.

### 7.4.10 Domain terminology

As with the **ontology presentation model**, the **data entry presentation model** uses domain terms from the mapped ontology in place of system terms where available (e.g. '*structure*' instead of **description object**, '*specimen*' instead of **high-level concept**).

## 7.5 Evaluation

It is difficult to measure data quality effectively in order to evaluate whether the data entry process improved it, especially when the data is subjective based upon the user's interpretation of real-world entities. Artificial data, real legacy and specimen data have however been successfully collected using this process, then exported to a database. These data were consistent with the original angiosperm ontology, with defined terms and consistent data structure.

Data quality is promoted in this process through the effects of constraining user behaviour to enforce ontology restrictions, especially where that results in selection of values rather than text entry. These constraints ensure the data has more defined terms,

## Chapter 7 - Data Entry Process

improving clarity; more comparable terms and data structure, improving comparability within and across data sets; less errors from text entry. Restricted selection rather than free entry, as well as the effects of defined choices in avoiding scoring drift (especially if pictures are used) improve consistency of data entry. Informed users are also necessary to have good data quality, the interface attempts to provide users with an understanding of the context of their data entry, for example via the **description object hierarchy** context and by making them aware of definitions where helpful.

A number of issues all need to be addressed in assessing the effectiveness of the data entry interface and its associated models. To do so the remainder of this section will address firstly whether users are able to express their data entry concepts, accurately recording what they wish to describe. Secondly, it will address whether these users are informed as to their data entry decisions. Lastly various issues affecting the efficiency of the process, particularly as relating to the effectiveness of the **data entry presentation model**.

### 7.5.1 Expressing data entry concepts

During testing, users were able to use the data entry interface to record the details of the specimens they were attempting to describe, based on **specialised domain models** they had detailed in the specialisation process. The ability of users to express their descriptive concepts in data entry obviously depends on their specialisation of the domain model (as described in chapter 6), but also on the ability of the system's domain model to capture the nuances of the data (e.g. multiple values). Generally, users believed they had been able to enter descriptive data which accurately described their observations of the specimens in the final wide user test (85% expressed positive opinions of this with 15% neutral and no negative opinions, although 18% of the positive opinions were conditional on the availability of modifiers). Full task test observations and informal user feedback backed up these findings, with users able to capture new specimen based data from various plant groups as well as legacy data on the '*alyxia*' group of plants [based on Middleton 2000, 2002].

## Chapter 7 - Data Entry Process

### 7.5.1.2 Use of ontology terms & data structure restrictions

Users are constrained to using the ontology terms included in the **specialised domain model** for a descriptive concept (expressed as one or more specialised **attributes**). Users showed no difficulties with working in the context of the structure of the domain model, as it was already understood from their earlier specialisation task.

Omissions in early versions of the angiosperm ontology were discussed in chapter 6. Such omissions had a less immediate effect upon data entry except where users' used workarounds for missing terms in the ontology. For example users used the modifier 'not' to modify the **value object** 'pubescent', a measure of hairiness, to effectively score smooth surfaces, as the domain term with this meaning ('glabrous') had initially been omitted from the ontology. The use of such workarounds is a concern for data quality and reinforces its dependency on the comprehensiveness of the underlying ontology.

Some users did question during testing whether it would be possible to select from **value objects** that were supported in the ontology for the **description object** and **attribute** in question, but which had not been included in the specialised **attribute** value domain. Whilst the **value object** definitions are assumed to be independent and thus should be able to be added, the idea of having the specialisation process is to ensure that data entry is consistent for each specimen. This consistency could not be guaranteed if users were permitted to enlarge the value domain beyond the specialisation at will. The intentions of the specialisation user cannot be second guessed at data entry stage, as there may be good reasons why they have not included some **value objects** permitted by the wider ontology, for their more specialised project subject. If the specialisation user is the same as the data entry user and has merely overlooked an option, they do have the facility to return to the specialisation process and add that option in. This behaviour was occasionally observed in the full tests, particularly during the entering of data on the first 1-2 specimens.

### 7.5.1.3 Use of multiple values

In order to record their descriptive concepts accurately, users need to distinguish between AND and OR multiple values. This question ignores the issues of users being

## Chapter 7 - Data Entry Process

able to make such a real-world distinction in their interpretation of the real-world specimen, which is a domain issue.

Users were observed to understand the distinction between **AND-ing** and **OR-ing** within the interface, including how to express those concepts. Using speak-aloud methodology, users were observed on a number of occasions expressing real-world descriptive concepts that equated to **AND-ing** or **OR-ing**. They were subsequently observed accurately expressing those concepts using the interface's AND/OR multiple value facilities. Feedback in the **attribute instantiation IO**'s informational data display was observed to serve as a useful check for users to ensure they had accurately expressed their AND or OR concepts.

Some users did require initial explanation on how to express alternate value concepts (**OR-ing**) in the interface, but were able to express that desire and following a brief explanation had no observed difficulties in subsequent use of the facilities.

### 7.5.1.4 Use of concrete **description object** instances

Only limited testing with concrete instances was possible due to time constraints, however a limited narrow test at the end of the 5<sup>th</sup> phase showed experienced users were able to comprehend and utilise concrete instances without observed difficulty.

### 7.5.1.5 Use of measurement units

Although generally straightforward to use, there were incidents during the user tests that gave cause for concern regarding the capture of numerical data with units of measurement. Some users were observed to enter numerical data without indicating what unit of measurement they were using. This applied to 15% of users during the final wide test. These users had not entered any preferred units during the specialisation process and were simply working within what they regarded as standard practice, assuming the units would be understood although they had not indicated them.

The mapped ontology can note preferred units for **attributes**, these can be overridden during specialisation or data entry as required. The angiosperm ontology does not include this information however, so it must be entered by users during specialisation, to be represented in the data entry interface. During user tests, more experienced users

## Chapter 7 - Data Entry Process

normally used the preferred unit facility, especially if they were intending to actually enter data for a number of specimens based on the specialisation. However that was not always the case, and one user was observed wasting a lot of time repeatedly entering units of measurement at data entry, because they had rushed the specialisation process, failing to set preferred units. A default ontology preferred unit that could be overridden would avoid this issue if a standard could be found.

Whilst there are occasions where no units of measurement are required, there should be some feedback to users that they should enter units for some **attributes** (e.g. ‘*length*’, ‘*height*’). The system would need to rely on the mapped ontology for the knowledge about whether an **attribute** should have units to properly implement such directed feedback.

### 7.5.1.6 Use of not scored mechanism

The not scored mechanism was observed during testing to be widely used where **attributes** could not be recorded due to the real-world state of the specimen. A number of users (22% in the final wide test) spontaneously commented upon the facility in a positive manner during speak aloud observation, appreciating the ability to make a positive comment that the **attribute** was considered but could not be scored.

During a narrow user test, users attempted to capture legacy data about the ‘*alyxia*’ group of plants. When capturing the legacy data, the not scored mechanism was useful for cases where the legacy descriptions omitted data or could not be clearly interpreted.

A similar positive reaction was achieved regarding the special presence **attribute** (with one experienced taxonomist for example commenting that it was “*very useful information to have*”), where users used the facilities to mark **description objects** that were not present on a particular **description object** as absent.

### 7.5.1.7 Use of modifiers

The ability to give extra qualitative statements about the data they recorded was seen as very important or helpful by a substantial minority of test users. Whilst they would not be likely to be useful in automatic database comparisons, they were seen to be important for added clarity regarding descriptive observations. To one user they were “*essential*”

## Chapter 7 - Data Entry Process

to accurately record their concepts. Most users however made only some occasional or rare use of the facility.

There was some concern that the use of some modifiers, specifically the ‘locator’ type modifiers (e.g. ‘*at/on base*’, ‘*at/on upper surface*’), would cause users to use these modifiers as an alternative to more accurate specifying of their descriptive concepts in the specialisation process. Instead of resorting to a locator modifier, users could specify the **attribute** for all the possible locations it could be observed in (or they could use a relative spatial modifier to relate the **attribute** to the other **description object**). Locator modifiers all referred to a universally applicable **description object**, such as ‘*base*’ or ‘*upper surface*’ that could exist as the child of any other **description object**. By using a locator modifier, the data might not be comparable with data that used the **description object hierarchy**. **Attribute** data for one specimen using locator modifiers might also not be comparable with data for another specimen if they had different locators. This would not necessarily be obvious if later comparisons were made with the data, as the comparison might ignore modifiers on the basis that they are primarily only useful for extra clarity of human interpretation.

There are however possible solutions to these issues. The locator modified data could be converted into more rigorous **description object hierarchy** based data, either within the domain model or when the data was mapped back to the database. To do so within the domain model would require that the transformation be recorded for each modifier in the mapped ontology.

A small number of users (15% in the final wide test for example) expressed an interest in being able to add their own notes to instantiated **attributes**. Such a facility could be incorporated easily, however there are some potential drawbacks. Whilst these free text notes could add clarity to the exact meaning of a user’s instantiated data, that clarity could only be for later interpretation, it could not be used for any sort of automatic comparison. A free text entry facility might also encourage users to bypass the constraints on data entry designed to uphold the ontology and data quality, in order to enter whatever they wanted without constraint.

If a notes facility was incorporated, one area to look at would be the facility not only to add text notes but also to add sketch drawings. Capturing a sketch drawing could be

## Chapter 7 - Data Entry Process

done either by scanning and attaching a file or by using a quick sketch program (e.g. diva.sketch's JSketch [Pederson 2006]). It was found during qualitative research and early storyboard development that such a drawing can be valuable in taxonomic description for relating the positions and attitudes of various elements of a specimen. In a general sense such a sketch facility could be of use in many domains, particularly where the visual medium was important and it was difficult to use text descriptions to capture every nuance of a descriptive feature or of the relationship of a set of such features.

### 7.5.2 Informed Decisions

In order to empower users to make good data entry decisions, the interface should ensure the user is well informed as to what they are instantiating, including what the **attribute** concept is, what the value choices are and what the **description object** context is. The interface must also make clear to users what data they have actually entered.

#### 7.5.2.1 Clear what being instantiated

Users appeared to be sure of the **description object** context in which they were operating from speak-aloud observation results. The **description object** context was repeated in the **description object hierarchy** tree, the heading to the **description object** instantiation panel and in the **description object** informative data display.

Upon beginning work with a new **description object**, users generally checked the prominent nametag with icon in the informative data display and then tracked over to the **description object hierarchy** tree to double check the context of the linked selected node. Users were observed occasionally to perform a quick check of the path of **description object** nametags to the root in the header. This usually occurred during the middle of working on a **description object's attributes**.

The only notable issue in early data entry tests involved clones, and as described in the presentation model section earlier, this issue was dealt with by extra text displays of any clones that the current **description object** could be confused with.



## Chapter 7 - Data Entry Process

Based upon speak-aloud observations and interviews in narrow tests, the display of the **attribute** identities (in the **attribute instantiation IO**'s header and informational data) was found to be adequate to keep users informed as to what the **attribute** was that was being instantiated. This generally included complex **attributes** with relative modifiers (e.g. a ratio or the relative hue of a leaf's surfaces), where the system attempted to represent the **attribute** name and relative modifiers with pseudo-natural language. However these could occasionally give user's pause for thought depending on how well the pseudo-natural language matched the user's original concept. When users had specialised the nametag during specialisation to one more in keeping with their conception, no hesitation was observed.

Value domains of **value objects** were clearly understood by users, with text labels or pictures to indicate what **value object** was represented. By displaying the pictorial definition, users were reminded of the defined concept, thus avoiding the problem of concept drift, where a descriptive concept can slowly alter over the course of a project, to take on different emphasis or even cover concepts that it originally did not. By making the picture representation so prominent, users were constantly reminded of the underlying concept. Other **value object** representations also had text definitions available on mouse-over and users were occasionally seen quickly checking these before making a data entry decision.

Where pictorial labels are used on selection buttons, the question arose of whether to use text labels as well. The nametag is displayed along with the definition on mouse-over but some testing was required to determine if the name should also be displayed normally. Part of the reason for promoting the use of multimedia definitions in taxonomy was that there were varying conceptions of what some commonly used terms actually meant. This suggested that displaying those contentious names might cause some users to misinterpret what was meant, whilst only using the multimedia aspect would side-step the contentious domain terminology issue. However, it was generally concluded that users would prefer to see the names (69% preferred to do so, 8% did not in the final wide test). From observation, it was seen that a majority of users commonly checked the names of all terms by mouse-over when entering data based on pictorial representations without names. Including the names by default would remove this step and save time. It was not however possible to measure the accuracy of the data inputted,

## Chapter 7 - Data Entry Process

except that no significant number of obvious errors were made by users that could be **attributed** to the presence or lack of text labelling.

### 7.5.2.2 Feedback on entered data

The data entry interface reflects the current status of the **specialised domain model** including the instantiation state of the current specimen. The **attribute instantiation IO** displays the details of its current instantiation, with coloured text (red for none, bold blue for instantiated data). From speak-aloud observation this display was found to provide good feedback on what had been entered, with users using it to confirm their data entry decisions were as expected. This was particularly noted as a conscious user action in cases where the entered data was complex due to alternate scores and/or modifiers.

Experienced users in narrow tests were also observed to scan down the **attribute IOs** to check for tell tale indicators of red text (as opposed to bold blue text) to check that they had not missed any **attributes** for instantiation before moving onto the next **description object**. Other indicators are available in the icons of nodes in the **description object hierarchy**. Users were observed checking these icons to confirm that all **attributes** of a **description object** were instantiated (or noted as ‘not scored’). One user in the final wide test said he got *“used to seeing the blue scores and I could tell from the icons in the structure hierarchy when I had not scored something”*. 92% of users in the final wide text gave positive or highly positive comments on system feedback on scoring status. One user could not distinguish the node colour due to colour blindness, which is to be expected based on the percentage of the population suffering from some form of this condition [Nielsen 1993]. Another indicator, such as icon shape should ideally back up the colour indicators in the interface and this is one of those more important areas where this should be done.

### **7.5.3 Efficient and effective usage**

Features of the various system models (task, presentation and domain) and underlying ontology can help support effective and efficient usage of the data entry interface. This section discusses some of the significant aspects not already covered.

## Chapter 7 - Data Entry Process

### 7.5.3.1 Abstract Interaction Objects for **Attribute** Instantiation

Expert assessments and user testing were used to develop the AIO selection strategy (see figure 7.10) and the library of AIOs. To effect the selection strategy requires access to data from the **specialised domain model**. Some of this data ultimately originates in and hence relies upon the underlying domain ontology (**attribute** value-constraint types, pictorial definitions). Other data primarily relies upon the specialisation process (concrete instance status, final value domain size).

The AIO selection strategy first considers whether the **attribute**'s value constraints are selection or user entry based. Most user entry in the system is likely to be numerical, as free text entry is minimised by using selection from ontology based terms.

AIOs for **attributes** with numerical entry constraints were fairly straightforward to develop, with text boxes constrained to numerical data entry and a unit of measurement IO. Ranges were supported by the domain model for abstract **description object** instances, which matched working practice. Concrete instances had only a single value in the domain model and an AIO that reflected that was developed. The use of modifiers allowed users to note if their data were approximations or non-scientific averages rather than accurate measurements. Where accurate statistical analysis was desired then concrete instances could be captured allowing such data analysis later. It was considered that such data analysis did not generally belong as part of data entry itself.

The AIO selection strategy emphasises using pictorial selection when available. Initially no pictures were available in the angiosperm ontology despite a vague aspiration to do so, resulting in an interface with only numerical text entry IOs and selection based IOs using text labelled checkboxes to represent **value objects**. Whilst acceptable, these interfaces had lower user satisfaction feedback than later interfaces including pictorial selection. Ontology developers were persuaded to add more picture definitions when sample appropriate pictures were artificially added to the domain model to show them the advantages in example data entry interfaces.

Initially pictorial selection based AIOs were only selected when all **value objects** in the value domain had pictorial definitions, but this was changed to a strategy emphasising them to a greater extent due to strong user preference in taxonomy for picture based selection. For instance the selection of pictorial selection AIOs was expanded to select

## Chapter 7 - Data Entry Process

them where only one **value object** was lacking a pictorial definition as there were no pictorial definitions for some concepts (such as '*glabrous*', a type of texture relating to hairs which indicates no hairs). Pictorial representation was important for some **attributes**, to such an extent that a pictorial importance tag was added to the **specialised domain model**. Due to strong user preference despite screen space considerations, pictorial importance was eventually defaulted to true in the angiosperm ontology.

Where the size of value domains of **value objects** grew large, the AIO selection strategy selected an AIO designed to deal with screen space considerations. In very large domains this resulted in an AIO that uses a pull down list widget as the primary **value object** selection IO. This was the only AIO based **attribute instantiation IO** that a majority of users expressed dissatisfaction with. Despite the screen space concerns, users believed they would wish to see all options on the screen at once. There are however presentation problems with presenting very large numbers of **value objects** in a value domain simultaneously, which are not necessarily obvious to users. Cognitively users cannot hold very large number of concepts such as **value object** data entry options in their mind simultaneously. The average human short term working memory (as demonstrated by digit span) is 7 (+/- 2) chunks of information, though educated academics can reasonably be expected to be slightly higher [Miller 1956, Wechsler 1997]. Users are thus forced to scan large value domains in sections in any case, to find targets or conceive their options. This factor may be partly to blame in the dissatisfaction with pull-down list IOs as they are only used in large value domains. Users are not used to having a constrained selection from a large defined value domain, either they have a small to medium value domain or have an unrestricted choice from their own knowledge. It is perhaps suggestive that users may need additional help in dealing with large value domains at data entry. Good, appropriate editing in the specialisation phase can reduce value domain size to a more manageable size whilst representing all reasonable options. If consistently supported by the ontology, sub-**attributes** that effectively split the value domains into manageable sections might be useful to represent through the AIOs. The ability to hold chunks of information simultaneously in a user's short-term memory can be related to their apparent preference for pictorial representation. As taxonomy users cognitively prefer to think in an iconic (i.e. visual) manner, they will be able to hold more concepts in short term memory if they are visual concepts rather than if they are abstract.

## Chapter 7 - Data Entry Process

For reasons of simplicity and practicality, the strategy does not attempt to calculate the detailed screen space footprint of IOs as a criterion. Some **attributes** can increase their screen footprint due to alternate values, so avoiding scrolling could not be guaranteed. It was also decided that keeping all of the **attributes** of a **description object** together was desirable to match user expectations and to simplify overriding the task order.

In some **attribute instantiation IOs** there can be a lot of unused white space depending upon screen resolution plus the representation type and number of **value objects**. Unused white space can also be generated by the knowledge-based grouping of one **description object's attributes** to one window, so that when only one or two **attributes** require instantiation for a **description object**, there may be unused space. The loss of utility of the unused space was considered to be more than balanced by the beneficial effects of a consistent and clear **attribute** representation. This grouped the interaction capability together for users as well as keeping the **attributes** of a **description object** grouped clearly together to help maintain the user's sense of context within the description space.

The grouping strategy can cause significant scrolling to be required in extreme cases, when there are very many **attributes** for one **description object**. The acceptance of the need for scrolling within the **description object** instantiation panel requires extra care and feedback to ensure users do not omit to instantiate **attributes** off the bottom of the screen. Testing found feedback adequate to avoid this problem, although on very rare occasions there was a little user dissatisfaction where the **attribute** instantiation task order did not match their expectations and substantial scrolling down and up was required to override it. Better **attribute** task ordering would overcome this rare issue as discussed further below (7.5.3.2).

These interfaces were however still usable despite possible scrolling requirements and unused space. Some potential loss of usability was worthwhile to avoid the danger of generating a totally unusable interface that could develop from a more complex layout strategy, for as Vanderdonk [1994] says no layout strategy produces a usable interface in all cases.

### 7.5.3.2 Default task order

Matching real-world working practice and user expectations of tasks and data is generally beneficial to usability [Nielsen 1993]. Taxonomists work specimen by specimen. Managing specimens and overriding this aspect of the default task order is therefore not so vital. Using tab windows for each specimen was considered but rejected because the number of specimens in the system could be so high as to make tabs unusable.

The default task order represented in the **data entry task model** was found to be a good match for user's working practice, if users had interacted with the **description object** task order during the specialisation process. If the task order had not however been altered to fit with the standard domain concept of acryptic ordering, users did not find the default order to be natural. This matches the findings of users using the **description object hierarchy** tree in the specialisation process as discussed in chapter 6. If there is a strong domain order, then it would be useful to represent that order in the ontology, thus reducing the reliance on extra specialisation tasks.

The default order for **attribute** instantiation within a **description object** is primarily alphabetical. This arbitrary order was changed during development from one in which **attributes** that required (numeric) text entry were presented before those requiring selection to one that was purely alphabetical. This change was based on feedback from a very small sample of users in the narrow tests. However, during the final wide test, some users expressed a preference for having all measurements together. This preference was backed up by observation of user's practice in conducting their examination of the specimen for data entry. When measurement tools were utilised, it was usually observed to be marginally easier to do all measurements on the one structure, one after another whilst tools were readily at hand. Focussing on one structure at a time though was still appropriate as the overhead for readying measurement tools was low i.e. doing all measurements for all structures was not generally desirable.

The default **attribute** instantiation order can be easily overridden as it is just enforced by the order of IOs in the same window. However to improve the match with user's expectations, the facility to alter the task order during the specialisation stage could be expanded to include **attribute** order, thus allowing individual users to specialise the order to fit their own preferences.

### 7.5.3.3 User autonomy to complete tasks as required

Users are empowered to work in a natural way, overriding the default task order as discussed above. To achieve this autonomy they must be able to navigate the interface to carry out their data entry tasks. To do so requires users to navigate **high-level concepts, description objects and attributes**. Testing showed users were generally able to navigate the interface effectively.

The **description object hierarchy** tree is the primary vehicle for overriding the default task order within the instantiation of a specimen. The navigation evaluation of this view was discussed in detail in chapter 6. During the data entry process, this view is filtered to only include the **specialised domain model**, making it easier to navigate generally, as this tends to result in a smaller tree. The order of the nodes also reflects any specialisation of the task model, which can improve navigation via domain knowledge as the order better reflects the user's cognitive model of the data. Evaluation of overriding using this view during the data entry process was very positive with users observed to be confident and capable in navigating the filtered view during the final wide test and the narrow tests.

### 7.5.3.4 Timing

As discussed in chapters 2 and 6, the time question is a particularly important one for taxonomists and for adoption of this sort of theoretically more rigorous approach to data entry. Whilst data entry is an expected process to any data entry user, it is a process that can be repeated many times for a project unlike specialisation. The time costs are thus potentially very large. Direct statistical comparisons with current practice cannot be made, as current practice is not based on a structured, ontology-based data model.

Time costs for data entry were measured in the final wide test. These costs were measured for users entering data based upon examination of real life specimens using a data entry interface that acted upon a **specialised domain model** which had been specialised by the same user in the preceding specialisation tests. The total time costs include the whole data entry process including time spent examining the specimen. Users were each given one specimen chosen randomly from a shortlist of four appropriate specimens with a similar level of detail.

## Chapter 7 - Data Entry Process

Users spent up to 3 minutes getting a feel for the specimen through a general examination before beginning data entry (not included in below time costs). The size of the **specialised domain model** for instantiation varied depending upon the user's earlier specialisation as some users added more specialised **attributes** than others, based on the same data. Users instantiated a mean of 19.2 (+/- 4.2 SD) **attributes**, excluding special presence **attributes**, of 9-12 **description objects** (excluding those **description objects** with no specialised **attributes**). The time cost varied from 10 to 24 minutes, with a mean of 16.8 (+/- 4.6) minutes.

These time costs should only be seen as generally indicative. Many variables such as the nature of the specimens and the complexity of the **attribute** instances will influence any actual results. Users believed that the data entry process was approximately as quick (46%) or slightly quicker (54%) than current data entry methods. Those who believed it was quicker **attributed** this to the speed and ease of selecting values rather than entering them. They particularly thought the picture box representations of values for selection, allowed them to consider the options quickly, especially where they could compare what they viewed on the data entry interface to what they observed on the real world specimen instance.

From observation of the tests, it appeared that the majority of the time cost was attributable to the user's observation of the specimen, particularly in the case of numerical measurement. During selection from qualitative **value objects**, users were sometimes observed to move their attention between the displayed interface and the specimen, comparing what they observed against the data entry options. More experienced taxonomists did this less than those inexperienced in the group of plants or with less general domain experience.

No significant correlation was seen between time costs and experience with the interface, except when entering alternate values, where less experienced users were initially more hesitant, spending extra time checking that the system's feedback on data entered matched what they had wished to enter.



### 7.6 Conclusion

This chapter has described the process by which users enter data using an automatically generated interface, based upon the specialisation of a mapped ontology and the known task of data entry. The interface is generated by two presentation models, one to control an overview of description space and one to control the main data entry instantiation panels. These system presentation models have been described and discussed, along with the domain model extensions for capturing instantiation data.

The results of the process are **specialised domain models** instantiated for a real world specimen, which are consistent with the underlying ontology. Each instantiation can be exported to a file for transfer to a database. To simplify the mapping process back to a database, an ontology based unique identifier, for each ontology term used to form a descriptive element, is captured when the ontology is first mapped to the database. These identifiers are included with the exported XML.

Generally evaluation suggested the final system for data entry was effective at providing a basis for supporting the needs of high quality data collection using a specialised ontology.

Some domain model concepts are utilised to capture the nuances of the entered data. The special **presence attribute** was developed by our approach and does not rely on the ontology, however mapping it to an ontology element will make it easier to map back to a database. The presentation and domain model understand this **attribute**, and its instantiation state is used to effect logical consequences on presentation and instance data. This special **attribute** can also improve the clarity of the captured data, as can the ‘**not scored**’ statement for **attributes**. The domain model also required mechanisms for distinguishing between various cases of multiple instantiated values for one **attribute**, in order to both record the data and allow the presentation model to present it appropriately. Modifiers enabled other nuances of data to be captured, without resorting to free text entry. A small but significant number of users found this facility to be very useful. The modifiers, their grouping and applicability do rely on the ontology. The use of **preferred units** can reduce the time costs and reduce errors (when units are omitted or wrongly selected) in data entry if they are enabled in the specialisation process. Defining default units in the ontology could reduce this dependency.

## Chapter 7 - Data Entry Process

The default order of the **description object** instantiations relied upon the specialisation process to alter the default order to match working practice as this was not represented in the ontology. The default order of **attribute** instantiation within a **description object** was less significant due to their grouped representation in the same window, however final evaluations suggest that providing the facility to specialise this order would be valuable to further match working practice and improve usability. However, the ability of users to act autonomously, overriding the default task order during data entry, means the operating of the interface is not totally reliant on the default task order.

Sensible constraining relationships in the ontology reduce but by no means eliminate the reliance on the specialisation process to ensure that sensible data entry choices can be presented.

Users responded well to the automatically generated interfaces controlled by the presentation models. During the final wide test no negative user replies were logged on the discussion subject of ease of use, instead positive comments such as “*very easy to use*” and “*straightforward*” were typical. Users were generally observed to be able to express their descriptive concepts within the constraints of the specialised ontology with appropriate feedback and information to inform their decisions.

The presentation models effectively present the appropriate elements of the **specialised domain model** to users for instantiation. To be effective it must be clear to users what they are instantiating and in what context they are operating.

The actual data entry panels use knowledge-based identification, being based upon one **description object** from the domain model that has specialised **attributes** to instantiate. This grouping level matches well with the taxonomist users’ working practice, enhancing the usability of this window. Within the **description object** instance representation, each of its **attributes** has a grouped set of IOs to instantiate it. The grouping level offsets one of the traditional drawbacks of automatic generation, that users require information from multiple objects in one window [Szekely 1996a], as all the required information to make an informed data entry decision is available. Easy access to definitions, **attribute** nametags and where possible pictorial representations of data entry options, are used to promote clarity of what is being instantiated in an **attribute instantiation IO**. Feedback on the current state of instantiation is also

## Chapter 7 - Data Entry Process

utilised, with visible natural language representations of entered instance data along with the use of at a glance indicators (e.g. colour and format of the text) of the instantiation state.

The **attribute instantiation IOs** are generated by the **data entry presentation model** using a selection strategy that relies upon data both from the specialisation of the domain model (concrete instance status, picture representation importance tag, final value domain size) and the underlying ontology (**attribute** value-constraint types, presence of pictorial definitions). This strategy selects an abstract interaction object from a system library that is then instantiated with the data of the actual **attribute** in the **specialised domain model**. The strategy developed for taxonomy favours pictorial representations of **value object** selection options for instantiating **attributes**. These have a high screen space cost, but provide a clear visual representation of the data entry options supporting the use of definitions and the visual cognitive processes of taxonomists. Users continue to express a desire to extend pictorial scoring with more pictures. During the final wide test, users expressed a wish to see pictures **for value objects** that were appropriately tailored for the structure being scored. This would require alterations to the ontology to track the multimedia definitions for states (**value objects**) dependent on structural context (no changes to the domain model would be required).

In taxonomy it is likely that in line with current practice the data entry user will be one and the same as the specialisation user. With a clearer method of specifying the data entry data requirements, such as the ontology-based system, this need not be the case. Users require sufficient domain knowledge to make informed observation of real-world specimens, with the back up of defined descriptive element choices in the interface. This suggests that users do not need to typically have as great a level of domain expertise as the specialisation process requires. A limited investigation of the effects of having separate data entry users from the specialisation users was conducted by having the wide user test subjects enter data from a specimen based on part of a **specialised domain model** for *prunus*, as developed by another expert. Although more care was taken to be sure of the meaning of descriptive elements, users showed no more difficulty in entering data based upon another's specialisation than when based upon their own. Evaluation suggests that the learning curve of the interface is easy, with users rapidly gaining familiarity with the primary interface functionality.

## Chapter 8

### Application in other domains

#### 8.1 Introduction

This work has introduced a domain ontology based tool for the semi-automatic generation of data entry interfaces. The tool allows domain users to specialise the data entry for individual projects without requiring the intervention of an IT expert. The approach is model-based, focussing on domain and presentation models. The approach has been developed as a solution to the difficulties of taxonomic description data collection. Taxonomists have tested the system and it has generally been found to offer a number of significant advantages over current practice.

Finding domains where automatic interface generation techniques can be used successfully and effectively remains a challenge for researchers [Nichols 2005]. Many automatic interface generators have tried yet failed to introduce a universal method for generating interfaces. By restricting the approach to specific domains, greater success may be achieved (e.g. DIGBE [Penner 2002]). Our approach in the taxonomy domain does demonstrate a specific domain where automatic generation can be successful. The hypothesis of our approach in using domain ontologies however suggested that the approach could be more generally successful. It does not attempt to be a universal approach, but by fixing the task to that of data entry for a database, the approach can work for domains outside taxonomic description. The needs of generalised data entry were considered in development of the tool. To show generalisation however, another ontology needed to be mapped into the system and the approach shown to work.

This chapter will look at applying the approach to domains other than taxonomy. An example of such an application was tested after the tests with the RBGE taxonomists and the angiosperm ontology were completed.

## 8.2 Importing domain ontologies

Initially the mapping of domain ontology conceptual models to the **abstract domain model** (see figure 5.1) was captured within a programmatic java class. To ease the burden on the IT expert performing the mapping, a transfer XML format was developed into which the ontology was transformed either programmatically or manually. The system then understands the ontology as presented in the XML transfer format. Figures 8.1, 8.2 represent excerpts of the XSD template for this XML format, which captures all possible elements of the **abstract domain model** that can be mapped from a domain ontology.

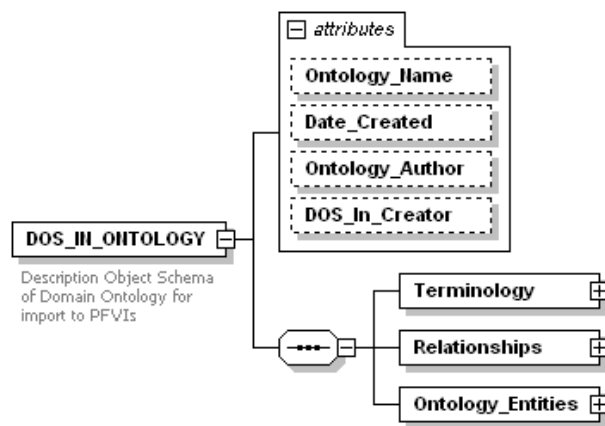


Figure 8.1: Transfer XML format (.xsd file). Top level elements.

## Chapter 8 - Application in other domains

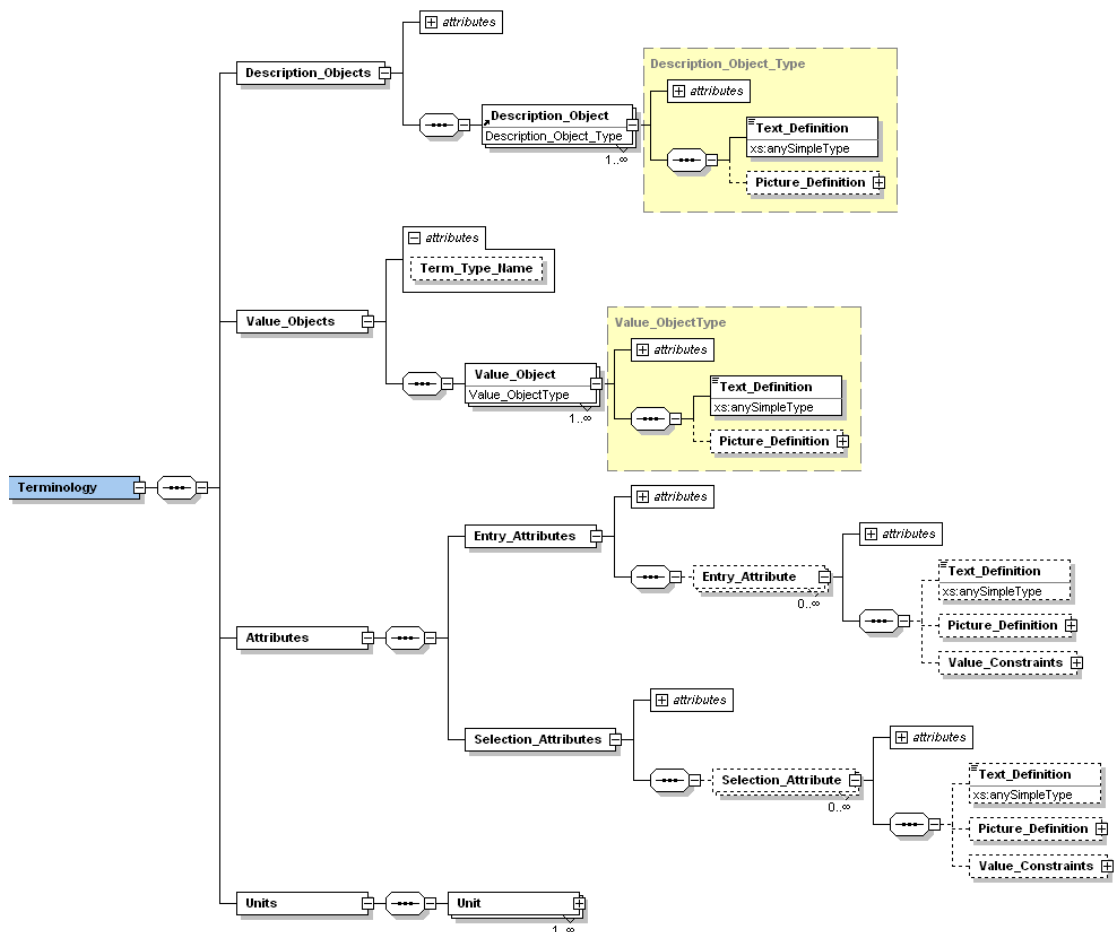


Figure 8.2: Transfer XML format (.xsd file). Terminology elements.

### 8.3 Other domain ontologies

The approach is designed to work by mapping existing domain ontologies rather than having to develop them specifically for the application or a related application.

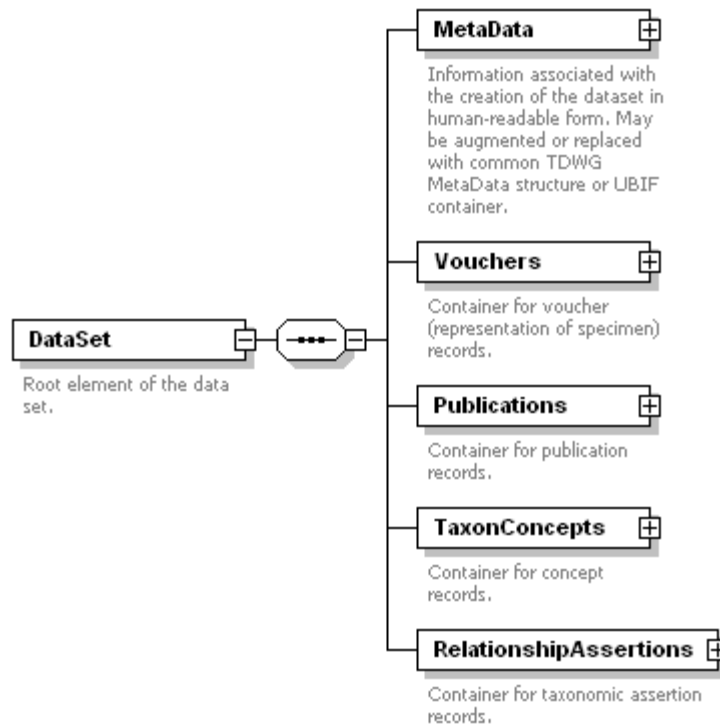
Acquiring useful domain ontologies that are suitable for constraining the description of entities of interest can be difficult especially in accessible formats. Popular ontology systems such as Protégé[Gennari 2002] can be adapted, but Protégé does not generally model relationships between its classes other than subsumption, which may not relate well to a domain user's concept of the data structure, so may require some extra work to map effectively a primary organising relationship between the mapped **description objects**.

## **Chapter 8 - Application in other domains**

XML schemas are however readily available and might be able to be used as a lightweight ontology for mapping. If so, it would open up a new source of ontologies to use the system with. There are some tools to generate data entry interfaces on the basis of XML schemas, such as XML Spy which can create data entry interfaces for a XML schema but must define a style sheet to do so. It does not offer definition support, description context, feedback, etc however, it is simply another forms generator based on value-types. Another example is Microsoft InfoPath 2003, but it does not automatically generate an interface, it requires users to drag and drop the XML elements to build form templates. The following section shows the use of one such schema as a domain ontology.

### **8.4 TDWG Taxonomic Transfer Concept Schema**

A suitable ontology and user community was found in TDWG's Taxonomic Concept Transfer Schema (TCS) [Kennedy 2006]. TCS is an XML transfer format developed for taxonomists, ecologists and other bioinformatics experts to exchange data about classification names and classification concepts (it does not include taxonomic description data). Whilst in the same general field of bioinformatics, the schema was used to capture very different information than taxonomic specimen description. By using the TCS XML format as a domain ontology, the extent of the adaptability of the ontology-based approach could be demonstrated. As the TCS was not developed as an ontology, if the approach can use it as such, then there are potentially more sources of domain ontologies for other data entry applications.



**Figure 8.3: Excerpt from TCS schema (version 0.88), showing top-level xml elements.**

### 8.4.1 Mapping the ontology

After a briefing on the domain model of the approach developed in this research, two IT experts were able to come up with a mapping between the TCS schema and the **abstract domain model** (as represented in the transfer XML). Mappings were initially made between XML `xs:elements` and **description objects**, with element `xs:attributes` as **attributes** and **value objects** were formed from XML `xs:enumerations`. Appropriate permitted relationships between **description objects**, **attributes** and **value objects** were derived from the circumscription of the mapped xml elements. Names were mapped from `xs:element` names, `xs:attribute` names and `xs:enumeration` values. The value-type for **attributes** was derived from `xs:type` of the `xs:attributes`. Definitions were derived from the `xs:documentation` where available, otherwise the nametags themselves were used as definitions in the absence of any other data.

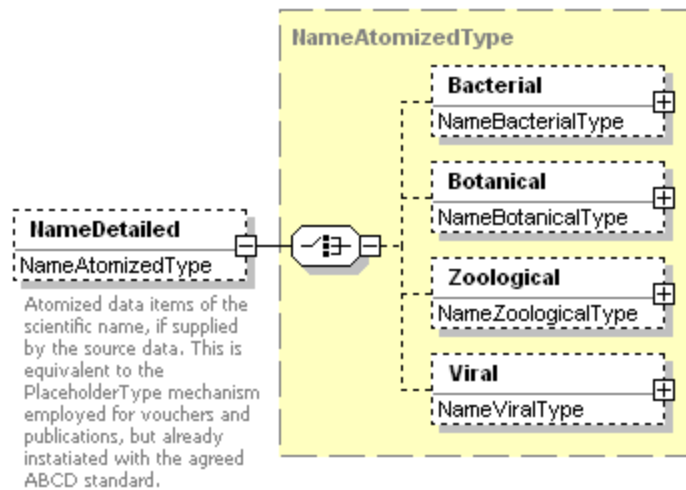
No modifiers, units or multimedia definitions were mapped as no such data was present. Basic units could have been added from another source to the mapping, but none were, as none were required for any of the possible data entry items. The primary organising



## Chapter 8 - Application in other domains

relationship (for **description objects**) was mapped to the schema's xs:element hierarchy structure. Whilst this mapping was fairly simple to implement, there were some complications that arose.

The xs:element 'NameDetailed' had a choice of sub-elements. These each represented a different set of **attributes** for capturing name data for concepts in the various fields (bacterial, botanical, zoological or viral). Only one type would be used in any one 'TCS dataset' instantiation and in fact in any one project group of TCS datasets, as the types were specific to different types of data and users. The **abstract domain model** does not capture the information that a **description object** can only have one of its child **description objects** included in a **specialised domain model** nor that only one child **description object** can be instantiated for a given high-level concept. A similar situation was touched upon in the taxonomic description domain, where only one of a number of **description object** clones could be present on a given specimen. In that case, the responsibility for avoiding nonsensical data entry was laid on the data entry domain user. They could use the prominent presence **attribute** that was required for every **description object** to ensure only one was marked as present, while those not present on their specimen should be marked as absent. The same decision was reached in this case, however it did raise the question again as to how far data entry users should be trusted and whether the special alternative **description object** relationships should be represented and their rules enforced.



**Figure 8.4: TCS alternative types of names. (XML Spy view).**

Another issue, which did in this case result in an alteration of the mapping, was that a number of `xs:elements` that had no child elements, were being used merely to capture one simple piece of data such as an id or other text string. It was felt that these would be more accurately represented as **attributes** of the parent **description object** rather than as **description objects** in their own right. Thus where an `xs:element` only contained one **attribute** and had no child elements, it would be mapped to an **attribute** with an **attribute** relationship to its parent. An example of this can be seen below with `xs:element` 'PublicationDetailed' which had a large number of sub-elements which were each designed to capture one piece of data. These sub-elements were mapped to **attributes** (e.g. `xs:element` 'Author' is mapped to an entry **attribute** 'Author' with value-type 'string', `xs:element` 'type' is mapped to a selection **attribute** 'PublicationDetailed Type' with value domain relationships to **value objects** mapped from the various enumerations such as `xs:enumeration` 'Audio-visual Material').

## Chapter 8 - Application in other domains

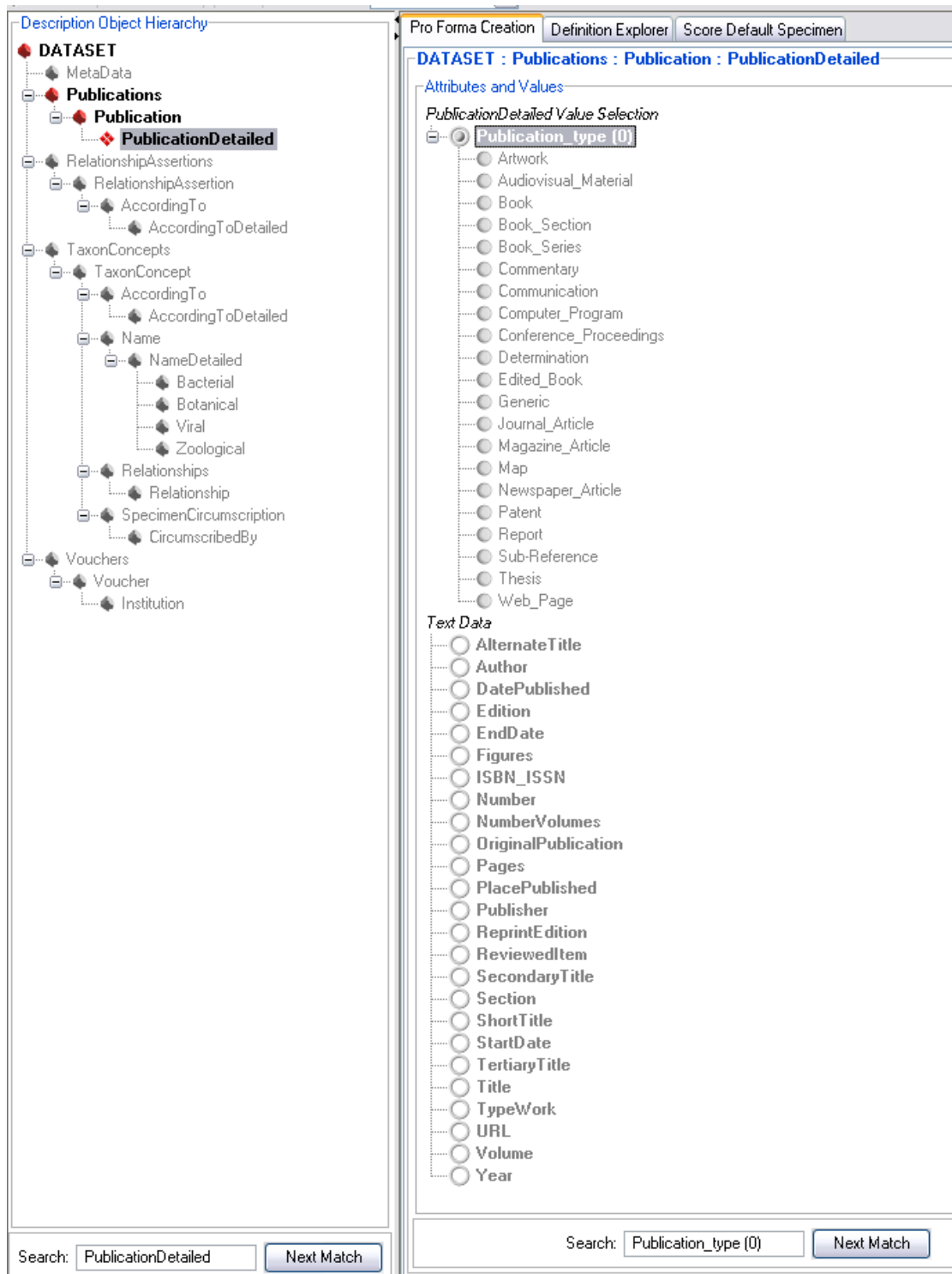
```
<xs:element name="PublicationDetailed" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Reference broken down into individual components. (The current version is based on
R.Pyles's Taxonomer data model, which in turn is based on Endnote 7.) [A]</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Author" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Year" type="xs:string" minOccurs="0"/>
      <xs:element name="Title" type="xs:string" minOccurs="0"/>
      <xs:element name="SecondaryTitle" type="xs:string" minOccurs="0"/>
      <xs:element name="Publisher" type="xs:string" minOccurs="0"/>
      <xs:element name="PlacePublished" type="xs:string" minOccurs="0"/>
      <xs:element name="Volume" type="xs:string" minOccurs="0"/>
      <xs:element name="NumberVolumes" type="xs:string" minOccurs="0"/>
      <xs:element name="Number" type="xs:string" minOccurs="0"/>
      <xs:element name="Pages" type="xs:string" minOccurs="0"/>
      <xs:element name="Section" type="xs:string" minOccurs="0"/>
      <xs:element name="TertiaryTitle" type="xs:string" minOccurs="0"/>
      <xs:element name="Edition" type="xs:string" minOccurs="0"/>
      <xs:element name="DatePublished" type="xs:string" minOccurs="0"/>
      <xs:element name="TypeWork" type="xs:string" minOccurs="0"/>
      <xs:element name="ShortTitle" type="xs:string" minOccurs="0"/>
      <xs:element name="AlternateTitle" type="xs:string" minOccurs="0"/>
      <xs:element name="ISBN_ISSN" type="xs:string" minOccurs="0"/>
      <xs:element name="OriginalPublication" type="xs:string" minOccurs="0"/>
      <xs:element name="ReprintEdition" type="xs:string" minOccurs="0"/>
      <xs:element name="ReviewedItem" type="xs:string" minOccurs="0"/>
      <xs:element name="Figures" type="xs:string" minOccurs="0"/>
      <xs:element name="StartDate" type="xs:string" minOccurs="0"/>
      <xs:element name="EndDate" type="xs:string" minOccurs="0"/>
      <xs:element name="URL" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="type" use="optional">
      <xs:annotation>
        <xs:documentation>Enumerated list of publication source types.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Generic"/>
          <xs:enumeration value="Artwork"/>
          <xs:enumeration value="Audiovisual Material"/>
          <xs:enumeration value="Book"/>
          <xs:enumeration value="Book Section"/>
          <xs:enumeration value="Book Series"/>
          <xs:enumeration value="Computer Program"/>
          <xs:enumeration value="Conference Proceedings"/>
          <xs:enumeration value="Edited Book"/>
          <xs:enumeration value="Journal Article"/>
          <xs:enumeration value="Magazine Article"/>
          <xs:enumeration value="Map"/>
          <xs:enumeration value="Newspaper Article"/>
          <xs:enumeration value="Patent"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

## Chapter 8 - Application in other domains

```
<xs:enumeration value="Report"/>
<xs:enumeration value="Thesis"/>
<xs:enumeration value="Communication"/>
<xs:enumeration value="Sub-Reference"/>
<xs:enumeration value="Determination"/>
<xs:enumeration value="Commentary"/>
<xs:enumeration value="Web Page"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
```

**Figure 8.5: Excerpt from TCS xsd containing PublicationDetailed data. [Kukla 2005]**

## Chapter 8 - Application in other domains



**Figure 8.6: Description object ‘PublicationDetailed’ mapped from xs:element ‘PublicationDetailed’ and its sub-elements as represented in the specialisation interface.**

Some TCS elements were marked up as bounded, meaning that there was a limit to the number of these elements that could be included in an instantiated xml file. The **abstract domain model** does not however capture this information, as it allows all

## Chapter 8 - Application in other domains

**description objects** to be cloned as often as desired or have concrete instance data for a theoretically infinite number of instances captured. It is not likely users would want to capture data on a number of **description object** instances, where only one in the real world could exist, however not enforcing this constraint from the ontology, does expose a risk, however slight, of generating a **specialised domain model** instantiation that is not compatible with the underlying domain. Future work would need to close this omission by ensuring the constraint information could be captured and enforced.

The mapping failed to capture the implication that unbounded elements in TCS were generally intended to capture repeated instances of the element within one taxon concept dataset (a high-level concept). This should have caused the **description objects** mapped from unbounded elements to be marked for concrete instance data by default, so as to more accurately represent the underlying data to users in the specialisation process. Users in that process can mark the data in such a way, but it should be captured originally in the mapping to avoid relying on users realising and essentially reinventing the wheel. The implications of concrete and abstract data entry need to be made clear to IT experts performing the mapping.

The unbounded issue also re-raises a minor issue from some of the later narrow tests in taxonomy, that it would be useful on occasion to expand the concrete instance data from grouping just the **attributes** of one **description object** to grouping the **attributes** of a **description object** and its descendent **description objects**. In taxonomy this was not a major issue as most **description objects** were focussed on individually by users performing observations and where this was not the case, cloning could be used to group observations and relative modifiers used to relate observations about different **description objects**. In TCS as in a number of other domains, a series of concrete instances will want to be captured where the data for the main **description object** instance needs to include its child **description objects**. Cloning at specialisation can manage a small number of instances, but where a large number of instances will be determined at data entry, an expanded concrete mechanism in the domain model to tie together a concrete instance over multiple **description objects** is more appropriate. This minor adjustment to the domain model would make the system more generally applicable. It would also involve minor adjustments to the presentation models to indicate the extent of concrete instances, clarify navigation between instances and refine the window identification strategy to group the expanded concrete instance.

## Chapter 8 - Application in other domains

Equally TCS elements are marked as optional or not. In the domain model, all **description objects** and **attributes** are optional at the discretion of the specialisation user. Non-optional TCS elements can always be left as blank however, so as long as the mapping process for transferring data out of the system ensures that the unrepresented non-optional elements are actually represented by blank elements (or **attributes**) then the mapped xml instances output will be compatible with TCS. However this work around does not disguise that this information should be captured. The mapping could have captured the data that the mapped **description objects** and/or **attributes** (or more precisely the **description object – description object** and **description object - attribute** relationships) were obligatory to be included in the **specialised domain model**. This would require a minor addition to the domain model to capture the constraint and the **ontology presentation model** should represent this added aspect in its representations of **description objects** and **attributes**. Alternatively a lesser constraint could suffice of simply including them by default, allowing the specialisation user to remove them if desired, but representing them as uninstantiated blank elements in the exported instances. This approach would still rely on the mapping to ensure compatibility of exported instances but would not require any changes to the models or system.

One final mapping complication was not unexpected. It occurred where the value constraints of an **attribute** were selected from a set of user instantiated elements from elsewhere in the schema. This occurred for example with elements of '*RelationshipAssertion*' which has **attributes** that reference element instantiations for '*TaxonConcept*', requiring the same instance to potentially be referenced multiple times. The system does not support value-types of instantiated **description object** instances. To do so would practically require enforcing a task order that ensured the referenced instances forming the values were first instantiated before anything instantiating anything that referenced them. In the TCS case though, a viable work-around was used, whereby the **attributes** with references were mapped to **attributes** with a entry value-type 'string' which would allow users to enter the Ids of the referenced concepts as all such referenced elements in TCS would have an unique ID.

The mapping could have possibly better matched the likely real world task order by mapping a **description object hierarchy** relationship to link mapped referencing

## Chapter 8 - Application in other domains

elements to another instance of the referenced element. For example *'TaxonConcept'* xs:element has a *'publication'* **attribute** that references instances of the xs:element *'publication'* from elsewhere in the schema. The mapping by the IT expert, simply mapped a *'Publication Id'* **attribute** of value-type *'string'* to the mapped *'TaxonConcept'* **description object**. Elsewhere in the **description object hierarchy** there was a **description object** representing the referenced publications where users could enter publication details. However users may find it more natural to simply add publications, as they need them when entering their taxon concept datasets, such as when instantiating the *'Publication'* **attribute** of a *'TaxonConcept'*. Thus it might fit better with the natural task order, if an instance of the *'Publication'* **description object** was included as a child **description object** of *'TaxonConcept'* in the **concrete domain model** (and could thus be included as such in the **specialised domain model**). This issue shows the importance of considering the working practices and cognitive *'natural'* data models of data entry users when performing the initial mapping and looking beyond the IT-based data structure. Specialisation users cannot add new relationships not supported by the mapped domain model, so all those relationships that can be useful should be mapped.

Whilst the referencing of other **description object** instances within the high-level concept instantiation does not arise in domains using descriptive observation such as taxonomic description, it is common in a number of knowledge acquisition domains and so should be supported in any future work. Adding an *'instance'* value-type would not involve any fundamental alterations to the domain or presentation model, although warnings would need to be given that displayed alternatives could only reflect the current state of data entry.

Overall the mapping was relatively easily completed in approximately 2 hours work by two IT experts (plus another 30 minutes for later refinements), including time to determine the rules of the mapping (c. 1 hour) and time for one expert to manually transform the XML (using cut and paste techniques) based on the mapping rules. Feedback from the IT experts indicated that they believed the mapping was relatively easy to develop, though fiddlesome to manually implement. To do the mapping robustly a programmatic or style sheet transformation would be required to be developed to ensure the mapping could be repeated for different versions of the TCS schema (as was done for the taxonomic description angiosperm ontologies). A tool to allow developers



## Chapter 8 - Application in other domains

to specify their mapping would be of value in supporting the process. Although only done once for a given ontology model, such a tool would definitely reduce the burden on the IT expert in the system. It could also clarify the options and thus improve the mapping.

### 8.4.2 Effectiveness of presentation

The mapped TCS schema was successfully imported into the domain model and presented in the specialisation interface (see figure 8.7). Domain experts in this scenario are ecologists, taxonomists and other bio-informatics experts who wish to share knowledge about a number of concepts (TCS ‘datasets’) with other bio-informatics experts. Different users from different fields will have different data defining their concepts and so may wish to enter data based only on a sub-set of the TCS format. Users can specify the sub-set for the datasets they wish to enter, in the specialisation interface. They would then be presented with data entry interfaces based only upon the fields and elements they used, whilst still being consistent with the TCS schema for compatibility with their colleagues. The users are not IT experts and would be unfamiliar with xml formats for editing. Expert assessment from IT experts who were familiar with the needs of TCS users was used to provide evaluation of the resultant interfaces.

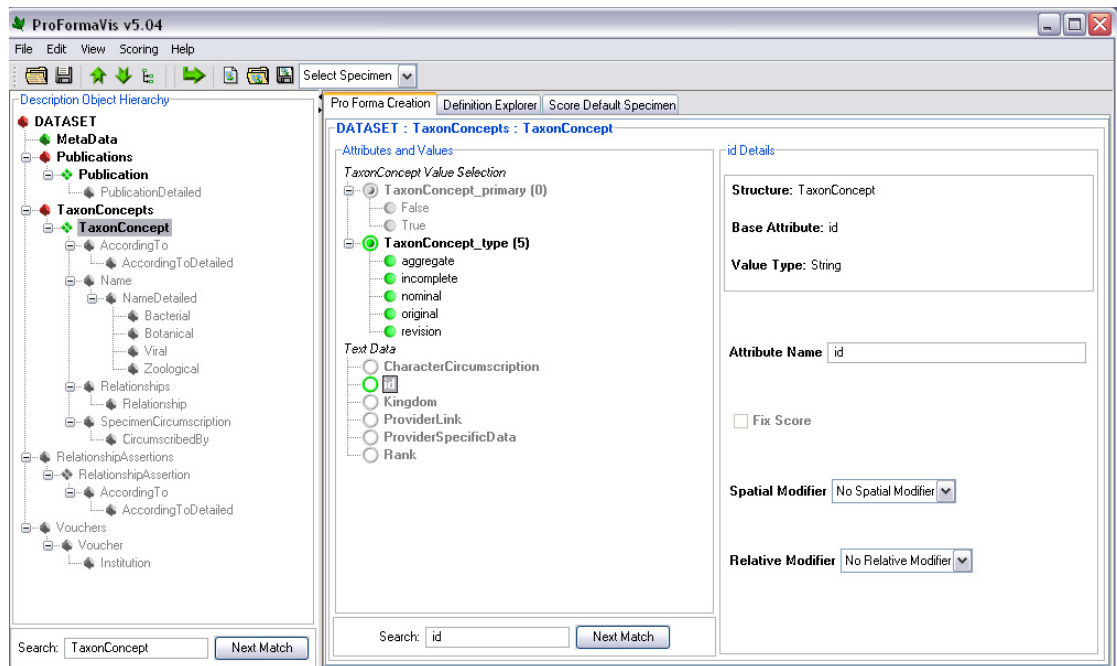
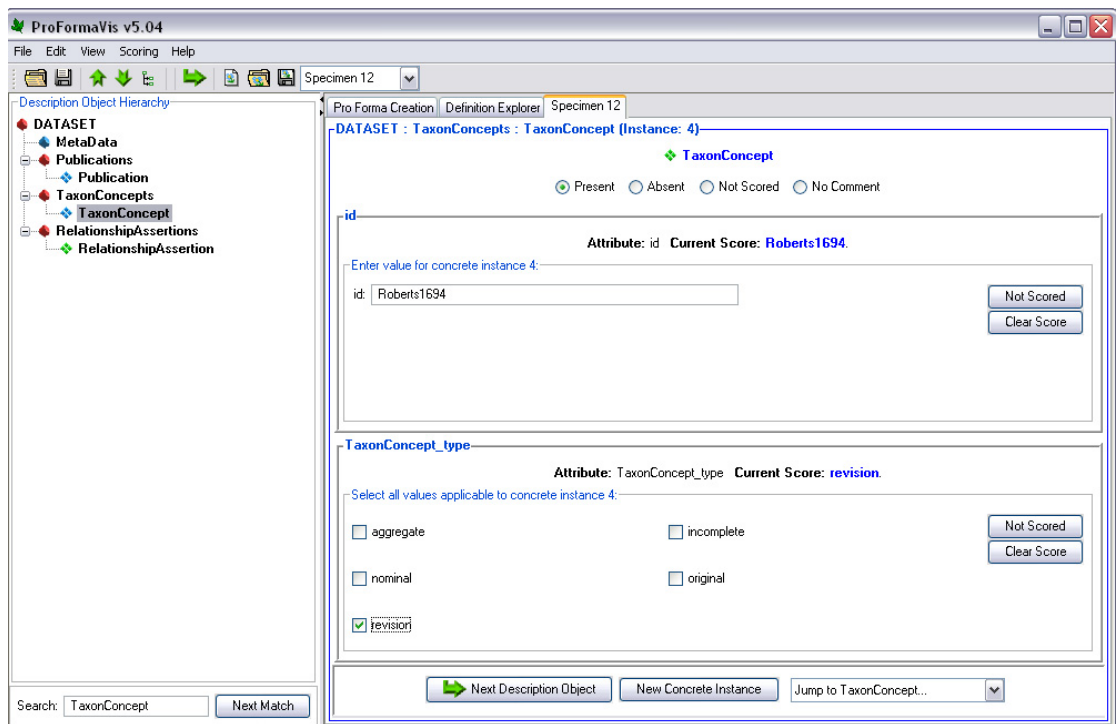


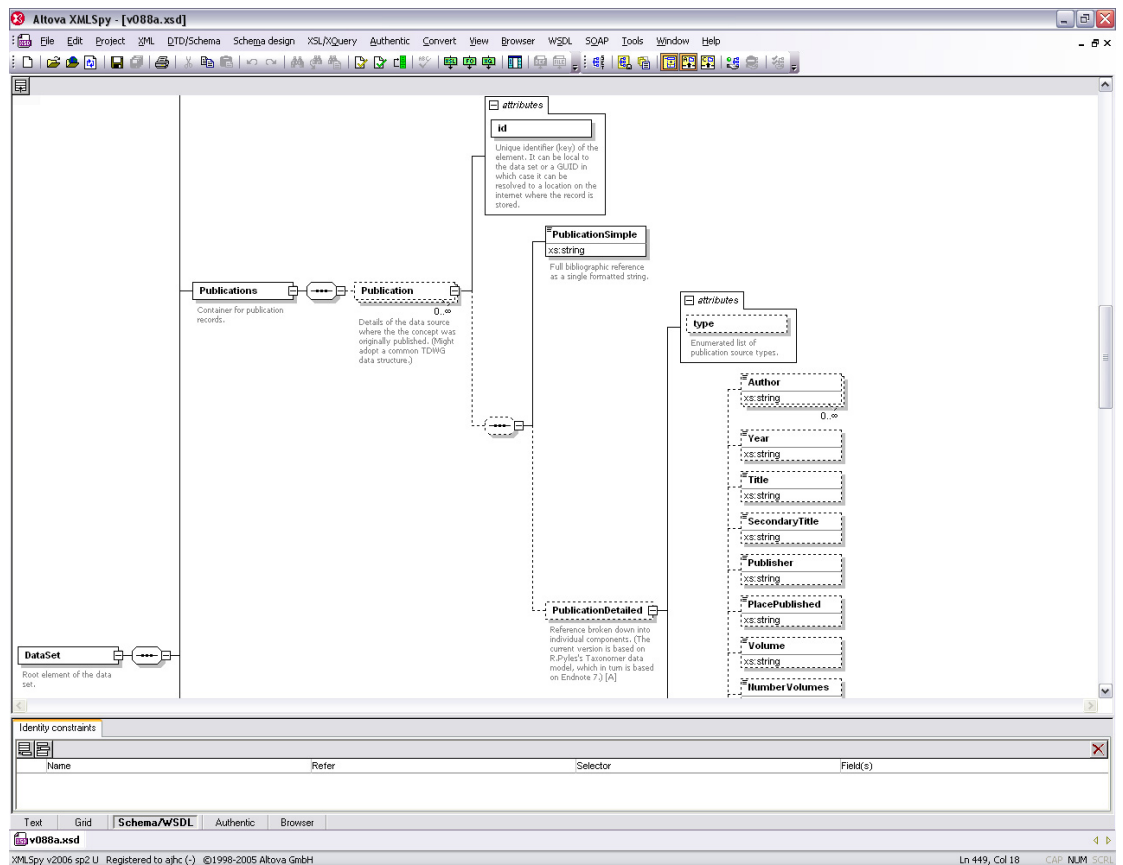
Figure 8.7: Specialisation interface example for TCS ontology.



**Figure 8.8: Data entry interface example for TCS ontology.**

No particular difficulties were noted with the specialisation process or interface. It performed well at representing TCS for specialisation. A comparison with using a popular XML visualisation tool, XML Spy [Altova 2006], was made by the experts, who were used to using this tool for working with XML structures. The specialisation interface was able to concisely present TCS's structure and relevant content more effectively on the screen, as the graphical representation of the xml hierarchy in XML Spy more rapidly extended off the window. Figure 8.9 shows an example of XML Spy's graphical representation, which while useful for IT experts looking at technical details, it is not as useful for giving overviews of system status nor for non-IT experts due to its technical nature. Representing TCS in a browser as indented text of various formats was also believed to be unfriendly to non-IT users and not suitable for their understanding or editing. In any case, for the purposes of specialisation, directly editing the schema without safeguards to ensure compatibility would not be safe. In summary, based on the informal expert evaluation, no significant problems relating to the TCS-based interface were identified and the specialisation interface was considered to work well for specialising TCS for individual needs of data entry, with the **ontology presentation model** providing a useful view of the TCS schema.

## Chapter 8 - Application in other domains



**Figure 8.9: XML spy representation of TCS schema. Most of the elements are off-screen in this tree view.**

Sample data entry interfaces were generated based upon various test specialisations of TCS based domain models (see figure 8.8). Generally the presentation was similar to the angiosperm based ones. The **data entry presentation model**'s AIO selection strategy and library was able to provide appropriate AIOs to represent the **attributes** to be instantiated. However there were no multimedia definitions in TCS, so selections were generally represented by text labels and checkbox based IOs and not pictorially. But any pictorial representation would not be as important to grasp their meaning, due simply to the nature of most of the **value objects**. There were a large proportion of **attributes** that utilised user entered text strings for their instantiation, which worked perfectly reasonably, but offer little advantage over traditionally generated forms. Again this was due to the nature of the TCS ontology which lacked other domain constraints for these entry fields.

The **data entry presentation model** identified one data entry window for one **description object**. This identification strategy was less ideal for TCS than it had proved to be for taxonomic description. In TCS **description objects** generally had less

## Chapter 8 - Application in other domains

applicable **attribute** relationships, so very often specialised **description objects** would only have one or two specialised **attributes** for instantiation, leaving a lot of unused space. This was exacerbated by the lack of modifiers and frequency of simple text entry in TCS, which often left the **attribute** instantiation IOs with a lot of unused space themselves. It was felt there was also a less important link with working practice in the TCS domain than in taxonomic description. If tailoring of the **data entry presentation model** was possible, the most important tailoring would thus be to change the grouping strategy to allow the presentation of multiple **description objects** within the one window for instantiation. Changing this grouping strategy and indeed any alteration of the **data entry presentation model** would be made by an IT expert during the initial mapping process. A tool to expose the model for editing would be useful, presenting a developer with access to the AIO selection strategy and data entry window identification strategy. Extra AIOs could be added to the AIO library to allow them to be featured in the AIO selection strategy. More extensively AIOs could be developed in such a tool, which along with a mapping tool would really transform the system into a complete model-based UIDE.

### 8.5 Conclusion

This chapter has demonstrated the application of our approach to a domain outside taxonomy, showing how a XML Schema designed to control the transfer of data for a specific domain could be adopted as a domain ontology. Further work on adopting our approach to other domains would need to take account of being able to tailor the presentation model at the time of the mapping, to the needs of the domain. Further investigation of what domain ontologies could be adopted would also be of value to test the extent of this sort of approach.

## Chapter 9

# Conclusion

### 9.1 Introduction

This work has introduced a domain ontology based tool for the semi-automatic generation of data entry interfaces. The tool allows domain users to specialise the data entry for individual projects without requiring the intervention of an IT expert. The approach is model-based, focussing on domain and presentation models with the known task of good quality data entry. The approach has been developed as a solution to the difficulties of taxonomic description data collection. Taxonomists have tested the system and it has generally been found to offer a number of significant advantages over current practice.

Testing showed the general approach to match well with user needs in taxonomy. Users quickly grasped the methodology as it outwardly matched well with their current practice of creating proformas and entering data. The system is designed to support iterative single-person working process, although it would also support a multi-person working process where one user specialised data requirements and others entered data on that basis.

### 9.2 Discussion

This research began by investigating taxonomic working practice with the aim of identifying problems and developing answers to some of them using computerised tools. Bioinformatics research, derived from the union of biology and computer science, has made great effort into state of the art biological database related research such as capturing genome sequencing (e.g EMBL Swiss Prot [Boeckmann 2003]), and integrating diverse data sets [e.g. SEEK 2006]. However database research has generally neglected the traditional tasks of botanical taxonomy. This research has modelled the working practice of the traditional task of classification and investigated

## Chapter 9 - Conclusion

some of the related issues particularly as regards traditional taxonomic description. A number of areas where IT solutions could aid taxonomists in their working practice were identified within the modelled framework.

The general problem area addressed by the rest of this project was how to use IT tools to encourage, enable and support the capture of an improved quality of taxonomic specimen description data. The main problems of inconsistent character selection, lack of character/terminology definitions and general data loss cause the difficulties in the clarity, comparability and re-use of descriptive character data for both taxonomy and the wider biological sciences. A system supporting the capture of structured descriptive data with formally defined terms was developed to alleviate some of the difficulties in data quality. The descriptive data concepts that taxonomists wished to capture about their specimens varied depending upon the project of work, but was generally consistent within the project. To avoid the need for IT developer involvement with the variations of every taxonomic project, the system uses domain experts to specify the descriptive data concepts for use in a given project.

Using a simple glossary of defined terms with in-built data model rules as a basis for users to construct their proformas challenged the aim of completing tasks in a simple and timely manner. Effectively users were defining a domain model, within the light strictures of the in-built data model rules, for each project. An ontology-based solution was thus developed in which a domain ontology of descriptive data served as an initial structured domain model, reducing burden on the user of specifying of a description composition hierarchy task, in favour of a selection based task.

The description ontology we used is unusual in that it constrains the description of plants by defining a compositional hierarchy that cannot exist in real-life as it covers all possibilities from the wide variation of flowering plants. Thus only a sub-set of the ontology is ever used at a time.

A presentation model is used to interpret and display the domain model that is the system's understanding of the ontology. To allow a non-IT expert to specialise the ontology-based domain model, requires the system to display the ontology for constrained editing. Existing ontology tools that display ontologies for editing are mostly aimed at developers rather than the needs of domain experts. This applies to both

## Chapter 9 - Conclusion

those aimed at creating knowledge-acquisition tools (e.g. Protégé-2000 [Grosso 1999]) and those aimed at knowledge sharing through ontologies (e.g. Ontolingua server [Farqhar 1997]). Some of these have become more popular ontology modelling environments since this work began, but even so without custom add-ons they primarily continue to display and address the ontology in the terms of its IT structure, such as Protégé's class subsumption hierarchies, slots and facets, which in addition to being unfriendly to non-IT users, does not necessarily match well with domain users' conceptions of the data. Unlike most of these approaches the purpose of our ontology editing was not to actually add to or change the underlying ontology, as is generally the case with ontology editors, but instead to essentially create a specialised sub-set of it, using the defined descriptive objects and existing allowed relationships. Forms editors generally only hide data entry fields and change concrete interaction widgets, in effect they allow the making of interface design decisions and not the editing of a consistent proforma based ontology. They do not support informed high-quality data entry.

Creating appropriate, good quality data entry interfaces for databases is traditionally a difficult and time-consuming process for an expert developer. Even if only adjusting an interface, the involvement of a developer for every taxonomic project to generate a project specific interface would be impractical in taxonomy as in many other domains. Our described approach automatically generates a data entry interface based on the user's specialisation of the ontology (as represented by the domain model).

In order to effectively present an ontology for domain experts to specialise in an informed manner, the approach took advantage of the domain knowledge in the ontology itself. For example, generating an effective specialisation interface was shown to be possible by making use of an organising relationship (the `part_of` structure hierarchy) from the ontology that matched well with taxonomists' conceptions of their data and which allowed them to navigate the description space effectively. To capture that domain knowledge we used a domain model in a similar fashion to model-based user interface environments (MB-UIDEs). While these systems can also involve modelling and editing domain models (e.g. Mecano [Puerta 1994], Janus [Balzert 1996]), such modelling is again directed at the developer rather than the domain expert and so to adopt the approach we would need to extend it to allow this. The described ontology-based approach does meet the criteria to be considered as a model-based UIDE. Schlungbaum identifies these as: 1. including a high-level, abstract, explicitly

## Chapter 9 - Conclusion

represented model about the system to be developed and 2. exploiting a clear and computer supported relation from that model to the desired and running user interface [Schlungbaum 1996]. The system does have an explicitly represented domain model and a presentation model to interpret the domain model into the data entry interface. Szekely [1996a] and other researchers classify MB-UIDEs by the models they use and the extent to which they attempt to automate the interface design as opposed to providing design assistance to developers. In those terms the approach is a domain based automated interface generation tool. Differences can be drawn however between the described ontology-based tool and other domain model based automated interface generation tools.

Previous domain-model based automatic interface generation tools have tended to be general in scope and unable to capture human knowledge of tasks and domain requirements [Szekely 1996a, Penner 2002]. Task model based tools still have to capture and specify the task models of individual domains and in any case tends to result in design assistance rather than automatic generation tools. The approach proposed in this work can capture the task detail as it restricts itself to the known task of high-quality data entry to a database and it utilises a domain ontology to provide specific domain knowledge to tailor the generated interface.

The other approaches also still require substantial investment by a developer to specify the models, particularly if they are to be successful in creating a useful domain specific interface, and as Novak has observed '*Nobody will create applications using specifications (models), if they can do it faster directly editing*' [Novak 2003]. This is doubtless one of the reasons that model based approaches have so far failed to achieve widespread commercial adoption, despite a strong research base [Traeteberg 2004]. The described approach in contrast uses domain experts to specify the domain model for specific project applications, based on the mapping of an existing ontology. Whilst the mapping of ontologies is not a simple task, it likely to be easier than designing a domain model from scratch and in any case need only been done once for a given ontology model. For example in taxonomy, were someone to develop another ontology for a different group of organisms, that used the same ontology model, they could use the same mapping as the angiosperm ontology without any extra input from an IT expert.



## Chapter 9 - Conclusion

Harning [1996] identifies a number of properties for a good interface that Szekely [1996a] believes the automated domain model based tools fail on. Firstly users require windows that show information from multiple objects. As this requires some knowledge of tasks, pure data model based systems (e.g. Janus [Balzert 1996]) do not meet this property unlike our system which does manage this as the basic data entry task is known and the ontology relationships allow suitable grouping mechanisms to be used. Equally the need of users for re-structured and summarised information is met by our approach, which presents users with appropriate information due again to a known general task, the ontology relationships and the ability of specialisation users to indicate re-structured names. Lastly graphical displays are often more effective than tables and forms. In our approach collapsible tree displays were shown to be effective when we extended their display and behaviour for our use. New AIOs for specialist graphical displays of special data could also be developed and added to the data entry presentation model if there was a major need for such representation in a specific domain.

Success has been generated with domain specific approaches to automatic generation such as interfaces for remote controls [Nichols 2002]. These approaches whilst valuable in their own domains are not designed to be portable to other domains and do not address the needs of high quality data entry to databases. Finding such domain specific approaches that work could help improve knowledge of how to adapt automatic generation for related domains. We have presented one such application for taxonomic description.

Finally to return to the instance field of taxonomy in which the approach was developed, the approach has a number of implications for taxonomic working practice if it were to be adopted, other than the immediate effects for the data collector in improving their data quality. A few of these are touched upon below.

Normally only one user performs the taxonomic description task, other users are discouraged from participating as different users have difficulty communicating their descriptive concepts and tend to use different descriptive terms with varying non-explicit meanings. By enabling domain experts to clearly define the data entry options and requirements through the use of the defined descriptive elements and being able to identify locations on the structure hierarchy, this issue could be overcome allowing

## Chapter 9 - Conclusion

other users to participate based on the expert's specialisation. This would free the expert from having to describe all specimens and promote further collaborative working.

The data generated by the approach is comparable with other data that is consistent with the same ontology. More comparable data would enable users to compare data across datasets more easily and with suitable database tools, automatic comparisons could inform classification decisions and revisions. By improving the clarity of description data, there would be fewer requirements for taxonomists to return to original specimens to comprehend other taxonomists' classification concepts. By capturing the descriptive data for a database, the loss of original descriptive data could be reduced and the accessibility of the dataset significantly improved. In addition, with a suitably large database of comparable clear high-quality descriptions, users would be able in future to use electronic specimen descriptions for their projects if they had been described previously, possibly saving the work of repeating a lot of data collection. Other electronic data format taxonomic description tools [Dallwitz 1980, Maddison 1997, CBIT 2003] do not offer the same ontology-based semantic standardisation advantages or the supporting tailored and richly featured data entry interface generation of this approach.

### 9.3 Main Contributions

The main contributions of this work are:

- We have modelled taxonomic working practice and identified areas for potential IT support and research therein.
- We have proposed and demonstrated an approach which applies and extends model-based user interface development environments to:
  - Use a domain ontology for controlling description, in place of a domain model.
  - Use domain experts to specialise the domain model reducing the labour of expert developers.
  - Automatically generate an appropriate data entry interface, based on the specialised domain model, specifically for the needs of high quality data collection for databases.

## Chapter 9 - Conclusion

- We have utilised this approach to support taxonomists to address taxonomic description data quality problems, by capturing specimen description data consistent with an ontology of descriptive terminology, thus potentially improving the clarity, comparability and re-use of descriptive character data for both taxonomy and the wider biological sciences. We showed how the potential for data quality improvements could be further boosted by harnessing a domain ontology within the developed tool to inform the taxonomist during data entry. The approach also answers some specific challenges laid out in 2.6.1:
  - The approach does not add significantly to time pressures felt by taxonomists despite improving the quality of the data. This challenge was answered by utilising selection techniques for specialisation and data entry in a user-friendly appropriate interface.
  - Taxonomists' visual cognitive process is supported using multimedia aspects of the ontology in the generated interfaces.
  - Individualistic working practices are supported by giving users a good degree of autonomy within the constraints of the task and ontology.
- Our approach shows how a descriptive ontology can be used for controlling and improving data entry for high quality defined data.
  - It shows how the 'super-plant' compositional hierarchy could be effectively harnessed to generate project specific ontologies for use in controlling a data entry interface.
  - It showed where strengths and weaknesses in the developed angiosperm ontology were and how such an ontology could be improved for generating data entry interfaces.
- We have developed specific models to support our approach, which could be extended further:
  - A presentation model to effectively present an ontology for editing by domain experts, using two linked collapsible trees which have been extended to support informed usage and harness domain knowledge. The requirements of the domain ontology were identified for the approach to function.

## Chapter 9 - Conclusion

- A second presentation model to subsequently present a data entry interface for high quality data entry, emphasising the use of an ontology and multimedia definitions was demonstrated.
- A domain model which can have domain knowledge mapped to it from a descriptive ontology of the sort described and which can capture both concrete and abstract descriptive data including nuances of data such as the presence of plant structures and cloned structures.
- Using our approach, two viable specific domains for the application of model-based automatic interface generation techniques have been demonstrated.

In summary this work makes an advance in the automatic model-based generation of interfaces for high quality data entry, with specific application for addressing the problems of taxonomic description. It further contributes to how best to present ontologies to domain users for constrained editing and how to use ontologies to support high-quality data entry.

### 9.4 Future Work

Through the thesis a number of areas of future work have been identified.

The domain model itself could be enriched to handle concepts such as synonymy, improve the handling of using other description object instances as value objects and incorporate the other recommendations found in the relevant thesis sections. Equally being able to handle a history of actions for undo and redo operations would improve usability.

It has been seen that the interface relies on having a good ontology for providing suitable relationships and terms to build an effective descriptive hierarchy that is appropriate for the user's working practice and can be presented using the developed presentation models for the needs of high quality data entry. The description ontology used for taxonomists was relatively effective with some limitations. It would be valuable to continue to extend such an ontology based on the lessons of this work. The angiosperm case showed a weakness in handling of attributes (properties/state groups), and for taxonomic description we need a better classification method for them. Is this

## Chapter 9 - Conclusion

however a general problem? If so it is a limitation of the approach that would need addressed.

Equally it would be valuable to see the effect of using ontologies of the same descriptive type but for other description domains, (e.g fish, fungi). The issues of how wide a domain such an ontology should cover could be investigated and its consequent effects on our approach. With a narrower focus, the ontology could be more accurate and controlling, allowing an even greater degree of selection for specialisation but at the cost of how widely the ontology could apply. Equally with a wider focus, the ontology would place more reliance on the user defining relationships in the specialisation interface and in sorting through the alternative relationships to form a consistent description composition hierarchy. However a wider ontology would have a better amount of re-use and hence presumably could justify more development resources to ensure quality. The use of our approach was shown to be useful for ontology builders, in that domain users gave the most feedback on the ontology when they could view it in a user-friendly visualisation and had to use it for normal domain tasks. Certainly improving the development of descriptive super-entity ontologies would be valuable for developing new application domains for our approach. Current ontology building techniques are focussed on different areas.

Outside the realm of descriptive ontology, we have extended our approach to show how a XML schema for transferring knowledge can be harnessed. Certainly our approach could potentially be of use in promoting the re-use of ontologies; using only part of an ontology for an application is not an issue restricted to taxonomy, for example only part of an ontology developed for petroleum remediation [Chen 2000] was used in a system supporting the elimination of contaminants from the air [Wang 2002]. Testing what sort of other ontologies could be adapted and at what point the approach could not be sustained would be worthwhile. An enhancement for our approach would be to develop a tool to support the mapping of the domain ontologies to the abstract domain model. This would be of significant aid in expanding the approach to other domains.

One final area that could extend the applicability of our approach would be to expose in an explicit fashion the data entry presentation model to enable it to be tailored for different domains. This would make the system even closer to the model-based UIDE approaches by enacting another model to specify, with the dangers of becoming too

## **Chapter 9 - Conclusion**

generic that previous MB-UIDE approaches have floundered on. However if done in a suitably lightweight fashion, this might not impose a much greater burden on developers.

## References

- Agrawal R., Gehani N., Srinivasan J. (1990) Ode-View: The Graphical Interface to Ode, in Proceedings ACM SIGMOD Conference, May 1990
- Ahlberg, C., Shneiderman, B., Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, Proc. ACM SIGCHI '94, pp. 313-317
- Aiken S. G., Dallwitz M. J., Consaul L. L., Boles R. L., Elven R., LeBlanc M. E. (2001) Flora of the Canadian Arctic Archipelago. Volume 1. Pteridophytes and Monocotyledons: Descriptions, Illustrations, Identification and Information Retrieval. Version 16<sup>th</sup> March 2001, URL: <http://www.mun.ca/biology/delta/articf/>
- Allkin, R. (1984) Handling Taxonomic Descriptions by Computer. In: Allkin R., Bisby F.A. (eds.): Databases in Systematics. Academic Press London
- Altova (2006) XML Spy, URL: <http://www.altova.com/>
- Andrews K., Wolte J., Pichler M. (1997) Information Pyramids: A New Approach to Visualising Large Hierarchies in Late Breaking Hot Topics Proc., IEEE Visualization'97, Phoenix, Arizona, October 1997, pp. 49-52
- Bailey, J. (ed.) (1999) The Penguin Dictionary of Plant Sciences. Penguin, London
- Baldock, R. (2002) Gene Expression Information Resource Project, URL: <http://genex.hgu.mrc.ac.uk>
- Ball, T., Eick, S. (1996) Software visualization in the large, IEEE Computer, 29(4), pp. 33-43
- Balzert, H., Hofmann, F., Kruschinski, V., Niemann, C. (1996) The JANUS Application Development Environment-Generating More than the User Interface. In Proceedings of CADUI'96, Namur: Presses Universitaires de Namur, pp. 183-207
- Barclay P., Griffiths T., McKirdy J., Paton N. W., Cooper R., Kennedy J. (1999) The Teallach Tool: Using Models For Flexible User Interface Design, in proceedings of 3rd International Conference on Computer-Aided Design of User Interfaces (CADUI'99), Louvain-la-Neuve (Belgium), 21-23 October 1999
- Barclay, P., Griffiths, T., McKirdy, J., Kennedy, J., Cooper, R., Paton, N. and Gray, P. (2003). Teallach - a Flexible User-Interface Development Environment for Object Database Applications. The Journal of Visual Languages and Computing, 14(1), pp. 47-77
- Beaudoin L., Parent, M. A., Vroomen L. C. (1996) Cheops: A Compact Explorer for Complex Hierarchies, in Proc. IEEE Visualization '96, pp. 87-92, San Francisco, USA, October 27 - November 1 1996, Computer Society Press

## References

- Becker, R. A., Eick, S. G., Wilks, A. R. (1995). Visualizing network data. IEEE Transactions on Visualization and Computer Graphics, 1(1) pp. 16-28.
- Benford, S., Snowdon, D., Greenhaigh, C., Ingram, R., Knox, I., Brown, C. (1995). Vr-vibe: A virtual environment for co-operative information retrieval. In Proceedings of Eurographics '95, 30 August - 1 September, Maastricht, The Netherlands, pp. 349-360
- Bertin J. (1967) Semiology of Graphics: Diagrams, Networks, Maps, University of Wisconsin Press, Madison WI
- Bertin J., (1981) Graphics and Graphic Information Processing, De Gruyter, Berlin
- Blackwelder, R.E.(1967) Taxonomy: A text and reference book. John Wiley New York
- Boag, S., Chamberlin D., Dernadez M., Florescu D., Robie J., Siméon J. (2005), XQuery 1.0: An XML Query Language W3C Working Draft 04 April 2005, URL: <http://www.w3c.org/TR/xquery/>.
- Bodart F., Hennebert A., Leheureux J., Vanderdonckt J. (1994) Towards a Dynamic Strategy for Computer-Aided Visual Placement, in proceedings of 2<sup>nd</sup> ACM Workshop on Advanced Visual Interfaces, 1-4 June 1994, ACM Press, New York
- Bodart F., Hennebert A., Lehereux J., Vanderdonckt J. (1995) Computer-Aided Window Identification in TRIDENT in proceedings of 5<sup>th</sup> IFIP TC13 International Conference on Human-Computer Interaction, 27-29 June 1995, Chapman & Hall, London, pp. 331-336
- Boeckmann B., Bairoch A., Apweiler R., Blatter M.-C., Estreicher A., Gasteiger E., Martin M.J., Michoud K., O'Donovan C., Phan I., Pilbout S., Schneider M. (2003) The Swiss-Prot Protein Knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res. 31, pp. 365-370
- Bryce, D., Hull, R. (1986) SNAP: A Graphics-Based Schema Manager, Proc. of IEEE Int'l Conf. on Data Engineering, Feb. 1986, pp. 151-164.
- Cannon, A. J., MacDonald, S. (2001) Prometheus II – Qualitative Research Case Study, URL: [www.dcs.napier.ac.uk/~prometheus/](http://www.dcs.napier.ac.uk/~prometheus/)
- Cannon, A., Kennedy, J., Paterson, T. and Watson, M. (2004). Ontology-Driven Automated Generation Of Data Entry Interfaces. In Williams, H., Mackinnon, L. (Eds.), Key Technologies for Data Management (Proc. BNCOD21): Lecture Notes in Computer Science 3112 pp150-164. : Springer Verlag.
- Card,S. K., Pirolli, P., Mackinlay, J. (1994) The Cost-of-Knowledge Characteristic Function: Display Evaluation for Direct-Walk Dynamic Information Visualizations in Proceedings of the SIGCHI conference on Human factors in computing systems, Boston, MA, pp. 238-244
- Card, S. K., MacKinlay, J., Schneiderman, B. (Eds) (1999) Readings in Information Visualization, Morgan Kaufmann Publishers, San Fransisco



## References

- Carey, M., Haas, L. M., Schwarz, P. M., Arya, M., Cody, W. F., Fagin, R., Flickner, M., Luniewski, A. W., Niblack, W., Petkovic, D., Thomas, Williams J. H., Wimmers, E. L. (1995) Towards Heterogeneous Multimedia Information Systems: The Garlic Approach, in Proceedings of the IEEE RIDE Workshop, March 1995
- Carey, M., Haas, L., Maganty, V., Williams, J. (1996) 'PESTO: An Integrated Query/Browser for Object Databases' in Vijayarama T M, Buchmann A, Mohan C, Sarda N L, Proceedings of the 22<sup>nd</sup> International Conference on Very Large Data Bases, Mumbai, India, 3-6 Sept 1996
- Carrière, J. and Kazman, R. (1995) Interacting with huge hierarchies: Beyond cone trees. In Proceedings of Information Visualization '95 Symposium (Atlanta, GA, October 30-31, 1995), pp. 90-96. IEEE.
- Casner S. (1991) Task-Analytic Approach to the Automated Design of Graphic Presentations, ACM Transactions on Graphics, 10(2, April)
- Chen, C., (1997) Structuring and visualising the WWW by generalised similarity analysis, in Proc. Eighth ACM Conference on Hypertext (Hypertext '97), pp. 177-186, Southampton, UK, April 6-11 1997. ACM Press
- Chen, L., Chan, C.W. (2000) Ontology Design and Its Application in the Petroleum Remediation Domain. PRICAI Workshops 2000, pp.16-23
- Chi E. H., Pitkom, J., Makinlay, J., Pirolli, P., Gossweiler, R., Card, S. K. (1998) Visualizing the Evolution of Web Ecologies, CHI 98, 18-23 April 1998
- Cleveland, W. S., McGill, M. E. (1988) Dynamic Graphics for Statistics, Wadsworth and Brooks, Pacific Grove CA
- Cockburn, A. (2001) Writing Effective Use Cases, Boston, Addison-Wesley
- Colless, D. H. (1985) On 'character' and related terms. Systematic Zoology 34, pp. 229-233
- Comai, S. (2001) Graph-based GUIs for querying XML data: the XML-GL experience. SAC 2001: 269-274
- Dallwitz, M. J (1980) A General System for Coding Taxonomic Descriptions, Taxon 29, pp. 41-46
- Dallwitz, M. J., Paine, T. A., Zurcher, E. J. (1993) DELTA user's guide. A general system for processing taxonomic descriptions, 4th edition. CSIRO, Australia
- Dallwitz, M. J (1999) Desirable Attributes for Interactive Identification Programs, July 1999, URL: <http://biodiversity.uno.edu/delta/www/idcrit.htm>
- Davidson, D., Bard, J., Brune, R., Burger, A., Dubreuil, C., Hill, W., Kaufman, M., Quinn, J., Stark, M., Baldock, R. (1997) The mouse atlas and graphical gene-expression database, Seminars in Cell and Developmental Biology 8(5), pp. 509-517

## References

- Davis, P. H., Heywood, V. H. (1963) Principles of angiosperm taxonomy. Van Nostrand, Princeton, N. J
- Diedrich, J., Fortuner, R., Milton, J.(1997) Construction and Integration of Large Character Data Sets for Nematode Morpho-Anatomical Data, *Fundam appl. Nematol.*, 20(5), pp. 409-424
- Diederich, J., Fortuner, R., Milton, J. (2000) Genisys and computer-assisted identification of nematodes. *Nematology* 2, pp. 17-30
- Donath, J. S. (1995) Visual Who: Animating the affinities and activities of an electronic community, in Proc. ACM MultiMedia '95, pp. 99-108, San Francisco, USA, November 5-9 1995. ACM Press
- Eick, S. G., Steffen, J. L. (1992). Visualizing code profiling line oriented statistics. In Proceedings of Visualization '92, pp. 210-217. IEEE.
- Elwert, T., Schlungbaum, T. (1995)Modelling and Generation of Graphical User Interfaces in the TADEUS Approach, in Design Specification, and Verification of Interactive Systems (eds. P. Palanque and R. Bastide). Wien, Springer, pp. 193–208
- Engler, A., *Das Pflanzenreich (1900-1953)*, Verlag von Wilhelm Enelmann, Leipzig & Berlin, 1900-1953
- Fairchild, K., Poltrok, S., Furnas, G. (1988). SemNet: Three-Dimensional Graphic Representations of Large Knowledge Bases, pp. 201-233. Lawrence Erlbaum.
- Farquhar, A., Fikes, R., Rice, J. (1997) The Ontolingua Server: A tool for collaborative ontology construction, *International Journal of Human Computer Studies*, 46(6), pp 702-728
- Feiner, S. K. (1988) A grid-based approach to automating display layout in proc graphics interface pp. 192-197 June 1988
- Feiner, S. K., McKeown, K., (1990), Co-ordinating Text and Graphics in Explanation Generation, in Proceedings of AAIA-90, Boston, 19 July – 3 Aug 1990, pp. 442-449
- Fensel, D., Horrocks, I., Harmelen, F. V., Decker, S., Erdmann, M., Klein M. (2000) OIL in a nutshell In: Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000), R. Dieng et al. (eds.), Lecture Notes in Artificial Intelligence, LNAI, Springer-Verlag, October 2000.
- Fowler, R. H., Fowler, W. A. L. (1996) Williams J L, Document Explorer visualizations of WWW document and term Spaces. Department of Computer Science, Technical Report, NAG9-551, #96-6
- Fowler, R. H., Wilson, B. A., Fowler, W. A. L. (1992) INFORMATION NAVIGATOR: An Information System using Associative Networks for Display and Retrieval Department of Computer Science, Technical Report NAG9-551, #92-1

## References

- Frese, M., Brodbeck, F., Heinbokel, T., Mooser, C., Schleiffenbaum, E., Thiemann, P. (1991) Errors in training computer skills: On the positive function of errors. *Human-Computer Interaction* 6, 1, pp. 77-93
- Fristrup, K. (1992) Character: current usages. In: Keller, E.F., Lloyd, E.A. (eds.): *Keywords in evolutionary biology*. Harvard University Press Cambridge, pp.45-51
- Furnas, G. W., (1986) "Generalized Fisheye Views" in *Proceedings of ACM SIGCHI '86 Boston, MA, 16-23, April 1986*
- Furnas, G. W., Zacks, J. (1994). Multitrees: Enriching and reusing hierarchical structures. In *Human Factors in Computing Systems CHI '94 Conference Proceedings*, pp. 330-336.
- Furnas, G. W. (1997) Effective View Navigation in *Proceedings of the SIGCHI conference on Human factors in computing systems, Atlanta, GA, pp. 367 – 374*
- Furtado, E., Furtado, V., Sousa, K. S., Vanderdonckt, J., Limbourg, Q. (2004) KnowiXML: a knowledge-based system generating multiple abstract user interfaces in USIXML. In *Proceedings of the 3rd Annual Conference on Task Models and Diagrams (Prague, Czech Republic, November 15 - 16, 2004)*. TAMODIA '04, vol. 86. ACM Press, New York, NY, pp. 121-128
- Gajos, K., Weld, D. S. (2004) SUPPLE: automatically generating user interfaces. In *Proceedings of the 9th international Conference on intelligent User interface (Funchal, Madeira, Portugal, January 13 - 16, 2004)*. IUI '04. ACM Press, New York, NY, pp. 93-100.
- Gennari, J., Musen, M.A., Ferguson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F. (2002) Tu, S.W., *The Evolution of Protégé: An Environment for Knowledge-Based Systems Development*, URL: [http://www.smi.stanford.edu/pubs/SMI\\_Reports/SMI-2002-0943.pdf](http://www.smi.stanford.edu/pubs/SMI_Reports/SMI-2002-0943.pdf), <http://protege.stanford.edu/index.html>
- Gershon, N., Card, S. K. and Eick, S. G.,(1998)*Information Visualization Tutorial*, in *Proc. ACM CHI '98: human factors in computing systems*, pp. 109-110, Los Angeles, USA, April 18-23, ACM Press
- Gillo, X., Vanderdonckt, J. (1994) Visual Techniques for Traditional and Multimedia Layouts, in *Proceedings of 2<sup>nd</sup> Workshop on Advanced Visual Interfaces, Bari, 1-4 June 1994*, ACM Press, New York, pp. 95-104
- Goldstein, J., Roth, S. F. (1994) Using Aggregation and Dynamic Queries for Exploring Large Data Sets, in *Human Factors in Computing Systems CHI '94 Conference Proceedings*, pp. 23-29
- Graham, M., Kennedy, J. B. (2001) Combining linking & focusing techniques for a multiple hierarchy visualisation, in *Proc. IV 2001*, pp. 425-432, London, UK, July 25-27, 2001. IEEE Computer Society Press

## References

- Gray, P., Cooper, R., Kennedy, J., McKirdy, J., Barclay, P., Griffiths, T. (1998), A Lightweight Presentation Model for Database User Interfaces, ERCIM'98, Stockholm, October 1998
- Griffiths, T., McKirdy, J., Forrester, G., Paton, N., Kennedy, J., Cooper, R., Barclay, P. J., Goble, C., Gray, P. (1998a) Exploiting Model Based Techniques for User Interfaces to Databases, in procs. of Visual Databases 4 (VDB4), L'Aquila, Italy, May, 1998
- Griffiths, T., Barcla, P. J., McKirdy, J., Paton, N. W., Gray, P. D., Kennedy, J. B., Cooper, R., Goble, C., West, A., Smyth, M. (1999) Teallach: A Model-Based User Interface Development Environment for Object Databases, in procs. of User Interfaces to Data Intensive Systems (UIDIS'99), Edinburgh, Scotland, 5-6 September, pp. 86-96, IEEE Computer Society Publishers, Norman W. Paton and Tony Griffiths (eds.)
- Griffiths, T., Barclay, P., Paton, N., McKirdy, J., Kennedy, J., Gray, P., Cooper, R., Goble, C., Pinherio da Silva, P. (2001). Teallach: A Model-Based User Interface Development Environment for Object Databases. *Interacting with Computers*, 14/1, pp. 33-72
- Gruber, T. R.(1993a) A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition* 5, pp. 199-220
- Gruber, T. R. (1993b) Towards principles for the design of ontologies used for knowledge sharing, presented at Padua workshop on Formal Ontology, March 1993
- Guarino, N., Giaretta, P. (1995) Ontologies and Knowledge Bases: Towards a Terminological Clarification. In Mars, N.J.I. (ed.), *Towards Very Large Knowledge Bases*, IOS Press
- Harning, M.B. (1996) An Approach to Structured Display Design - Coping with Conceptual Complexity, *Proceedings of 2nd International Workshop on Computer-Aided Design of User Interfaces CADUI '96*, pp. 121-140
- Hearst, M. A. (1995) TileBars: Visualization of Term Distribution Information in Full-Text Information Access, in *Proc. ACM CHI '95*, pp. 55-66, Denver, Colorado, USA, May 7-11 1995, ACM Press
- Hendley, R., Drew, N., Wood, A., and Beale, R. (1995). Narcissus: Visualizing information. In *Proceedings of Information Visualization '95 Symposium* (Atlanta, GA, October 30-31, 1995), pp. 90-96. IEEE
- Huang. M. L. Eades, P., Wang, J. (1998) On-line Animated Visualization of Huge Graphs using a Modified Spring Algorithm, *Journal of Visual Languages and Computing*, 9 (6), pp. 623-645
- Huang, T. C. (ed.) (1993) *Flora of Taiwan*, 2nd Edition Volume 3. Taipei.
- Hyam, R., (2002), pers com., Unpublished.
- IBM (2000) IBM Visualization Data Explorer QuickStart Guide, URL: <http://www.research.ibm.com/dx/docs/legacyhtml/pages/qikgu010.htm>

## References

- Index Filicum(2004) Index Filicum, The International Plant Names Index URL: [www.ipni.org](http://www.ipni.org)
- Index Kewensis (2004) Index Kewensis, The International Plant Names Index: About the Index Kewensis, URL: [http://www.ipni.org/ik\\_blurb.html](http://www.ipni.org/ik_blurb.html)
- Inselberg, A., Dimsadale, B., (1990) Parallel Co-ordinates: A Tool for Visualising Multi-Dimensional Geometry, Proceedings of IEEE Visualization '90Conference, Los Alamitos CA, pp. 361-375
- Jacobs, M. (1969) Large Families – Not Alone!, Taxon 18, pp253-262, June 1969
- Janssen, C., Weisbecker, A., Zeigler, J. (1993) Generation user interfaces from data models and dialogue net specifications. In: Proceedings of INTERCHI '93. Addison-Wesley, pp. 418-423
- Jeong, C. S., Pang, A. (1998) Reconfigurable Disc Trees for Visualizing Large Hierarchical Information Space, in Proc. IEEE InfoVis '98, pp. 19-25, Research Triangle, North Carolina, USA, October 19-20 1998. Computer Society Press
- Johnson, B., Shneiderman, B. (1991) Treemaps: A Space-Filling Approach to the Visualisation of Hierarchical Information Structures, Proceedings of IEEE Visualization '91 Conference, San Diego, pp. 284-291
- Keim, D. A., Kreigel, H.(1999) VisDB: Database Exploration Using Multimedia Visualisation in Card S K, MacKinlay, J., Schneiderman, B. (Eds), Readings in Information Visualization, Morgan Kaufmann Publishers, San Fransisco
- Kelly, M. G., Bayer, M. M., Hürlimann, J., Telford, R. J. (2002)Human error and quality assurance in diatom analysis In: Automatic Diatom Identification. Series in Machine Perception and Artificial Intelligence. Eds. du Buf, J. M. H., Bayer, M. M., pp. 75-92, World Scientific
- Kennedy, J., Hyam, R., Kukla, R. and Paterson, T. (2006). A Standard Data Model Representation for Taxonomic Information. *to appear in OMICS: A Journal of Integrative Biology*
- Kukla, R., Kennedy, J., Paterson, T. (2005) Taxonomic Concept Transfer Schema URL: <http://tdwg.napier.ac.uk/index.php>
- Kuntz, M.,Melchert, R. (1989) Pasta-3: A Complete Integrated Graphical Direct Manipulation Interface for Knowledge Bases, IFIP Congress 1989, pp.547-552
- Lamping, J., Rao, R. (1994). Laying out and visualizing large trees using a hyperbolic space. In Proceedings of UIST '94, pp. 13-14.
- Lamping, J., Rao, R., Pirolli, P. (1995) A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies in Proceedings of ACM SIGCHI '95, Denver, CO, 401408, May 1995
- Lamping J., Rao R. (1996) Visualising Large Trees Using the Hyperbolic Browser, CHI 96, 13-18 April 1996

## References

- Lawrence, G. H. M. (1951) *Taxonomy of Vascular Plants*. The Macmillan Company, New York
- Lewis, C. (1982). Using the 'thinking-aloud' method in cognitive interface design. Research Report RC9265, IBM T J Watson Research Center, Yorktown Heights, NY.
- Lonczewski, F., Schreiber, S.(1996) The FUSE-System: An Integrated User Interface Design Environment. In: J. Vanderdonck (ed.): *Computer-Aided Design of User Interfaces*. Namur: Namur University Press, pp. 57-74
- Lydon, S. J., Wood, M. M. (2003) Huxley R, Sutton D, Data Patterns in Multiple Botanical Descriptions: implications for automatic processing of legacy data, in *Systematics and Biodiversity* 1 (2), pp. 151-157
- Lucas, P., Roth, S. F. (1996) *Exploring Information with Visage*, Video Proceedings of the CHI '96 Conference on Human Factors in Computing Systems. New York: ACM
- Middleton, D. (2000) Revision of *Alyxia* (Apocynaceae), Part 1: Asia and Malesia. *Blumea* 45, pp.1-146
- Middleton, D. (2002) Revision of *Alyxia* (Apocynaceae), Part 2: Australia and Pacific Islands. *Blumea* 47, pp. 1-93
- Multiflora Project (2003) URL: <http://www.cs.man.ac.uk/ai/Software/MultiFlora/>
- Machatschki-Laurich, B. (1926). Die Arten der Gattung *Biscutella* L. sectio *Thlaspidium* (Med.) DC. *Botanisches Archiv Koenigsberg* 13: pp. 1–115
- Mackinlay, J. D., Robertson, G. G., and Card, S. K. (1991). The perspective wall: Detail and context smoothly integrated. In CHI '91 Conference on Human Factors in Computing Systems, pp. 173-179.
- MacKinlay, J. (1999) Automating the Design of Graphical Presentations of Relational Information in Card S K, MacKinlay J, Schneiderman B (Eds), *Readings in Information Visualization*, Morgan Kaufmann Publishers, San Francisco
- Maddison, D.R., Swofford, D.L., Maddison, W.P.(1997) NEXUS: An extensible file format for systematic information. *Systematic Biology* 46, pp. 590-621
- McDonald, S. M., Raguenaud, C., Pullan, M. R., Kennedy, J., B., Russell, G., Watson, M. F. (2002) The Prometheus II Description Model: an objective approach to representing taxonomic descriptions. URL: [http://www.dcs.napier.ac.uk/~prometheus/prometheus\\_2/publications2.html](http://www.dcs.napier.ac.uk/~prometheus/prometheus_2/publications2.html)
- Menglan, S., Zehui, P., Fading, P., Watson, M. F., Cannon, J. F. M., Kljuykov, E. V., Phillippe, L. R., Pimenov, M. G. (2004) *Apiaceae* (Umbelliferae). In: Wu, Z. Y., Raven, P. H. (eds.), *Flora of China*, 14. Science Press, Beijing & Missouri Botanical Garden Press, Saint Louis.
- Mihalisin, T., Timlin, J., Schwegler (1991) Visualizing Multivariate Functions, Data and Distributions, *IEEE Computer Graphics and Applications*, 11(13), pp. 28-35

## References

- Miller, G. A. (1956) The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information, *Psychological Review*, 63, pp. 81-97
- Mitchel, K. J. (1997) Three Dimensional Database Environments, PhD Thesis, Napier University
- Molina, P. A. (2004) Review to Model-Based User Interface Development Technology. In Trætteberg, H., Molina, P., Nunes, N., (Eds.): Proceedings of the First International Workshop on Making model-based user interface design practical: usable and open methods and tools, Funchal, Madeira, Portugal, January 13, 2004. CEUR Workshop Proceedings 103 CEUR-WS.org
- Motro, A., D'Atri, A., Tarantino, L. (1988) The Design of KIVIEW: An Object Orientated Browser, Proceedings of the 2<sup>nd</sup> Internatinal Conference on Expert Database systems, April 1988.
- Mukherjee, P. K., Constance, L. (1993) *Umbelliferae (Apiaceae) of India*, Oxford & IBH Publishing Company, New Dehli
- Myers, B., Hudson, S. E., Pausch, R. (2000) Past, present, and future of user interface software tools. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (Mar. 2000), 3-28
- Newman, M. (2001) pers com, unpublished.
- Nichols, J., Myers, B. A., Higgins, M., Hughes, J., Harris, T. K., Rosenfeld, R., and Pignol, M. (2002). Generating remote control interfaces for complex appliances. In Proceedings of the 15th Annual ACM Symposium on User interface Software and Technology (Paris, France, October 27 - 30, 2002). UIST '02. ACM Press, New York, NY, pp. 161-170
- Nichols, J., Myers, B. A., Litwack, K. (2004) Improving automatic interface generation with smart templates. In Proceedings of the 9th international Conference on intelligent User interface (Funchal, Madeira, Portugal, January 13 - 16, 2004). IUI '04. ACM Press, New York, NY, pp. 286-288.
- Nichols, J., Faulring, A. (2005) Automatic Interface Generation and Future User Interface Tools in proceedings of the workshop on the future of user interface design tools CHI 2005, Portland Or.
- Nielsen, J. (1992). Finding usability problems through heuristic evaluation, *Proc. ACM CHI'92*, (3-7 May, Monteray, CA), pp. 373-380.
- Nielsen, J. (1993) *Usability Engineering*, Academic Press, London, ISBN 0125184069
- Nielsen, J. (1994) Heuristic evaluation” in Nielsen, J., and Mack, R. L., (Eds.), *Usability Inspection Methods*. John Wiley & Sons, New York, pp. 25-62
- Norman, D. (1988) *The design of everyday things*. New York, NY: Doubleday
- Novak, G. S. (2003) Novak's rule, URL: <http://www.cs.utexas.edu/users/novak/>

## References

- Oviatt, S. L. (1996), Multimodal Interfaces for Dynamic Interactive Maps, Proceedings of the CHI'96 Conference on human Factors in Computing Systems, ACM, New York
- Paterson, T., Cannon, A., Raguenaud, C., Russell, G., Armstrong, K., MacDonald, S., Pullan, M., Watson, M. and Kennedy, J. (2003). A Methodology for Composing Well-Defined Character Descriptions, (*Unpublished*) URL:  
<http://www.soc.napier.ac.uk/publication/op/getpublication/publicationid/5947465>
- Paterson, T., Kennedy, J., Pullan, M. R., Cannon A. J., Armstrong, K., Watson M. F., Raguenaud C., McDonald S. M., Russell G. (2004) A Universal Character Model and Ontology of Defined Terms for Taxonomic Description, DILS 2004, LNBI 2994, pp. 63-78
- Pederson, D. O. (2006) Center for Electronic Systems Design, University of California, Berkeley, diva.sketch demo, URL:  
<http://embedded.eecs.berkeley.edu/diva/demo/sketch.html>
- Penner, R. R., Steinmetz, E. S. (2002) Model-based automation of the design of user interfaces to digital control systems. IEEE Transactions on Systems, Man, and Cybernetics, Part A 32(1): pp. 41-49
- Petropoulos, M., Papakonstantinou, Y., Vassalos, V. (2005) Graphical query interfaces for semistructured data: the QURSED system. ACM Trans. Inter. Tech. 5, 2 (May. 2005), pp. 390-438
- Pirolli, P., Card, S., Van der Wege, M. (2001) Visual Foraging in a Focus and Context Visualization, Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 506 – 513, March 2001
- Prometheus (2001) Prometheus 2, Proposal, URL:  
[http://www.dcs.napier.ac.uk/~prometheus/prometheus\\_2/descriptionII.html](http://www.dcs.napier.ac.uk/~prometheus/prometheus_2/descriptionII.html)
- Prometheus II. (2005) Prometheus II Project, URL:  
<http://www.dcs.napier.ac.uk/~prometheus/>
- Puerta, A. R., Eriksson, H., Gennari, J. H., Mussen, M. A. (1994) Beyond Data Models For Automated User Interface Generation. In People and Computers IX HCI'94, pp. 353-366
- Puerta, A. R. Maulsby, D. (1997) Management of interface design knowledge with MOBI-D. In Proceedings of the 2nd international Conference on intelligent User interfaces (Orlando, Florida, United States, January 06 - 09, 1997). J. Moore, E. Edmonds, and A. Puerta, Eds. IUI '97. ACM Press, New York, NY, pp. 249-252
- Pullan, M. R., Watson, M. F., Kennedy, J. B., Raguenaud, C., Hyam, R., (2000) The Prometheus Taxonomic Model: a practical approach to representing multiple taxonomies, Taxon 49(1), pp. 55-75, February 2000, ISSN 0040-0262
- Pullan, M., Armstrong, K., Paterson, T., Cannon, A., Kennedy, J. (2005). The Prometheus Description Model: An examination of the taxonomic description building process and its representation. Taxon, 543, pp. 751-765.



## References

- Raggett, D., Le Hors, A., Jacobs, I. (1999) XHTML HTML 4.01 Specification W3C Recommendation 24 December 1999, URL: <http://www.w3.org/TR/html4/>
- Rao, R., Card, S. K. (1994). The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In Human Factors in Computing Systems CHI '94 Conference Proceedings, pp. 318-322.
- Rheingold, H.(1991) Virtual Reality, the Revolutionary Technology of Computer Generated Artificial Worlds - and How it Promises and Threatens to Transform Business and Society, Summit Books
- Robertson, G. G., Mackinlay, J. D., Card, S. K. (1991) Cone trees: Animated 3D visualizations of hierarchical information. in Proceedings of the ACM CHI'91 Conference on Human Factors in Computing Systems, Information Visualization. ACM, New York, NY, pp. 189-194,
- Rogers, T., Cattell, R.(1988) Entity-Relationship Database User Interfaces in Stonebraker M (Ed), Readings in Database Systems, Morgan Kaufman
- Roth, S. F., Mattis, J. (1990) Data Characterisation for Intelligent Graphics Presentation, Proceedings of CHI'90, ACM Conference on Human Factors in Computing Systems, New York
- Roth, S. F., Chuah, M. C., Kerpedijev, S., Kolojejchick, J., Lucas, P. (1997) Towards an Information Visualization Workspace: Combining Multiple Means of Expression' Human-Computer Interaction Journal, Volume 12, Numbers 1 & 2, pp.131-185
- Schaffer, D., Zuo, Z., Greenberg, S., Bartam, L., Dill, J., Dubs, S., Roseman, M. (1996) Navigating Hierarchically Clustered Networks through Fisheye and Full-zoom methods, ACM Transactions on Computer-Human Interaction, Vol 3, No 2, June 1996, pp. 162-188
- Schneiderman, B. (1983) Direct Manipulation: A Step Beyond Programming Languages, IEEE Computer, 16
- Schneiderman, B. (1998) Designing the user interface: Strategies for effective human-computer interaction (3rd ed.), Reading, MA: Addison-Wesley Publishing
- Schneiderman, B. (1999b) Dynamic Queries for Visual Information Seeking in Card S K, MacKinlay J, Schneiderman B (Eds), Readings in Information Visualization, Morgan Kaufmann Publishers, San Francisco
- Schneiderman, B. (1999a) Dynamic Queries, Starfield Displays, and the Path to Spotfire, URL: <http://www.cs.umd.edu/hcil/spotfire>
- Schlunbaum, E. (1996) Model-based User Interface Software Tools - Current state of declarative models. Technical Report 96-30, Graphics, Visualization and Usability Center, Georgia Institute of Technology, Atlanta
- SEEK (2006) Science Environment for Ecological Knowledge, URL: <http://www.seek.ecoinformatics.org/>

## References

- Shan, R. H., L. Constance, L. (1951) The genus *Sanicula* (Umbelliferae) in the Old World and New. University of California Publication in Botany. University of California Press, Berkeley and Los Angeles.
- Sivarajan, V. V. (1991) Introduction to the Principles of Plant Taxonomy. Cambridge University Press Cambridge
- Smith, M. (2003) Ontology, in L. Floridi (ed.), Blackwell Guide to the Philosophy of Computing and Information, Oxford: Blackwell, pp.155–166
- Stace, C. A. (1989) Plant Taxonomy and Biosystematics. Cambridge University Press
- Stanford Medical Informatics (2005a) Protégé-Frames User's Guide, URL: <http://protege.stanford.edu/doc/users.html>
- Stanford Medical Informatics (2005b) Getting Started with Protégé-Frames, URL: <http://protege.stanford.edu/doc/users.html>
- Stanford Medical Informatics (2006) Protégé Frames, URL: <http://protege.stanford.edu/overview/protege-owl.html>
- Stearne, W. T. (1983) Botanical Latin: History, Grammar, Syntax, Terminology and Vocabulary. David & Charles, London
- Stuessy, T.F. (1990) Plant taxonomy: the systematic evaluation of comparative data. Columbia University Press New York
- Szekely, P., Luo, P., Neches, R. (1992) Facilitating the Exploration of Interface Design Alternatives: The HUMANOID Model of Interface Design", Proc. CHI 92, pp. 507-515
- Szekely, P. (1996a) Retrospective and challenges for model-based interface development. In F. Bodart and J. Vanderdonck, editors, Design, Specification and Verification of Interactive Systems '96, pp. 1-27, Wien, Springer-Verlag
- Szekely, P., Sukaviriya, P., Castells, P., Muhtkumarasamy, J., Salcher, E. (1996b) Declarative Interface Models For User Interface Construction Tools: The MASTERMIND Approach, in Engineering For Human-Computer Interaction
- TDWG (International Working Group on Taxonomic Databases) subgroup (2000) Structure of Descriptive Data.: Subgroup session report at the TDWG meeting in Frankfurt URL: [www.tdwg.org/tdwg2000/sddreport.htm](http://www.tdwg.org/tdwg2000/sddreport.htm)
- TDWG (International Working Group on Taxonomic Databases) (2005), URL: [www.tdwg.org](http://www.tdwg.org)
- Tognazzini, B. (1992) Tog on Interface, Addison-Wesley, Reading, ISBN: 0201608421
- Tognazzini, B., (2003), First Principles of Interaction Design, URL: <http://www.asktog.com/basics/firstPrinciples.html>
- Trætteberg, H., Molina, P. J., Nunes, N. J. (2004) Making model-based UI design practical: usable and open methods and tools. In Proceedings of the 9th international

## References

- Conference on intelligent User interface (Funchal, Madeira, Portugal, January 13 - 16, 2004). IUI '04. ACM Press, New York, NY, pp. 376-377
- Tufte, E. R. (1983) *The Visual Display of Quantitative Information*, CT: Graphics Press, Cheshire
- Tukey, J. W. (1977) *Exploratory Data Analysis*, Addison-Wesley, Reading MA
- Turo, D., Johnson, B. (1992) Improving the Visualization of Hierarchies with Treemaps: Design Issues and Experimentation, in Proc. IEEE Visualization '92, pp. 124-131, Boston, Massachusetts, USA, October 19-23, 1992. Computer Society Press
- Vanderdonckt, J., Ouedraogo, M., Ygquier, B. (1994) A Comparison of Placement Strategies for Effective Visual Design, in Proceedings of HCI'94 "People and Computers IX", G. Cockton, S.W. Draper, G.R.S. Weir (Eds.), Cambridge University Press, Cambridge, pp. 125-143
- Vanderdonckt, J. (1995) Knowledge-Based Systems for Automated User-Interface Generation: the TRIDENT Experience', in Proceedings of CHI 95 Workshop on Knowledge-Based Support for the User Interface Design Process, Denver, 7-9 May 1995, pp. 21-33
- Vanderdonckt, J., Bodart, F. (1996) The Corpus Ergonomicus: A Comprehensive and Unique Source for Human-Machine Interface in Ozok A F, Sanvendy G (Eds), Proceedings of 1<sup>st</sup> International Conference on Applied Ergonomics ICAE 96, Istanbul, 21-24 May 1996, USA Publishing, West Lafayette, pp. 62-169
- Wang, X., Chan, C. W., Hamilton, H. J. (2002) Design Of Knowledge-Based Systems With The Ontology-Domain-System Approach. SEKE 2002, pp. 233-236
- Watson, L. (1971) Basic Taxonomic Data: The Need for Organisation over Presentation and Accumulation, *Taxon* 20(1), pp. 131-136, February 1971
- Wechsler, D. (1997) *Wechsler Adult Intelligence Scale-III*. The Psychological Corporation, San Antonio, TX
- Weiss-Lijn, M., McDonnell, J. T., James, L. (2001) Visualising Document Content with Metadata to Facilitate Goal-Directed Search. In Proceedings of the Fifth international Conference on information Visualisation (Iv'01) (July 25 - 27, 2001). IV. IEEE Computer Society, Washington, DC, 71.
- Wiecha, A., Bennett, W., Boies, S., Gould, J., Greene, S. (1990) ITS: A Tool For Rapidly Developing Interactive Applications. *ACM Transactions on Information Systems* 8(3), July 1990, pp. 204-236
- Wiley, E. O. (1981) *Phylogenetics: the Theory and Practice of Phylogenetic Systematics*. John Wiley New York
- Wise, J. A., Thomas, J. J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., and Crow, V. (1995) Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In Proceedings of Information Visualization '95 Symposium (Atlanta, GA, October 30-31, 1995), pp. 51-58. IEEE.

## References

- Wood, A., Drew, N., Beale, R., Hendley, B. (1995). Hyperspace: Web browsing with visualisation. In Third International World-Wide Web Conference Poster Proceedings, April 10-14 1995, Darmstadt, Germany, URL: <http://www.cs.bham.ac.uk/~amw/hyperspace/www95/>.
- Xiong, R., Smith, M., Drucker, S. (1998) Visualizations of Collaborative Information for End-Users, Technical Report MSR-TR-98-52, Microsoft Research
- Yang, C., Kao, C. (1999) Visualizing large hierarchical information structures in digital libraries, In Proceedings of the Second Asian Digital Library Conference, Taipei, Taiwan, ROC, pp. 217-225, November 8-9, 1999
- Zloof, M. M. (1998) Selected Ingredients in End-User Programming', in T Catarci, MF Costabile, G Santucci, L Tarantino (Eds), Proceedings of the Working Conference on Advanced Visual Interfaces, L'Aquila, Italy, 24-27 May 1998

## Appendix A: Use Cases

### USE CASES: SCORING TAXONOMIC DESCRIPTION DATA FOR SPECIMENS IN A PLANT TAXONOMY PROJECT

#### Use Case Scenarios:

##### 1. Current usage

- Upper Level
  - Level 2: Create pro forma
    - Level 3: Add description section to pro forma
      - Level 4: Add label for feature/character into pro forma
  - Level 2: Scoring taxonomic description for a specimen
  - Level 2: Capture additional description details for all specimens

##### 2. General computer interface (abstract)

- Upper Level
  - Level 2: Create pro forma
    - Level 3: Add description section to pro forma
      - Level 4: Add quantitative description character
        - Level 5: Find structure definition
      - Level 4: Add qualitative description character
        - Level 5: Find defined states
  - Level 2: Scoring taxonomic description for a specimen
    - Level 3: Scoring a quantitative description character
    - Level 3: Scoring a qualitative description character

##### 3. Computer interface (guidance heavy interface)

- Upper Level
  - Level 2: Create pro forma
    - Level 3: Add description section to pro forma
  - Level 2: Scoring taxonomic description for a specimen

##### 4. Computer interface (multi-pane interface)

- Upper Level
  - Level 2: Create pro forma
    - Level 3: Add description section to pro forma
      - Level 4: Add qualitative description character
  - Level 2: Scoring taxonomic description for a specimen

## Use Case 1: Current Usage

### Upper Level:

**Primary Actor:** Taxonomist (T)

**Secondary Actors:** None

**System Type:** Current Usage

**System Scope:** Capture taxonomic descriptions for all specimens in a project

**Pre-requisites:**

- Crude sort completed.
- Basic knowledge gathering -> Familiar with specimen and project subject area
- WP package started.

**Goal:** Capture all relevant descriptive data for all specimens in project

T creates WP pro forma document with labels for features of interest\*. T prints a copy of the pro forma for each specimen. T takes a random first specimen and records details of the specimen on the pro forma, filling in the various label sections\*. T repeats process for each specimen. (*Exception: T sees feature on specimen that requires to be scored. T adds feature to Pro forma. T records feature for all previously scored specimens*). T reviews completed pro formas. T adds further details to each scored specimen pro forma\*, until T believes sufficient details recorded to usefully support the taxonomic process of delimiting groups, creating a taxonomic hierarchy and writing taxon descriptions.

### Level 2

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Current Usage

**System Scope:** Create pro forma to score taxonomic description specimen details upon for a taxonomic project.

**Pre-requisites:**

- As upper level

**Goal:** Create pro forma for project, for use in capturing specimen descriptive data

T starts new document. T selects bold typeface and adds title to top of document. Under the title, T adds Specimen ID section heading. T deselects bold typeface. T adds detailed labels with blank space for later writing, for recording specimen ID data (*accession numbers, etc*). T adds section separator line.

T adds description sections with feature labels. T reviews whole pro forma, checking to insure all relevant features are included. T uses WP formatting tools to ensure presentation is relatively clear. T uses WP print command to print 50 copies (1 for each specimen in the study).

### Level 3

**Primary Actor:** Taxonomist

**Secondary Actors:** None

## Appendix A - Use Cases

**System Type:** Current Usage

**System Scope:** Add description section into pro forma.

**Pre-requisites:**

- As upper level
- Begun Pro forma WP document
- Have idea for feature/character

**Goal:** Add description section to pro forma

T selects header-formatting style. T adds section heading. T selects general label formatting style. T adds individual feature labels with blank space for later writing until all considered labels are added. T rechecks existing descriptions of the general description section in other publications and compares features recorded against the ones just added to the pro forma. (*Exception: T sees other features in existing descriptions that T wishes to use: T adds more labels.*) (*Exception: T finds different features that leads T to want to revise the features in the pro forma: T edits feature labels in section*). T adds section separator line.

**Level 2/3** (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Current Usage

**System Scope:** Create pro forma to score taxonomic description specimen details upon for a taxonomic project.

**Pre-requisites:**

- As upper level

**Goal:** Create pro forma for Umbrellifer project, for use in capturing specimen descriptive data

T starts new document. T selects bold type face and adds title 'Umbellifer Proforma' to top of document. Under the title, T adds Specimen ID section heading. T deselects bold type face. T adds detailed labels with blank space for later writing, for recording specimen ID data (*accession numbers, etc*). T adds section separator line.

T selects bold type face and adds section heading 'General Features'. T deselects bold type face. T adds detailed labels for individual features ('*Habit*', '*Stem*', '*Other*') with blank space for later writing. T adds section separator line.

T selects bold type face and adds section heading 'Leaf'. T looks up relevant existing publications with leaf descriptions of similar plant types and looks at what features are recorded in them. T deselects bold type face. T adds detailed labels for individual features ('*shape*', '*indumentum*', '*petiole*', '*stipules*', '*texture*', '*size*', '*base*', '*apex*', '*other*') with blank space for later writing. T rechecks existing leaf descriptions in other publications and compares features recorded against the ones just added to the pro forma. T adds section separator line.

T selects bold type face and adds section heading 'Inflorescence'. T looks up relevant existing publications with inflorescence descriptions of similar plant types and looks at what features are recorded in them. T deselects bold type face. T adds detailed labels for individual features ('*type*', '*bracts*', '*bractioles*', '*flower*

## Appendix A - Use Cases

*colour', 'sepals', style', etc, 'other')* with blank space for later writing. T rechecks existing inflorescence descriptions in other publications and compares features recorded against the ones just added to the pro forma. T adds section separator line.

T selects bold type face and adds section heading 'Fruit'. T looks up relevant existing publications with fruit descriptions of similar plant types and looks at what features are recorded in them. T deselects bold type face. T adds detailed labels for individual features (*'shape', 'ornamentation', 'size', 'vittae', 'other'*) with blank space for later writing. T rechecks existing fruit descriptions in other publications and compares features recorded against the ones just added to the pro forma. T adds section separator line.

T reviews whole pro forma, checking to insure all relevant features are included. T uses WP formatting tools to ensure presentation is relatively clear. T uses WP print command to print 50 copies (1 for each specimen in the study).

### Level 4

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Current Usage

**System Scope:** Add label for feature/character into pro forma.

**Pre-requisites:**

- As upper level
- Section created
- Have idea for feature/character

**Goal:** Add identifiable label to pro forma

T decides on name for label and types it into pro forma, under previous label. T leaves space for writing scores onto pro forma and moves cursor to place for next label.

### Level 4 (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Current Usage

**System Scope:** Add label for feature/character into pro forma.

**Pre-requisites:**

- As upper level
- Section created
- Have idea for feature/character

**Goal:** Add identifiable label to pro forma

T decides on name 'texture' for label and types it into pro forma, under previous label 'shape'. T leaves space for writing scores onto pro forma and moves cursor to place for next label.

### Level 2



## Appendix A - Use Cases

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Current Usage

**System Scope:** Scoring taxonomic description for a specimen based on a pro forma.

**Pre-requisites:**

- Crude sort completed.
- Basic knowledge gathering -> Familiar with specimen and project subject area
- Pro forma created
- Specimen and measuring tools available

**Goal:** Score all relevant descriptive data for one specimen

T takes copy of empty pro forma. T reviews pro forma. T opens specimen and looks at it, gaining impression of general structure. T records ID data from Specimen Label, onto pro forma next to the pro forma labels such as *accession number*.

T looks at first description section and the first label, then examines specimen. T decides what characteristics the specimen exhibits that are relevant to the feature label and makes a note of them on the pro forma in the space next to the label (*Exception: T find no characteristics to comment upon or the feature is not present: T leaves area next to label blank and moves to next label*) (*Exception: T unable to fully express unusual characteristic in words: T uses sketch*)(*Exception: T uses sketch as shorthand for complex characteristic*). T re-examines the specimen, and notes observations on pro forma next to label, until T believes all relevant characteristics for the label are recorded. T then moves onto next label and repeats process until T reaches end of the pro forma.

T then scans the specimen, checking no outstanding or unusual features have been missed. (*Exception: if find non-recorded feature, record it on pro forma and decide if merits recording for other specimens. If so return to previously recorded specimens and record new feature in ad hoc blank space.*) T then puts pro forma down, closes specimen and returns it to its pile. T then moves onto next specimen.

### Level 2 (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Current Usage

**System Scope:** Scoring taxonomic description for a specimen based on a pro forma.

**Pre-requisites:**

- Crude sort completed.
- Basic knowledge gathering -> Familiar with specimen and project subject area
- Pro forma created
- Specimen and measuring tools available

**Goal:** Score all relevant descriptive data for one specimen

T takes copy of empty pro forma. T reviews pro forma. T opens specimen and looks at it, gaining impression of general structure. T records ID data from Specimen Label, onto pro forma next to the pro forma labels such as *accession*

## Appendix A - Use Cases

*number*. T looks at next section 'General' and the first label 'habit', then examines specimen. Deciding the specimen is a perennial herb, the T makes a note on the pro forma in the space next to the habit label.

T looks at next label 'stem/branches' and examines specimen for these features. Deciding the stem and branches are terete, the T writes that on the pro forma next to the stem/branches label. Examine stem/branches again and deciding they are striated, record that as well next to terete. T continues examining specimen and sees white hairs. Deciding these hairs are sparse, T records sparse white hairs next to striate. Examine specimen again and decide can see nothing else worth recording about stem/branches. T then moves onto next label 'other', but has nothing to add, so moves onto next section 'Leaf'.

Looking at the first leaf label, shape, T examines specimen to decide what leaf shape is. Decide the basic structure is 2-pinnate, and record that on pro forma as '2-pinn'. Looking at structure, T decides that 2 pinnate does not fully describe the structure and makes a sketch of basic structure on blank space on pro forma, near to shape label. T then looks at the individual leaf shapes and seeing they are all basically the same, makes a sketch, next to the previous sketch in the leaf section. Looking up leaf shapes in a book whose leaf shape definitions the T likes, T decides the leaf shape is like deltoid and oblong, so records deltoid-oblong next to the shape sketch. Deciding that shape has been sufficiently described, T checks the pro forma for the next label and sees it is indumentum. T then examines indumentum and decides it is glabrous and writes 'glabrous' next to the indumentum label. T then reads next label 'petiole' and examines the petioles on the specimen, deciding they are strigose and hairy, so T records strigose hairy. T also notices that the hairs are particularly hairy on the ribs and margins, and adds a note 'part. on ribs and margins' next to strigose hairy. T reads next label 'stipules' but knowing from the on-going examination of the specimen that there are no stipules on this specimen, T skips this label and goes on to next label. T repeats the process for other labels in leaf section.

T then fills in inflorescence section of pro forma in similar manner. Looking at next section 'fruit', T skips this section, as the specimen has no fruit.

Seeing that the end of the pro forma has been reached, T then scans the specimen, checking no outstanding or unusual features have been missed. (*Exception: if find non-recorded feature, record it on pro forma and decide if merits recording for other specimens. If so return to previously recorded specimens and record new feature in ad hoc blank space.*) T then puts pro forma down, closes specimen and returns it to its pile. T then moves onto next specimen.

### Level 2

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Current Usage

**System Scope:** Capture additional description details on all specimens in a taxonomic project

**Pre-requisites:**

- Crude sort completed.

## Appendix A - Use Cases

- Basic knowledge gathering -> Familiar with specimen and project subject area
- Pro forma created
- Initial pro forma scored
- Scored pro formas for specimens reviewed
- Have idea(s) for additional features of interest to be scored

**Goal:** Add and score additional feature(s) to existing pro forma

T decides to add other features (*or more detailed feature details for one of recorded existing pro forma feature labels*) to pro forma. T looks up similar features in other published works. T goes through each specimen, recording the new feature scores in blank unused areas of pro forma (preferably close to other related pro forma features).

## Use Case 2: Abstract Computer System

### Upper Level:

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Scoring taxonomic descriptions for specimens in a project

**Pre-requisites:**

- Crude sort completed.
- Basic knowledge gathering -> Familiar with specimen and project subject area
- Interface system started.

**Goal:** Capture all relevant descriptive data for specimens in project

T opens a new project and the system creates a new project folder with a new pro forma file containing a default specimen label ID section. T gives new project a title. System applies the title to pro forma. T reviews specimen ID labels and edits out unwanted labels and adds others.

T builds a pro forma. System saves the pro forma.

T opens specimen-scoring interface and selects new specimen. System assigns a default specimen ID. T takes random first specimen and uses the pro forma interface to record details of the specimen, selecting and/or entering scores for the characters on the pro forma. T indicates the specimen is scored and saves the work. (*Exception: T sees a character(s) on the specimen that requires to be scored, which is not in the pro forma. T adds character(s) to pro forma. T records character(s) for all previously scored specimens*). The system saves the specimen file and exports the description into the Prometheus II DB, in Prometheus II data format. T repeats process for each specimen.

T reviews pro formas and is happy that sufficient details are recorded to usefully support the taxonomic process of delimiting groups, creating a taxonomic hierarchy and writing taxon descriptions. (*Exception: T decides to add other characters to pro forma: T looks up similar characters in other published works and in Prometheus II DB. T reopens pro forma and adds new character(s) to it. System flags all specimen descriptions in the project that have not had this character scored as incomplete. T goes through each specimen recording the new characters. T reviews pro formas and if necessary repeats adding characters until is happy with results.* )

### Level 2

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Create pro forma to score taxonomic description specimen details upon for a taxonomic project.

**Pre-requisites:**

- As upper level
- New project folder created
- New pro forma file opened

## Appendix A - Use Cases

**Goal:** Create pro forma for current project, for use in capturing specimen descriptive data

T navigates to and reviews default Specimen ID section. T edits Specimen ID section to contain the desired labels.

T adds description sections with characters, until T believes has entered all the characters initially thought of.

T calls up a view of the whole pro forma. System provides a default view. T is content with default view (*Exception: T wants another view, and asks system for a different view. The system provides a choice of views, from which T selects one, which they system then provides*) and explores it, ensuring it is what was meant and is inclusive of all desired characters. T saves this version of project pro forma (*Exception: T is not content with pro forma: T selects and edits characters and/or sections then reviews whole pro forma again*).

### Level 3

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Create description section in pro forma.

**Pre-requisites:**

- As upper level
- New project folder created
- New pro forma file opened
- Specimen Id section completed
- New description section selected

**Goal:** Create description section in pro forma for current project, for use in capturing specimen descriptive data

T adds description section data. System translates section data into structure data for a Description Unit (DU).

T adds a character to the description section. System translates the character to one (or more) partial Description Elements (DE). T repeats adding characters in the section. System translates each character into one (or more) partial DEs as they are created.

T calls up a view of the created section and the system displays the default view (*Exception: T wants another view, and asks system for a different view. The system provides a choice of views, from which T selects one, which they system then provides*). T is content with section (*Exception: T is unhappy with portions of section, and selects characters for editing. T edits or adds characters. T reviews section again*).

### Level 2/3 (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

## Appendix A - Use Cases

**System Scope:** Create pro forma to score taxonomic description specimen details upon for a taxonomic project.

**Pre-requisites:**

- As upper level
- New project folder created
- New pro forma file opened

**Goal:** Create pro forma for current project, for use in capturing specimen descriptive data

T navigates to and reviews default Specimen ID section. T edits Specimen ID section to contain the desired labels. T adds first description section 'General Plant'. System translates section data into structure data for 'Plant' Description Unit (DU). T adds a character to the first description section. System translates character to one (or more) partial Description Element (DE). T repeats adding characters in 'General Plant' section. System translates each character into one (or more) partial DE as they are created. T calls up view of created General Plant section and system displays default view. T is content with section, and adds new description section 'leaf'. (*Exception: T is unhappy with portions of section, and selects characters for editing.*) The system translates the section data into a Leaf DU and T adds characters one by one to leaf section as above, then repeats adding a section for inflorescence and fruit. T calls up a view of the whole pro forma. System provides a default view. T is content with default view (*Exception: T wants another view, and asks system for a different view. The system provides a choice of views, from which T selects one, which they system then provides*) and explores it, ensuring it is what was meant and is inclusive of all desired characters. T saves this version of project pro forma.

### Level 4

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Add character into pro forma.

**Pre-requisites:**

- As upper level
- Description Section (DU) created and selected
- Have concept idea for character

**Goal:** Add Quantitative character to pro forma

T requests a new character. System begins new character interface. T specifies the extra detail of what part of the plant is being described in this character, exploring filtered views of the database, choosing a structure term (*Exception: Term does not exist: T creates new term and definition*) and selecting a definition (*Exception: T does not wish to use existing definition: T adds new definition*). T selects a quantitative abstract property from a list of possible properties (*Exception: T selects non-quantitative property: Fail to add quantitative character*). The system flags the character as quantitative, based on the property selection. T indicates to the system that the property is not relative (*Exception: T indicates property is relative: Failure to add basic quantitative character*). T indicates that preferred measurement units are cm. System creates default label for the character. T accepts the default label for the character (*Exception: T does not accept default label: T edits label. System saves new label*).

## Appendix A - Use Cases

System translates the character into partial DE. T reviews partial DE. T indicates the character is completed. (*Exception: T returns and edits character further.*)

### Level 4 (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Add character into pro forma.

**Pre-requisites:**

- As upper level
- 'Leaf' Section (DU) created and selected
- Have concept idea for character

**Goal:** Add Quantitative character 'leaf petiole length' to pro forma

T requests a new character. System begins new character interface. T specifies the extra detail of what part of the plant is being described in this character, exploring filtered views of the database, choosing a structure term 'petiole' (*Exception: Term does not exist: T creates new term and definition*) and selecting a definition (*Exception: T does not wish to use existing definition: T adds new definition*). T then selects the abstract property 'length' from a list of possible properties. The system flags the character as quantitative, based on the property selection. T indicates to the system that the property is not relative. T indicates that preferred measurement units are cm. T accepts the default label 'leaf petiole length' for the character. System translates the character into partial DE. T reviews partial DE. T indicates the character is completed. (*Exception: T returns and edits character further.*)

### Level 5 (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Explore definition DB

**Pre-requisites:**

- As upper level
- 'Leaf' Section (DU) created and selected
- Have concept idea for character
- DU created
- Pro forma does not include required structure
- Character editing interface running

**Goal:** Find definition for petiole

T navigates to DB exploration interface. T selects Db filter 'structure terms'. System displays all available structure terms. T searches available structure terms and selects 'petiole'. System displays all available definitions for petiole. T reviews available definitions and selects one. System adds selected defined term to character being built. T navigates back to character editing interface.

## Level 4

## Appendix A - Use Cases

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Add character into pro forma.

**Pre-requisites:**

- As upper level
- Description Section (DU) created and selected
- Have concept idea for character

**Goal:** Add Qualitative character to pro forma

T requests a new character. System begins new character interface. T decides what part of the plant is being described in this character. T explores the pro forma to determine if the required structure(s) is already present and selects those elements. (*Exception: Structure is not present in part or in full: T explores filtered views of the database, selecting sufficient structure terms to fully describe where the character exists on the specimen* (**Exception: Term does not exist: T creates new term and definition**) and selecting definitions for every structure term (**Exception: T does not wish to use existing definition: T adds new definition**)). System begins construction of new partial DE by adding composite structure. System displays list of properties and T selects a qualitative property (*Exception: T selects non-qualitative property: Failure to add qualitative character. Add quantitative character*). System interprets the property choice and flags the partial DE as qualitative. T explores DB for relevant possible defined states, selecting and adding to the possible states until all possible states desired for this character have been added. (*Exception: T cannot find all possible desired states in DB: T adds new defined state. System adds new defined states to DB (provisional addition until confirmed by at project completion) and assigns the new state to the possible states for the character*). T indicates to the system that the property is not relative (*Exception: T indicates property is relative: Failure to add basic quantitative character. Add relative character*). System translates character into series of possible DEs list of states selected. T reviews possible DEs and indicates character is complete (*Exception: T returns and edits character further.*)

### Level 4 (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Add character into pro forma.

**Pre-requisites:**

- As upper level
- 'Leaf' Section (DU) created and selected
- Have concept idea for character

**Goal:** Add Qualitative character 'leaf petiole length' to pro forma

T requests a new character. System begins new character interface. T decides what part of the plant is being described in this character. T explores the pro forma to determine if the required structure(s) is already present and selects those elements ('Leaf + petiole'). (*Exception: Structure is not present in part or in full: T explores filtered views of the database, choosing a structure term 'petiole'* (**Exception: Term does not exist: T creates new term and definition**) and selecting a definition (**Exception: T does not wish to use existing definition: T adds new definition**)). System begins construction of new partial



## Appendix A - Use Cases

DE by adding composite structure. System displays list of properties and T selects 'shape'. System interprets the property choice and flags the partial DE as qualitative. T explores DB for relevant possible defined states, selecting and adding to the possible states until all possible states desired for this character have been added. *(Exception: T cannot find all possible desired states in DB: T adds new defined state. System adds new defined states to DB (provisional addition until confirmed by at project completion) and assigns the new state to the possible states for the character).* T indicates to the system that the property is not relative. System translates character into series of possible DEs list of states selected. T reviews possible DEs and indicates character is complete *(Exception: T returns and edits character further.)*

### Level 5 (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Explore definition DB

**Pre-requisites:**

- As upper level
- 'Leaf' Section (DU) created and selected
- Have concept idea for character
- DU created
- Character editing interface running
- Constructed

**Goal:** Find defined states for leaf petiole shape character

T navigates to DB exploration interface.

T selects Db filter 'states' and 'property: shape'. System displays all available states.

T selects a known state to begin exploration *(Exception: T does not have a starting state idea: T uses different criteria such as author to begin exploration).* System displays alternate definitions for selected term, along with limited view of defined terms related by other criteria such as author.

T explores defined terms and selects one.

*(Exception: T does not find an acceptable defined term: T adds new defined term to database.)*

*(Exception: T finds character has been constructed incorrectly: T returns to character editing interface and edits character.)*

*(Exception: T finds defined term, which is not useful for current character but will be for later character: T selects defined term and indicates that it is potentially useful. System flags defined term for later use by T.)*

System adds selected defined term to list of possible states for character under construction. T selects defined states until all defined states are selected for the character *(Exception: T cannot find all possible defined terms for states and is unwilling to currently add new defined state: T indicates the character is*

## Appendix A - Use Cases

*incomplete: System flags character as incomplete. T begins new character). T navigates back to character editing interface.*

### Level 2

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Scoring taxonomic description for a specimen based on a pro forma.

**Pre-requisites:**

- Crude sort completed.
- Basic knowledge gathering -> Familiar with specimen and project subject area
- Pro forma created
- Specimen and measuring tools available
- Pro forma scoring interface started

**Goal:** Capture all relevant descriptive data

T opens a view of the pro forma. The system displays the requested view and T reviews the pro forma. T opens specimen and looks at it, gaining impression of general structure. T navigates to the Specimen ID section of the pro forma, and enters the *accession number* the specimen label into the relevant field. T then enters the other ID section fields one by one, and enters the ID data.

T scores and inputs the characters in the description sections for the specimen. System translates completed characters to completed DEs. (*Exception: specimen is missing structure. T indicates structure is missing and pro forma creates structure not present DE, and marks dependent characters as non-collectable.*)

Seeing that the end of the pro forma has been reached, T requests a view of the completed description, which the system provides. T then explores the description checking it has been properly recorded (*Exception: Find improperly recorded character, and re-enter scoring process*) and checks all characters completed. T then scans the specimen, checking no outstanding or unusual features have been missed. (*Exception: if find non-recorded character, decide if merits recording and if so enter the edit pro forma interface and add characters as required. Then return to previously recorded specimens and record new character scores.*) T then saves description, closes specimen and returns it to its pile. T then moves onto next specimen.

### Level 3

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Scoring taxonomic character of a specimen based on a pro forma.

**Pre-requisites:**

- As previous

**Goal:** Rigorously and accurately score selected quantitative character.

T reads the pro forma character from the displayed interface. T requests any extra required detail on any of the terms used in the character and the system displays the

## Appendix A - Use Cases

requested stored definition(s). T then looks at the specimen and takes measurements (*Exception: feature does not exist on specimen: T records feature not present. System creates appropriate DE and marks all other dependent characters non-completable for this specimen*). T enters the value into the interface, and checks to see that the correct unit of measurement has been used. (*Exception: If the used measurement is not the default measurement, T either selects the utilised measurement or alters the data to be accurate*). T indicates that no other modifier is needed and navigates to the next character.

### Level 3 (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Scoring taxonomic character of a specimen based on a pro forma.

**Pre-requisites:**

- As previous

**Goal:** Rigorously and accurately score selected quantitative character.

T reads the pro forma character 'leaf petiole length' from the displayed interface. T requests more detail on the term 'petiole' and system displays the stored definition. T then requests more detail on the use of length and the system displays the definition of length in this instance. T then looks at the specimen and takes measurements. Discovering the petiole is 1.5 cm long using the pro forma definition of how to measure petiole length, T enters the value into the interface, and checks to see that cm is the default measurement unit (*Exception: If cm is not the default measurement, T either selects cm or alters the data to be accurate*). T indicates that no other modifier is needed and navigates to the next character.

### Level 3

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Scoring taxonomic character of a specimen based on a pro forma.

**Pre-requisites:**

- As previous

**Goal:** Rigorously and accurately score selected qualitative character.

T reads qualitative pro forma character from the displayed interface. T requests any extra-required detail on any of the terms used in the character and the system displays the requested stored definition(s).

T then examines the relevant portions of the specimen. T determines, referencing back to the states displayed on the interface, which of the possible states from the pro forma, that the specimen has for the character. (*Exception: T does not find the possible states cover the specimen and enters a new state into the pro forma, the system then flags all completed pro formas before this one, as requiring checked for this character*). T

## Appendix A - Use Cases

selects the state that applies. *(Exception: T wishes to select more than one state: T selects more than one state. The system finds it has more than one state for one partial character and requires further data. T indicates whether the states are present on a single instance of the structure. The system translates the characters and scores into the appropriate numbers of DEs and requests confirmation. T gives confirmation (Exception: T does not confirm and re-scores the character).*

T selects any required modifiers to apply to the character. System adds any selected modifiers to DE. T indicates that no other modifiers are needed and navigates to the next character.

### Level 3 (instance)

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (abstract)

**System Scope:** Scoring taxonomic character of a specimen based on a pro forma.

**Pre-requisites:**

- As previous

**Goal:** Rigorously and accurately score selected qualitative character.

T reads the pro forma character 'leaf shape' from the displayed interface. T requests more detail on the term 'leaf' and system displays the stored definition. T then requests more detail on the possible states and the system displays the various possible states for this character. T then looks at the specimen, carefully examining the leaves. T determines, referencing back to the states displayed on the interface, that the leaves are mostly closest to the deltoid definition, but sometimes closer to oblong *(Exception: T does not find the possible states cover the specimen and enters a new state into the pro forma, the system then flags all completed pro formas before this one, as requiring checked for this character).* T decides that the shape of the leaves does not seem to correlate with other distinctions between the leaves, but is not worth changing the pro forma to include two different types of leaf at this point. T selects the state deltoid and the state oblong from the interface. T then indicates another modifier is required and requests a list of possible frequency modifiers. T selects 'mostly' for deltoid and 'sometimes' for oblong. The system finds it has two states for one partial character and requires further data. T indicates that the two states are not present on a single instance of leaf, and that the leaf section should not be repeated. The system translates the characters and scores into two DEs and asks for confirmation of what T has entered. T gives confirmation *(Exception: T does not confirm and re-scores the character).* T indicates that no other modifiers are needed and navigates to the next character.

### Use Case 3: Guidance Heavy Interface

#### Upper Level:

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (guidance heavy interface)

**System Scope:** Capture taxonomic descriptions for all specimens in a project

**Pre-requisites:**

- Crude sort completed.
- Basic knowledge gathering -> Familiar with specimen and project subject area
- Interface system started.

**Goal:** Capture all relevant descriptive data for specimens in project

T opens a new project. System starts new project wizard. T selects standard or custom set-up. T selects from set-up options, what specimen ID labels to apply to the pro forma and selects next page. System displays next page of wizard. T inputs title and selects next. System displays next page of set-up wizard. T selects any desired default description sections to include and selects finish. System generates pro forma, with specimen ID section, title and description sections as selected by T. System displays view of partial pro forma.

T builds a pro forma. System saves the pro forma.

T opens specimen-scoring interface and selects new specimen. System assigns a default specimen ID. T takes random first specimen and uses the pro forma interface to record details of the specimen, selecting and/or entering scores for the characters on the pro forma, as prompted by the system. T reaches end of pro forma scoring wizard and selects finish. (*Exception: T sees a character(s) on the specimen that requires to be scored, which is not in the pro forma. T selects edit pro forma. T adds character(s) to pro forma. T records character(s) for all previously scored specimens*). The system saves the specimen file and exports the description into the Prometheus II DB, in Prometheus II data format. T repeats process for each specimen.

T reviews pro formas and is happy that sufficient details are recorded to usefully support the taxonomic process of delimiting groups, creating a taxonomic hierarchy and writing taxon descriptions. (*Exception: T decides to add other characters to pro forma: T looks up similar characters in other published works and in Prometheus II DB. T reopens pro forma wizard and adds new character(s). System flags all specimen descriptions in the project that have not had this character scored as incomplete. System reminds T these flagged items require scored. T selects score flagged items. T goes through each specimen recording the new characters. T reviews pro formas and if necessary repeats adding characters until is happy with results.* )

#### Level 2

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (guidance heavy interface)

## Appendix A - Use Cases

**System Scope:** Create pro forma to score taxonomic description specimen details upon for a taxonomic project.

**Pre-requisites:**

- As upper level
- New project folder created
- New pro forma file opened
- Specimen ID section edited

**Goal:** Create pro forma for current project, for use in capturing specimen descriptive data

T adds description sections with characters, until T believes has entered all the characters initially thought of.

T indicates all characters added. System provides a default view of whole pro forma. T is content with default view (*Exception: T wants another view, and asks system for a different view. The system provides a choice of views, from which T selects one, which they system then provides*) and explores it, ensuring it is what was meant and is inclusive of all desired characters. T indicates is content with pro forma and system saves this version of project pro forma (*Exception: T is not content with pro forma: T indicates is not content. T selects and edits characters and/or sections then reviews whole pro forma again*).

### Level 3

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (guidance heavy interface)

**System Scope:** Create description section in pro forma.

**Pre-requisites:**

- As upper level
- New project folder created
- New pro forma file opened
- Specimen Id section completed

**Goal:** Create description section in pro forma for current project, for use in capturing specimen descriptive data

System begins add description section window of create pro forma wizard. System displays list of default common description sections, which display pop-up window definitions on mouse over. T selects a description section. System translates section data into structure data for a Description Unit (DU). (*Exception: T browses to find defined structure term from database. System uses this defined structure as basis for DU.*) (*Exception: T adds own defined term for description section. System uses this new defined term as basis for DU.*)

System begins add character window. T adds a character to the description section. System translates the character to one (or more) partial Description Elements (DE). System repeats adding characters in the section until T indicates all added. System translates each character into one (or more) partial DEs as they are created.

System displays default view of the created section (*Exception: T wants another view, and asks system for a different view. The system provides a choice of views, from which*

## Appendix A - Use Cases

*T selects one, which they system then provides). T indicates is content with section. System asks whether another section is needed. (Exception: T is unhappy with portions of section, and selects characters for editing. T edits or adds characters. T reviews section again).*

### Level 2

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (guidance heavy interface)

**System Scope:** Scoring taxonomic description for a specimen based on a pro forma.

**Pre-requisites:**

- Crude sort completed.
- Basic knowledge gathering -> Familiar with specimen and project subject area
- Pro forma created
- Specimen and measuring tools available
- Pro forma scoring interface started

**Goal:** Capture all relevant descriptive data

System displays an overview view of the pro forma and first specimen ID 'character' to be scored. T navigates to overview of pro forma and reviews the pro forma. T opens specimen and looks at it, gaining impression of general structure. T navigates back to main scoring interface and enters the *accession number* from the specimen label into the relevant field. System saves the entered data and displays next ID section 'character'. T then enters the ID data for the other ID section fields one by one as requested.

System displays first actual character to be scored. System displays extra information on terms as requested by T. T scores the character as requested, navigating to the pro forma overview to gain perspective as required. System translates completed character to completed DE(s) and updates the overview. *(Exception: specimen is missing structure. T indicates structure is missing and pro forma creates structure not present DE, and marks dependent characters as non-collectable.) (Exception: T is unable to score character for other reasons. System flags character as not scored and moves to next character).* System displays next character to be scored and T scores character as above until all characters entered.

System indicates pro forma completed, displays any problems and/or unscored characters and requests confirmation of completion. T navigates to the completed pro forma overview, checking it has been properly recorded *(Exception: Find improperly recorded character, and re-enter scoring process).* T then scans the specimen, checking no outstanding or unusual features have been missed. *(Exception: if find non-recorded character, decide if merits recording and if so enter the edit pro forma interface and add characters as required. Then return to previously recorded specimens and record new character scores.)* T then confirms completion. System saves the description, and T closes the specimen and returns it to its pile. T then moves onto next specimen.

## Use Case 4: Free Form Multi-Pane Constructor Interface

### Upper Level:

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (multi-pane constructor interface)

**System Scope:** Capture taxonomic descriptions for all specimens in a project

**Pre-requisites:**

- Crude sort completed.
- Basic knowledge gathering -> Familiar with specimen and project subject area
- Interface system started.

**Goal:** Capture all relevant descriptive data for specimens in project

T opens a new project. The system creates a new project folder with a new pro forma file containing a default specimen label ID section. System updates view of pro forma. T gives new project a title. System applies the title to pro forma and updates display. T selects specimen ID section from overview pane and indicates desire to edit it. System opens section in editing pane. T reviews specimen ID labels and edits out unwanted labels and adds others.

T builds a pro forma. System saves the pro forma. T shuts editing windows.

T opens specimen-scoring window and selects new specimen. System assigns a default specimen ID. T takes random first specimen and uses the pro forma interface to record details of the specimen, selecting and/or entering scores for the characters on the pro forma. T indicates the specimen is scored and saves the work. (*Exception: T sees a character(s) on the specimen that requires to be scored, which is not in the pro forma. T opens editing windows. T adds character(s) to pro forma. T closes editing windows. System flags all previously scored specimens as incomplete. T records character(s) for all previously scored specimens*). The system saves the specimen file and exports the description into the Prometheus II DB, in Prometheus II data format. T repeats process for each specimen.

T reviews pro formas and is happy that sufficient details are recorded to usefully support the taxonomic process of delimiting groups, creating a taxonomic hierarchy and writing taxon descriptions. (*Exception: T decides to add other characters to pro forma: T looks up similar characters in other published works and in Prometheus II DB. T reopens pro forma and adds new character(s) to it. System flags all specimen descriptions in the project that have not had this character scored as incomplete. T goes through each specimen recording the new characters. T reviews pro formas and if necessary repeats adding characters until is happy with results.* )

### Level 2

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (multi-pane constructor interface)

**System Scope:** Create pro forma to score taxonomic description specimen details upon for a taxonomic project.



## Appendix A - Use Cases

### Pre-requisites:

- As upper level
- New project folder created
- New pro forma file opened
- Specimen ID section edited

**Goal:** Create pro forma for current project, for use in capturing specimen descriptive data

T uses the editing pane to add description sections and characters, until T believes has entered all the characters initially thought of.

T explores pro forma in overview window, ensuring it is what was meant and is inclusive of all desired characters. T saves this version of project pro forma (*Exception: T is not content with pro forma: T selects and edits characters and/or sections then reviews whole pro forma again*).

### Level 3

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (multi-pane constructor interface)

**System Scope:** Create description section in pro forma.

### Pre-requisites:

- As upper level
- New project folder created
- New pro forma file opened
- Specimen Id section completed

**Goal:** Create description section in pro forma for current project, for use in capturing specimen descriptive data

T explores the defined structure terms and selects desired defined term for description section. T drags term to pro forma structure pane and drops it as the root of a new description section. (*Exception: structure term already in pro forma as part of another description section: T copies term to a new description section within pro forma structure pane*). System translates section data into structure data for a Description Unit (DU).

### Level 4

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (multi-pane constructor interface)

**System Scope:** Add character into pro forma.

### Pre-requisites:

- As upper level
- Description Section (DU) created and selected
- Have concept idea for character

**Goal:** Add Qualitative character to pro forma

## Appendix A - Use Cases

T explores pro forma structure and decides whether needs to add any additional structures to indicate where character is found. As required, T explores defined structure terms in exploration pane, copying selected defined terms to pro forma structure pane as required, until all structure data for character is present (*Exception: Desired defined term does not exist: T creates new term and definition*). System saves this structure data provisionally, but only translate to Prometheus II DB format when structure data used in a character. T selects structure elements and system displays selected structure in editing pane.

T selects qualitative property in editing pane from list of properties. (*Exception: T selects non-qualitative property: Failure to add qualitative character. Add quantitative character.*) System flags character as qualitative and configures editing pane to support alternative states.

T explores defined terms in exploration pane for relevant possible defined states, copying defined terms to the editing pane as possible character states, until all possible states desired for this character have been added. (*Exception: T cannot find all possible desired states in DB: T adds new defined state. System adds new defined states to DB (provisional addition until confirmed by at project completion) and assigns the new state to the possible states for the character*).

T indicates to the system that the property is not relative (*Exception: T indicates property is relative: Failure to add basic quantitative character. Add relative character*). System translates character into a series of possible DEs. T reviews possible DEs and indicates character is complete (*Exception: T returns and edits character further.*)

### Level 2

**Primary Actor:** Taxonomist

**Secondary Actors:** None

**System Type:** Computer pro forma interface (multi-pane constructor interface)

**System Scope:** Scoring taxonomic description for a specimen based on a pro forma.

**Pre-requisites:**

- Crude sort completed.
- Basic knowledge gathering -> Familiar with specimen and project subject area
- Pro forma created
- Specimen and measuring tools available
- Pro forma scoring interface started

**Goal:** Capture all relevant descriptive data

T reviews the pro forma in the overview pane. T opens specimen and looks at it, gaining impression of general structure. T navigates to the Specimen ID section of the pro forma using the overview or specialist navigation pane, and enters the *accession number* the specimen label into the relevant field. T then enters the other ID section fields one by one, and enters the ID data.

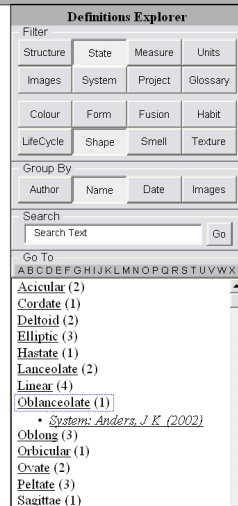
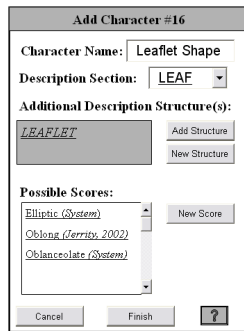
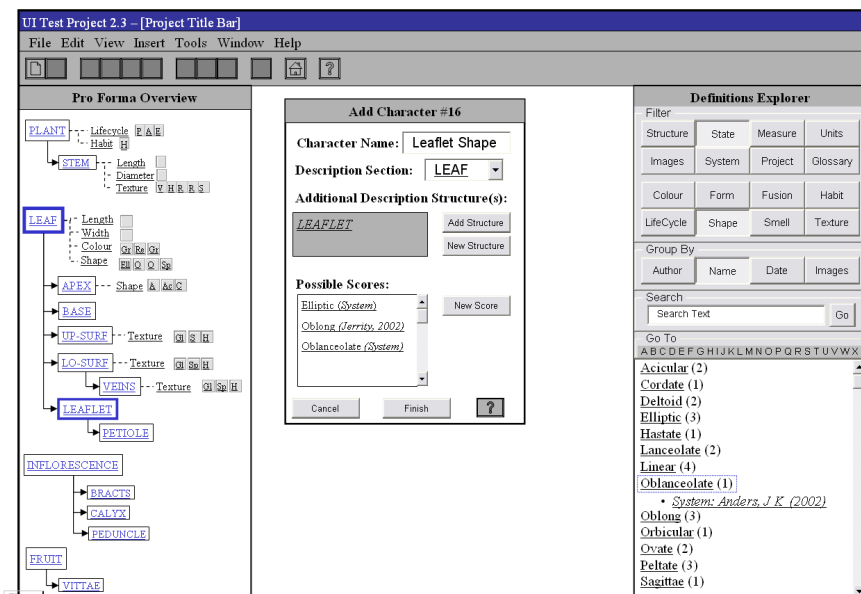
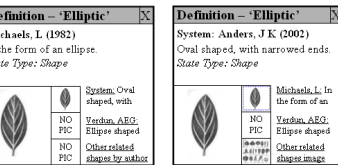
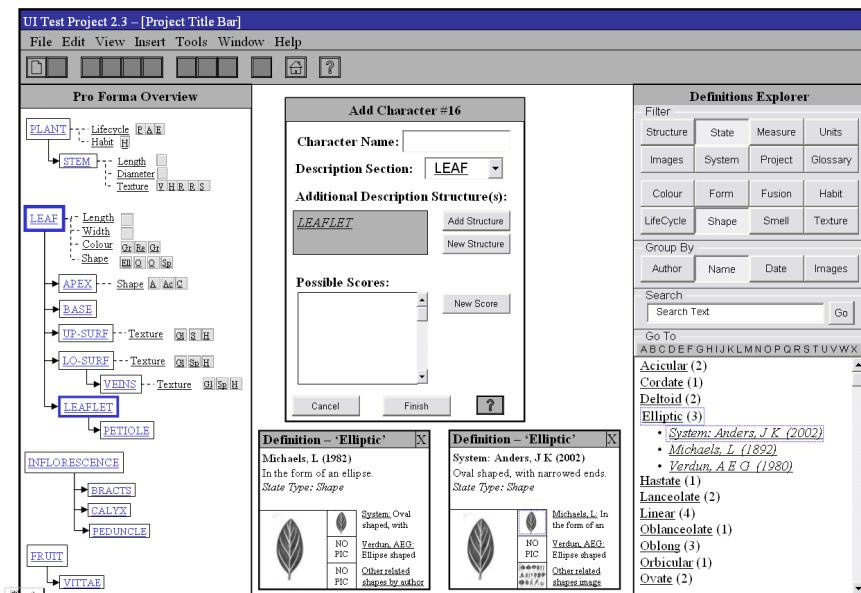
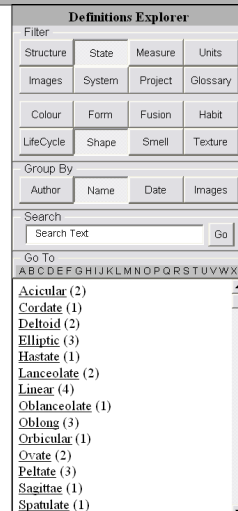
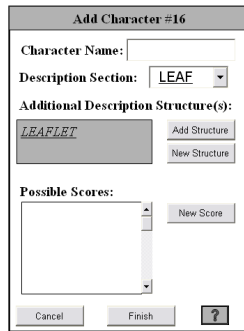
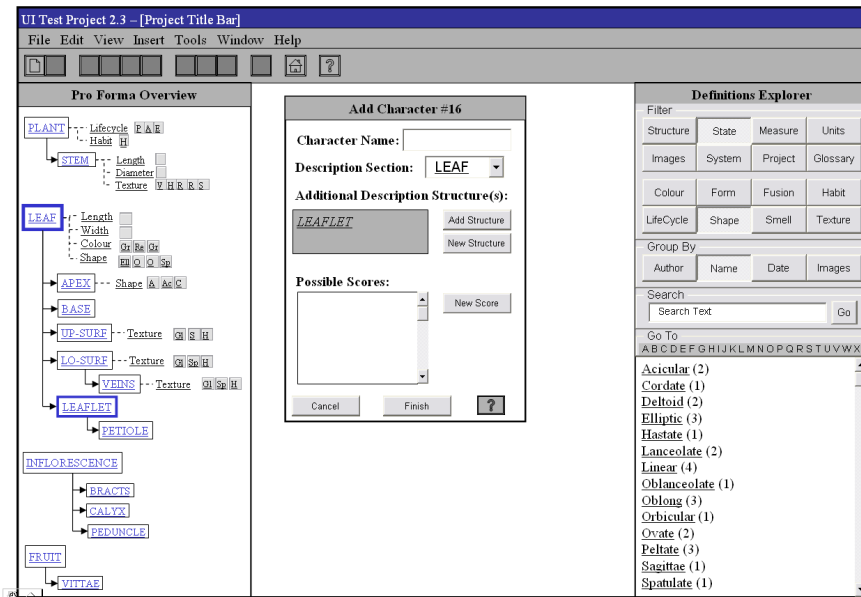
T selects characters from the overview or navigation pane. System displays the selected character (or characters) in the main scoring pane. T scores and inputs the characters. System translates completed characters to completed DEs. (*Exception: specimen is missing structure. T indicates structure is missing and pro forma creates structure not*

## Appendix A - Use Cases

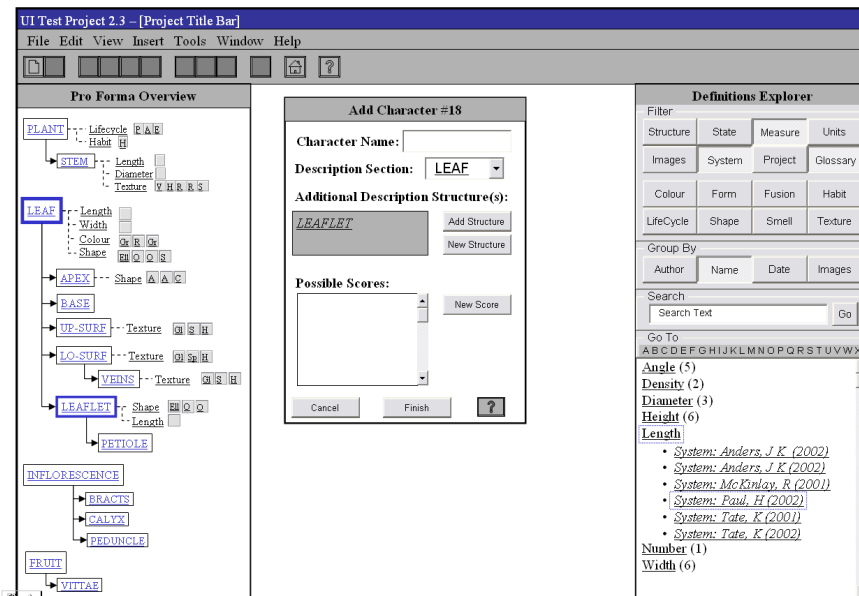
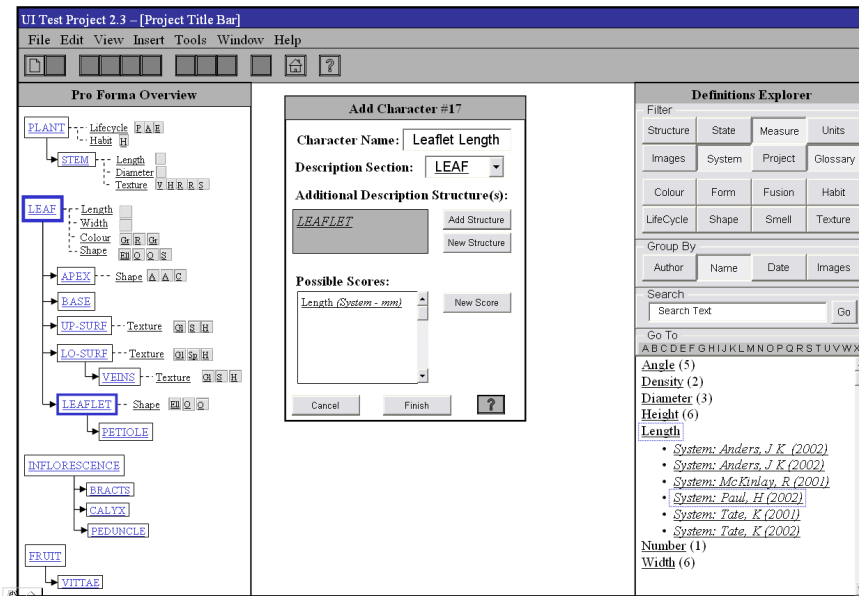
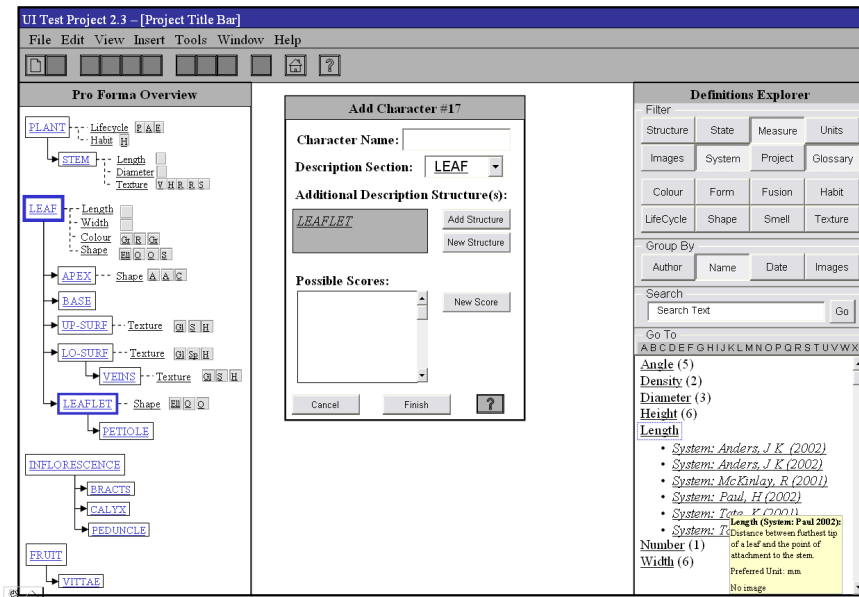
*present DE, and marks dependent characters as non-collectable.)* The system marks the characters as scored in the pro forma overview and navigation panes. T repeats this process until scored all characters that T desires to score.

Seeing that the end of the pro forma has been reached, T explores the description checking it has been properly recorded (*Exception: Find improperly recorded character, and re-enter scoring process*) and checks all characters completed. T then scans the specimen, checking no outstanding or unusual features have been missed. (*Exception: if find non-recorded character, decide if merits recording and if so enter the edit pro forma interface and add characters as required. Then return to previously recorded specimens and record new character scores.*) T then saves description, closes specimen and returns it to its pile. T then moves onto next specimen.

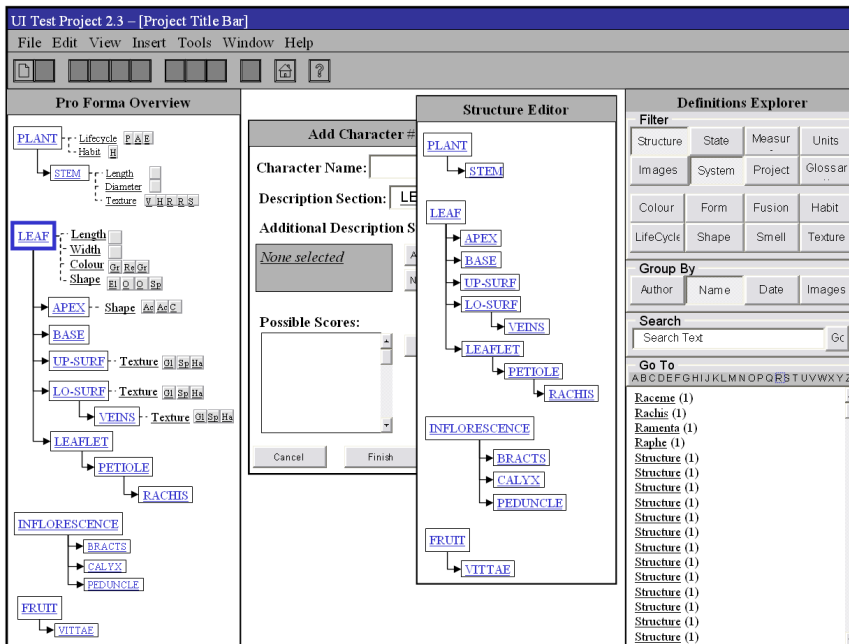
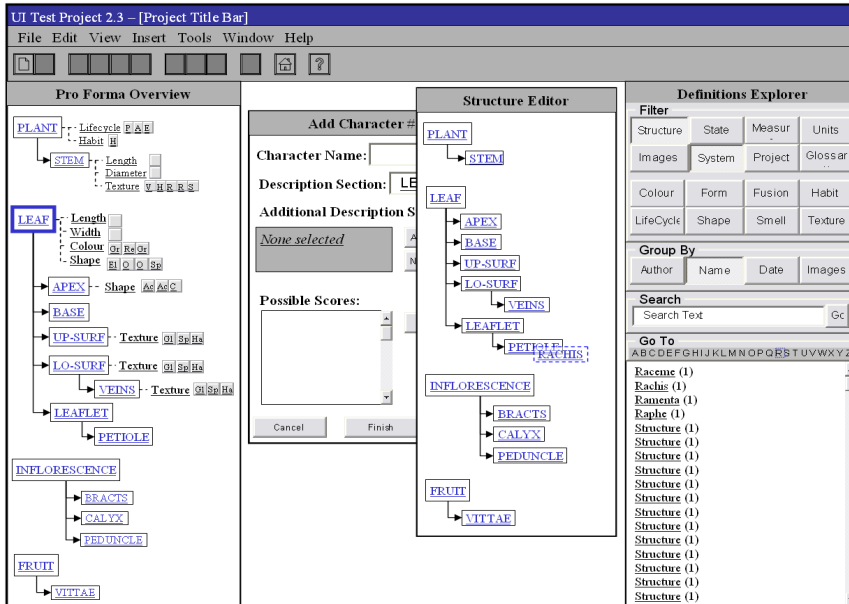
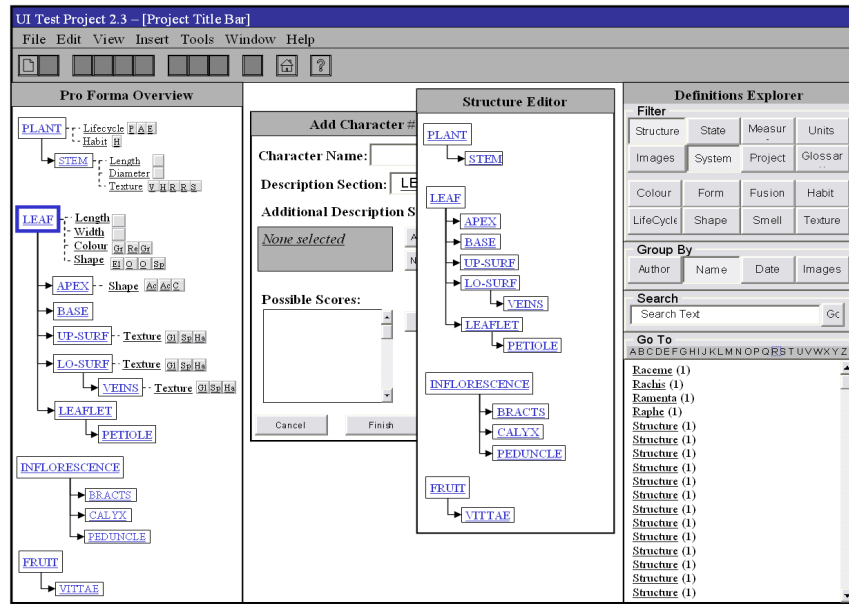
## Appendix B – Storyboard Examples



## Appendix B – Storyboard Examples



## Appendix B – Storyboard Examples



## Appendix B – Storyboard Examples

UI Test Project 2.3 – [Project Title Bar]  
File Edit View Insert Tools Window Help

**Specimen #32 Overview**

- PLANT
  - Lifecycle: E A R
  - Habit: H
  - STEM
    - Length: A
    - Diameter: R
    - Texture: W H R E S
  - LEAF
    - Length: P
    - Width: W
    - Colour: G R G
    - Shape: G Q S
    - APEX
      - Shape: A A C
    - BASE
      - Shape: A A C
    - UP-SURF
      - Texture: G S H
    - LO-SURF
      - Texture: G S H
    - VEINS
      - Texture: G S H
    - LEAFLET
      - Shape: G Q S
      - Length: L
    - PETIOLE
      - Length: L
- INFLORESCENCE
  - BRACTS
  - CALYX
  - PEDUNCLE
- FRUIT
  - VITTAE

**Score Character #16 : Leaflet Shape**

Description Section: LEAF

LEAF → LEAFLET

Elliptic: Oval shaped, with narrowed ends. System: Anders, J K (2002)

Oblanceolate: A leaf broader at the distal third than at the middle and tapering towards the base. System: Anders, J K (2002)

Ovate: An organ, such as a leaf, that is about four times as long as it is broad, and which tapers at both ends. System: Anders, J K (2002)

Buttons: Edit Character, Feature Not Present, Add Modifier, Add Sketch, New Score, Reset, See Other Results, Cancel, Skip, Finish, ?

UI Test Project 2.3 – [Project Title Bar]  
File Edit View Insert Tools Window Help

**Specimen #32 Overview**

- PLANT
  - Lifecycle: E A R
  - Habit: H
  - STEM
    - Length: A
    - Diameter: R
    - Texture: W H R E S
  - LEAF
    - Length: P
    - Width: W
    - Colour: G R G
    - Shape: G Q S
    - APEX
      - Shape: A A C
    - BASE
      - Shape: A A C
    - UP-SURF
      - Texture: G S H
    - LO-SURF
      - Texture: G S H
    - VEINS
      - Texture: G S H
    - LEAFLET
      - Shape: G Q S
      - Length: L
    - PETIOLE
      - Length: L
- INFLORESCENCE
  - BRACTS
  - CALYX
  - PEDUNCLE
- FRUIT
  - VITTAE

**Score Character #16 : Leaflet Shape**

Description Section: LEAF

LEAF → LEAFLET

Elliptic: Oval shaped, with narrowed ends. System: Anders, J K (2002)

Oblanceolate: A leaf broader at the distal third than at the middle and tapering towards the base. System: Anders, J K (2002)

Ovate: An organ, such as a leaf, that is about four times as long as it is broad, and which tapers at both ends. System: Anders, J K (2002)

Buttons: Edit Character, Feature Not Present, Add Modifier, Add Sketch, New Score, Reset, See Other Results, Cancel, Skip, Finish, ?

UI Test Project 2.3 – [Project Title Bar]  
File Edit View Insert Tools Window Help

**Specimen #32 Overview**

- PLANT
  - Lifecycle: E A R
  - Habit: H
  - STEM
    - Length: A
    - Diameter: R
    - Texture: W H R E S
  - LEAF
    - Length: P
    - Width: W
    - Colour: G R G
    - Shape: G Q S
    - APEX
      - Shape: A A C
    - BASE
      - Shape: A A C
    - UP-SURF
      - Texture: G S H
    - LO-SURF
      - Texture: G S H
    - VEINS
      - Texture: G S H
    - LEAFLET
      - Shape: G Q S
      - Length: L
    - PETIOLE
      - Length: L
- INFLORESCENCE
  - BRACTS
  - CALYX
  - PEDUNCLE
- FRUIT
  - VITTAE

**Score Character #17 : Leaflet Length**

Description Section: LEAF

LEAF → LEAFLET

Enter Length:  mm

Buttons: Edit Character, Feature Not Present, Add Modifier, See Other Results, Cancel, Reset, Skip, Finish, ?

## Appendix C

### Prometheus 2 Data Model

The Prometheus 2 descriptive model was developed by the separate Prometheus 2 project [Prometheus 2005] alongside this research. While this research influenced the development of the project, the Prometheus 2 database model does not form part of this research's contribution. The Prometheus 2 data model was designed to store rigorous descriptive data in a database. This section describes the model and its terminology as it was currently developed during the development of static prototypes of this research. Later versions of the model are available in [Paterson 2003, 2004].

#### Model Building Blocks

##### Terminology

To avoid confusion, the term 'description element' is used instead of the term 'character'. A description element is one descriptive statement regarding one aspect of one defined structure. A description element may contain the following components:

- **Structure:** a whole plant, any portion of a plant or a compound group of portions of the plant (e.g. leaf)
- **Property:** an abstract aspect of the structure that is being described, taken from a fixed list (e.g., shape).
- **State:** a qualitative score that the structure that is being described can take (e.g. obovate).
- **Value:** a quantitative score that can be assigned to a property associated with a structure (e.g. 6).
- **Units:** the units included in a quantitative score (e.g. cm)

Description units are everything recorded about one defined structure within one description.

A description is everything that is recorded for one specimen or taxon by one taxonomist in one publication. A specimen description consists of all the description units of one specimen. Taxa descriptions consist of one or more specimen (or virtual specimen) descriptions.

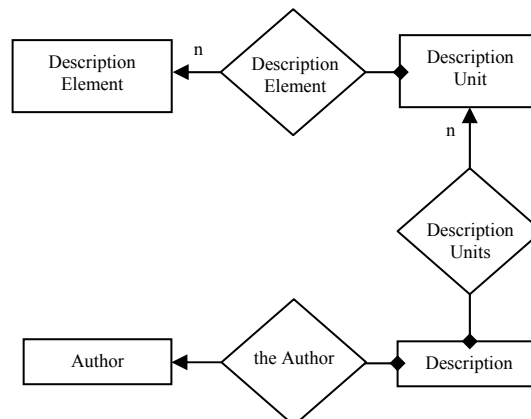


Figure C.1: Prometheus II model – Description components

##### Defining terms

The Prometheus approach requires each use of a term to be associated with a definition. Terms are simply the bare words that are found in a standard description (e.g. leaf). There may be



## Appendix C - Prometheus 2 Data Model

several definitions available for each term. Definitions consist of a textual definition and a literature reference (including the author of the definition). Multimedia definition data (e.g. pictures, diagrams) are included in a definition where possible. Associating the use of a term with a definition creates a defined term. Each defined term also has an author associated with it.

Definitions are assigned to terms to create defined structures, defined properties, defined qualitative states and defined units. The list of structure terms is open-ended and user defined. Users may either draw from the list of previously used structure terms or may create a new term. Structures may have multiple definitions. Properties are only defined when used in qualitative description elements. Additions to the property term list are possible only in exceptional circumstances. Definitions of qualitative states differ from other definitions by including a reference to one or more property terms. Associating a state term with a property term in its definition allows queries such as ‘find all the states that are shapes’ to be handled. Allowing a state definition to reference multiple property terms handles state terms such as ‘radiating’, where it is difficult to conclusively assign the term to only one property. Defined units are used in conjunction with values.

### Building Descriptions

Basic descriptions of plant features are constructed by combining a number of description element components, in one description element.

#### Compound Structures

Description elements can either contain simple structures (e.g. leaf), or compound structures (e.g. leaf margin). The creation of compound structures is performed during the description construction process. Compound structures are thus part of a description and are not new defined structures.

Individual defined structures are arranged in a hierarchy to describe compound structures using a ‘part of’ relationship. For example, as shown below, ‘leaf’ is related to ‘margin’ which is in turn related to ‘teeth’ using a ‘part of’ relationship to form the compound structure ‘leaf margin teeth’. Description units are determined by the structure at the top of the compound structure hierarchy. The example below would be held in the ‘leaf’ description unit not the ‘teeth’ description unit.

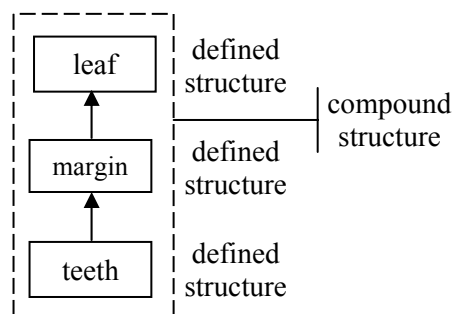


Figure C.2: Example of a Compound Structure

#### Dealing with multiple states for the same property

The semantics of AND and OR between states with the same property (e.g. leaf green and brown vis-à-vis leaf green or brown) are conveyed by the description element. For the AND case, a single description element is created with two or more states that must be assigned to the same property term. For the OR case the two alternatives are recorded as separate description elements, but the states must be assigned to the same property term.

## Appendix C - Prometheus 2 Data Model

### Ranges

The model does not handle qualitative ranges. There is no method in the model for descriptions and definitions to record an author's categorisation, thus another user could not determine what intermediate states might exist in a qualitative range. Quantitative ranges are handled as described under relative modified description elements.

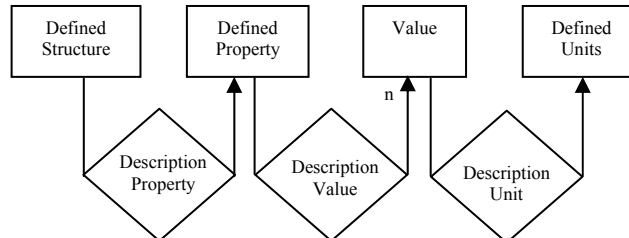
### Types of Description Element

There are two basic types of description elements: quantitative and qualitative. Quantitative description elements describe the results of measurement. Qualitative description elements describe a score by categorisation, (e.g., flowers red). These types of description element require the inclusion of different pieces of data to be explicit. The split of types is not because there is believed to be any real difference between qualitative and quantitative data. Visualisation of the two types of description element thus need only differ according to the demands of the data itself.

### Quantitative Description Element

In a quantitative description element, the property and relevant units must be included as well as the value. A quantitative description element therefore must include a defined structure, a defined property term, a value and the appropriate defined units. The property must be explicit in a quantitative description element. For example, the meaning of the statement 'leaflet 3 cm' is not clear, whereas 'leaflet length 5 cm' is clear.

When creating a quantitative description element, the property term is associated with one or more values. Values are individual numbers or number ranges (e.g. 5, 5 to 10). Values must also be associated with a defined unit. The taxonomist is free to choose whichever unit applies to their score. For quantitative statements that do not have units, for example number of petals, 'count' is defined as a unit. Defined units are assigned when results are recorded (or when



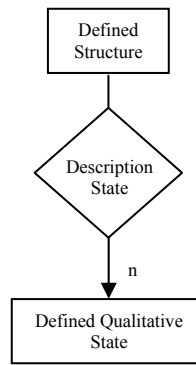
proforma is created).

**Figure C.3: Quantitative Description Element**

### Qualitative Description Element

For a qualitative element to be explicit, it must include a defined structure (simple or compound) and a defined qualitative state. The property being recorded is implicit and is captured by the relationship between the defined state and one or more property terms included in the definition of that defined state. It is not necessary to specifically highlight the property in visualising qualitative statements, as all taxonomists will be able to clearly understand the meaning without its inclusion. If they wish to know, the data can be accessed on demand from the definition of the qualitative state.

## Appendix C - Prometheus 2 Data Model



**FigureC.4: Qualitative Description Element**

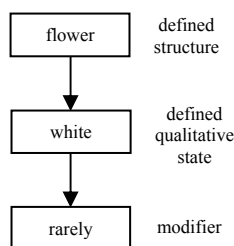
### Modifying Description Elements

Basic quantitative and qualitative elements are the most frequently used elements in a description. It will however be necessary to add further information to description elements to adequately cover more complicated observations. The model, therefore, includes a mechanism by which qualitative and quantitative description elements can be modified. It is also possible to modify a description unit by using a partially formed description element (i.e. a description element where a defined structure is not related to a defined qualitative state or is only related to a property term without a value). There are four kinds of modifiers to description elements: frequency, relative, spatial, and temporal. A description element may have one or more modifiers.

### Frequency Modified Description Element

To capture statements that relate to the frequency of assigned scores, description elements can be qualified using frequency modifiers. A frequency modifier can be attached to each description element. The fixed list of frequency modifiers is: often, usually, sometimes, mostly, rarely, mean.

An example of a frequency modified description element is 'flowers rarely white'. The description element 'flowers, white' is related to the frequency modifier 'rarely' via the description element modifier relationship.



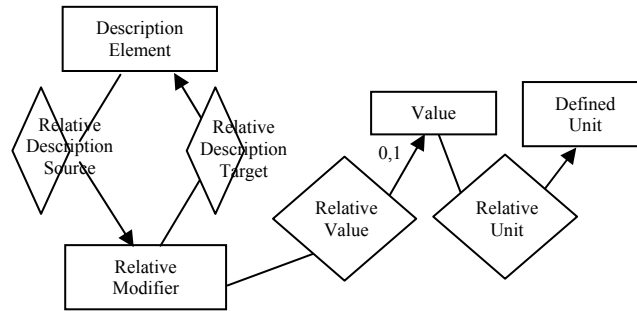
**Figure C.5: Example of Frequency Modified Description Element**

### Relative Modified Description Element

A relative statement is one that compares one aspect of the structure being described to another aspect of the same or a different structure, (e.g. bracteole length greater than pedicel length). The relative description modifier is attached to the first description element and the modifier gives direction to the statement. This allows the identification of the structure that is being referred to and the structure that is making the reference. For instance in the statement 'bracteole length greater than pedicel length' the direction of the reference is from bracteole length to pedicel length.

Generally in relative modified description elements the values are 'undefined'. There is a finite list of relative modifiers (such as greater than, equal, ratio, etc).

## Appendix C - Prometheus 2 Data Model



**Figure C.6: Relative Modified Description Elements**

Ratio is one relative modifier, which is treated slightly different, as the relative modifier contains a value. This value is the ratio and so has no units.

Relative description modifiers are also used to represent quantitative ranges. The extremes of the ranges are separate description elements related by a relative modifier ‘to’. For example, the range ‘stem length 5–10 cm’ would be represented as two quantitative description elements: ‘stem length 5 cm’ and ‘stem length 10 cm’. The relative modifier ‘to’ would then link these two description elements.

Relative modified description elements break the general tree structure of a description by introducing directional cyclical links, as will be discussed further in later sections.

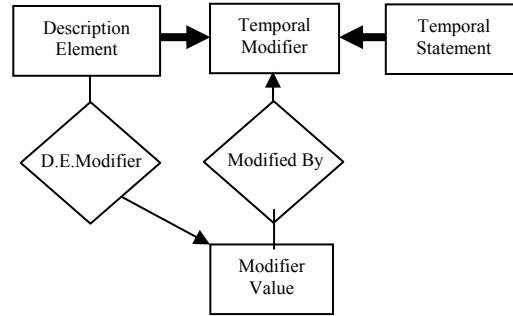
### **Spatial Information – Landmarks**

Landmarks allow the location of a measurement on a structure to be recorded. For example, the diameter of a tree trunk could be measured at various points. The model handles this by associating description elements with defined landmarks via a landmark modifier. The modifier can then target two kinds of objects. One kind is a defined structure, which plant taxonomists so far can only imagine being used for groups other than plants. Alternatively, the modifier could target a defined landmark, which can be a free text statement defined by the user. This would cover statements such as ‘trunk diameter at breast height’. The landmark modifiers can only take the values: at, above, below. There is some uncertainty over whether this modifier is actually required at all and it may be eliminated from the model.

### **Temporal Information**

Some phenomena only appear at certain periods of the year (e.g. flowers in spring) or when other phenomena have already appeared (e.g. fruits after flowers). The model thus allows the recording of the point in time at which a structure has a particular state. Temporal modifiers relate a description element to another description element or to a temporal statement. A temporal statement is a free text object that allows the representation of abstract temporal concepts such as seasons. Temporal modifiers can only have one of the following temporal values: after, before, while. Like relative modified elements, temporal modified elements that relate two description elements also break the general tree hierarchy of descriptions.

## Appendix C - Prometheus 2 Data Model



**Figure C.7: Temporal Modified Description Elements**

## Appendix D

### Third phase development

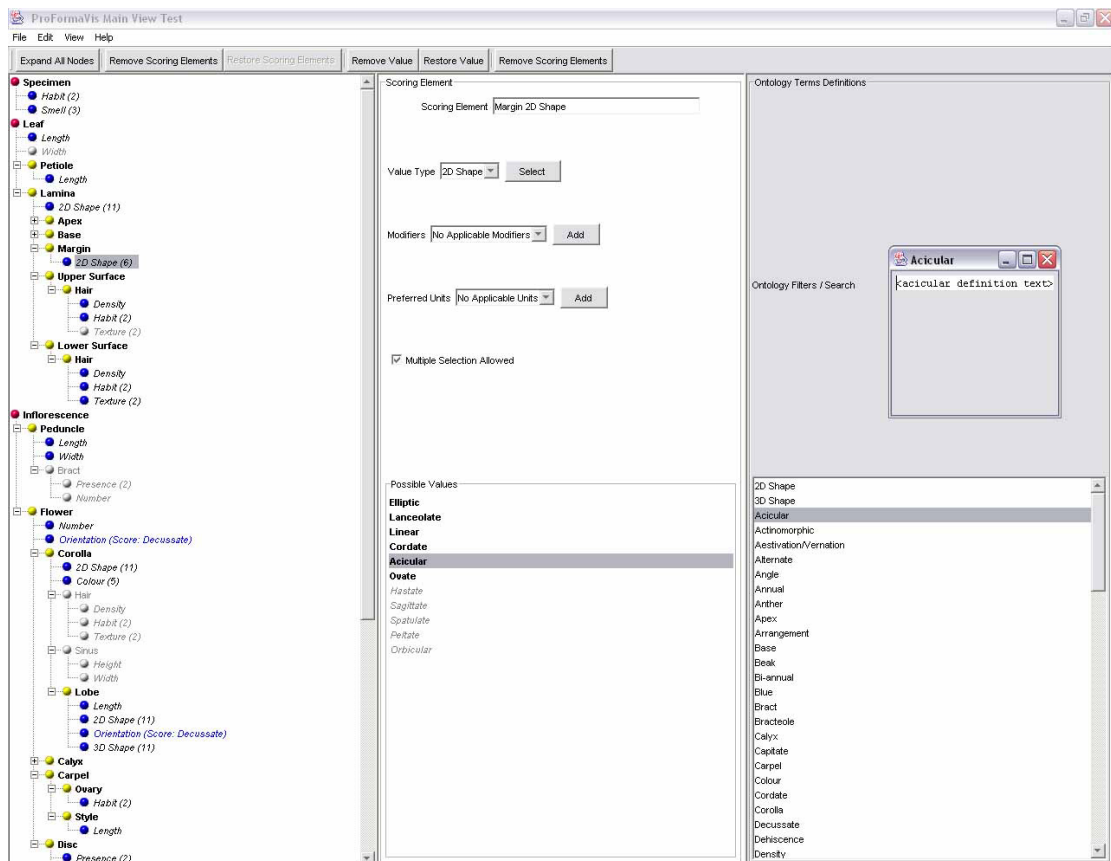
#### D.1 Introducing the approach

In order to solve the problems (articulated in chapter 4) of having taxonomists create the compositional structure hierarchy as well as attribute and value domain relationships, we presented the possible relationships to users and allow them to select those they wished to use.

At this stage, the full angiosperm ontology had not been developed. The description data from a proforma developed for the *codnopsis* group of plants was transformed for use as an ontology to test the functionality of the specialisation interface hierarchical description view.

The presentation of the interface was initially loosely based upon the developed storyboards (see 4.2.3) in that the description hierarchy would primarily be represented by collapsible tree metaphor with nodes for description objects and their attributes as shown in fig. D.1. The inclusion of all the possible members of the value domain as nodes in the tree view would make that view very large and unwieldy. Using an alternate space-saving technique (e.g. fig. 4.3) would make the representation of each value object very small, making it difficult to compare all options at a glance. Accordingly, the value domain was represented in a linked separate column view. For similar reasons the display of all attributes was restricted. Initially there were no fixed restrictions on the applicability of an attribute to description objects. There was however the concept of common properties for a description object and these were mapped to attributes that were included in the specialised domain model and hence the tree. Other attribute relationships could be defined by users as required at which point they would be represented in the tree.

## Appendix D: Third Phase Development



**Fig. D.1: 3<sup>rd</sup> Development Phase specialisation interface based on codnopsis descriptions.**

In fig. D.1 description objects (yellow and red icons) and attributes (blue icons) are represented as nodes in the collapsible tree on the left. Removed elements are greyed-out. The value domain for selected properties is represented on the linked middle column. A view of all defined terms from the ontology is in the right column from which definitions are available. Definitions are also available on tool tips upon mouse-over of the nodes.

To specify a specialised domain model, users edit the hierarchy, removing the elements that are not of interest, and adding elements they are interested in that are not represented (such as other attributes or cloned description objects). For each qualitative property must select which states wish included from the value domain of possible states attached to the property.

## Appendix D: Third Phase Development

### D.2 Evaluation

Initial informal feedback on prototypes was received from 2 RBGE taxonomists and from computer scientists on the Prometheus 2 project staff. A formal user test with 3 RBGE taxonomists was undertaken at the end of the phase (see Appendix E for task details).

Evaluation showed that the basic approach was easily understood by taxonomists using a proforma building metaphor. Removing most of the requirement to define the composition hierarchy was seen as a big improvement over the previous approach outlined in chapter 4. From the effect of their editing turning the coloured nodes into greyed-out nodes, users quickly got the idea of how they were creating the 'proforma'. The initial approach adopted had been to primarily remove description objects from an initially included state. A comparison was made with the beginning with all description objects greyed-out (i.e. not included) and the users having to select and include them. This latter approach was preferred with users saying they saw it as a more positive action and that it was easier to conceptualise finding the elements they were interested in, rather than finding those that they were not.

Working interactively users confirmed that they could navigate the hierarchical display using their domain knowledge of plant structures, although they commented they would prefer to see the hierarchy in acryptic order. There were also some minor concerns about how much of the hierarchy users could see when there were lots of attributes included.

Tooltips for definitions were found to be intuitive and users quickly understood the concept that they could access the definition on mouse-over anywhere in the interface. The view of all defined terms was not utilised during tasks to find definitions of terms and users were uncertain they would need it, since they could just explore the description hierarchy for terms.

The concept of cloning description objects (5.3.4.4) was understood, although it was seen that the name labels would need to be differentiated to avoid confusion. Other minor usability findings fed into later development.



## **Appendix D: Third Phase Development**

### **D.3 Following development phases**

Following the 3<sup>rd</sup> stage evaluation, the interface was substantially refined for the 4<sup>th</sup> development phase when the angiosperm ontology was used. The approach adopted would present the ontology for users to select those elements they were interested in as the primary paradigm. Only where there were no explicit ontological relationships, would users be required to specify their own relationships based on permissible rules.

## User Test: Creating the Pro Forma (3<sup>rd</sup> Stage)

1. **Discuss Stage of Interface Development**
2. **Demonstrate Basic Functionality of Interface**
  - **Explain Interface Elements**
    - Main Tree View
    - Scoring Element Details
      - Name
      - Modifiers
      - Value Type
      - Value Domain
      - Definition Explorer
  - **Explain Basic Functionality**
    - **Main View Editing**
      - Select SE
      - Remove Structure / SE
      - Restore Structure / SE
      - Add new SE/ Region / Generic
      - Clone
    - **Scoring Element Details Editing**
      - Remove Value
      - Restore Value
      - Add Attribute to Blank SE
      - Edit Name
    - **Misc Functions**
      - Hide/Show Removed toggle
      - Expand / Contract
      - Tool tips
      - Definition Box

## Appendix D: Third Phase Development

### 3. Pro Forma Creation Tasks

#### 3.1. Exploration

Find the following using **basic angiosperm** main tree and scoring element view:

- Leaf (*hint: under entire Plant*)
- Fruit
- Arrangement of the anther on the stamen of the androecium of the main inflorescence flower
- Leaf blade length
- Definition of leaf
- Possible shapes of the root
- Possible flower processes
- Definition of flower
- Definition of anthesis (a flower process)
- Possible lifecycle scores of plant

#### 3.2 Editing – Selection

- Modify the possible scores for the 3D shape of the leaf to include only the choice of Acicular and Falcate.
- Add plicate back into the previous list of possible scores for the 3D shape of the leaf.
- Modify the possible scores for the apical shape of the leaf to indicate that all the leaves in the study are apiculate. (*Then use View>Refresh Main View.*)
- Remove outline shape of the leaf from the pro forma.

## Appendix D: Third Phase Development

- Remove leaf cuticle from the pro forma
- Remove inflorescence, infructescence, root, ochrea, husk, seedling, and cupule from the pro forma
- Replace the inflorescence flower in the pro forma

### 3.3 Editing – Creation

- Insert a new scoring element to hold another measurement of leaf length
- Insert a new scoring element to hold another 3D shape with a choice of all possible values
- Insert regions and generic structures to score the length of the spines on the ridges of the stem.
- Insert a second shoot structure which must have an ascending arrangement

### 3.4 Review Pro Forma

- Toggle off View Removed Elements
- View interpretation of Kate's Pro Forma
- View all attributes on basic angiosperm
- View SE names instead of property names

### 3.5. Select the Pro Forma

Start with basic angiosperm with all deselected

- Use restoration to include a possible score for the 3D shape of the leaf to include only the choice of Acicular and Falcate.

#### **Appendix D: Third Phase Development**

- Remove outline shape of the leaf from the pro forma.
- Remove leaf cuticle from the pro forma
- Select entire Plant lifecycle, Leaf (except cuticle), shoot, stem and inflorescence flower into the pro forma.

#### **4. Discussion**

- General method
  - Selection
  - Part-of organisation
  - Value domain
  - Default SEs
  
- Simple Scoring Elements included at beginning?
  
- Effect of all Scoring Elements
  
- Order of potential Scoring
  
- Definition Viewer and Box
  - Popup nature of boxes – arrange as wish
  
- Complex scores
  - Definition of quantitative properties (landmarks, etc)
  
- Use of warning colours
  
- Tree Icons
  
- Tooltips
  
- Other Thoughts

#### **5. Note observed use of:**

- Definitions
- Tool tips
- RC Menus
- Tree icons
- Expand/contract

## Appendix D: Third Phase Development

### User Feedback:

#### Creating the Pro Forma

##### Navigation

- Differentiate labels where clone structure e.g. leaf A and leaf B
- Generally found use of hierarchical tree structure intuitive for searching and finding areas of interest. Users showed a use of their biological knowledge to find elements of interest in the tree.
- The user using his biological knowledge to find things in the tree was put of when he found siblings in an unexpected order. This user found the arbitrary order to be a disconcerting aspect of the visualisation and would expect to see the elements in acryptical order, as they tend to work (Bottom up and flowers outside to inside). The ordering of scoring elements was not a problem.
- Being able to rearrange the order of siblings was seen to be a valuable possibility, although not a complete substitute for seeing the initial tree in acryptical order.
- Acryptical ordering would require institution at the ontology stage as it would be too much work to continually re-order the ontology before beginning work on creating a pro forma, especially as the basic ordering decisions would likely be the same each time. Ordering the pro forma ontology after the initial creation would be more likely task to fine tune ordering for a particular study and for scoring purposes.
- Navigating the tree was slower when large number of options at a given node were presented if they were already all expanded. One user liked to shut nodes he was not interested in.

##### Tooltips/Definitions

- Tooltips for definitions were quite intuitive and users quickly got concept that they worked everywhere.
- Tooltips were however on occasion too long due to having a long definition. Users suggested having a short version of the definition would be helpful.
- Get long definitions from anywhere too on R.C. would be helpful.
- Were uncertain of possible use of definition explorer.
  - Could see a use for exploring terms not automatically included in SE.
  - Useful for search...but could just replace with a simple search.
  - Agreed would not want to see it all the time.

##### Editing

- Extensive use of right click menu for editing.
- General happiness with process. Like greyed out vs. normal coloured nodes. Quickly get idea of how creating pro forma.
- Preference for the selection as opposed to deselection process. Reasons given: More positive action; Easier to conceptualise finding the elements interested in and selecting than finding those not

## Appendix D: Third Phase Development

interested in; Easier when have to add little. However possibility other way around when subtracting little from pro forma.

- Possibly mix both processes under user control.
- Cloning seen as intuitive task. (Possibly some difference in conceptualisation with one user over why cloning structures – he may have believed were cloning for instantiating each actual structure on specimen as opposed to creating a sufficiently different description – however the system does not change for either reason behind cloning structures).
- Cloning scoring elements was the preferred method of creating a new scoring element. Users used cloning (copy) as opposed to insert a new SE, even when that meant more editing. Users were not surprised to see that when they cloned a score, the value domains were edited as the original was.
- One user found it off putting to see the possible simple Ses appear on newly cloned structures. He found it removed control from the user. The other user did not find it off putting and expected to see them.
- Restoring also intuitive operation.
- Working with greyed out elements not a problem.
- Users felt values should be greyed out when SE was greyed out.
- User Observation: Indicator for whether a SE is numeric might be helpful.
- Indicators for size of value domain found helpful, incl. Warning colours. However users did not always pick up on the fact that score was displayed in tree for Ses with only 1 possible.
- Drag for horizontal width of window.
- Prefer perm. Def'n box to popup. Other users not bothered.
- At Scoring need to indicate if closed nodes hide non-scored SE.
- At Scoring – possibly list of large scores and scroll down – more than 1 on screen. But able to jump around list with tree.
- Suggestion of select and drag method for creating pro forma, although agreed could lead to confusion over whether could edit structure hierarchy.



# Appendix E

## Wide User Tests

### 1<sup>st</sup> Wide User Test

#### 1. Discuss Stage of Interface Development

#### 2. Demonstrate Basic Functionality of Interface

- **Explain Interface Elements**
  - Structure Hierarchy
  - Attributes and Values Display
  - Attribute Editing
  - Scoring Panel
- **Explain Basic Functionality**
  - **Main View Editing**
    - Enable Structure
    - Enable Attribute
    - Enable Value
    - Enable All Values of Sub Group
    - Clone
    - Add New Structure
    - Add New Attribute
    - Edit Attribute Name
    - Set Preferred Unit
    - Set Relational Modifier
  - **Misc Functions**
    - Hide/Show Removed toggle
    - Expand / Contract
    - Tool tips
    - Find Structure Match
  - **Move to Score**
    - New Specimen
    - Switch Specimens
    - Score Box
      - i. Basic Select/Entry
      - ii. Modifiers
      - iii. Clear
      - iv. AND/OR
      - v. Not Scored
    - Next Structure

### 3. Pro Forma Creation Tasks

#### 3.1. Exploration

Find the following using the **Angiosperm Structure Hierarchy** and associated **Attributes and Values display**:

- Flower
- Definition of flower
- Possible flower positions
- Definition of marcescent (a flower lifestyle)
- Disc (of flower)
- Any Other 'Disc' structures
- Androecium of flower
- Any other Androecium structures
- Arrangement of the anther on the stamen of the androecium of the main inflorescence flower
- Flower petal length
- Flower petal pubescent and glabrous

#### 3.2 Editing

- Enable Leaf blade shapes with choice of : cordate, lanceolate, linear, obovate, oval, ovate
- Enable Leaf blade length-width ratio
- Enable Leaf blade base: acute, rounded and truncate
- Enable score(s) to measure leaf blade upper and lower surfaces pubescent and glabrous
- Enable leaflet length (mm)
- Enable male and female flower structures
- Enable Perianth: Calyx: Lobe - Reflexed, Erect

### 3.3 Review Pro Forma

Toggle off View Removed Elements

### 3.4 Scoring

- Open Codnopsis Pro Forma ('codnopsis.xml')
- Collapse nodes and see if can tell where enabled structure exist below top level
- Toggle off View Removed Elements
- Import previous specimen Specimen #101
- Score a new Specimen
- Score Plant Erect, Stipitate
- View entire plant scores of specimen #101 and return to current specimen
- Score Plant Height 10 ft
- Score Flower terminal and axillary
- Score Flower filament usually setose or rarely glabrous
- Score Ovary Not Scored
- Score Style length 5-10 cm
- Score Fruit length 0.3 – 1.5 cm
- Score stem architecture climbing
- Save description
- Scoring Present – Not (discuss)

#### 4. Discussion

- General method
  - Selection from all disabled
  - Initial level of structures open
  - Alphabetical structures
  - State group usefulness
  
- Order of potential Scoring
  
  
- Complex scores
  - AND/OR
  - Freq Mods
  
- Use of warning colours
  
  
- Tree Icons
  
  
- Tooltips
  
  
- Shortening / format of path labels
  
  
- Renaming structures
  
- Incrementer for quantitative scores
  
- Single click enlarge tree
  
- Rename To relative modifier / value of this
  
  
- Other Thoughts

#### 5. Note

- Use of Definitions
- Tool tips
- RC Menus
- Tree icons
- Expand/contract

## Appendix E: Wide User Tests

### Results Summary

- Specialisation:
  - Improved ability to cope with choices with all initially disabled
  - Continued comprehension and ability to navigate structure hierarchy with expert domain knowledge
  - Initial confusion over selection methods of attributes/values...quickly understood though
  - Difficulty finding desired state terms because property is not always obvious
  - Lots of technical terms only used for individual families, complicate moving through structure hierarchy, especially if not familiar with them.
  - Some property choices were contentious; particularly the 2D vs 3D shape split with strong opinions on whether would use 2D and 3D shapes together.
  - Warning icons were noticed
  - *Desire Type structures to form some characters*
  - *More cloning support to avoid getting confused as to which is which*
- Data Entry
  - Scoring straightforward...split by structure seems to fit with taxonomic working pattern
  - Order however does not fit with working practice
  - Full task user tended to alter working practice to fit with default task order, following next structure buttons. Other users (with more description experience) expressed desire not to alter working practice.
  - Occasional wish for multiple structures on same screen where the multiple structures all refer to individual parts of larger structure character concept e.g. hairiness of leaf upper surface, lower surface, apex, base.
  - Rough timing estimate indicates no significant time increase for scoring.
  - Do not expect to see sub-structures when have scored parent as not present.
  - *Wish to differentiate between scores that have the same state but one is more so than the other (e.g. sharply vs finely serrate)*
  - *Wish easier method of noting exact location of a state (e.g. where on leaf the hairs are), rather than having to predetermine all possible regions in specialisation*

### Development Priorities

- Spatial Modifiers at PF Creation
  - Simple spatial mods at scoring e.g. at base, at apex, at upper surface, at lower surface
- Type attributes support
- Concrete structures support
- Search for term
  - Simple searches for structure and value
  - Advanced term search, incl disallowed and types and synonyms
- Picture definition support
  - incl. Picture scoring boxes
- Present/Absent score for each Structure.
  - Only score if present.

## Appendix E: Wide User Tests

- Include effect on scoring of dependent lower structures.
- Changing Task Order
- Improve relational specification interface
  - Ratio attributes to come under relational header in attribute-values tree
  - Improve depiction of relational path data
- Fix usability issues in Attribute/Value Selection
- Alternative Single Click Expansion/contraction
- Increase identification of structure path at scoring
- Feedback on what scored
  - Structure hierarchy icons
  - Other summary
  - OR-ing
- Clone support
  - Customise structure name display labels
  - Fix single state score at PF creation.

## 2<sup>nd</sup> Wide User Test: Creating the Pro Forma

- 13 RBGE taxonomist users (10 inexperienced in system, 1 semi-experienced, 2 experienced). Varying levels of domain experience, but all representative end-users.

During the second wide user test, users were timed in completing a number of tasks including specialising the angiosperm ontology for a project based on the *prunus* group of plants. Then they entered data for 1-2 actual specimens (chosen randomly from a shortlist of 4 appropriate specimens with a similar level of detail.) Users were encouraged to talk through their tasks using a think-aloud methodology.

As each user would not necessarily have the requisite level of detailed background domain knowledge of this particular group of plants to complete this task, they were given a list of taxonomic characteristics in the form of common terms used in the related subject literature for describing the inflorescence. An expert who had previously worked with the group of plants created the list. The list utilised terms to be found in the ontology (other elements of the testing procedure dealt with issues of searching for terms not in the ontology).

This part of the wide test was restricted to specialising a representative section of the ontology due to time constraints. The inflorescence section was chosen, as it is the most detailed and most commonly important discrete area of flowering plants for taxonomic description purposes. It is not believed this would significantly affect results as the inflorescence forms a discrete area of the **description object hierarchy** and expert botanical knowledge can readily distinguish its constituents from other areas. The resulting **specialised domain models** contained a representative sample of specialised **attributes** based on using the system for specialising in other narrow tests.

Users were also encouraged to specify extra characteristics from their own domain knowledge but the time cost for these extra operations are excluded from the timing results.

11 users were timed in completing the resultant tasks. These users were all taxonomists of varying levels of experience who were unfamiliar with the system. They all received an explanation and walkthrough of the interface before the tasks and were given an initial introductory guided task for specialising two simple **attributes** before timing began. The same task given to 2 users (angiosperm ontology developers) who were very familiar with the system and ontology.

The users' test data was all saved for later analysis.

Follow-up interview was immediately carried out to get feedback and follow-up any questions arising from the test.

## Ontology Based Proforma Creation and Scoring Interface RBGE User Test: June 2004

- Alan Cannon – Napier University

### Part 1 – Introduction

#### Introductions

- Names
- Purpose – test an interface to create proformas and record base specimen description data for capture in an ontology supported database

#### Explain test procedure

- Test of system interface & method, not of person
- Speak aloud methodology – what you are doing and why. Comments & feedback encouraged.
- Time expected 60 mins

#### Explain system & methodology

- Open PFVis
- Load pf-test.xml
- Proforma Creation
  - Structure Hierarchy
    - All possible permutations of sub-structures supported by ontology
    - Generic structures/regions
    - Start all disabled, enable what desired
    - How to expand/contract incl. Contract button
    - How to search. May be multiple matches for search string.
    - How to enable – double click toggle or RC menu or enable a value for it
    - How to change scoring label name
    - How to change order
    - Definitions access – Mouse over or RC menu for box
    - Icons: Red – Nothing to be said, Green – Something can be said, blue – Scored, Purple - absent
  - Attribute/Value Hierarchy – for each structure
    - All values and attributes. Attributes can be measurements or groups of related states or relational stuff
    - How to enable – double click to toggle or RC menu or enable value for attribute
    - Colours – Red Attribute for not scorable (no values), Green for enabled
    - Definitions access and search as structures
  - Edit panel
    - Change name



## Appendix E: Wide User Tests

- Preferred unit of measurement
- Spatial modifier – specify another structure in proforma to relate this attribute to
- Relational modifier – ratios and relative values (>, <, =)
- Fix Score – fix the states for this attribute as always scored in this proforma
  
- Scoring Interface – brief
  - Structure Hierarchy – much as before
  - Specimen Panel – Scores – Default specimen is sample to see how looks.
    - Presence
    - Modifier boxes
    - AND/OR
    - Not Scored
    - Next Structure
  
- Definitions Explorer – alphabetical listing of all terms

## Part 2 – Proforma Creation

### Scripted Creation:

- Scripted create: quantitative (Plant height – mm)
- Scripted create: (Inflorescence: terminal; axillary, lateral)

### Create Proforma for Inflorescence:

- Inflorescence: racemose; solitary; fasciculate
  - Number of flowers per inflorescence
  - Pedicel length
  - Petal: obovate; suborbicular; oval; ovate
  - Petal: white; red; orange-red; purple
  - Petal length; width
  - Calyx lobe: triangulate; oblong; ovate
  - Calyx lobe length; width
  - Calyx lobe apex: rounded; acute
  - Sepal: lanate; glabrous; pubescent
  - Hypanthium: campanulate; tubular; cup-shaped; funnellform
- Add 1-2 of own features if desired
  - Change structure order if desired
  - Save proforma

### Optional supplementary characters if time permits

- Petal length : width ratio
  - Bud scales surrounding hypanthium
  - Hypanthium length; width
  - Corolla length; width
  - Sepal: green; red; purple
- Fruit: ellipsoid; globose
  - Fruit: tomentose; glabrous; velutinous
  - Fruit length; width
  - Fruit: orange; green; red; blue; black; purple; yellow-orange; purple-violet; violet; purple-red
- Petiole length
  - Petiolar glands: present, absent
  - Leaf blade: elliptic; lanceolate; oblong; ovate
  - Leaf blade length; width
  - Leaf blade apex: acuminate; acute; rounded
  - Leaf blade base: cuneate; rounded; subcordate,
  - Leaf blade margin: entire; serrate; serrulate; crenate
  - Leaf blade: glabrous; pubescent
  - Under-surface of leaf color hue: lighter than; darker than; equal to lower surface
  - Leaf domatia present

### Part 3 – Score Specimens

**Load previously create proforma or use pre-generated one (prunus220604.xml)**

#### **Explain scoring procedure**

- Structure Hierarchy – much as before
  - Icons – blue & purple
  - Cannot edit
- Specimen Panel – Scores – Default specimen is sample to see how looks.
  - Presence
  - Modifier boxes
  - AND/OR
  - Not Scored
  - Next Structure

#### **Scripted Scores**

- Scripted score plant height
- Scripted score selection inflorescence terminal / axillary

#### **Score 1-2 specimens**

**Save specimens normally and easy xml**

#### **Advanced Scripted Scores (if not done and time permits)**

- Use modifiers in score
- Use AND in score
- Use OR in score
- Use auto absent -fruit not present
  
- Load pro forma concrete-test.xml
- Concrete score – petal

**Check Specimens – walk through simple xml to see what scored**

## Part 4 – Discussion

- Ability to express / constraints on expression
- Ability to find terms
- Ease of Proforma creation
- System feedback on proforma creation status
- Ease of scoring
- System feedback on scoring status
- Definitions access
- Pictorial scoring
- Pictures With or without names
- Anything other feedback
  
- Thanks

## Proforma Creation Task

**Create Proforma for a group of specimens** (including *Prunus cornuta*, *P. cerasoides*, & *P. persica* specimens from Bhutan)

1. Create score for Entire Plant Height in meters.
  - a. Find and click on Entire Plant in Structure Hierarchy
  - b. Enable measurement 'height' in central panel
  - c. Select mm in preferred unit box
2. Create score for Inflorescence: terminal, lateral or axillary
  - a. Find and click on Inflorescence
  - b. Find and enable the possible values terminal, lateral and axillary (found in attribute Arrangement:Position:General)
3. Using the below hypothetical list of terms, create the remainder of the proforma for the inflorescence and its substructures:

**Terms commonly used in this subject area about inflorescence (from initial knowledge and literature):**

- Inflorescence: racemose; solitary; fasciculate
  - Number of flowers per inflorescence
  - Pedicel length
  - Petal: obovate; suborbicular; oval; ovate
  - Petal: white; red; orange-red; purple
  - Petal length; width
  - Calyx lobe: triangulate; oblong; ovate
  - Calyx lobe length; width
  - Calyx lobe apex: rounded; acute
  - Sepal: lanate; glabrous; pubescent
  - Hypanthium: campanulate; tubular; cup-shaped; funnelform
4. Add 1-2 other likely characteristics you might wish to score about this group from your own knowledge. (This need not be taxonomically accurate for this type of specimen).
  5. Review the proforma.
  6. Alter the default structure scoring order of the proforma using the move up and move down arrows on the menu bar, if desired.
  7. Save the proforma as your-name.xml

## Supplementary Proforma Creation

**Terms commonly used in this subject area about inflorescence, infructescence and leaf (from initial knowledge and literature):**

- Corolla length : width ratio
  - Use the Relative Modifier box to create Petal Length: Petal Width ratio
  - Select appropriate units of measurements (if any)
- Create score for Petal length : width ratio (requires second length score)
  - Right click on Petal and select Add New Attribute from menu to create new length score.
- Bud scales surrounding hypanthium
- Sepal: green; red; purple
- Hypantheum length; width
- Fruit: ellipsoid; globose
- Fruit: tomentose; glabrous; velutinous
- Fruit length; width
- Fruit: orange; green; red; blue; black; purple; yellow-orange; purple-violet; violet; purple-red
- Petiole length
- Petiolar glands: present, absent
- Leaf blade: elliptic; lanceolate; oblong; ovate
- Leaf blade length; width
- Leaf blade apex: acuminate; acute; rounded
- Leaf blade base: cuneate; rounded; subcordate,
- Leaf blade margin: entire; serrate; serrulate; crenate
- Leaf blade: glabrous; pubescent
- Under-surface of leaf color hue: lighter than; darker than; equal to lower surface
- Leaf domatia present

## Score Specimen Task

**Score specimen** (one of *Prunus cornuta*, *P. cerasoides*, & *P. persica* specimens from Bhutan)

1. Unselect Proforma Editing View (under View menu)
2. Score a new specimen
  - a. Click on Score New specimen on menu bar (or under Scoring menu)
  - b. Enter new specimen details
3. Score Entire Plant Height
  - a. Click on Specimen # tab
  - b. Enter score for plant height
  - c. Click Next Structure
4. Select relevant checkbox(s) under Inflorescence – Arrangement:Position:General for terminal and/or axillary
5. Continue to score the specimen till complete proforma
6. Check all structures scored
7. Save description as your-name-specimen#.xml
8. Export Viewable Description (under File menu)
9. Review description in web browser