GRAPHICAL USER INTERFACE REPRESENTATION AND GENERATION USING SYSTEM ENTITY STRUCTURES

By

Lahiru Ariyananda

_____

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN COMPUTING ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

2007

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of the requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: _____

APPROVED BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

_____ _____

Bernard P. Zeigler Date Date
Professor of Electrical and Computer Engineering

ACKNOWLEDGEMENTS

My heartfelt gratitude goes to my advisor, Professor Bernard Zeigler, for his invaluable and reassuring advice, and for the freedom he gave me to focus on areas of my interest which made this study not only educational but also enjoyable for me. I would like to take this opportunity to specially appreciate and remember the kind support given by Dr Doohwan Kim and Mr Robin Moore during the course of this study. I would also like to extend my sincere gratitude to Dr Roman Lysecky and Dr Susan Lysecky for their flexibility and time taken to serve on my defense committee. Last but not least, many thanks to all my colleagues at ACIMS lab for patiently sharing my moments of joy and temporary anguish.

DEDICATION

To my late beloved Father

Your undying love, guidance and encouragement will always be my inspiration.

TABLE OF CONTENTS

LIST OF FIGURES **7**

LIST OF TABLES **8**

ABSTRACT **9**

CHAPTER 1 INTRODUCTION **10**

1.1 OBJECTIVE 11
1.2 MOTIVATION 11
1.3 CHAPTER INTRODUCTIONS 12

CHAPTER 2 SYSTEM ENTITY STRUCTURES IN A NUTSHELL **13**

2.1 DEFINITION AND STRUCTURE OF AN SES 13
2.2 AXIOMS 15
2.3 VISUALLY DEPICTING AN SES TREE 16
2.4 PRUNED ENTITY STRUCTURES 18

CHAPTER 3 BACKGROUND & IMPLEMENTATION CHOICES **21**

3.1 PROGRAMMING LANGUAGE & GRAPHICAL LIBRARY 21
3.2 SWT MAPPING TO SES TREE 22
3.3 DATA MODEL 27
3.4 INTRODUCTION TO XML 27
3.5 TRANSLATING THE VISUAL SWT SES TO TEXTUAL XML 30
3.6 NATURAL LANGUAGE DESCRIPTION OF AN SES 33
3.7 NATURAL LANGUAGE TO XML PROCESS 37

CHAPTER 4 USER MODIFICATIONS: PRUNING & EDITING **41**

4.1 PRUNING THE SWT SES 44
4.2 ENTERING DATA INTO VARIABLES : 45
4.3 VALIDATING THE PES 48

CHAPTER 5 AUTO GENERATION OF GUIS **51**

5.1 SWTPARSER 51
5.2 THE TESTING AND EXECUTION ENVIRONMENT 52
5.3 GUI CODE GENERATION PROCESS 53

TABLE OF CONTENTS - Continued

CHAPTER 6 SES TO GUIS 58
6.1 MAPPING A SES TO THE SWT SES 60
6.2 GROUPS & FILL LAYOUTS 64
6.3 FURTHER REDUCING PRUNING AND MODIFICATION TIME ? 67
6.4 GENERATING PROFESSIONAL GUIS USING SWTPARSER 69
SUMMARY 74
RELATION OF THE SES AXIOMS ? 76
CONCLUSION AND FUTURE WORK 78
APPENDIX 1 SWT SES IN NL 80
APPENDIX 2 CANDIDATE PES IN XML GENERATED FROM DTD 83
APPENDIX 3 SIMPLEGUI1.XML 99
APPENDIX 4 SIMPLEGUI2.XML 102
APPENDIX 5 EXGUI3.XML 109
APPENDIX 6 EXGUI4.XML 116
REFERENCES 122

LIST OF FIGURES

Figure 1 SES of a Computer 16
Figure 2 A PES of the Computer SES 19
Figure 3 PES of a Book 20
Figure 4 SES of a basic SWT tree 23
Figure 5 Alternate SES of a Basic SWT Tree 24
Figure 6 Alternate view of SWT SES 26
Figure 7 XML code fragment 28
Figure 8 Process flow : Visual SES to XML 31
Figure 9 Shell described in NL 35
Figure 10 Tabfolder Description in NL 35
Figure 11 Decomposition of “Components” into widgets 36
Figure 12 Segment of SWT SES in NL 36
Figure 13 Vitual Work Table Web Interface 37
Figure 14 Virtual Work Table panels 1 & 2 38
Figure 15 Virtual Work Table panes 1 - 3 39
Figure 16 Virtual Work Table panes 1 -4 40
Figure 17 One PES for SWT SES 42
Figure 18 SimpleGUI1 43
Figure 19 Altova XMLSPY 45
Figure 20 Validate PES in Virtual Work Table 48
Figure 21 Process flow : PES in XML to GUI 51
Figure 22 Eclipse IDE with SWTParser open 53
Figure 23 SimpleGUI1 executed 55
Figure 24 SimpleGUI2 56
Figure 25 ExGUI3 57
Figure 26 Variant of Computer SES 59
Figure 27 PES of SWT SES with Group 61
Figure 28 PES of SWT SES with Group 63
Figure 29 ExGUI4.xml code fragment 64
Figure 30 EXGUI4 66
Figure 31 A semantically correct version of EXGUI4 67
Figure 32 Another ExGUI4.xml code fragment 68
Figure 33 GENETSCOPE Experimental Frame 70
Figure 34 GENETSCOPE EF mapping to SWT SES 71
Figure 35 PES for GenetScope Experimental Frame 73

LIST OF TABLES

Table 1 SES concept summary..... 14
Table 2 Attach variables for Widgets..... 33

ABSTRACT

System Entity Structures have been mainly used as a language for hierarchical-system modeling, simulation and knowledge representation in the past. In this work the possibility of extending its knowledge representation features to account for Graphical User Interfaces is investigated. The fact that a basic Standard Widget Toolkit (SWT) library can be mapped into a SES tree based on the concepts of hierarchical decomposition and specialization is identified. On the base recognition that SES is an ontology framework that is closer to XML, it is used as the data storage model to make use of its salient features so that sufficient and quantitative knowledge is represented by it to auto generate a GUI. The GUI generation will be done using a tool named SWTParser which was specifically developed for this study. SWTParser reads in a user customized Pruned Entity Structure in XML format to generate fully executable GUI code in java. Also a simple methodology that would map a given instance of a physical system defined in SES format to a functional GUI is presented. Finally, methods to reduce the over head time involved in the user customization process and possible expansions to the current framework will be discussed.

CHAPTER 1 INTRODUCTION

Graphical User Interface design and Application Programming are like Abbot and Costello. Without each complimenting the other, their effectiveness is circumscribed in modern day software development. It could be a tedious task for a developer to create especially larger GUIs using code descriptions of a programming language. Though there are tools developed by software development companies which allow a user to develop GUIs without the direct involvement of a programming language, they are mainly reliant on the user's creativity. Hence GUI development can be considered an Art as well as a Science.

Though the behaviors and characteristics of Hierarchical Knowledge-Based systems are well understood [1], very little research has been done in the past two decades which involves methodologies for GUI development using such systems. User Management Systems are often used to help user interface designs. [2] However, a few User Interface Management Systems exists which are designed explicitly for hierarchical knowledge based systems [3][4].

1.1 Objective

The main objective of this research would be to exploit the hierarchical nature of a knowledge representation language called the System Entity Structures(SES) [1]and study the possibility of mapping a graphical library into the SES format so that sufficient and quantitative knowledge is represented by it to auto generate a GUI. Also a simple methodology will be presented to map a given instance of a physical system described using an SES to a translated GUI. It should be clearly noted that a full fledged implementation with all bells and whistles was not the intention of this research. It is the firm belief of the author that, once a basic foundation has been laid, expansions and innovations could be introduced as future endeavors.

1.2 Motivation

The initial motivation for this research was derived when the author was working on the GENETSCOPE project [5] to study a feasible method for mapping its existing GUI to a System Entity Structure mainly as knowledge representation mechanism. Further interest was generated to study the feasibility of reversing this process to recreate GUIs from the SES information.

1.3 Chapter Introductions

SES is a modeling language that describes hierarchically decomposable physical systems such as computer systems, buildings, and robot systems [2]. SES can also be applied to describe conceptual hierarchical systems such as tree systems and special-purposed languages [6]. Chapter 2 will introduce the concepts and axioms behind SES through examples.

As mentioned before, in order to represent a GUI through an SES, a mapping of a graphical library needs to be made. Also as SES are usually depicted as visual trees, a method to transform the relevant information into an electronic format using a hierarchical data model will have to be introduced. Chapter 3 will be a discussion on the above processes and choices made on the implementation framework.

Chapter 4 will venture out to acquaint the user with pruning and modifications processes needed to make the aforementioned SES into a viable structure for data extraction for GUI generation. Chapter 5 will attempt to introduce the usage of the SWTParser tool, which was developed as a part of this study to generate GUIs from the above data files.

Chapter 6 will introduce a methodology to map an existing SES based on a physical system to a GUI from the information discussed in the previous chapters. Also a discussion will be presented to reduce the overhead time needed to do the tasks in chapter 4 with its pros and cons.

CHAPTER 2 SYSTEM ENTITY STRUCTURES IN A NUTSHELL

2.1 Definition and Structure of an SES

As the crux of this research depends on using a System Entity Structure (SES) as the model of choice for data representation and storage, this chapter will venture out to introduce the language and its axioms. Zeigler proposed the System Entity Structure as a language for hierarchical-system modeling, simulation and knowledge representation in [7][1]. This study will be mainly focusing on its use as an effective way of knowledge representation and data storage with the aforementioned ultimate goal of GUI generation.

If one is to quote the keywords of the basic structural composition of an SES, they would be “Entity”, “Variable”, “Aspect”, “Specialization” and “Multiple Aspect”. Part of the following summarized explanation was extracted from [8]. Entities represent things that exist in the real world or sometimes in an imagined world. A house would be an example. Aspects represent ways of decomposing things into even smaller ones. An example would be living room, bath room, kitchen etc of a house. Multi-aspects are aspects for which the components are all of the same kind. In other words multi aspects can be regarded as a special case of an aspect in which the entities of the aspect are homogeneous in nature. An example would be rooms. Specializations represent categories or families of specific forms that a thing can assume. A family of colors or

size would be an example. Variables are an attribute of an Entity. The address of a house would be a variable. The below table extracted from [8] summarizes the basic concepts mentioned above.

item	**denotes**	**when to use**
entity	a thing in the real or modeled world	Use to represent a thing that stands alone or is a part or variant of another thing.
aspect	the relationship between a thing and its components when decomposed from a certain perspective	Use when you want to represent an "and" connective among sub-things of a thing – where the "and" denotes the necessity that all of the sub-things must appear together to comprise the thing.
multi-aspect	a special kind of aspect in which all the components are homogeneous in nature	Use for the same objective as an aspect except that the components are all from the same classes.
specialization	the relationship between a thing and its variants from a given family	Use when you want to represent an "or" connective among sub-things of a thing – where the "or" denotes the fact that a choice of one of the variants can replace the thing.

Table 1 SES concept summary

2.2 Axioms

The SES was originally characterized by its axioms by Zeigler [1] and later by Zhang and Zeigler [9]. Accordingly the SES was defined as labeled tree with attached variable types which satisfies the following axioms :

uniformity: Any two nodes which have the same labels have identical attached variable types and isomorphic sub trees.

strict hierarchy: No label appears more than once down any path of the tree.

alternating mode: Each node has a mode which is either entity, aspect, or specialization; if the mode of a node is entity then the modes of its successors are aspect or specialization, if the mode of a node is aspect or specialization, then the modes of its children are entity. The mode of the root is entity.

valid brothers: No two brothers have the same label.

attached variables: No two variable types attached to the same item have the same name.

inheritance: every entity in a specialization inherits all the variables, aspects and specializations from the parent of the specialization

2.3 Visually depicting an SES tree

The most common way to visually depict an SES is by a tree structure. In this depiction, the items (entities, aspects, specializations, and multi-aspects) are displayed as nodes and the relationships of aspects, specializations and multi-aspects to entities are shown as vertical lines, arrows, and triple parallel lines, respectively [8]. A key observation of a visual tree SES would be that entities alternate with the other items. Let us consider the SES in the below figure to further clarify the concepts mentioned before.

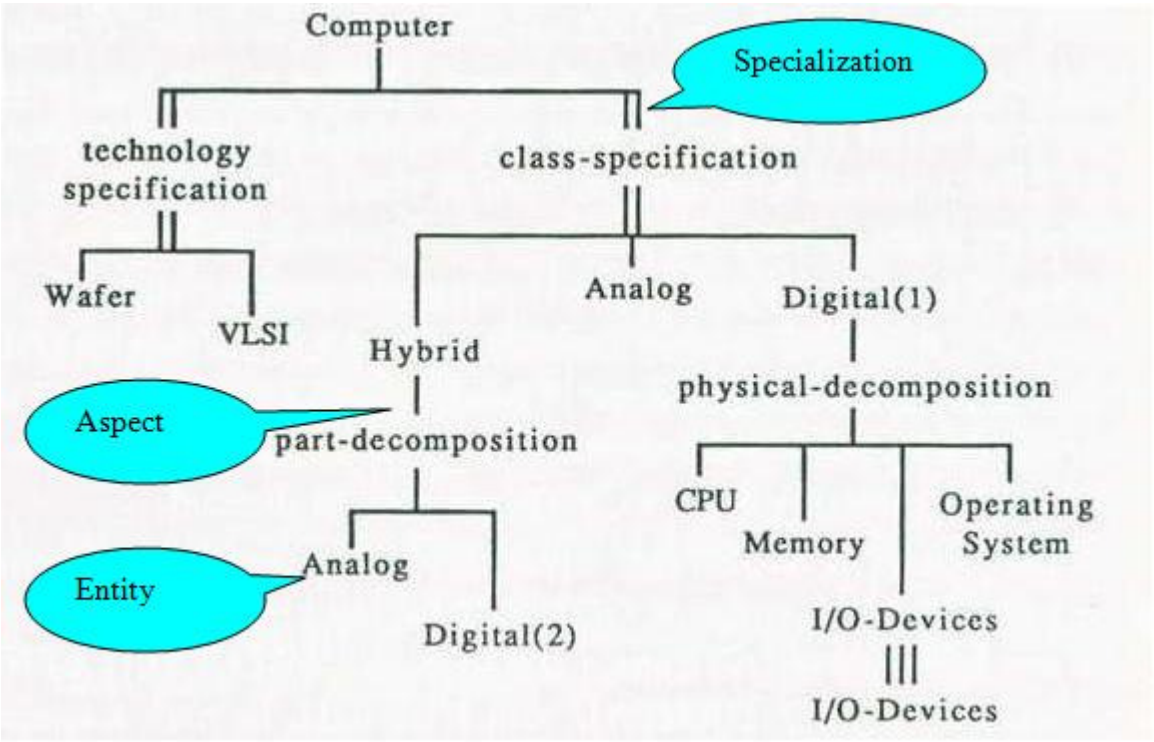

Figure 1 SES of a Computer

The “Computer” is called the root entity and it has two specializations shown by two vertical lines, called “class-specialization” & “technology-specialization”. The class-spec has entities “Analog”, “Digital”, and “Hybrid”. In such a specialization relation, Computer is referred to as a generic type relative to the entities, Analog, Digital, and Hybrid, which are called special types [7]. Though the special types can have their own distinctive attributes, they inherit all of the attributes (variables and substructures) possessed by Computer.

Hybrid special type is shown as having “part decomposition” into two components, namely Analog and Digital. As mentioned before these components are now entities themselves. Likewise Digital has a “physical-decomposition” in four components namely, CPU, Memory, I/O-Devices and Operating System. By the uniformity axiom, the Digital part of a Hybrid computer has the same physical-decomposition shown under the occurrence of Digital as a special type of Computer. It should also be noted that Hybrid, being decomposed into Digital and Analog components, inherits properties from both through the *uniformity axiom*.

In the above figure , the triple vertical bars connecting “I/O-Devices” and “I/O-Device” depicts the special type of aspect decomposition discussed before called the multiple decomposition. A multi-aspect decomposition is used to represent entities whose

number in a system may vary. For example a Digital computer may have 0, 1, 2, or more I/O-Devices.

2.4 Pruned Entity Structures

A pure entity structure is defined as one having no specializations and at most one aspect hanging from every entity [7]. The pruning process is used to create such a pure SES. The Pruned Entity Structure (PES) is the intermediate result of the pruning process which would contain fewer aspects and specializations than the original and therefore specifies a smaller family of alternative models than the latter. The eventual termination to the pruning process will result in a pure entity structure that specifies the synthesis of a particular hierarchical model.

If we consider the SES in figure 1, an example of a Pruned Entity Structure of the original is shown in figure 2 below.

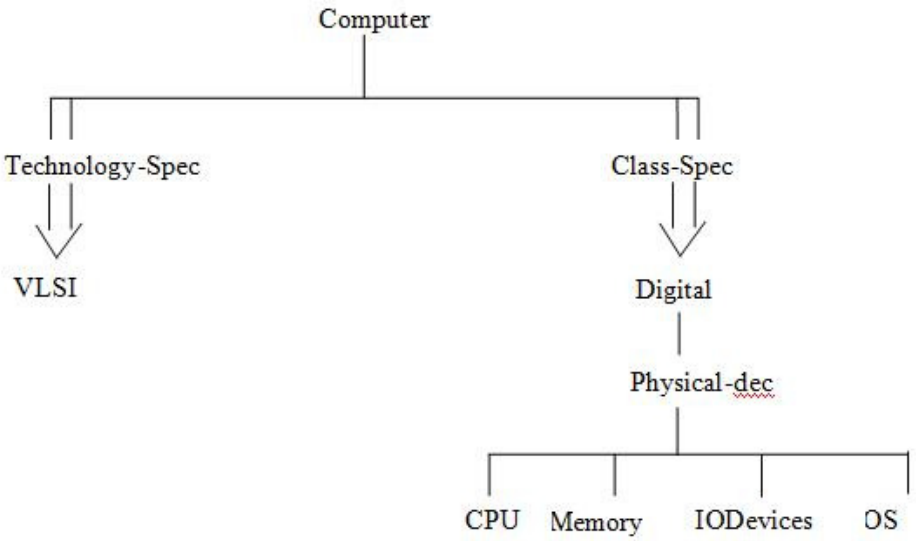

Figure 2 A PES of the Computer SES

Let us consider another instance where an entity of an SES can have more than one aspect attached to it. The following example was extracted from [8]. Fig 3 shows an SES for a book that provides two aspects, contentDec and physicalDec, corresponding to decompositions from the perspectives of content and physical constitution, respectively.

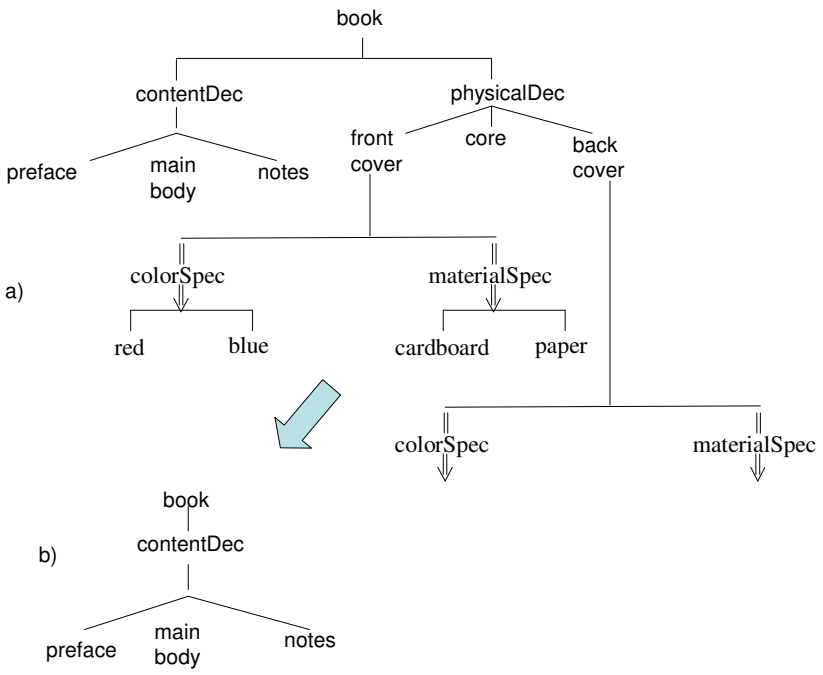

Figure 3 PES of a Book

One example PES of the SES shown in a) is would be b) in the above figure. Another valid example of a PES for the above SES would be if one chooses the physicalDec aspect and then chooses a single color (ex red) for the colorSpec and a solitary material (ex paper) for the materialSpec. Hence, pruning of an SES can create not just one PES but a family of valid PESs. Pruning will be revisited in chapter 4.

CHAPTER 3 BACKGROUND & IMPLEMENTATION CHOICES

3.1 Programming Language & Graphical Library

In choosing a relevant programming language to implement the objective of this research, as with many other instances, there were many choices. As the implementations will mainly involve the generation and testing of Graphical User Interfaces (GUIs), the leading contenders could be narrowed down to C/C++, Java and Visual Basic (VB) amongst many. Out of them, Java was the author's language of choice due to several factors. The author's experience of the ease of using Java's strong graphics was the main forerunner. Alongside this reason, other benefits such as Platform Independence, Object Orientation and Distributed Computing were key deciding factors which would also pave way for future expansion and research.

Once the preference has been made in choosing Java as the programming language, it should be deemed equally as important to select a GUI library to support it. At this juncture it should be clearly noted that a full fledged implementation with all bells and whistles, is totally out of context of this research. It is the firm belief of the author that, once a basic foundation has been laid, expansions and innovations could be introduced as a future endeavor. Keeping this in mind, SWT (Standard Widget Toolkit) was chosen over Swing. The following discussion would serve as a justification as to why SWT was chosen as the GUI library of choice.

3.2 SWT Mapping to SES tree

As it has been discussed before, System Entity Structures clearly follow a hierarchical format. Visualizing SWT and Swing from a top down structural format, the author observed that SWT has a more simplified hierarchy compared to Swing. As mentioned above, this simplified structure suits favorably the implementation objectives. The below figure 4, depicts a basic SWT hierarchy visualized and mapped into the SES format. It should be noted that this is not a comprehensive SWT widget/component mapping into SES, but only a simplified version which would suit the research objectives.

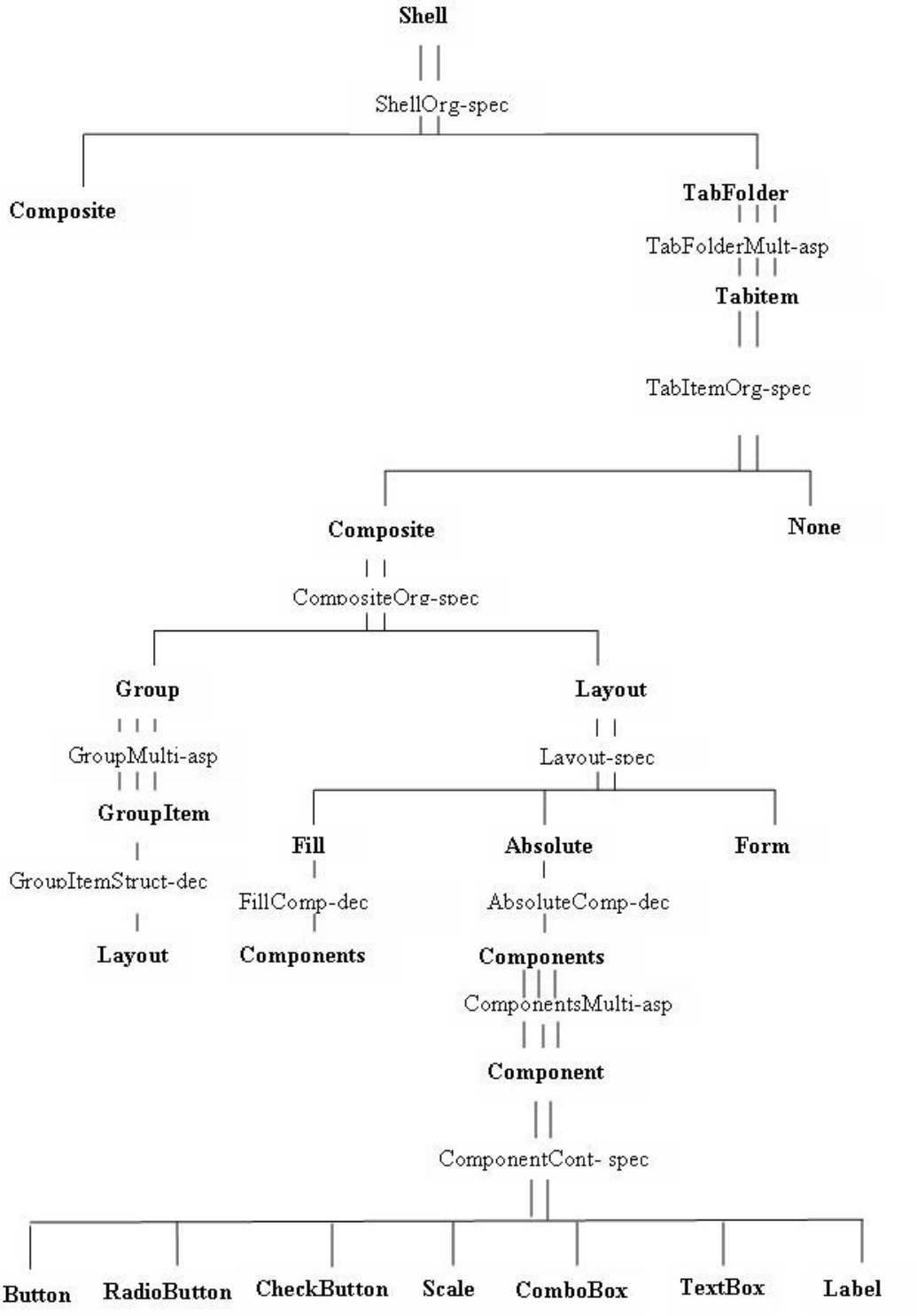

Figure 4 SES of a basic SWT tree

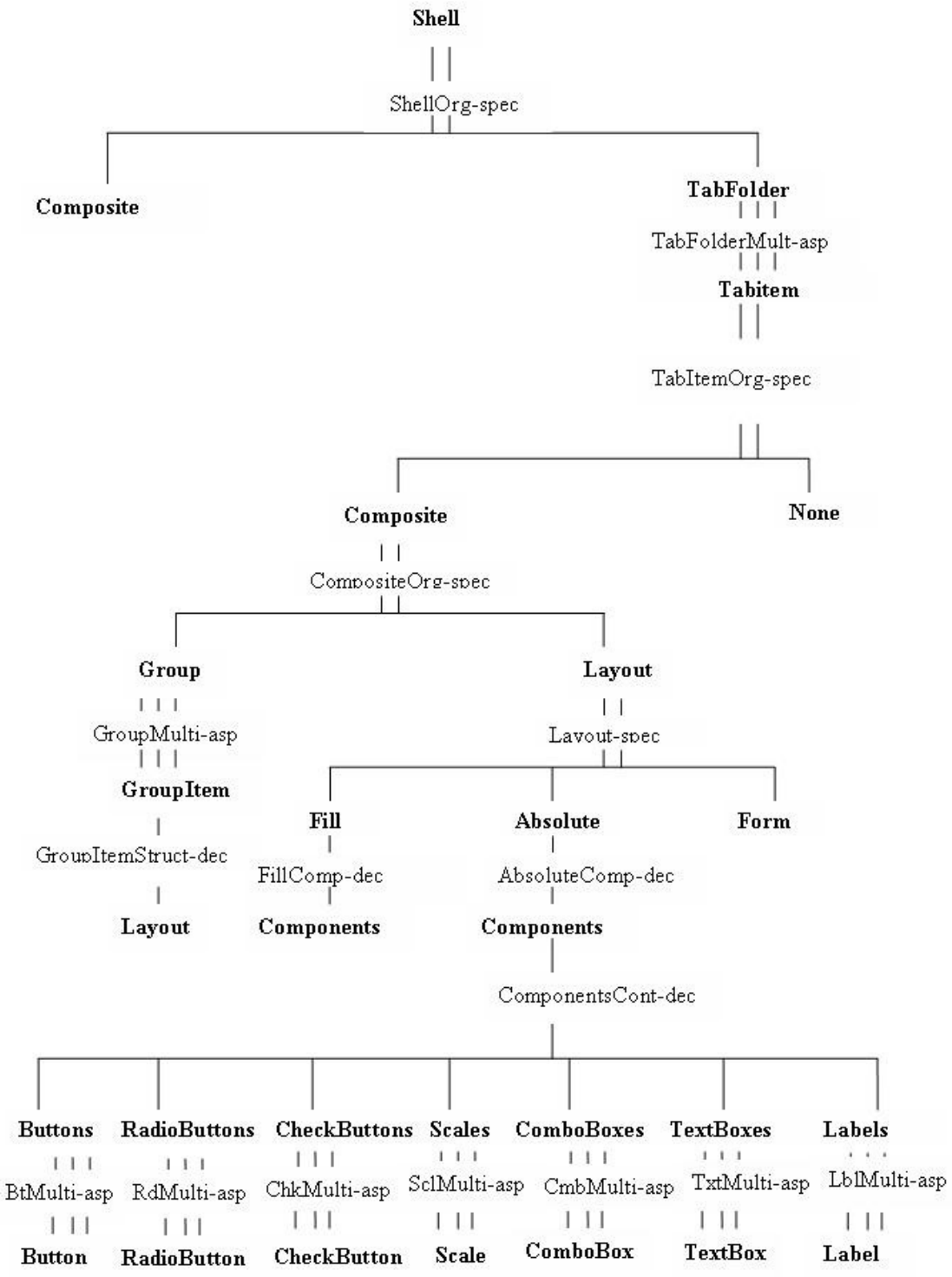

Figure 5 Alternate SES of a Basic SWT Tree

When developing this SES, some factors had to be taken into consideration. Given a hierarchical system, it could be understood, that the SES which could be formed from it can vary from the view of its creator. In other words, there can be several implementations of an SES derived from an open ended hierarchical system depending on the perspective of the creator. For example the SES shown in figure 5 was finalized after several initial attempts at deriving different versions from different perspectives. Figure 5 is an alternate way of representing the same SES of figure 4 by restructuring the multi-asp at the "components" level. The main advantage of using the "restructures" figure 5 version over figure 4 is twofold. Depending on the user preference, the components widgets at the leaf level of the tree (Button, TextBox etc) may or may not be used in a GUI. Hence as mentioned before in the sub section 2.3, defining them as a multiple decomposition as opposed to a specialization will allow for zero (0) items of the widget when necessary. Also defining them as entities of a specialization (ie figure 4), will only let the user pick one of the widgets at the pruning level. What if the GUI consists of multiple widgets? This restructuring process is further discussed in [. On the same note, a valid question could be raised as to why the "TabFolderMult-asp" was not restructured in the same spirit in figure 5. Also note that the "TabFolder" could have being decomposed into several "Tabitems" as shown in figure 6 as opposed to the multiple aspect decomposition in figure 4. Though both these versions are also well within the allowable syntax of System Entity Structures, the author foresaw problems of doing so with respect to future objectives. For one, these methods would pose a

restriction to the number of "Tabitems" a user can have in his/her GUI. It would be only three in figure 6 case. What if the user wanted to have 5 or 6 tabs? If the "TabFolder" multi-asp were restructure as in the "Components" case, the concept of a "Tabitem" will be lost and would be very confusing to the user . Also, as it would be discussed later, when this visual SES is converted to some data model, which would allow for data extraction and processing, there will be complexity issues deriving data from it. In both these versions each "Tabitem" is treated as a separate entity which would complicate things as opposed to the *multi- aspect* version where all "Tabitems" are considered to possess the same characteristics. Also the multi-aspect perspective would allow the ultimate user to define the number of tab items according to his/her preferences.

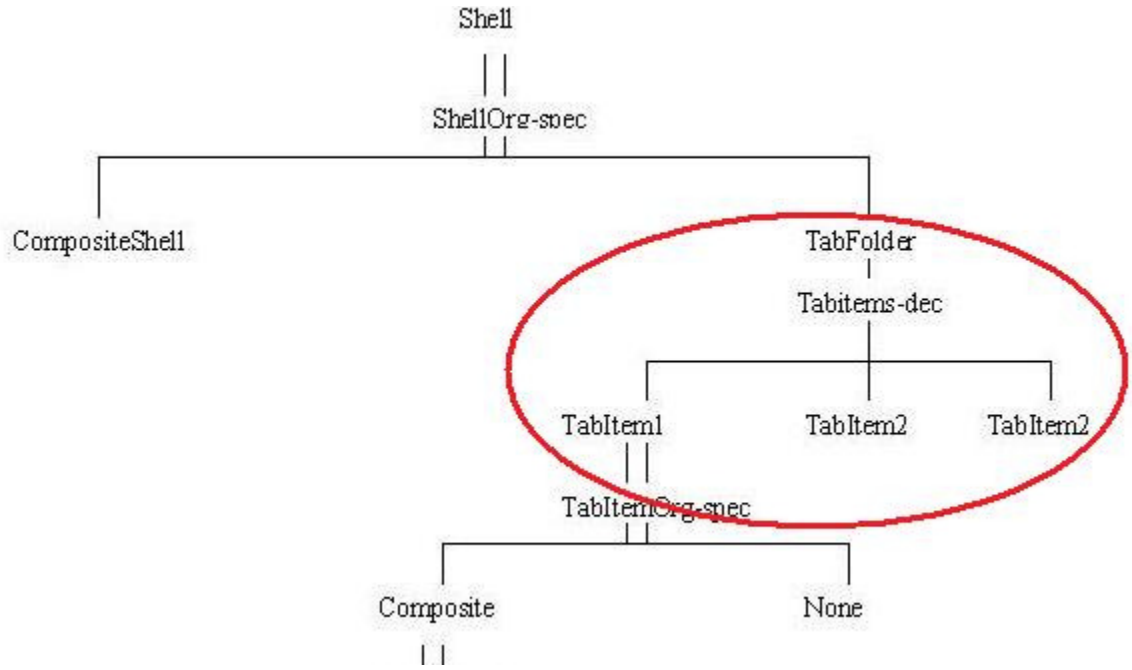

Figure 6 Alternate view of SWT SES

3.3 Data Model

Once the System Entity Structure mapping of a SWT GUI tree has been visualized and established as above , the information need to be stored in some data model so that it can be read and processed by a programming language (ie Java in this case) easily. In researching for a data model to meet the needs of the research objective, a key factor needs to be kept in mind. As all SES mappings follow hierarchy, a data format that would support tree structures would be naturally the best contender.

The eXtensible Markup Language (XML) was chosen as the best fit due to the above reason and the reasons mentioned later. Before going any further, a very brief introduction to XML would be considered appropriate at this stage.

3.4 Introduction to XML

XML could be described as a "nonprocedural programming language, which means that things written in the language are not so much commands as they are descriptions of a condition or state. Like almost all programming languages, XML is written as human-readable text, in such a form that all humans as well as programs can read and understand the instructions" [10]. XML can markup data so that the reader (either a human or a program) can identify each piece of the data set and determine its characteristics by examining the "tags" it contains. A tag can be named anything the creator of a XML document wishes but the reader (a parser in this case) should be told the meaning of them to retrieve whatever information which is contained within the tag's

“attributes” and “elements”. Tags have to be paired together so that an open tag also has an ending tag. It would be more enlightening to follow these concepts through as example.

```
1  − <textboxes>
2    − <aspectsOftextboxes>
3      − <textboxes-multMultiAsp numContainedIntextboxes="1">
4          <textbox textboundlx="204" textboundly="29" textboundux="111" textbounduy="20" textbgdcolor="white"> </textbox>
5        </textboxes-multMultiAsp>
6      </aspectsOftextboxes>
7    </textboxes>
```

Figure 7 XML code fragment

In the above figure “textboxes-multMultiAsp” in line 3 is contained in a “<” and “>” pair is called a starting tag name which has to be matched with an ending tag as shown. The “numContainedIntextboxes” is a name of an attribute which should be followed by a value (ie 1 in this case). All the information from the beginning of a start tag to the closing of the end tag, and any information in between is called an element. Hence, the whole of line 4 is an element. Though there is one element in this case with the name “textbox” there can be many elements. It should be also noticed that the tag in line 3 is contained within the tag with the name “aspectsOftextboxes” and it within tag with the name “textboxes”. Hence this tree structure maintains the hierarchy which was discussed before.

The term “XML document” which was mentioned before is precisely defined by XML specifications published by the W3C [11] World Wide Web consortium. In very loose terms, an XML document can be viewed as a “well formed” data file model that

meets the W3C requirements and contains a hierarchical tree with tags, attributes and elements. The detailed rules an XML document must follow is out of scope for this brief introduction. A parser that will "read" an XML document will also do checks for their validity [12]. Further more the process we will be using to generate a well formed XML file will be discussed in detail later in chapter 4.

As mentioned before, besides hierarchy, there were a few other reasons for choosing XML as the data model of choice. Since information coded in XML is human readable and easy to understand, it would also prove to be beneficial to the user who will ultimately have to customize and modify data of the figure 5 to include a specific SES which he/she would like to be auto generated into a GUI. This process with examples will be discussed in detail in Chapter 4. As the name says, Extensibility feature of XML will also prove ideal to the objectives as new tags and attributes can be easily introduced when necessary. XML does not need to have an ordered fixed set of tags. Also the robustness feature of XML explained by [13] could be used to advantage as SWT GUI widget names can be tagged and its attributes and elements can be read through a parser. Examples in chapter 4 & 5 will clarify these features.

3.5 Translating the visual SWT SES to textual XML

Though the figure 5 depicting the SWT System Entity Structure might not appear to be too complicated to the human eye, the process involving it's conversion to XML format has to be given a lot more thought. Naturally this process will include multiple steps. The information contained in the figure is only in a visual human readable format and involves a high level of abstraction. It will be necessary to expand the SES with additional explicit data variables introduced at each level where there is an entity in order to make it a valid candidate for the ultimate objective of data extraction and auto GUI generation. For example, at the "Shell" root entity level in figure 5, if a GUI needs to be defined, a minimum of a shell height and width variables should be introduced. If the user is given the opportunity to customize and modify the XML based SWT SES (to be discussed) to include an explicit one, for example maybe of figure 1, he or she should also be given an addition variable to include the name of the root entity, ie "Computer". Hence, it was apparent that the SES needs to be transformed from its visual format to some textual format (along with the inclusion of the additional variables), so that a computer program could read it and convert it to an XML format for data extraction. This process is shown in the below figure.

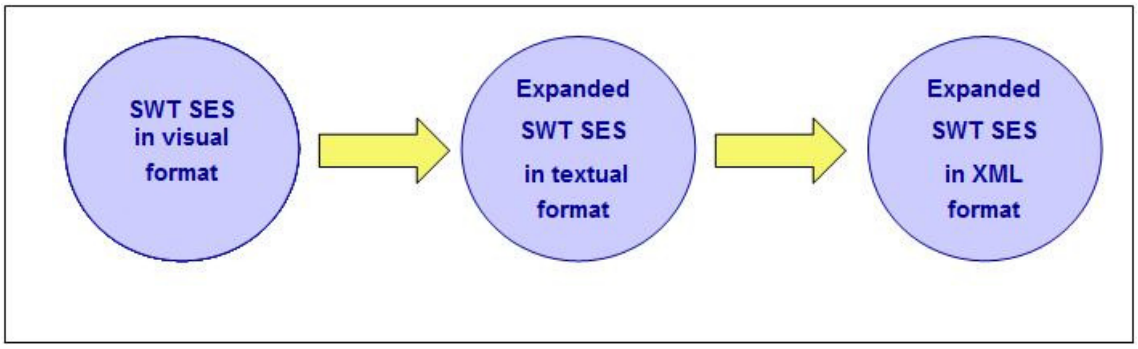

Figure 8 Process flow : Visual SES to XML

The first hurdle was to identify the easiest process to translate the SES information in some text based format. At the second level in the above figure, it should be understood that the SES will grow much larger with the inclusion of the additional variables. Zeigler and Hammond, in the book titled "Modeling & Simulation-Based Data Engineering" identifies that "from a static point of view, the SES is an ontology framework that is much closer to XML than that of Semantic Web ontology [8]. Further more, they go on to describe and offer a tool to automatically generate valid XML Schema and instances from an intuitively developed structured model.

Prior to a discussion of the above process, further research needs to be done to determine what additional variables are needed to be introduced to the SES of figure 5 to make is a viable candidate for data extraction and GUI generation. After a careful analysis and further study of SWT components and widgets, the variables shown in table 2 were introduced. It should be noted that only seven widgets, namely buttons, radiobuttons, checkbuttons, labels, textboxes, comboboxes and sliders were chosen as a test bed for this implementation and their selection were based on their common usage as

seen in regular GUI applications. Also it should be mentioned that each SWT widget is capable of supporting more variables, to accommodate for more customizations than the few selected in this implementation. For example, a button could have an alignment variable or a variable to set its background color or a label could have one to change its font type. The variables chosen (for some of the widgets) were based on the minimum needed to place the corresponding widget in an "absolute layout" environment in SWT. Introducing additional variables to enhance options will be an additional task for the future and not for an experimental research of this caliber. However a few exceptions were made to illustrate the above point and the additional variables used for setting the scale minimum, maximum, increments and page increments will serve as an example. Also the color variable in labels and textboxes are another example.

Entity	Variables
Shell	name ,height,width
TabFolder	tabux ,tabuy ,tablx,tably
TabItem	tabname
GroupItem	groupname
Button	buttonname,buttonboundux,buttonbounduy, buttonboundlx, buttonboundly
CheckButton	chkbuttonname,chkbuttonboundux, chkbuttonbounduy, chkbuttonboundlx, chkbuttonboundly
ComboBox	comboboundux, combobounduy, comboboundlx, comboboundly
Labels	lblname,lblbgdcolor,lblboundux, lblbounduy, lblboundlx, lblboundly
RadioButton	rdbuttonname,rdbuttonboundux, rdbuttonbounduy, rdbuttonboundlx, rdbuttonboundly
Scale	sclbgdcolor,sclboundux, sclbounduy, sclboundlx, sclboundly, sclsetmax, sclsetmin, sclsetincr, sclsetpgincr
TextBox	textbgdcolor,textboundux, textbounduy, textboundlx, textboundly

Table 2 Attach variables for Widgets

3.6 Natural Language description of an SES

As the detailed syntax for the Natural Language (NL) described by Zeigler et al can be found in [8], a comprehensive description here is considered unnecessary. However, in the next few pages an attempt will be made to introduce the key processes used to convert the visual SWT SES tree (enhanced with variables) to the NL. It is deemed that the best way to do this is by way of example.

Let us recall that the Root Entity “shell” can be specialized in “design” either into a “composite” and a “tabfolder”.

Hence in line 1 of figure 9, “shell”, “composite”, “tabFolder” and “design” are the words of our choice. The words “A”, “can be”, “or” and “in” are key and compulsory in the NL description which would essentially explain the specialization format. The “!” marks the end of any sentence. Line 2 shows the inclusion of the aforementioned variables name, height and width of shell. Again the underlined are the words of our choice and the rest are compulsory keywords. Line 3 says that a “name” is of the value type “string”. This would restrict the ultimate user to enter a name in only string format. Note that an intentional space is kept after “values” and the “!” in line 3. This is to allow the user to enter any name of the string format. Also a variable can be of keyword value type integer (“int”) which would be the obvious choice for a height and a width. Line 4 and 5 places restrictions on the allowable values which can be specified by the user. Hence the ultimate user can only enter integers from 300 to 400 as a valid value. Note that at the pruning stage, these values can be checked for their validity by the framework (to be discussed) upon user’s choice.

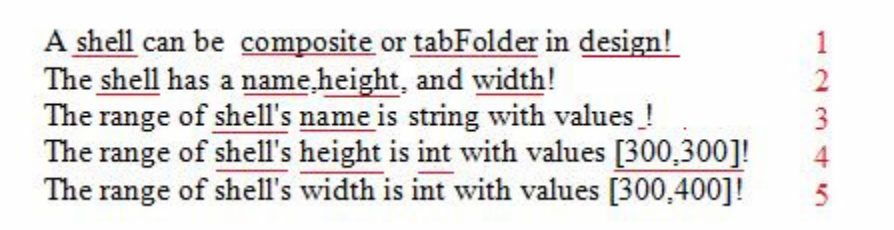

Figure 9 Shell described in NL

Recalling that in SWT, a tabfolder is a container that can include multiple tabitems, the below figure 10 explains by example how this could be achieved. Again the underlined are the word of our choice.

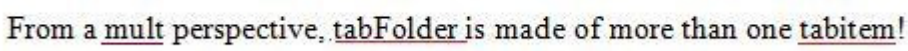

Figure 10 Tabfolder Description in NL

Out of the many widgets that can be selected to form a GUI, recall that seven were chosen as an experimental case for this study. Hence the entity "components" can be decomposed into seven further "parts" from a component widget view. This process of breaking down components into widgets namely buttons, radiobuttons, checkbuttons, labels, scales, comboboxes, and textboxes is described by the below example in figure 10. Again the underlined are the words of our choice while the "From", "perspective", " is made of" and "and" described the aspect of the entity components.

From compwidgets perspective, components is made of
buttons,radiobuttons,checkbuttons, labels, scales, comboboxes,and textboxes !

Figure 11 Decomposition of "Components" into widgets

The below text block in figure 12 is a part of the NL translation of the SWT SES which recaps all of the concepts mentioned above (Specializations, aspects, multiple aspects and variables).

The complete Natural Language translation for the above mentioned SES can be found in Appendix 1.

The layout can be absolute,fill, or form in layouttype!

From absolutecomp perspective, absolute is made of components!

From fillcomp perspective, fill is made of components!

From compwidgets perspective, components is made of
buttons,radiobuttons,checkbuttons, labels, scales, comboboxes,and textboxes !

From a mult perspective, textboxes are made of more than one textbox!

The textbox has a textbgdcolor, textboundux,textbounduy,textboundlx, and textboundly!

The range of textbox's textbgdcolor is string with values white,yellow, and green!
The range of textbox's textboundux is int with values [0,392]!
The range of textbox's textbounduy is int with values [0,366]!
The range of textbox's textboundlx is int with values [0,392]!
The range of textbox's textboundly is int with values [20,350]!

Figure 12 Segment of SWT SES in NL

3.7 Natural Language to XML Process

Once the Natural Language description for the SWT SES is in hand, the tool offered by Zeigler et al can be put into good use to convert the SES into an XML version. A practical implementation of the tool can be found at [14]. The figure 12 shows the "Virtual Working Table" Web User Interface which will be used to first generate our SES in XML and then a PES which will be the ultimate file the user will need to modify to generate a GUI. This process will be explained in chapters 4 and 5.

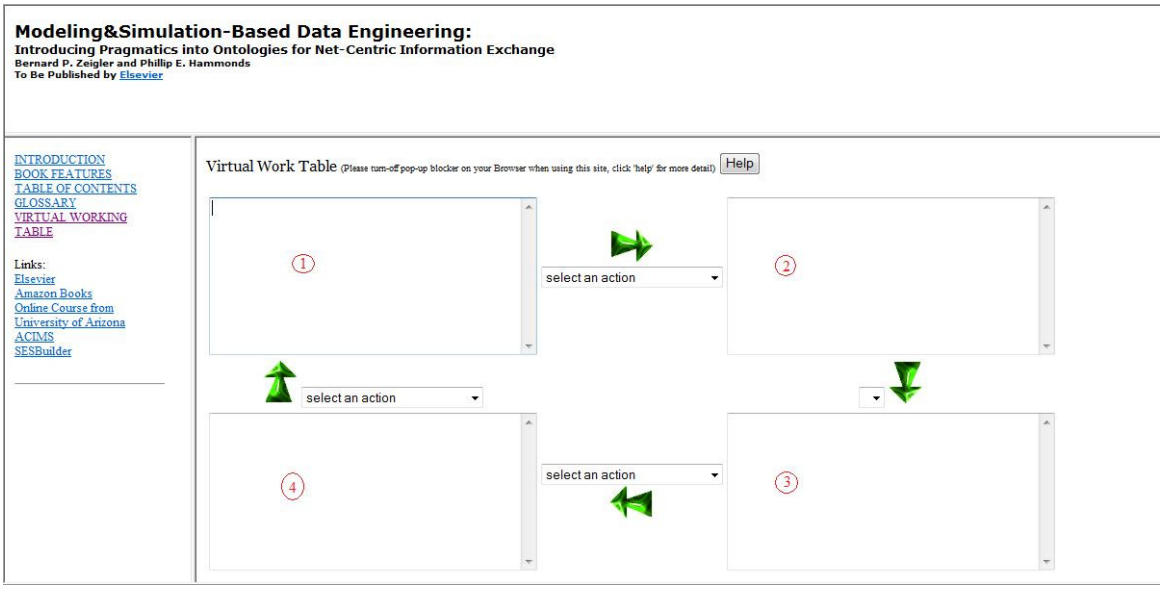

Figure 13 Vitual Work Table Web Interface

The conversion of the SWT SES in NL to a one which could be "Pruned" by a user, involves a 3 step process.

Step 1 :

The SWT SES in its Natural Language format (found in Appendix 1) will be copied into the text area of panel 1 in the above figure. Afterwards from the drop down list between panel 1 and 2, “NL to SESinXML” needs to be selected and then the “green arrow” clicked. Figure 14 shows this process with the SES in XML auto generated in Panel 2.

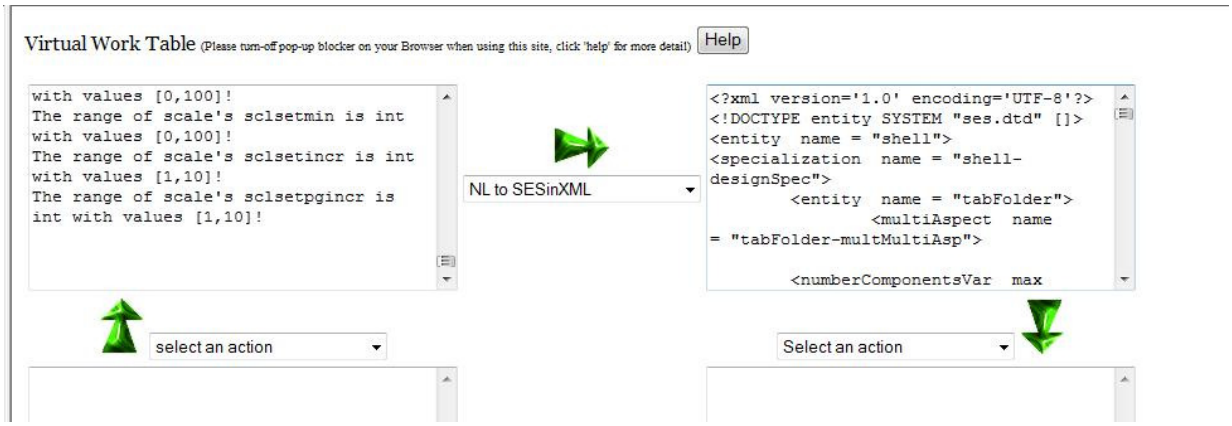

Figure 14 Virtual Work Table panels 1 & 2

Step 2:

Once the XML version of the SES is generated in Panel 2, “SESinXML to DTD” needs to be selected from the dropdown list and the “green arrow” between panel 2&3 clicked to generate a DTD of the SES in XML. Figure 15 illustrates this process. A Document Type Definition (DTD) defines the legal building blocks of an XML document. It defines the document structure with a list of legal elements and attributes.

As the DTD conversion is only an additional step to accommodate for the usage flow of the tool, further technical explanations will be considered unnecessary.

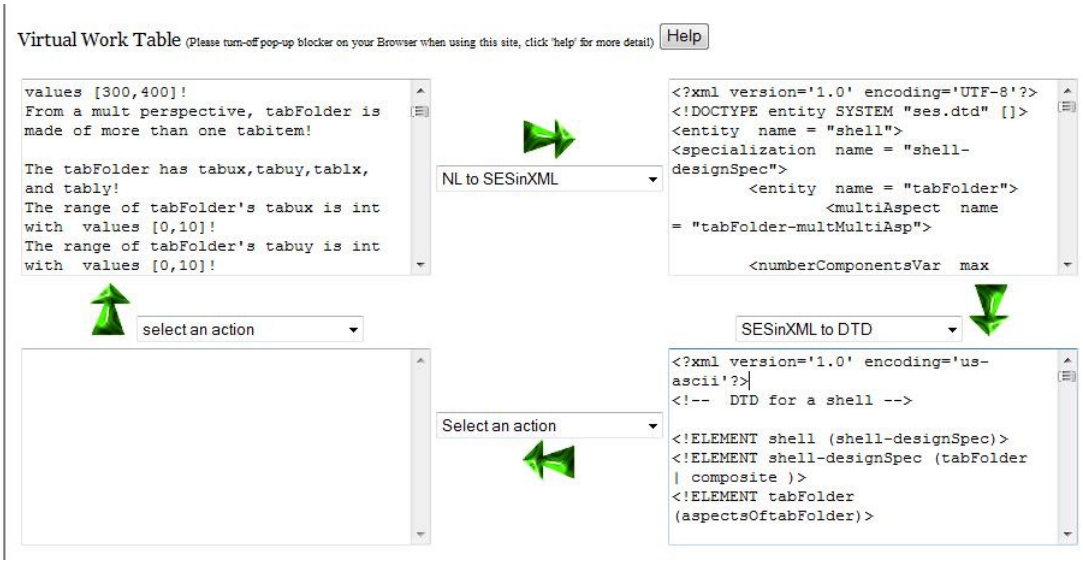

Figure 15 Virtual Work Table panes 1 - 3

Step 3:

After the DTD is generated in panel 3, "DTD to PESinXML" needs to be selected and the "green arrow" clicked between panel 3&4. The panel 4 will show a candidate SES in XML format for a Pruned Entity Structure (PES). Figure 16 illustrates this process. As Pruning the Entity Structures remain a very important step prior to a GUI can be generated from it, a detailed introduction will be given in the next chapter.

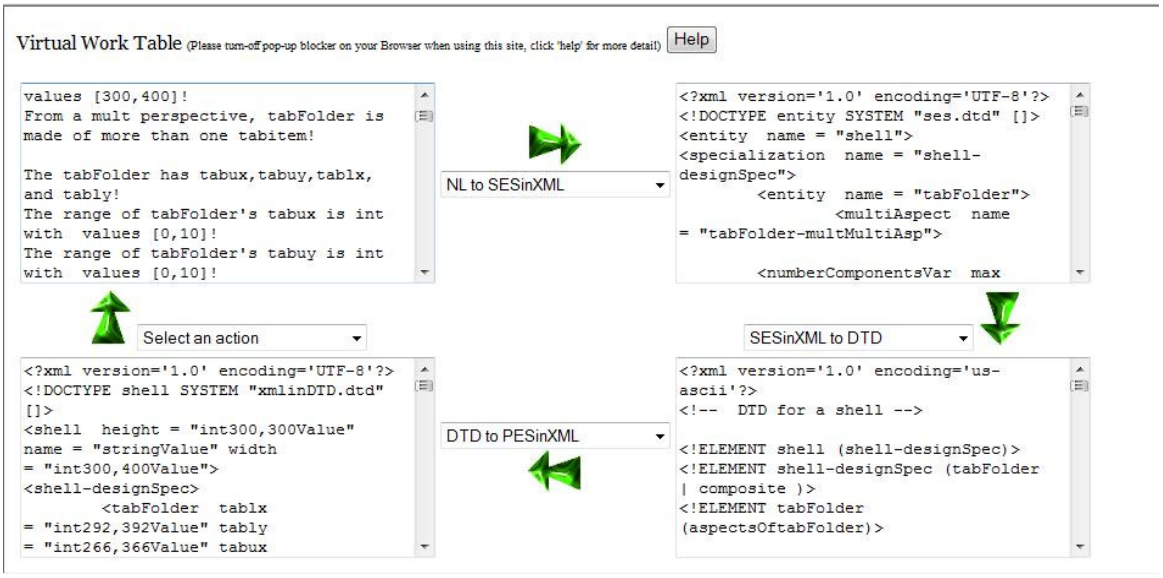

Figure 16 Virtual Work Table panes 1 -4

CHAPTER 4 USER MODIFICATIONS: PRUNING AND EDITING

This chapter will explain in detail the process of Pruning and Knowledge Representation of the SWT SES which will be a required task before a GUI can be generated from it. As the pruning process will Solely determine the format and layout of the GUI (to be Generated later in the next chapter) , it will be of utmost importance. Efforts will be made to introduce the inherent steps by way of example. As explained in the previous chapter, the SES generated in XML in step 3 (found in Appendix 2) will be the candidate for modification by the user . It will be made into a PES by the user to meet his/her specific requirements.

As we saw in Chapter 2, the Pruning of an SES can create a family of PESs. Hence as a start we will consider the specific PES shown in figure 16 below and discuss how it could be made a valid candidate for data representation and extraction. The path marked in RED is the intended PES to be derived from the original SES.

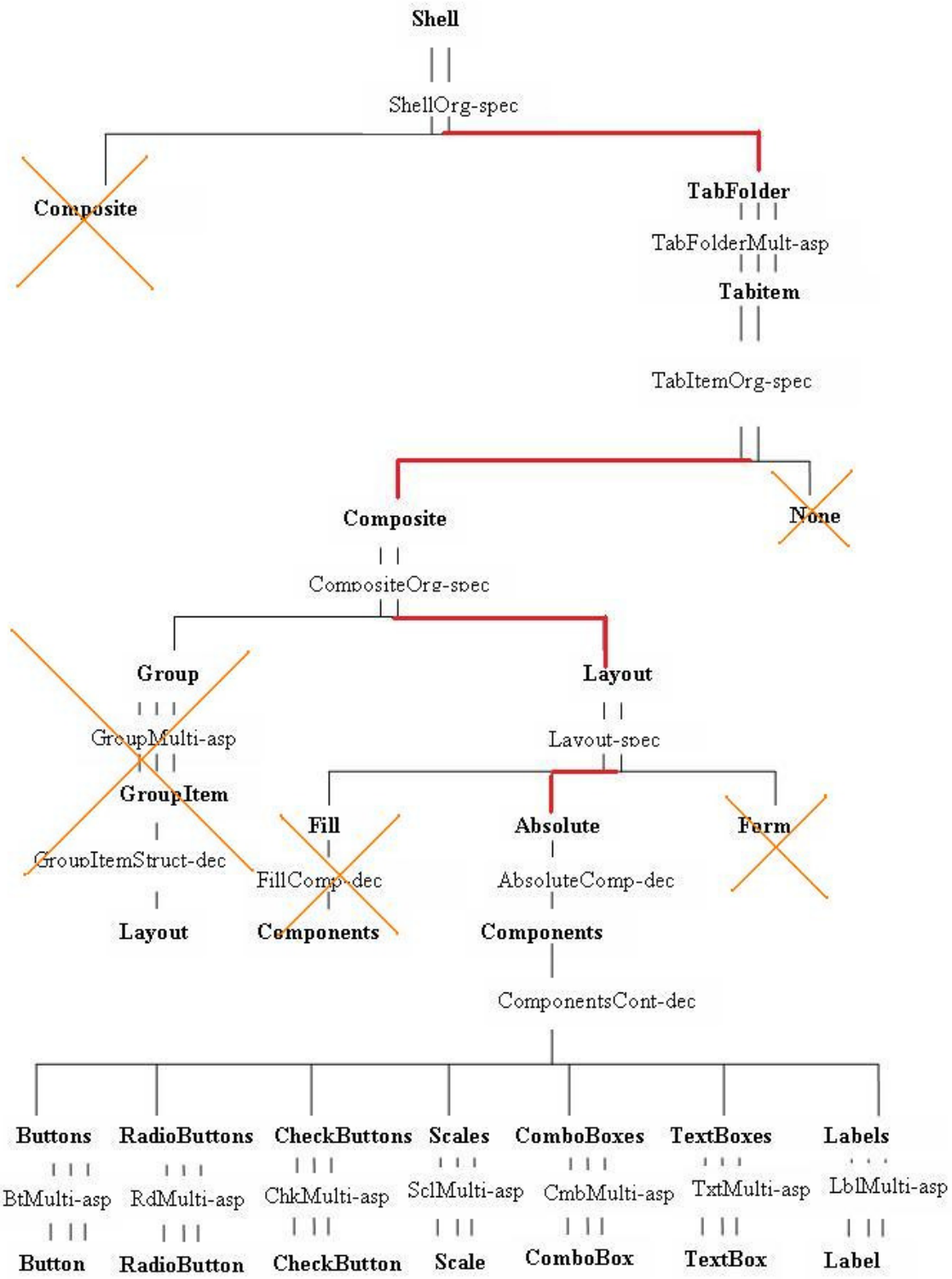

Figure 17 One PES for SWT SES

A reverse engineering approach will be used to best explain the process. Let us consider the Simple GUI shown in figure 18 below with one tabitem.

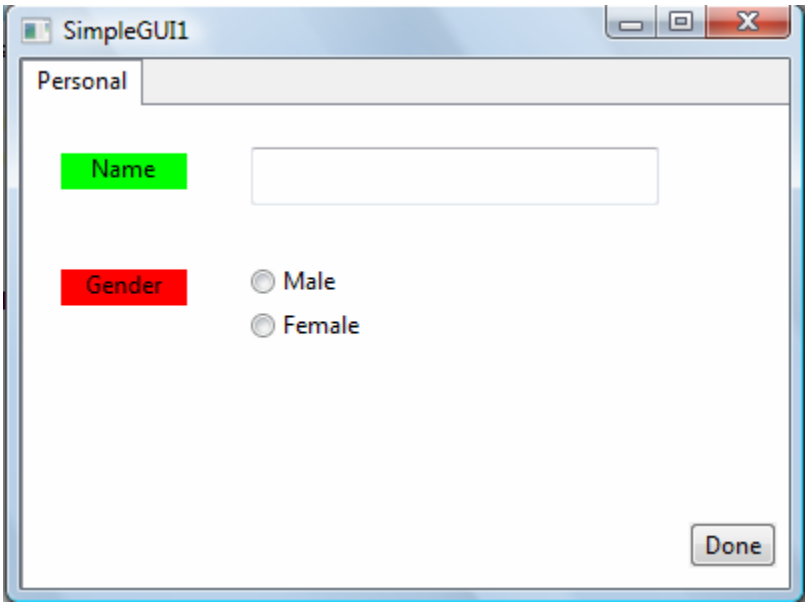

Figure 18 SimpleGUI1

The process required by a User wishing to generate the above GUI can be broken into a two steps.

1) Prune the SES in XML shown in Appendix 2 to one that reflect the figure 16 . This would result in a much smaller PES (in XML) to edit in the next step

2) Once the PES is derived, the user can embed relevant information into its already existing variables to make it viable for data extraction

The two step process is discussed in details below.

4.1 Pruning the SWT SES

As mentioned before we first need to prune the SES shown in Appendix 2 to derive the specific PES shown in figure 17. A special note should be made that the line "<!DOCTYPE shell SYSTEM "xmlinDTD.dtd" []>", which would usually appear as the second in the SES generated in Panel 4 (refer to Step 3 in previous chapter) was intentionally deleted in the one shown in Appendix 2 . This is due to the fact that the tool which generates the Automatic GUIs (discussed in the next chapter) does not require it.

Though the aforementioned SES seems to be rather big, it should be noticed that it carries lot of extra redundant information in this particular case. The crossed out entities needs to be first removed from the SES to make it into the PES of our choice.

Though the SES could be edited within the Panel 4 of figure 13 itself, it might be more convenient to use an XML editor to do the tasks at hand. An editor would let you directly modify or delete an entity along with all its attributes and elements. This would prove to expedite the process than deleting it manually by hand. Also an editor would let you directly fill in values for variables. The below screenshot shown in figure 19 is

extracted from Altova XMLSPY User Interface [15]. It shows the SES (ready for pruning) in Appendix 2 opened and ready for editing. The big oval in yellow covering the blue colored area in the figure is one to be deleted. It is the “Composite” entity which needs to be deleted as shown in figure 16 . After the pruning process, the PES become significantly compact in size as a quick glance at the PES in Appendix 2 would show.

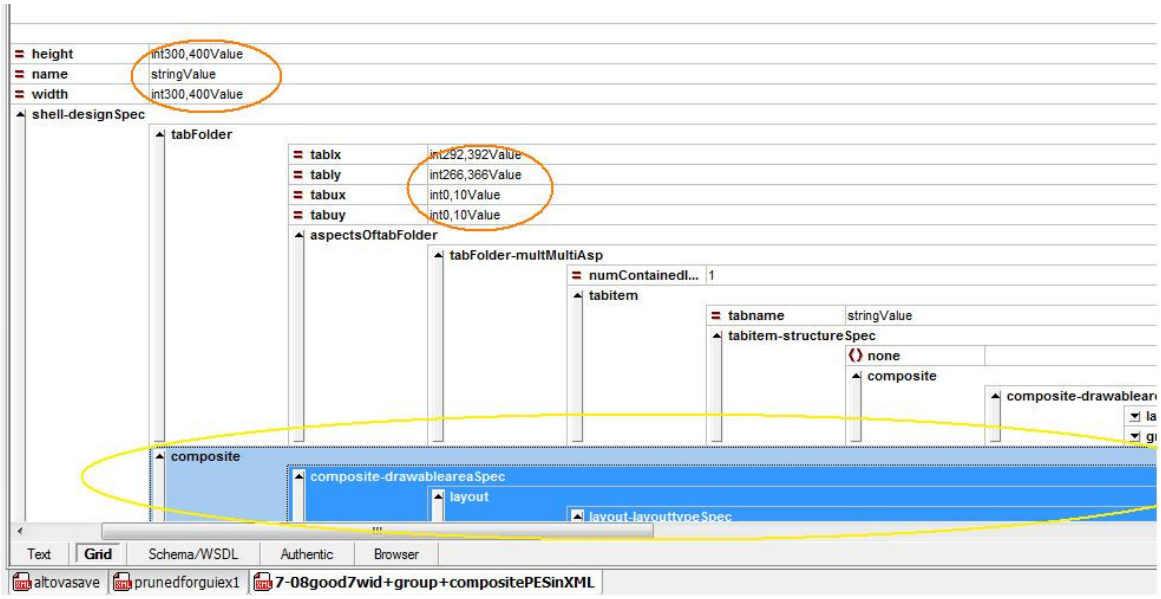

Figure 19 Altova XMLSPY

4.2 Entering data into variables :

Before the User can modify the variable data values of the PES, he/she needs to understand the basics of how the GUI in figure 18 correlates and maps to the existing PES.

Let us try to describe the GUI in hand. It is a GUI with the title “SimpleGUI1” displayed in a single Tab with the name “Personal”. Within the Tab the following widget components are placed strategically. Namely, a Textbox, 2 Labels, 2 Radiobuttons and a Button. Respectively, the names of the labels, radiobuttons and the button are “Name”, “Gender”, “Male”, “Female” and “Done”. The background colors of the two labels are green and red.

With this information is it easy for the User to comprehend that the Shell name is ”SimpleGU1”, the Tabfolder should contains 1 Tabitem with name “Personal”, and the components widgets should contain only the 3 mentioned above.

As mentioned before the multi-asp feature allows the inclusion of 0 (zero) items. Hence this feature could be put into good use when the user does not want to include any other widgets.

For example in this case, the “numContainedIncheckbuttons” variable should be set to 0 and anything between the <checkbutton> and </checkbutton> tags should removed. Looking at the code in Appendix 3, would clarify this.

Once the basics have been understood values for variables could be entered into the PES in XML. The red ovals in the above figure show example areas where direct values could be keyed in for values. For example the top oval shows the area where values should be entered for Shell’s height, width and name. Similarly the user can key in the values for rest of the variables of the PES. The Shell height would be 300 and the width would be 400 with the name modified to “SimpleGUI1”.

As the “Absolute Layout” was the pruned choice for this particular example, a question might arise how the user would know the coordinate values to fill in. Let us revisit the SES in NL , shown in Appendix 1, and observe the below block of code extracted from it.

```
The range of tabFolder's tabux is int with  values [0,10]!
The range of tabFolder's tabuy is int with  values [0,10]!
The range of tabFolder's tablx is int with  values [292,392]!
The range of tabFolder's tably is int with  values [266,300]!
```

When the NL for the SES was initially defined, calculated restrictions were imposed on the range of values the User can enter.

Note the line 5 of the generated SES in XML of Appendix 2,

“<tabFolder tablx="int292,392Value" tably="int266,366Value" tabux="int0,10Value" tabuy="int0,10Value">”.

The indicated ranges show up within the area whether the value has to be keyed in. For example tablx= “int292,392Value” suggests a value between those ranges. So the User can simply chose a concrete value such as tablx= “392” or lblbgdcolor= “red” for the label color.

4.3 Validating the PES

It would be worthwhile to mention that the tool introduced in the previous chapter has a built in feature which checks the validity of the values entered in the PES with the original ranges defined by the SES in NL. This is an optional choice for the user but could prove to be useful. After the specific values has been entered into the variables, and the PES finalized, the user can select the "Validate PES" from the dropdown list between panel 4 & 1 and click the green arrow(refer to figure 13) . The below figure illustrates this process for the modified PES SimpleGUI1.xml in Appendix 3

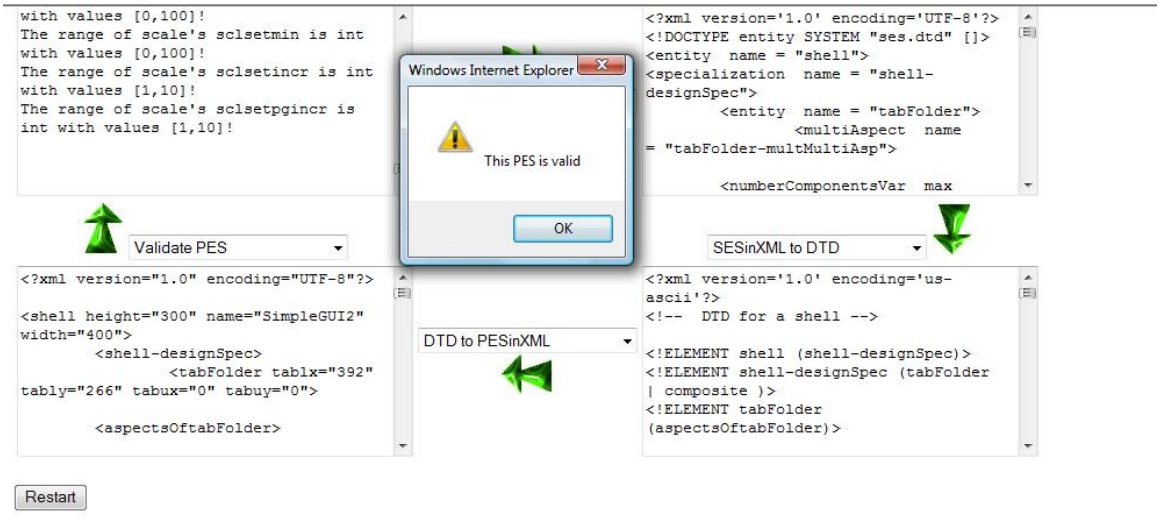

Figure 20 Validate PES in Virtual Work Table

Once the PES in XML is finalized, it should be saved for later use. Though one might feel that even with these features to make it more user friendly, a PES with Absolute Layout can prove to be too cumbersome for the User to edit and modify, it should be noted that it has its own benefits including total user control of widget placements. As we will see later in Chap 6 , using a PES with the "Fill" Layout can drastically reduce this overhead time to figure and fill out the location variables, but likewise it will have it's own disadvantages.

Also an argument can be made that there is a time overhead for this modification process for what appears to be conservative gain. Though it is agreed that the initial pruning and modification process can take a while, the benefits of this method does not reside in simple GUIs as the one discussed here. For example and argument sake, if we wish to have 3 tabs with each containing the same layout configuration within the tab as the one shown above, but only the tabname and widget names change, the modification process would be nothing simpler.

All it takes is to

a) modify the "numContainedIntabFolder" value to 3

b) make copies of every thing within the "tabitem" tags (line 7-81 in Appendix 3) two more times in succession

c) change the variables containing names to reflect the newer ones.

An example SimpleGUI2.xml created by this process can be found in Appendix 4.

Though it seems appropriate that a few more examples using different SWT PES is discussed at this juncture, the author deems is necessary to go into the GUI generation process and then revisit more examples in the following chapters.

CHAPTER 5 AUTO GENERATION OF GUIs

From the User perspective, once a Pruned Entity Structure reflecting a specific GUI is defined in XML, the GUI generation process proves to be a comparatively trivial task. To generate GUIs, the user only needs to feed the XML file into a tool named "SWTParser" which was written as a part of this research effort.

5.1 SWTParser

The SWTParser is a tool written in java by the author which takes a predefined SWT PES in XML format as input, and writes out an auto generated .java file to a specified path. Figure 21 illustrates this process. The written java file is a fully functional GUI based on SWT libraries which only needs to be compiled and executed by the user. The execution process and its features will be discussed later in the chapter.

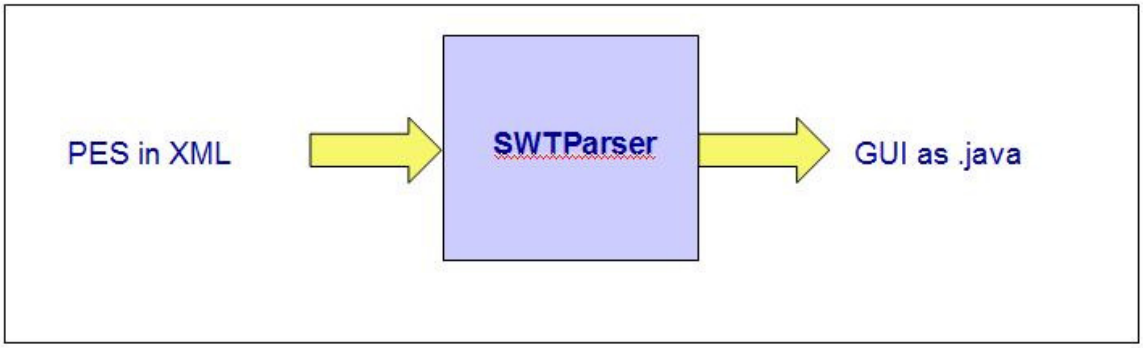

Figure 21 Process flow : PES in XML to GUI

5.2 The testing and execution environment

The tool was written and tested on a Pentium III machine running JDK 1.5.0_11 with 256MB of memory. However for processing of large GUIs it is recommended that a faster processor with at least 1GB memory is used for speedy code generation. As java code is portable and can be compiled in cross platforms, there should be no issues in running this tool on a Linux or MacOSX based machine.

For demonstration, execution and testing purposes, henceforth we will be using the Eclipse SDK Version: 3.2.2.r322_v20070104 IDE for Windows which could be download as freeware [16]. For Linux and MacOSX enthusiasts there are compatible Eclipse IDEs also downloadable at the above site. It will be presumed that JDK and SWT libraries are installed and imported into Eclipse and ready for execution. The below figure shows the basic Eclipse IDE with SWTparser .java open.

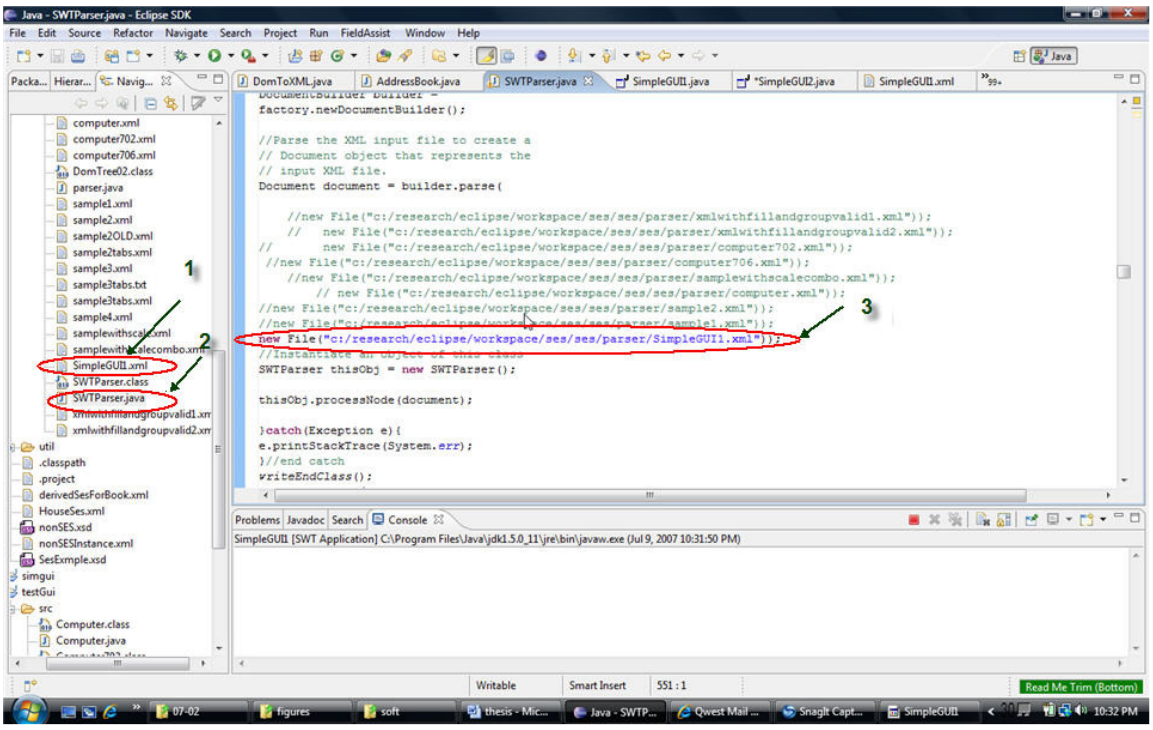

Figure 22 Eclipse IDE with SWTParser open

5.3 GUI code generation process

From a User perspective the auto generation of a GUI is a simple 4 step process

Step 1 :

Copy the Pruned SES in XML (SimpleGUI1.xml in this example, which was created in the previous chapter and can be found in Appendix 3) to some folder. In this case it is placed in the same folder as the SWTParser.java for convenience. Shown with 1 &2 in figure 21.

Step 2 :

Include path of the PESinXML (ie SimpleGUI1.xml) as below within the main function.**File("c:/research/eclipse/workspace/ses/ses/parser/SimpleGUI1.xml"));"**

Note that only one file can be read at a time.

Step 3 :

Set the global variable "**outdir"** located in the SWTParser class to a convenient path of the User's choice. This is output path is where the user will find the auto generated java file for the GUI. In this case it was set to

outdir**="c:/research/eclipse/workspace/testGui/src";**

as shown by 3 in figure 22.

Step 4 :

Once the GUI java code is generated (SimpleGUI1.java) it can be compiled and executed to produce the actual GUI. The below figure shows the output of the SimpleGUI1.java running in the Eclipse IDE foreground.

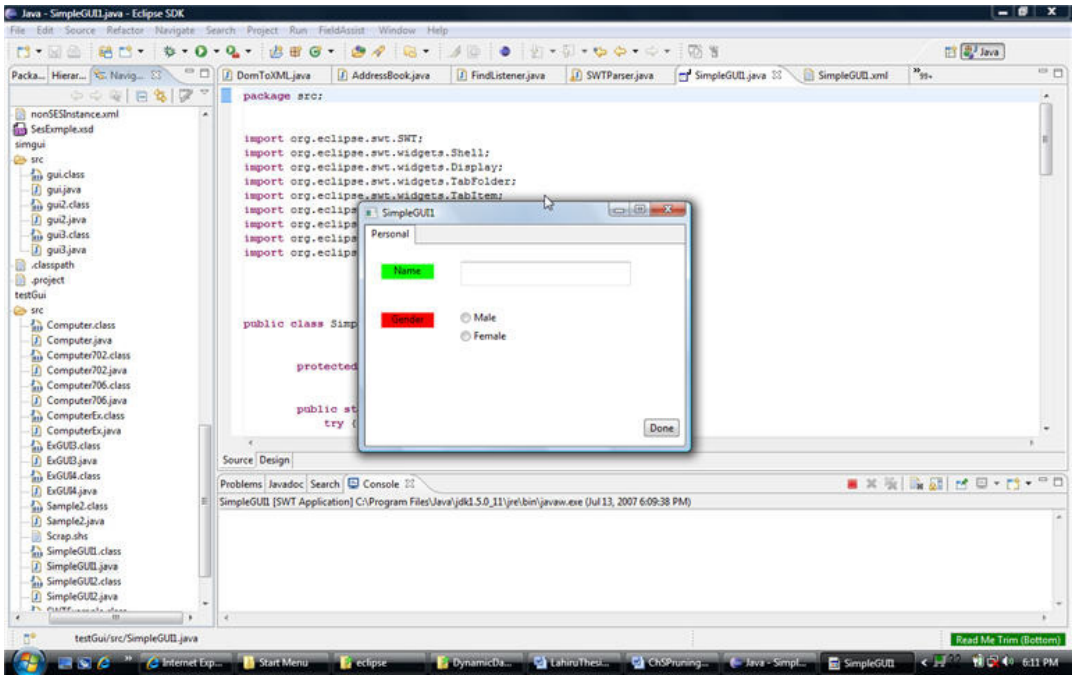

Figure 23 SimpleGUI1 executed

Following the same four steps discussed above, the SimpleGUI2.xml which was discussed in the previous chapter can be processed through SWTParser to generate the SimpleGUI2.java. Below figure contains the output of the exercise.

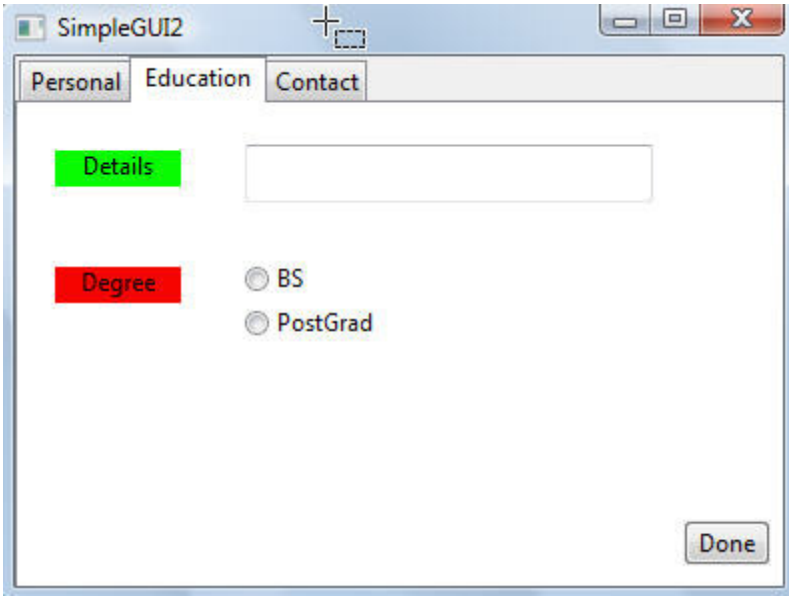

Figure 24 SimpleGUI2

When referring to the SimpleGUI2.xml one can notice that both the “Education” and “Contact” tabs have a label named “Details” within them. It should be noted that the tool has built in intelligence to generate proper output even in such cases. However the user needs to be careful not to name 2 widgets of the same category (eg 2 labels) with identical names within a single tab.

On the same topic, it should also be noted that if a user inputs a shell size , for example as 300 height and 400 width, and enters bad coordinates for tabfolder location variables (ie tabux, tabuy etc) , the tool has built in intelligence to recognize the bad coordinates and re-adjust the tabfolder to fit well within the shell. (User entering bad values can be reduced through the mechanism discussed in chapter 4) The author point this out only as an attempt to show that the tool can be further extended to add such features when necessary

.

The ExGUI3.xml (it's not a simple GUI anymore) code attached in Appendix 5 will clarify the above feature on auto resizing. Also, note that the "Personal" tab as seen in the below figure has been expanded to demonstrate the use of other widgets. The below ExGUI3 was generated and executed using the same methods discussed above.

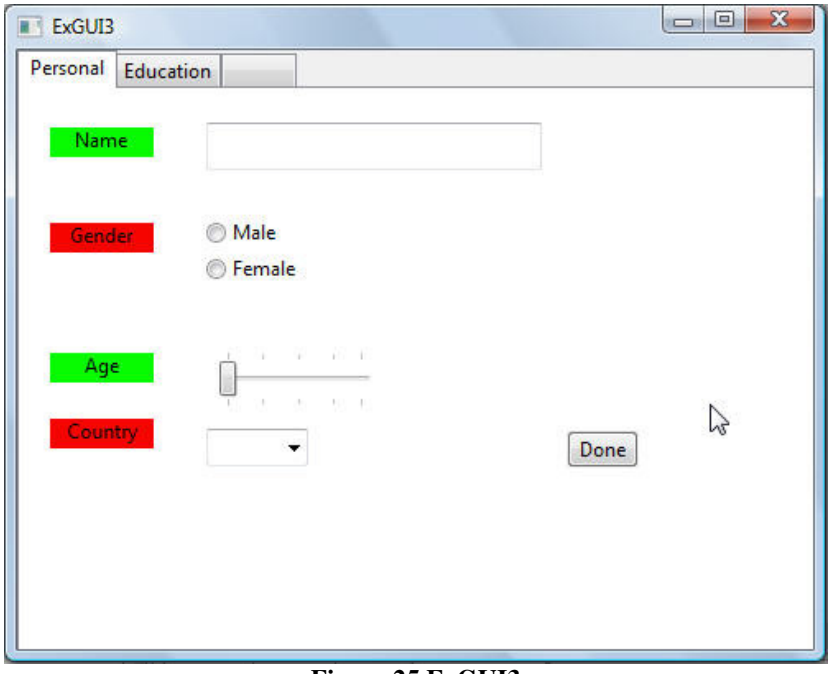

Figure 25 ExGUI3

CHAPTER 6 SES TO GUIs

In the previous chapters, we discussed how we can include GUI information into a PES in XML format and auto generate the desired GUI from it. As we initially took a reverse engineering approach and worked forwards, it would probably not be apparent whether a specific instance of given SES (maybe a physical hierarchical system) could be mapped into the existing framework and then a GUI could be generated from it. This chapter will try to introduce a simple methodology to perform the above function given an SES.

Let us consider a slight modified version of the SES in figure 1 shown in the below figure.

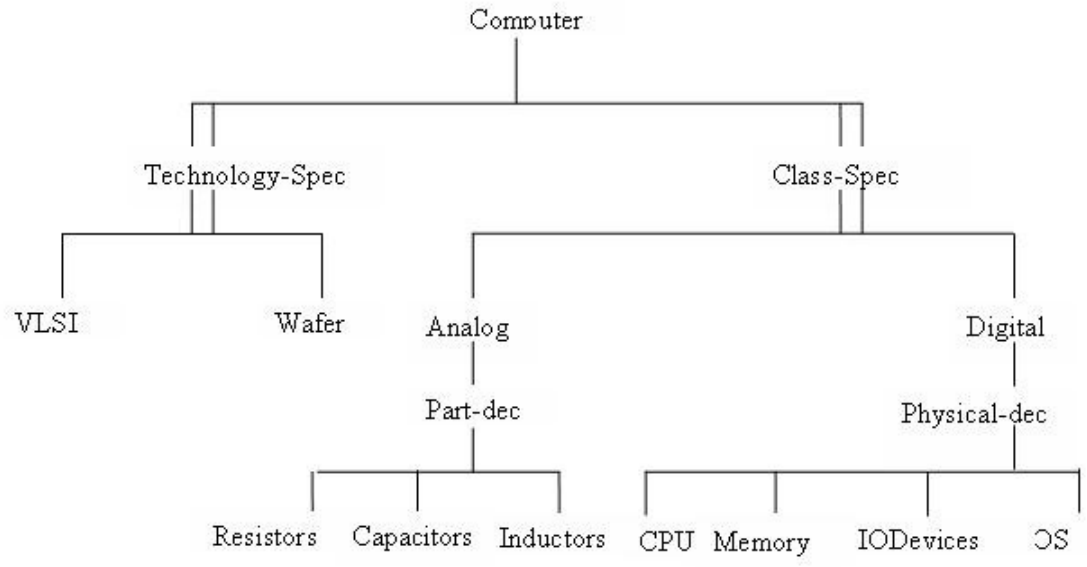

Figure 26 Variant of Computer SES

Is it possible for us to transform this SES into a GUI ? A User might say, "Okay, I see that the root entity will be the Computer, but how do I go about mapping rest of the Entities to a SWT PES to generate a GUI?" As a formal approach to this process is not defined yet, the above question will prove to be valid. It should noted that before information from the above SES is translated to a SWT PES by a user, there is a intermediate step which is still undefined. Hence, the next sub section will try to introduce a methodology to take the above SES and relate it with the existing SWT SES.

6.1 Mapping a SES to the SWT SES

The figure below presents a way a user can relate the Computer SES to the SWT one. The Computer is considered the root entity and hence the name "Computer" can be embedded into the shell name variable. A specialization name (Technology & Class) is thought of as a tabitem name. The corresponding entities of the specializations (VLSI, Wafer, Analog & Digital) should be contained in a group as *groupitems*. The groupitem is used to organize an area (within a tab in this case) to sub areas. Each of these sub areas can be given a name to visually recognize them as we will see later. A "composite" also defines an area in SWT, but they do not have visible name attached to them. The user should be aware that in SWT, there is only a concept called group and no physical element called a groupitem as in tabitem. However , for this research , the concept of a group and a groupitem was introduced at the SWT SES level. A quick glance at its description in NL(Appendix 1) and the corresponding PES of figure 26 shown in Appendix 6 would clarify this further. The SWTParser will look at how many groupitems are defined in a group and translate them into valid SWT code. This process is hidden from the user as he/she doesn't need to be aware of the SWT code generation semantics. The User will only need to be aware of the conversion process of the given SES to a SWT PES in XML.

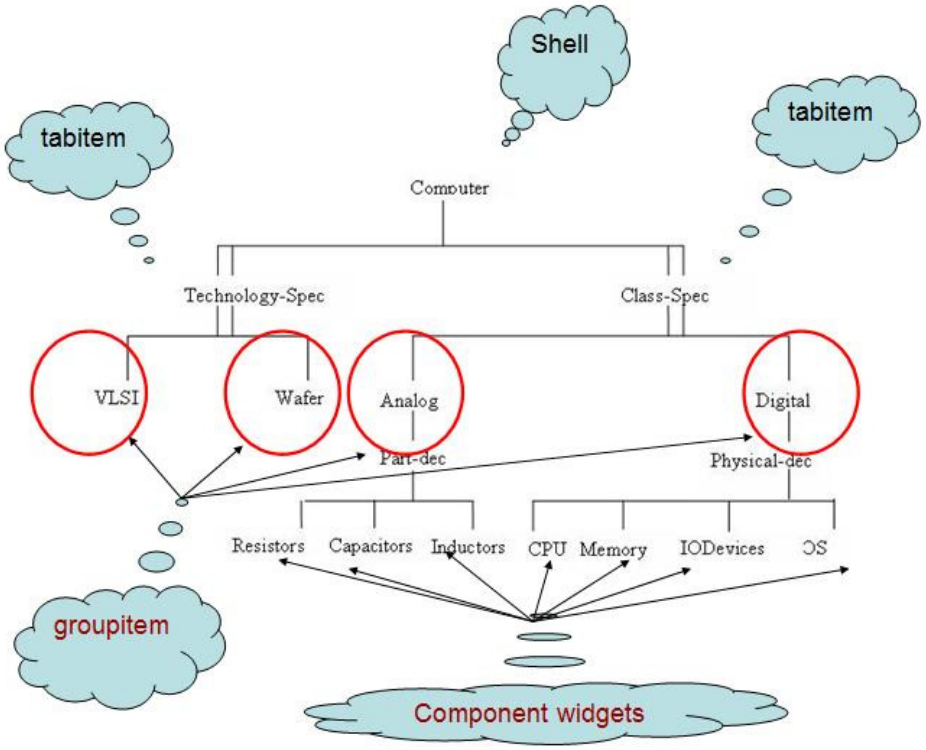

Figure 27 PES of SWT SES with Group

When a specialized entity is further decomposed as in this case, the leaf level entities (Resistors, Inductors, Capacitors, CPU, Memory, IODevices and OS) will be component widgets. Of course one could ask what happens if an aspect is further decomposed ? As mentioned in the introduction the objective of this research was not to cover every level of possibility. Hence, further research in this area will be needed to expand on the concepts introduced here to cover issues as such.

Once the User has a clear method of formalism in mind, the Computer SES can be transformed into a SWT PES which could be read and a GUI generated from it. The figure below shows the path (in Red) the SWT SES will have to take to create the PES for this example.

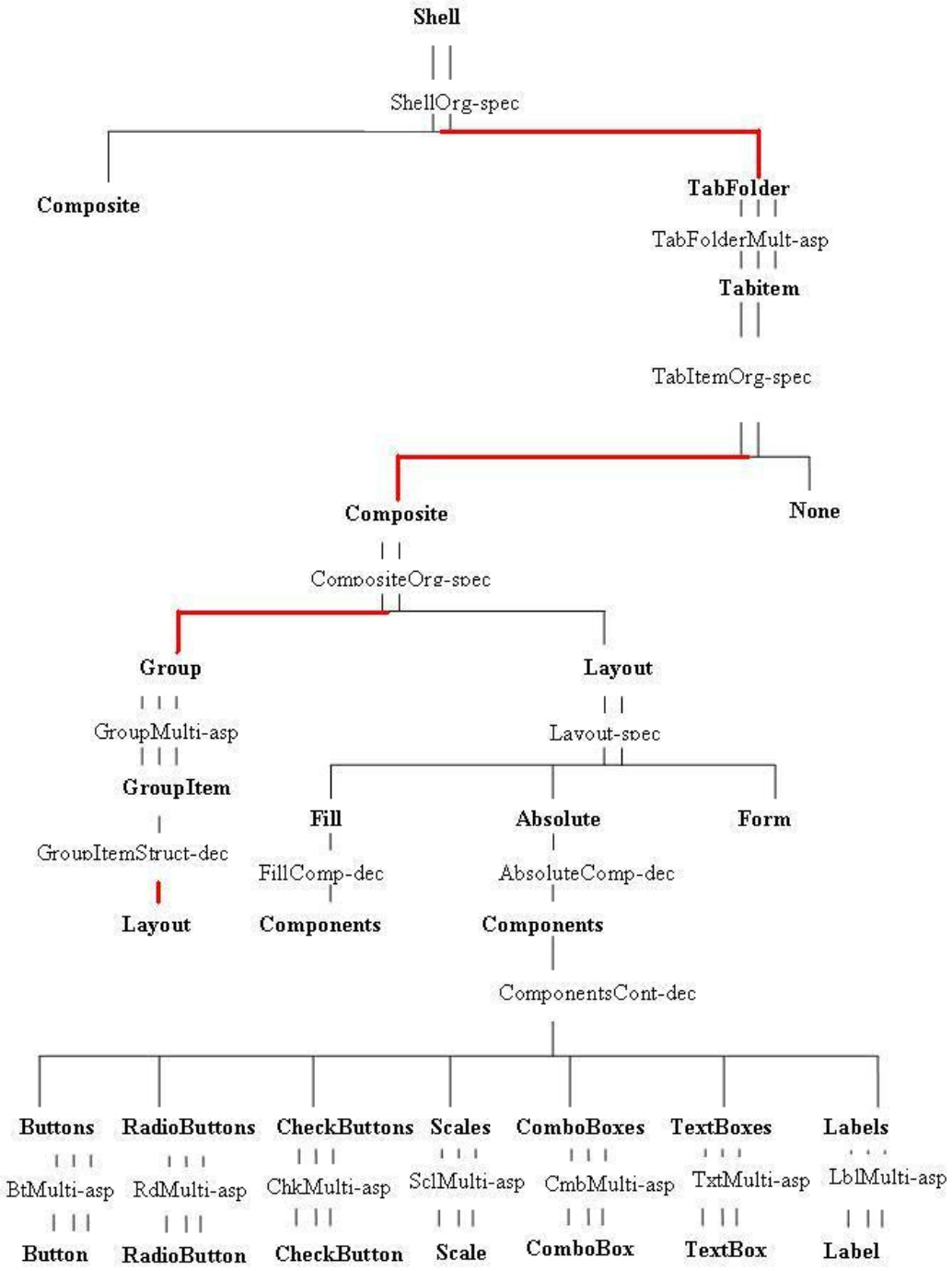

Figure 28 PES of SWT SES with Group

6.2 Groups & Fill layouts

To understand more thoroughly the concepts of “group” , “groupitems” and “fill” layout , let us consider the segments of XML extracted from the PES , ExGUI4.xml from Appendix 6.

```
7   <tabitem tabname="Technology">
8     <tabitem-structureSpec>
9       <composite>
10        <composite-drawableareaSpec>
11          <group>
12            <aspectsOfgroup>
13              <group-multMultiAsp numContainedIngroup="2">
14                <groupitem groupname="VLSI">
15                  <aspectsOfgroupitem>
16                    <groupitem-groupitemstrucDec>
17                      <layout>
18                        <layout-layouttypeSpec>
19                          <fill>
```

Figure 29 ExGUI4.xml code fragment

In the above figure, note that the *composite-drawableareSpec* has been pruned to include “group”. In SWT a group is different from a composite container as it is allowed to have a visible name and a border. Note , in all previous examples the pruned path was composite -> layout. As there are two candidate groups (ie VLSI and Wafer) under “Technology” tabitem, the multAsp number will be 2 and two groupitems needs to be created with the above names. Attention should be drawn to the uniformity axiom of SES as the *layout* under groupitem is the very same as the *layout* entity we have seen before. However, in previous examples we were using “absolute” as layouttypeSpec but in this case we are using “fill”. As the “VLSI” and “Wafer” groups are not further

decomposed there will be no component widgets within them. One can note this fact by observing in Appendix 6, that all component widgets have num 0.

The main advantage of using fill layout is that the user will be relieved of the burden of figuring out the coordinate locations for each component as in absolute layout. This drastically cuts down the PES generation time. Hence the reason for specifying <tabFolder tablx="0" tably="0" tabux="0" tabuy="0"> in line 5 of the xml code. However the downside is that the user will lose control of exact widget placements. When using the fill layout, any component which is added to the container (ie shell, composite or group) will be equally spaced as shown in the below figure. The figure 30 shows the output generated using SWTParser for ExGUI4.xml using the same method discussed in the last chapter. Note that the SWTParser automatically adds components to a container in fill layout horizontally by default. It also sets the shell and composite containers to use fill layout automatically if the User chooses fill as a group layout. Note that a bug in SWT will not allow one to specify a shell as “absolute” and add widgets to its sub containers using fill layout. Actually, though code can be written as such and will compile, when it is executed, the widgets will not appear as expected! These corrected actions are hidden from the user as he/she does not need to be aware of it, but it will be visible in the auto generated code. A small extension would be needed to the original SWT SES in NL, if one wishes to add components vertically.

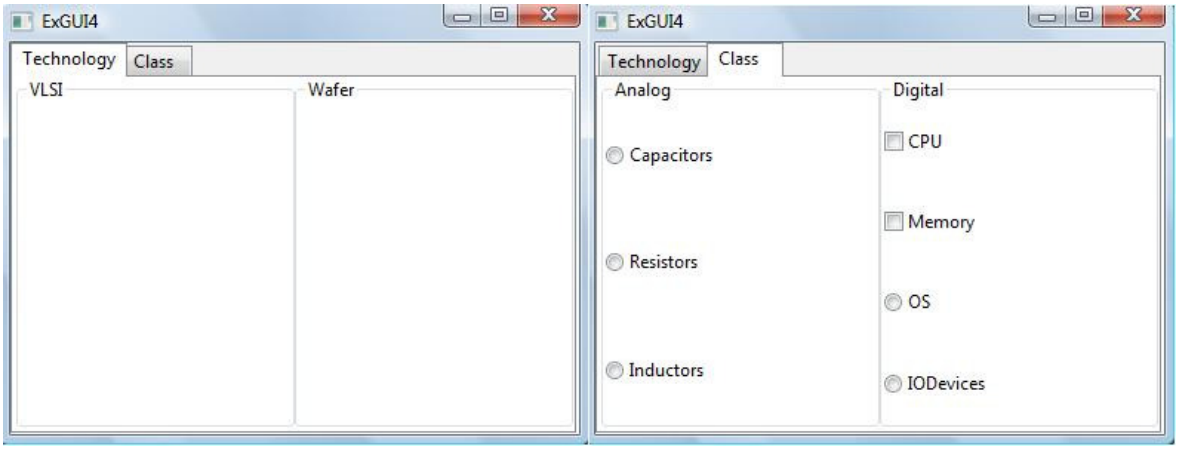

Figure 30 EXGUI4

It should also be noted that though the above GUI will appear visually correct to a regular user's eye, to a user who is proficient in the semantics of SES will find it rather irregular. The reason is due to the fact that radio buttons and check buttons were chosen as widget options for the decomposed entities at the leaf level. The decomposed entities practically should not be items of choice. A radio or a check button is a widget which can be chosen active or otherwise. Hence a discrepancy would arise. The below figure depicted with textboxes containing the component names would be a more appropriate choice given the restriction of the seven widgets to select from.

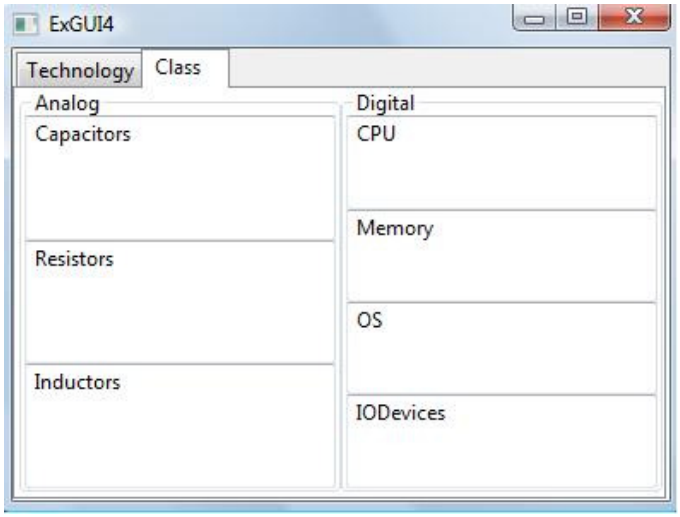

Figure 31 A semantically correct version of EXGUI4

6.3 Further reducing pruning and modification time ?

As we noted before, when using the fill layout, the User conveniently does not have to figure out the location coordinates of the components. Hence the reason tabux, tabuy etc was set to 0. If the location coordinates are unimportant, why does the User have to fill in 0s? This takes way some of the time saving benefits. Can a User, for example, leave the original tags as it were without modifications? For example can the user leave a radio button description line as

<rdbutton rdbuttonboundlx="int0,382Value" rdbuttonboundly=" int20,350Value" rdbuttonboundux="int0,392Value”rdbuttonbounduy="int0,366Value" rdbuttonname="Capacitors"> ?

The quick answer is Yes. This feature was thought about when the SWTParser tool was written. The tool ignores all the location tags when reading a PES using fill layout.

Let us consider the below code extracted from appendix 6.

```
142                                 <groupitem groupname="Analog">
143                                   <aspectsOfgroupitem>
144                                     <groupitem-groupitemstrucDec>
145                                       <layout>
146                                         <layout-layouttypeSpec>
147                                           <fill>
148                                             <aspectsOffill>
149                                               <fill-fillcompDec>
150                                                 <components>
151                                                   <aspectsOfcomponents>
152                                                     <components-compwidgetsDec>
153                                                     <radiobuttons>
154                                                         <aspectsOfradiobuttons>
155                                                           <radiobuttons-multMultiAspnumContainedInradiobuttons="3">
156                                                             <rdbutton rdbuttonboundIx="0" rdbuttonboundIy="0"
    rdbuttonboundux="0" rdbuttonbounduy="int0,366Value" rdbuttonname="Capacitors">
157                                                           </rdbutton>
158                                                             <rdbutton rdbuttonboundIx="0" rdbuttonboundIy="0"
    rdbuttonboundux="0" rdbuttonbounduy="int0,366Value" rdbuttonname="Resistors">
159                                                           </rdbutton>
160                                                           <rdbutton rdbuttonboundIx="0" rdbuttonboundIy="0"
    rdbuttonboundux="0" rdbuttonbounduy="int0,366Value" rdbuttonname="Inductors">
161                                                           </rdbutton>
162
163                                                           </radiobuttons-multMultiAsp>
164                                                       </aspectsOfradiobuttons>
165                                                   </radiobuttons>
166                                                 </components-compwidgetsDec>
```

Figure 32 Another ExGUI4.xml code fragment

Note that one of the location radio button variables was intentionally left as it were originally to display this feature and the SWTParser would still work. (One can leave out all the 4 location variables as they were initially and SWTParser would still work). Also note that as we are using three radio buttons to define “Resistors”, “Inductors” and “Capacitors” in the “Analog” group, the rest of the widgets were

completely left out intentionally within compents-compwidgetsDec. Previously when a widget was not needed we made the multAsp num to 0 as in the below code

```
<checkbuttons>
        <aspectsOfcheckbuttons>
                <checkbuttons-multMultiAsp
numContainedIncheckbuttons="0"/>
    </aspectsOfcheckbuttons>
</checkbuttons>
```

Leaving these out would further reduce the user pruning and modification time.

However there is an important fact to make note in both the above cases. When validating a PES as discussed in the subsection “Validating the PES”, leaving the original variable values as they are (ie not giving them an explicit value like 0) will cause an error. Also leaving out the rest of the widgets completely as opposed to making their multAsp number to 0 would result in “number of entities in components-compwidgetDec in SES and PES not equal”. Hence there will be a trade off with this process if one wishes to validate the PES using the “Virtual Working Table” before copying it to file.

6.4 Generating Professional GUIs using SWTParser

As mentioned in chapter 1, the initial motivation for this research was derived when the author was working on the GENETSCOPE project to study a feasible method for mapping its existing GUI to a System Entity Structure. Hence it would be interesting to extract one of its frames as shown in figure 32 and discuss the possibility of regenerating a similar GUI with the processes presented in this study.

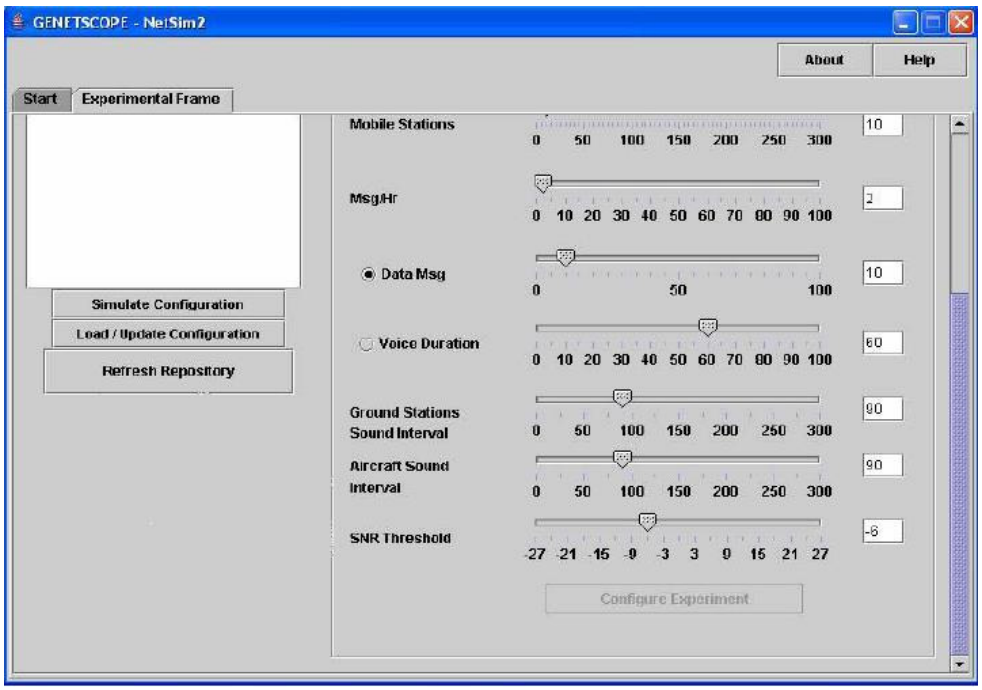

Figure 33 GENETSCOPE Experimental Frame

The relationships defined in figure 33 could be established by comparing the above figure with the knowledge gathered through this study.

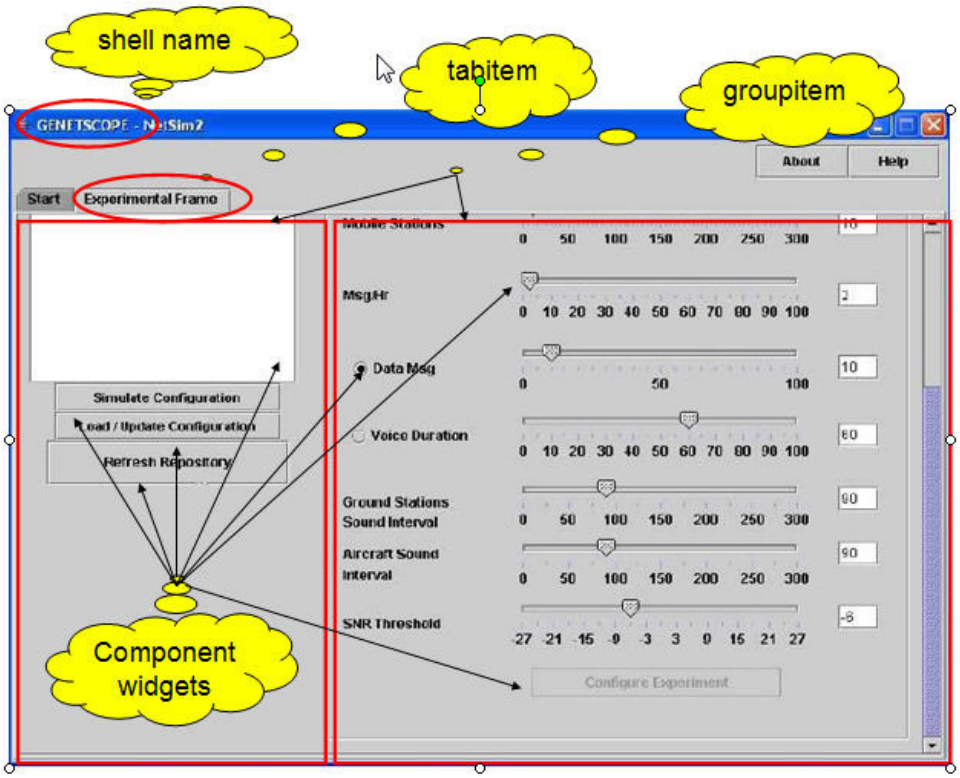

Figure 34 GENETSCOPE EF mapping to SWT SES

Note that by using a fill layout, it will reduce the times involved in producing a PES for this GUI. But as mentioned before, by using the fill layout , the user will not be able to exactly place the widgets where he/she wants and hence an absolute layout might be more suitable for this particular GUI. Using the concepts discussed in chapter 4 , 5 and 6 one should be able to generate a somewhat "similar" but not exact GUI.

However, the discussed mapping would not be able to fully include this GUI frame as one would see. Adding the missing "slider" as a component widget to the Original SWT SES might not be a complicated task. The area where the "About" and "Help" buttons are located still needs to be specified. Hence a suggestion could be made

that a PES following the “path in red” in the below figure should take care of this issue somewhat as the composite can be sub divided into 2 areas mapping to 2 group items . The first would involve the groupitem area where the “About” & “Help” buttons can be placed. But the second groupitem should be able to add a tabfolder to it to be able to specify the two tab items (Start & Experimental Frame) discussed above. As the current SWT SES doesn’t support that feature, the entity GroupItem(circled in red in figure 34) should further be extended to include a tabfolder.

Hence as one could see, this research can be further expanded to support for more complicated GUI. As mentioned in the introduction adding bells and whistles are left alone for a further study now that a basic frame work has been developed.

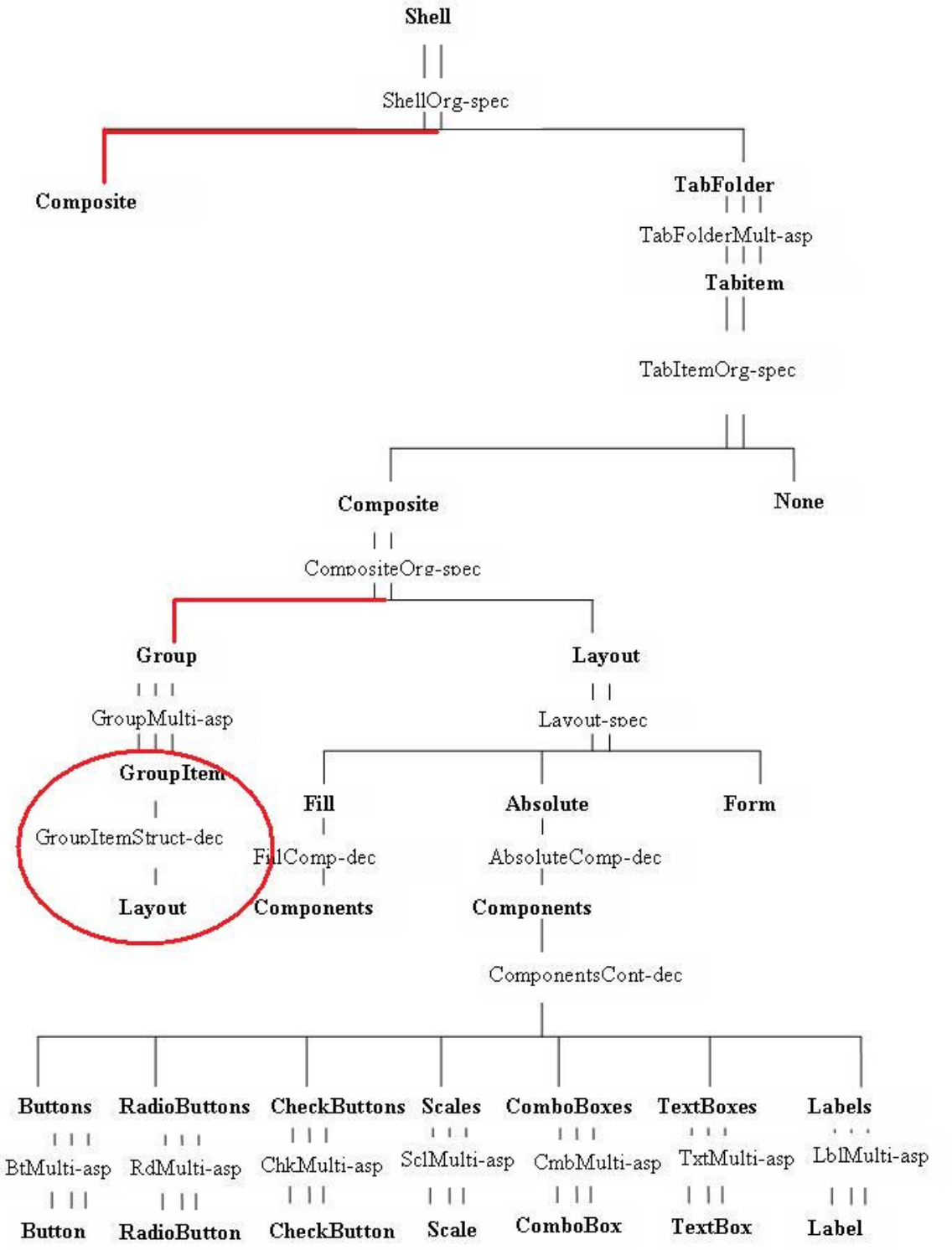

Figure 35 PES for GenetScope Experimental Frame

SUMMARY

The main focus of this study was to extend the knowledge representation features of a System Entity Structure to account for Graphical User Interfaces, and develop a framework where the user can customize data in order to auto generate fully functional GUI code from it. As the eventual auto generation of GUI code need, as a first step, sufficient data to be embedded into a data model for analysis and extraction, a graphical library needed to be identified and mapped. Due to its simple hierarchy, the Standard Widget Toolkit (SWT) library was chosen as the candidate to be mapped into an SES tree. The key concepts of hierarchical decomposition and specialization of SESs were used to accommodate for the mappings. As SESs are usually depicted as visual trees, a method for translation from the format into electronic data had to be identified. Based on the fact that XML is a hierarchical ontology framework that was recognized of having closer connections with SES, it was used as the data storage model of choice. The process conversion from a visual tree SES to a candidate System Entity Structure in XML that could be customized by a user through pruning and modifications involved several intermediate steps. Initially, a Natural Language was used to translate the aforementioned SWT SES from the visual to textual format. Several key variables were also introduced to accommodate for the user customization of data at each level of the tree. A web hosted tool named Virtual Working Table was used to do the process translations in XML. Once the candidate XML was generated, through examples, the pruning and customization

process was introduced so that a final version of a Pruned Entity Structure could be generated by the user as input for GUI code generation.

The usage of the tool named SWTParser which would generate fully functional GUI code in java taking the user customized PES mentioned above as input was also discussed. SWTParser was developed as part of this research effort. Once it could be presumed that a user would be comfortable with the customization and the GUI generation processes, an extension of the framework to map a given instance of an SES to a GUI was discussed. It was not the author's intension to introduce the methodology as a comprehensive system that would cover any instance of a given SES (which it is not), but as an example process that would make use of the concepts from the framework discussed in the study. It was found that professional GUIs of a more complicated nature could be supported through this framework with small extensions and future enhancements to it. It was seen that the user customization of a PES could take significantly more time compared to the GUI generation process which is almost trivial. Hence methods to reduce the over head time involved in this process along with its pros and cons were discussed.

During different stages of this study, the author needed to relate to the SES axioms and concepts discussed in chapter 2. Hence a brief discussion emphasizing the key relations which he was able to establish during the process is deemed appropriate.

Relation of the SES Axioms ?

The *uniformity axiom* was made use of during this study in several occasions. Let us be reminded that the axiom states "any two nodes which have the same labels have identical attached variable types and isomorphic sub trees." Note that when using the PES in figure 28, the entity "GroupItem" was decomposed to a "Layout" entity. This Layout entity remains identical to the one that appears in the pruning path of figure 17. Also in figure 28, the "Components" used under "Fill" layout remains the same as the "Components" used in figure 17. Also note that the suggested use of the "Composite" entity in figure 34 is the same as the one that appears in the pruning path of figure 17.

The *valid brothers* axiom which states "No two brothers have the same label" was implicitly used during this study. Note that the sub section 5.3 states that "However the user needs to be careful not to name 2 widgets of the same category (eg 2 labels) with identical names within a single tab." Though reference was not made to it at that point the notion behind it supports the axiom explicitly.

The *strict hierarchy, alternating mode* and *attached variables* axioms were also followed in this study. One can confirm this by referring to any visual SES figure and the attached code in Appendices 1 and 2.

Finally the *inheritance* axiom which states that "every entity in a specialization inherits all the variables, aspects and specializations from the parent of the specialization" was made use of on several occasions. Note that the 'TabFolder" and "Composite" entities under the 'ShellOrg" specialization in figure 5 will inherit all of the variables of

its parent “Shell” entity. This feature had to be used in the SWTParser when tabfolder auto resizing has to be done based on the Shell’s height and width variable values. Also entities “Composite” and “None” of TabItemOrg specialization and “Fill”, “Absolute” and “Form” of Layout specialization follows the axiom.

CONCLUSION AND FUTURE WORK

As mentioned in the Objectives sub section in 1st chapter, the framework discussed in this study was never meant to be introduced as a fault tolerant full fledged system. It was presented as an innovative approach which would hopefully pave way for further research and study in this field. It is said that "Little drops of water makes the mighty ocean".

The SES based GUI development framework has its own merits. As the framework offers portability, and a platform free representation of a Graphical User Interface, it would prove to be advantageous in many regards. The current adaptation (and future expansion) of pruning rules and other constraints would support iteration and thus provide an advantage over drag and drop methods.

The author recognizes the fact that the framework it is still in its elementary stages and will need further expansion to make it even more useful and user friendly. As we saw, it has its inherent limitations. As a start the user customization process needs to be made more user-friendly which could further reduce the overhead time. Maybe a Graphical Interface to input the various variable values after pruning the SWT SES would help this cause. The current framework only supports a certain class of GUIs which could be derived from it. The main restrictions are based at the SWT SES level

itself as there are few choices for pruning. Also adding more variables for user customization at various levels of the SWT SES tree will obviously add to the enhancement of features. On the hind side, more variables to modify mean more time it takes for the user to customize. The brief discussion made in sub section 6.4 to support for more complex GUIs would clarify this further. Also extending the SWT SES to represent, and the SWTParser to generate event handlers could significantly reduce programming effort over conventional methods.

Hence, while expansions to the initial SWT SES will account for greater breadth and depth to the study, the complexity of the framework will also grow in parallel. However, if such enhancements are made, it should be noted that the SWTParser should also be modified to support for the new additions.

As a final note, the author hopes that the readers will recognize the potentials in this study so that future research and development in this field will be encouraged.

APPENDIX 1 SWT SES in NL

A shell can be composite or tabFolder in design!

The shell has a name,height, and width!
The range of shell's name is string with values !
The range of shell's height is int with values [300,300]!
The range of shell's width is int with values [300,400]!
From a mult perspective, tabFolder is made of more than one tabitem!

The tabFolder has tabux,tabuy,tablx, and tably!
The range of tabFolder's tabux is int with values [0,10]!
The range of tabFolder's tabuy is int with values [0,10]!
The range of tabFolder's tablx is int with values [292,392]!
The range of tabFolder's tably is int with values [266,300]!

The tabitem can be none or composite in structure!
The tabitem has a tabname!
The range of tabitem's tabname is string with values !

The composite can be layout or group in drawablearea!

From a mult perspective, group are made of more than one groupitem!
The groupitem has a groupname!
The range of groupitem's groupname is string with values !

From a groupitemstruc perspective groupitem is made of layout!

The layout can be absolute,fill, or form in layouttype!

From absolutecomp perspective, absolute is made of components!

From fillcomp perspective, fill is made of components!

From compwidgets perspective, components is made of
buttons,radiobuttons,checkbuttons, labels, scales, comboboxes,and textboxes !

From a mult perspective, textboxes are made of more than one textbox!
The textbox has a textbgdcolor, textboundux,textbounduy,textboundlx, and textboundly!

The range of textbox's textbgdcolor is string with values white,yellow, and green!
The range of textbox's textboundux is int with values [0,392]!

The range of textbox's textbounduy is int with values [0,266]!
The range of textbox's textboundlx is int with values [0,392]!
The range of textbox's textboundly is int with values [20,250]!
From a mult perspective, comboboxes are made of more than one combobox!
The combobox has a comboboundux,combobounduy,comboboundlx, and comboboundly!

The range of combobox's comboboundux is int with values [0,392]!
The range of combobox's combobounduy is int with values [0,366]!
The range of combobox's comboboundlx is int with values [0,392]!
The range of combobox's comboboundly is int with values [20,350]!

From a mult perspective, checkbuttons are made of more than one checkbutton!
The checkbutton has a chkbuttonname, chkbuttonboundux, chkbuttonbounduy, chkbuttonboundlx, and chkbuttonboundly!

The range of checkbutton's chkbuttonname is string with values !
The range of checkbutton's chkbuttonboundux is int with values [0,392]!
The range of checkbutton's chkbuttonbounduy is int with values [0,366]!
The range of checkbutton's chkbuttonboundlx is int with values [0,382]!
The range of checkbutton's chkbuttonboundly is int with values [20,350]!

From a mult perspective, buttons are made of more than one button!

The button has a buttonname, buttonboundux,

buttonbounduy,buttonboundlx,and buttonboundly!

The range of button's buttonname is string with values !
The range of button's buttonboundux is int with values [0,392]!
The range of button's buttonbounduy is int with values [0,366]!
The range of button's buttonboundlx is int with values [0,382]!
The range of button's buttonboundly is int with values [20,350]!

From a mult perspective radiobuttons are made of more than one rdbutton!

The rdbutton has a rdbuttonname, rdbuttonboundux,
rdbuttonbounduy,rdbuttonboundlx,and rdbuttonboundly!

The range of rdbutton's rdbuttonname is string with values !
The range of rdbutton's rdbuttonboundux is int with values [0,392]!
The range of rdbutton's rdbuttonbounduy is int with values [0,366]!
The range of rdbutton's rdbuttonboundlx is int with values [0,366]!
The range of rdbutton's rdbuttonboundly is int with values [10,350]!

From a mult perspective, labels are made of more than one label!
The label has a lblname,lblbgdcolor, lblboundux,lblbounduy,lblboundlx, and lblboundly!

The range of label's lblname is string with values !
The range of label's lblbgdcolor is string with values white,red,yellow, and green!

The range of label's lblboundux is int with values [0,392]!
The range of label's lblbounduy is int with values [0,366]!
The range of label's lblboundlx is int with values [0,366]!
The range of label's lblboundly is int with values [10,350]!

From a mult perspective, scales are made of more than one scale!
The scale has a sclbgdcolor, sclboundux,sclbounduy,sclboundlx,sclboundly,
sclsetmax,sclsetmin,sclsetincr, and sclsetpgincr!

The range of scale's sclbgdcolor is string with values ,white,red,yellow, and green !
The range of scale's sclboundux is int with values [0,392]!
The range of scale's sclbounduy is int with values [0,366]!
The range of scale's sclboundlx is int with values [0,366]!
The range of scale's sclboundly is int with values [10,350]!
The range of scale's sclsetmax is int with values [0,100]!
The range of scale's sclsetmin is int with values [0,100]!
The range of scale's sclsetincr is int with values [1,10]!
The range of scale's sclsetpgincr is int with values [1,10]!

APPENDIX 2 Candidate PES in XML generated from DTD

```
1   <?xml version='1.0' encoding='UTF-8'?>
2
3   <shell height = "int300,400Value" name = "stringValue" width = "int300,400Value">
4   <shell-designSpec>
5     <tabFolder tablx = "int292,392Value" tably = "int266,366Value" tabux = "int0,10Value" tabuy = "int0,10Value">
6       <aspectsOftabFolder>
7         <tabFolder-multMultiAsp numContainedIntabFolder = "1">
8           <tabitem tabname = "stringValue">
9             <tabitem-structureSpec>
10            <none/>
11              <composite>
12                <composite-drawableareaSpec>
13                  <layout>
14                    <layout-layouttypeSpec>
15                      <absolute>
16                        <aspectsOfabsolute>
17                          <absolute-absolutecompDec>
18                            <components>
19                              <aspectsOfcomponents>
20                                <components-compwidgetsDec>
21                                  <checkbuttons>
22                                    <aspectsOfcheckbuttons>
23                                      <checkbuttons-multMultiAsp numContainedIncheckbuttons = "1">
24                                        <checkbutton chkbuttonboundlx = "int0,382Value" chkbuttonboundly = "
    int20,350Value" chkbuttonboundux = "int0,392Value" chkbuttonbounduy = "int0,366Value" chkbuttonname = "stringValue">
25                                      </checkbutton>
26                                    </checkbuttons-multMultiAsp>
27                                  </aspectsOfcheckbuttons>
28                                </checkbuttons>
29                                <textboxes>
30                                  <aspectsOftextboxes>
31                                    <textboxes-multMultiAsp numContainedIntextboxes = "1">
32                                      <textbox textbgdcolor = "string with values yellow ,green , and whiteValue
    textboundlx = "int0,392Value" textboundly = "int20,350Value" textboundux = "int0,392Value" textbounduy = "int0,366Value">
33                                    </textbox>
34                                  </textboxes-multMultiAsp>
35                                </aspectsOftextboxes>
36                              </textboxes>
37                              <scales>
38                                <aspectsOfscales>
39                                  <scales-multMultiAsp numContainedInscales = "1">
40                                    <scale sclbgdcolor = "string with values white ,red ,yellow ,green , and Value
    sclboundlx = "int0,366Value" sclboundly = "int10,350Value" sclboundux = "int0,392Value" sclbounduy = "int0,366Value"
    sclsetincr = "int1,10Value" sclsetmax = "int0,100Value" sclsetmin = "int0,100Value" sclsetpgincr = "int1,10Value">
41                                    </scale>
42                                  </scales-multMultiAsp>
43                                </aspectsOfscales>
44                              </scales>
```

```
45                                        <comboboxes>
46                                          <aspectsOfcomboboxes>
47                                            <comboboxes-multMultiAsp numContainedIncomboboxes= "1">
48                                              <combobox comboboundIx = "int0,392Value" comboboundly = "
    int20,350Value" comboboundux = "int0,392Value" comboboundуy = "int0,366Value">
49                                              </combobox>
50                                            </comboboxes-multMultiAsp>
51                                          </aspectsOfcomboboxes>
52                                        </comboboxes>
53                                        <labels>
54                                          <aspectsOflabels>
55                                            <labels-multMultiAsp numContainedInlabels= "1">
56                                              <label lblbgdcolor = "string with values red ,yellow ,green , and whiteValue
    lblboundIx = "int0,366Value" lblboundly = "int10,350Value" lblboundux = "int0,392Value" lblboundуy = "int0,366Value" lblname
    = "stringValue">
57                                              </label>
58                                            </labels-multMultiAsp>
59                                          </aspectsOflabels>
60                                        </labels>
61                                        <buttons>
62                                          <aspectsOfbuttons>
63                                            <buttons-multMultiAsp numContainedInbuttons= "1">
64                                              <button buttonboundIx = "int0,382Value" buttonboundly = "int20,350Value"
    buttonboundux = "int0,392Value" buttonbounduy = "int0,366Value" buttonname = "stringValue">
65                                              </button>
66                                            </buttons-multMultiAsp>
67                                          </aspectsOfbuttons>
68                                        </buttons>
69                                        <radiobuttons>
70                                          <aspectsOfradiobuttons>
71                                            <radiobuttons-multMultiAsp numContainedInradiobuttons= "1">
72                                              <rdbutton rdbuttonboundIx = "int0,366Value" rdbuttonboundly = "
    int10,350Value" rdbuttonboundux = "int0,392Value" rdbuttonbounduy = "int0,366Value" rdbuttonname = "stringValue">
73                                              </rdbutton>
74                                            </radiobuttons-multMultiAsp>
75                                          </aspectsOfradiobuttons>
76                                        </radiobuttons>
77                                      </components-compwidgetsDec>
78                                    </aspectsOfcomponents>
79                                  </components>
80                                </absolute-absolutecompDec>
81                              </aspectsOfabsolute>
82                            </absolute>
83                          <form/>
84                            <fill>
85                              <aspectsOffill>
86                                <fill-fillcompDec>
87                                  <components>
```

```
88                                  <aspectsOfcomponents>
89                                    <components-compwidgetsDe>
90                                      <checkbuttons>
91                                        <aspectsOfcheckbuttons>
92                                          <checkbuttons-multMultiAsp numContainedIncheckbuttons= "1">
93                                            <checkbutton chkbuttonboundlx = "int0,382Value" chkbuttonboundly = "
    int20,350Value" chkbuttonboundux = "int0,392Value" chkbuttonbounduy = "int0,366Value" chkbuttonname = "stringValue">
94                                            </checkbutton>
95                                          </checkbuttons-multMultiAsp>
96                                        </aspectsOfcheckbuttons>
97                                      </checkbuttons>
98                                      <textboxes>
99                                        <aspectsOftextboxes>
100                                         <textboxes-multMultiAsp numContainedIntextboxes= "1">
101                                           <textbox textbgdcolor = "string with values yellow ,green , and whiteValue
    textboundlx = "int0,392Value" textboundly = "int20,350Value" textboundux = "int0,392Value" textbounduy = "int0,366Value">
102                                           </textbox>
103                                         </textboxes-multMultiAsp>
104                                       </aspectsOftextboxes>
105                                     </textboxes>
106                                     <scales>
107                                       <aspectsOfscales>
108                                         <scales-multMultiAsp numContainedInscales= "1">
109                                           <scale sclbgdcolor = "string with values white ,red ,yellow ,green , and Value
    sclboundlx = "int0,366Value" sclboundly = "int10,350Value" sclboundux = "int0,392Value" sclbounduy = "int0,366Value"
    sclsetincr = "int1,10Value" sclsetmax = "int0,100Value" sclsetmin = "int0,100Value" sclsetpgincr = "int1,10Value">
110                                           </scale>
111                                         </scales-multMultiAsp>
112                                       </aspectsOfscales>
113                                     </scales>
114                                     <comboboxes>
115                                       <aspectsOfcomboboxes>
116                                         <comboboxes-multMultiAsp numContainedIncomboboxes= "1">
117                                           <combobox comboboundlx = "int0,392Value" comboboundly = "
    int20,350Value" comboboundux = "int0,392Value" comboboundy = "int0,366Value">
118                                           </combobox>
119                                         </comboboxes-multMultiAsp>
120                                       </aspectsOfcomboboxes>
121                                     </comboboxes>
122                                     <labels>
123                                       <aspectsOflabels>
124                                         <labels-multMultiAsp numContainedInlabels= "1">
125                                           <label lblbgdcolor = "string with values red ,yellow ,green , and whiteValue
    lblboundlx = "int0,366Value" lblboundly = "int10,350Value" lblboundux = "int0,392Value" lblbounduy = "int0,366Value" lblname
    = "stringValue">
126                                           </label>
127                                         </labels-multMultiAsp>
128                                       </aspectsOflabels>
```

```
129                                    </labels>
130                                    <buttons>
131                                      <aspectsOfbuttons>
132                                        <buttons-multMultiAsp numContainedInbuttons= "1">
133                                          <button buttonboundlx = "int0,382Value" buttonboundly = "int20,350Value"
    buttonboundux = "int0,392Value" buttonbounduy = "int0,366Value" buttonname = "stringValue">
134                                          </button>
135                                        </buttons-multMultiAsp>
136                                      </aspectsOfbuttons>
137                                    </buttons>
138                                    <radiobuttons>
139                                      <aspectsOfradiobuttons>
140                                        <radiobuttons-multMultiAsp numContainedInradiobuttons= "1">
141                                          <rdbutton rdbuttonboundlx = "int0,366Value" rdbuttonboundly = "
    int10,350Value" rdbuttonboundux = "int0,392Value" rdbuttonbounduy = "int0,366Value" rdbuttonname = "stringValue">
142                                          </rdbutton>
143                                        </radiobuttons-multMultiAsp>
144                                      </aspectsOfradiobuttons>
145                                    </radiobuttons>
146                                  </components-compwidgetsDec>
147                                </aspectsOfcomponents>
148                              </components>
149                            </fill-fillcompDec>
150                          </aspectsOffill>
151                        </fill>
152                      </layout-layouttypeSpec>
153                    </layout>
154                    <group>
155                      <aspectsOfgroup>
156                        <group-multMultiAsp numContainedIngroup= "1">
157                          <groupitem groupname = "stringValue">
158                            <aspectsOfgroupitem>
159                              <groupitem-groupitemstrucDec>
160                                <layout>
161                                  <layout-layouttypeSpec>
162                                    <absolute>
163                                      <aspectsOfabsolute>
164                                        <absolute-absolutecompDec>
165                                          <components>
166                                            <aspectsOfcomponents>
167                                              <components-compwidgetsDec>
168                                                <checkbuttons>
169                                                  <aspectsOfcheckbuttons>
170                                                    <checkbuttons-multMultiAsp numContainedIncheckbuttons= "1"
171                                                      <checkbutton chkbuttonboundlx = "int0,382Value"
    chkbuttonboundly = "int20,350Value" chkbuttonboundux = "int0,392Value" chkbuttonbounduy = "int0,366Value"
    chkbuttonname = "stringValue">
172                                                    </checkbutton>
```

```
173                                                    </checkbuttons-multMultiAsp>
174                                                 </aspectsOfcheckbuttons>
175                                              </checkbuttons>
176                                              <textboxes>
177                                                 <aspectsOftextboxes>
178                                                    <textboxes-multMultiAsp numContainedIntextboxes= "1">
179                                                       <textbox textbgdcolor = "string with values yellow ,green ,
and whiteValue" textboundlx = "int0,392Value" textboundly = "int20,350Value" textboundux = "int0,392Value" textbounduy = "
int0,366Value">
180                                                       </textbox>
181                                                    </textboxes-multMultiAsp>
182                                                 </aspectsOftextboxes>
183                                              </textboxes>
184                                              <scales>
185                                                 <aspectsOfscales>
186                                                    <scales-multMultiAsp numContainedInscales= "1">
187                                                       <scale sclbgdcolor = "string with values white ,red ,yellow
,green , and Value" sclboundlx = "int0,366Value" sclboundly = "int10,350Value" sclboundux = "int0,392Value" sclbounduy = "
int0,366Value" sclsetincr = "int1,10Value" sclsetmax = "int0,100Value" sclsetmin = "int0,100Value" sclsetpgincr = "int1,10Value
">
188                                                       </scale>
189                                                    </scales-multMultiAsp>
190                                                 </aspectsOfscales>
191                                              </scales>
192                                              <comboboxes>
193                                                 <aspectsOfcomboboxes>
194                                                    <comboboxes-multMultiAsp numContainedIncomboboxes= "1">
195                                                       <combobox comboboundlx = "int0,392Value" comboboundly
= "int20,350Value" comboboundux = "int0,392Value" combobounduy = "int0,366Value">
196                                                       </combobox>
197                                                    </comboboxes-multMultiAsp>
198                                                 </aspectsOfcomboboxes>
199                                              </comboboxes>
200                                              <labels>
201                                                 <aspectsOflabels>
202                                                    <labels-multMultiAsp numContainedInlabels= "1">
203                                                       <label lblbgdcolor = "string with values red ,yellow ,green ,
and whiteValue" lblboundlx = "int0,366Value" lblboundly = "int10,350Value" lblboundux = "int0,392Value" lblbounduy = "
int0,366Value" lblname = "stringValue">
204                                                       </label>
205                                                    </labels-multMultiAsp>
206                                                 </aspectsOflabels>
207                                              </labels>
208                                              <buttons>
209                                                 <aspectsOfbuttons>
210                                                    <buttons-multMultiAsp numContainedInbuttons= "1">
211                                                       <button buttonboundlx = "int0,382Value" buttonboundly = "
int20,350Value" buttonboundux = "int0,392Value" buttonbounduy = "int0,366Value" buttonname = "stringValue">
```

```
212                                                    </button>
213                                                  </buttons-multMultiAsp>
214                                                </aspectsOfbuttons>
215                                              </buttons>
216                                              <radiobuttons>
217                                                <aspectsOfradiobuttons>
218                                                  <radiobuttons-multMultiAsp numContainedInradiobuttons= "1">
219                                                    <rdbutton rdbuttonboundlx = "int0,366Value" rdbuttonboundly
    = "int10,350Value" rdbuttonboundux = "int0,392Value" rdbuttonbounduy = "int0,366Value" rdbuttonname = "stringValue">
220                                                    </rdbutton>
221                                                  </radiobuttons-multMultiAsp>
222                                                </aspectsOfradiobuttons>
223                                              </radiobuttons>
224                                            </components-compwidgetsDec>
225                                          </aspectsOfcomponents>
226                                        </components>
227                                      </absolute-absolutecompDec>
228                                    </aspectsOfabsolute>
229                                  </absolute>
230                                <form/>
231                                  <fill>
232                                    <aspectsOffill>
233                                      <fill-fillcompDec>
234                                        <components>
235                                          <aspectsOfcomponents>
236                                            <components-compwidgetsDec>
237                                              <checkbuttons>
238                                                <aspectsOfcheckbuttons>
239                                                  <checkbuttons-multMultiAsp numContainedIncheckbuttons= "1"
240                                                    <checkbutton chkbuttonboundlx = "int0,382Value"
    chkbuttonboundly = "int20,350Value" chkbuttonboundux = "int0,392Value" chkbuttonbounduy = "int0,366Value"
    chkbuttonname = "stringValue">
241                                                    </checkbutton>
242                                                  </checkbuttons-multMultiAsp>
243                                                </aspectsOfcheckbuttons>
244                                              </checkbuttons>
245                                              <textboxes>
246                                                <aspectsOftextboxes>
247                                                  <textboxes-multMultiAsp numContainedIntextboxes= "1">
248                                                    <textbox textbgdcolor = "string with values yellow ,green ,
    and whiteValue" textboundlx = "int0,392Value" textboundly = "int20,350Value" textboundux = "int0,392Value" textbounduy = "
    int0,366Value">
249                                                    </textbox>
250                                                  </textboxes-multMultiAsp>
251                                                </aspectsOftextboxes>
252                                              </textboxes>
253                                              <scales>
254                                                <aspectsOfscales>
```

```
255                                                    <scales-multMultiAsp numContainedInscales= "1">
256                                                      <scale sclbgdcolor = "string with values white ,red ,yellow
,green , and Value" sclboundlx = "int0,366Value" sclboundly = "int10,350Value" sclboundux = "int0,392Value" sclbounduy = "
int0,366Value" sclsetincr = "int1,10Value" sclsetmax = "int0,100Value" sclsetmin = "int0,100Value" sclsetpgincr = "int1,10Value
">
257                                                      </scale>
258                                                    </scales-multMultiAsp>
259                                                  </aspectsOfscales>
260                                                </scales>
261                                                <comboboxes>
262                                                  <aspectsOfcomboboxes>
263                                                    <comboboxes-multMultiAsp numContainedIncomboboxes= "1">
264                                                      <combobox comboboundlx = "int0,392Value" comboboundly
= "int20,350Value" comboboundux = "int0,392Value" combobounduy = "int0,366Value">
265                                                      </combobox>
266                                                    </comboboxes-multMultiAsp>
267                                                  </aspectsOfcomboboxes>
268                                                </comboboxes>
269                                                <labels>
270                                                  <aspectsOflabels>
271                                                    <labels-multMultiAsp numContainedInlabels= "1">
272                                                      <label lblbgdcolor = "string with values red ,yellow ,green ,
and whiteValue" lblboundlx = "int0,366Value" lblboundly = "int10,350Value" lblboundux = "int0,392Value" lblbounduy = "
int0,366Value" lblname = "stringValue">
273                                                      </label>
274                                                    </labels-multMultiAsp>
275                                                  </aspectsOflabels>
276                                                </labels>
277                                                <buttons>
278                                                  <aspectsOfbuttons>
279                                                    <buttons-multMultiAsp numContainedInbuttons= "1">
280                                                      <button buttonboundlx = "int0,382Value" buttonboundly = "
int20,350Value" buttonboundux = "int0,392Value" buttonbounduy = "int0,366Value" buttonname = "stringValue">
281                                                      </button>
282                                                    </buttons-multMultiAsp>
283                                                  </aspectsOfbuttons>
284                                                </buttons>
285                                                <radiobuttons>
286                                                  <aspectsOfradiobuttons>
287                                                    <radiobuttons-multMultiAsp numContainedInradiobuttons= "1">
288                                                      <rdbutton rdbuttonboundlx = "int0,366Value" rdbuttonboundly
= "int10,350Value" rdbuttonboundux = "int0,392Value" rdbuttonbounduy = "int0,366Value" rdbuttonname = "stringValue">
289                                                      </rdbutton>
290                                                    </radiobuttons-multMultiAsp>
291                                                  </aspectsOfradiobuttons>
292                                                </radiobuttons>
293                                              </components-compwidgetsDec>
294                                            </aspectsOfcomponents>
```

```
295                                                    </components>
296                                                  </fill-fillcompDec>
297                                                </aspectsOffill>
298                                              </fill>
299                                            </layout-layouttypeSpec>
300                                          </layout>
301                                        </groupitem-groupitemstrucDec>
302                                      </aspectsOfgroupitem>
303                                    </groupitem>
304                                  </group-multMultiAsp>
305                                </aspectsOfgroup>
306                              </group>
307                            </composite-drawableareaSpec>
308                          </composite>
309                        </tabitem-structureSpec>
310                      </tabitem>
311                    </tabFolder-multMultiAsp>
312                  </aspectsOftabFolder>
313                </tabFolder>
314                <composite>
315                  <composite-drawableareaSpec>
316                    <layout>
317                      <layout-layouttypeSpec>
318                        <absolute>
319                          <aspectsOfabsolute>
320                            <absolute-absolutecompDec>
321                              <components>
322                                <aspectsOfcomponents>
323                                  <components-compwidgetsDec>
324                                    <checkbuttons>
325                                      <aspectsOfcheckbuttons>
326                                        <checkbuttons-multMultiAsp numContainedIncheckbuttons= "1">
327                                          <checkbutton  chkbuttonboundlx= "int0,382Value" chkbuttonboundly= "int20,350Value"
chkbuttonboundux= "int0,392Value" chkbuttonbounduy= "int0,366Value" chkbuttonname = "stringValue">
328                                          </checkbutton>
329                                        </checkbuttons-multMultiAsp>
330                                      </aspectsOfcheckbuttons>
331                                    </checkbuttons>
332                                    <textboxes>
333                                      <aspectsOftextboxes>
334                                        <textboxes-multMultiAsp  numContainedIntextboxes= "1">
335                                          <textbox  textbgdcolor = "string with values yellow ,green , and whiteValue" textboundlx = "
int0,392Value" textboundly = "int20,350Value" textboundux = "int0,392Value" textbounduy = "int0,366Value">
336                                          </textbox>
337                                        </textboxes-multMultiAsp>
338                                      </aspectsOftextboxes>
339                                    </textboxes>
340                                    <scales>
```

```
341                                    <aspectsOfscales>
342                                       <scales-multMultiAsp numContainedInscales= "1">
343                                          <scale sclbgdcolor = "string with values white ,red ,yellow ,green , and Valuë sclboundlx
= "int0,366Value" sclboundly = "int10,350Value" sclboundux = "int0,392Value" sclbounduy = "int0,366Value" sclsetincr = "
int1,10Value" sclsetmax = "int0,100Value" sclsetmin = "int0,100Value" sclsetpgincr = "int1,10Value">
344                                          </scale>
345                                       </scales-multMultiAsp>
346                                    </aspectsOfscales>
347                                 </scales>
348                                 <comboboxes>
349                                    <aspectsOfcomboboxes>
350                                       <comboboxes-multMultiAsp numContainedIncomboboxes= "1">
351                                          <combobox comboboundlx = "int0,392Value" comboboundly = "int20,350Value"
comboboundux = "int0,392Value" combobounduy = "int0,366Value">
352                                          </combobox>
353                                       </comboboxes-multMultiAsp>
354                                    </aspectsOfcomboboxes>
355                                 </comboboxes>
356                                 <labels>
357                                    <aspectsOflabels>
358                                       <labels-multMultiAsp numContainedInlabels= "1">
359                                          <label lblbgdcolor = "string with values red ,yellow ,green , and whiteValuë lblboundlx = "
int0,366Value" lblboundly = "int10,350Value" lblboundux = "int0,392Value" lblbounduy = "int0,366Value" lblname = "stringValue
">
360                                          </label>
361                                       </labels-multMultiAsp>
362                                    </aspectsOflabels>
363                                 </labels>
364                                 <buttons>
365                                    <aspectsOfbuttons>
366                                       <buttons-multMultiAsp numContainedInbuttons= "1">
367                                          <button buttonboundlx = "int0,382Value" buttonboundly = "int20,350Value"
buttonboundux = "int0,392Value" buttonbounduy = "int0,366Value" buttonname = "stringValue">
368                                          </button>
369                                       </buttons-multMultiAsp>
370                                    </aspectsOfbuttons>
371                                 </buttons>
372                                 <radiobuttons>
373                                    <aspectsOfradiobuttons>
374                                       <radiobuttons-multMultiAsp numContainedInradiobuttons= "1">
375                                          <rdbutton rdbuttonboundlx = "int0,366Value" rdbuttonboundly = "int10,350Value"
rdbuttonboundux = "int0,392Value" rdbuttonbounduy = "int0,366Value" rdbuttonname = "stringValue">
376                                          </rdbutton>
377                                       </radiobuttons-multMultiAsp>
378                                    </aspectsOfradiobuttons>
379                                 </radiobuttons>
380                              </components-compwidgetsDe
381                           </aspectsOfcomponents>
```

```
382                        </components>
383                      </absolute-absolutecompDec>
384                    </aspectsOfabsolute>
385                  </absolute>
386                <form/>
387                  <fill>
388                    <aspectsOffill>
389                      <fill-fillcompDec>
390                        <components>
391                          <aspectsOfcomponents>
392                            <components-compwidgetsDec>
393                              <checkbuttons>
394                                <aspectsOfcheckbuttons>
395                                  <checkbuttons-multMultiAsp numContainedIncheckbuttons= "1">
396                                    <checkbutton chkbuttonboundlx = "int0,382Value" chkbuttonboundly = "int20,350Value"
chkbuttonboundux = "int0,392Value" chkbuttonbounduy = "int0,366Value" chkbuttonname = "stringValue">
397                                    </checkbutton>
398                                  </checkbuttons-multMultiAsp>
399                                </aspectsOfcheckbuttons>
400                              </checkbuttons>
401                              <textboxes>
402                                <aspectsOftextboxes>
403                                  <textboxes-multMultiAsp numContainedIntextboxes= "1">
404                                    <textbox textbgdcolor = "string with values yellow ,green , and whiteValue textboundlx = "
int0,392Value" textboundly = "int20,350Value" textboundux = "int0,392Value" textbounduy = "int0,366Value">
405                                    </textbox>
406                                  </textboxes-multMultiAsp>
407                                </aspectsOftextboxes>
408                              </textboxes>
409                              <scales>
410                                <aspectsOfscales>
411                                  <scales-multMultiAsp numContainedInscales= "1">
412                                    <scale sclbgdcolor = "string with values white ,red ,yellow ,green , and Value sclboundlx
= "int0,366Value" sclboundly = "int10,350Value" sclboundux = "int0,392Value" sclbounduy = "int0,366Value" sclsetincr = "
int1,10Value" sclsetmax = "int0,100Value" sclsetmin = "int0,100Value" sclsetpgincr = "int1,10Value">
413                                    </scale>
414                                  </scales-multMultiAsp>
415                                </aspectsOfscales>
416                              </scales>
417                              <comboboxes>
418                                <aspectsOfcomboboxes>
419                                  <comboboxes-multMultiAsp numContainedIncomboboxes= "1">
420                                    <combobox comboboundlx = "int0,392Value" comboboundly = "int20,350Value"
comboboundux = "int0,392Value" combobounduy = "int0,366Value">
421                                    </combobox>
422                                  </comboboxes-multMultiAsp>
423                                </aspectsOfcomboboxes>
424                              </comboboxes>
```

```
425                                 <labels>
426                                   <aspectsOflabels>
427                                     <labels-multMultiAsp numContainedInlabels = "1">
428                                       <label lblbgdcolor = "string with values red ,yellow ,green , and whiteValue lblboundlx = "
    int0,366Value" lblboundly = "int10,350Value" lblboundux = "int0,392Value" lblbounduy = "int0,366Value" lblname = "stringValue
    ">
429                                       </label>
430                                     </labels-multMultiAsp>
431                                   </aspectsOflabels>
432                                 </labels>
433                                 <buttons>
434                                   <aspectsOfbuttons>
435                                     <buttons-multMultiAsp numContainedInbuttons = "1">
436                                       <button buttonboundlx = "int0,382Value" buttonboundly = "int20,350Value"
    buttonboundux = "int0,392Value" buttonbounduy = "int0,366Value" buttonname = "stringValue">
437                                       </button>
438                                     </buttons-multMultiAsp>
439                                   </aspectsOfbuttons>
440                                 </buttons>
441                                 <radiobuttons>
442                                   <aspectsOfradiobuttons>
443                                     <radiobuttons-multMultiAsp numContainedInradiobuttons = "1">
444                                       <rdbutton rdbuttonboundlx = "int0,366Value" rdbuttonboundly = "int10,350Value"
    rdbuttonboundux = "int0,392Value" rdbuttonbounduy = "int0,366Value" rdbuttonname = "stringValue">
445                                       </rdbutton>
446                                     </radiobuttons-multMultiAsp>
447                                   </aspectsOfradiobuttons>
448                                 </radiobuttons>
449                               </components-compwidgetsDec>
450                             </aspectsOfcomponents>
451                           </components>
452                         </fill-fillcompDec>
453                       </aspectsOffill>
454                     </fill>
455                   </layout-layouttypeSpec>
456                 </layout>
457                 <group>
458                   <aspectsOfgroup>
459                     <group-multMultiAsp numContainedIngroup = "1">
460                       <groupitem groupname = "stringValue">
461                         <aspectsOfgroupitem>
462                           <groupitem-groupitemstrucDec>
463                             <layout>
464                               <layout-layouttypeSpec>
465                                 <absolute>
466                                   <aspectsOfabsolute>
467                                     <absolute-absolutecompDec>
468                                       <components>
```

```
469                                    <aspectsOfcomponents>
470                                      <components-compwidgetsDe>
471                                        <checkbuttons>
472                                          <aspectsOfcheckbuttons>
473                                            <checkbuttons-multMultiAsp numContainedIncheckbuttons= "1">
474                                              <checkbutton chkbuttonboundlx = "int0,382Value" chkbuttonboundly = "
int20,350Value" chkbuttonboundux = "int0,392Value" chkbuttonbounduy = "int0,366Value" chkbuttonname = "stringValue">
475                                              </checkbutton>
476                                            </checkbuttons-multMultiAsp>
477                                          </aspectsOfcheckbuttons>
478                                        </checkbuttons>
479                                        <textboxes>
480                                          <aspectsOftextboxes>
481                                            <textboxes-multMultiAsp numContainedIntextboxes = "1">
482                                              <textbox textbgdcolor = "string with values yellow ,green , and whiteValue
textboundlx = "int0,392Value" textboundly = "int20,350Value" textboundux = "int0,392Value" textbounduy = "int0,366Value">
483                                              </textbox>
484                                            </textboxes-multMultiAsp>
485                                          </aspectsOftextboxes>
486                                        </textboxes>
487                                        <scales>
488                                          <aspectsOfscales>
489                                            <scales-multMultiAsp numContainedInscales= "1">
490                                              <scale sclbgdcolor = "string with values white ,red ,yellow ,green , and
Value" sclboundlx = "int0,366Value" sclboundly = "int10,350Value" sclboundux = "int0,392Value" sclbounduy = "int0,366Value"
sclsetincr = "int1,10Value" sclsetmax = "int0,100Value" sclsetmin = "int0,100Value" sclsetpgincr = "int1,10Value">
491                                              </scale>
492                                            </scales-multMultiAsp>
493                                          </aspectsOfscales>
494                                        </scales>
495                                        <comboboxes>
496                                          <aspectsOfcomboboxes>
497                                            <comboboxes-multMultiAsp numContainedIncomboboxes= "1">
498                                              <combobox comboboundlx = "int0,392Value" comboboundly = "
int20,350Value" comboboundux = "int0,392Value" comboboundbuy = "int0,366Value">
499                                              </combobox>
500                                            </comboboxes-multMultiAsp>
501                                          </aspectsOfcomboboxes>
502                                        </comboboxes>
503                                        <labels>
504                                          <aspectsOflabels>
505                                            <labels-multMultiAsp numContainedInlabels = "1">
506                                              <label lblbgdcolor = "string with values red ,yellow ,green , and whiteValue
lblboundlx = "int0,366Value" lblboundly = "int10,350Value" lblbounduy = "int0,392Value" lblbounduy = "int0,366Value" lblname
= "stringValue">
507                                              </label>
508                                            </labels-multMultiAsp>
509                                          </aspectsOflabels>
```

```
510                                    </labels>
511                                    <buttons>
512                                      <aspectsOfbuttons>
513                                        <buttons-multMultiAsp numContainedInbuttons= "1">
514                                          <button buttonboundlx = "int0,382Value" buttonboundly = "int20,350Value"
    buttonboundux = "int0,392Value" buttonbounduy = "int0,366Value" buttonname = "stringValue">
515                                          </button>
516                                        </buttons-multMultiAsp>
517                                      </aspectsOfbuttons>
518                                    </buttons>
519                                    <radiobuttons>
520                                      <aspectsOfradiobuttons>
521                                        <radiobuttons-multMultiAsp numContainedInradiobuttons= "1">
522                                          <rdbutton rdbuttonboundlx = "int0,366Value" rdbuttonboundly = "
    int10,350Value" rdbuttonboundux = "int0,392Value" rdbuttonbounduy = "int0,366Value" rdbuttonname = "stringValue">
523                                          </rdbutton>
524                                        </radiobuttons-multMultiAsp>
525                                      </aspectsOfradiobuttons>
526                                    </radiobuttons>
527                                  </components-compwidgetsDec>
528                                </aspectsOfcomponents>
529                              </components>
530                            </absolute-absolutecompDec>
531                          </aspectsOfabsolute>
532                        </absolute>
533                      <form/>
534                        <fill>
535                          <aspectsOffill>
536                            <fill-fillcompDec>
537                              <components>
538                                <aspectsOfcomponents>
539                                  <components-compwidgetsDec>
540                                    <checkbuttons>
541                                      <aspectsOfcheckbuttons>
542                                        <checkbuttons-multMultiAsp numContainedIncheckbuttons= "1">
543                                          <checkbutton chkbuttonboundlx = "int0,382Value" chkbuttonboundly = "
    int20,350Value" chkbuttonboundux = "int0,392Value" chkbuttonbounduy = "int0,366Value" chkbuttonname = "stringValue">
544                                          </checkbutton>
545                                        </checkbuttons-multMultiAsp>
546                                      </aspectsOfcheckbuttons>
547                                    </checkbuttons>
548                                    <textboxes>
549                                      <aspectsOftextboxes>
550                                        <textboxes-multMultiAsp numContainedIntextboxes = "1">
551                                          <textbox textbgdcolor = "string with values yellow ,green , and whiteValue
    textboundlx = "int0,392Value" textboundly = "int20,350Value" textboundux = "int0,392Value" textbounduy = "int0,366Value">
552                                          </textbox>
553                                        </textboxes-multMultiAsp>
```

```
554                                          </aspectsOftextboxes>
555                                        </textboxes>
556                                        <scales>
557                                          <aspectsOfscales>
558                                            <scales-multMultiAsp numContainedInscales= "1">
559                                              <scale sclbgdcolor = "string with values white ,red ,yellow ,green , and
    Value" sclboundlx = "int0,366Value" sclboundly = "int10,350Value" sclbounddux = "int0,392Value" sclbounduy = "int0,366Value"
    sclsetincr = "int1,10Value" sclsetmax = "int0,100Value" sclsetmin = "int0,100Value" sclsetpgincr = "int1,10Value">
560                                              </scale>
561                                            </scales-multMultiAsp>
562                                          </aspectsOfscales>
563                                        </scales>
564                                        <comboboxes>
565                                          <aspectsOfcomboboxes>
566                                            <comboboxes-multMultiAsp numContainedIncomboboxes= "1">
567                                              <combobox comboboundlx = "int0,392Value" comboboundly = "
    int20,350Value" comboboundux = "int0,392Value" comboboundduy = "int0,366Value">
568                                              </combobox>
569                                            </comboboxes-multMultiAsp>
570                                          </aspectsOfcomboboxes>
571                                        </comboboxes>
572                                        <labels>
573                                          <aspectsOflabels>
574                                            <labels-multMultiAsp numContainedInlabels= "1">
575                                              <label lblbgdcolor = "string with values red ,yellow ,green , and whiteValue
    lblboundlx = "int0,366Value" lblboundly = "int10,350Value" lblbounddux = "int0,392Value" lblbounduy = "int0,366Value" lblname
    = "stringValue">
576                                              </label>
577                                            </labels-multMultiAsp>
578                                          </aspectsOflabels>
579                                        </labels>
580                                        <buttons>
581                                          <aspectsOfbuttons>
582                                            <buttons-multMultiAsp numContainedInbuttons= "1">
583                                              <button buttonboundlx = "int0,382Value" buttonboundly = "int20,350Value"
    buttonbounddux = "int0,392Value" buttonbounduy = "int0,366Value" buttonname = "stringValue">
584                                              </button>
585                                            </buttons-multMultiAsp>
586                                          </aspectsOfbuttons>
587                                        </buttons>
588                                        <radiobuttons>
589                                          <aspectsOfradiobuttons>
590                                            <radiobuttons-multMultiAsp numContainedInradiobuttons= "1">
591                                              <rdbutton rdbuttonboundlx = "int0,366Value" rdbuttonboundly = "
    int10,350Value" rdbuttonbounddux = "int0,392Value" rdbuttonbounduy = "int0,366Value" rdbuttonname = "stringValue">
592                                              </rdbutton>
593                                            </radiobuttons-multMultiAsp>
594                                          </aspectsOfradiobuttons>
```

```
595                                        </radiobuttons>
596                                      </components-compwidgetsDec>
597                                    </aspectsOfcomponents>
598                                  </components>
599                                </fill-fillcompDec>
600                              </aspectsOffill>
601                            </fill>
602                          </layout-layouttypeSpec>
603                        </layout>
604                      </groupitem-groupitemstrucDec>
605                    </aspectsOfgroupitem>
606                  </groupitem>
607                </group-multMultiAsp>
608              </aspectsOfgroup>
609            </group>
610          </composite-drawableareaSpec>
611        </composite>
612    </shell-designSpec>
613    </shell>
614
```

APPENDIX 3 SimpleGUI1.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <shell height="300" name="SimpleGUI1" width="400">
3    <shell-designSpec>
4      <tabFolder tablx="392" tably="266" tabux="0" tabuy="0">
5        <aspectsOftabFolder>
6          <tabFolder-multMultiAsp numContainedIntabFolder="1">
7            <tabitem tabname="Personal">
8              <tabitem-structureSpec>
9                <composite>
10                 <composite-drawableareaSpec>
11                   <layout>
12                     <layout-layouttypeSpec>
13                       <absolute>
14                         <aspectsOfabsolute>
15                           <absolute-absolutecompDec>
16                             <components>
17                               <aspectsOfcomponents>
18                                 <components-compwidgetsDec>
19                                   <checkbuttons>
20                                     <aspectsOfcheckbuttons>
21                                       <checkbuttons-multMultiAsp numContainedIncheckbuttons="0"/>
22                                     </aspectsOfcheckbuttons>
23                                   </checkbuttons>
24                                   <textboxes>
25                                     <aspectsOftextboxes>
26                                       <textboxes-multMultiAsp numContainedIntextboxes="1">
27                                         <textbox textbgdcolor="white" textboundlx="204" textboundly="29"
   textboundux="111" textbounduy="20">
28                                       </textbox>
29                                       </textboxes-multMultiAsp>
30                                     </aspectsOftextboxes>
31                                   </textboxes>
32                                   <scales>
33                                     <aspectsOfscales>
34                                       <scales-multMultiAsp numContainedInscales="0"/>
35                                     </aspectsOfscales>
36                                   </scales>
37                                   <comboboxes>
38                                     <aspectsOfcomboboxes>
39                                       <comboboxes-multMultiAsp numContainedIncomboboxes="0"/>
40                                     </aspectsOfcomboboxes>
41                                   </comboboxes>
42                                   <labels>
43                                     <aspectsOflabels>
44                                       <labels-multMultiAsp numContainedInlabels="2">
45                                         <label lblbgdcolor="green" lblboundlx="63" lblboundly="18" lblboundux="16"
   lblbounduy="23" lblname="Name">
46                                       </label>
```

```
47                                                  <label lblbgdcolor="red" lblboundlx="63" lblboundly="18" lblboundux="18"
    lblbounduy="81" lblname="Gender">
48                                                  </label>
49                                                  </labels-multMultiAsp>
50                                                </aspectsOflabels>
51                                              </labels>
52                                              <buttons>
53                                                <aspectsOfbuttons>
54                                                  <buttons-multMultiAsp numContainedInbuttons="1">
55                                                    <button buttonboundlx="44" buttonboundly="23" buttonboundux="330"
    buttonbounduy="207" buttonname="Done">
56                                                  </button>
57                                                  </buttons-multMultiAsp>
58                                                </aspectsOfbuttons>
59                                              </buttons>
60                                              <radiobuttons>
61                                                <aspectsOfradiobuttons>
62                                                  <radiobuttons-multMultiAsp numContainedInradiobuttons="2">
63                                                    <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="79" rdbuttonname="Male">
64                                                  </rdbutton>
65                                                    <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="101" rdbuttonname="Female">
66                                                  </rdbutton>
67                                                  </radiobuttons-multMultiAsp>
68                                                </aspectsOfradiobuttons>
69                                              </radiobuttons>
70                                            </components-compwidgetsDec>
71                                          </aspectsOfcomponents>
72                                        </components>
73                                      </absolute-absolutecompDec>
74                                    </aspectsOfabsolute>
75                                  </absolute>
76                                </layout-layouttypeSpec>
77                              </layout>
78                            </composite-drawableareaSpec>
79                          </composite>
80                        </tabitem-structureSpec>
81                      </tabitem>
82                    </tabFolder-multMultiAsp>
83                  </aspectsOftabFolder>
84                </tabFolder>
85              </shell-designSpec>
86            </shell>
87
```

APPENDIX 4 SimpleGUI2.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- edited with XMLSpy v2007 rel. 3 (http://www.altova.com) by lahiru (Acims)-->
3  <shell height="300" name="SimpleGUI2" width="400">
4     <shell-designSpec>
5        <tabFolder tablx="392" tably="266" tabux="0" tabuy="0">
6           <aspectsOftabFolder>
7              <tabFolder-multMultiAsp numContainedIntabFolder="3">
8                 <tabitem tabname="Personal">
9                    <tabitem-structureSpec>
10                      <composite>
11                         <composite-drawableareaSpec>
12                            <layout>
13                               <layout-layouttypeSpec>
14                                  <absolute>
15                                     <aspectsOfabsolute>
16                                        <absolute-absolutecompDec>
17                                           <components>
18                                              <aspectsOfcomponents>
19                                                 <components-compwidgetsDec>
20                                                    <checkbuttons>
21                                                       <aspectsOfcheckbuttons>
22                                                          <checkbuttons-multMultiAsp numContainedIncheckbuttons="0"/>
23                                                       </aspectsOfcheckbuttons>
24                                                    </checkbuttons>
25                                                    <textboxes>
26                                                       <aspectsOftextboxes>
27                                                          <textboxes-multMultiAsp numContainedIntextboxes="1">
28                                                             <textbox textbgdcolor="\white" textboundlx="204" textboundly="29"
    textboundux="111" textbounduy="20">
29                                                          </textbox>
30                                                          </textboxes-multMultiAsp>
31                                                       </aspectsOftextboxes>
32                                                    </textboxes>
33                                                    <scales>
34                                                       <aspectsOfscales>
35                                                          <scales-multMultiAsp numContainedInscales="0"/>
36                                                       </aspectsOfscales>
37                                                    </scales>
38                                                    <comboboxes>
39                                                       <aspectsOfcomboboxes>
40                                                          <comboboxes-multMultiAsp numContainedIncomboboxes="0"/>
41                                                       </aspectsOfcomboboxes>
42                                                    </comboboxes>
43                                                    <labels>
44                                                       <aspectsOflabels>
45                                                          <labels-multMultiAsp numContainedInlabels="2">
46                                                             <label lblbgdcolor="green" lblboundlx="63" lblboundly="18" lblboundux="16
    lblbounduy="23" lblname="Name">
```

```
47                                                  </label>
48                                                    <label lblbgdcolor="red" lblboundlx="63" lblboundly="18" lblboundux="16"
    lblbounduy="81" lblname="Gender">
49                                                  </label>
50                                                  </labels-multMultiAsp>
51                                                </aspectsOflabels>
52                                              </labels>
53                                              <buttons>
54                                                <aspectsOfbuttons>
55                                                  <buttons-multMultiAspnumContainedInbuttons="1">
56                                                    <button buttonboundlx="44" buttonboundly="23" buttonboundux="330"
    buttonbounduy="207" buttonname="Done">
57                                                  </button>
58                                                  </buttons-multMultiAsp>
59                                                </aspectsOfbuttons>
60                                              </buttons>
61                                              <radiobuttons>
62                                                <aspectsOfradiobuttons>
63                                                  <radiobuttons-multMultiAspnumContainedInradiobuttons="2">
64                                                    <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="79" rdbuttonname="Male">
65                                                  </rdbutton>
66                                                    <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="101" rdbuttonname="Female">
67                                                  </rdbutton>
68                                                  </radiobuttons-multMultiAsp>
69                                                </aspectsOfradiobuttons>
70                                              </radiobuttons>
71                                            </components-compwidgetsDec>
72                                          </aspectsOfcomponents>
73                                        </components>
74                                      </absolute-absolutecompDec>
75                                    </aspectsOfabsolute>
76                                  </absolute>
77                                </layout-layouttypeSpec>
78                              </layout>
79                            </composite-drawableareaSpec>
80                          </composite>
81                        </tabitem-structureSpec>
82                      </tabitem>
83                      <tabitem tabname="Education">
84                        <tabitem-structureSpec>
85                          <composite>
86                            <composite-drawableareaSpec>
87                              <layout>
88                                <layout-layouttypeSpec>
89                                  <absolute>
90                                    <aspectsOfabsolute>
```

```
 91                                  <absolute-absolutecompDec>
 92                                    <components>
 93                                      <aspectsOfcomponents>
 94                                        <components-compwidgetsDec>
 95                                          <checkbuttons>
 96                                            <aspectsOfcheckbuttons>
 97                                              <checkbuttons-multMultiAspnumContainedIncheckbuttons="0"/>
 98                                            </aspectsOfcheckbuttons>
 99                                          </checkbuttons>
100                                          <textboxes>
101                                            <aspectsOftextboxes>
102                                              <textboxes-multMultiAspnumContainedIntextboxes="1">
103                                                <textbox textbgdcolor="white" textboundlx="204" textboundly="29"
    textboundux="111" textbounduy="20">
104                                              </textbox>
105                                              </textboxes-multMultiAsp>
106                                            </aspectsOftextboxes>
107                                          </textboxes>
108                                          <scales>
109                                            <aspectsOfscales>
110                                              <scales-multMultiAspnumContainedInscales="0"/>
111                                            </aspectsOfscales>
112                                          </scales>
113                                          <comboboxes>
114                                            <aspectsOfcomboboxes>
115                                              <comboboxes-multMultiAspnumContainedIncomboboxes="0"/>
116                                            </aspectsOfcomboboxes>
117                                          </comboboxes>
118                                          <labels>
119                                            <aspectsOflabels>
120                                              <labels-multMultiAspnumContainedInlabels="2">
121                                                <label lblbgdcolor="green" lblboundlx="63" lblboundly="18" lblboundux="16
     lblbounduy="23" lblname="Details">
122                                              </label>
123                                                <label lblbgdcolor="red" lblboundlx="63" lblboundly="18" lblboundux="16"
    lblbounduy="81" lblname="Degree">
124                                              </label>
125                                              </labels-multMultiAsp>
126                                            </aspectsOflabels>
127                                          </labels>
128                                          <buttons>
129                                            <aspectsOfbuttons>
130                                              <buttons-multMultiAspnumContainedInbuttons="1">
131                                                <button buttonboundlx="44" buttonboundly="23" buttonboundux="330"
    buttonbounduy="207" buttonname="Done">
132                                              </button>
133                                              </buttons-multMultiAsp>
134                                            </aspectsOfbuttons>
```

```
135                                        </buttons>
136                                        <radiobuttons>
137                                          <aspectsOfradiobuttons>
138                                            <radiobuttons-multMultiAspnumContainedInradiobuttons="2">
139                                              <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="79" rdbuttonname="BS">
140                                              </rdbutton>
141                                              <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="101" rdbuttonname="PostGrad">
142                                              </rdbutton>
143                                            </radiobuttons-multMultiAsp>
144                                          </aspectsOfradiobuttons>
145                                        </radiobuttons>
146                                      </components-compwidgetsDec>
147                                    </aspectsOfcomponents>
148                                  </components>
149                                </absolute-absolutecompDec>
150                              </aspectsOfabsolute>
151                            </absolute>
152                          </layout-layouttypeSpec>
153                        </layout>
154                      </composite-drawableareaSpec>
155                    </composite>
156                  </tabitem-structureSpec>
157                </tabitem>
158                <tabitem tabname="Contact">
159                  <tabitem-structureSpec>
160                    <composite>
161                      <composite-drawableareaSpec>
162                        <layout>
163                          <layout-layouttypeSpec>
164                            <absolute>
165                              <aspectsOfabsolute>
166                                <absolute-absolutecompDec>
167                                  <components>
168                                    <aspectsOfcomponents>
169                                      <components-compwidgetsDec>
170                                        <checkbuttons>
171                                          <aspectsOfcheckbuttons>
172                                            <checkbuttons-multMultiAspnumContainedIncheckbuttons="0"/>
173                                          </aspectsOfcheckbuttons>
174                                        </checkbuttons>
175                                        <textboxes>
176                                          <aspectsOftextboxes>
177                                            <textboxes-multMultiAsp numContainedIntextboxes="1">
178                                              <textbox textbgdcolor="white" textboundlx="204" textboundly="29"
    textboundux="111" textbounduy="20">
179                                              </textbox>
```

```
180                                          </textboxes-multMultiAsp>
181                                        </aspectsOftextboxes>
182                                      </textboxes>
183                                      <scales>
184                                        <aspectsOfscales>
185                                          <scales-multMultiAsp numContainedInscales="0"/>
186                                        </aspectsOfscales>
187                                      </scales>
188                                      <comboboxes>
189                                        <aspectsOfcomboboxes>
190                                          <comboboxes-multMultiAsp numContainedIncomboboxes="0"/>
191                                        </aspectsOfcomboboxes>
192                                      </comboboxes>
193                                      <labels>
194                                        <aspectsOflabels>
195                                          <labels-multMultiAsp numContainedInlabels="2">
196                                            <label lblbgdcolor="green" lblboundlx="63" lblboundly="18" lblboundux="16
    lblbounduy="23" lblname="Details">
197                                            </label>
198                                            <label lblbgdcolor="red" lblboundlx="63" lblboundly="18" lblboundux="16"
    lblbounduy="81" lblname="Preference">
199                                            </label>
200                                          </labels-multMultiAsp>
201                                        </aspectsOflabels>
202                                      </labels>
203                                      <buttons>
204                                        <aspectsOfbuttons>
205                                          <buttons-multMultiAsp numContainedInbuttons="1">
206                                            <button buttonboundlx="44" buttonboundly="23" buttonboundux="330"
    buttonbounduy="207" buttonname="Done">
207                                            </button>
208                                          </buttons-multMultiAsp>
209                                        </aspectsOfbuttons>
210                                      </buttons>
211                                      <radiobuttons>
212                                        <aspectsOfradiobuttons>
213                                          <radiobuttons-multMultiAsp numContainedInradiobuttons="2">
214                                            <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="79" rdbuttonname="Phone">
215                                            </rdbutton>
216                                            <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="101" rdbuttonname="Mail">
217                                            </rdbutton>
218                                          </radiobuttons-multMultiAsp>
219                                        </aspectsOfradiobuttons>
220                                      </radiobuttons>
221                                    </components-compwidgetsDe
222                                  </aspectsOfcomponents>
```

```
223                                    </components>
224                                  </absolute-absolutecompDec>
225                                </aspectsOfabsolute>
226                              </absolute>
227                            </layout-layouttypeSpec>
228                          </layout>
229                        </composite-drawableareaSpec>
230                      </composite>
231                    </tabitem-structureSpec>
232                  </tabitem>
233                </tabFolder-multMultiAsp>
234              </aspectsOftabFolder>
235            </tabFolder>
236          </shell-designSpec>
237        </shell>
238
```

APPENDIX 5 ExGUI3.xml

```
1  <?xml version='1.0' encoding='UTF-8'?>
2  <shell height = "400" name = "ExGUI3" width = "500">
3  <shell-designSpec>
4     <tabFolder tablx = "392" tably = "266" tabux = "0" tabuy = "0">
5        <aspectsOftabFolder>
6           <tabFolder-multMultiAsp numContainedIntabFolder= "3">
7              <tabitem tabname= "Personal'>
8                 <tabitem-structureSpec>
9
10                   <Composite>
11                      <Composite-drawableareaSpec>
12                         <Layout>
13                            <Layout-layouttypeSpec>
14                               <absolute>
15                                  <aspectsOfabsolute>
16                                     <absolute-componentDec>
17
18                                     <checkbuttons>
19                                                <aspectsOfcheckbuttons>
20                                                   <checkbuttons-multMultiAspnumContainedIncheckbuttons="0"/>
21                                                </aspectsOfcheckbuttons>
22                                     </checkbuttons>
23                                        <textboxes>
24                                           <aspectsOftextboxes>
25                                              <textboxes-multMultiAsp numContainedIntextboxes= "1">
26                                                 <textbox textboundlx = "204" textboundly = "29" textboundux = "111" textbounduy = "
   20" textbgdcolor = "white">
27                                                 </textbox>
28                                              </textboxes-multMultiAsp>
29                                           </aspectsOftextboxes>
30                                        </textboxes>
31
32
33 <comboboxes>
34                                           <aspectsOfcomboboxes>
35                                              <comboboxes-multMultiAsp numContainedIncomboboxes= "1">
36                                                 <combobox comboboundlx = "62" comboboundly = "21" comboboundux = "111"
   comboboundy = "206">                                  </combobox>
37                                              </comboboxes-multMultiAsp>
38                                           </aspectsOfcomboboxes>
39                                        </comboboxes>                                  <labels>
40                                           <aspectsOflabels>
41                                              <labels-multMultiAsp numContainedInlabels= "4">
42 <label lblbgdcolor = "green" lblboundlx = "63" lblboundly = "18" lblboundux = "16" lblbounduy = "160" lblname = "Age">      </
   label>
43                                                 <label lblbgdcolor = "green" lblboundlx = "63" lblboundly = "18" lblboundux = "16"
   lblbounduy = "23" lblname = "Name">                 </label>
44                                 <label lblbgdcolor = "red" lblboundlx = "63" lblboundly = "18" lblboundux = "16" lblbounduy = "81"
```

```
    lblname = "Gender">          </label>
45                      <label  lblbgdcolor = "red" lblboundlx = "63" lblboundly = "18" lblbounddux = "16" lblbounduy = "200"
    lblname = "Country">         </label>
46
47                                          </labels-multMultiAsp>
48                                       </aspectsOflabels>
49                                    </labels>
50                                    <buttons>
51                                       <aspectsOfbuttons>
52                                          <buttons-multMultiAsp numContainedInbuttons = "1">
53                                             <button  buttonboundlx = "44" buttonboundly = "23" buttonbounddux = "330"
    buttonbounduy = "207" buttonname = "Done">
54                                             </button>
55                                          </buttons-multMultiAsp>
56                                       </aspectsOfbuttons>
57                                    </buttons>
58                                    <radiobuttons>
59                                       <aspectsOfradiobuttons>
60                                          <radiobuttons-multMultiAsp numContainedInradiobuttons= "2">
61                                             <rdbutton  rdbuttonboundlx = "83" rdbuttonboundly = "16" rdbuttonbounddux = "111"
    rdbuttonbounduy = "79" rdbuttonname = "Male"> </rdbutton>
62
63  <rdbutton  rdbuttonboundlx = "83" rdbuttonboundly = "16" rdbuttonbounddux = "111" rdbuttonbounduy= "101" rdbuttonname = "
    Female">
64                                 </rdbutton>
65
66
67                                          </radiobuttons-multMultiAsp>
68                                       </aspectsOfradiobuttons>
69                                    </radiobuttons>
70
71
72
73                                    <scales>
74                                       <aspectsOfscales>
75                                          <scales-multMultiAsp numContainedInscales= "1">
76                                             <scale  sclbgdcolor = "" sclboundlx = "108" sclboundly = "50" sclbounddux = "111"
    sclbounduy = "155" sclsetincr = "4" sclsetmax = "20" sclsetmin = "1" sclsetpgincr = "5">
77                                             </scale>
78                                          </scales-multMultiAsp>
79                                       </aspectsOfscales>
80                                    </scales>
81                                 </absolute-componentDec>
82                              </aspectsOfabsolute>
83                           </absolute>
84
85                        </Layout-layouttypeSpec>
86                     </Layout>
```

```
87
88                    </Composite-drawableareaSpec>
89                  </Composite>
90                </tabitem-structureSpec>
91              </tabitem>
92
93      <tabitem tabname="Education">
94                  <tabitem-structureSpec>
95                    <composite>
96                      <composite-drawableareaSpec>
97                        <layout>
98                          <layout-layouttypeSpec>
99                            <absolute>
100                             <aspectsOfabsolute>
101                               <absolute-absolutecompDec>
102                                 <components>
103                                   <aspectsOfcomponents>
104                                     <components-compwidgetsDec>
105                                       <checkbuttons>
106                                         <aspectsOfcheckbuttons>
107                                           <checkbuttons-multMultiAspnumContainedIncheckbuttons="0"/>
108                                         </aspectsOfcheckbuttons>
109                                       </checkbuttons>
110                                       <textboxes>
111                                         <aspectsOftextboxes>
112                                           <textboxes-multMultiAspnumContainedIntextboxes="1">
113                                             <textbox textbgdcolor="white" textboundlx="204" textboundly="29"
    textboundux="111" textbounduy="20">
114                                           </textbox>
115                                           </textboxes-multMultiAsp>
116                                         </aspectsOftextboxes>
117                                       </textboxes>
118                                       <scales>
119                                         <aspectsOfscales>
120                                           <scales-multMultiAspnumContainedInscales="0"/>
121                                         </aspectsOfscales>
122                                       </scales>
123                                       <comboboxes>
124                                         <aspectsOfcomboboxes>
125                                           <comboboxes-multMultiAspnumContainedIncomboboxes="0"/>
126                                         </aspectsOfcomboboxes>
127                                       </comboboxes>
128                                       <labels>
129                                         <aspectsOflabels>
130                                           <labels-multMultiAspnumContainedInlabels="2">
131                                             <label lblbgdcolor="green" lblboundlx="63" lblboundly="18" lblboundux="16
    lblbounduy="23" lblname="Details">
132                                           </label>
```

```
133                                                    <label lblbgdcolor="red" lblboundlx="63" lblboundly="18" lblboundux="16"
    lblbounduy="81" lblname="Degree">
134                                                    </label>
135                                                    </labels-multMultiAsp>
136                                                  </aspectsOflabels>
137                                                </labels>
138                                                <buttons>
139                                                  <aspectsOfbuttons>
140                                                    <buttons-multMultiAspnumContainedInbuttons="1">
141                                                      <button buttonboundlx="44" buttonboundly="23" buttonboundux="330"
    buttonbounduy="207" buttonname="Done">
142                                                    </button>
143                                                    </buttons-multMultiAsp>
144                                                  </aspectsOfbuttons>
145                                                </buttons>
146                                                <radiobuttons>
147                                                  <aspectsOfradiobuttons>
148                                                    <radiobuttons-multMultiAspnumContainedInradiobuttons="2">
149                                                      <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="79" rdbuttonname="BS">
150                                                    </rdbutton>
151                                                      <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="101" rdbuttonname="PostGrad">
152                                                    </rdbutton>
153                                                    </radiobuttons-multMultiAsp>
154                                                  </aspectsOfradiobuttons>
155                                                </radiobuttons>
156                                              </components-compwidgetsDec>
157                                            </aspectsOfcomponents>
158                                          </components>
159                                        </absolute-absolutecompDec>
160                                      </aspectsOfabsolute>
161                                    </absolute>
162                                  </layout-layouttypeSpec>
163                                </layout>
164                              </composite-drawableareaSpec>
165                            </composite>
166                          </tabitem-structureSpec>
167                        </tabitem>
168
169    <tabitem>
170                          <tabitem-structureSpec>
171                            <composite>
172                              <composite-drawableareaSpec>
173                                <layout>
174                                  <layout-layouttypeSpec>
175                                    <absolute>
176                                      <aspectsOfabsolute>
```

```
177                                 <absolute-absolutecompDec>
178                                   <components>
179                                     <aspectsOfcomponents>
180                                       <components-compwidgetsDec>
181                                         <checkbuttons>
182                                           <aspectsOfcheckbuttons>
183                                             <checkbuttons-multMultiAspnumContainedIncheckbuttons="0"/>
184                                           </aspectsOfcheckbuttons>
185                                         </checkbuttons>
186                                         <textboxes>
187                                           <aspectsOftextboxes>
188                                             <textboxes-multMultiAspnumContainedIntextboxes="1">
189                                               <textbox textbgdcolor="white" textboundlx="204" textboundly="29"
    textboundux="111" textbounduy="20">
190                                             </textbox>
191                                             </textboxes-multMultiAsp>
192                                           </aspectsOftextboxes>
193                                         </textboxes>
194                                         <scales>
195                                           <aspectsOfscales>
196                                             <scales-multMultiAspnumContainedInscales="0"/>
197                                           </aspectsOfscales>
198                                         </scales>
199                                         <comboboxes>
200                                           <aspectsOfcomboboxes>
201                                             <comboboxes-multMultiAspnumContainedIncomboboxes="0"/>
202                                           </aspectsOfcomboboxes>
203                                         </comboboxes>
204                                         <labels>
205                                           <aspectsOflabels>
206                                             <labels-multMultiAspnumContainedInlabels="2">
207                                               <label lblbgdcolor="green" lblboundlx="63" lblboundly="18" lblboundux="16
    lblbounduy="23" lblname="Details">
208                                             </label>
209                                               <label lblbgdcolor="red" lblboundlx="63" lblboundly="18" lblboundux="16"
    lblbounduy="81" lblname="Preference">
210                                             </label>
211                                             </labels-multMultiAsp>
212                                           </aspectsOflabels>
213                                         </labels>
214                                         <buttons>
215                                           <aspectsOfbuttons>
216                                             <buttons-multMultiAspnumContainedInbuttons="1">
217                                               <button buttonboundlx="44" buttonboundly="23" buttonboundux="330"
    buttonbounduy="207" buttonname="Done">
218                                             </button>
219                                             </buttons-multMultiAsp>
220                                           </aspectsOfbuttons>
```

```
221                                          </buttons>
222                                          <radiobuttons>
223                                            <aspectsOfradiobuttons>
224                                              <radiobuttons-multMultiAspnumContainedInradiobuttons="2">
225                                                <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="79" rdbuttonname="Phone">
226                                                </rdbutton>
227                                                <rdbutton rdbuttonboundlx="83" rdbuttonboundly="16" rdbuttonboundux="
    111" rdbuttonbounduy="101" rdbuttonname="Mail">
228                                                </rdbutton>
229                                              </radiobuttons-multMultiAsp>
230                                            </aspectsOfradiobuttons>
231                                          </radiobuttons>
232                                        </components-compwidgetsDec>
233                                      </aspectsOfcomponents>
234                                    </components>
235                                  </absolute-absolutecompDec>
236                                </aspectsOfabsolute>
237                              </absolute>
238                            </layout-layouttypeSpec>
239                          </layout>
240                        </composite-drawableareaSpec>
241                      </composite>
242                    </tabitem-structureSpec>
243                  </tabitem>
244
245
246
247            </tabFolder-multMultiAsp>
248          </aspectsOftabFolder>
249        </tabFolder>
250
251    </shell-designSpec>
252    </shell>
253
```

APPENDIX 6 ExGUI4.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<shell height="300" name="ExGUI4" width="400">
  <shell-designSpec>
    <tabFolder tablx="0" tably="0" tabux="0" tabuy="0">
      <aspectsOftabFolder>
        <tabFolder-multMultiAsp numContainedIntabFolder="2">
          <tabitem tabname="Technology">
            <tabitem-structureSpec>
              <composite>
                <composite-drawableareaSpec>
                  <group>
                    <aspectsOfgroup>
                      <group-multMultiAsp numContainedIngroup="2">
                        <groupitem groupname="VLSI">
                          <aspectsOfgroupitem>
                            <groupitem-groupitemstrucDec>
                              <layout>
                                <layout-layouttypeSpec>
                                  <fill>
                                    <aspectsOffill>
                                      <fill-fillcompDec>
                                        <components>
                                          <aspectsOfcomponents>
                                            <components-compwidgetsDec>
                                              <checkbuttons>
                                                <aspectsOfcheckbuttons>
                                                  <checkbuttons-multMultiAspnumContainedIncheckbuttons="0'
/>
                                                </aspectsOfcheckbuttons>
                                              </checkbuttons>
                                              <textboxes>
                                                <aspectsOftextboxes>
                                                  <textboxes-multMultiAsp numContainedIntextboxes="0"/>
                                                </aspectsOftextboxes>
                                              </textboxes>
                                              <scales>
                                                <aspectsOfscales>
                                                  <scales-multMultiAsp numContainedInscales="0"/>
                                                </aspectsOfscales>
                                              </scales>
                                              <comboboxes>
                                                <aspectsOfcomboboxes>
                                                  <comboboxes-multMultiAspnumContainedIncomboboxes="0"
                                                </aspectsOfcomboboxes>
                                              </comboboxes>
                                              <labels>
                                                <aspectsOflabels>
                                                  <labels-multMultiAsp numContainedInlabels="0"/>
```

```
48                                                  </aspectsOflabels>
49                                                </labels>
50                                                <buttons>
51                                                  <aspectsOfbuttons>
52                                                    <buttons-multMultiAspnumContainedInbuttons="0"/>
53                                                  </aspectsOfbuttons>
54                                                </buttons>
55                                                <radiobuttons>
56                                                  <aspectsOfradiobuttons>
57                                                    <radiobuttons-multMultiAspnumContainedInradiobuttons="0"/>
58                                                  </aspectsOfradiobuttons>
59                                                </radiobuttons>
60                                              </components-compwidgetsDec>
61                                            </aspectsOfcomponents>
62                                          </components>
63                                        </fill-fillcompDec>
64                                      </aspectsOffill>
65                                    </fill>
66                                  </layout-layouttypeSpec>
67                                </layout>
68                              </groupitem-groupitemstrucDec>
69                            </aspectsOfgroupitem>
70                          </groupitem>
71                          <groupitem groupname="Wafer">
72                            <aspectsOfgroupitem>
73                              <groupitem-groupitemstrucDec>
74                                <layout>
75                                  <layout-layouttypeSpec>
76                                    <fill>
77                                      <aspectsOffill>
78                                        <fill-fillcompDec>
79                                          <components>
80                                            <aspectsOfcomponents>
81                                              <components-compwidgetsDec>
82                                                <checkbuttons>
83                                                  <aspectsOfcheckbuttons>
84                                                    <checkbuttons-multMultiAspnumContainedIncheckbuttons="0"
    />
85                                                  </aspectsOfcheckbuttons>
86                                                </checkbuttons>
87                                                <textboxes>
88                                                  <aspectsOftextboxes>
89                                                    <textboxes-multMultiAsp numContainedIntextboxes="0"/>
90                                                  </aspectsOftextboxes>
91                                                </textboxes>
92                                                <scales>
93                                                  <aspectsOfscales>
94                                                    <scales-multMultiAsp numContainedInscales="0"/>
```

```
ERROR: stackunderflow
OFFENDING COMMAND: ~

STACK:
```