FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Department of Computer Science & Applied Mathematics

Weizmann Institute

Department of Mathematics

Biography

Research

Publications

Home

D. Harel and M. Politi, *Modeling Reactive Systems with Statecharts: The STATEMATE Approach,* (with M. Politi), McGraw-Hill, 1998.

(Early version titled: *The Languages of STATEMATE*, I-Logix, Inc., Andover, MA, 1991.)

This book is no longer in print at McGraw-Hill. It can be downloaded in its entirety here: Download Book

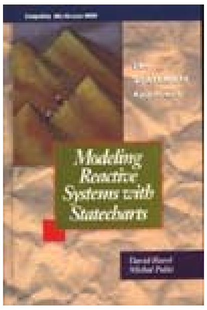

Table of Contents
Preface

~ Amazon
~ Barnes and Noble

Table of Contents

Chapter 1. Introduction

1.1 System Development and Methodologies
1.2 Modeling Reactive Systems
1.3 The Modeling Languages
1.4 The Statemate Toolset

Chapter 2. The Functional View: Activity-Charts

2.1 Functional Description of a System
2.2 Activities and Their Representation
2.3 Flow of Information Between Activities
2.4 Describing the Behavioral Functionality of Activities
2.5 Connectors and Compound Flow-Lines

Chapter 3. Information Elements

3.1 Information Elements in the Model
3.2 Events
3.3 Conditions
3.4 Data-Items
3.5 User-Defined Types

Chapter 4. The Behavioral View: Statecharts

4.1 Behavioral Description of a System
4.2 Basic Features of Statecharts
4.3 The Hierarchy of States

4.4 Orthogonality
4.5 Connectors and Compound Transitions
4.6 More About Transitions

Chapter 5. The Textual Expression Language

5.1 Event, Condition and Data-Item Expressions
5.2 Actions
5.3 Time-Related Expressions
5.4 Static Reactions

Chapter 6. The Semantics of Statecharts

6.1 Execution of the Model
6.2 Handling Time
6.3 Nondeterministic Situations

Chapter 7. Connections Between the Functional and Behavioral Views

7.1 Dynamics in the Functional Decomposition
7.2 Dynamics of Activities
7.3 Controlling the Activities
7.4 Specifying Behavior of Basic Activities

Chapter 8. Communication Between Activities

8.1 Communication and Synchronization Issues
8.2 Controlling the Flow of Information
8.3 Examples of Communication and Synchronization Issues

8.4 Activities Communicating Through Queues

Chapter 9. The Structural View: Module-Charts

9.1 Structural Description: High-Level Design
9.2 Internal and External Modules
9.3 Communication Lines Between Modules
9.4 Connectors and Compound Flow-Lines

Chapter 10. Connections Between the Functional and Structural Views

10.1 Relating the Functional and Structural Models
10.2 Activity-Chart Describing a Module
10.3 Activities Implemented by Modules
10.4 Activities Associated with a Module's Activities
10.5 Object-Oriented Analysis with Module-Charts

Chapter 11. Splitting Up Charts

11.1 Separating a Chart into Multiple Pages
11.2 Offpage Charts
11.3 Connecting Offpage Charts by Matching Flows
11.4 Connecting Offpage Statecharts Using Connectors

Chapter 12. Putting Things Together

12.1 Relationships Among the Three Kinds of Charts
12.2 A Chart in a Model

12.3 Hierarchy of Charts
12.4 Entities External to the System under Description

Chapter 13. Scope and Resolution of Elements

13.1 Visibility of Elements and Information Hiding
13.2 Defining, Referencing and Resolving Elements
13.3 The Scope of Charts and Graphical Elements
13.4 The Scope of Textual Elements
13.5 Global Definition Sets (GDSs)

Chapter 14. Generic Charts

14.1 Reusability of Specification Components
14.2 Definition and Instances of Generic Charts
14.3 Parameter of Generic Charts
14.4 Referring to Elements in Instances

Chapter 15. Related Approaches

15.1 An Overview of Specification Methods
15.2 Methods Based on Structured Analysis
15.3 ECSAM
15.4 Methods Based on Objects
15.5 MIL-STD-498 (DOD-STD-2167A)

Chapter 16. Transition to Design

16.1 Statemate Models in the Development Process
16.2 Mapping Models to Design Structures

Appendix A. Names and Expressions
A.1 Names
A.2 Expressions
A.3 Predefined Functions
A.4 Reactions and Behavior of Activities
A.5 Flow of Information

Appendix B. Early Warning System Example: Functional Decomposition Approach

B.1 Textual Description of the System
B.2 The Model

References

Back to Top

Preface

This book provides a detailed description of a comprehensive set of languages for modeling reactive systems. The approach is dominated by the language of Statecharts, which is used to describe behavior, combined with Activity-charts, which are used for describing the system's activities (i.e., its functional building blocks, capabilities, and objects) and the data that flows between them. These two languages are used to develop a conceptual model of the system, which can be combined with the system's physical, or structural, model described in our third language, Module-charts. These three languages are highly diagrammatic in nature, constituting full-fledged visual formalisms, complete with rigorous semantics. They are accompanied by a Data Dictionary for specifying additional parts of the model that are textual in nature.

The approach described here lies at the heart of the Statemate system, which the authors have helped design and build at I-Logix, Inc. since 1984. Statemate is most beneficial in requirements analysis, specification, and high-level design. In addition to supporting the modeling effort using the aforementioned language set, Statemate provides powerful tools for inspecting and analyzing the resulting models, via model execution, dynamic testing, and code synthesis.

This book discusses the modeling languages in detail, with an emphasis on the language of Statecharts, because it is the most important and intricate language in the set and the most novel. Statecharts are used to specify the behavior of activities, whether they represent functions in a functional decomposition or objects in an object decomposition. We describe the syntax in a precise and complete manner and discuss the semantics in a way that is intended to render the model's behavior clear and intuitive. Our presentation is illustrated extensively with examples, most of which come from a single sample model of an early warning system (EWS). Appendix B provides a summarized description of this model.

Whenever possible, we have tried to explain our motivation in including the various features of the languages. We also provide hints and guidelines on such methodological issues as decomposition criteria and the order in which charts are to be developed.

While we do provide a brief description of the Statemate system in Sec. 1.4, this book is not intended to be a user manual for it but, rather, a definitive description of its languages and a guide to their use. For more on Statemate's capabilities, we refer the reader to the documentation supplied by I-Logix, Inc.

This book should be of interest to a wide variety of systems developers (both in software and hardware) and to teachers and students of software
and hardware engineering.

Back to Top