

# Contents

<b>1</b>	<b>The Science of Software Development</b>	<b>7</b>
1.1	Software Engineering as Theory Construction . . . . .	7
1.2	Software Engineering Paradigms . . . . .	9
1.3	The Path from Problem to Program . . . . .	15
1.4	Knowledge-Based Software Development . . . . .	19
<b>2</b>	<b>Knowledge Representation as a Basis of Specifying Requirements</b>	<b>28</b>
2.1	Demands on a Requirements Specification Language . . . . .	28
2.2	Languages to Formulate Requirements Specifications . . . . .	32
2.3	The FRORL Requirements Specification Language . . . . .	35
2.4	LSSGR Protocol Example . . . . .	41
2.5	Developing Requirements Specifications using FRORL . . . . .	45
<b>3</b>	<b>Nonmonotonic Logic Foundation of the Requirements Specification Language</b>	<b>50</b>
3.1	The Formal Foundation of FRORL . . . . .	50
3.2	Soundness and Completeness of FRORL . . . . .	58
3.3	Representing FRORL Constructs . . . . .	65
<b>4</b>	<b>A Requirements Specification Language for Real-Time Distributed Software Systems</b>	<b>70</b>
4.1	Characteristics of Real-Time Distributed Software Systems . . . . .	70
4.2	Demands on Requirements Specification Languages for Real-Time Distributed Systems . . . . .	71
4.3	Languages to Formulate Real-Time Requirements Specifications . . . . .	73
4.4	Modeling Mechanisms for Concurrent Distributed Systems . . . . .	78
4.5	Modeling Constructs for Real-Time Processes, Timing Constraints, and Temporal Properties . . . . .	81
<b>5</b>	<b>Temporal Logic Foundation of the Real-Time Distributed Requirements Specification Language</b>	<b>84</b>
5.1	The Temporal Fix-Point Calculus . . . . .	84
5.2	Model Checking . . . . .	89

5.3	Expressing the Temporal Aspects of a FRORL Specification . . . . .	91
<b>6</b>	<b>Verification of Requirements Specifications</b>	<b>95</b>
6.1	Analysis through Resolution Refutation . . . . .	95
6.2	Model Checking . . . . .	98
6.3	Timing Constraints Consistency Analysis . . . . .	109
<b>7</b>	<b>Development, Specification, and Verification of Knowledge-Based Systems</b>	<b>117</b>
7.1	Difficulties in Verifying Knowledge-Based Systems . . . . .	117
7.2	Correctness Problems in Knowledge Bases . . . . .	119
7.3	Approaches to the Verification of Knowledge-Based Systems . . . . .	121
7.4	Dynamic Verification . . . . .	126
<b>8</b>	<b>Knowledge-Based Implementation</b>	<b>136</b>
8.1	Automated Program Construction . . . . .	136
8.2	Canonicalization . . . . .	140
8.3	Data Dependency and Control Flow Analysis . . . . .	154
8.4	Determination of Execution Sequence . . . . .	163
8.5	Removal of Unnecessary Nondeterminacy . . . . .	167
<b>9</b>	<b>Specification Debugging</b>	<b>174</b>
9.1	Knowledge-Based Debugging . . . . .	174
9.2	Debugging of FRORL Specifications . . . . .	177
<b>A</b>	<b>Example Specifications</b>	<b>186</b>
A.1	Alternate Bit Protocol . . . . .	186
A.2	Subscriber-Line Controller of a Telephone Exchange . . . . .	189
<b>B</b>	<b>Formal Grammar of FRORL</b>	<b>197</b>
<b>C</b>	<b>Some Results about Fix-Points</b>	<b>200</b>
<b>D</b>	<b>References</b>	<b>203</b>
<b>E</b>	<b>Index</b>	<b>231</b>