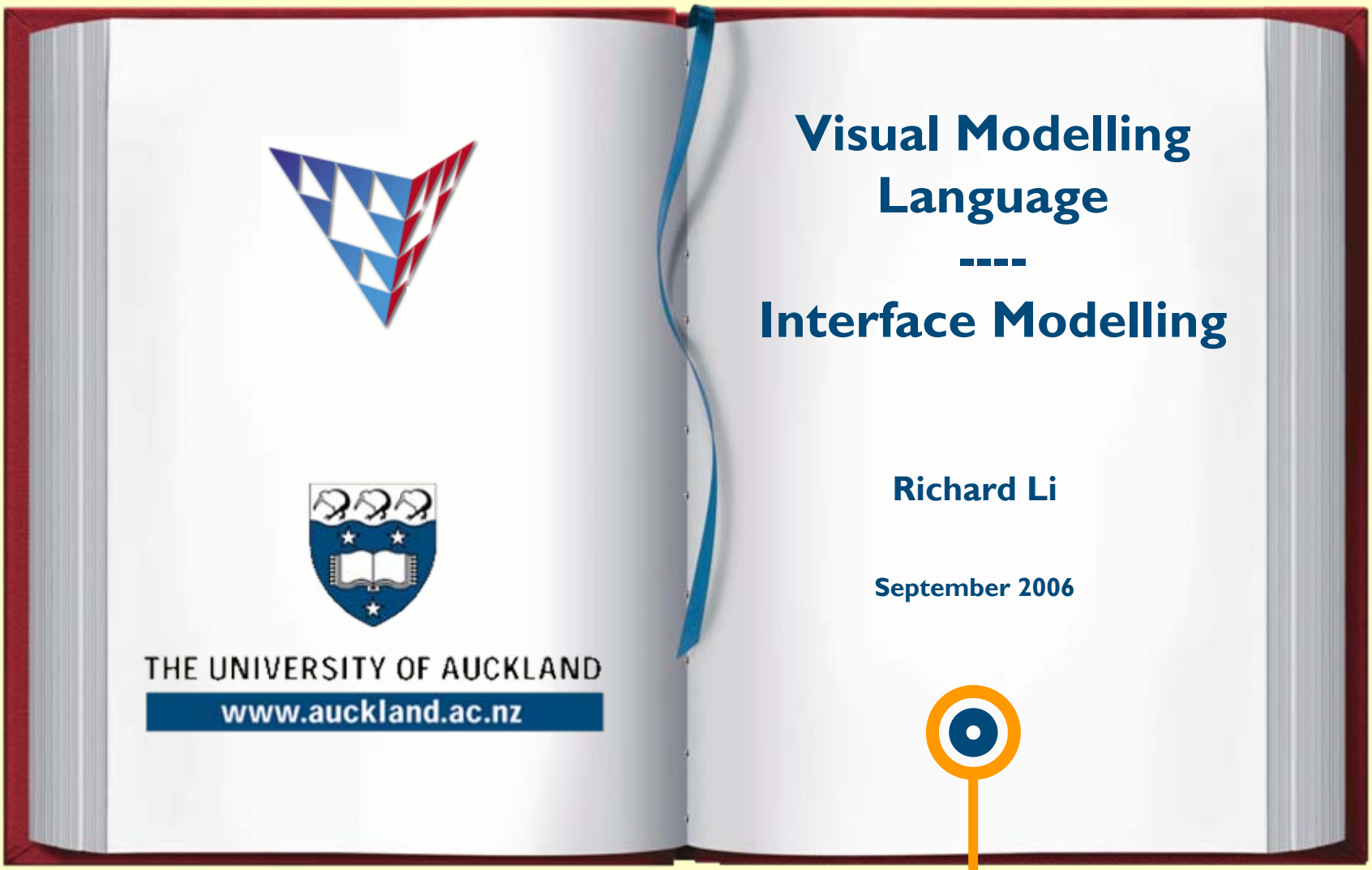


Visual Modelling Language

Interface Modelling

Richard Li

September 2006

THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

What is Visual Modelling Language

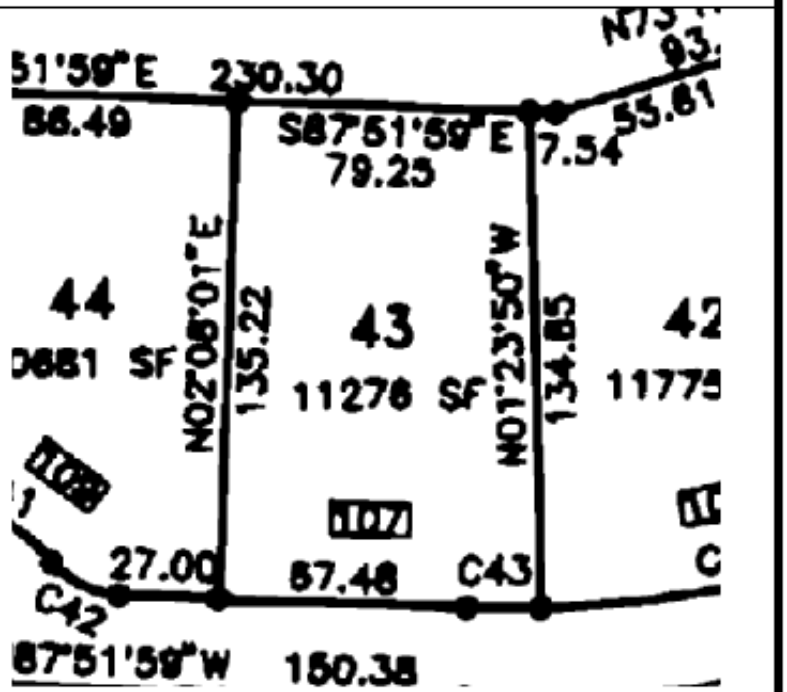
- *"If you were supposed to understand it, we wouldn't call it code."*
- *Visual* modeling languages use a diagram techniques with named symbols that represent concepts and lines that connect the symbols and that represent relationships and various other graphical annotation to represent constraints

Why???

Enhance Communication

Legal "Code" vs. Picture

BEING all of Lot 43, Franklin Chase Subdivision, Phase One-A, according to a plat recorded in Book of Maps 1993 at Page 1132 and rerecorded in Book of Maps 1993 at Page 1279, …



Why???


Caveman
+hunt()
Homosapien
public class Caveman
extends Homosapien {
public void hunt(Animal prey) {

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	JAVA	C++	BASIC
↓	↓	↓	↓
Delphi	J Builder	Visual C++	Visual Basic

+ Reduce the Amount of Code

+ Quickly Construct Graphical Interfaces

+ Achieve a Consistent Look and Feel

Research Motivation

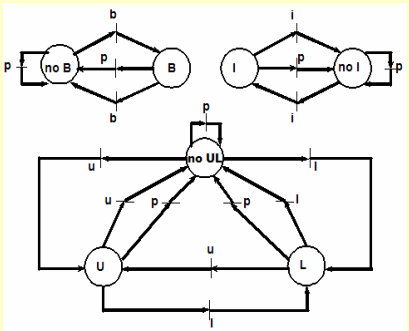
Interface Builders

Research Motivation

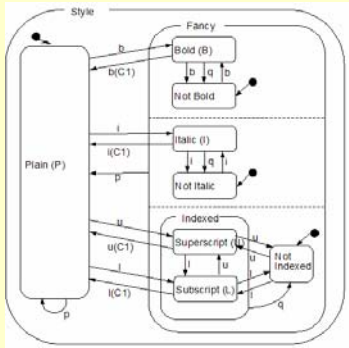
Dialogue Notations

Dialogue Models Provide a Behavioral Description for the User Interface

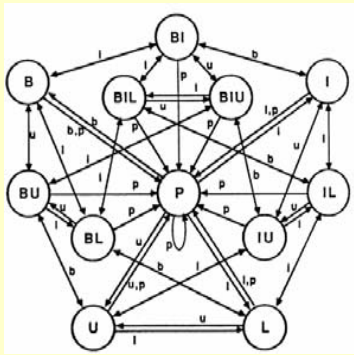
Petri Nets



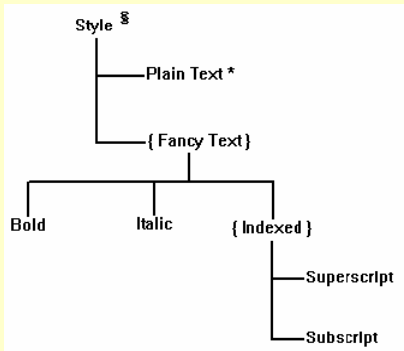
State Charts



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

Department of Computer Science

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Research Motivation

2006

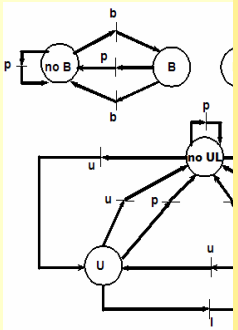
COMPUTER SCIENCE

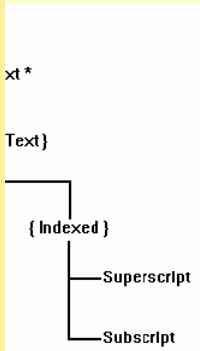
The University of Auckland

Dialogue Notations

Dialogue I **nterface**

Petri N **uisine+**





+ Using a V **JI**

+ Directly I **ntation**

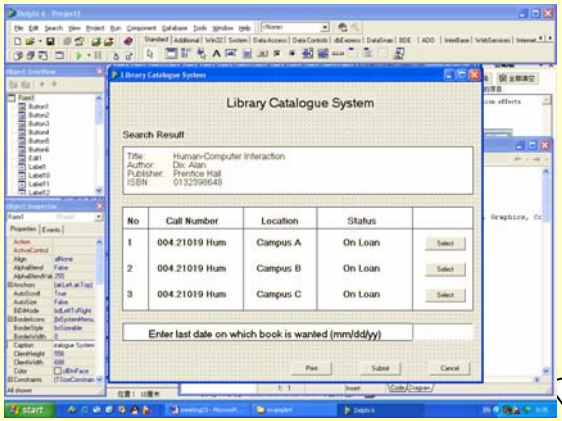
+ The Desc

GAP

Department of Computer Science

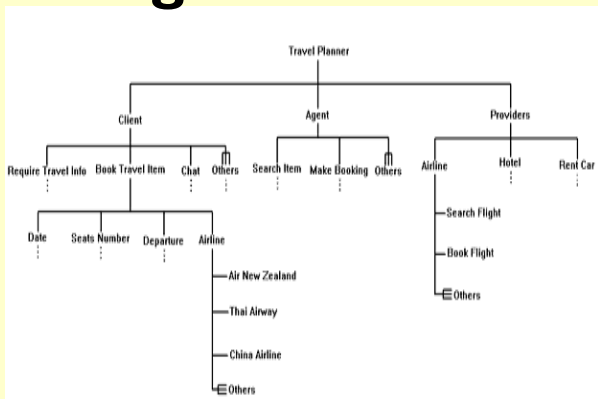
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Research Motivation



Interface Builders

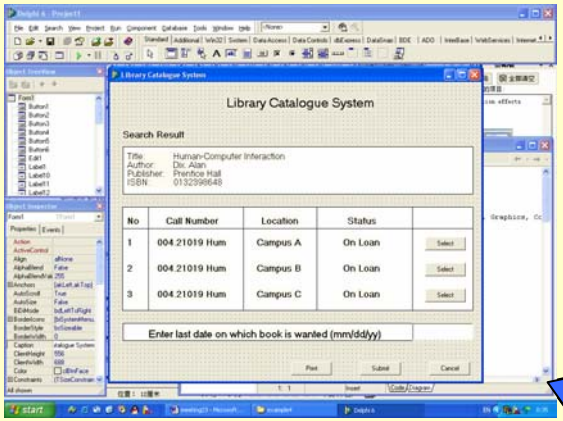
Dialogue Notations



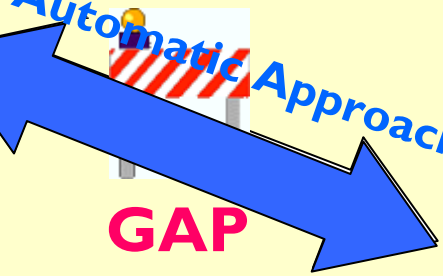
Department of Computer Science

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

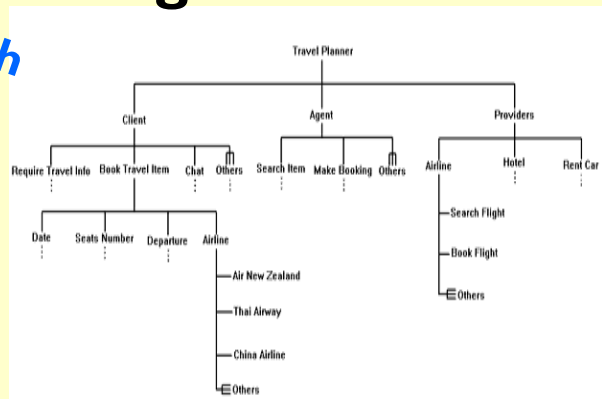
Research Motivation



Interface Builders



Dialogue Notations



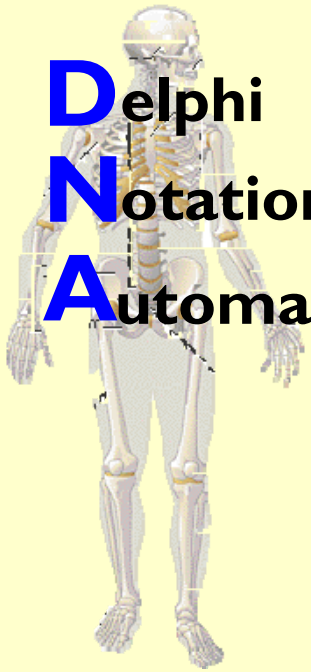
Department of Computer Science

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

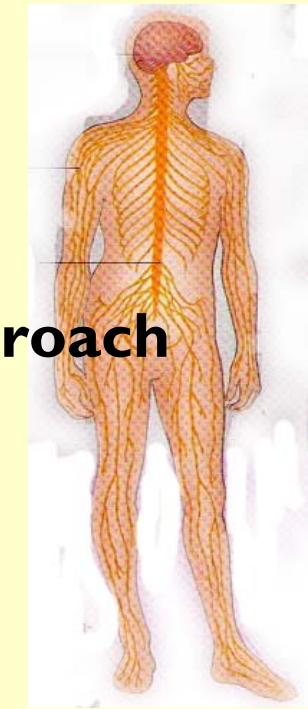
Research Perspective

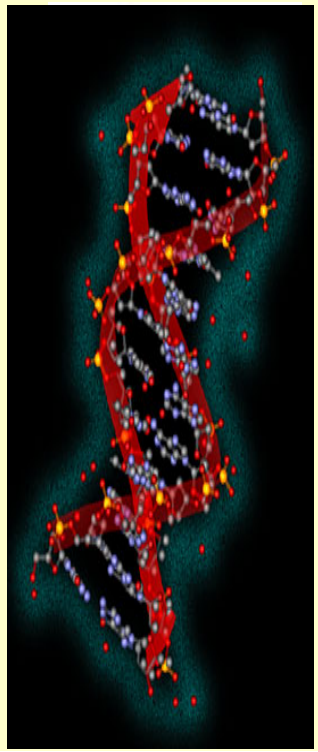
Delphi **Notation** **Automatic Approach**

Interface Builder **Lean Cuisine+** **Methodology**

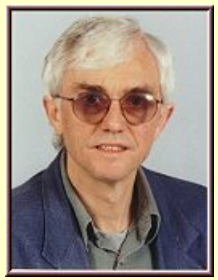


Delphi
Notation
Automatic Approach





Lean Cuisine+ History

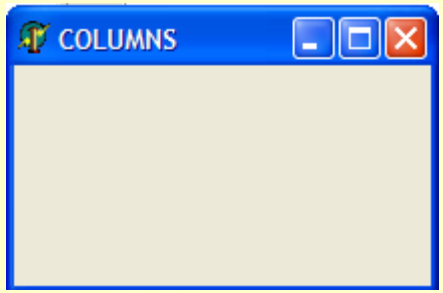


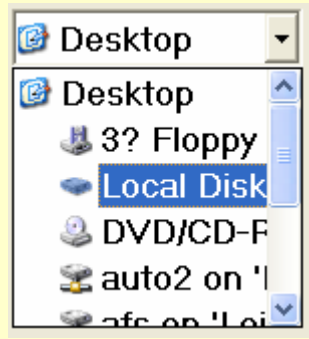
Lean Cuisine+ was developed by Dr Chris Philips at Massey University

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

Lean Cuisine+ Introduction

Interface Components --- Meneme





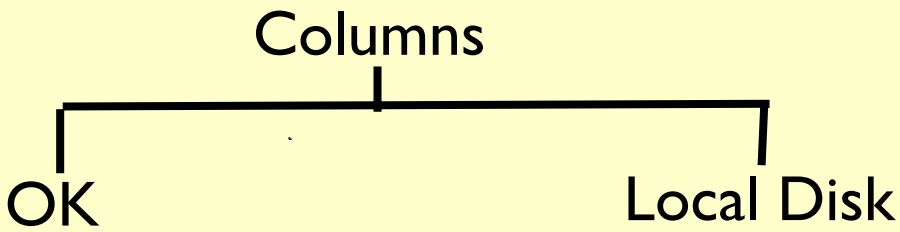
OK

Columns

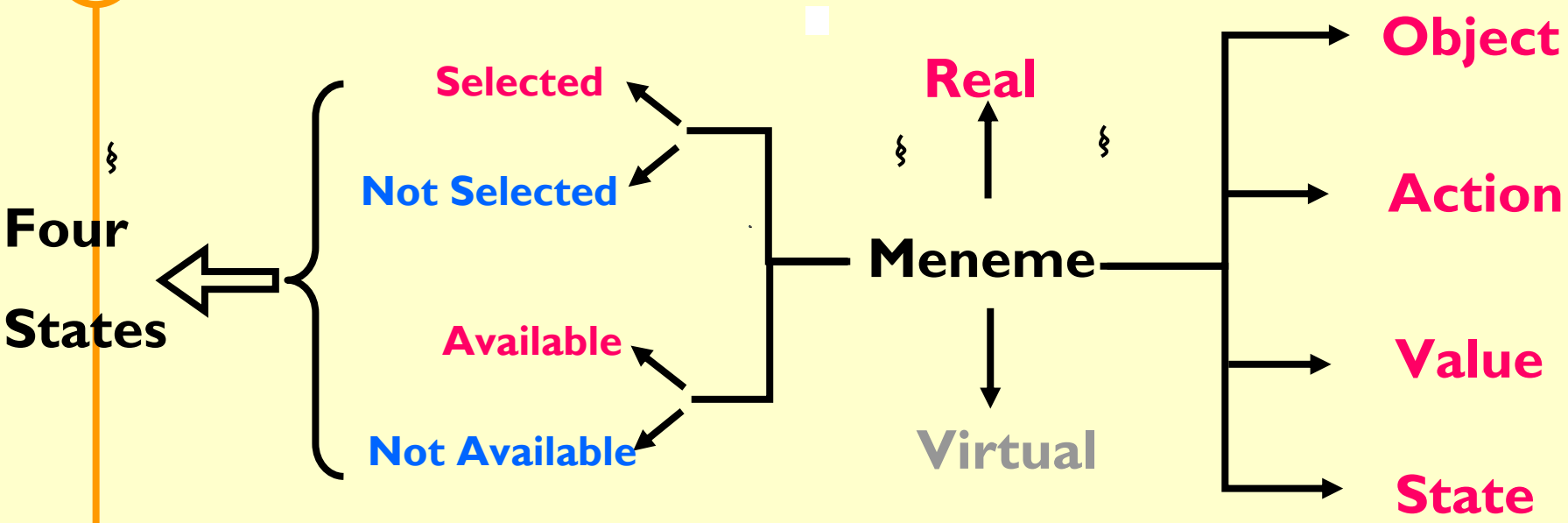
Local Disk

Lean Cuisine+ Introduction

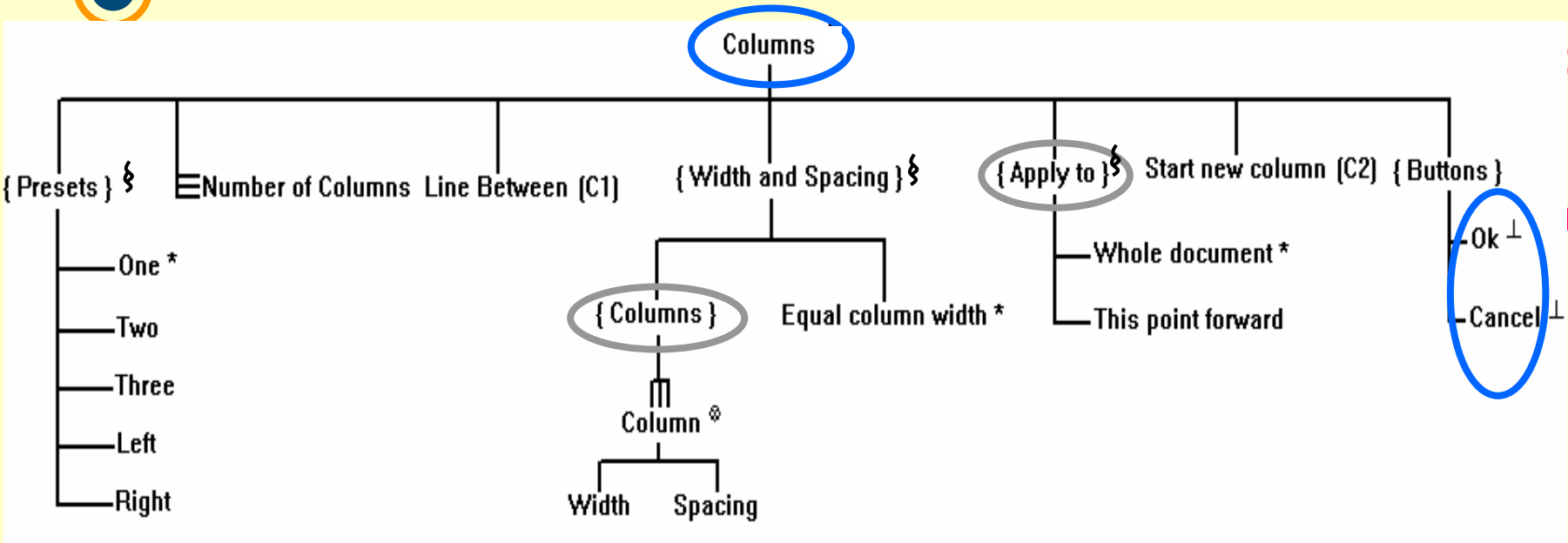
Tree Structure



Lean Cuisine+ Introduction



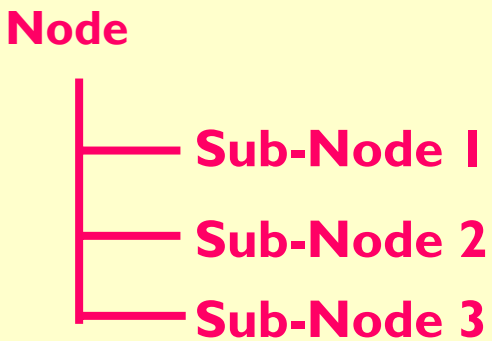
Lean Cuisine+ Introduction



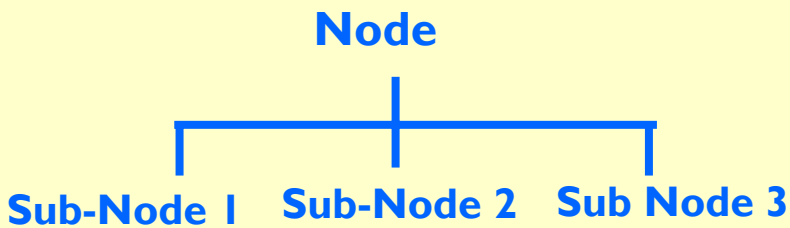
Lean Cuisine+ Introduction

Two Grouping Constraints on Behavior

Mutually Exclusive
(1 from N)

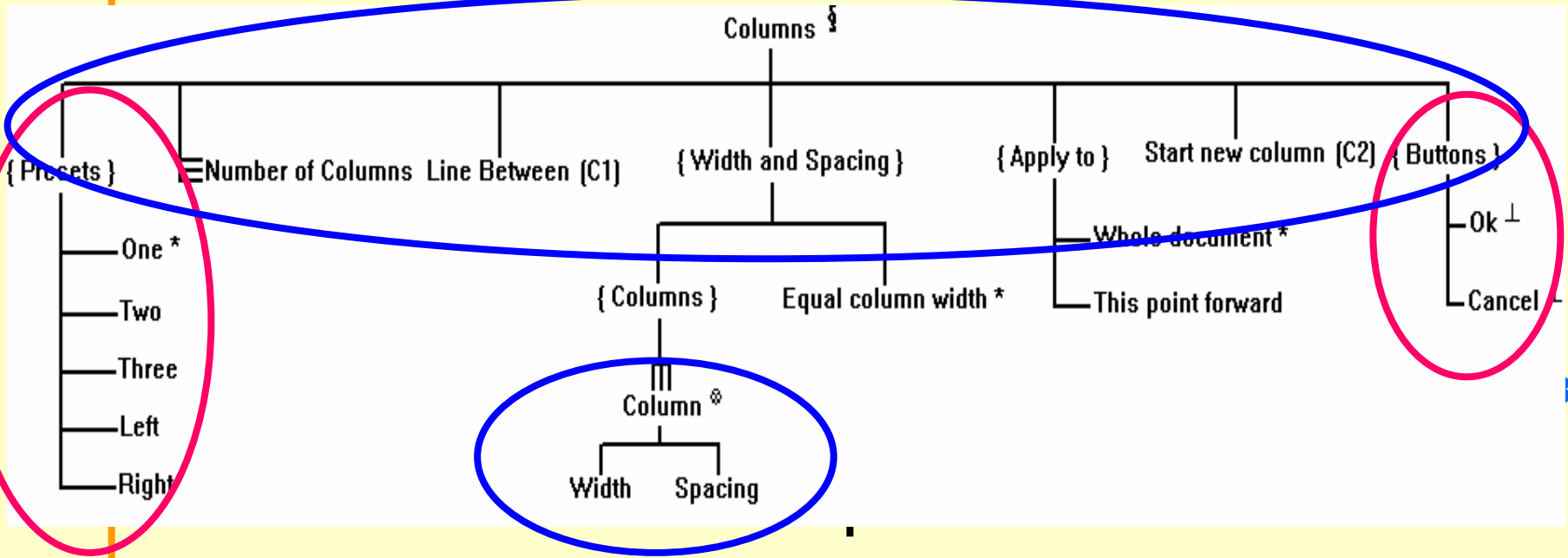


Mutually Compatible
(M from N)



Lean Cuisine+ Introduction

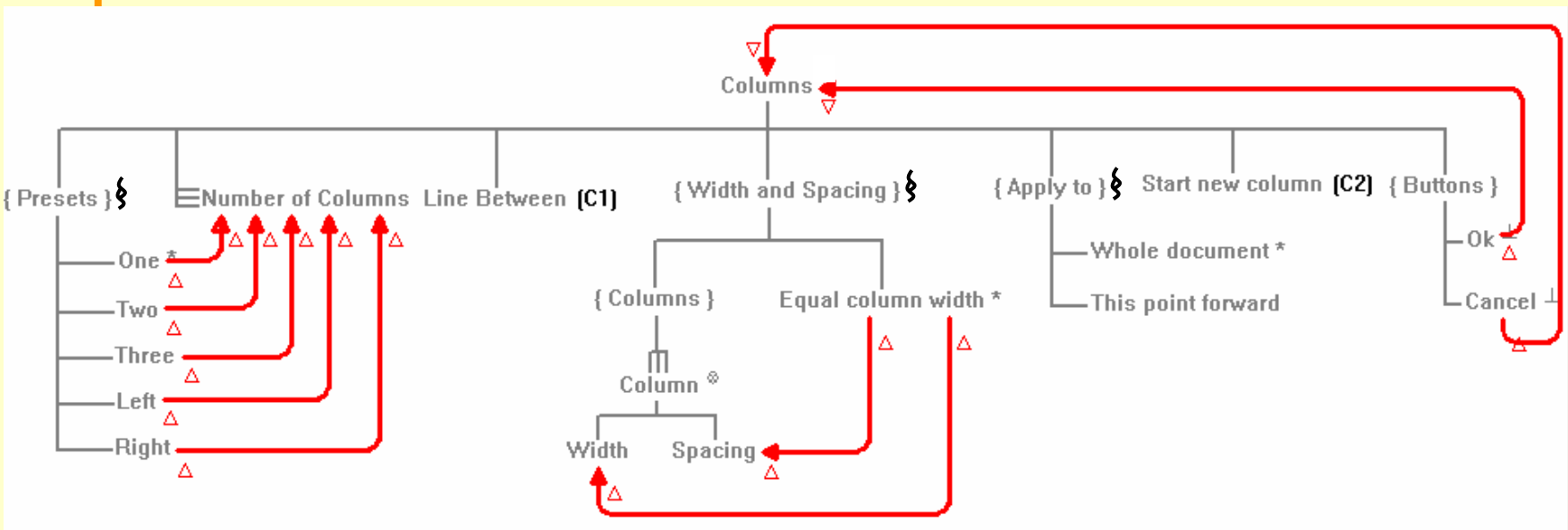
Two Grouping Constraints on Behavior



Lean Cuisine+ Introduction

Triggers

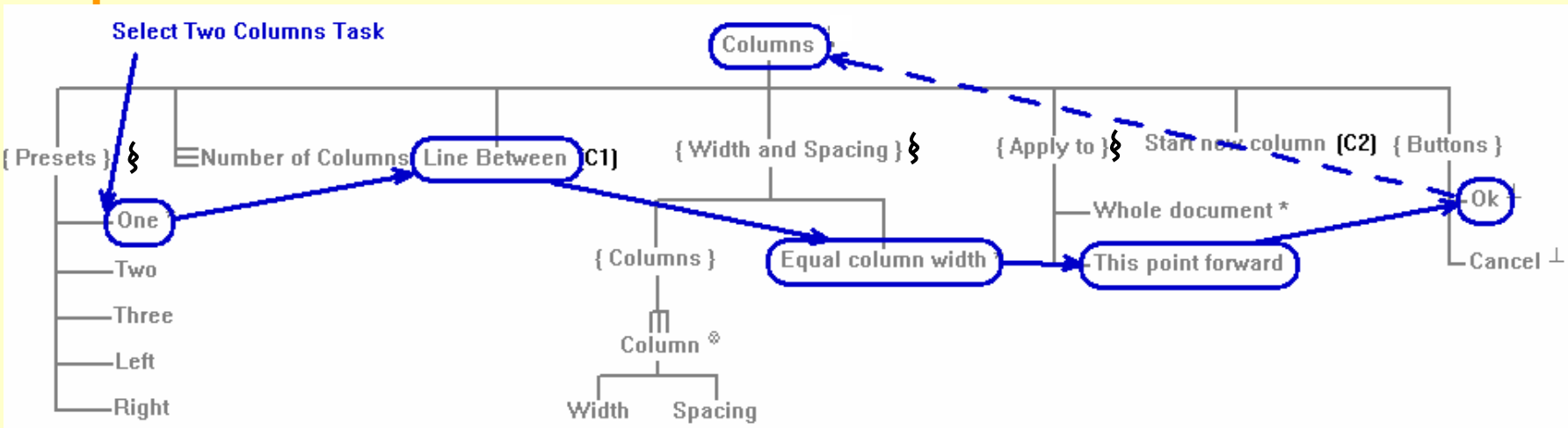
are used to show System Internal Selections



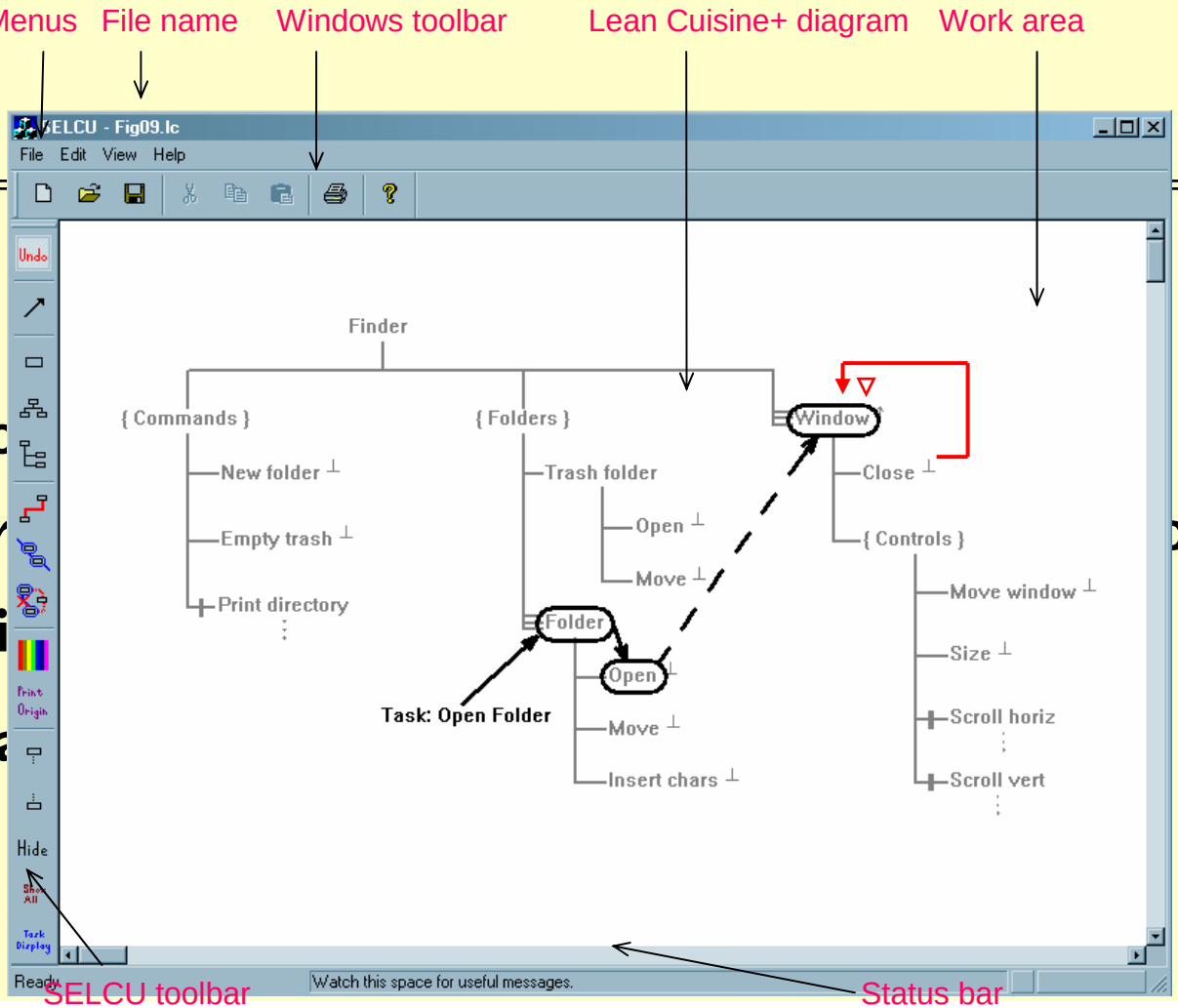
Lean Cuisine+ Introduction

Tasks

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



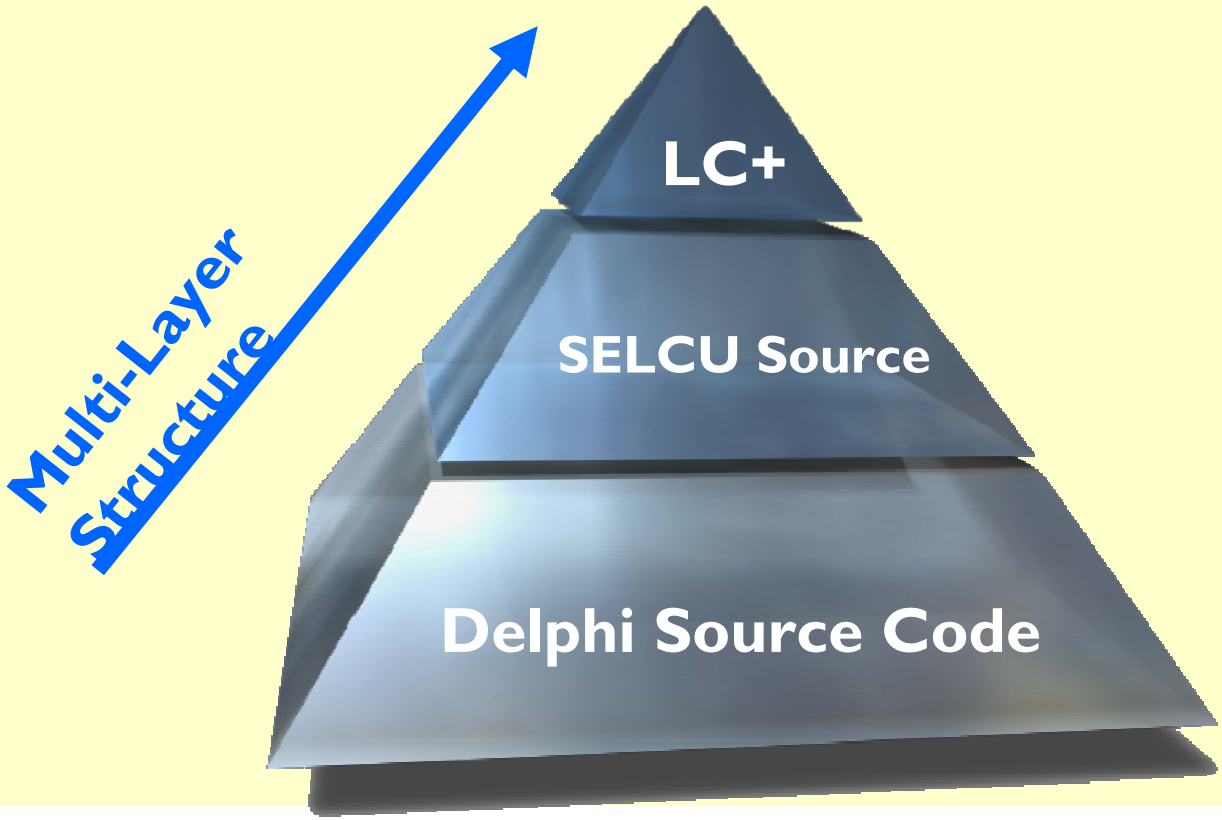
SELCU



- Sc
- Pr
- Di
- Da

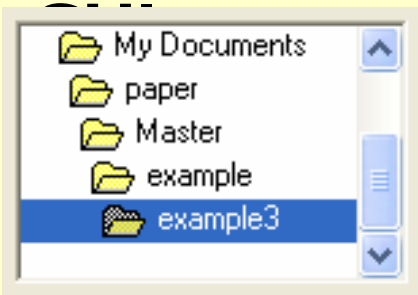
on LC+

Automatic Generation Structure

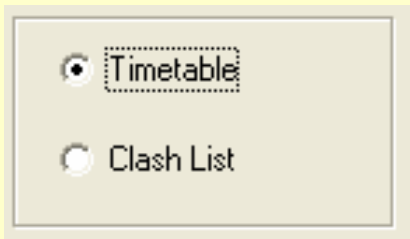


Automatic Generation Approach

Delphi

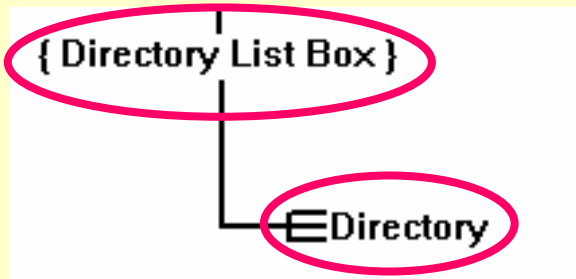


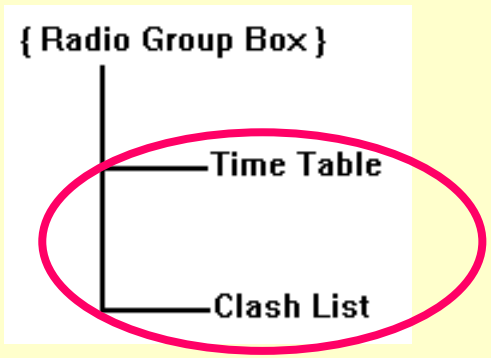
List Box / Combo Box



Radio Group

Lean Cuisine+ Model



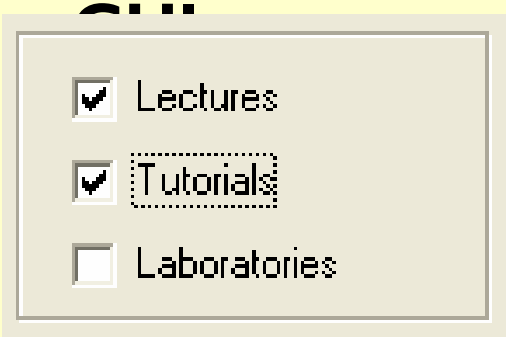


Department of Computer Science

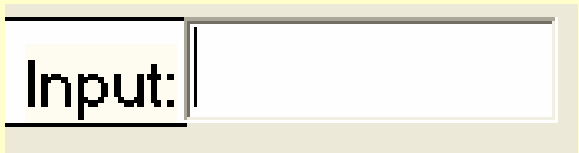
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Automatic Generation Approach

Delphi

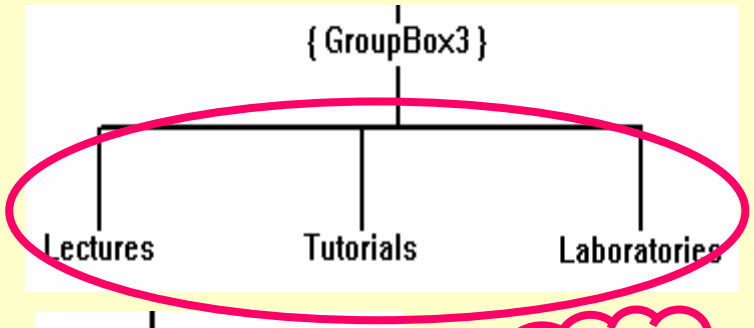


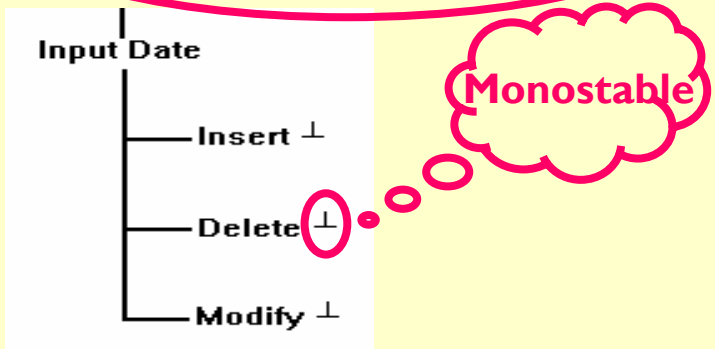
Check Box



Data Input Box

Lean Cuisine+ Model

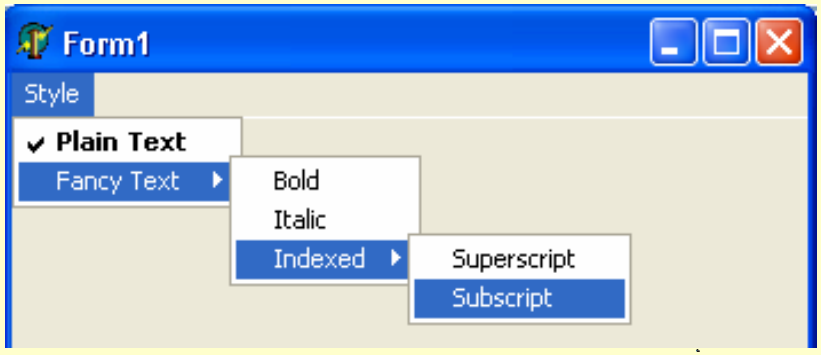




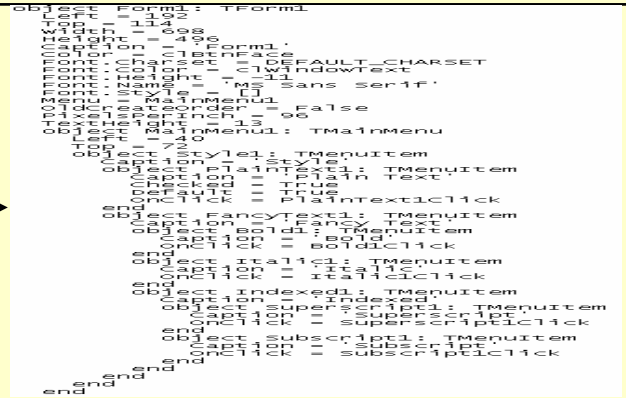
Department of Computer Science

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

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**

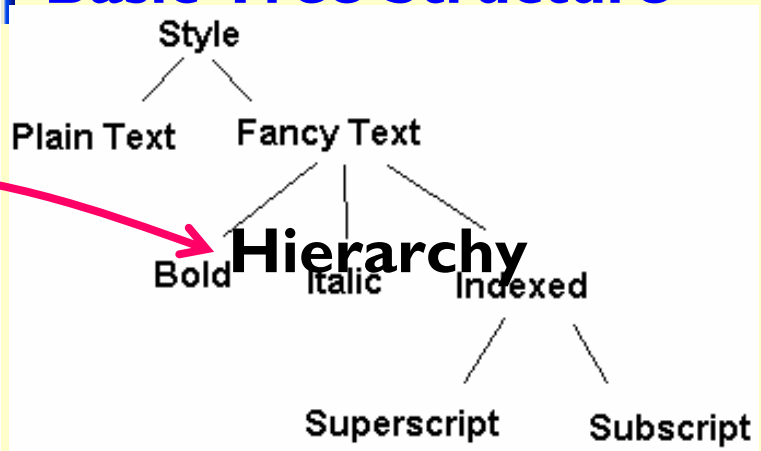
Automatic Generation Approach

```
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
    object Indexed1: TMenuItem
      Caption = 'Indexed'
      object Superscript1: TMenuItem
        Caption = 'Superscript'
        OnClick = Superscript1Click
      end
      object Subscript1: TMenuItem
        Caption = 'Subscript'
        OnClick = Subscript1Click
```

.dfm File

Meneme
Name
System Default
Option

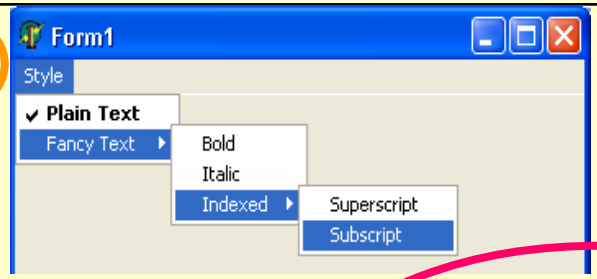
Basic Tree Structure



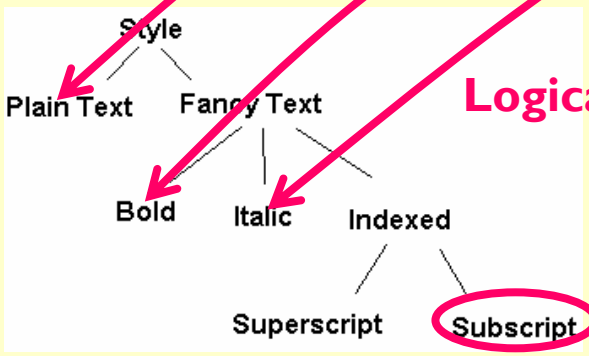
Department of Computer Science

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Automatic Generation Approach

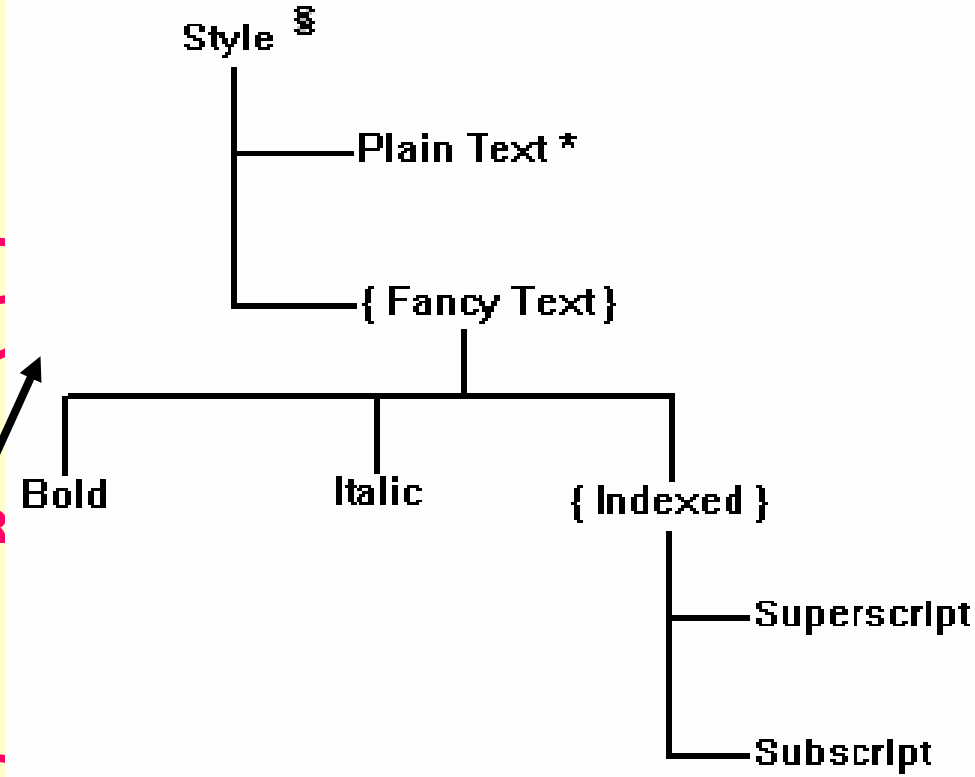


Delphi Interface



Logical R

Basic Tree Structure



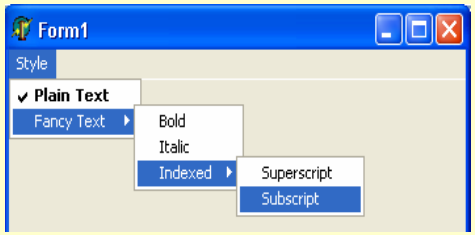
end;

Lean Cuisine+ File Model

Department of Computer Science

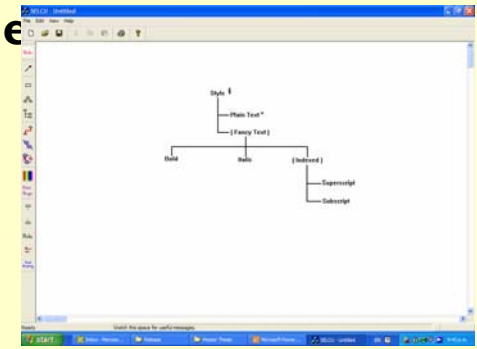
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

Automatic Generation Process Sequence

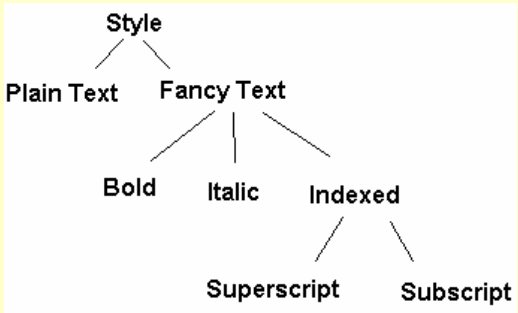


Delphi Interface

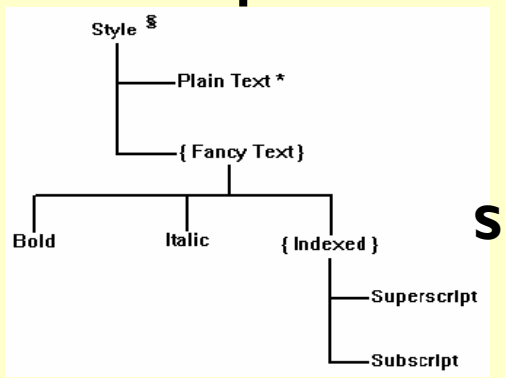
SELCU Compatible Text File (in HD)



SELCU Graphical View (on Screen, in HD)



Basic Tree Structure (in Memory)

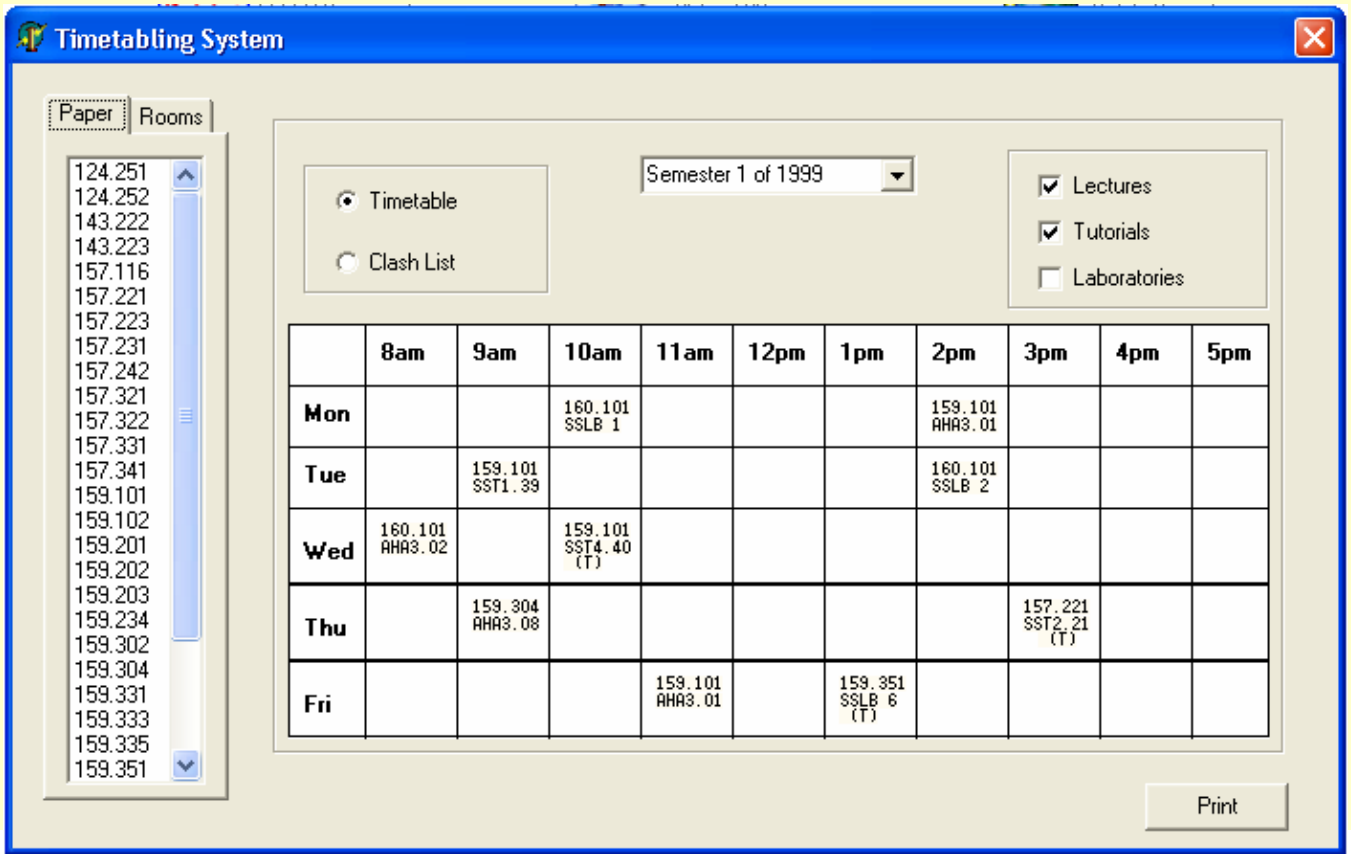


Lean Cuisine+ Model (in Memory)

Department of Computer Science

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

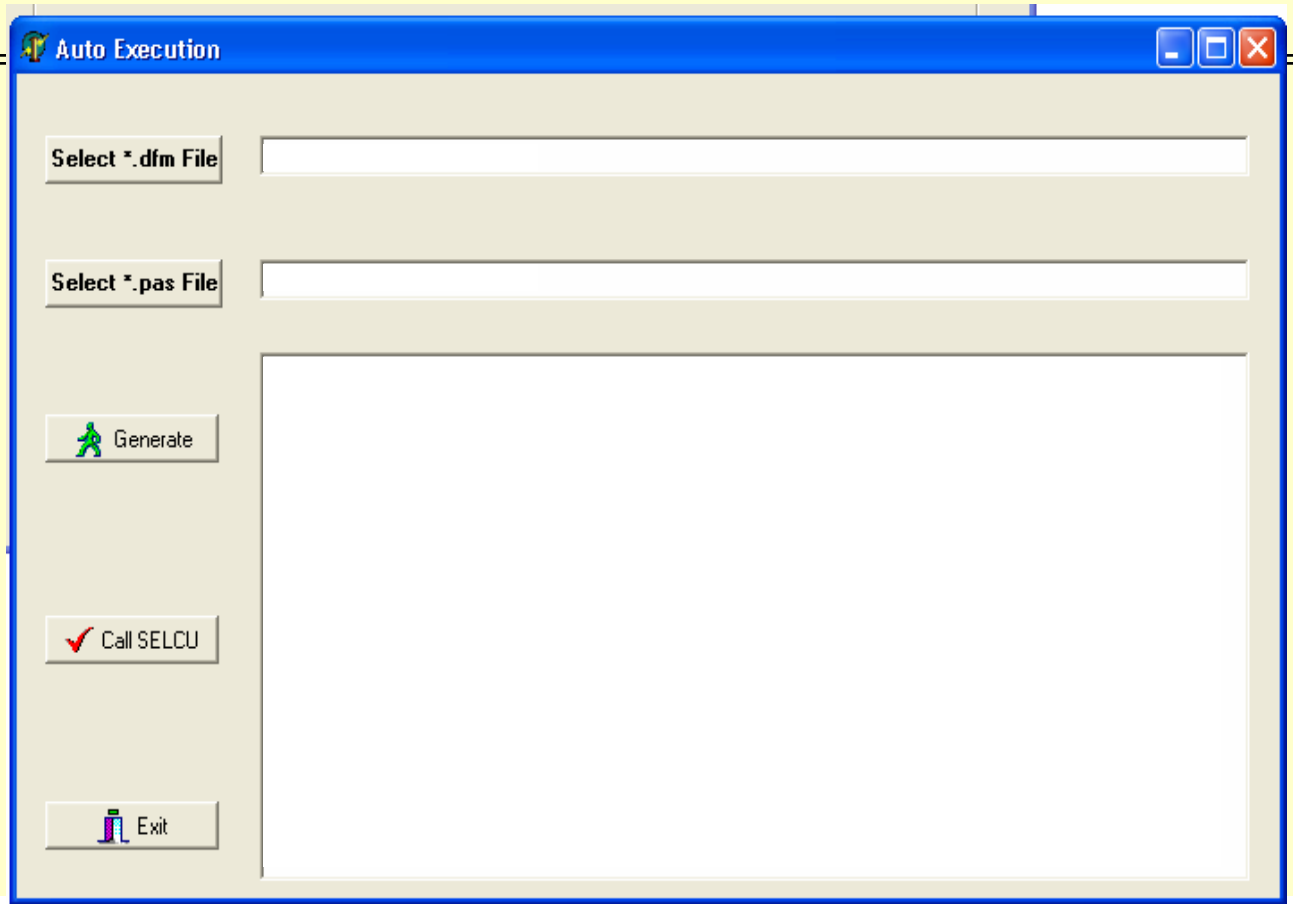
Case Study --- Timetabling System



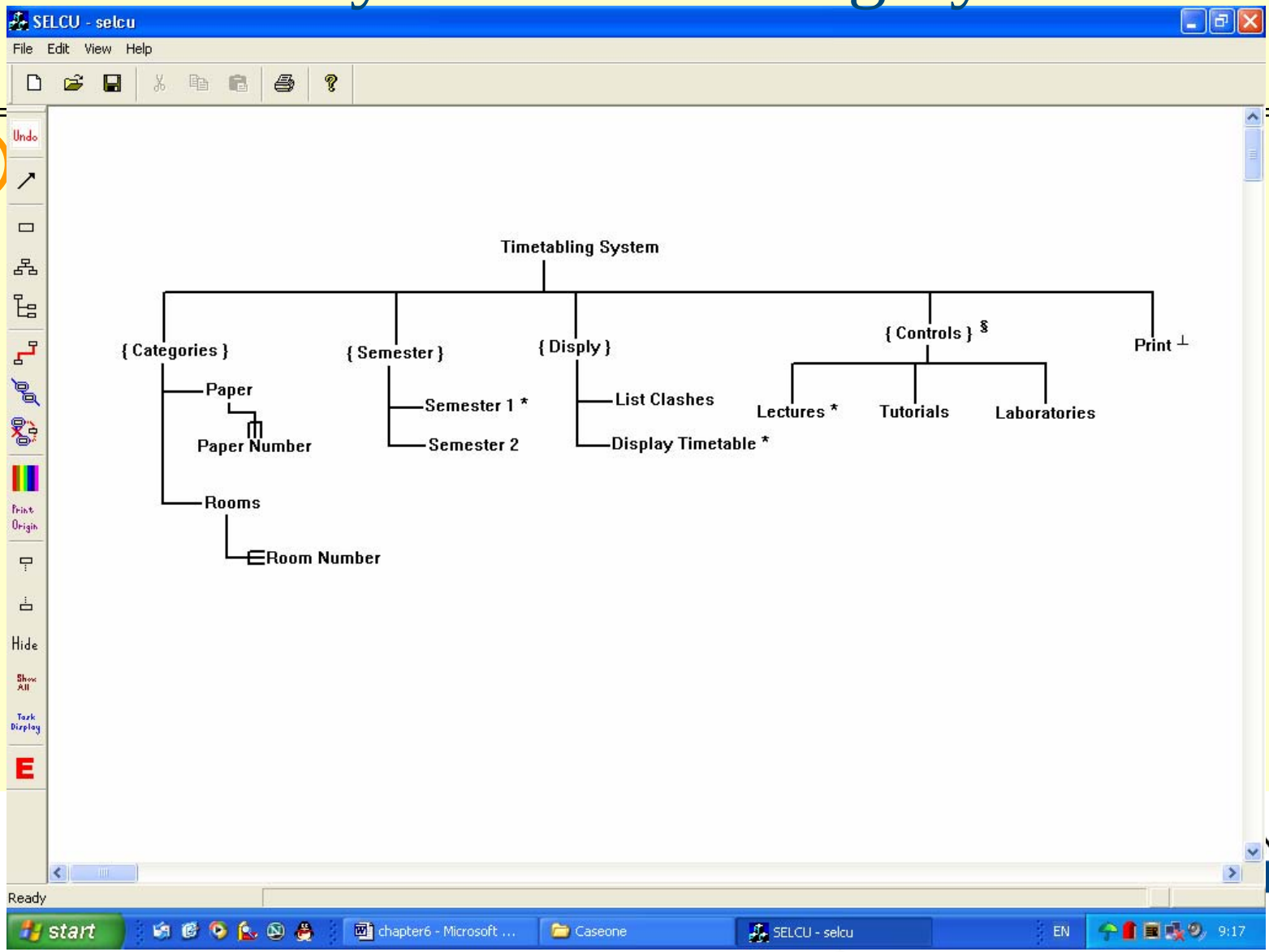
Department of Computer Science

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

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+

Department of Computer Science

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND