

<<软件体系结构>>

图书基本信息

书名：<<软件体系结构>>

13位ISBN编号：9783540743422

10位ISBN编号：3540743421

出版时间：2008-04-28

出版商：Springer

作者：Zheng Qin,Jian-Kuan Xing,Xiang Zheng

页数：339

字数：719000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

Building software nowadays is far more difficult than it can be done several decades ago. At that time, software engineers focused on how to manipulate the computer to work and then solve problems correctly. The organization of data and implementation of algorithm were the crucial process of software designing then? However, more and more tasks in low level, such as memory management and network communication, have been automatized or at least can be reused with little effort and cost. Programmers and designers, with the help of high level programming languages and wieldy development tools, can pay more attention to problems, rather than bury themselves into the machine code manuals. However, the side effect of these utilities is that more complicated problems are given according to the requirements from military, enterprise and SO on, in which the complexity grows rapidly day by day. We believe that software architecture is a key to deal with it. Many people become aware of the existence of software architecture just recently. Nevertheless, it in fact has a long history, which may surprise you. Before the invention of C++ or even C, some computer scientists had begun to notice the concept of software structure and its influence to software development. In the 1990s, software architecture started its journey of bloom, when several communities, workshops and conferences were held with a great amount of published articles, books and tools. Today, software architect, the job of taking software designing, analysis and dealing with different concerns and requirements from different stakeholders, is considered as the center of development team. But there is an ironical problem that most existing architects in fact do not take any study or training in this field, some of whom even do not realize that software architecture is a kind of realm requiring academic effort, just as artificial intelligence or data mining. The reason is that software architecture has no widely-accepted definitions and standards of basic theories and practical methods, which leads to that there is almost no universal course about this subject. Meanwhile, the rapid growth and division of software architecture result in too many branches and sub-fields, most of which still keep non-dominant and unified.

<<软件体系结构>>

内容概要

Part of the new series , Advanced Topics in Science and Technology in China. this book aims to introduce the theoretical foundations , various sub-fields , current research , and practical methods of software architecture . Readers can acquire basic knowledge of software architecture , including Why software architecture is necessary , how we can describe system ' S architecture with formal language , what architecture styles are popular in practice , and how we can apply software architecture to the development of systems . Case studies , data , illustrations , and other-materials released within recent years will be used to show the latest status in software architecture.

<<软件体系结构>>

作者简介

Dr . Zheng Qin is doctoral mentor of the computer science and technology departments at Tsinghua and Xi ' an Jiaotong University . He has been Associate Dean of the School of Software , Tsinghua University, and Chair of the Institute of E—commerce , Xi ' an J

<<软件体系结构>>

书籍目录

1 Introduction to Software Architecture 1.1 A Brief History of Software Development 1.1.1 The Evolution of Programming Language Abstract Level 1.1.2 The Evolution of Software Development Concerns 1.1.3 The Origin and Growth of Software Architecture 1.2 Introduction to Software Architecture 1.2.1 Basic Terminologies 1.2.2 Understanding IEEE 1471—2000 1.2.3 Views Used in Software Architecture 1.2.4 Why We Need Software Architecture 1.2.5 Where Is Software Architecture in Software Life Cycle 1.3 Summary References

2 Architectural Styles and Patterns 2.1 Fundamentals of Architectural Styles and Patterns 2.2 Pipes Filters 2.2.1 Style Description 2.2.2 Study Case 2.3 Object-oriented 2.3.1 Style Description 2.3.2 Study Case 2.4 Event-driven 2.4.1 Style Description 2.4.2 Study Case 2.5 Hierarchical Layer 2.5.1 Style Description 2.5.2 Study Case 2.6 Data Sharing 2.6.1 Style Description 2.6.2 Study Case 2.7 Virtual Machine3 Application and Analysis of Architectural Styles

4 Software Architecture Description

5 Design Strategies in Architecture Level

6 Software Architecture IDE

7 Evaluating Software Architecture

8 Flexible Software Architecture

9 A Vision of Software Architecture

Index

章节摘录

components , hence , it supports the function modular level reuse . Existed filters in the system can be easily applied to new systems which are to be designed . Second , the system composed by pipes and filters can be easily maintained and extended . The maintenance is mainly incarnated in a system's evolution . The filter only needs to consider components' input , output and inner implementation , and not needs to consider the filter's maintenance and modification . If we want to replace a certain filter , we only need to design a filter that has the same input , output with the original one . The extension mainly incarnate on the system functions' expansion . For instance , if we want to add a new function to the original system , add new data output , we can finish it by adding new output port to the original filter . Third , in the Pipes Filters style , the independence of filter component provides convenience for system's performance analysis , such as data throughput , deadlock analysis and computing accuracy , etc . Fourth , it supports concurrency computing . Systems based on Pipes-Filters style may have many Parallel filters ; these filters can run concurrently . so that the whole performance of the system is improved . Meanwhile , the Pipes-Filters style has some disadvantage : Filters may have some restrictions to the input and output data , so this style is not proper for interactive systems . In fact , when the pipes filters style is brought forward , the application does not have high interactive requirement . In the early days of computer design , this type of style met the requirement of processing multiple tasks . For some application design that needs sharing much data , it is not proper to use this type of style . The exchanging of data between filters needs large data access space . and the transmission of data will occupy much system running time .

2 . 2 . 2 Study Case In this part , we will give a typical example about digital communication system , and introduce in detail how to organize each component using Pipes-Filters style . From this , we can obviously know that software architecture is production produced when system analysis , creation and management technologies have got many research results . Software architecture does not limit itself to computer software or other concrete subjects , it has strong general utility . The goal of communication is transferring information . Messages have a variety of forms , such as symbols , text , voice , music , graph , image , etc . , according to the difference of messages . We can classify the communication operation into telegraph , telephone , fax , data transferring and visible telephone , etc . In fact , the basic peer to peer communication is always transferring data from one point to another point . ……

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>