



THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

Visual Modelling Language ---- Interface Modelling

Richard Li

September 2006



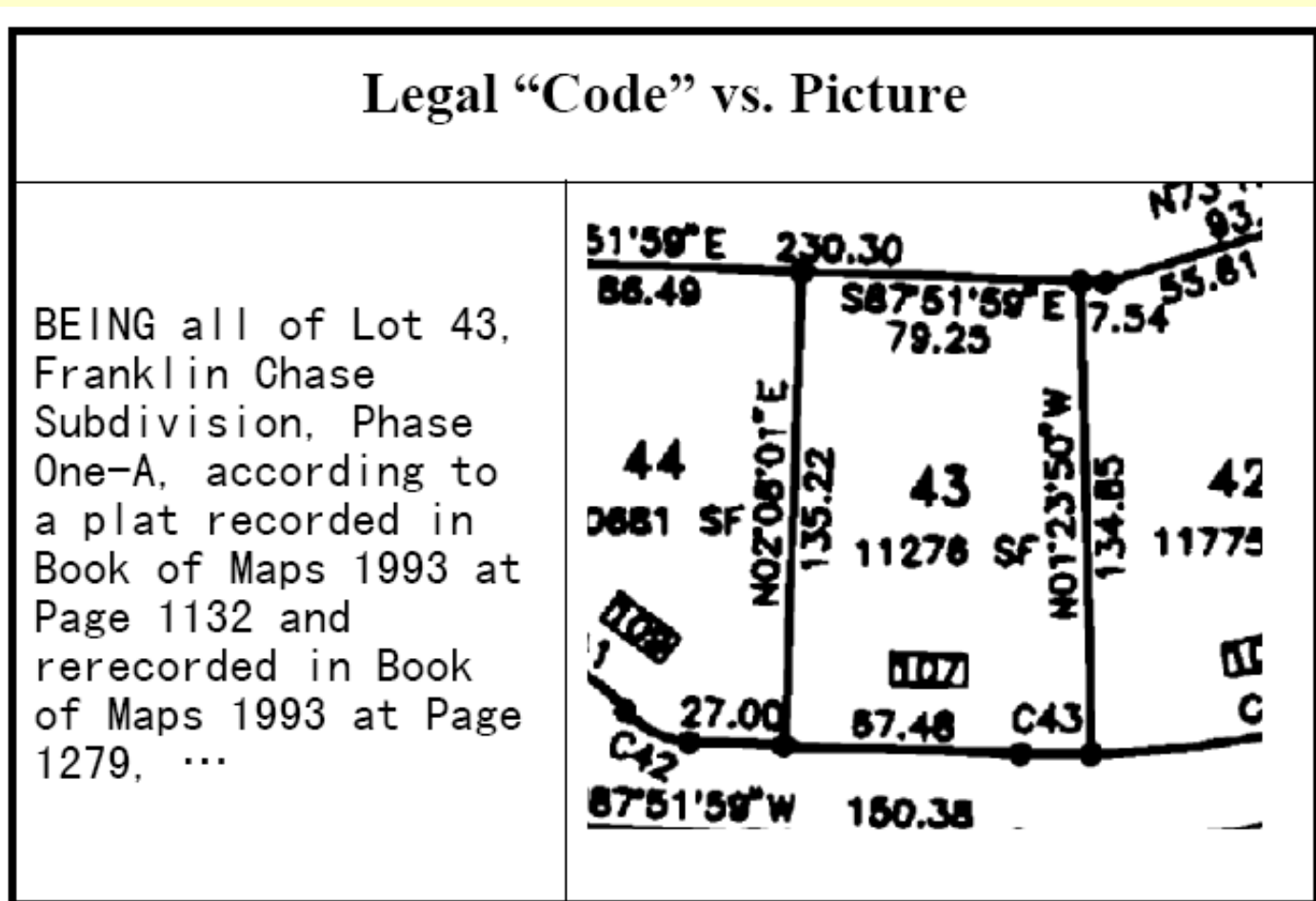
What is Visual Modelling Language

G "If you were supposed to understand it, we wouldn't call it code."

= Visual modeling languages use a diagram techniques with named symbols that represent concepts and lines that connect the symbols and that represent relationships and various other graphical annotation to represent constraints

Why???

Enhance Communication



Why???



Why???

Enable Planning and Reduce Risk

- = The two greatest risks to any software project are time and quality. Modeling can reduce both of these risks.
- = By using visual models for software projects, we can see the big picture for the project. We can organize overarching concepts into high level diagrams, which can be drilled into via a more fine-grained diagram.

Types of VML

Overall architecture of the system

- System dependencies,
- Business requirements
- Database organization and structure

Source code – including almost every aspect of object-oriented development

- Deployment configurations

GUI Modelling

Research Motivation

Interface Builders

Most of Popular Programming Languages have their own IDEs

PASCAL



Delphi

JAVA



J Builder

C++



Visual C++

BASIC



Visual Basic

- + Reduce the Amount of Code
- + Quickly Construct Graphical Interfaces
- + Achieve a Consistent Look and Feel

Research Motivation

Interface Builders

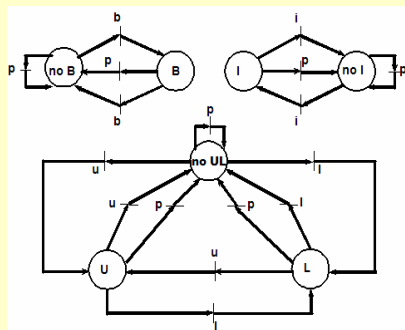
NO MODEL

Research Motivation

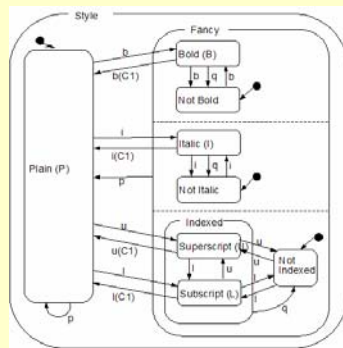
Dialogue Notations

Dialogue Models Provide a Behavioral Description for the User Interface

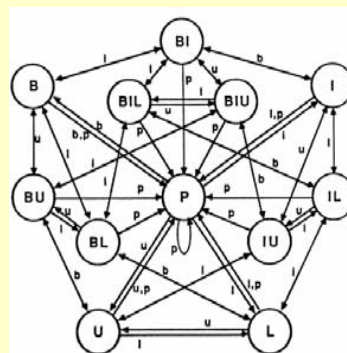
Petri Nets



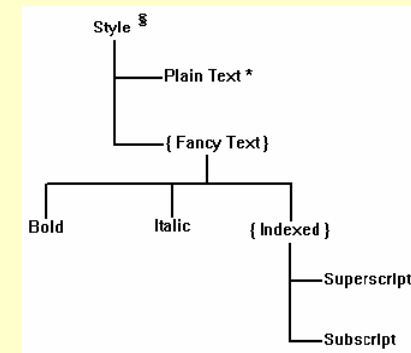
State Charts



STN



Lean Cuisine+



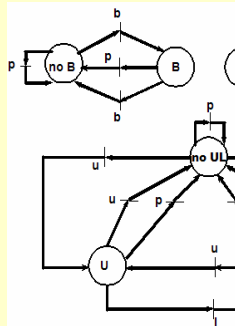
- + Using a Visual Approach to Abstract a High Level Structure for the GUI
- + Directly Executable; Easy to Understand; Not Constrain the Implementation
- + The Description Can be Subjected to Analysis

Research Motivation

Dialogue Notations

Dialogue

Petri N



+ Using a V

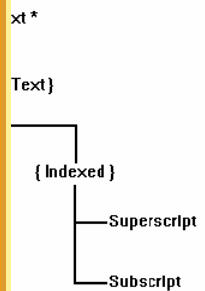
+ Directly I

+ The Desc



Interface

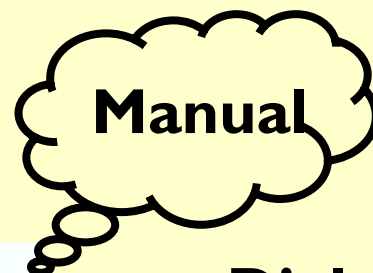
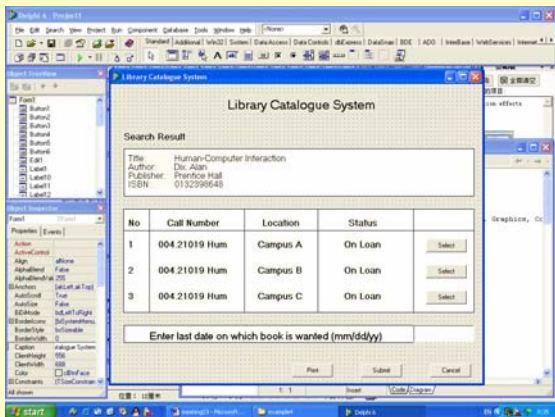
isine+



Jl

itation

Research Motivation

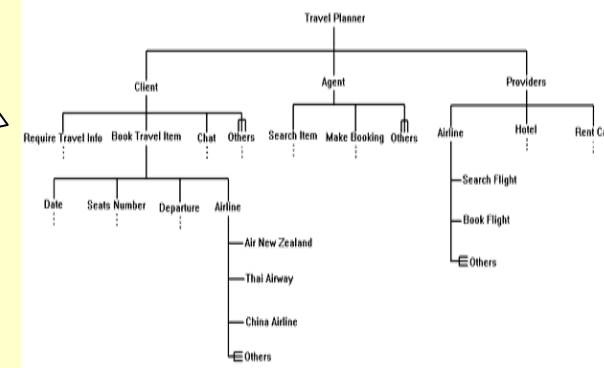


Interface Builders

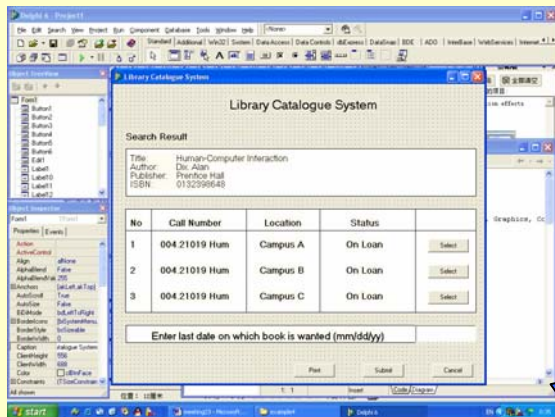


GUI Expert

Dialogue Notations



Research Motivation

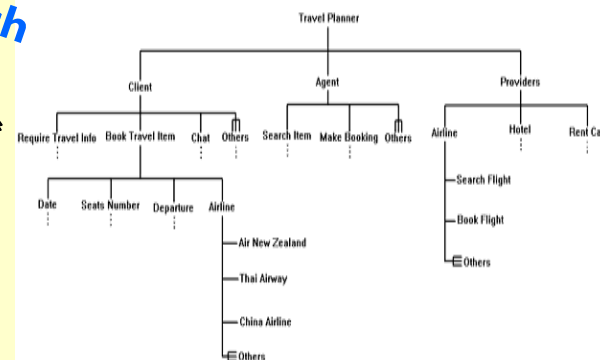


Interface Builders

Automatic Approach

GAP

Dialogue Notations



Research Perspective

Delphi

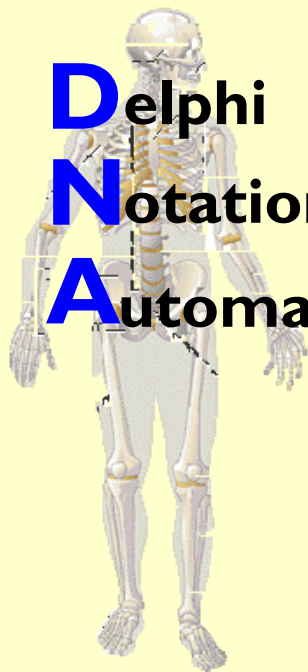
Notation

Automatic Approach

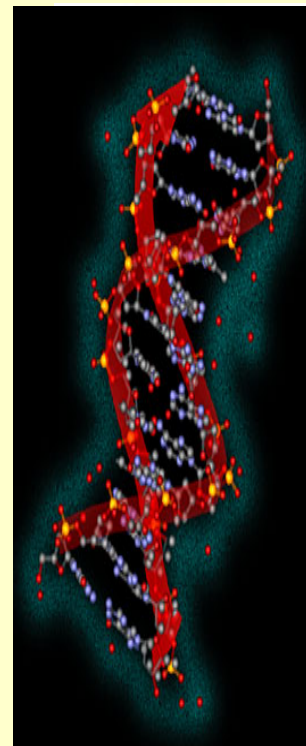
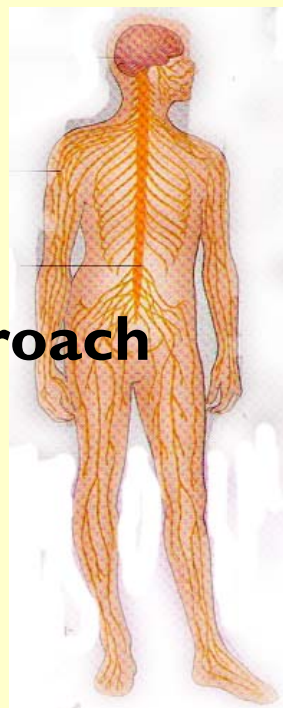
Interface Builder

Lean Cuisine+

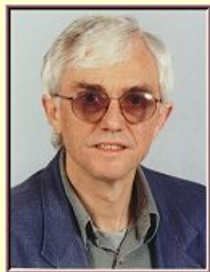
Methodology



Delphi
Notation
Automatic Approach



Lean Cuisine+ History



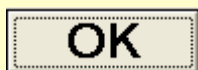
Lean Cuisine+ was developed by Dr Chris Philips at Massey University



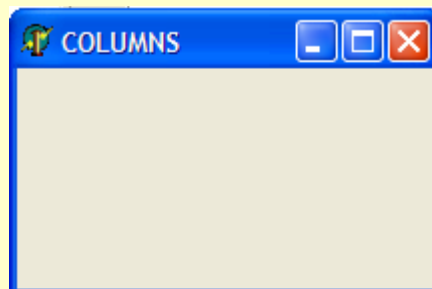
SELCU (Support Environment) has been built by Dr Chris Scogings at Massey University

Lean Cuisine+ Introduction

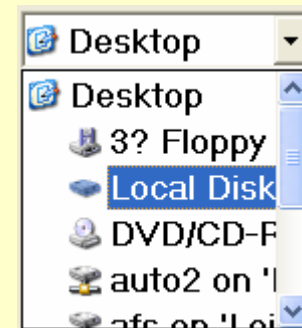
Interface Components --- Meneme



↓
OK



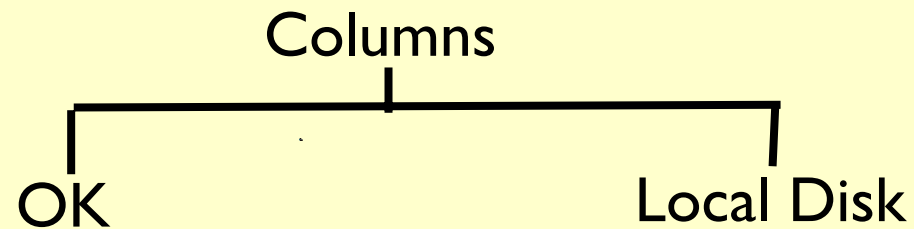
↓
Columns



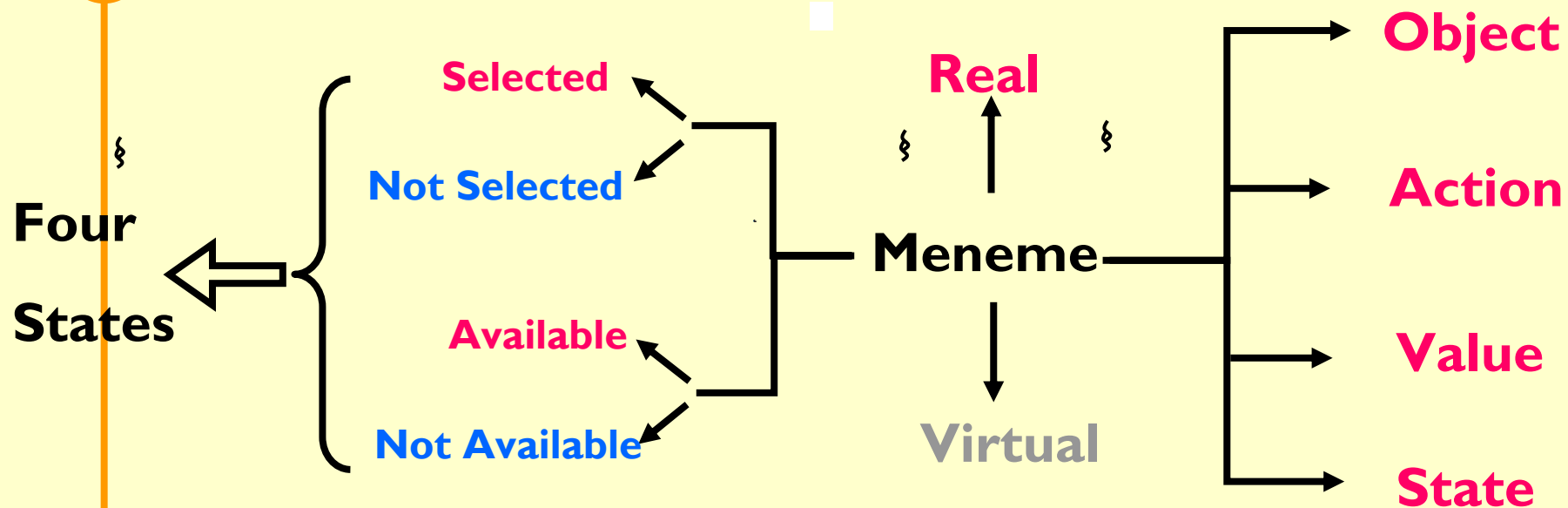
↓
Local Disk

Lean Cuisine+ Introduction

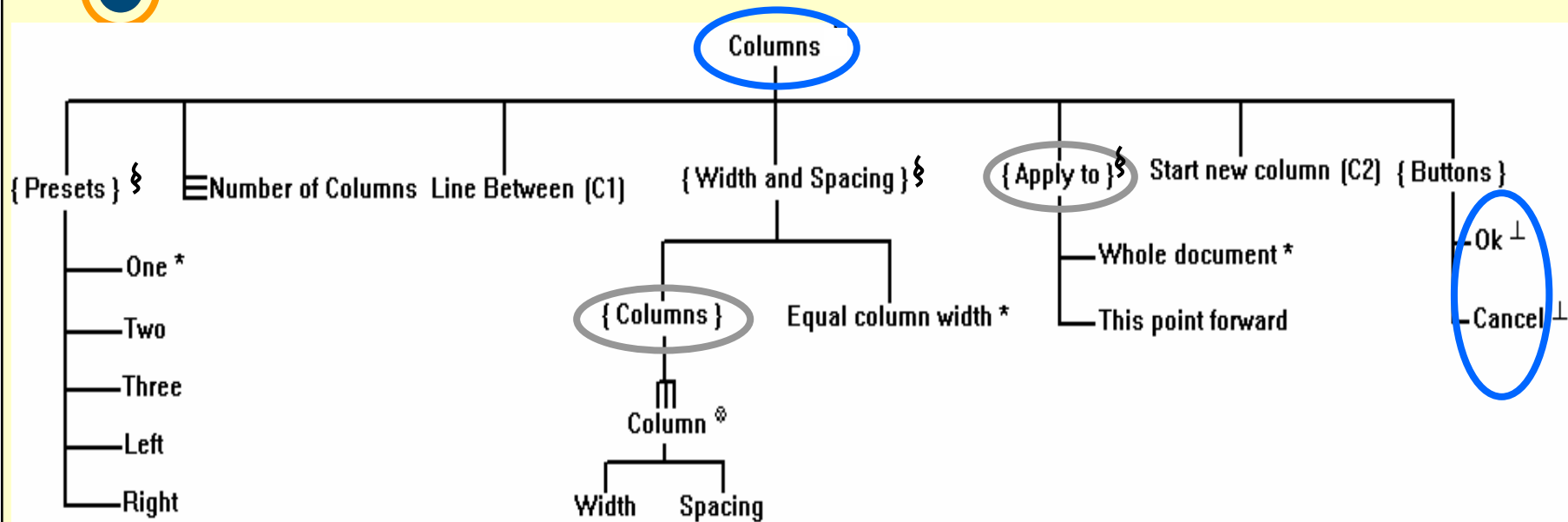
Tree Structure



Lean Cuisine+ Introduction



Lean Cuisine+ Introduction

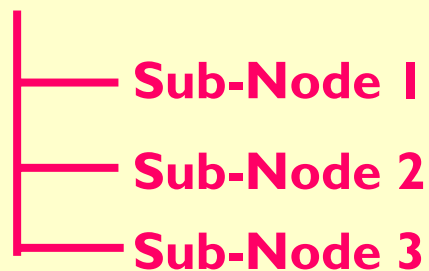


Lean Cuisine+ Introduction

Two Grouping Constraints on Behavior

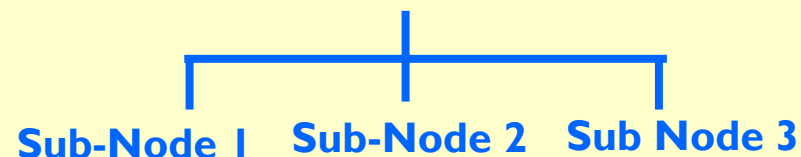
Mutually Exclusive
(1 from N)

Node



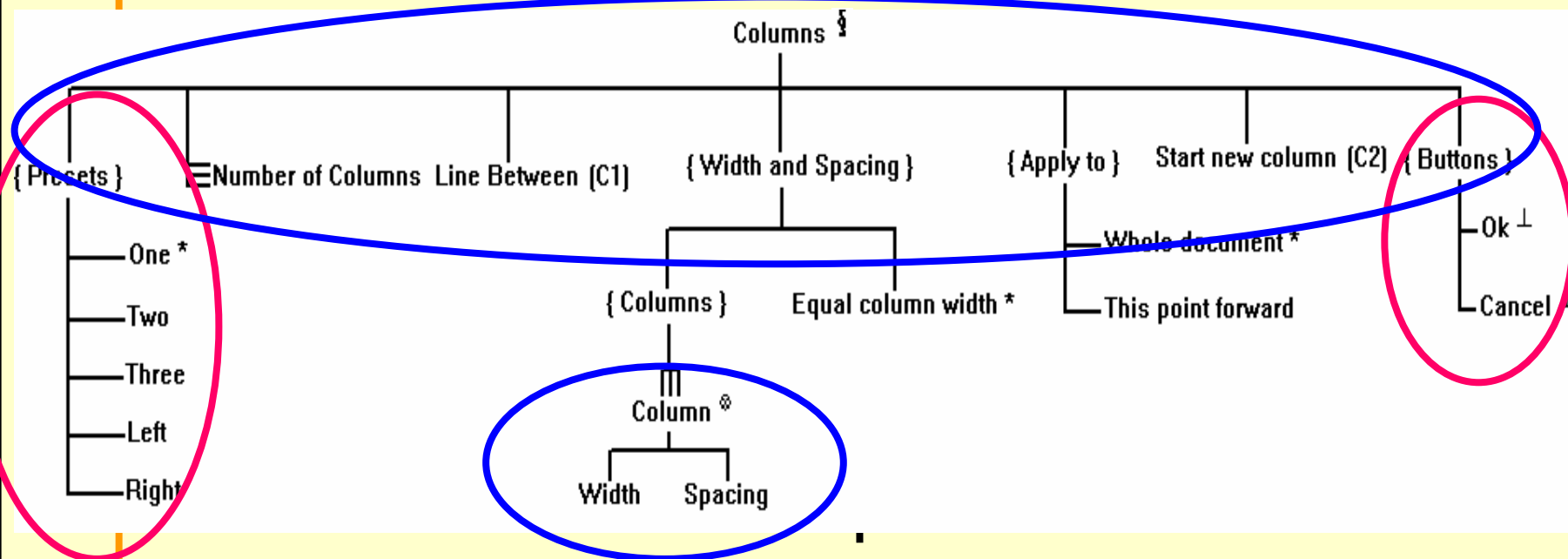
Mutually Compatible
(M from N)

Node



Lean Cuisine+ Introduction

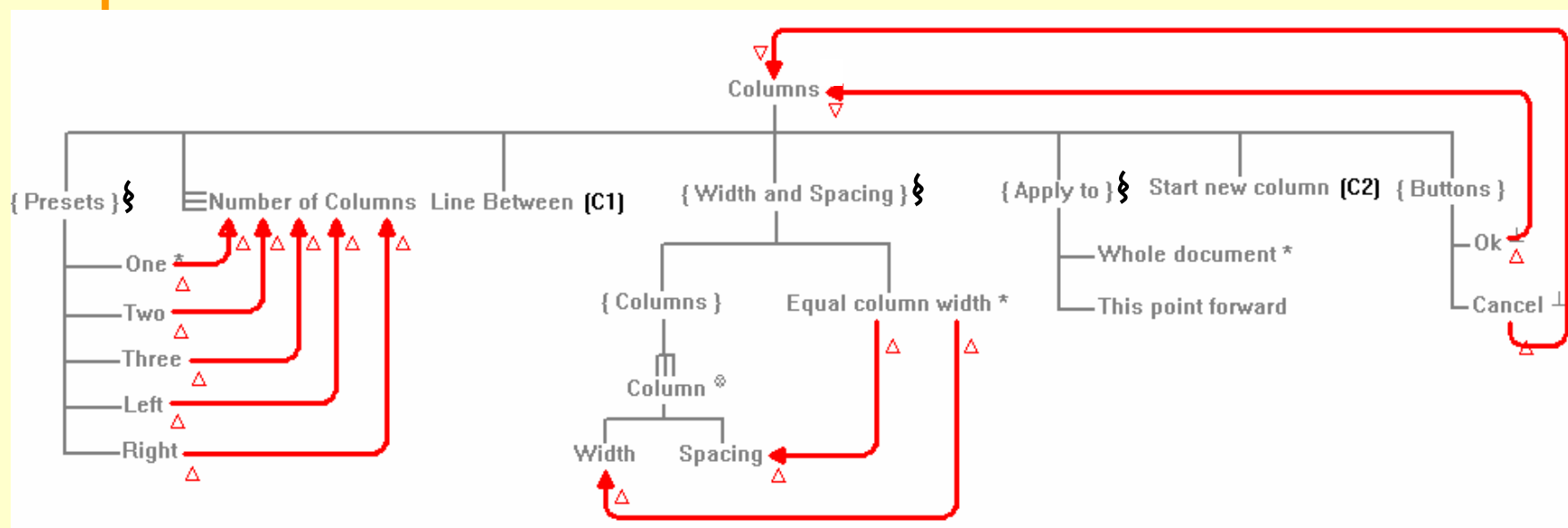
Two Grouping Constraints on Behavior



Lean Cuisine+ Introduction

Triggers

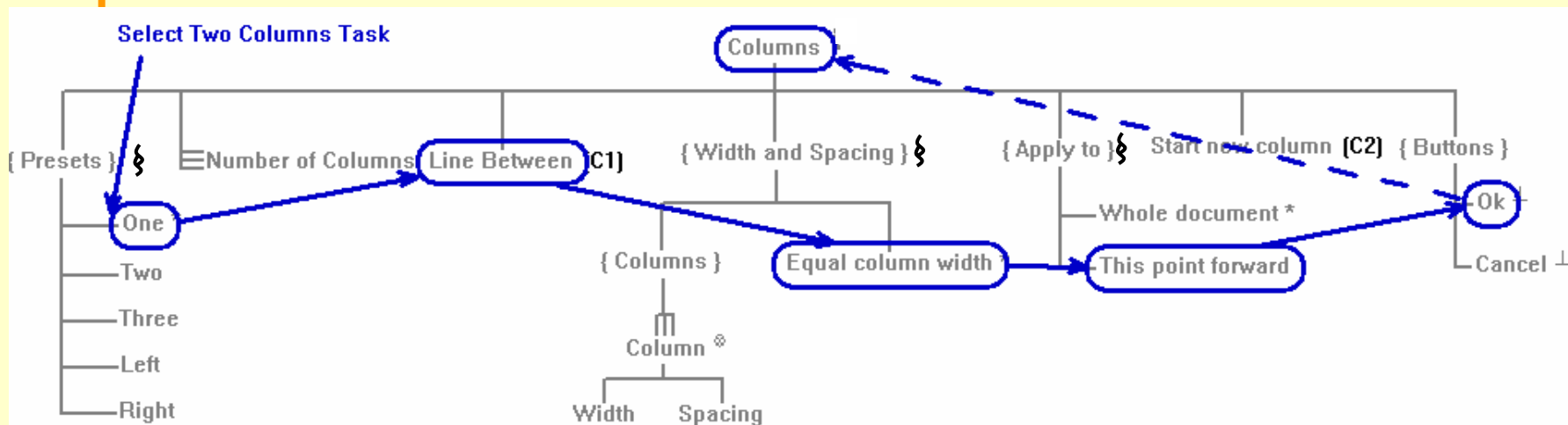
are used to show System Internal Selections



Lean Cuisine+ Introduction

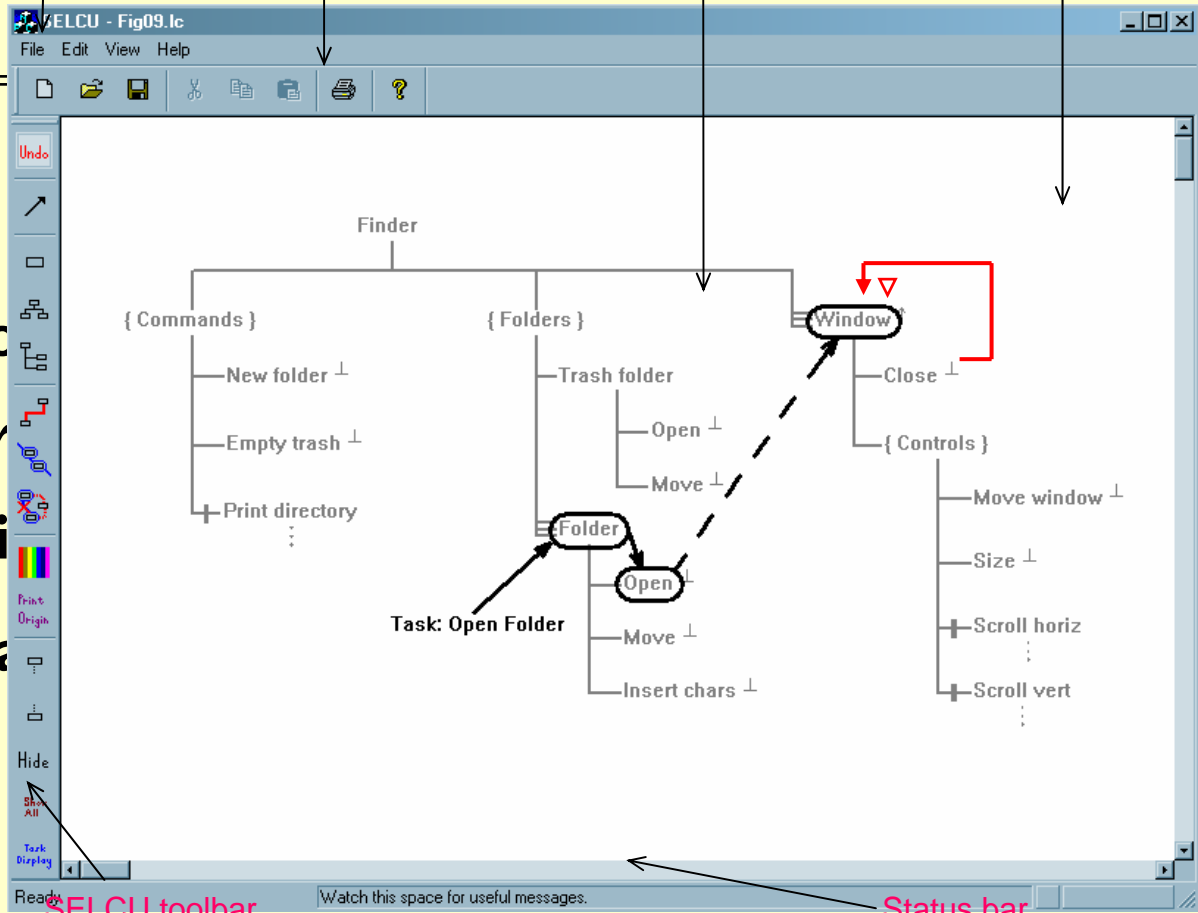
Tasks

- Represent as an **Overlay of Linked Menemes**
- Superimposed on the **Base Diagram**



SELCU

Menus File name Windows toolbar Lean Cuisine+ diagram Work area



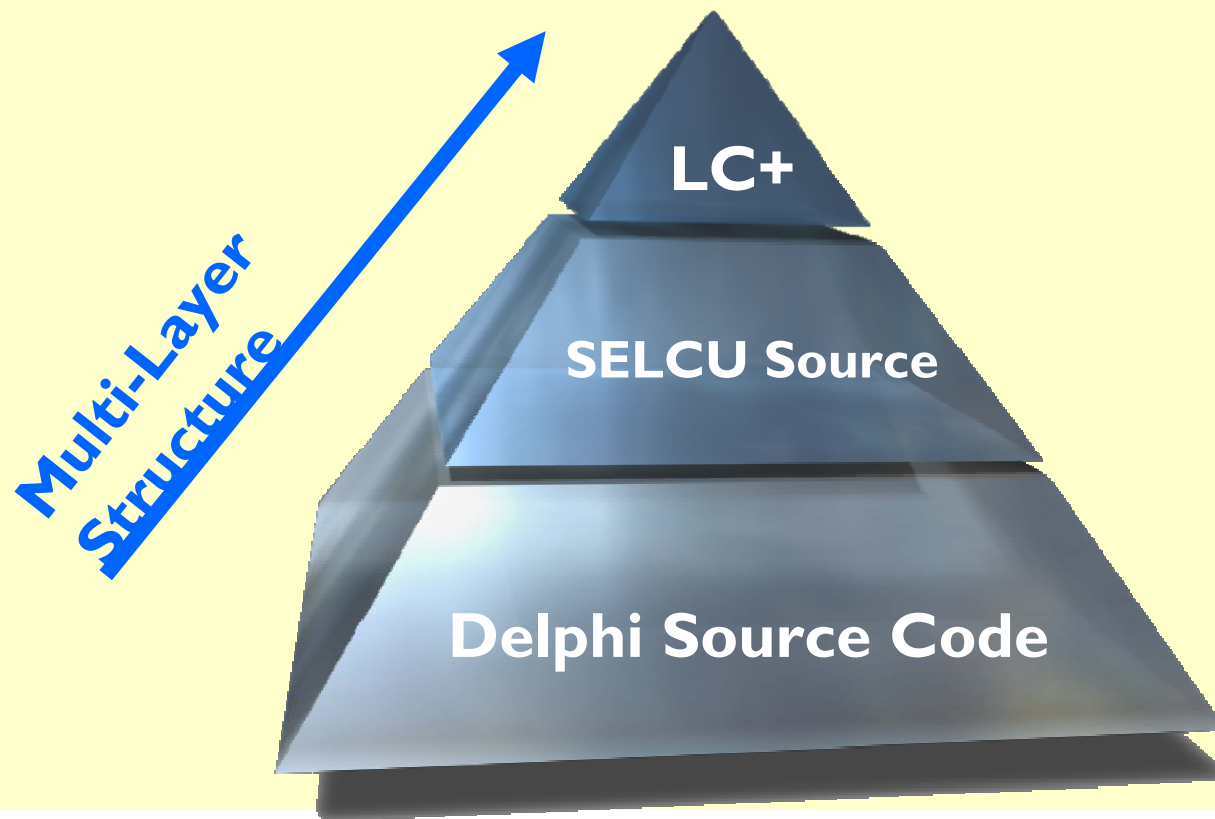
- Sc
- Pr
- Di
- Da

on LC+

SELCU toolbar

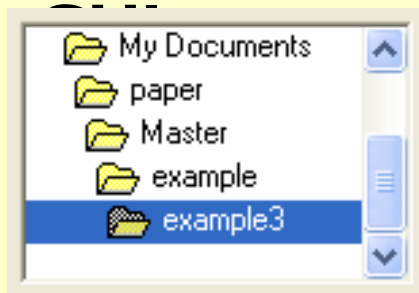
Status bar

Automatic Generation Structure

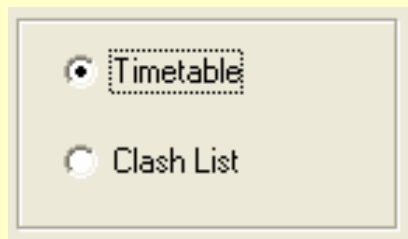


Automatic Generation Approach

Delphi

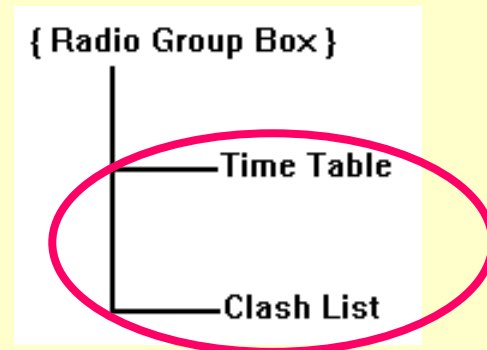
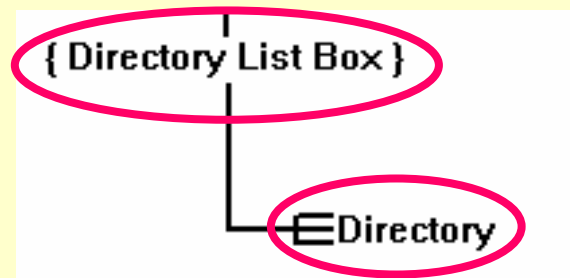


List Box / Combo Box



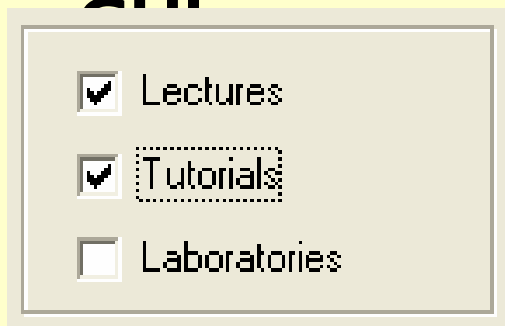
Radio Group

Lean Cuisine+ Model

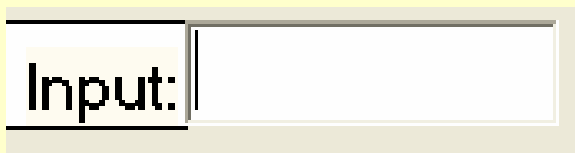


Automatic Generation Approach

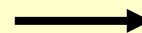
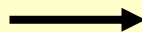
Delphi GUI



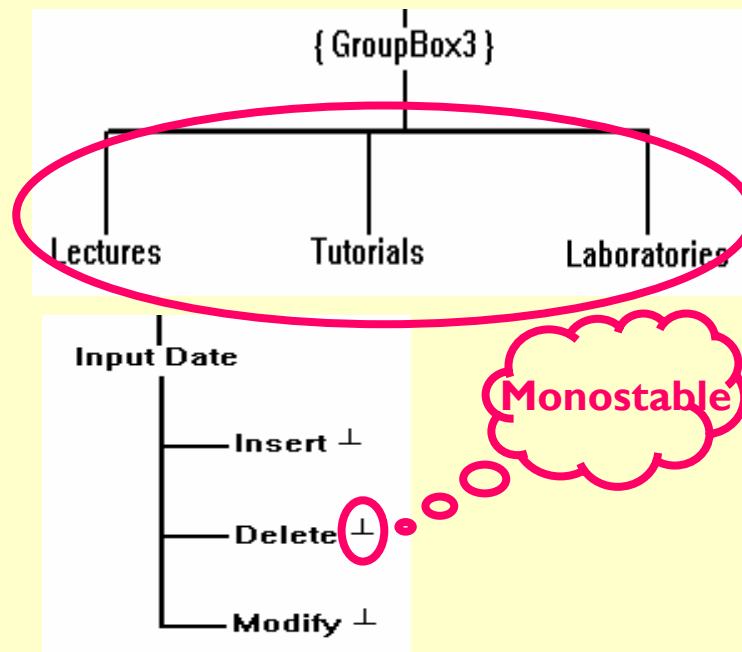
Check Box



Data Input Box



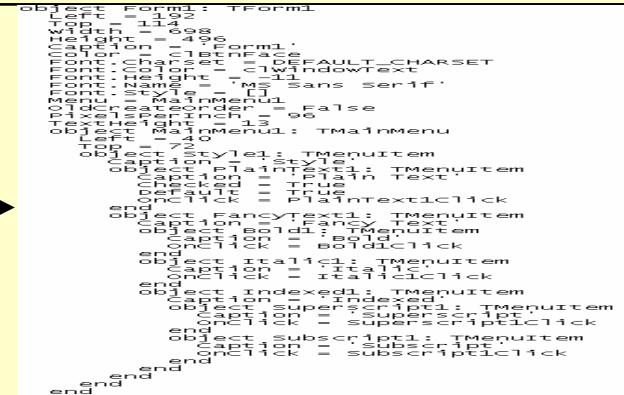
Lean Cuisine+ Model



Automatic Generation Approach



Delphi Interface



Delphi Source Code

- Using .dfm file to Generate Basic Tree Structure
- Using .pas file to Add Meneme Attributes and Logical Relationships

***.dfm File**

***.pas File**

Automatic Generation Approach

```

object style1: TMenuItem
  Caption = 'Style'
  object PlainText1: TMenuItem
    Caption = 'Plain Text'
    Checked = True
    default = True
    onclick = PlainText1Click
  end
  object FancyText1: TMenuItem
    Caption = 'Fancy Text'
    object Bold1: TMenuItem
      Caption = 'Bold'
      onclick = Bold1Click
    end
    object Italic1: TMenuItem
      Caption = 'Italic'
      onclick = Italic1Click
    end
  end
  object Indexed1: TMenuItem
    Caption = 'Indexed'
    object Superscript1: TMenuItem
      Caption = 'Superscript'
      onclick = Superscript1Click
    end
    object Subscript1: TMenuItem
      Caption = 'Subscript'
      onclick = Subscript1Click
    end
  end
end

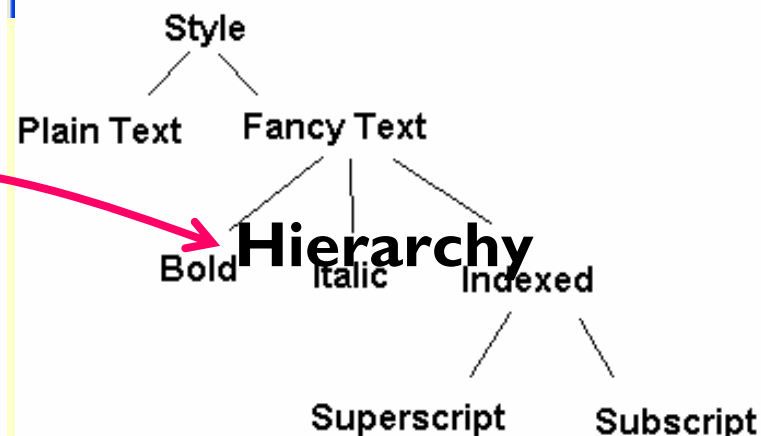
```

.dfm File

**Meneme
Name**

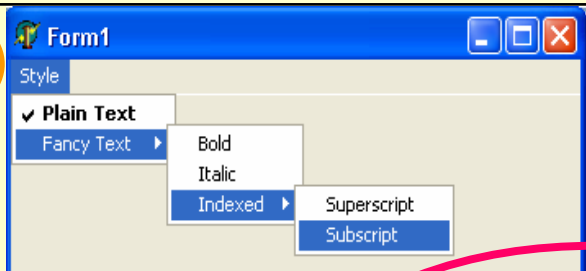
**System Default
Option**

Basic Tree Structure

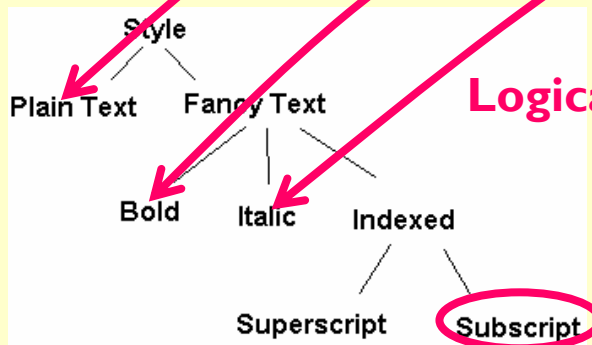


Hierarchy

Automatic Generation Approach

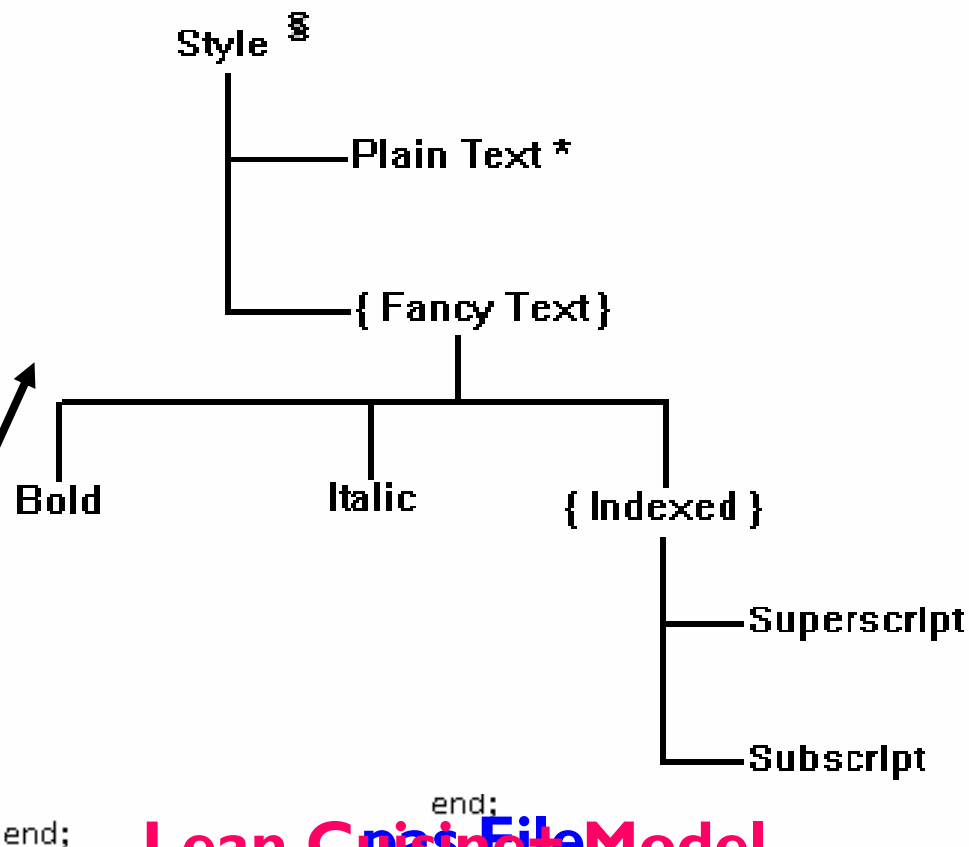


Delphi Interface



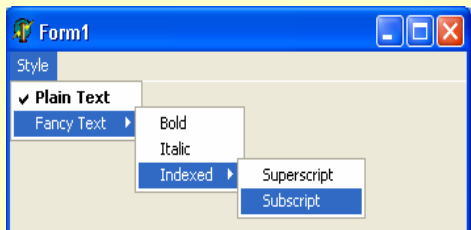
Basic Tree Structure

Logical R



Lean Code File Model

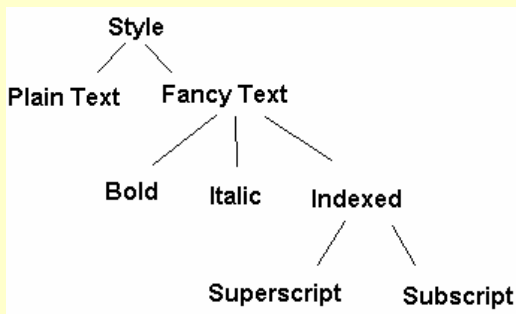
Automatic Generation Process Sequence



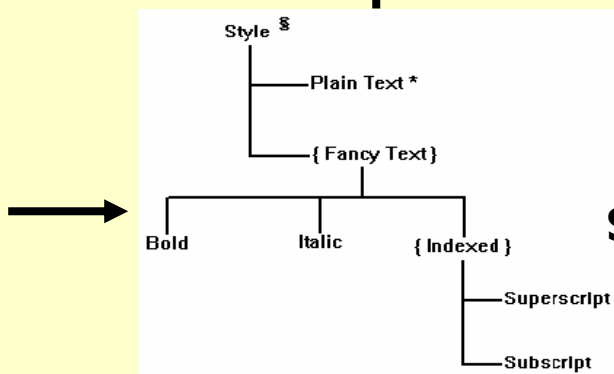
Delphi Interface

```
System, 16, 7, 0, 0, 700, 0, 0, 0, 0, 1, 2, 2, 34, 10,
1, [450-132617-154]VdIcNi; : CL[46, 132, 46, 0][46, 22, 46, 54][ 334, 54, 655, 54], Timetabling System
3, [93- 235, 185- 257]VdIcNi; : EL[47, 48, 47, 0][46, 22, 46, 153][0, 0, 0, 0], Categories
4, [1295-237, 401- 259]VdIcNi; : EL[53, 51, 53, 0][46, 22, 46, 103][0, 0, 0, 0], Semester
5, [1487- 231, 569- 253]VdIcNi; : EL[41, 45, 41, 0][42, 22, 42, 108][0, 0, 0, 0], Display
6, [834- 217, 531- 239]VdIcNi; : CL[48, 51, 48, 0][46, 22, 46, 40][ 90, 40, 164, 40], Controls
7, [1063- 230, 1120- 252]VdIcNi; : mL[22, 44, 22, 0][0, 0, 0, 0], Print
8, [154- 274, 209- 296]VdIcNi; : CL[ 39, 11, 0, 11][27, 22, 27, 32][27, 32, 53, 32], Paper
9, [153- 387, 218- 409]VdIcNi; : EL[ 38, 11, 0, 11][25, 22, 25, 65][0, 0, 0, 0], Rooms
10, [374- 230, 473- 312]VdIcNi; : mL[33, 11, 0, 11][0, 0, 0, 0], Semester 1
11, [377- 329, 478- 351]VdIcNi; : mL[ 36, 11, 0, 11][0, 0, 0, 0], Semester 2
12, [864- 294, 688- 306]VdIcNi; : mL[ 25, 11, 0, 11][0, 0, 0, 0], List Dates
13, [561- 328, 714- 300]VdIcNi; : mL[32, 11, 0, 11][0, 0, 0, 0], Display Timetable
14, [706- 236, 783- 318]VdIcNi; : mL[38, 39, 38, 0][0, 0, 0, 0], Lectures
15, [829- 236, 907- 318]VdIcNi; : mL[39, 39, 39, 0][0, 0, 0, 0], Tutorials
16, [844- 237, 1053- 315]VdIcNi; : mL[40, 54, 0, 0][0, 0, 0, 0], Laboratories
17, [146- 331, 268- 353]VdIcNi; : mL[61, 25, 61, 15][0, 0, 0, 0], Paper Number
18, [215- 443, 337- 465]VdIcNi; : mL[ 37, 9- 15, 9][0, 0, 0, 0], Room Number
A, Triggers ###
[17, 8, 82]c[11, 8, 73]c[228, 372, 228, 372]0;
```

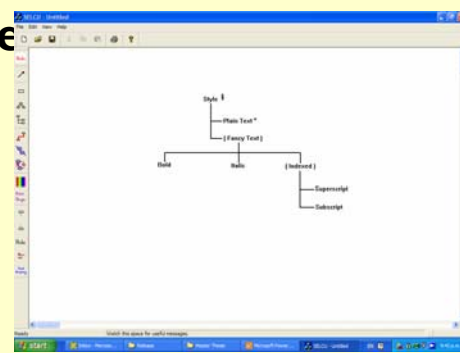
SELCU Compatible Text File (in HD)



Basic Tree Structure (in Memory)



Lean Cuisine+ Model (in Memory)



SELCU Graphical View (on Screen, in HD)

Case Study --- Timetabling System

Timetabling System

Paper Rooms

124.251
124.252
143.222
143.223
157.116
157.221
157.223
157.231
157.242
157.321
157.322
157.331
157.341
159.101
159.102
159.201
159.202
159.203
159.234
159.302
159.304
159.331
159.333
159.335
159.351

Semester 1 of 1999

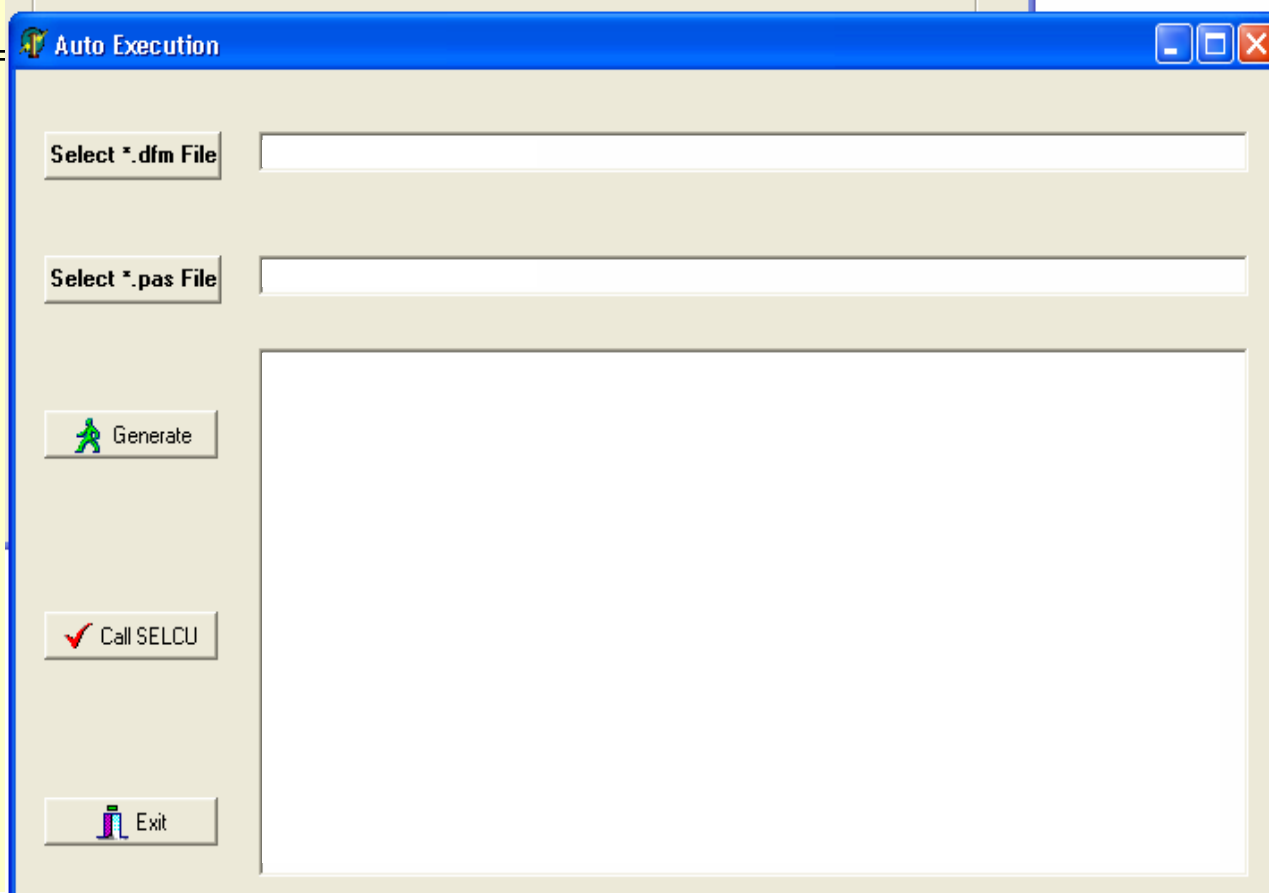
Timetable
 Clash List

Lectures
 Tutorials
 Laboratories

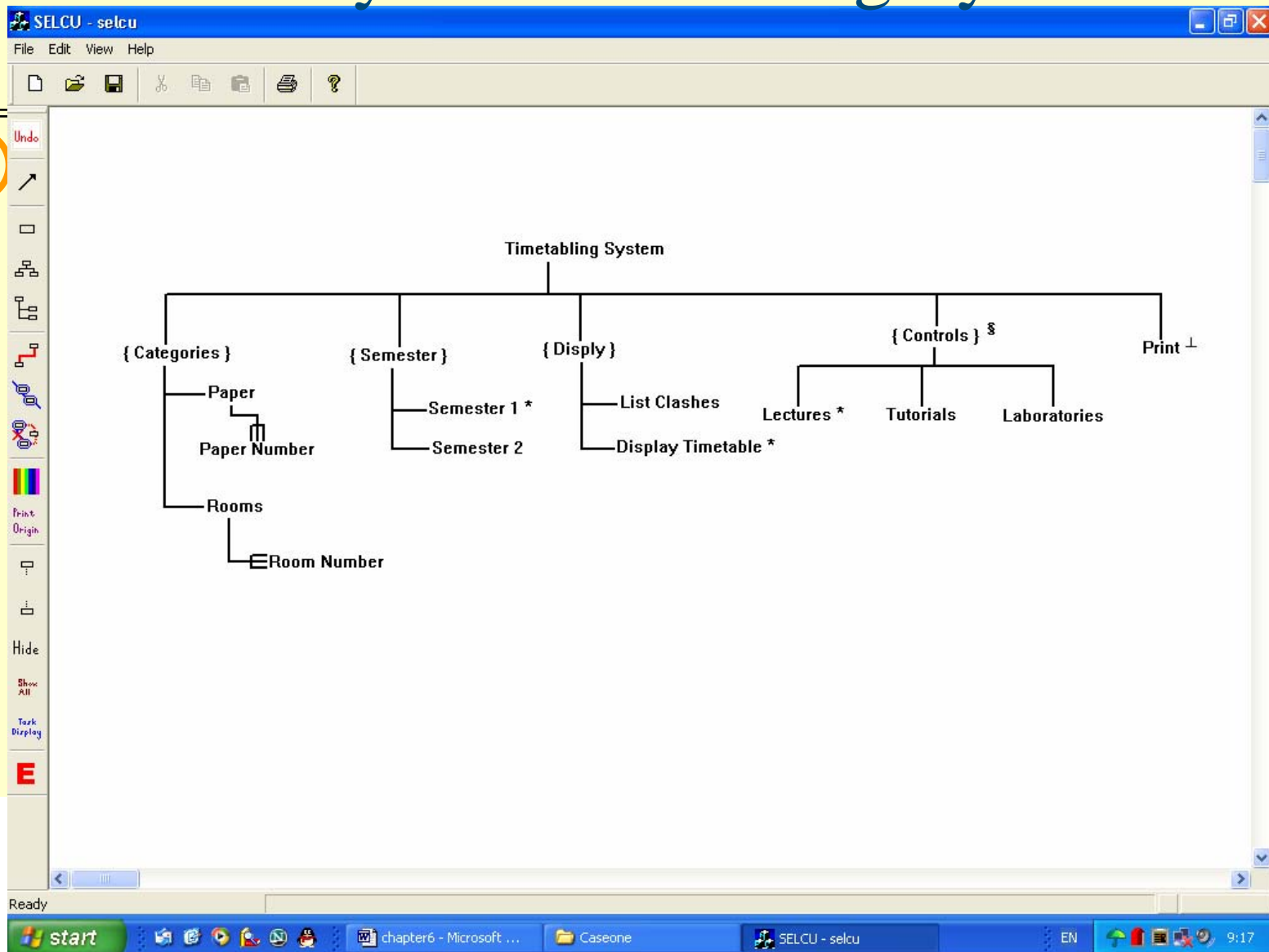
	8am	9am	10am	11am	12pm	1pm	2pm	3pm	4pm	5pm
Mon			160.101 SSLB 1				159.101 AHA3.01			
Tue		159.101 SST1.39					160.101 SSLB 2			
Wed	160.101 AHA3.02		159.101 SST4.40 (T)							
Thu		159.304 AHA3.08						157.221 SST2.21 (T)		
Fri				159.101 AHA3.01		159.351 SSLB 6 (T)				

Print

Case Study --- Timetabling System



Case Study --- Timetabling System



Conclusion

The goal is to explore the possibility of Automatically Generating Behavioral Description of Graphical User Interfaces

Integrate a Industry Standard Programming Environment (Delphi IDE) and a Dialogue Notation (Lean Cuisine+)

In addition to supporting analysis of interaction, the generated dialogue model provides useful documentation of the system behavior

Future Work

The approach described in this presentation could be extended to other programming environments (e.g. Java, C++ ...)

Auto-Generation Software needs to be extended to handle a wider range of Delphi interface components

The final generation result could be specified in a more general / popular standard (e.g. XML style)

THANKS

LEAN CUISINE+



THE UNIVERSITY OF AUCKLAND
NEW ZEALAND