

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

123
6059

**DIALOGUE ACTIVATION:
AN APPROACH TO USER CENTRED CONSTRUCTIONAL
MODELLING OF
DIRECT MANIPULATION INTERFACES**

**A dissertation presented
in partial fulfilment of the requirements
for the degree of
Doctor of Philosophy in Computer Science
at Massey University**

Paul Stuart Anderson

1995

Acknowledgements

I would like to thank my supervisors, Mark Apperley and John Hudson for many helpful discussions and suggestions over the course of this research.

I should also like to thank my family (Kathy, Christopher, Miles, Philip and Rosemary) without whose support this project would not have been possible.

Finally I should like to thank Chris Phillips for reading a draft of this thesis and giving me a great deal of useful feedback, and encouragement to finish it.

This work was funded in part by a grant from the Massey University Research Fund.

Abstract

Early stages in the development of interfaces involve the construction of models that aid interface analysis prior to construction. These behavioural models generally take a user-centred perspective. In contrast, subsequent implementation models tend to take a system-centred view of the interface. As a consequence of this change in viewpoint, the task of translating an analysis model into its implementation equivalent is extremely difficult.

This thesis proposes a constructional modelling approach for direct manipulation user interfaces (DMUI) that takes a user action viewpoint. Based on a hierarchy of dialogue groups and the notion of dialogue activation, sequence and concurrency within the interface can be described. Dialogues can be in one of two possible states, active or inactive. An active dialogue is one with which the user is able to interact. A dialogue becomes active only if its parent is active, and it receives one of a set of possible activating events. A second set of deactivating events can also exist. In this way a dialogue can be specified in terms of both a user's actions and the sequences in which those actions may be carried out. Dialogue Activation Language (DAL), a language for describing such models is developed, and shown to be applicable to a range of interaction styles. An architecture capable of implementing the dialogue activation model is proposed, and a user interface development system (PIPS), based on this architecture and using DAL is described.

It is argued that DAL takes the same view of an interface as would be used in its initial analysis, and as a consequence, facilitates the translation of these early interface models into working prototypes. In addition, it is proposed that taking the DAL approach to modelling DMUI allows great flexibility in describing interaction and encourages experimentation with entirely new interaction styles.

Contents

Chapter 1

Motivation and Scope	1
1.1 Introduction.....	1
1.2 The Behavioural and Constructional Domains	2
1.3 Direct Manipulation	6
1.3.1 A Definition of Direct Manipulation.....	7
1.4 Tools for User Interface Development	10
1.5 Motivation	12
1.6 Research Objectives	13
1.7 Thesis Outline	13

Chapter 2

Dialogue Modelling in the Behavioural Domain	15
2.1. Introduction.....	15
2.3. Dialogue Models.....	19
2.3.1. User Action Notation	19
2.3.2. Lean Cuisine	25
2.3.3. GOMS	29
2.3.4. Command Language Grammar	29
2.3.5. Notations for Comparison of Interactions	32
2.4. Summary.....	34

Chapter 3

Dialogue Modelling in the Constructional Domain.....	37
3.1. Introduction.....	37
3.2. Structural Models	37
3.2.1. Linguistic based architectures.....	38
3.2.2. Agent based architectures	40
3.2.3. Surface Interaction	43
3.3. Dialogue Models.....	43
3.3.1. Context Free Grammars	45
3.3.2. State Transition Networks	49
3.3.3. Statecharts	54

3.3.4.	Event Dispatch Models	57
3.3.5.	Propositional Production Systems	63
3.3.6.	Process Algebras	64
3.3.7.	Other methods.....	69
3.4.	Summary.....	69
Chapter 4		
Behavioural to Constructional Model Translation		71
4.1.	Introduction.....	71
4.2.	Model Translation	71
4.3.	Translation of Lean Cuisine	72
4.4.	Conclusion	78
Chapter 5		
Dialogue Activation Language		83
5.1.	Introduction.....	83
5.2.	Dialogue Activation	83
5.3.	Specification of Events	88
5.4.	Specification of Activation & Deactivation.....	89
5.5.	Action Sequences.....	90
5.6.	Concurrency	91
5.7.	Visualisation of Concurrency	92
5.8.	Dialogue Specification	94
5.9.	Variables	97
5.10.	Guard Conditions.....	98
5.11.	Scope of Events	99
5.12.	Presentation and Application Attachment	101
5.12.1.	Display statements.....	102
5.12.2.	Application statements	104
5.13.	Object Oriented Features of DAL.....	106
5.14.	Dynamic Creation of μ dialogues.....	111
5.15.	Architectural Context.....	115
Chapter 6		
Formal Semantics of DAL		121
6.1.	Introduction.....	121
6.2.	The DAL Framework	121
6.3.	System States	122
6.4.	State Transitions and Event Propagation.....	125
6.4.1.	Action Event Propagation	126

6.4.2.	System Event Broadcasting	133
6.4.3.	Anonymous Transitions	137
6.5.	Variables	137
6.6.	Concurrency	139
6.7.	Potential System Problems	142
6.7.1.	Concurrency Related Faults	142
6.7.2.	DAL Specific Faults	147
6.8.	Scope Of This Analysis	158
Chapter 7		
The PIPS Development Environment		161
7.1.	Introduction.....	161
7.2.	Implementation Options	161
7.3.	The PIPS 'agent' Model.	163
7.4.	The PIPS Architecture	164
7.5.	DAL Compilation	166
7.5.1.	Class expansion	166
7.5.2.	Variable and event identifier assignment	168
7.5.3.	Statement translation.....	170
7.6.	Run-Time Components	170
7.6.1.	Framework agents	171
7.6.2.	Interface agents	172
7.7.	Framework Agents	172
7.7.1.	Action event queue	172
7.7.2.	System event handler	177
7.7.3.	Template agent	179
7.7.4.	Interface agent	180
7.8.	Interface Agents	180
7.8.1.	Dialogue agents	180
7.8.2.	Widgets.....	181
7.8.3.	Application interface	183
7.9.	Optimisation Of The Prototype System	183
7.10.	PIPS Debugging Facilities	187
7.10.1.	Activation tracing	187
7.10.2.	Event record tracing.	188
Chapter 8		
Design Examples		191
8.1.	Introduction.....	191

8.2. Dialogue Box	191
8.3. A Pin-up, Pop-up Menu	192
8.4. Drag and Drop	194
8.5. Gesture Recognition	199
Chapter 9	
Conclusions and Further Work	205
9.1. Introduction	205
9.2. Contributions of this work	205
9.3. Questions and answers	206
9.4. Further Work	209
Bibliography	215
Appendix A	
BNF Definition of DAL	225
Appendix B	
Event Cycle Detection	231
Appendix C	
Example Dialogues	241
1. Widget and Application Classes	241
2. Closebox	245
3. Simple paint dialogue	249
4. Pulldown Menu	256
5. Pin-up Pop-up Menu	266
6. Brush Shape Dialogue Box	275
7. Simple Drag and Drop	284
8. Drag and Drop with Multiple Destinations	289
9. Dynamic "drag and drop"	295
10. Simple Gesture Recognition	305
11. Compound Gesture Recognition Tablet	310