List of Figures

1-1	A simple model of people	23
1-2	A more complicated model of people	25
1-3	Representing gender by the value of an enumeration type	30
1-4	Representing gender by subtypes	30
1-5	Template for a mapping program with prior checking of target model constraints	34
1-6	Transference of constraints for schema-to-schema mapping	35
1-7	A transformed source model	36
2-1	A set of program generation commands	47
2-2	A sample overlay	52
2-3	A high-level view of a program's architecture	58
2-4	Reengineering = reverse engineering + forward engineering	60
3-1	Case-based reasoning	71
3-2	Example of a case	73
3-3	Linear case library organisation	80
3-4	Hierarchical case library organisation	82
4-1	A model of people	109
4-2	Top-level architecture of the constraint-understanding system	110
4-3	Architecture of the constraint-understanding system, with its case library	113
5-1	A simple example of a frame	114
5-2	A frame linked to another frame	115
5-3	A model for the frame classes in Figures 5-1 and 5-2	116
5-4	A sample FIL file	119
5-5	A frame class model for Figure 5-6	122
5-6	An FIL file to illustrate defined type instances	122
5-7	A frame class model for Figure 5-8	123

5-8 An FIL file to illustrate subtyping in FIL 123
5-9 A model of students and supervisors 125
5-10 An FIL file to be input to the frame system 125
5-11 The frames in Figure 5-10, after being processed by the frame system 126
5-12 A grammar for FIL 128

6-1 An EXPRESS model to be parsed into frames 131
6-2 Frame representation of the model in Figure 6-1 132
6-3 Some HLC frames which would be extracted from the model in Figure 6-1 135
6-4 Conjunction frames 136
6-5 An entity frame, tagged with a conjunction frame to hold the HLCs extracted from that entity 137
6-6 A case for extracting a discrete, inclusive, lower bound from '>=' 140
6-7 A case for extracting a range constraint from the conjunction of two bounds 144
6-8 A case for extracting a conjunction from an `AND` operation 145

7-1 Table to show the numbers of higher-level constraints of each kind extracted from three sample models 153

B-1 Representing a person's gender by the value of an enumeration type 188
B-2 Representing a person's gender by subtypes 189
B-3 A case for extracting 'exactly one' HLCs from models like Figure B-1 191
B-4 A case for extracting 'exactly one' HLCs from models like Figure B-2 192
B-5 'Exactly one' can be realised as a select type 192
B-6 'Exactly one' can be realised as constrained boolean flags 193
B-7 The sixteen cases for extracting bound constraints 196
B-8 A numeric range constraint 198
B-9 A symbolic range constraint 198
B-10 A model of rectangles in which area is constrained by a domain rule 199

B-11 A model of rectangles in which area is constrained by a derived attribute 199
B-12 A case for extracting an 'equation' HLC from a domain rule 200
B-13 A case for extracting a 'multiplier' HLC from an equation of the form $z = x * y$ 201
B-14 A case for extracting a 'multiplier' HLC from an equation of the form $x = z / y$ 202
B-15 Encoding that an attribute is mandatory by omitting the `OPTIONAL` keyword 204
B-16 Encoding that an attribute is mandatory by a domain rule 204
B-17 An example of a more complicated existence constraint 205
B-18 The existence constraint implied by an inverse attribute, expressed as an explicit rule 206
B-19 A record company's model of albums and their tracks 207
B-20 An alternative model of albums and their tracks 208
B-21 An explicit conjunction 209
B-22 An implicit conjunction 209

D-1 An entity template for visualising sources of constraints 250
D-2 Shallow indexing (hash table) case library organisation 255
D-3 `match`: the top-level case matching algorithm 261
D-4 `matchFrames`: algorithm to match a frame from the case to a frame from the retrieval prompt 262
D-5 `matchFillers`: algorithm to match a value from the case to a value from the retrieval prompt 264
D-6 `extract`: the top-level algorithm for post-match extraction 268
D-7 `extractFrame`: algorithm to reinstantiate a frame from the case's solution for the actual model 269
D-8 `extractFiller`: algorithm to process a value from the case's solution for the actual model 270

E-1 FIL augmented with HTML tags 293
E-2 Screenshots of the HTML files in Figure E-1, as rendered by Netscape Navigator 3.01 294

G-1 An adaptation case 324
G-2 An entity from the AP214 model 325
G-3 HLCs extracted from Figures B-1 and B-2 330