
Contents

1	Introduction	1
1.1	The Viewpoint Model of Software Systems Development	2
1.2	Integration of Specifications	5
1.2.1	Admissible Interpretations, Correspondences, and Consistency	10
1.2.2	Language- and Method-Independent Integration	13
1.3	Requirements of Reference Models and Their Usage	16
1.4	The Transformation Systems Reference Model	17
1.4.1	Transformation Systems	18
1.4.2	Development Operations and Relations	20
1.4.3	Composition	22
1.4.4	Granularity	25
1.5	Organisation of the Book	27
2	Transformation Systems	29
2.1	Transition Graphs and Data Spaces	30
2.2	Examples	38
2.3	Data Spaces from Other Specification Frameworks	49
2.4	Objects and Object References	52
2.5	Discussion	57
3	Specification of Properties	63
3.1	Data Space Specification	64
3.2	Control Flow Specification	71
3.3	Examples	78
3.4	Rewriting Algebras with Transformation Rules	85
3.5	Specification with Other Formulae	95
3.6	Discussion	99

4	Development of Transformation Systems	103
4.1	Development Operations	105
4.2	Extension and Reduction	112
4.3	Categorical Structure	119
4.4	Refinement and Implementation	122
4.5	Examples	132
4.6	Preservation of Properties	140
4.7	The Institution of Transformation Systems	144
4.8	Development w.r.t. Other Specification Frameworks	146
4.9	Discussion	157
5	Composition of Transformation Systems	161
5.1	Binary Composition via Connection Relations	163
5.2	Categorical Structure	174
5.3	Composition-by-Limits	180
5.4	Compositional Semantics	184
5.5	Compositionality of Properties	186
5.6	Compositionality of Developments	187
5.7	Morphisms of Transformation Systems with Distributed Data .	195
5.8	Construction of General Compositions by Global Limits	202
5.9	Sequential Composition	213
5.10	Composition w.r.t. Other Specification Frameworks	217
5.11	Discussion	224
6	Applications to UML Software Specifications	229
6.1	Class Diagram Semantics	230
6.1.1	Architecture: Class Graphs and Object Graphs	232
6.1.2	Internal Structure: Class Signatures and Object States .	236
6.1.3	Signature Diagrams and System States	238
6.1.4	A Language for Object Systems	242
6.1.5	Evaluation of Expressions	246
6.1.6	Further Static Features of Class Diagrams	249
6.1.7	State Transformations	250
6.2	State Machine Semantics	251
6.2.1	Control and Data States	252
6.2.2	Transitions and Transformations	254
6.3	Composition of State Machines	258
6.3.1	Asynchronous Communication	259
6.3.2	Synchronous Communication	261
6.4	Integration of Class Diagrams and State Machines	266
6.5	Sequence Diagram Semantics	268
6.6	Discussion	271

7	Conclusion	277
7.1	Summary	278
7.2	Further Developments and Applications	282
7.2.1	UML Integration	282
7.2.2	Integration Methods	285
7.2.3	Architecture Description	286
7.3	Related Approaches	287
7.3.1	Integration of Static States and Dynamic Changes	287
7.3.2	Categorical Composition of Theories and Models	291
7.3.3	Consistency and Integration of Viewpoint Specifications	293
7.3.4	Semantic Unification of Programming Languages	294
7.4	Methodological Remarks	296
A	Partial Algebras and Their Specification	303
	References	311
	Index	325