

Claude Girault  
Rüdiger Valk

# Petri Nets for Systems Engineering

A Guide to Modeling, Verification,  
and Applications

With 190 Figures and 9 Tables



Springer

# Contents

<b>List of Authors and Affiliations . . . . .</b>	<b>XV</b>
---	-----------

<b>Introduction: Purpose of the Book . . . . .</b>	<b>1</b>
--	----------

---

## **Part I. Petri Nets — Basic Concepts**

---

<b>1. Introduction . . . . .</b>	<b>7</b>
<b>2. Essential Features of Petri Nets . . . . .</b>	<b>9</b>
2.1 Locality and Concurrency . . . . .	10
2.2 Graphical and Algebraic Representation . . . . .	12
2.3 Concurrency, Conflict, and Confusion . . . . .	15
2.4 Refinement and Composition . . . . .	18
<b>2.5 Net Morphisms . . . . .</b>	<b>23</b>
<b>3. Intuitive Models . . . . .</b>	<b>29</b>
3.1 Arc-Constant Nets . . . . .	29
3.2 Place/Transition Nets . . . . .	32
3.3 Coloured Nets . . . . .	34
3.4 Foldings . . . . .	38
<b>4. Basic Definitions . . . . .</b>	<b>41</b>
4.1 Formal Definition of Place/Transition Nets . . . . .	41
4.2 Formal Definition of Arc-Constant Nets . . . . .	43
4.3 Formal Definition of Coloured Nets . . . . .	45
<b>5. Properties . . . . .</b>	<b>53</b>
5.1 Basic Properties . . . . .	54
5.2 An Introduction to the Analysis . . . . .	58
5.2.1 Verification Based on the Reachability Graph . . . . .	60
5.2.2 Verification Based on Linear Invariants . . . . .	68
<b>6. Overview of the Book . . . . .</b>	<b>73</b>

---

**Part II. Modelling**

---

7. Introduction . . . . .	81
8. Modelling and Analysis Techniques by Example . . . . .	85
8.1 Nets, Refinement, and Abstraction . . . . .	85
8.2 Place/Transition Nets and Resource Management . . . . .	92
8.3 Coloured Nets, Abstraction, and Unfolding . . . . .	97
9. Techniques . . . . .	105
9.1 Building Blocks . . . . .	105
9.2 Combining Nets . . . . .	108
9.2.1 Place Fusion . . . . .	108
9.2.2 Arc Addition . . . . .	110
9.2.3 Transition Fusion . . . . .	111
9.3 High-Level Nets . . . . .	112
9.3.1 Coloured Nets . . . . .	<b>112</b>
9.3.2 Fairness, Priority, and Time . . . . .	<b>115</b>
9.4 Decomposing Nets . . . . .	116
9.5 Conclusion . . . . .	117
10. Methods . . . . .	119
10.1 State-Oriented Modelling . . . . .	120
10.1.1 Specification . . . . .	123
10.1.2 Design . . . . .	124
10.1.3 Implementation . . . . .	133
10.1.4 Conclusion . . . . .	134
10.2 Event-Oriented Modelling . . . . .	135
10.2.1 High-Level Modelling . . . . .	<b>135</b>
10.2.2 Protocol Modelling . . . . .	137
10.2.3 Verification . . . . .	142
10.2.4 Conclusion . . . . .	145
10.3 Object-Oriented Modelling . . . . .	146
10.3.1 Objects vs. Petri Nets . . . . .	146
10.3.2 Integration Approaches . . . . .	148
10.3.3 A Multi-Formalism Approach Including Nets . . . . .	152
10.3.4 Conclusion . . . . .	157
11. Case Studies . . . . .	159
11.1 State-Oriented Approach . . . . .	162
11.1.1 Specification . . . . .	162
11.1.2 Design . . . . .	162
11.1.3 Implementation . . . . .	166
11.2 Event-Oriented Approach . . . . .	166

<b>11.2.1</b>	Modelling a Node	166
11.2.2	Verification	170
11.2.3	Adding Colour	172
11.3	Object-Oriented Approach	173
11.3.1	Structure of the NodeCoordinator	173
11.3.2	The NodeCoordinator in OF-Class Formalism	175
11.3.3	Net Synthesis from the NodeCoordinator Specification	176
11.3.4	Verification of Protocol Correctness	176
<b>12.</b>	<b>Conclusion</b>	179

---

### Part III. Verification

---

<b>13.</b>	<b>Introduction: Issues in Verification</b>	183
13.1	Classification of Nets	184
<b>13.1.1</b>	Restriction of Nets	184
13.1.2	Extension of Nets	184
13.1.3	Abbreviation of Nets	185
13.1.4	Parametrisation of Nets	187
13.2	Properties	188
13.3	Classification of Methods	190
13.4	Verification Process	197
13.5	Overview	199
<b>14.</b>	<b>State-Space-Based Methods and Model Checking</b>	201
<b>14.1</b>	Properties, Temporal Logic, and Fairness	202
14.1.1	The Temporal Logic CTL*	204
14.2	On-the-Fly Approaches	215
14.3	Partial-Order-Based Approaches	218
14.3.1	Traces and Verification Issues	219
14.3.2	Persistent Set Searches	223
14.3.3	Sleep Set Searches	227
14.3.4	Covering Step Graphs	229
14.3.5	Branching Process Techniques	231
14.3.6	Conclusion	239
14.4	Symbolic and Parametrised Approaches	241
<b>14.4.1</b>	Symbolic Reachability Graph	241
14.4.2	Symmetries in Nets	259
14.4.3	Parametrised Reachability Graph	260
14.5	Implementation Issues	266
14.5.1	State-Space Caching	267
14.5.2	Hashing Compaction	267
14.5.3	Boolean Manipulation	268
14.5.4	Symbolic Model Checking	273

14.5.5	Concluding Remarks on Implementation Issues. . . . .	273
14.6	Synthesis and General Concluding Remarks. . . . .	274
<b>15.</b>	<b>Structural Methods</b> . . . . .	277
15.1	Net System Reductions. . . . .	278
15.1.1	A Basic Kit of Reduction Rules. . . . .	280
15.1.2	Implicit Places. . . . .	281
15.2	Linear Algebraic <b>Techniques</b> . . . . .	285
15.2.1	Bounds and Boundedness. . . . .	287
15.2.2	Deadlock-Freeness and Liveness. . . . .	288
15.2.3	Structural Liveness and <b>Liveness</b> . . . . .	292
15.2.4	Reversibility and Liveness. . . . .	295
15.3	Siphons and Traps. . . . .	297
15.4	Analysis of Net Subclasses. . . . .	299
15.4.1	Some Syntactical Subclasses. . . . .	300
15.4.2	Fairness and Monopolies. . . . .	303
15.4.3	Confluence and Directedness. . . . .	304
15.4.4	Reachability and <b>the State Equation</b> . . . . .	306
15.4.5	Analysis of Liveness and Boundedness. . . . .	306
15.5	Invariants and Reductions for Coloured Petri Nets. . . . .	307
15.5.1	Invariants. . . . .	307
15.5.2	Reductions. . . . .	312
<b>16.</b>	<b>Deductive and Process-Algebra-Based Methods</b> . . . . .	317
<b>16.1</b>	<b>A Rewriting Semantics for Algebraic Nets</b> . . . . .	318
16.1.1	Algebraic Specifications. . . . .	320
16.1.2	Rewriting Specifications. . . . .	324
16.1.3	Algebraic Nets. . . . .	328
16.1.4	<b>Rewriting Semantics</b> . . . . .	334
16.1.5	Final Remarks. . . . .	337
16.2	Assertional Reasoning. . . . .	338
16.2.1	State Predicates and Functions. . . . .	341
16.2.2	Basic Assertions. . . . .	341
16.2.3	Safety Assertions. . . . .	342
16.2.4	Liveness Assertions. . . . .	347
16.2.5	Elementary Compositkmality. . . . .	351
16.2.6	A Simple Example. . . . .	353
16.2.7	Extensions of the Logic. . . . .	359
16.2.8	Combination with Other Methods. . . . .	360
16.2.9	Final Remarks. . . . .	361
16.3	A Logic of <b>Enablement</b> . . . . .	361
16.3.1	Morphisms, Reductions, and Simulation. . . . .	362
16.3.2	A Temporal Logic for Nets. . . . .	364
16.3.3	The Concept of a Test Net. . . . .	367
16.3.4	Example: Mutex. . . . .	368

16.4	Linear Logic and Petri Nets. . . . .	370
16.4.1	Basic Relationship. . . . .	371
16.4.2	Specification of Net Properties. . . . .	374
16.4.3	Linear Logic for Representation of Coloured Nets. . . . .	375
16.4.4	The Principle of Backward Reasoning . . . . .	377
16.4.5	<b>Nondeterministic</b> Transitions. . . . .	377
16.4.6	Bibliographic Remarks. . . . .	381
16.5	Verifying Petri Net Models Using Process Algebra . . . . .	382
16.5.1	Method. . . . .	382
16.5.2	Hierarchical Place/Transition Nets. . . . .	384
16.5.3	A Brief Introduction to Process Algebra. . . . .	385
16.5.4	The Production Unit . . . . .	388
16.5.5	Concluding Remarks. . . . .	397
<b>17.</b>	<b>Conclusion</b> . . . . .	<b>399</b>

---

**Part IV. Validation and Execution**

---

<b>18.</b>	<b>Introduction</b> . . . . .	<b>403</b>
<b>19.</b>	<b>Systems Engineering and Validation</b> . . . . .	<b>405</b>
19.1	Software Life-Cycle and Validation. . . . .	405
19.2	Validation. . . . .	406
19.3	Prototyping as an Approach. . . . .	408
19.3.1	The Original Problems. . . . .	408
19.3.2	Prototyping Taxonomy. . . . .	409
19.3.3	Key Issues in Prototyping. . . . .	410
19.3.4	Extended Definition of Prototyping . . . . .	411
19.4	Tools. . . . .	412
<b>20.</b>	<b>Net Execution</b> . . . . .	<b>417</b>
20.1	Centralised Control. . . . .	420
20.2	Distribution of Control over Places. . . . .	421
20.3	Distribution of Control over Edges. . . . .	427
20.4	Multithreading and Synchronisation . . . . .	428
20.5	Asynchrony. . . . .	429
20.6	Conclusion . . . . .	431
<b>21.</b>	<b>Code Generation</b> . . . . .	<b>133</b>
21.1	Petri Net Approaches to Code Generation. . . . .	435
21.1.1	State of the Art. . . . .	435
21.1.2	Parallel Interpretation of a Petri Net . . . . .	437
21.2	A Petri Net Partitioning Algorithm. . . . .	440
21.2.1	Transformation into a Structural Model. . . . .	440
21.2.2	Computation and Selection of Positive Place Invariants 441	441

21.2.3	Evaluation of Partitioning Properties. . . . .	442
21.2.4	Computation of Prototype Objects. . . . .	443
21.2.5	Speeding Up the Algorithm. . . . .	445
21.2.6	Net Transformation When the Algorithm Fails. . . . .	446
21.3	Some Aspects of Code Generation from Petri Nets. . . . .	448
21.3.1	On the Implementation of Prototype Objects. . . . .	448
21.3.2	Prototype and Execution Environment. . . . .	453
21.3.3	Mapping Processes onto a Given Architecture. . . . .	454
21.3.4	Place Invariants and Pipeline Detection. . . . .	459
21.4	Code Generation from a High-Level Net. . . . .	461
21.4.1	Association with a High-Level Formalism. . . . .	462
21.4.2	An Example of Work Based on a Pre-Existing High-Level Formalism. . . . .	462
21.4.3	An Example of a High-Level Formalism Dedicated to Code Generation: H-COSTAM. . . . .	463
21.4.4	Implementation of Enhanced Prototype Objects. . . . .	466
21.5	Conclusion. . . . .	467
<b>22.</b>	<b>Conclusion. . . . .</b>	<b>469</b>

---

## Part V. Application Domains

---

<b>23.</b>	<b>Introduction. . . . .</b>	<b>473</b>
23.1	Putting Petri Nets to Work. . . . .	473
23.2	Domains of Application. . . . .	474
23.2.1	Manufacturing. . . . .	474
23.2.2	Workflow Management. . . . .	475
23.2.3	Telecommunications. . . . .	475
23.2.4	Other Application Domains. . . . .	476
<b>24.</b>	<b>Flexible Manufacturing Systems. . . . .</b>	<b>479</b>
24.1	A Brief Overview of the Domain. . . . .	479
24.2	Using Petri Nets in FMS. . . . .	484
24.3	A Design Approach. . . . .	490
24.3.1	An Intuitive Introduction to a Class of Nets. . . . .	490
24.3.2	Automation of the Modelling Process. . . . .	494
24.3.3	Using Structural Analysis for System Control. . . . .	497
24.4	Conclusion. . . . .	505
<b>25.</b>	<b>Workflow Systems. . . . .</b>	<b>507</b>
25.1	An Overview of the Domain. . . . .	507
25.2	Motivation. . . . .	510
25.2.1	Formal Language. . . . .	512
25.2.2	Analysis Techniques. . . . .	513

25.3	Design Methodology . . . . .	513
25.3.1	Tasks and Transitions . . . . .	516
25.3.2	Logistics and Transitions . . . . .	517
25.3.3	Case and Tokens . . . . .	520
25.3.4	Case Study: Justice Department . . . . .	520
25.3.5	Business Process Definition . . . . .	522
25.4	Workflow Analysis . . . . .	523
25.4.1	Structural Analysis . . . . .	524
25.4.2	Dynamic Analysis . . . . .	530
25.5	Lessons Learned: The Sagitta-2000 Case . . . . .	537
25.6	Conclusion . . . . .	539
26.	Telecommunications Systems . . . . .	541
26.1	Overview of the Domain . . . . .	541
26.1.1	The IN Architecture . . . . .	542
26.1.2	The IN Service Processing . . . . .	543
26.1.3	Conclusion . . . . .	545
26.2	Motivation . . . . .	545
26.3	Design Methodology . . . . .	547
26.3.1	The OF-Class Model of a Basic Telecommunications System . . . . .	547
26.3.2	The OF-Class Model of a CFU Telecommunications System . . . . .	553
26.3.3	The <b>OF-Class</b> Model of an IN Telecommunications System . . . . .	557
26.3.4	From OF-Class to OF-CPN: The Principles of the Transformation . . . . .	559
26.3.5	From OF-Class to OF-CPN: Illustration of the Transformation . . . . .	560
26.4	Analysis . . . . .	561
26.4.1	Overview of Analysis with Petri Nets in the Area of Telecommunications Systems . . . . .	561
26.4.2	Analysis of the IN Model: Detection of Feature Interaction . . . . .	562
26.5	<b>Conclusion</b> . . . . .	566
27.	<b>Conclusion</b> . . . . .	567
27.1	Common Modelling Problems . . . . .	567
27.2	Shared Analysis Results . . . . .	568
	<b>References</b> . . . . .	571
	<b>Index</b> . . . . .	601