

“十一五”国家重点图书

计算机科学与技术学科前沿丛书

计算机科学与技术学科研究生系列教材（中文版）

New Technologies for Software Components

软件构件新技术

王洪泊 编著

清华大学出版社

“十一五”国家重点图书

计算机科学与技术学科前沿丛书

计算机科学与技术学科研究生系列教材（中文版）

New Technologies for Software Components

软件构件新技术

王洪泊 编著

清华大学出版社
北京

内 容 简 介

本书系统地介绍现代软件构件的主流技术,并深入剖析热点技术。全书共分10章,内容包括绪论、软件构件概述、面向Agent的软件构件分析与建模、面向服务计算的构件技术、面向智能工作流的构件技术、基于Agent Zeus的软件构件技术、面向移动Agent的软件构件技术、SPADE Agent平台及其构件技术、JADE Agent平台及其构件技术、工作流jBPM平台及其构件技术。本书以现代构件主流技术为主线,以提升兴趣为先导,以开源实战、团队合作为目标,采用探索式理念,以引导学习者提高软件构件研发能力。

本书既可作为计算机科学与技术专业与软件工程研究方向硕士生的专业核心课教材,也可供软件开发人员自学参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件构件新技术/王洪泊编著. --北京:清华大学出版社,2015

计算机科学与技术学科研究生系列教材(中文版)

ISBN 978-7-302-39790-8

I. ①软… II. ①王… III. ①软件工程—高等学校—教材 IV. ①TP311.5

中国版本图书馆CIP数据核字(2015)第080931号

责任编辑:龙启铭 战晓雷

封面设计:傅瑞学

责任校对:时翠兰

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:16.25

字 数:374千字

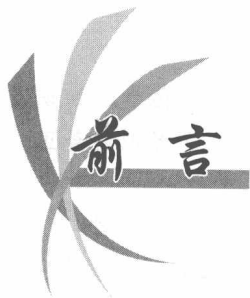
版 次:2015年9月第1版

印 次:2015年9月第1次印刷

印 数:1~2000

定 价:35.00元

产品编号:063452-01



近年来,随着计算机网络技术迅速发展,企事业等用人单位对计算机专业的本科及硕士毕业生提出更新、更高的实践动手能力要求,“软件构件新技术”课程开设以来,受到本科生及硕士研究生的广泛欢迎,选课人数逐年递增。

“软件构件新技术”是计算机科学技术本科及硕士生的重要专业必修课。为了适应计算机网络技术发展趋势,配合我校国家级计算机应用特色专业建设以及相关人才培养工作的迫切需要。注重从软件构件技术的源头出发,既有“热点”上深入浅出的详细讲解与剖析,更有“面上”的系统梳理,使学生们可以用发展的眼光看清楚智能构件的实质及其发展趋势;本书以现代构件的主流技术为主线,以提升兴趣为先导,以开源实战、团队合作为目标,采用探索式理念,鼓励学生们在协作中自觉遵循软件构件习惯与智能构件开发标准,逐步锻炼提升软件构件研发能力。

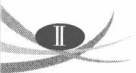
本书共分 10 章,内容包括:绪论、软件构件概述、面向 Agent 的软件构件分析与建模、面向服务计算的构件技术、面向智能工作流的构件技术、基于 Agent Zeus 的软件构件技术、面向 Mobile Agent 的软件构件技术、SPADE Agent 平台及其构件技术、JADE Agent 平台及其构件技术和工作流 jBPM 平台及其构件技术。

本书力求概念准确,论述严谨,内容新颖,图文并茂;围绕基本原理和技术细节的阐述,力求将相关研究的最新进展反映出来。

结合本书的撰写,作者深入开展研究型教学实践尝试,按照夯实理论学习和动手能力培养两方面对学生进行全面素质培养。同时,积极创造机会,为精品课程建设打好基础,为计算机科学及技术特色专业建设贡献力量。

本书已经列入北京科技大学研究生院“十二五”教改建设规划,本书的顺利出版得益于校院系各级领导的关怀和帮助,在此表示衷心感谢。

中国科学院计算技术研究所史忠植研究员和清华大学计算机系史美林教授在参加笔者的博士论文答辩时就建议尽快撰写软件构件新技术的专著。北京邮电大学杜军平教授在本书的审稿过程中提出了许多宝贵建议。



作者在此向他们表示衷心的感谢。

本书是我们关于软件构件新技术教学改革、科研工作的阶段总结,由于该学科知识及相关技术发展迅速和限于作者水平,书中难免有不妥之处,希望相关专家学者批评斧正。

作 者

2015 年 5 月



第 1 章 绪论 /1

1.1	软件构件研发的新挑战	1
1.2	从个性化独立设计到团队高效协作	1
1.2.1	多方支持、关系和谐	2
1.2.2	用户、开发者需真诚合作	2
1.2.3	工作有序、阶段推进	2
1.2.4	网络沟通、流程开放	2
1.2.5	系统渐进、功能渐用	2
1.2.6	按需办事、技术适用	2
1.3	构件系统开发策略	3
1.3.1	最小构件系统	3
1.3.2	原型构件系统	3
1.3.3	进化构件系统	3
1.4	理顺软件构件开发与系统维护的关系	4
1.4.1	采用软件构件工程理论,规范管理开发过程	4
1.4.2	采用软件复用理论,面向复用开发	4
1.4.3	科学设计软件体系结构,提高原型系统可扩展性	5
1.5	软件构件开发流程的标准化之路	5
1.5.1	软件模块的组件、构件化趋势	5
1.5.2	基于网络构件的智能软件体系结构	6
1.5.3	构件模型及其运行机制的研究	6
1.5.4	基于递阶协调联盟框架的反射式动态配置机制	7
1.5.5	用户需求驱动的动态演化意图的构造与实现	7
小结		7

第 2 章 软件构件概述 /8

2.1	构件概念的演化	8
2.1.1	构件概念起源	8
2.1.2	三大构件标准之一: CORBA 技术	9
2.1.3	三大构件标准之二: COM 技术	16

2.1.4	三大构件标准之三: EJB 技术	21
2.1.5	软件构件的特点	25
2.2	构件的定义	25
2.2.1	构件的预先创建性	25
2.2.2	构件的黑盒性	26
2.2.3	构件的可分离性	26
2.2.4	构件的可组装和易部署性	26
2.2.5	构件与复用	26
2.3	构件的理解	26
2.3.1	从对象的封装到构件的包容	27
2.3.2	构件与对象	28
2.3.3	构件的 5 个要素	29
2.3.4	构件规格说明	29
2.3.5	构件接口	29
2.3.6	构件模型	30
2.4	软件构件式开发的利与弊	31
2.4.1	软件复用的优势	31
2.4.2	软件复用的不足	32
小结		32

第 3 章 面向 Agent 的软件构件分析与建模 /33

3.1	Agent 技术概述	33
3.1.1	Agent 的定义	33
3.1.2	Agent 的基本属性	33
3.1.3	Agent 的分类	34
3.1.4	单 Agent 系统和多 Agent 系统	34
3.2	面向 Agent 分析技术	36
3.2.1	知识获取	36
3.2.2	建立需求体系结构模型	36
3.3	面向 Agent 建模技术	37
3.3.1	Agent 的接口设计	37
3.3.2	Agent 与对象化构件	38
小结		38

第 4 章 面向服务计算的构件技术 /39

4.1	SOA 基础	39
4.1.1	SOA 的定义	39
4.1.2	SOA 的特征	40



4.1.3	SOA 的标准	41
4.1.4	SOA 的设计原则	41
4.1.5	SOA 与 Web Services 之关系	41
4.2	SOA 的体系结构	42
4.2.1	SOA 的体系结构	42
4.2.2	SOA 的生命周期	42
4.3	基于 Web Services 实现 SOA	43
4.3.1	Web Services 概述	43
4.3.2	Web Services 模型	45
4.3.3	Web Services 协议栈	45
4.4	SOA 核心实现技术	47
4.4.1	面向 SOA 构建企业服务总线	47
4.4.2	面向 SOA 的模型驱动的开发方法	49
4.4.3	简单对象访问协议	49
4.4.4	Web 服务描述语言	53
4.4.5	统一描述、发现和集成	55
4.5	SOA 的安全标准	59
4.5.1	XML 数字签名	59
4.5.2	XML 加密	60
4.5.3	WS-Security	60
4.5.4	安全声明置标语言	61
4.5.5	可扩展访问控制置标语言	61
4.5.6	XML 密钥管理规范 XKMS	61
4.5.7	联合身份框架 ID-FF	61
4.5.8	WS-Trust	62
4.6	商业流程执行语言	62
4.6.1	商业流程执行语言概述	62
4.6.2	BPEL 语言结构	62
4.6.3	组合 Web Services 的方式	64
	小结	65

第 5 章 面向智能工作流的构件技术 /66

5.1	基本概念	66
5.1.1	协调智能调度工作流	67
5.1.2	参考模型	68
5.2	智能工作流构建	70
5.2.1	智能算子与智能操作	70
5.2.2	智能操作模型的基本组织方式	71



5.2.3	广义算子模型	75
5.2.4	多重广义算子模型	76
5.3	workflow模型的设计	80
5.3.1	workflow模型设计原则	80
5.3.2	workflow模型设计	81
5.3.3	workflow相关数据模型	83
5.4	workflow引擎设计与实现	84
5.4.1	基本功能	84
5.4.2	关键问题	84
5.4.3	总体设计	84
5.4.4	实现	85
5.5	典型应用模型举例	89
5.5.1	系统管理	89
5.5.2	业务管理	93
5.5.3	项目管理	96
5.5.4	任务管理	97
小结		100

第6章 基于 Agent Zeus 的软件构件技术 /101

6.1	Zeus 开发平台	101
6.1.1	Zeus 系统概述	101
6.1.2	Zeus 的几个核心概念	101
6.1.3	Zeus 工具包分析	103
6.1.4	Zeus 开发方法	108
6.2	基于 Zeus 的电子商务系统 ECS 的总体设计	109
6.2.1	选择角色模型	110
6.2.2	Agent 职责分配	111
6.2.3	领域知识模型构建	112
6.3	基于 Zeus 的电子商务系统 ECS 实现	112
6.3.1	实体创建	113
6.3.2	任务 Agent 创建	113
6.3.3	效用 Agent 配置	115
6.3.4	任务 Agent 配置	115
6.3.5	代码的产生和实现	116
6.3.6	电子商务系统 ECS 展示	117
小结		119

第 7 章 面向移动 Agent 的软件构件技术 /120

7.1 移动 Agent 概述	120
7.1.1 移动 Agent 体系结构	120
7.1.2 移动 Agent 理论模型	121
7.1.3 移动 Agent 的历史与现状	121
7.2 Aglet 开放平台	122
7.2.1 Aglet 简介	122
7.2.2 Aglet 应用程序接口 (Aglet API)	123
7.2.3 运行层	124
7.2.4 通信层	125
7.3 Aglet 环境搭建	126
7.3.1 变量设置与 Aglets 的安装	126
7.3.2 Tahiti 的调试	127
7.4 基于移动 Agent 的物流订单系统详细设计	130
7.4.1 Aglet 相关设计	131
7.4.2 界面设计与功能分析	133
7.4.3 详细图示	136
7.4.4 数据存储	138
7.4.5 可行性分析	139
7.5 基于移动 Agent 的物流订单系统运行	139
7.5.1 运行环境搭建	139
7.5.2 成果展示	139
小结	143

第 8 章 SPADE Agent 平台及其构件技术 /144

8.1 SPADE Agent 平台概述	144
8.2 基于 SPADE 的 Web 会议智能调度系统需求	145
8.2.1 需求分析	145
8.2.2 技术路线	145
8.2.3 调度协商模型	146
8.2.4 存储结构	146
8.3 基于 SPADE 的 Web 会议智能调度算法构建	148
8.3.1 空闲度	148
8.3.2 平均空闲度	148
8.3.3 成功率	148
8.3.4 时段权重	148
8.4 基于 SPADE 的 Web 会议智能调度系统设计	149



8.4.1	总体设计方案	149
8.4.2	系统开发平台与技术	149
8.5	基于 SPADE 的 Web 会议智能调度系统实现	150
8.5.1	Django 数据表设计	150
8.5.2	系统数据表设计	150
8.5.3	多 Agent 系统设计	156
8.5.4	MSA	156
8.5.5	CA 类	158
8.5.6	Interact 类	158
8.5.7	消息格式	159
8.5.8	消息传递	161
8.5.9	调度算法	161
8.5.10	调度流程	163
8.5.11	Agent 生命周期控制	165
8.6	模型	166
8.6.1	多 Agent 系统模型	166
8.6.2	Django 模型	166
8.7	控制器	167
8.7.1	admin	167
8.7.2	user	167
8.7.3	agenda	168
8.7.4	meeting	170
8.8	视图	177
8.9	基于 SPADE 的 Web 会议智能调度系统测试	178
8.9.1	用户场景设置	178
8.9.2	日程表	178
8.9.3	会议参数	180
8.9.4	其他条件	180
8.9.5	后台进程	180
8.9.6	测试过程与结果	180
小结		185

第 9 章 JADE Agent 平台及其构件技术 /186

9.1	JADE Agent 平台概述	186
9.1.1	JADE 的包的组成	186
9.1.2	JADE 的特点	187
9.1.3	用 JADE 创建多 Agent 系统	188
9.2	基于 Agent 的任务 workflow 管理系统设计	189



9.2.1	Agent 技术特点	189
9.2.2	workflow 管理系统概述	189
9.2.3	workflow 执行服务和 workflow 引擎	190
9.2.4	workflow 任务管理系统开发策略	191
9.2.5	构建 workflow 管理系统时可能遇到的问题	191
9.3	基于 JADE Agent 的任务 workflow 管理系统实现	193
9.3.1	用 JADE 平台创建多 Agent 系统 DF 服务和 AMS 服务	193
9.3.2	Agent 的执行	194
9.3.3	ACL 通信语言	196
9.3.4	Sniffer 监控通信功能	197
9.4	任务管理系统在 JADE 平台的开发	198
9.4.1	JADE 安装及初步使用	198
9.4.2	系统需求分析	199
9.4.3	系统概要分析	202
9.5	基于 JADE Agent 的任务 workflow 管理系统的实例	203
9.5.1	利用 JADE 平台实现任务管理系统实例背景	203
9.5.2	设定任务申请的 Ontology	203
9.5.3	任务发布及执行过程的实现	203
9.5.4	流程监控的实现	206
小结		207

第 10 章 workflow jBPM 平台及其构件技术 /208

10.1	Jboss jBPM 术语	208
10.1.1	流程定义	208
10.1.2	节点详解	209
10.1.3	服务详解	210
10.1.4	任务管理	212
10.1.5	任务实例	212
10.1.6	任务分配	212
10.2	基于 jBPM 的建设项目监理系统的需求和设计	212
10.2.1	系统目标分析	212
10.2.2	系统设计原则	213
10.2.3	系统环境搭建	213
10.3	业务需求分析	214
10.3.1	注册模块	214
10.3.2	登录验证模块	214
10.3.3	用户管理模块	214
10.3.4	任务管理模块	215



- 10.3.5 流程管理模块..... 215
- 10.3.6 系统数据库分析..... 216
- 10.4 基于 jBPM 的建设项目监理系统实现..... 229
 - 10.4.1 费用申报流程定义..... 229
 - 10.4.2 工程质量问题通知单流程定义..... 230
 - 10.4.3 流程部署..... 231
 - 10.4.4 特殊流程模式处理..... 231
 - 10.4.5 基于角色的访问控制..... 232
 - 10.4.6 流程监控的实现..... 232
 - 10.4.7 系统测试结果展示..... 233
- 小结..... 241

软件构件新技术已经成为信息社会必要的基础设施,也是时代特征最重要的体现之一。网络与信息技术的迅速发展与广泛普及,为大型应用软件构件系统应对现代社会(特别是企业)生产过程自动化、工艺流程智能化的趋势提供了新环境、新条件,也带来了新机遇、新问题。

1.1 软件构件研发的新挑战

随着飞速发展的计算机、智能手机、物联网终端等应用系统灵活性的不断增强,相关软件系统的功能需求变得越来越复杂,相关性能要求也愈来愈高。目前缺乏快速开发各种满足质量要求、安全、可靠的软件的实用研发技术,软件的生产能力远远满足不了飞速发展的实际需求。在互联网、物联网时代,软件规模的增长给软件开发带来了许多的具体困难,例如:

- 技术和需求的变化问题;
- 软件难以维护问题;
- 重复不必要的工作问题;
- 软件质量和成本有效性问题;
- 应用集成的问题。

围绕上述困难,核心问题是:复杂网络背景下,异型、异构、异步的各大型应用软件子系统如何相互协调、协同运行。

多年来,国内外学者们从不同侧面对大型应用软件系统及其开发、维护技术进行了深入研究,工业界侧重于实用,关注软件系统的服务质量(效率、安全性和可靠性等);而学术界则侧重于增强软件系统的表达能力(如语义方面)和对新的应用环境的支持。针对现存的主要问题,有不同的研究工作和技术成果,但从大型应用软件系统的核心技术来看,一些关键问题仍有待解决,特别是网络环境下新需求的产生,还有更为广阔的研究空间。

1.2 从个性化独立设计到团队高效协作

大型软件系统的开发策略是一个系统工程问题,应根据具体的实际系统进行全面分析,制定切实可行的开发策略。开发过程中应该遵循以下原则。

1.2.1 多方支持、关系和谐

在大型软件系统的开发过程中,协调信息部门的主要领导对系统的重视和支持对于系统成功与否有直接的影响,并能够起到决定性作用。特别是在系统分析、规划、决策、指挥、协调、评审过程中,提出软件系统目标,明确系统的任务,更需要领导的参与。而在系统的组织和工程实施中,必须获得领导的重视和支持。只有这样,才能有效地进行人力、财力和物力调度,努力创造有利条件,克服可能的各种阻力,协调各方面的关系,以保证高效地进行大型软件系统的开发。

1.2.2 用户、开发者需真诚合作

大型软件系统的开发可采用独立开发、委托开发、合作开发和引进软件等各种方式,都涉及开发者和用户的密切合作问题。因为协调智能工作流的开发是一个共同研究、相互协作的过程,所以,用户和开发者之间需建立真诚合作的伙伴关系。开发者应力求满足用户的需求,用户应尽量支持开发者的工作。

1.2.3 工作有序、阶段推进

大型软件系统的开发过程本身是一项系统工程,也需要进行科学的组织协调调度。然而,这是包括用户和开发者双方或多方的计划协调工作。需要在合同制约下,按照系统工程和软件工程的原则和规律,全面规划,打好基础,根据进度计划循序渐进。

1.2.4 网络沟通、流程开放

大型软件系统的设计、开发、运行过程是伴随着用户协调调度体制、组织机构的改革、开放的过程。大型软件系统既要适应和满足用户现行管理体制的工作需要,又要跟踪和促进用户管理体制的改革和更新,才能提高科学信息化水平,实现信息流程自动化。

1.2.5 系统渐进、功能渐用

用户的各级人员、运行值班、系统维护人员的培训工作是大型软件系统开发和应用能否成功的关键问题。应当结合需求分析、系统设计、安装调试及运行评审等开发过程,做好用户人员培训工作。

1.2.6 按需办事、技术适用

大型软件系统的软、硬件设备,可以采用一次性购置和分阶段购置的方法。一次性购置是根据大型软件系统的总体设计方案,统一购置所有的设备。分阶段购置是根据大型软件系统的进度计划,合理分解,逐步开发。这里,应当按需办事,量力而行,按照实际工作需要,考虑技术发展趋势,根据性能价格比,选取适用的技术。

1.3 构件系统开发策略

为了延伸大型软件构件系统的生命周期,适应和跟踪其服务对象的发展与变化,根据大型软件系统的开发原则,建议大型软件构件系统的设计与开发过程采用进化系统的开发策略,如图 1.1 所示。

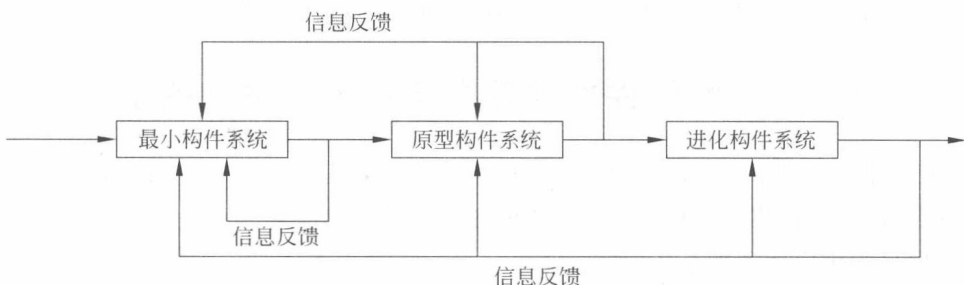


图 1.1 构件式进化系统开发策略

这是具有多重反馈、逐步进化的开发策略,分为 3 步: 最小构件系统→原型构件系统→进化构件系统。利用反馈信息,逐步进化。根据用户需求,改善人-机协调特性,充实大型软件构件系统的功能,扩展、提高构件系统的智能水平。

1.3.1 最小构件系统

最小构件系统(minimum component system)主要是指规模最小的工程试验系统,其任务是验证系统设计方法和实现技术的可行性、适用性。一般至少使用 2~3 个应用实例来进行验证,以评审系统的可行性和适用性。

1.3.2 原型构件系统

原型构件系统(prototype component system)主要是指可供实用的、第一个版本的或第一个型号的协调智能工作流。原型系统开发成功后,应通过合同约定的、实际运行的考验,随后进行技术鉴定与效益评审。

1.3.3 进化构件系统

进化构件系统(evolutionary component system)是在原型系统基础上再版或新型的系統,可以在同行业、同类服务对象中推广的、定型的商品化系统。进化构件系统开发策略是根据我国国情以及大型软件系统开发方面的实践经验,在原型构件系统、生命周期开发策略的基础上提出的大型软件系统的开发策略。通常,用户和开发者之间可能存在对同一问题的不同理解,由于缺乏有关知识,用户有时也不清楚系统的详细需求,难以把用户意图向开发者完全表述明白。也就是说,大型软件系统的所有需求并非都能够预先确定。往往需要在系统的开发过程中,通过最小系统的工程试验和原型系统的实际运行,相互协商,逐步明确。

进化系统开发策略的特点如下：

- 加强开发过程中用户参与和合作的意识；
- 在生命周期中的早期提供系统测试，验证用户需求的环境；
- 便于用户与开发者的相互协商、密切合作；
- 有助于合同的顺利执行和工程开发过程的有效组织；
- 便于培训用户的值班、维护人员，提高系统运行可靠性；
- 有利于延伸系统的生命周期，适应环境变化，开发新型系统。

1.4 理顺软件构件开发与系统维护的关系

系统必须经过科学合理的设计，充分考虑系统将来的可扩展性，尽量做到系统高内聚、低耦合，能够满足将来根据实际需求不断添加新功能的要求。为了实现系统的高度可扩展性，应该采用以下多种方法。

1.4.1 采用软件构件工程理论，规范管理开发过程

软件工程是一套关于软件开发各个阶段的定义、任务、作用和过程控制的一门工程学科。其目的是解决软件危机的诸多问题，指导软件开发者利用科学、有效的方法来开发软件，提高及保证软件开发的效率和软件产品的质量。软件工程包括方法、工具、过程 3 个要素，其中方法、工具属于技术范畴，过程属于管理范畴。按照软件工程理论，软件原型系统开发需经历完整的软件生命周期 (software life cycle)，包括可行性分析、需求分析、概要设计、详细设计、编程实现、测试部署、运行维护等几个阶段。通过采用软件工程方法可避免软件开发中可能遇到的诸多问题。

1.4.2 采用软件复用理论，面向复用开发

麦肯锡在研究了世界上最成功的 100 家软件企业后得出结论：一种新的采用了标准化的复用构件技术最终会取代昂贵的定制编程和集成，这种方法是未来软件生产力提高的主要来源，也将成为软件行业未来前景中的核心部分。软件复用能够避免软件开发过程中的重复劳动，提高软件开发效率和产品质量。在开发原型系统的过程中，可采用不同层次的软件复用。系统整体框架在底层代码实现过程中采用面向对象的方法，充分利用面向对象技术封装、继承、多态的特征以及面向接口编程等方法进行代码级的复用。通过软件复用，使软件更符合高内聚、低耦合的要求，这些可复用组件能够满足以后添加软件系统新功能的需求，从而提高系统的可扩展性。

软件复用的原则如下：

- 在一个整体的概念和框架下，而不是在随意的方式下定义一个构件，以便增大复用的机会；
- 构件设计的上下文无关：构件在新的环境中被使用时，无须将使用它的构件一起迁移过来；
- 变化影响局部化：变化影响代码局限在一个小的范围内；