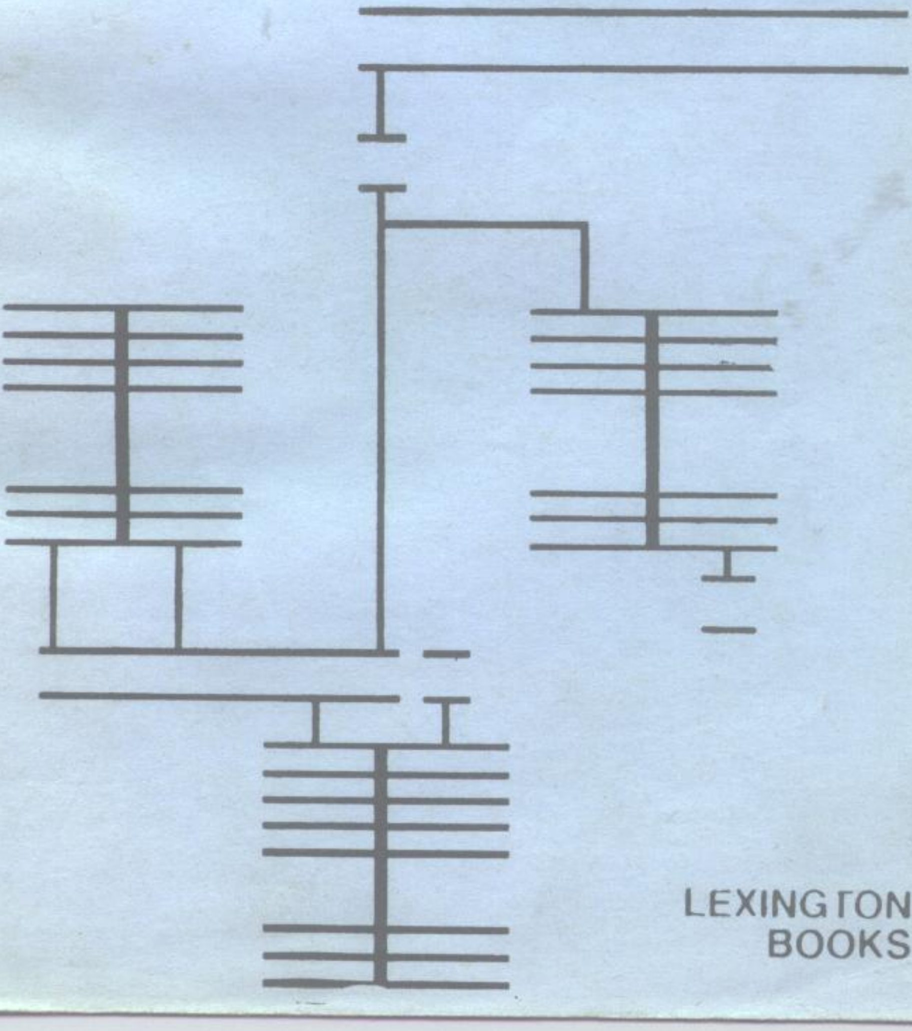
Yaohan Chu

# Software Blueprint and Examples



LEXINGTON
BOOKS

73.87221
C559

# Software Blueprint and Examples

**Yaohan Chu**
University of Maryland

中国科学院电子学研究所
1984 10
图书館藏

**LexingtonBooks**
D.C. Heath and Company
Lexington, Massachusetts
Toronto

5506615

DR87/92

**Library of Congress Cataloging in Publication Data**

Chu, Yaohan, 1920-
Software blueprint and examples.

Includes bibliographical references and index. 1. Electronic digital computers—Programming.
I. Title.

| | | |
|---|---|---|
| QA76.6.C457 | 001.64′2 | 81-48268 |
| ISBN 0-669-05329-5 | | AACR2 |

*Copyright © 1982 by D.C. Heath and Company*

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage or retrieval system, without permission in writing from the publisher. The author retains translation rights to the following languages: Chinese, Spanish, Polish, Hungarian, Rumanian, Japanese, and Russian.

Published simultaneously in Canada

Printed in the United States of America

International Standard Book Number: 0-669-05329-5

Library of Congress Catalog Card Number: 81-48268

# Preface and Acknowledgments

Software engineering is the application of engineering concepts, techniques, methodologies, principles, and practices to the development of software. A software blueprint is an application of the engineering blueprint to software. In traditional engineering methodology and practice, the engineer first develops a blueprint that describes the design of a product completely and understandably; the engineer, or another, then constructs the product according to the blueprint. This long-practiced methodology contrasts with the current practice of student programmers, who attempt to write code in the early stages of development without fully designing the software first.

The application of engineering methods and practices to software development implies (1) the separation of software design from software implementation and (2) a software-blueprint interface between software design and software implementation. In the nine chapters of this book, the methodology of the software blueprint is presented, software design is introduced, and six examples of the software blueprint are shown.

Many of the serious problems of today's software development can be attributed to the lack of a readable, concise, understandable, and complete design description. The software blueprint has been developed to be such a description. Chapter 1 presents the software blueprint methodology. A unique feature of this methodology is the design of software in three levels, described by corresponding software blueprints in three levels. These three levels are structured hierarchically and described in a uniform syntax. The description of a lexical scanner design is chosen to illustrate the software blueprint.

Chapter 2 first introduces the software development life-cycle, and then develops software design by taking an engineering approach. With this approach, five subjects are identified: software elements, software modules, software architecture, software design, and software design description. These subjects are analogous to the five subjects of computer design: computer elements, computer modules, computer architecture, computer design, and computer design description. This engineering approach gives a unified approach to the design of computer hardware and software. Software elements are identified by analyzing existing software; this is the first level of the formulation. Software elements are selected and organized into modules; this is the second level of the formulation. A software architecture is built up to represent a software organization; this is the third level of the formulation. A software design creates a software organization to meet the design specifications; this is the fourth level of the formulation. After the software design is completed, it needs to be formally

5506615

but understandably described; this is the fifth level of the formulation. These five subjects—software elements, software modules, software architecture, software design, and software design description—are presented in detail.

Chapter 3 presents the design of an interactive direct-execution programming system. In a direct-execution programming system, the tokens of a high-level-language program are scanned and then directly executed token by token without compilation, linkage editing, or loading. This chapter first introduces three types of interactive programming systems. It next presents the conceptual design of the direct-execution programming system and shows the designs of the monitor, the text editor, the input processor, and the direct-execution processor. The monitor is the supervisor that communicates with the user by means of four monitor commands. It manages the text editor, the input processor, and the direct-execution processor. The text editor has six edit commands, and the input processor allows the user to input a source program from the terminal. The direct-execution processor, the heart of the system, accepts a small subset of ALGOL-60. The design is described by a level-A blueprint and a level-B blueprint.

Chapter 4 applies the structured-design technique to the design of an interactive text editor. The text consists of fixed-length lines organized into a doubly linked list structure. This text editor operates in two modes—input and edit—and includes a limited set of eleven edit commands. It has been broken down into twenty-two procedures organized in three levels. This structured design is described by a level-A blueprint and a level-C blueprint.

Chapter 5, a revised version of a file system originally written by Edward Raymond Cannon (a graduate of University of Maryland), presents the design of a disk-pack file system intended for a single user. This chapter describes the five file statements, the data structures and their formats, and the twenty-two procedures of the system. The level-A design is first developed and presented in a level-A software blueprint. The level-B design is then developed and presented in a level-B software blueprint. These two levels are designed and described using many diagrams.

The designs in chapters 6, 7, and 8 involve process synchronization, message communication, hardware/software interaction, and the like. Chapter 6 presents the design of an I/O system involving device drivers and interrupt handlers. The drivers and handlers are hardware dependent (they were designed for a PDP-11 computer system). These device drivers and interrupt handlers are used again in the designs in chapters 7 and 8. Software design language SDL-2, which will have a syntax compatible with SDL-1, is being developed for describing these constructs. Some constructs of SDL-2 are introduced and used in these three chapters.

Chapter 7 presents the design of an output spooling system that spools data records from a disk to a printer. This chapter introduces a buffer pool, buffer queues, and semaphore queues. Included in the design are the selected

此为试读，需要完整PDF请访问： www.ertongbook.com

data structures, which include process control blocks, descriptors, semaphores, and I/O buffers. Four system processes are described: the disk buffer process, the printer buffer process, the sleep process, and the quit process. The scheduler, trap handler, device drivers, and interrupt handlers are designed. The design is intended for implementation on a PDP-11 computer system.

Chapter 8 sketches the design of a simple multiprogramming system making use of the I/O system of chapter 6 and the output spooling system of chapter 7. By using the process flow trace technique, the multiprogramming of processes can be observed in detail. The systems in chapter 6, 7, and 8 were projects in the senior course on operating systems at the Department of Computer Science at the University of Maryland.

Chapter 9 presents the software-design language SDL-1. This language is developed in concert with a formulation of software design given in chapter 2. SDL-1 was developed, above all, with design understandability in mind. Particular attention was paid to the visibility of the overall design structure. SDL-1 has constructs delineating data, control data, procedures, and reference structures. The control constructs and the design structure are simple, but the data constructs are rich in data types, data structures, and data operations. There are two types of statements; control flow statements and data flow statements. These describe the control flow and data flow of a software design. The syntax and semantics of SDL-1 are so formulated that the resulting design is a structured design.

This book presents an engineering (as opposed to a mathematical) approach to the design and description of software. The seven examples provided are of significant size and complexity. They are relevant and pedagogical, and they constitute a major part of the book. Implementations of these software designs in certain programming languages may become available.

*Software Blueprint and Examples* can be used in several areas of study. It can be used in colleges and universities as a textbook in a senior course on software design, or in a second- or third-year programming course in which the software blueprints are used for programming projects. The examples presented in this book can be used as software implementation projects in courses on subjects such as compilers, interpreters, systems programming, and operating systems. It can also serve as a textbook in a software-engineering laboratory or as a self-study reference for programmers in the software industry who are relatively new to software engineering.

## Acknowledgments

I wish to acknowledge that the design examples in chapters 6, 7, and 8 were originated by Dr. Virgil Gligor of the Department of Computer Science of

the University of Maryland. I also wish to acknowledge the contributions made by many of my students and to express my appreciation to Carmen T. Radelat and Robin Endelman for their assistance in preparing the manuscript.

# Contents

List of Figures and Tables xi

Preface and Acknowledgments xvii

Chapter 1 Software Blueprint 1

1.1 Software Engineering 1
1.2 Software Blueprint Methodology 4
1.3 Specification of a Lexical Scanner 10
1.4 Level-A Design 14
1.5 Level-B Design 20
1.6 Level-C Design 25
1.7 Concluding Remarks 29
1.8 References 29
Problems 31

Appendix 1A—Software Blueprint:
Level-A Design of the Lexical Scanner 33
Appendix 1B—Software Blueprint:
Level-B Design of the Lexical Scanner 37
Appendix 1C—Software Blueprint:
Level-C Design of the Lexical Scanner 45

Chapter 2 A New Approach to Software Design 53

2.1 Introduction 53
2.2 Software Elements 55
2.3 Software Modules 62
2.4 Software Architecture 72
2.5 Software Design 78
2.6 Software Design Descriptions 86
2.7 Concluding Remarks 89
2.8 References 90
Problems 93

Chapter 3 An Interactive Direct-Execution System 95

3.1 Interactive Systems 95
3.2 Design Specification 99
3.3 Level-A Design 102
3.4 Level-B Design for MONITOR 111
3.5 Level-B Design for Text Editor 114
3.6 Level-B Design for Input Processor 117

3.7 Level-B Design for Direct-Execution Processor 119
3.8 References 129
Problems 131

Appendix 3A—Software Blueprint: Level-A Design for an Interactive Direct-Execution System 135
Appendix 3B—Software Blueprint: Level-B Design for an Interactive Direct-Execution System 141

**Chapter 4** **An Interactive Text Editor** 167

4.1 Design Specification 167
4.2 Level A-Design 174
4.3 Level-C Design: Data Elements 182
4.4 Level-C Design: Procedures 184
4.5 Level-C Design: Reference Structure 195
4.6 Level-C Software Blueprint 197
4.7 References 197
Problems 199

Appendix 4A—Software Blueprint: Level-A Design of an Interactive Text Editor 201
Appendix 4B—Software Blueprint: Level-C Design of an Interactive Text Editor 205

**Chapter 5** **A File System** *Edward Raymond Cannon* and *Yaohan Chu* 239

5.1 Introduction 239
5.2 Design Specification 240
5.3 Level-A Design 243
5.4 Level-B Design 252
5.5 Level-B Software Blueprint 274
5.6 Special Classes of File Systems 280
5.7 Concluding Remarks 281
5.8 References 281
Problems 283

Appendix 5A—Software Blueprint: Level-A Design for a Disk-Pack File System 285
Appendix 5B—Software Blueprint: Level-B Design for a Disk-Pack File System 291

Chapter 6 **Design Example of an I/O System** 319

6.1 A PDP-11 Computer System 319
6.2 Design Specification 323
6.3 Level-A Design 323
6.4 Level-B Design 330
6.5 Interrupt Flow Trace 339
6.6 References 342

Appendix 6A—Software Blueprint: Level-A Design of an I/O System 345
Appendix 6B—Software Blueprint: Level-B Design of an I/O System 349

Chapter 7 **An Output Spooling System** 359

7.1 Design Specification 359
7.2 Level-A Design 359
7.3 Level-B Design 379
7.4 Reference Structures 386
7.5 Process Flow Trace 387
Problems 395

Appendix 7A—Software Blueprint: Level-A Design of an Output Spooling System 397
Appendix 7B—Software Blueprint: Level-B Design of an Output Spooling System 403

Chapter 8 **A Simple Multiprogramming System** 419

8.1 Design Specification 419
8.2 Level-A Design 420
8.3 Processes 425
8.4 Trap Handler 431
8.5 Process Flow Trace 435
8.6 References 438
Problems 439

Appendix 8A—Software Blueprint: Level-B Design of a Multiprogramming System 441

Chapter 9 **Software-Design Language** 467

9.1 Introduction 467
9.2 Design Structure 469

9.3 Control Flow 470
9.4 Control Data Statements 479
9.5 Data Elements 482
9.6 Data Flow 486
9.7 Writing Style and Design Practice 488
9.8 Concluding Remarks 492
9.9 References 493
Problems 495

Appendix 9A—Syntax Description of SDL-1 497

**Bibliography** 503

**Index** 513

**About the Author**

此为试读,需要完整PDF请访问: www.ertongbook.com

# List of Figures and Tables

Figures

1-1 Engineering Blueprint: A Description of an Engineering Design 2

1-2 Software Blueprint: A Description of a Software Design 3

1-3 Examples of Data Structure Visibility 7

1-4 Examples of Control Structure Visibility 8

1-5 Level Numbers and Structure Reserved Words for Design Structure Visibility 9

1-6 Input String and Output String of a Lexical Scanner 11

1-7 BNF Description of the Lexical Rules of MICRO-ALGOL 14

1-8 Data Flow of the Lexical Scanner (Level-A Design) 15

1-9 Data Flow of the Lexical Scanner (Level-B Design) 21

1-10 Data Elements of the Lexical Scanner (Level-C Design) 26

2-1 Types of Software Elements 56

2-2 Types and Structures of Lexical Elements 57

2-3 Types and Structures of Data Elements 58

2-4 Control Data Elements 61

2-5 Types and Structures of Control Elements 61

2-6 Data Type Operations 63

2-7 Data Structure Operations 65

2-8 Control Data Operations 67

2-9 Control Operations 68

2-10 Structured Control Operations 70

| | | |
|---|---|---|
| 2-11 | Architectural Diagram of the Lexical Scanner Design, Showing the Data Elements and a Snapshot of Data Flow | 73 |
| 2-12 | Interprocedure Structure Diagram with Abnormal Returns | 74 |
| 2-13 | Interprocedure Structure Diagram with Direct and Indirect Recursion | 74 |
| 2-14 | Procedure Reference Structure of the Lexical Scanner Design | 75 |
| 2-15 | Data Reference Structure of the Lexical Scanner Design | 76 |
| 2-16 | Switch Reference Structure of the Lexical Scanner Design | 77 |
| 2-17 | Types of Interprocedure Structures | 82 |
| 2-18 | HIPO Charts | 88 |
| 3-1 | Terminal Output of an Example Session | 101 |
| 3-2 | Organization of an Interactive Direct-Execution System Showing Data Flow | 103 |
| 3-3 | System Control-Flow of the Interactive Direct-Execution System | 105 |
| 3-4 | Data Elements of the Interactive Direct-Execution System | 112 |
| 3-5 | Data Elements of the Text Editor | 114 |
| 3-6 | Data Elements of the Direct-Execution Processor | 121 |
| 3-7 | A Cross Section of the Procedure Reference Structure for the Direct-Execution Processor | 126 |
| 3-8 | A Second Cross Section of the Procedure Reference Structure for the Direct-Execution Processor | 127 |
| 3-9 | A Third Cross Section of the Procedure Reference Structure for the Direct-Execution Processor | 128 |
| 3-10 | Syntax Description of MICRO-ALGOL Language | 132 |
| 4-1 | A Terminal Session Using the Interactive Text Editor | 172 |
| 4-2 | Text-File Organization | 176 |
| 4-3 | Data Elements of the Text Editor | 178 |

4-4 Data Elements of the Text Editor 183

4-5 Flowchart for the Main Procedure of the Text Editor 186

4-6 Flowchart of Procedure INPUT 188

4-7 Flowchart of Procedure GO_EDIT 189

4-8 Flowchart of Procedure CONNECT 190

4-9 Flowchart of Procedure INSERT_CHAIN 191

4-10 Pointer Manipulation for Procedure INSERT_CHAIN 192

5-1 Drawing of a Disk Pack 240

5-2 Structure of the Five File-System Procedures 243

5-3 Format of the File Information Table (FIT) 244

5-4 FILE_MAP Format 245

5-5 Flowchart of Procedure FILE_DECLARE (Level-A Design) 245

5-6 Flowchart of Procedure FILE_OPEN (Level-A Design) 246

5-7 Flowchart of Procedure FILE_READ (Level-A Design) 249

5-8 Flowchart of Procedure FILE_WRITE (Level-A Design) 250

5-9 Flowchart of Procedure FILE_CLOSE (Level-A Design) 251

5-10 Procedures and Data Storage Areas 253

5-11 Table FILE Format 254

5-12 Table APT Format 255

5-13 Table VTOC Format and One User's File 256

5-14 File AVAIL_DISK Format 257

5-15 Flowchart of Procedure FILE_DECLARE (Level-B Design) 258

5-16 Flowchart of Procedure FILE_OPEN (Level-B Design) 259

5-17 Flowchart of Procedure VTOC_SEARCH (Level-B Design) 260

5-18 Flowchart of Procedure CHECK__PACK (Level-B Design) 260

5-19 Disk Pack Label 261

5-20 Flowchart of Procedure FILE__READ (Level-B Design) 261

5-21 Flowchart of Procedure ADDRESS__CONVERT (Level-B Design) 262

5-22 Format of User File on Disk 263

5-23 Formats of FILE Table in Main Memory and User File on Disk 263

5-24 Flowchart of Procedure FILE__WRITE (Level-B Design) 264

5-25 Flowchart of Procedure ALLOCATE__TRACK (Level-B Design) 265

5-26 Flowchart of Procedure FILE__CLOSE (Level-B Design) 266

5-27 Flowchart of Procedure DFILE__ENTER (Level-B Design) 266

5-28 Flowchart of Procedure VTOC__DELETE (Level-B Design) 267

5-29 Data Storage Areas of the File System 268

5-30 Disk Format 269

5-31 Volume Table of Contents (VTOC) Format 270

5-32 Table Disk File (DFILE) Format 271

5-33 File AVAIL__DISK Format 271

5-34 Format of Active Pack Table (APT) Entry 272

5-35 Table FILE Format 273

6-1 Configuration of a PDP-11 Computer System 320

6-2 Formats for Peripheral Device Registers 322

6-3 Data Flow of the I/O System 324

6-4 Interprocedural Structure Diagram 337

7-1 An Output Spooling System 360

7-2 An Output Spooling System Showing Queues 362

7-3 Queue DBREQ after Initialization 362

7-4 Queue EMPTYQ and Pool Buffers 363

7-5 Queue FULLQ 364

7-6 Queue READYQ 364

7-7 Queue AVAILQ 364

7-8 An Output Spooling System Showing Queues and Semaphores 365

7-9 Two Elements of a Semaphore 365

8-1 A Multiprogramming System 420

8-2 A Multiprogramming System Showing System Processes and Queues 422

9-1 Level Numbers and Structure Reserved Words of an SDL-1 Design 469

9-2 An Example of a Procedure Reference Structure with Recursive Calls 471

9-3 A Partial Procedure Reference Structure with an Unwind Return 473

9-4 An Example of a Procedure Definition 475

9-5 Examples of the Case Statement 476

9-6 An Example of a Loop Statement with a Simple Loop Structure 477

9-7 An Example of a Loop Statement with a Loop-Until Structure 478

9-8 An Example of the Unwind Statement 479

9-9 An Example of a Switch Declaration and a Switch Reference Structure 480

9-10 Examples of Primitive and Structured Data Types 483

9-11 Examples of Data Declarations 484

9-12 Examples of Data Declaration Elements 485

9-13 Examples of Expressions 488

9-14 Data-Flow Statements 489

9-15 An Example of a Nested Control Structure 490

9-16 An Example Showing the Use of the Case Statement to Replace a Nested If Statement 491

## Tables

1-1 Internal Codes of the Tokens 12

1-2 Terminals for an ALGOL-60 Subset 13

3-1 Types of Interactive High-Level-Language Systems 97

4-1 Terminals 168

4-2 External Edit Commands 169

4-3 Edit Commands 170

4-4 Output Messages 171

此为试读，需要完整PDF请访问：www.ertongbook.com