

第 1 章概述	54
1. 1 信息系统与信息系统工程	54
1.2 建立信息系统所涉及的问题	55
1.2.1 系统建设前企业应具备的条件	55
1.企业高层领导应具有对企业信息系统建设规律性的认识.....	55
2.企业必须认真分析建立计算机信息系统的实际需求.....	55
3.管理的科学化是企业信息系统建立的基础和保证.....	55
4.企业文化和管理人员的组织结构应能满足系统建设的需要.....	55
5.规范和齐全的数据是建立企业计算机信息系统的必要条件.....	55
6.企业有必要的开发期和维护期的资金保证.....	56
1.2.2 系统建设中企业应具备的条件	56
1.企业高层领导介入系统建设	56
企业领导介入系统建设的必要性在于.....	56
2.吸收相关企业信息系统建设中的经验和教训.....	56
3.选择适合本企业实际情况的开发方式.....	56
企业信息系统的开发可以有多种方式的选择。传统的方式有.....	56
• 委托开发方式	56
• 合作开发方式	56
• 自行开发方式	56
4.建立系统开发组织机构和选择成员.....	57
5. 系统开发策略的制定和开发方法、开发工具的选择.....	57
6.组织基础数据的收集和预处理, 实施数据工程.....	57
实施数据工程会涉及 3 方面的工作.....	57
• 确定收集数据的范围和数量并提出质量要求.....	57
• 有规范的标准的数据格式.....	57
• 完善对主题数据库的设计.....	57
7. 设计并确定系统目标, 进行投资估算.....	57
8.合理设计信息部门在企业机构中的位置.....	58
9.应用自动化的手段来开发系统	58
1.认真做好系统的验收工作	59
2.着力优化系统的功能和性能.....	59
3.重视文档的整理和接收	59
4.重视系统维护队伍的建设	59
1.3 信息系统工程所涉及的技术内容	59
1.3.1 管理科学的应用	59
1. 3. 2 方法论的发展与应用	60
1. 基于经验的开发	60
2.软件危机与软件工程	60
3.自底向上和自顶向下	61
自底向上方法的优点有	61
自底向上方法的缺点有	61
自顶向下方法的优点有	61
自顶向下方法的缺点有	61
4.模型化	61
(1)瀑布模型.....	62
(2)螺旋模型.....	62
它在笛卡儿坐标的 4 个象限上反映出 4 方面的活动.....	62
制定计划	62
风险分析	62
工程实施	62

工程评估	62
(3)喷泉模型.....	62
1 . 3.3 从软件工程到信息工程	62
1.4 系统分析员及其培养	63
1.5 系统分析员教程的内容组织	65
第 2 章信息与系统	65
2. 1 信息与信息化	65
2. 1. 1 信息时代与国家信息化	65
关于信息社会的特征有很多说法，主要有以下几点.....	66
国家信息化体系包括 6 个因素	66
·信息资源.....	66
·信息网络.....	66
·信息技术应用	66
·信息产业.....	66
·信息化人才.....	66
·信息化政策、法规、标准和规范.....	66
2.1.2 信息与数据	66
2.1.2.1 信息与数据的定义	66
2.1.2.2 信息的属性	67
· 真伪性	67
· 层次性	67
· 不完全性	67
· 滞后性.....	67
· 扩压性	67
· 分享性	67
2. 1. 2. 3 信息量和信息熵	67
2.1.3 信息与管理	67
1.管理信息及其特征	67
根据以信息为依据的管理唯物论的基本原理，可以将管理信息的定义分解为以下几点.....	67
.....	67
2.信息与信息收集	68
2. 1. 4 信息与决策	68
2.2 系统与系统工程	68
2.2.1 系统的概念	68
1.一般系统论概述	68
2.系统的分类	68
·按系统抽象程度分	69
·按系统的功能分	69
3.系统的特性	69
· 整体性	69
· 层次性	69
· 相关性.....	69
· 目的性.....	69
· 环境适应性	69
2.2.2 系统与环境	69
1.系统的能控与能观	69
2.系统的接口与耦合	69
3.系统的自组织性	70
2. 2. 3 系统工程与系统方法	70
三维结构由时间维、逻辑维和知识维组成一个立体结构.....	70

(5)安装调试阶段.....	70
(6)运行阶段.....	70
(7)更新阶段.....	70
逻辑维	70
(1)问题确定.....	70
(2)目标确定.....	70
(3)系统综合.....	70
(4)系统分析.....	70
(5)最优化.....	70
(6)系统决策.....	70
(7)计划实施.....	70
2.3 信息系统工程	70
1.信息系统的特征	71
2. 信息工程的发展	71
3. 信息工程的复杂性及解决方案.....	71
SEI的 5 级管理能力模式如下	71
第 3 章结构化分析与设计方法	71
3. 1 方法概述	71
3.1.1 系统开发生命周期	71
3.1. 2 结构化方法的基本思想	72
3.1.2.1 结构化分析	72
1.结构化系统分析思想	72
2.结构化分析方法的内容	72
3.结构化分析方法的特点	72
4.结构化分析方法的局限	72
3. 1.2. 2 结构化设计	73
结构化设计方法内容主要包括.....	73
• 系统总体结构	73
• 系统设备配置	73
• 系统分类编码方案	73
• 数据库结构图	73
• I/O设计方案	73
• HIPO图	73
• 处理逻辑和存储方案	73
3.1.3 系统开发的阶段划分	73
1.总体规划阶段	73
总体规划的作用可以分成以下几点.....	73
• 指明组织中建立信息系统的范围和目标.....	73
• 指导信息系统开发	73
• 合理分配和利用各种资源.....	73
• 通过规划过程找出企业中存在的问题.....	73
2.系统分析阶段、	73
3.系统设计阶段	73
4. 系统实施阶段	73
5.系统运行和评价阶段	74
3.1.4 系统开发中的管理	74
3.1.4.1 项目管理	74
1.任务划分	74
2.计划安排	74
3.经费管理	74

4.审计控制	74
5. 风险管理	74
归纳起来, 风险主要有以下几方面.....	75
6.质量保证	75
3. 1. 4. 2 人员组成与管理	75
1.人员的构成	75
2.组织形式	75
可以采取的组织形式有下面 3 种.....	75
• 共同工作小组	75
• 主管负责制	75
• 主管负责下的专业分工制	75
3.对人员的选择	75
3.1.4.3 系统开发中全面质量管理	76
为了在信息系统的建设过程中实施全面的质量控制, 主要采取下述措施.....	76
• 实行工程化的开发方法.....	76
• 实行阶段性冻结与改动控制.....	76
• 进行原型演化	76
3.2 总体规划	76
3. 2. 1. 总体规划概述	76
这个阶段的主要任务是	76
• 制定信息系统的发展战略	76
• 确定组织的主要信息需求, 形成信息系统的总体结构方案, 安排项目开发计划.....	76
• 制定系统建设的资源分配计划.....	76
3.2.1.1 总体规划主要步骤	76
• 一般调查	76
• 信息需求初步调查	76
3. 2. 1.2 总体规划方法	76
1.关键成功因素法(CSF).....	77
2.战略目标集转化法(SST).....	77
具体步骤如下	77
3. 2. 2 目标系统框架分析	77
3.2.2.1 管理目标分析	77
进行管理目标分析的步骤是	77
3.2. 2.2 系统目标分析	77
通常, 信息系统应该在下面几个方面发挥作用.....	77
3.2. 2. 3 系统范围及功能	78
确定系统范围和功能的原則如下.....	78
按照上述原則, 确定系统的范围和功能应采取的步骤是.....	78
3.2.2.4 系统总体结构及投资概算	78
1.系统总体结构	78
2.投资概算	78
• 计算机系统软、硬件设备投资.....	78
• 系统开发费	78
• 系统安装和维护费用	78
• 人员培训费	78
3. 2. 3 可行性分析及总体规划报告	79
3.2.3.1 可行性分析的内容	79
一般来说建立信息系统的必要性大概有 3 种情况.....	79
• “显见” 的必要性	79
• “预见” 的必要性	79

• “隐见” 的必要性	79
建立信息系统的可能性主要有以下内容.....	79
• 经济可行性	79
• 技术可行性	79
• 管理上的可行性	79
• 开发环境的可行性	79
3.2.3.2 可行性分析报告	79
1.引言	79
2.现行系统调查与分析	79
3.新系统建设方案	79
4. 其他	80
3. 3 系统分析与建立逻辑模型	80
3. 3. 1 系统分析概述	80
3.3.1.1 系统分析的任务和目的	80
3. 3. 1. 2 系统分析的主要步骤	80
系统分析过程一般按如图 3. 8 所示的逻辑进行.....	80
(4)对目标系统的逻辑模型具体化(物理化), 建立目标系统的物理模型.....	80
按照图 3.8 所示, 可将系统分析阶段的主要工作步骤分为.....	80
3.3.2 详细调查	80
3. 3. 2. 1 详细调查的主要内容	80
1.静态信息调查: 组织结构的调查.....	80
2.静态信息调查: 功能体系的调查.....	81
3.动态信息调查: 业务流程的调查.....	81
4.动态信息调查: 数据流程调查.....	81
3.3.2.2 详细调查的原则	81
1.自顶向下全面展开	81
2.存在的 不一定是合理的	81
3.分工和协作相结合	81
4.点面相结合展开调查	81
5.主动沟通的工作方式	81
3.3.2.3 详细调查的方法	81
• 收集资料	81
• 开调查会	81
• 个别访问	81
• 书面调查.....	81
• 参加业务实践	82
• 发电子邮件	82
在系统调查时, 应注意下面的一些问题.....	82
• 事先计划	82
• 调查态度	82
• 调查顺序	82
• 研究分析	82
3.3.3 需求分析	82
1.系统范围与目标分析	82
• 确定系统范围	82
• 确定系统需求	82
2.系统组织结构与功能分析	82
• 了解组织结构及各部分的功能.....	82
• 了解相关部门职能上的各种联系.....	82
• 分析组织结构的合理性	82

• 分析组织结构设置的必要性和合理性	82
• 发现其中的问题	82
• 提出改进的意见	82
在系统组织结构与功能分析中, 有以下几个主要的工具可以应用	82
• 组织结构图	82
• 组织/业务关系图	82
• 业务功能一览表	82
3. 系统性能分析	82
3. 3. 4 业务流程详细调查与分析	82
业务流程分析的步骤可以总结如下	83
1. 组织结构与业务流程详细调查	83
2. 业务流程图和系统概况图	83
业务流程图的基本符号	83
3. 业务流程优化与再造	83
企业流程再造(BPR) 应遵循以下原则	83
• 有一个明确的、具有启发性的目标, 即共同远景	83
• 充分考虑顾客的价值	83
• 必须服从统一指挥	83
• 充分做好横向及纵向沟通	84
• 认识流程再造的两大要素—信息技术/信息系统和人员组织管理	84
• 树立典范、逐步推进, 充分利用变革的涟漪效应	84
3.3.5 数据流程分析	84
3. 3. 5.1 数据流及数据流图	84
• 物资流	84
• 货币流	84
• 人员流	84
• 机器及设备流	84
• 数据流	84
采用数据流图的方式进行数据流程分析一般应遵循以下原则	85
• 明确系统边界	85
• 在总体上遵循自顶向下逐层分解的原则	85
• 在局部上遵循由外向里的原则	85
3.3.5.2 数据流图的绘制与检验	85
1. 识别系统的输入和输出	85
2. 绘制系统内部数据流	85
3. 对复杂加工进行分解	85
4. 对草图进行检查和合理布局	85
5. 和用户交流	85
6. 检查、修改、完善	85
分层数据流图便于人们理解和使用, 但在绘制时应注意以下事项	85
① 自顶向下、逐层分解	86
② 数据流必须经过加工环节, 即必须进入加工环节或从加工环节流出	86
③ 数据存储环节一般作为两个加工环节的界面来安排	86
④ 编号	86
⑤ 只绘制所描述的系统稳定工作情况下的数据流图	86
数据流图的正确性可从以下几方面检查	86
• 数据守恒	86
• 文件使用	86
• 子图和父图平衡	86
• 加工和数据流的命名	86

如果数据流图的可读性	86
• 简化加工之间的联系	86
• 分解应当均匀	86
• 命名应当恰当	86
但数据流图在描述系统逻辑功能和有关信息内容的细节方面仍存在较大的局限性	87
3.3.5.3 数据流图绘图举例	87
数据流图由 4 种基本符号组成.....	87
对数据流的表示通常有以下约定.....	87
②数据处理	87
加工的作用主要是	87
③数据存储	87
④外部实体	87
3.3.6 数据字典	87
编写数据字典的基本要求是	88
3.3.6.1 数据字典项目描述内容举例.....	88
数据字典中有 6 类条目	88
1. 数据元素	88
• 名称	88
• 别名	88
• 类型	88
• 取值范围和取值的含义	88
• 长度	88
2. 数据结构	88
• 任选项	88
• 必选项	88
• 重复项	88
3. 数据流	88
• 数据流的来源	88
• 数据流的去处	88
• 数据流的组成	88
• 数据流的流通量	88
• 高峰时的流通量	88
4. 数据存储	88
5. 外部实体	88
6. 处理	88
3.3.6.2 数据量统计及分析	89
3.3.7 基本加工处理描述	89
1. 基本加工处理概述	89
编写加工说明的时候，有如下要求.....	89
2. 结构化语言	89
3. 决策树	89
4. 决策表	89
3.3.8 建立新系统逻辑模型	89
对新系统的信息处理方案的确定包括如下几部分	90
1. 新系统组织机构及业务流程.....	90
首先要做的，就是确定新系统组织机构和业务流程。主要工作有	90
2. 新系统目标及范围	90
3. 新系统逻辑结构及数据分布.....	90
4. 新系统数据流图及数据字典.....	90
应该确定合理的数据和数据流程。主要工作有	90

5.新系统数据分析及数据量统计	90
6.新系统实施策略及计划	90
7.新系统投资预算及策略	91
3. 3.9 系统分析报告	91
系统分析报告主要有以下 3 个作用.....	92
一份完整的系统分析报告应该包括下述内容.....	92
①组织情况概述	92
②现行系统概述	92
③系统逻辑模型	92
④新系统在各个业务处理环节拟采用的管理方法、算法或模型.....	92
⑤与新的系统相配套的管理制度和运行体制的建立.....	92
⑥系统设计与实施的初步计划.....	92
⑦用户领导审批意见	92
3.4 系统设计	92
3.4.1 系统设计概述	92
项目开发过程并不总是能按总体计划分阶段顺利推进，甚至造成反复，究其原因有	92
3.4.1.1 系统设计的内容和步骤	93
系统设计的基本任务大体上可以分为两个步骤.....	93
3.4.1.2 系统结构设计的原则	93
①分解—协调原则	93
在系统中，应按以下要求分解.....	93
协调的依据主要是	93
②自顶向下的原则	93
③信息隐蔽、抽象的原则	93
④一致性原则	93
⑤明确性原则	93
⑥模块之间的耦合尽可能小，模块内部组合要尽可能紧凑.....	93
⑦模块的扇入系数和扇出系数要合理.....	93
⑧模块的规模适当	94
3.4.2 系统总体结构设计	94
3.4.2.1 子系统划分	94
1.子系统划分的原则	94
• 子系统要具有相对独立性.....	94
• 子系统之间数据的依赖性尽量小.....	94
• 子系统划分的结果应使数据冗余较小.....	94
• 子系统的设置应考虑今后管理发展的需要.....	94
• 子系统的划分应便于系统分阶段实现.....	94
2.系统划分方法的分类	94
3.4.2.2 子系统结构设计	94
3.4.2.3 网络设计	94
3. 4. 2. 4 硬件设备及配置	95
3. 4. 3 系统模块结构设计	95
3.4. 3. 1 模块的概念	95
一个模块应具备以下 4 个要素.....	95
• 处理功能	95
• 内部数据	95
• 程序代码	95
3. 4. 3.2 模块结构图	95
为了确保系统设计工作的顺利进行，结构设计应遵循如下原则.....	95
模块结构图	95

• 模块	95
• 调用	95
• 控制信息	96
• 转接符号	96
3. 4.3.3 模块的变换型分析与事务型分析.....	96
变换型模块结构	96
事务型系统	96
1.变换型分析	96
(3)设计顶层模块和第一层模块。	97
2.事务型分析	97
3.4.3.4 模块的藕合与内聚	97
模块的藕合方式有 3 种	97
• 数据藕合	97
• 控制藕合	97
• 非法藕合	97
模块的内聚方式	97
• 巧合内聚	97
• 时间内聚	98
• 过程内聚	98
• 通信内聚	98
• 功能内聚	98
3.4.4 系统详细设计	98
3.4.4.1 代码设计	98
代码设计应该遵循以下基本原则.....	98
• 惟一性	98
• 合理性	98
• 可扩充性	98
• 简单性	98
• 适用性	98
• 规范性	98
• 系统性	98
代码设计可以按照以下步骤进行.....	98
目前常用的编码归纳起来有如下几种形式.....	98
• 顺序码	98
• 数字码	98
• 字符码	99
• 混合码	99
在实际分类时必须遵循如下几点.....	99
3.4.4.2 输出设计	99
输出设计包括以下几方面的内容.....	99
• 确定输出内容	99
• 选择输出设备与介质	99
• 确定输出格式	99
最终输出方式常用的只有两种：一种是报表输出，另一种是图形输出.....	99
3. 4. 4. 3 输入设计	99
为此,输入设计应遵循以下原则：	99
• 最小量原则	99
• 简单性原则	99
• 早检验原则	100
• 少转换原则	100

输入设计的内容包括	100
• 确定输入数据内容	100
• 输入方式设计	100
• 输入格式设计	100
• 校对方式设计	100
3.4.4.4 处理过程设计	100
1.程序流程图	100
2.盒图(NS图).....	100
3. 形式语言	100
4.决策树	100
5.决策表	101
3.4.4.5 数据存储设计	101
• 数据资源分布	101
• 数据的安全保密	101
3.4.4.6 用户界面设计	101
1.菜单方式	101
2.会话管理方式	101
纠错、容错的目的是保证会话的正确性，提高会话的效率，在系统中可采用如下方法	101
• 提示法	101
• 确认回答法	101
• 无效处理法	101
• 返回处理法	101
• 延时处理法	101
• 帮助处理法	101
3. 提示方式与权限管理	101
3. 4. 4. 7 安全控制设计	102
从数据环境和数据处理两方面看，影响系统安全的因素有	102
• 环境性因素	102
• 数据处理因素	102
3. 4. 5 系统设计报告	102
3. 5 系统实施	104
3. 5. 1 系统实施概述	104
1.系统实施的目的和任务	104
系统实施阶段的主要任务是	104
• 按总体设计方案购置和安装计算机网络系统.....	104
• 软件准备	104
• 人力培训	104
• 数据准备	104
• 投入切换和试运行	104
2.系统实施的步骤	104
3.5.2 程序设计	105
程序设计的主要依据是系统设计阶段的HIPO图以及数据库结构和编码设计.....	105
1.程序设计的方法	105
• 结构化程序设计方法	105
• 快速原型式的程序开发方法	105
2.程序设计基本模块	105
①控制模块	105
②输入模块	105
③输入数据校验模块	105

④输出模块	105
⑤处理模块	105
• 文件更新模块	105
• 分类合并模块	105
• 计算模块	105
• 数据检索模块	105
• 预测或优化模块	105
3.5.3 系统测试与调试	106
3.5.3.1 系统测试的意义及目的	106
根据测试的概念和目的，在进行信息系统测试时应遵循以下基本原则	106
• 应尽早并不断地进行测试	106
• 测试工作应该避免由原开发软件的人或小组承担	106
• 设计测试方案的时候，不仅要确定输入数据，而且要根据系统功能确定预期输出结果	106
• 在设计测试实例时，不仅要设计有效合理的输入条件，也要包含不合理、失效的输入条件	106
• 在测试程序时，不仅要检验程序是否做了该做的事，还要检测程序是否做了不该做的事	106
• 严格按照测试计划来进行，避免测试的随意性	106
• 妥善保存测试计划、测试例子，作为软件文档的组成部分，为维护提供方便	106
3.5.3.2 测试过程	106
一个规范化的测试过程通常包括以下基本的测试活动	106
(1)拟定测试计划	106
(2)编制测试大纲	106
(3)根据测试大纲设计和生成测试例子	106
(4)实施测试	106
(5)生成测试报告	106
3.5.3.3 测试策略与测试方法	106
1.人工测试	106
• 个人复查	107
• 抽查	107
• 会审	107
2.机器测试	107
①黑盒测试也称为功能测试	107
②白盒测试也称为结构测试	107
3.软件测试步骤	107
(1)单元测试	107
①模块接口	107
②局部数据结构	107
③重要的执行路径	108
④出错处理	108
⑤边界条件	108
在对每个模块进行测试时，需要开发两种模块	108
• 驱动模块	108
• 桩模块	108
(2)组装测试	108
(3)确认测试	108
• 有效性测试	108
• 软件配置审查	108
• 验收测试	108

(4)系统测试.....	108
• 恢复测试	109
• 安全性测试	109
• 强度测试	109
• 性能测试	109
• 可靠性测试	109
• 安装测试	109
3.5.3. 4 调试	109
目前常用的调试方法有如下几种.....	109
• 试探法	109
• 回溯法	109
• 对分查找法	109
• 归纳法	109
• 演绎法	109
3. 5. 4 系统文档	109
• 用户与系统分析人员在系统规划和系统分析阶段通过文档进行沟通.....	110
• 系统开发人员与项目管理人员通过文档在项目期内进行沟通.....	110
• 前期开发人员与后期开发人员通过书面文档进行沟通.....	110
• 系统测试人员与系统开发人员通过文档进行沟通.....	110
• 系统开发人员与用户在系统运行期间进行沟通.....	110
• 系统开发人员与系统维护人员通过文档进行沟通.....	110
• 用户与维护人员在运行维护期间进行沟通.....	110
3.5.5 系统转换	110
系统试运行阶段的工作主要有	110
•	110
•	110
新旧系统之间的转换方式有直接转换、并行转换和分段转换.....	110
•	110
•	110
•	110
3.6 系统维护与评价	111
3. 6.1 系统维护概述	111
3.6.1.1 系统可维护性概念	111
1.系统的可维护性的评价指标.....	111
• 可理解性	111
• 可测试性	111
• 可修改性	111
2.维护与软件文档	111
3.软件文档的修改	111
3.6.1.2 系统维护的内容及类型	111
1.硬件维护	111
2.软件维护	111
软件维护的内容一般有以下几个方面.....	112
• 正确性维护	112
• 适应性维护	112
• 完善性维护	112
• 预防性维护	112
3.数据维护	112
3. 6. 1. 3 系统维护的管理和步骤.....	112
(1)提出维护或修改要求	112

(2)领导审查并做出答复,如同意修改则列入维护计划	112
(3)领导分配任务,维护人员执行修改	112
(4)验收维护成果并登记修改信息	112
3.6.2 系统评价	113
3.6.2.1 系统评价的目的和任务	113
• 立项评价	113
• 中期评价	113
• 结项评价	113
3.6.2.2 系统评价的指标	113
一、系统质量	113
二、技术水平	114
三、运行质量	114
四、用户需求	114
五、系统成本	114
六、系统效益	114
七、财务评价	114
3.6.3 系统运行管理	114
3.6.3.1 运行管理制度	114
1.各类机房安全运行管理制度	114
2.信息系统的其他管理制度	115
3.6.3.2 日常运行管理内容	115
1.系统运行情况的记录	115
2.、审计踪迹	115
• 语句审计	115
• 特权审计	115
• 对象审计	115
3.审查应急措施的落实	115
4.系统资源的管理	116
3.6.3.3 系统软件及文档管理	116
(1)系统软件的管理除日常维护以外,还包括版本更新和升级等	116
(2)信息系统文档的管理	116
• 文档管理的制度化	116
• 文档要标准化、规范化	116
• 文档管理的人员保证	116
• 维护文档的一致性	116
• 维持文档的可追踪性	116
第4章企业系统规划方法	116
4.1 概述	116
4.1.1 BSP的概念	116
1.一个信息系统必须支持企业的战略目标	116
2.一个信息系统的战略应当表达出企业的各个管理层次的需求	117
• 战略计划层	117
• 管理控制层	117
• 操作控制层	117
3.一个信息系统应该向整个企业提供一致的信息	117
4.一个信息系统应该适应组织机构和管理体制的改变	117
5.一个信息系统的战略规划,应当由总体信息系统结构中的子系统开始实现	117
4.1.2 BSP的目标	117
4.2 BSP方法的研究步骤	117
4.2.1 研究项目的确立	117

4. 2. 2 研究准备工作	118
4.2.3 研究的主要活动	118
1.研究开始阶段	118
2.定义企业过程	118
3.定义数据类	118
4.分析现存系统支持	118
5.确定管理部门对系统的要求	118
6.提出判断和结论	118
7.定义信息总体结构	118
8.确定总体结构中的优先顺序	118
9. 评价信息资源管理工作	118
10.制定建议书和开发计划	119
11. 研究成果报告	119
4.3 定义企业过程	119
4.3.1 过程定义的目的和条件	119
定义企业过程的目的和作用可归纳为	119
过程定义以前, 下列几点是研究的成功必要条件	119
4.3.2 产品和资源的生命周期	119
生命周期的各个阶段可描述如下	119
4. 3. 3 定义过程的基本步骤	119
4. 3. 3. 1 计划和控制过程	119
4.3.3.2 产品/服务过程	119
1.识别企业的产品/服务	120
2.按产品/服务的生命周期的各个阶段识别过程	120
3.画出产品/服务过程总流程图	120
4. 写出每一过程的说明	120
(1)生产计划	120
(2)采购	120
4.3.3.3 支持资源过程	120
1.支持资源的描述	120
4. 3. 3.4 过程的归并和分析	120
1.过程的归并	120
2.画出过程组合表和完成过程说明	121
3.建立企业过程与组织的联系	121
4.识别企业成功的关键过程	121
4.3.3.5 结果和应用	121
一般从定义企业过程中, 应获得以下结果和资料	121
4. 4 定义数据类	121
4. 4. 1 识别数据类	121
识别数据类是为了解决下列问题	121
4. 4. 2 给出数据类定义	122
4. 4. 3 建立数据类与过程的关系	122
4. 5 分析当前业务与系统的关系	122
4.5.1 分析现行系统支持	122
1.考察信息系统对过程的支持	122
2.识别当前的数据使用情况	122
4.5.2 确定管理部门对系统的要求	123
面谈的目的有以下几方面	123
(1)面谈的一般准备	123
(2)针对面谈对象的特别准备	123

(3) 进行面谈	123
(4) 总结和分析每次谈话结果.....	123
BSP还要求及时总结面谈结果，其内容包括下面几方面.....	123
4.5.3 提出判断和结论	124
总结目的有如下几方面	124
• 与管理人员进行交流	124
• 为提出实施计划提供依据	124
• 为建立总体结构优先次序提供依据.....	124
• 为信息结构中的子系统描述提供基础材料.....	124
下面介绍提出判断和结论的步骤	124
(1) 检查前期工作完成情况	124
(2) 确定判断和结论的范畴	124
① 目标	124
② 机构	124
③ 计划	124
④ 度量和控制	124
⑤ 运营	124
(3) 根据以上范畴将问题分类.....	125
(4) 将判断和结论写成报告	125
(5) 将问题分类以确定总体结构优先次序.....	125
4. 6 定义系统总体结构	125
4. 6. 1 企业的信息结构图	125
4. 6. 2 确定主要系统	125
4. 6. 3 数据流向表示	125
4.6.4 识别子系统	126
BSP给出子系统的以下有关概念	126
根据其对数据类的产生和使用特点可将子系统分类如下.....	126
4.6.5 先决条件的分析	126
4.6.6 信息结构的使用计划	127
4. 7 确定系统的优先顺序	127
4. 7. 1 确定选择的标准	127
确定子系统优先顺序应考虑下述问题.....	127
而确定逻辑优先顺序的主要判断标准可归结成 4 方面.....	127
• 潜在的利益分析	127
• 对企业的影响	127
• 成功的可能性	127
• 需求	127
4. 7. 2 子系统的排序	127
4. 7. 3 优先子系统的描述	127
对优先子系统的基本描述应包括以下几项.....	127
• 一般性描述和目标	127
• 主要问题	127
• 潜在的效益	127
• 受影响的企业过程	127
• 输入和输出	127
• 影响的组织层次	127
• 先决条件	127
4. 7. 4 实施方法的选择	128
4. 8 信息资源管理	128
• 资源管理的方向和控制	128

• 建立企业信息资源指导委员会	128
• 建立信息资源的组织机构	128
4.9 制定建议书和开发计划	128
通过BSP研究而提出的具体建议有或可能有下面 4 方面	128
• 信息结构	128
• 信息系统管理	128
• 分布信息系统规划	128
• 总体结构优先顺序	128
每个开发计划应包括下列内容	128
• 项目的范围、主题和目标	128
• 预期成果	128
• 进度	128
• 潜在的效益	128
• 人员和职能	128
• 工具和技术	128
• 人员培训	128
• 通信	128
• 后勤	128
• 控制	128
4. 10 成果报告和后续活动	128
4.11 结论	129
第 5 章战略数据规划方法	129
5.1 概述	129
5.1.1 方法的来源	129
5. 1.2 内容概述	129
5.1.3 系统开发策略	129
考虑系统开发战略和策略的根本出发点在于.....	129
从普遍原理的角度，必须考虑下列几方面的问题.....	130
1. 企业建立信息系统总体规划的必要性.....	130
2.自顶向下规划与局部设计相结合.....	130
3.高层管理人员的参与	130
4.处理部门与管理者之间有交流与联系.....	130
James Martin认为，如果在数据处理部门和最高管理者之间存在着隔阂，下面的措施将会起到沟通作用	131
5.提高数据处理生产率的途径.....	131
• 应用的微小变化，可能导致程序的系列变化.....	131
• 数据格式的不一致、工作文件的不同表示形式，导致数据的共享性差，需要不同的应用程序来适应不同的数据格式并加以维护.....	131
• 企业的应用程序中存在着许多重复的逻辑结构，而其中有很多功能是相同的，本应由统一的程序来处理	131
• 高级数据库语言比大多数商用程序设计语言有更高的生产率.....	131
• 已存在的适当的数据库系统，对于一定类型的事务，可以缩短和简化系统分析过程；直接应用高效的软件开发工具(如应用第四代语言)可加快开发	131
6.选择快速收回投资的应用项目.....	131
7. 数据库费用的支付	131
8. 信息工程	131
5.2 自顶向下规划的组织	132
5.2.1 规划工作的组织	132
5.2.2 信息资源规划	132
5. 2. 3 数据规划的基本步骤	132

数据规划的步骤可粗略规划如下	132
(1) 企业模型的建立	132
① 开发一个表示企业各职能范围的模型	133
② 扩展上述模型, 使它们表示企业各处理过程	133
③ 继续扩展上述模型, 使它能表示企业各处理过程	133
(2) 确定研究的边界	133
• 在一个小型企业或密集型的一体化企业中, 研究的范围应包括整个企业	133
• 假若自顶向下规划的范围太广且涉及到几个独立的单位, 那么及时控制和实现数据库的开发是困难的	133
• 战略规划的研究范围与企业的管理方式有联系	133
(3) 建立业务活动过程	133
(4) 实体和活动的确定	133
(5) 对所得规划结果进行审查	133
5. 3 企业模型的建立	133
5. 3. 1 企业职能范围	133
5.3.2 业务活动过程	134
5. 3. 3 企业模型图	134
• 完整性	135
• 适用性	135
• 持久性	135
5. 3. 4 战略业务规划	135
5. 3. 5 关键成功因素	135
下面是James Martin给出的各类企业的关键成功因素的示例部分	135
5.4 主题数据库及其组合	136
5.4.1 主题数据库的概念	136
5.4.2 主题数据库的选择	136
5. 4. 3 主题数据库的组合	136
5.4.4 4 类数据环境	136
1. 文件环境	137
2. 应用数据库环境	137
3. 主题数据库环境	137
4. 信息检索系统环境	137
5.5 战略数据规划的执行过程	137
5.5.1 企业的实体分析	137
1. 企业的实体	138
2. 实体的确定	138
3. 实体间的联系	138
4. 实体图和数据模型	138
5. 自顶向下的规划和自底向上的设计	138
6. 结构化实体图	139
用计算机绘制和维护实体图的过程如下	139
7. 把实体聚集成超级组	139
实体图联系强度的五种划分如下	139
5. 5. 2 实体活动分析	140
1. 企业功能的分解	140
2. 绘制层次结构图	140
3. 基本活动	140
4. 相关活动的特征	141
5. 实体活动的映象	141
6. 可更新的自顶向下规划	141

5.5.3 企业的重组	141
5. 5. 4 亲合性分析	142
5.5.5 分布数据规划	143
1.分布式数据的 6 种形式	143
• 复制的数据	143
• 子集数据	143
• 重组数据	143
• 划分数据	143
• 独立模式数据	143
2.同步数据与不同步数据	143
3.没有地理位置的规划	144
4.分布矩阵	144
5.分布数据规划的相关内容	144
6.分布数据规划过程	144
5.6 战略数据规划过程提要	145
(1)得到企业最高管理层的委托.....	145
(2)选择一套合适使用的方法，并实施其方法.....	145
(3)定义业务职能范围.....	145
(4)拟定职能、活动和实体样本.....	145
(5)把每一职能范围划分成一些业务活动过程，这些业务活动过程可通过各个用户小组进行审查.....	145
(6)把所有业务过程分配给不同的用户分析员以便复审.....	145
(7)把每个业务过程分解成功能和活动.....	145
(8)可选择项.....	146
(9)建立超级组(主题数据库)与业务过程的对应关系.....	146
(10)绘制出现存的数据处理系统.....	146
(11)选择需要采访的高级管理人员.....	146
(12)画出每个业务活动过程出现的位置.....	146
(13)向用户分析员送交主题数据库、信息系统和分布系统产生的各种图表.....	146
(14)设置实施过程的优先级别.....	146
(15)建立职责，确保对自顶向下的规划进行不断的更新.....	146
(16)准备和呈交一份结束报告.....	146
5.7 结论	146
第 6 章信息工程方法	147
6.1 信息工程基本概念	147
6.1.1 信息工程发展过程	147
强调了实施信息工程的关键因素是.....	147
6.1.2 信息工程概念	147
然而，信息工程方法无论如何变化，它都应该具有以下特征或保持以下关键成分.....	148
6.1.3 信息工程的组成	148
归纳上述，可以将信息工程的组成总结如下.....	148
• 系统的方法论	148
• 完备的工具集	148
• 成熟经验总结	148
6.2 信息工程方法	148
6. 2. 1 信息工程金字塔表示	148
6.2.2 信息工程步骤:	148
(1)信息战略规划(Information Strategy plan, ISP)	148
(2)业务领域分析(Business Area Analysis, BAA)	148
(3)业务系统设计(Business System Design; BSD)	149

(4)技术系统设计(Technical System Design,TSD)	149
(5)系统构成(System Construction, SC)	149
(6)系统转换(System Transition, ST).....	149
6. 3 信息战略规划	149
6. 3. 1 信息战略规划的任务	149
根据上述流程, 信息战略规划的具体任务应包括如下内容	149
(1)在实施信息战略规划前, 必须制定信息战略规划项目的计划	149
(2)初始评估	149
(3)定义信息结构	149
(4)评估当前的环境	149
(5)确定业务系统结构	149
(6)完成信息战略规划项目, 提交信息战略规划报告	149
6. 3. 2 信息战略规划的实施	149
6. 3. 2. 1 数据和资料的收集	149
1.有关制定企业计划的资料	149
2.有关组织结构资料	150
3.有关业务活动的资料	150
以下按层次来说明与业务活动或称活动有关的概念	150
·业务功能	150
·业务活动	150
·主题域	150
·实体类型	150
·实体关系	150
4.现有系统环境资料	150
如果将计算机系统按其性质划分为 4 类, 则可区分现有系统的性质和功能	150
·决策性系统	150
·规划性系统	150
·控制性系统	150
·操作性系统	151
5.当前技术环境的资料	151
·硬件产品	151
·软件产品	151
·网络产品	151
·应用系统	151
6. 3. 2. 2 审查文档资料	151
·业务文档	151
·技术文档	151
·系统文档	151
6.3. 2. 3 通过采访获取资料	151
1.采访的准备	151
2. 采访的对象	151
·最高管理者	151
·中层管理者	151
·其他相关人员	151
3. 采访的技术	151
信息工程方法根据经验形成了所谓结构化采访技术, 其内容可归纳为	151
·采访人员	151
·采访准备	151
·进行采访	151
·整理结果	151

·时间和次数	152
6.4 建立企业模型	152
6.4.1 识别企业的组织机构	152
·信息来源	152
·输入信息	152
·输出信息	152
6.4.2 企业的任务、目标和关键成功因素.....	152
信息工程方法将企业任务、目标和CSF的识别规范为以下过程.....	152
·信息来源	152
·输入信息	152
·输出信息	152
可分 5 步完成上述过程	152
以下给出关键成功因素的例子	152
6.4.3 信息需求分析	154
在信息工程方法中，将信息需求的识别规范成以下过程.....	154
·输入信息	154
·输出信息	154
可分 3 步完成上述过程	154
(1) 识别和记录信息需求特征.....	154
(2) 建立信息需求/组织单元矩阵.....	154
(3) 给出评价每个目标完成的方法.....	154
·满意度	154
·重要性因素	154
·需求权值	154
6.4.4 企业模型的建立	154
信息工程方法将建立企业模型的过程规范为以下过程.....	154
·输入信息	154
可分 4 步完成上述过程	154
(1) 确定业务处理的主题域	154
(2) 建立主题域图表.....	154
(3) 确定高层次的业务功能	155
(4) 分解成业务过程	155
6.5 确定企业信息结构	155
6.5.1 企业业务功能的确定	155
信息工程方法将功能分解过程规范成以下过程.....	155
·输入信息	155
·输出信息	155
可分 3 步进行	155
(1) 利用IEF中的规划工具箱的活动层次图表表示工具，把功能层次图所示的功能继续分解成为更低层的功能或业务过程.....	155
在功能分解的过程中应参考下列分解原则.....	155
(2) 利用IEF中规划工具箱的活动依赖图工具构造功能依赖图.....	155
(3) 将业务功能映射到组织单元上，建立业务功能/组织单元矩阵.....	156
6.5.2 实体分析与实体关系	156
信息工程方法将实体分析规范成以下过程.....	156
·输入信息	156
·输出信息	156
可分 5 步实现	156
(1) 确定实体类型	156
(2) 定义实体类关系	156

项目规划中，应确定一个关系的如下信息.....	156
• 关系的名字	156
• 关系的基数	156
(3) 可利用IEF的规划工具箱的数据建模工具建立实体关系图.....	156
(4) 可利用IEF的规划工具箱的矩阵处理器，建立实体类/信息需求矩阵，其矩阵元素表示对应的信息需求所要求的实体类.....	156
(5) 记录业务功能所使用的实体类，建立实体类/业务功能矩阵.....	156
该矩阵应遵循如下规定	156
6. 5. 3 企业环境评估	157
信息工程在评价企业当前环境时，将包含以下内容.....	157
1. 现有系统和数据存储清单	157
将企业现有的计算机应用系统和数据库及文件系统经调查核实后列出清单，规范成以下过程	157
• 输入信息	157
• 输出信息	157
可分 3 步实现	157
(1) 确定和列出当前系统清单，包括系统名、说明和状态.....	157
(2) 列出当前数据库和文件清单，确定已使用和仅规划的，记录其名称、状态信息	157
(3) 用IEF中的规划工具箱的矩阵处理器工具，建立当前系统/数据存储矩阵	157
2. 信息结构范围	157
确定信息结构的范围可规范成以下过程.....	157
• 输入信息	157
• 输出信息	157
可分 3 步实现	157
(1) 建立业务功能/当前系统矩阵.....	157
(2) 建立实体类/当前存储矩阵.....	157
(3) 分析业务功能/当前系统矩阵和业务功能/当前数据存储矩阵	157
3. 信息需求列表	158
其规范的过程如下	158
• 输入信息	158
• 输出信息	158
可分 3 步实现	158
(1) 确定信息需求表中的每一项信息需求的满意度.....	158
(2) 定义需求权值	158
(3) 产生新的信息需求表.....	158
4. 信息系统组织评估	158
其规范的过程如下	158
• 输入信息	158
• 输出信息	158
可分 5 步实现	158
(1) 扩展企业的组织层次图和功能层次图，从而确定信息系统的组织单元和业务功能	158
(2) 建立与信息系统组织的责任(R人权力(A)、知识(E)和工作(W)相关的RAEW矩阵	158
(3) 考虑信息系统组织中需要新增加的角色.....	158
(4) 定义一个新的组织机构，并为每一个新的或变化的组织单元定义其职责	158
(5) 定义新的RAEW矩阵.....	158

6.5.4 现有技术环境分析	158
实现技术环境分析的规范过程如下.....	158
· 输入信息	159
· 输出信息	159
可分 4 步实现	159
(1) 列出以下企业使用的硬件设备和软件产品的技术清单.....	159
(2) 建立硬件设备/组织单元使用矩阵.....	159
(3) 确定技术环境中的非技术因素的约束.....	159
(4) 评价企业的技术地位	159
6.6 确定业务系统结构	159
6.6.1 业务领域划分与数据存储确定	159
将上述内容的获取规范为以下过程.....	159
· 输入信息	159
· 输出信息	159
可分 3 步实现	159
(1) 利用 IEF 规划工具箱中的自动聚合软件，自动调整业务功能/实体类的 CU 矩阵.....	159
(2) 依据业务功能/实体类 CU 矩阵，通过对实体类之间的亲合度分析来确定实体类的聚合，聚合后在一起的实体类组即为超级实体类组.....	159
(3) 利用 IEF 中的规划工具箱，建立实体类组/实体类矩阵	160
6.6.2 业务系统的识别和确定	160
可将其过程规范如下	160
· 输入信息	160
· 输出信息	160
可分两步实现	160
(1) 对业务功能之间的亲合度进行分析，从而确定业务功能组，即聚合的业务功能 ..	160
(2) 利用 IEF 工具箱中的工具，建立聚合业务功能组/业务功能矩阵	160
6.6.3 业务系统结构图的建立	160
上述过程可规范成以下过程	160
· 输入信息	160
· 输出信息	160
可分 3 步实现	160
(1) 根据处理特征，对预期的业务系统进行分类.....	160
(2) 建立预期系统之间的信息流.....	160
(3) 人工调整不规则情况，使预期系统成为实际系统.....	160
6.6.4 确定和组成业务领域	160
其规范过程如下	161
· 输入信息	161
· 输出信息	161
可分 4 步实现	161
(1) 从识别和确定预期的数据存储中，产生企业初步业务领域划分	161
(2) 建立预期业务系统/预期数据存储 CU 矩阵.....	161
(3) 建立业务领域/预期系统矩阵.....	161
(4) 建立业务领域/业务功能和业务领域/实体类矩阵	161
6.7 确定系统的技术结构	161
6.7.1 数据分布与数据分布矩阵	161
1. 分散管理的数据具有的特征	161
2. 集中管理的数据具有的特征	161
6. 7. 2 分布矩阵与业务系统分布矩阵.....	161
通过建立数据分布矩阵来分析企业数据分布状况，其规范过程如下	161
· 输入信息	161

·输出信息	161
(1) 确定每个预期的数据库和文件的地点要求	161
(2) 在预期的数据存储/地点矩阵上, 给出数据的分布决策	162
6.7.3 业务系统分布矩阵的确定	162
可分两步实现	162
(1) 确定每一地理位置的业务功能, 建立业务功能/地点矩阵	162
(2) 确定每一地理位置上对预期业务系统的要求	162
具体过程可规范如下	162
可分两步实现	162
(1) 对每个预期的业务系统, 进一步确认已收集到性能度量时收集的可利用信息, 包括	162
(2) 提供与每个业务系统有关的性能需求技术说明, 完成技术需求说明书, 如包括	162
6.7.4 技术分配要求的确定	162
确定技术分配要求可规范成以下过程	163
·输入信息	163
·输出信息	163
可分 6 步实现	163
(1) 建立因素矩阵, 对数据分布进行定性分析。	163
(2) 对数据分布进行定量分析, 合理安排数据 and 应用程序的位置。	163
(3) 建立有关地点的系统/数据存储矩阵	163
(4) 建立有关地点的业务系统和地点的数据库或文件之间交互关系矩阵	163
(5) 根据上述分析, 绘制成各地点的计算机、文件、数据库的组成, 反映各地点系统配置情况	163
(6) 制定出各计算机(主机、客户机、服务器)、各地理位置的业务系统连接成的企业整体网络规划	163
6. 7. 5 方案的确定与评估	163
6.8 信息战略规划报告	163
6. 8. 1 报告的组成和内容	163
一般认为, 信息战略规划报告应由 3 个主要部分组成	163
·摘要	163
·规划	163
·附录	163
摘要通常不要多于 5 页, 其内容应涉及下列主题	164
规划其主要内容包括	164
6.8.2 规划成果展示	164
6. 9 信息工程方法和环境	164
6.9.1 方法与工具的结合	164
6.9.2 信息工程设施	164
以下介绍信息工程设施的不同结构的设计	164
1. 知识件工具集(关Knowledge Ware Toolset)	164
信息库的基本内容包括	165
2. Composer	165
Composer主要由信息库和 5 个工具箱及通信设施组成	165
·信息库	165
·规划工具箱	165
·设计工具箱	165
·构成工具箱	165
·实现工具箱	165
6.10 小结	166

第7章应用原型化方法	166
7.1 概述	166
7.1.1 原型化的概念	166
7.1.2 原型化的内容	166
• 严格定义/预先定义	166
• 应用原型化	166
7.2 原型定义策略	167
7.2.1 需求定义的重要性	167
为了进行需求定义，有必要知道下述情况	167
• 约束	167
• 系统输出	167
• 系统输入	167
• 系统数据需求	167
• 数据元素	167
• 转换	167
• 功能	167
• 性能/可靠性	167
从实用上讲，一般认为，需求定义必须有下列的一些属性	167
• 完备的	167
• 一致的	167
• 可理解	167
• 可测试	167
• 可维护	167
• 正确的	167
• 必要的	167
7.2.2 严格定义的策略	167
1. 所有的需求都能被预先定义	168
2. 修改定义不完备的系统代价昂贵且实施困难	168
3. 项目参加者之间能够清晰而准确地进行通信	168
4. 静态描述或图形模型对应用系统的反映是充分的	168
文字叙述	168
图形模型	168
逻辑规则	168
数据字典	168
5. 严格方法的生命周期的各阶段的划分都是正确的	168
7.2.3 原型定义的策略	169
1. 并非所有的需求在系统开发以前都能准确地说明	169
2. 有快速的系统建造工具	169
用于完成原型化的较好的工具应包含以下几个部分	169
• 集成数据字典	169
• 高适应性的数据库管理系统	169
• 非过程的报告书写器	169
• 非过程查询语言	169
• 屏幕生成器	169
• 超高级语言	169
• 自动文档编排	169
• 原型人员工作台	169
3. 项目参加者之间通常都存在通信上的障碍	169
4. 需要实际的、可供用户参与的系统模型	169
5. 需求一旦确定，就可以遵从严格的方法	169

6.大盘的反复是不可避免的, 必要的, 应该加以鼓励.....	169
7.2.4 原型化的优点及其意义	170
应用原型化是一种系统开发的高级策略, 优点如下	170
7.2.5 原型化与预先定义的比较	170
7. 3 原型生命周期	170
7. 3. 1 原型生命周期划分	170
1.合适的(好的)选择.....	171
• 系统结构	171
• 逻辑结构	171
• 用户特征	171
• 应用约束	171
• 项目管理	171
• 项目环境	171
2.识别基本需求	171
3.开发工作模型一	171
4.模型验证	171
在迭代的初期	172
• 模型通过用户进行验收。.....	172
• 总体检查, 找出隐含错误。.....	172
• 在操作模型时, 使用户感到熟悉和愉快。.....	172
在迭代的后期	172
• 应发现丢失和不正确的功能。.....	172
• 测试思路和提出建议。.....	172
• 改善用户/系统界面。.....	172
5.修正和改进	172
6.判定原型完成	172
7. 判别细部说明	172
8.严格说明细部	172
9.判定原型效果	172
10. 整理原型和提供文档	172
原型化方法生命周期提供了一种完整的、灵活的、近于动态的需求定义技术, 它具有以下特征	172
7.3.2 原型化的准则与策略	173
7. 3. 2. 1 原型化的准则	173
1.大多数的应用系统都能从一个小的系统结构集合导出.....	173
可归纳成以下 8 个基本的模型结构.....	173
• 成批编辑/修改	173
• 成批生成报表	173
• 成批转换	173
• 成批对接	173
• 联机结构化的修改/查询	173
• 联机特殊查询	173
• 联机界面	173
• 联机报表生成	173
2.多数系统使用一个常用和熟悉的功能集合.....	173
• 确定应用系统需要的基本功能, 再分析应用系统的个别差异.....	173
• 分析应用系统中不常用的功能, 当在初始模型中为一热点时, 这些功能可放在迭代阶段来完成	173
3. 大多数的输入编辑能从一个小的编辑模型集中导出.....	173
• 识别每个输入所需要的一般编辑的子集.....	173

• 识别应用需要的特殊编辑。它可以留待以后的迭代中来完成.....	173
4. 基于一个 4 步的报表模型生成应用系统的报表.....	173
从数据库生成报表的 4 步过程为.....	173
(1)从数据库选择和拆卸数据.....	174
(2)按说明分类每个报告.....	174
(3)为了打印定格式和编辑数据.....	174
(4)打印该报告.....	174
5. 有一个“正确”的设计结构集合，对原型将会产生积累作用.....	174
7.3.2.2 原型化的策略.....	174
下列策略能用于快速建立原型及原型改进.....	174
1.用第三范式规范数据，建立应用系统的数据模型.....	174
2.大多数富有成效的建立模型的途径是利用组合工程.....	174
导出或得到一个系统需要的实体的的大多数的有效途径是.....	174
(1)利用一个已存在的实体.....	174
(2)全部从已存在的系统实体中装配此实体.....	174
3.最有成效的建立模型的途径是“剪裁和粘贴”.....	174
4.用系统举例.....	174
5.字典驱动的软件结构.....	174
6.文档的自动化.....	175
7.小的原型化队伍.....	175
8.交互式原型开发者的工作台.....	175
9.陈述性规格说明.....	175
过程性说明.....	175
陈述性说明.....	175
10.终端用户报表生成器.....	175
11.专业原型化人员.....	175
12.开发人员参加原型化.....	175
7.3.3 混合原型化策略.....	176
已介绍的原型生命周期意味着对自身的以下若干约束.....	176
建立一个完整的模型。.....	176
原型人员要建立初始模型。.....	176
原型化要从定义阶段开始。.....	176
实际系统将用自家的资源来建立。.....	176
下面是一些可供选择的方法，它们改变了上述某些约束。.....	176
1.仅对屏幕的原型化.....	176
2.使用购买到的应用系统作为初始模型.....	176
3.可行性分析中的原型化.....	176
4.子系统原型化.....	176
5.原型与需求建议.....	176
6.最终用户进行原型化.....	176
7.3.4 原型的实施.....	177
7.4 原型化中心.....	177
7.4.1 原型化中心的组织.....	177
• 开发中心.....	177
• 生产中心.....	177
• 信息中心.....	177
7.4.2 原型化中心的人员配备.....	178
现从以下几个方面说明组小的必要性.....	178
7. 4. 3 硬件需求.....	178
1.终端.....	178

• 用户终端	178
• 原型软件终端	178
• 打印终端	178
2. 个人计算机	178
7.4.4 软件需求	178
原型化软件需求内容可简要的归纳为	178
• 数据字典驱动。	179
• 有结构地支持组合工程。	179
• 从现有组件“剪裁和粘贴”出新的组件。	179
• 提供交互原型化工作台。	179
• 使用描述性文档而非过程化文档。	179
• 自动生成应用文档。	179
生成完整应用所必需的所有软件成分应包含在原型化结构中	179
7. 4. 5 原型工作环境	179
1. 项目工作室的建立	179
2. 快速响应的工作环境	179
3. 规范的原型构造过程	179
4. 文档资源	179
5. 演示/展示设施	179
6. 集中式/分散式原型开发中心	180
分散式的开发人员通过负责开发的组织接近用户具有以下明显的优点	180
7. 零件部门	180
7.5 原型化与项目管理	180
7.5.1 项目管理的必要性	180
7.5.2 项目管理的内容	180
1. 估计过程	180
对简单的估算规则有如下情况	180
(1) 对建立初始原型的估计	181
(2) 对原型的修改的估计	181
(3) 对建立初始原型和修改原型的估计	181
2. 费用重新分配	181
3. 变化控制	181
4. 活动停止	181
7.6 结论	181
原型化方法的介绍即将结束，可以得出 6 个结论	181
• 原型化从用户角度考虑是非常适当的	181
• 原型化从开发者角度考虑也是合适的	181
• 原型化可用于大规模的项目开发	181
• 原型化是可行的	181
• 原型的制作者相当于一个建筑师	181
• 原型制作的核心策略为处理过程提供了方便的工作环境	181
第 8 章 软件工程	181
8.1 软件生存期过程	181
1. 主要生存期过程(primary process)	182
(1) 获取过程	182
(2) 供应过程	182
(3) 开发过程	182
(4) 运行过程	182
(5) 维护过程	182
2. 支持生存期过程(supporting process)	182

(1) 文档编制过程	182
(2) 配置管理过程	182
(3) 质量保证过程	182
(4) 验证过程	182
(5) 确认过程	182
(6) 联合评审过程	182
(7) 审核过程	182
(8) 问题解决过程	182
3. 组织生存期过程(organizational process)	182
(1) 管理过程	183
(2) 基础设施过程	183
(3) 改进过程	183
(4) 培训过程	183
8.2 软件过程能力评估	183
8. 2. 1 软件过程评估的意义	183
8.2.1.1 软件过程改进的需要	183
1. 软件过程不断改进是软件工程的基本原理之一	183
• 按软件生存周期分阶段制定计划并认真实施	183
• 逐阶段进行确认	183
• 坚持严格的产品控制	183
• 使用现代程序设计技术	183
• 明确责任	183
• 用人少而精	183
• 不断改进开发过程	183
2. 软件过程改进是软件生存周期的基本过程之一	183
8.2.1.2 降低软件风险的需要	184
1. 软件采购者的需要	184
2. 软件承制者的需要	184
8. 2. 2 软件过程评估方法的产生	184
8.2.3 软件能力成熟度模型CMM(Capability Maturity Model)简介	184
8.2.3.1 模型概要	184
• 软件过程	184
• 软件过程能力	184
• 软件过程性能	185
• 软件过程成熟度	185
• 软件能力成熟度等级	185
• 关键过程域	185
• 关键实践	185
• 软件能力成熟度模型	185
8.2.3.2 模型的产生和原理	185
8. 2.3.3 不成熟和成熟软件组织的比较	185
8. 2. 3. 4 软件过程成熟度的 5 个等级	186
1. 等级 1—初始级	186
2. 等级 2—可重复级	186
3. 等级 3—已定义级	187
4. 等级 4—已定量管理级	187
5. 等级 5—优化级	187
8. 2.3.5 跳越成熟度等级	187
1. 跨越等级的现象	187
2. 跳越等级的错误	187

8.2.3.6 关键过程域	188
(1) 等级 2 上的关键过程域集中关注软件项目所关心的、与建立基本项目管理和控制有关的事情	188
·需求管理	188
·软件项目策划	188
• 软件项目跟踪和监督	188
• 软件子合同管理	188
• 软件质量保证	188
• 软件配置管理	188
(2) 等级 3 的关键过程域既涉及项目，又涉及组织，因为组织建立起了对所有项目都有效的、使软件工程过程和管理过程规范化的基础设施	188
• 组织过程焦点	188
• 组织过程定义	189
·培训大纲	189
·集成软件管理	189
• 软件产品工程	189
• 组际协调	189
·同行专家评审	189
(3) 等级 4 上的关键过程域的关注焦点是建立起对软件过程和正在构造的软件工程产品的定量了解。该等级上的两个关键过程域一定量过程管理和软件质量管理一是互相紧密依赖的	189
·定量过程管理	189
·软件质量管理	189
• 缺陷预防	189
·技术变更管理	189
·过程变更管理	189
8.2.3.7 关键实践	189
• 执行约定	189
• 执行的活动	190
·测量和分析	190
• 验证实施	190
通过实施这些关键实践，就能实现软件项目策划这个关键过程域的下列 3 个目标	190
• 对策划和跟踪软件项目所用的软件估计已建立文档	190
·软件项目的活动和约定是有计划的并已建立文档	190
• 受影响的组织和个人都同意他们关于软件项目的约定	190
8.2.3.8 CMM 的应用	190
CMM 有两个基本用途：软件过程评估和软件能力评价	190
1. 软件过程评估和软件能力评价的基本方法和步骤	190
图 8.4 概要地描述评估和评价中的共同步骤	190
(1) 建立一个小组	190
(2) 填写提问单	190
(3) 进行响应分析	190
(4) 进行现场访问	190
(5) 提出调查发现清单	191
(6) 制作关键过程域 (KPA) 剖面图	191
总之，软件过程评估和软件能力评价方法两者的共同点如下	191
• 采用成熟度提问单作为现场访问的出发点	191
• 采用 CMM 作为指导现场调查研究的导引图	191
• 利用 CMM 中的关键过程域生成明确地指出软件过程强项和弱项的调查发现清单	191

• 在对关键过程域目标满足情况进行分析的基础上，衍生出一个剖面	191
• 根据调查发现清单和关键过程域剖面，向合适的对象提出结论意见	191
2.软件过程评估和软件能力评价之间的差异	191
3.其他应用	191
8. 2. 3. 9 软件过程成熟度提问单	191
• 约定(commitment)	191
• 事件驱动的评审和活动(event-driven review/ activity)	191
• 方针(policy)	192
8.2.4 软件过程评估的国际标准概述	192
8.2.4.1 软件过程评估国际标准的制定	192
该标准的目的有 3 点	192
8.2.4.2 软件过程评估标准的组成	192
• 部分 1: 概念和引导指南(参考件)	192
• 部分 2: 过程和过程能力的参考模型(标准件)	192
• 部分 3: 进行评估(标准件)	192
• 部分 4. 进行评估的指南(参考件)	192
• 部分 5: 评估模型和指示器指导(参考件)	192
• 部分 6: 评估人员资格指南(参考件)	192
• 部分 7: 过程改进指南(参考件)	192
• 部分 8: 供应者过程能力评定指南(参考件)	192
• 部分 9: 词汇表(标准件)	193
8. 2 . 4.3 参考模型	193
1.过程维	193
2.过程能力维	193
8.2.4.4 评估框架	194
1.过程评估环境	194
2.过程改进环境	194
3.过程能力评定环境	194
4.对评估过程的要求	194
8. 2.4.5 软件过程评估标准的特点	195
8. 3 软件配置管理	195
8.3.1 软件配置管理的概念	195
8.3.1.1 软件配置项(software configuration item)	195
8.3.1.2 软件配置管理	196
1.什么是软件配置管理	196
①国际标准ISO 9000-3: 1997	196
②W. Babich的解释	197
③GB/T 11457:1995《软件工程术语》国家标准	197
• 对配置项的功能特性和物理特性进行标识和文件编制工作	197
• 控制这些特性的更动情况	197
• 记录并报告这些更动进行的处理和实现的状态	197
2.软件配置管理的任务	197
• 制定软件配置管理计划	197
• 确定配置标识规则	197
• 实施变更控制	197
• 报告配置状态	197
• 进行配置审核	197
3.软件配置管理与软件开发过程	197
8.3.1.3 软件配置管理的意义	198
1.软件项目的特点	198

必须重视软件项目以下的特点是.....	198
2.忽视软件配置管理可能导致的混乱现象.....	198
3.几类配置问题及其解决的对策.....	198
·多重维护	198
• 共享数据	198
·同时修改	198
·丢失版本号或是不知版本号	198
8.3.2 软件配置管理计划	198
配置管理计划标准IEEE 828-1990	199
8.3.3 软件配置标识	199
8.3.3.1 确定配置项	199
Roger S. Pressman认为至少以下所列配置项应该是受控的.....	199
8.3.3.2 配置项命名及其相关信息.....	200
1.配置项命名	200
2.对象的标识形式	200
每个对象用一组特征信息(名字、描述、一组资源、实现)唯一地标识	200
3.对象演变图	201
8.3.4 变更管理	201
8.3.4.1 软件变更	201
1.软件变更的不可避免性	201
2.软件变更的复杂性	201
3.变更管理的任务	202
• 分析变更	202
·记录和追踪变更	202
• 采取措施保证变更在受控状态下进行.....	202
8.3.4.2 配置库	202
1.配置库的作用	202
(1)记录与配置相关的所有信息，其中存放受控的软件配置项是很重要的内容 ...	202
(2)利用库中的信息可评价变更的后果，这对变更控制有着重要的意义	202
(3)从库中可提取各种配置管理过程的管理信息，可利用库中的信息查询回答许多配置管理的问题，例如	202
2.三类库	202
• 开发库(development library).....	202
• 受控库(controlled library)	202
8.3.4.3 配置基线	203
1.基线	203
2.基线的种类	203
·功能基线	203
·分配基线	203
• 产品基线	203
3.基线与配置项	203
8.3.4.4 变更控制	204
1. 变更控制组	204
2.变更请求与变更控制	204
(1)利用配置库实现变更控制。	204
(2)变更请求。	204
变更请求的主要内容有 3 个方面.....	204
• 变更描述	204
• 对变更的审批.....	204
• 有关变更实施的一些信息.....	204

(4)故障报告	204
故障报告包含的内容有	205
·FR ID(故障报告标识)	205
·CCB评估意见	205
3.变更记录	205
8.3.5 版本管理	205
8.3.5. 1 软件版本	205
8.3.5.2 版本管理(version management)也称版本控制(version control)	205
1. 号码版本标识	205
2.符号版本标识	205
8. 3. 6 配置审核	206
8. 3. 6. 1 什么是配置审核	206
这种验证包括	206
• 对配置项的处理是否有背离初始的规格说明或已批准的变更请求的现象	206
• 配置标识的准则是否得到了遵循	206
• 变更控制规程是否已遵循, 变更记录是否可供使用	206
• 在规格说明、软件产品和变更请求之间是否保持了可追溯性	206
配置审核工作主要集中在两个方面	206
8.3.6.2 为什么要实施配置审核	206
• 防止出现向用户提交不适合的产品, 如交付了用户手册的不正确版本	206
• 发现不完善的实现, 如开发出不符合初始规格说明或未按变更请求实施变更	206
• 找出各配置项间不匹配或不相容的现象	206
• 确认配置项已在所要求的质量控制审查之后作为基线入库保存	206
• 确认记录和文档保持着可追溯性	206
8.3.6.3 如何实施配置审核	206
1.实施配置审核的时机	206
• 软件产品交付或是软件产品正式发行前	206
• 软件开发的阶段工作结束之后	206
• 在维护工作中, 定期地进行	206
2.实施配置审核的责任人	206
3.配置审核工作的开展	206
(1)由项目经理决定何时进行配置审核工作	206
(2)质量保证组或软件组的配置管理组指定该项目的配置审核人员	206
(3)项目经理和配置审核员决定审核范围	206
(4)配置审核员准备配置审核检查单	206
(5)配置审核员安排时间审核文档和记录, 审核活动可能涉及到	206
(6)配置审核员在审核中发现不符合现象, 并作记录	206
(7)由项目经理负责消除不符合现象	206
(8)配置审核员验证所有发现的不符合现象确已得到解决	206
8. 3. 7 配置状态报告	206
8. 3. 7. 1 什么是配置状态报告	206
8. 3. 7. 2 配置状态报告信息	207
1.状态说明的实体关系	207
2.状态说明数据词典	207
3.定期提交的配置状态报告的内容示例	209
4.配置状态报告提供信息的利用示例	209
8.3.7.3 状态说明	209
8.4 面向对象的开发方法	209
8.4.1 面向对象分析	209
8. 4. 1. 1 电梯控制系统需求说明	210

8.4.1.2 标识对象和类	210
1.如何选择候选对象	210
应该严格地审查每个候选对象。对每个候选对象，可以审查以下这些方面的问题	211
还需要写下在系统范畴之外所有会发生的事件，再考虑以下问题	211
现在我们仍将注意力放在系统上而暂且不去考虑对象，那么	211
2.电梯控制系统的对象类层	211
8.4.1.3 发现和标识结构	213
1.如何发现和标识结构	213
发现和标识一般—特殊结构主要从以下几个方面考虑	213
• 考察类的属性和服务	213
• 考虑领域范围内的复用	213
发现和标识整体一部分结构主要从以下几个方面考虑	213
• 物理组装关系	213
• 空间包含关系	213
• 组织机构的上下级关系	213
• 概念上的组装关系	213
2.电梯控制系统的结构层	214
8.4.1.4 划分主题层	214
8.4.1.5 定义属性和实例连接	214
1.如何标识属性	215
可以从以下几个方面考虑标识属性	215
• 所标识的问题领域的概念中有许多都可以表达事物的特性	215
• 根据系统的功能，确定对象应该有什么属性	215
• 为了惟一地标识一个对象，往往需要建立一个标识数属性	215
• 为了区别对象的状态，往往需要增加一个属性	215
• 为了表示实例连接，需要设置相应的属性	215
对于初步标识的属性应该进行审查，可以审查以下这些方面的问题	215
• 继承结构中父类子类属性的一致性	215
• 可以从其他属性中导出的属性应该去掉	215
• 在标识一个对象时，如果其属性的值“不适用”，就应当对其重新考虑 ..	215
2.如何标识实例连接	215
3.电梯控制系统的属性层	216
8.4.1.6 定义服务和消息	216
1.如何发现和标识服务层	217
2.电梯控制系统的服务层	217
8.4.1.7 事件响应对象交互图	217
8.4.2 面向对象的设计	218
1. OOD模型	218
2.建立电梯控制系统的OOD模型	219
8.4.3 OOD文档的编写	220
8.5 软件复用技术	220
8.5.1 软件复用的概述	220
8.5.2 软件开发过程	221
8.5.2.1 以往的软件开发技术不能满足复用的需要	221
1. 工程、	221
• 缺乏界定“复用”的机制	221
• 缺乏制作可复用构件的方法	221
• 不成熟的体系结构设计致使可复用构件缺乏足够的灵活性	221
• 缺乏实施复用的工具	221

2.过程	221
3.组织管理	221
4.经营方式	221
8.5.2.2 软件复用需要改变软件开发过程.....	221
• 可复用资产的开发	222
• 复用	222
• 支持	222
• 管理	222
8.5.2.3 领域工程和应用系统工程	222
1.领域工程	222
2. 应用系统工程的变化	223
8.5.3 构件技术	223
1. 应用系统和应用系统族	223
2.应用系统与构件	223
3.构件系统	224
4.构件系统的门面	224
5.可变性和客户化	225
6.打包和编写文档	225
8.5.4 分层式体系结构	225
1.软件体系结构	225
2.良好的软件体系结构的重要作用.....	226
3.分层式的体系结构	226
• 第一层是最顶层或最高层，对于每种软件体系结构来说，最顶层总是应用系统层..	226
• 第二层是次顶层或次高层，它应当是“业务专化”(business-specific)	226
• 第三层是“中件层”(middleware-layer)	227
• 第四层是最低层，它是系统软件层.....	227
8.5.5 渐进地实施复用和复用单位的组织结构.....	227
8.5.5.1 软件复用需要改变开发单位的组织结构.....	227
8.5.5.2 渐进地系统地采用复用技术.....	228
1.采用系统的复用技术	228
2. 一个实例	228
(1)黑盒代码的复用.....	228
(2)库和工作成品的管理.....	228
(3)体系结构和系统.....	228
(4)应用工程和构件工程技巧.....	228
(5)面向复用的过程和组织的管理	228
(6)新工具和技术.....	228
3.渐进地采用复用技术	229
4.迭代式过渡	229
(1)第一轮迭代的目的是，初步理解应用族，重点关注体系结构，开发单位开始认识到复用	230
• 复用业务的高层设想(即概要式的设想)	230
• 初步的市场分析	230
• 目标组织结构的业务模型	230
• 约定关键的客户	230
• 建立过渡团队	230
(2)第二轮迭代的目的是建立体系结构	230
(3)第三轮迭代的目的是，组织更多的人员参与构件系统工程，开发并改进构件系统	230
(4)第四轮迭代的目的是，建成稳定状态的复用组织结构，并掌握若干源于客户合同	230

作为考验新的构件系统和(前一轮迭代开发的)构件系统新版的ASE过程.....	230
5.过渡计划实例	230
• 体系结构如何演变	230
• 要开发什么应用系统和构件系统.....	230
• 要定义哪些过程	230
• 要建立和培训哪些团队	230
• 如何安排复用的认识和实施进度.....	230
8.5.5.3 充分利用可共享复用成果	230
8.5.5.4 实施系统复用需要遵循的原则.....	231
第9章数据库与数据仓库	231
9.1 关系数据库系统	231
9.1.1 关系数据库系统概述	231
9.1.1.1 关系数据模型	231
1.关系数据结构	231
2.关系操作集合	231
所谓非过程化是指	232
• 用户不必请求DBA为他建立特殊的存取路径, 存取路径的选择由DBMS的优化 机制来完成	232
• 用户也不必求助于循环、递归来完成数据的重复操作	232
3.关系的完整性约束	232
• 与现实世界的应用需求的数据的相容性和正确性.....	232
• 数据库内数据之间的相容性和正确性.....	232
9.1.1.2 关系模型的数据结构和基本术语.....	232
(1)关系(relation)	232
(2)属性(attribute)和值域(domain).....	232
(3)关系模式(relation schema).....	233
(4)元组(tuple)	233
(5)分量(component)	233
(6)候选码(candidate key)或候选键.....	233
(7)主码(primary key)或主键.....	233
(8)主属性(primary attribute)和非主属性(nonprimary attribute)	233
(9)外码(foreign key)或外键.....	233
(10)参照关系(referencing relation)与被参照关系(referenced relation)	233
(11)关系的形式定义	233
• 用集合论的观点定义关系.....	233
• 用值域的概念来定义关系.....	233
(12)数据库对关系的限定	234
• 每一个属性是不可分解的.....	234
• 每一个关系模式中属性的数据类型以及属性的个数是固定的, 并且每个属性必须 命名, 在同一个关系模式中, 属性名必须是不同的	234
• 每一个关系仅仅有一种记录类型, 即一种关系模式.....	234
• 在关系中元组的顺序(即行序)是无关紧要的	234
• 在关系中属性的顺序可任意交换, 交换时应连同属性名一起交换才行	234
9.1.2 关系模型的完整性约束	234
1.实体完整性规则(entity integrity rule).....	235
对于实体完整性规则说明如下.....	235
• 实体完整性规则是针对关系而言的.....	235
• 现实世界中的实体是可区分的, 即它们具有某种惟一性标识.....	235
• 相应地, 关系模型中以主码作为惟一性标识.....	235
2.参照完整性规则(reference integrity rule)	235

• 或者取空值 (F 的每个属性值均为空值)	235
• 或者等于 S 中某个元组的主码值	235
3. 用户定义的完整性规则	236
• 当执行插入操作时	236
• 当执行删除操作时	236
• 当执行更新操作时	236
9.1.3 关系数据库标准语言 SQL	236
其主要特点如下	236
• 综合统一	236
• 高度非过程化	237
• 面向集合的操作方式	237
• 以同一种语法结构提供两种使用方式	237
9.1.3.1 SQL 数据库的体系结构	237
9.1.3.2 SQL 的数据定义	237
1. 基本表	238
2. 索引	238
9.1.3.3 SQL 的数据操纵	239
1. SQL 的查询语句	239
2. SQL 的修改语句	239
9.1.3.4 视图	239
1. 创建视图	239
但在下列 3 种情况下必须明确指定组成视图的所有列名	240
2. 删除视图	240
3. 查询视图	240
4. 修改视图	240
5. 视图的作用	240
• 视图能够简化用户的操作	240
• 视图使用户能以多种角度观察同一数据	241
• 视图对重构数据库提供了一定程度的逻辑独立性	241
9.1.3.5 SQL 的数据控制语句	241
1. 授予权限	241
2. 收回权限	241
9.1.3.6 嵌入式 SQL	241
把 SQL 嵌入主语言使用时必须解决以下 3 个问题	242
1. 区分 SQL 语句与主语言语句	242
2. 数据库工作单元和程序工作单元之间的通信	242
3. 协调 SQL 语言和主语言的处理方式	242
与游标有关的 SQL 语句有下列 4 个	242
(1) 游标定义语句 (如例 5 程序中④)	242
(2) 游标打开语句 (如例 5 程序中⑤)	242
(3) 游标推进语句 (如例 5 程序中⑥)	242
(4) 游标关闭语句 (如例 5 程序中⑦)	242
9.2 规范化理论与数据库设计	243
9.2.1 “不好”的关系模式	243
关系模式 SUPPLIER 有如下“毛病”	243
9.2.2 函数依赖	243
1. 函数依赖的定义	243
2. 函数依赖的逻辑蕴含	244
3. 码	244
4. 函数依赖的公理系统	244

设F是属性组U上的一组函数依赖，于是有如下推理规则	244
根据Armstrong公理系统的 3 条推理规则可以得到下面 3 条很有用的推理规则	245
9. 2. 3 关系模式的规范化	245
1.第一范式(1NF)及进一步规范化.	245
2. 第二范式(2NF).....	245
3.第三范式(3NF).....	246
4. Boyce-Codd范式(BCNF)	246
9.2.4 多值依赖和 4NF	246
1. 多值依赖	246
多值依赖具有以下性质	247
2.第四范式(4NF).....	247
9.2.5 关系模式的分解	247
1.模式分解的等价标准	247
2.关于模式分解的几个事实	248
9.2.6 数据库设计过程	249
1.需求分析	249
需求分析阶段的任务是	249
调查的重点是“数据”和“处理”。通过调查要从用户中获得对数据库的下列需求 ..	249
• 信息需求	249
• 处理需求	249
除数据流图外，还采用一些规范表格对于数据分析的结果描述做补充描述	249
• 数据项	249
• 数据结构	249
• 数据流说明	249
• 数据存储说明	249
• 加工过程	249
2.概念结构设计	249
概念模型应具备以下特点	249
• 有丰富的语义表达能力.....	250
• 易于交流和理解	250
• 易于变动	250
• 易于向各种数据模型转换，易于从概念模型导出与DBMS有关的逻辑模型.....	250
设计概念结构的策略有如下几种.....	250
• 自顶向下	250
• 自底向上	250
• 由里向外	250
• 混合策略	250
采用E-R方法的数据库概念结构设计可分为三步进行.....	250
(1)设计局部E-R模型	250
(2)设计全局E-R模型	250
当将局部的E-R图集成为全局的E-R图时，可能存在 3 类冲突.....	250
• 属性冲突	250
• 结构冲突	250
• 命名冲突	250
(3)全局E-R模型的优化	250
3.逻辑结构设计	250
E-R模型向关系模型转换的规则是.....	251
4.物理结构设计	251
(1)存储记录的格式设计.....	251
(2)存储方法设计	251

• 顺序存放	251
• 散列存放	251
• 聚簇(cluster)存放	251
(3) 存取方法设计	251
5. 数据库实施	252
• 建立实际的数据库结构	252
• 装入试验数据对应用程序进行测试, 以确认其功能和性能是否满足设计要求, 并检查对空间的占有情况	252
• 装入实际数据, 即数据库加载, 建立起实际的数据库	252
6. 数据库运行和维护	252
• 数据库的转储和恢复	252
• 数据库的安全性、完整性控制	252
• 数据库性能的监督、分析和改造	252
• 数据库的重组和重构造	252
9.2.7 规范化理论在数据库设计中的应用	252
9.3 数据仓库与联机分析处理、数据挖掘	253
用于决策支持的数据的存储和检索将涉及以下领域	253
• 联机分析处理(OLAP)	253
• 数据挖掘	253
9.3.1 OLAP系统与OLTP系统的比较	253
下面概述OLAP系统与OLTP系统的主要区别	253
• 所面向的用户和系统	253
• 数据内容	253
• 视图	254
• 访问模式	254
9. 3. 2 多维数据模型	254
1. 度量属性(measure attribute)	254
2. 维属性(dimension attribute)	254
3. 维的概念分层(concept hierarchy of a dimension)	254
4. 多维数据(multidimensional data)	254
5. 数据立方体(data cube)	254
6. 方体和数据立方体(cuboid and data cube)	254
9.3.3 数据仓库	254
9. 3. 3. 1 数据仓库基本概念	255
1. 数据仓库是面向主题的	255
2. 数据仓库的数据是集成的	255
3. 数据仓库的数据是相对稳定的	255
4. 数据仓库数据是反映历史变化的	255
数据仓库的数据是反映历史变化的, 这主要表现在以下 3 方面	255
• 数据仓库随时间变化不断增加新的数据内容	255
• 数据仓库随时间变化不断删去旧的数据内容	255
• 数据仓库中包含有大量的综合数据, 这些综合数据中很多跟时间有关	255
9. 3. 3. 2 数据仓库的数据模式	255
9. 3. 3. 3 数据仓库体系结构	256
数据仓库系统通常采用 3 层的体系结构	256
底层的数据仓库服务器	256
中间层OLAP服务器	256
顶层的前端工具	256
从结构的角度看, 有三种数据仓库模型	256
企业仓库(enterprise warehouse)	256

数据集市(data mart)	256
虚拟仓库(virtual warehouse)	257
9.3.3.4 数据仓库系统的开发	257
对于数据仓库系统的开发,一般推荐采用增量的、演进的方式	257
下面讨论创建一个数据仓库时的一些关键问题	257
• 何时如何收集数据	257
• 使用何种模式	257
• 数据清理	257
• 如何传播更新	257
• 汇总何种数据	257
9.3.4 联机分析处理的基本分析功能	257
1. 上卷(roll-up)	258
2. 下钻(drill-down)	258
3. 切片(slice)	258
4. 切块(dice)	258
5. 转轴(pivot or rotate)	258
9.3.5 数据挖掘	258
• 特征描述	259
• 聚类分析	260
第10章 计算机网络	260
10.1 计算机网络的产生和发展	260
1. 1个目标	260
2. 2个支撑	260
3. 3个融合	260
4. 4个热点	260
• 多媒体	260
• 宽带网	260
• 移动通信	260
• 信息安全	261
10.2 网络体系结构及协议	261
10.2.1 网络体系结构及协议的定义	261
协议的关键成分是	261
• 语法(syntax)	261
• 语义(semantics)	261
• 定时(timing)	261
10.2.2 开放系统互连参考模型	261
1. 物理层	262
物理连接是开放系统互连的基础,它可分为:永久连接或动态的交换连接,全双工传输或半双工传输,同步传输或异步传输	262
物理层应向链路提供下列服务:数据电路标识,物理连接及其端点,物理服务数据单元,排序,故障状态通知和服务质量参数	262
作为物理层协议,它具有如下处理功能:激活和拆除物理连接,传输物理服务数据单元,完成物理层一些管理工作	262
2. 数据链路层	262
数据链路层向网络层提供的功能有:在物理层提供物理连接的基础上建立、维护和释放数据链路,数据链路服务单元的透明传送,数据传送的流量控制,数据链路服务提供者的差错指示,服务质量管理	262
链路层协议标准分成两类:第一类是面向字符的传输控制规程;第二类是面向位的链路层规程	262
3. 网络层	262

网络层通过网络层服务访问点，给传输层提供如下服务：网络地址服务，网络连接及端点标识，网络服务数据单元(NSDU)，服务质量，差错通知，用于保证接收顺序和控制的排序，流量控制，加速网络服务数据单元，复位(网络层差错处理方法，可将顺序计数清零)，释放网络连接，接收确认.....	262
4.传输层	262
根据残留差错率和可通告差错率，可把网络服务分为 3 类:A型网络服务, B型网络服务, C型网络服务	262
由于网络层以下对传送的协议数据单元大小都有限制，而高层则没有，因而传输层提供了分段/合段功能以满足两方面的要求.....	262
传输实体向会话实体提供的传输服务由 3 个阶段组成：传输连接建立阶段，数据传送阶段，传输连接释放阶段	262
根据传输实体所用不同类型的网络服务，可将传输协议分为 5 类.....	262
5. 会话层	263
• 会话层的主要功能是数据交换，它分为 3 个阶段：会话的建立、使用和拆除	263
• 会话层的另一功能是对话管理	263
• 会话层另一个与同步密切相关的关键特性是活动管理	263
6. 表示层	263
有 4 个主要的功能：给用户提供一种执行会话服务的方式，提供一种确定复杂数据结构的方法，管理当前请求数据结构组，在内部和外部形式间实现数据转换.....	263
7. 应用层	263
10.2.3 TCP/IP的分层	263
1. TCP/IP分层模型	263
(2) 传输层	263
(4) 网络接口层	263
2. TCP/IP分层工作原理	263
3. TCP/IP模型的分界线	263
4.复用和分解	264
10. 2. 4 IP协议.....	264
1. Internet体系结构.....	264
这种不可靠的、无连接传送机制称为Internet协议，简称IP协议。IP提供了 3 个重要的定义	264
2. IP数据报.....	264
10.2.5 用户数据报协议	265
1. UDP协议功能	265
2. UDP报文格式	265
3. UDP的协议分层与封装	265
4.UDP的复用、分解与端口	265
10.2.6 可靠的数据流传输	266
TCP/IP的可靠传输服务有以下 5 个特征.....	266
• 面向数据流	266
• 虚电路连接	266
• 有缓冲的传输	266
• 全双工连接	266
10.2.7 传输控制协议	266
1. TCP功能	266
2. TCP报文格式	267
10 . 3 局域网技术、	267
10.3.1 局域网定义和特性	267
10.3.2 局域网标准	268
(1) IEEE 802: LAN标准—概观和体系结构(1997)	268

·802.1B: LAN/MAN管理(1995)	268
·802.1D: MAC网桥(1998)	268
·802.1E: 系统负载协议(1994)	268
·802.1 F: 用于IEEE 802 管理信息的公共定义和过程(1993)	268
·802.1G: 远程MAC桥接(1998)	268
·802.1H: 在局域网中以太网 2.0 版MAC桥接(1997)	268
·802.1Q: 虚拟桥接局域网(1998)	268
(2)IEEE 802.2: 逻辑链路控制(包括简单无连接、连接方式、带确定无连接等服务)(1998)	268
(3)IEEE 802.3: 带冲突检测的载波监听多路访问(CSMA/CD)的访问方法和物理层规范(1998)	268
·802.3ac: VLAN的帧扩展(1998)	268
·802.3ab: 1000BASE-T物理层参数和规范(1999)	268
·802.3ad: 多重链接分段的聚合协议(aggregation of multiple link segments)(2000)	268
(4) IEEE 802.4: 逻辑标记总线访问方法和物理层规范(1990).....	268
(5)IEEE 802.5: 标记环访问方法和物理层规范(1997).....	268
·802.5r: 专用的标记环的运行(1997)	268
·802.5t: 100Mb/s高速标记环访问方法(2000)	268
(6) IEEE 802.6: 城域网(metropolitan area network,MAN)访问方法和物理层规范(1994).....	268
(7) IEEE 802.9: 在 MAC和物理层上综合语音和数据(integrated voice and data,IVD)局域网技术(1996).....	268
(8) IEEE 802.10: 可互操作的局域网安全标准(standard for interoperable LAN security, SILS)(1998)	268
(9) IEEE 802.11: 无线局域网的MAC协议和物理层规范(1999).....	268
(10) IEEE 802.12: 需求优先(demand priority)协议	268
(11)IEEE 802.14: 利用CATV宽带通信的标准(1998)	268
(12) IEEE 802.15: 无线私人网(wireless personal area network, WPAN)	268
(13) IEEE 802.16: 宽带无线访问标准(broadband wireless access standards)	268
·802.16.1: 固定宽带无线访问的无线界面	268
·802.16.2: 宽带无线访问系统的共存	268
(14)ISO 9314: 光纤分布式数据接口(1989).....	268
10.3.3. 快速以太网	269
1.快速以太网类型	269
2.快速以太网产品	269
10.3.4 千兆位以太网	269
1. 千兆位以太网规程和标准	269
2.交换式LAN结构的千兆位以太网	270
10.4 广域网技术	270
10.4.1 点到点通信	270
10.4.2 分组交换网	271
1.分组交换网原理	271
2. X.25 分层协议	271
10.4.3 帧中继网	272
1.帧中继网产生背景	272
2.帧中继网与X.25 网比较.....	272
帧中继与X.25 分组交换的最主要区别有以下3点.....	272
在高速信道上提供帧中继服务, 对下面一些应用很有用.....	272
10.4.4 ATM网	273
10.4.4.1 ATM协议参考模型	273
1.物理层	273

2. ATM层	273
3. ATM适配层(AAL)	273
4.信元类型	273
• 有效信元	273
• 无效信元(物理层).....	273
• 指定的信元(ATM层)	273
• 非指定的信元(ATM层)	273
10. 4. 4. 2 ATM层	273
1. 信元结构	273
2.信元头的结构	273
3. ATM层原语	274
10. 4. 4. 3 ATM适配层	274
1. ATM适配层目的	274
在ATM信元流中会发生下列情况	274
2. AAL服务分类	274
3. AAL的子层	274
4. AAL类型	274
10.4.5 移动通信	274
1.移动通信网	274
2.全球移动通信系统	274
3.无线软件应用协议	274
4. 个人通信业务/个人通信网	275
10.5 网络管理与网络安全、	275
10.5.1 网络管理功能	275
1.配置管理	275
• 配置信息的自动获取	275
• 自动配置、自动备份及相关技术	275
• 配置一致性检查	275
• 用户操作记录功能	275
2.性能管理	275
• 性能监控	275
• 阈值控制	275
• 性能分析	276
• 实时性能监控	276
• 网络对象性能查询	276
3.故障管理	276
• 故障监测	276
• 故障报警	276
• 故障信息管理	276
• 排错支持工具	276
• 检索/分析故障信息	276
4. 安全管理	276
网络管理本身的安全由以下机制来保证.....	276
网络对象的安全管理有以下功能.....	276
• 网络资源的访问控制	276
• 告警事件分析	276
• 主机系统的安全漏洞检测.....	276
5.计费管理	276
• 计费数据采集.....	276
• 数据管理与数据维护	277

• 计费政策制定	277
• 政策比较与决策支持	277
• 数据分析与费用计算	277
• 数据查询	277
10.5.2 网络管理协议	277
10.5.2.1 SNMP	277
10.5.2.2 CMIS/CMIP	277
10.5.2.3 CMOT	277
10.5.2.4 LMMP	278
10.5.2.5 简单网络管理协议	278
1. SNMP管理控制框架	278
管理信息报文中包括以下两部分内容	278
访问权限检查涉及到以下因素	278
2. SNMP协议	279
SNMP中设计了4种基本协议交互过程。	279
• 管理进程从管理代理处提取管理信息	279
• 管理进程在管理代理的可见范围内遍历一部分管理对象实例	279
• 管理进程在管理代理中存储信息，即对管理代理的管理信息库MIB进行写操作（包括设置工作参数）	279
• 管理代理主动向管理进程报告事件	279
10.5.3 信息安全术语	279
1. 密码学	279
2. 鉴别	280
3. Kerberos鉴别	280
4. 公钥基础设施	280
5. 数字签名	281
6. 访问控制	281
10.5.4 网络安全技术	281
10.5.4.1 网络安全层次模型	281
10.5.4.2 防火墙技术	282
1. 包过滤防火墙	282
2. 应用层网关	282
3. 应用代理服务器	282
4. 状态检测防火墙	283
10.5.4.3 IP层安全性	283
10.5.4.4 传输层安全性	283
10.5.4.5 应用层安全性	284
10.5.4.6 WWW应用安全技术	284
10.6 Internet与Intranet	285
10.6.1 Internet路由结构	285
随着Internet的迅速发展，核心路由器体系的结构在实现时出现了不少问题	285
10.6.2 Internet地址	285
10.6.3 Internet域名系统	286
1. 域名系统原理	286
2. 域名的分级	287
3. Internet域名	287
10.6.4 Internet地址空间的扩展	287
但是，IPv6对协议细节作了许多修改。IPv6的修改可分成以下5大类：	288
• 更大的地址空间	288
• 灵活的报头格式	288

·增强的选项	288
·支持资源分配	288
·支持协议扩展	288
10. 6. 5 Intranet的定义和应用	288
10. 6. 5. 1 Intranet的定义	288
从这个定义出发, 可概括Intranet的若干要点如下	288
10. 6.5.2 Intranet的应用	288
1. 企业内部主页	288
2.通信处理	288
3. 支持处理	289
4.产品开发处理	289
5. 运行处理	289
6.市场和销售处理	289
7.客户支持	289
10.7 信息服务与网络应用	289
10.7.1 万维网	290
10.7.1.1 浏览器	290
1 . lynx	290
2. midasWWW	290
3 . Mosaic.....	290
4. Netscape.....	291
5 . Microsoft Internet Explorer	291
目前较为流行的WWW浏览器是: Netscape, IE, lynx.....	291
10. 7. 1. 2 Web服务器	291
1. NCSA Web服务器.....	291
2. CERN httpd	291
3. Plexus httpd	291
10.7.2 动态Web文档与CGI技术	291
1. Web文档的 3 种基本形式	291
·静态文档	291
·动态文档	291
·活动文档	291
2.动态文档的实现	292
3.通用网关接口	292
10. 7. 3 活动Web文档和Java技术	292
1.活动文档技术	292
2. Java技术	292
10.7.4 网络化经济的新模式	293
10.7.5 电子商务	294
可以简明地给出电子商务的几个要素或特征, 即 2P+ 3C.....	294
1.电子商务通用框架	294
2.电子商务的分类	294
根据贸易商务活动的伙伴或贸易对象分类如下	294
·企业或商户与个人消费者之间的电子商务, 即B to C(Business to Consumer,B2C)	294
·企业与企业之间的电子商务, 即B to B(Business to Business, B2B).....	294
·消费者与消费者之间的电子商务, 即C to C或P to P(Consumer to Consumer, C2C或Person to Person, P2P).....	294
·企业与政府之间的电子商务, 即B to G(Business to Government, B2G).....	294
·电子商店或虚拟商店(Virtual Store).....	294

• 电子商场(Virtual Electronic Merchant)	294
• 电子商厦或电子街道(Virtual Electronic Mall)	294
• 电子商城(Virtual Electronic Commerce City)	294
按执行的电子商务的业务性质分类如下	294
• 网络电子商情业务	294
• 网上交易业务	294
• 网络电子银行业务	294
按照网络环境分类如下	294
• 采用EDI专用网络	294
• 在Internet上开展	295
• 在Extranet/Intranet环境中进行	295
10.8 网络工程	295
10.8.1 网络规划	295
1.需求分析	295
• 地理布局	295
• 用户设备类型	295
• 网络服务	295
• 通信类型和通信量	295
• 容量和性能	295
• 网络现状	295
2. 系统可行性分析	295
• 传输	295
• 用户接口	295
• 服务器	295
• 网络管理能力	295
• 系统可行性的另一个重要影响因素是造价	296
10. 8. 2 网络设计	296
10. 8. 2. 1 网络设计原则	296
• 成熟性	296
• 开放原则	296
• 安全可靠原则	296
• 先进原则	296
• 完整性原则	296
• 可扩展性	296
10.8.2.2 网络体系结构	296
10.8.2.3 子网规划	296
划分子网的方式有多种, 经常使用的有	296
• 通过物理连接来实现	296
• 虚拟局域网(virtual LAN, VLAN)	296
10.8.2.4 逻辑网络设计	296
1.设计网络拓扑结构	296
现在常用的层次网络结构是 3 层结构	296
• 核心层	296
• 分布层	296
2.网络地址分配和命名策略	297
• 在分配地址之前设计结构化寻址模型	297
• 为寻址模型的扩充预留空间	297
• 以分层方式分配地址块, 以改进可伸缩性和可用性	297
• 为了避免组或个人移动所带来的问题, 应根据物理网络而不是组成成员分配地址块	297

• 分配网络地址时使用有意义的编号.....	297
• 为了最大限度满足灵活性, 而又使配置最少, 可以在用户端使用动态寻址.....	297
• 为了使安全性和适应性得到最大满足, 在IP环境中使用网络地址翻译(network address translation, NAT)技术, 在单位内部使用私用地址.....	297
3.选择桥接、交换和路由选择协议.....	297
4.设计网络安全和管理策略.....	297
安全性设计一般包括.....	297
网络管理设计包括.....	297
10. 8. 2. 5 网络技术和设备选型.....	297
1. LAN布线设计.....	297
2. LAN选型.....	298
(1) LAN技术选型.....	298
(2) LAN网络互连设备选型.....	298
选择网络互连设备的条件一般包括下列内容.....	298
对于网桥, 可以增加下列条件.....	298
对于交换机, 可以增加下列条件.....	298
对路由器(和有路由模块的交换机)可增加下列条件.....	298
3.远程访问设计.....	299
(1)远程访问技术选型。.....	299
(2)远程访问设备选型.....	299
4.广域网设计.....	299
(1)广域网带宽系统.....	299
• DS系列.....	299
• E系列.....	299
• 同步数字线路(SPH).....	299
(2)广域网接入技术.....	299
• 专线.....	299
• 同步光纤网络(SONET).....	299
• 帧中继.....	300
• ATM广域网.....	300
(3)广域网设备及服务提供商的选择.....	300
①选择广域网路由器.....	300
②选择广域网交换机.....	300
③选择广域网服务提供商.....	300
以下标准往往比费用更重要.....	300
在选择服务商时, 还应尽可能了解服务商所提供网络的以下特性.....	300
10.8.3 网络实施.....	300
1.工程实施计划.....	301
2.网络设备到货验收.....	301
3.设备安装.....	301
4.系统测试.....	301
5.系统试运行.....	301
6.系统切换.....	301
7.人员培训.....	301
10.8.4 网络测试.....	301
1.网络设备测试.....	301
2.网络系统和应用测试.....	301
第 11 章计算机系统与配置.....	302
11.1 计算机体系结构.....	303
11. 1. 1 计算机指令系统的发展.....	303

1.复杂指令系统计算机(CISC).....	303
2.精简指令系统计算机(RISC).....	303
大部分RISC机具有以下特点	303
同CISC机比较RISC机有以下优点.....	304
11.1.2 提高计算机系统运算速度的方法.....	304
对于单机系统(系统内含一个CPU)可采用下述方法.....	304
11.1.3 流水线技术	304
1.指令的重叠执行	304
2.流水线中的相关问题	304
3.程序转移对流水线的影响	305
4.其他	305
11. 1. 4 指令预取和无序执行.....	306
11. 1.5 存储系统的发展.....	306
11.1. 5.1 多层次存储系统.....	306
11. 1. 5. 2 cache存储器	307
1. cache工作原理	307
2.指令cache和数据cache	307
3.多层次cache	307
11. 1. 5. 3 虚拟存储器.....	308
11. 1. 5. 4 访问存储器(取指或存取数据)的工作过程	308
1.虚地址转换成实地址	308
2.根据实地址访问主存	309
11. 1. 5. 5 主存储器.....	309
1. DRAM的研究与发展	309
• 增强型DRAM(EDRAM)	309
• cache DRAM(CDRAM)	309
• EDO DRAM	309
• 同步DRAM(SDRAM).....	309
• Rambus DRAM(RDRAM)	309
2.交错存储器	310
11. 1.5. 6 相联存储器.....	310
11.1.6 系统总线和外设接口.....	310
11. 1. 7 超级标量处理机、超级流水线处理机和超长指令字处理机.....	311
1. 超级标量(Superscalar)处理机	311
2.超级流水线(super pipeline)处理机	312
3.超长指令字(VLIW)处理机	312
11.2 并行处理计算机	312
11.2.1 向量处理机	312
11.2.2 多处理机系统	312
1.多处理机系统结构	312
2.大规模并行处理机MPP和对称处理机SMP.....	313
3.互联网络ICN(inter connection network).....	313
常见的互联网结构如下	313
• 总线结构	313
• 交叉开关	313
• 多级互联网	313
4. cache一致性	313
5.非均匀存储存取NUMA(non uniform memory access)	313
下面以SGI公司的Origin服务器为例来介绍多处理机系统的结构.....	313
(1)SGI Origin的基本结构.....	314

(2) Origin的拓扑结构	314
11. 3 计算机系统的可靠性、可用性、可维护性技术和容错技术	314
11. 3. 1 计算机系统的可靠性	314
11. 3. 1. 1 计算机系统的可靠性指标	314
提高系统可靠性一般有两类技术方法，即避错法和容错法	314
硬件避错技术的作用是减少系统失效的可能性，主要包括	314
11.3. 1. 2 采用附加的数据校验码来提高计算机系统的可靠性	315
1.奇偶校验码	315
2.纠错码ECC	315
3.循环冗余检验码CRC	316
11.3.2 计算机系统的可用性	316
11.3.3 计算机系统的可维护性	316
在实际工作中，一般将维修分成 3 级	316
11.3.4 容错技术	317
冗余一般可分为下列几种类型	317
• 硬件冗余	317
• 软件冗余	317
• 信息冗余	317
• 时间冗余	317
故障可归结为永久性故障、间隙性故障和瞬时性故障 3 类	317
最常用的硬件冗余是硬件的重复设置。硬件冗余一般可分为 3 种类型：静态冗余、动态冗余和混合冗余	317
一个采用冗余技术的计算机系统在处理运行中产生故障时，通常采用以下 10 个步骤(或其中的一部分)	318
11.4 计算机性能评测	318
11. 4.1 计算机性能评测概述	318
11.4.1.1 计算机性能评测的度量项目	318
• 性能指标	318
• 可靠性、可用性和可维护性	319
• 环境适应性	319
• 兼容性	319
• 开放性	319
• 可扩充性	319
• 安全性	319
• 保密性	319
• 性能价格比	319
11.4. 1.2 评测方法	319
• 测量法	319
• 模型法	319
11.4.2 开放系统	319
开放系统应能做到	319
11. 4.3 系统兼容性	319
• 硬件设备或部件兼容	319
• 机器语言程序兼容	319
• 汇编语言程序兼容	320
• 高级语言程序兼容	320
• 系统软件兼容	320
• 软件系统兼容	320
11. 4.4 性能评估	320
1 . MIPS和MFLOPS	320

2. 吉普森混合法	320
3. 数据处理速率PDR(processing data rate)	321
4. 综合理论性能CTP(composite theoretical performance)	321
11.4.5 基准测试程序	322
1. Whetstone基准程序	322
2. Dhrystone基准程序	322
3. UNPACK基准程序	322
4. SPEC(system performance evaluation cooperative)基准程序	322
SPEC95 由两组基准测试程序组成	322
合成指标计算如下	323
·SPECint 95	323
·SPECint-base 95	323
·SPECfp 95	323
·SPECfp-base 95	323
5. TPC(transaction process performance council)基准程序	324
6. 对计算机性能评测的评估	325
第 12 章 信息安全技术	325
12. 1 访问控制机制和方法学	325
12.1.1 单点登录技术	326
一个理想的SSO产品具备以下的特征和功能	326
· 常规特征	326
· 终端用户管理灵活性	326
· 应用管理灵活性	326
· 移动用户管理	326
· 加密和认证	326
· 访问控制	326
· 可靠性和性能	326
12.1.2 集中式认证服务	326
1. AAA服务	327
AAA服务的主要特征包括	327
分布式安全模型将认证过程和通信过程分开	327
AAA服务器能够支持多种认证机制	327
从一个管理者的角度来看, AAA服务器具有下列优点	327
2 . RADIUS(remote authentication dial-in user service)	328
RADIUS是目前最常用的AAA服务, 其普遍性应归功于RADIUS的源代码的公开性	328
RADIUS有 8 种标准的事务类型	328
在RADIUS协议中, 授权不是一个独立的功能而是认证响应中的一部分	328
RADIUS是为远程访问认证所设计, 而不适用于主机以及应用认证	328
3. TACACS	328
TACACS认证有 3 种类型的包	328
TACACS的授权功能包含请求和响应AV对, 其主要用于以下目的	328
TACACS的记账功能使用类似于授权功能的格式	328
虽然TACACS是一个多用途而且稳健的协议, 很少的服务器使用它, 在NAS中的使用更加少	328
4. DIAMETER	328
DIAMETER是一个高度扩展的AAA框架, 能够支持多种认证、授权或记账方案以及连接类型	328
DIAMETER建立在RADIUS协议之上, 但是对其进行了补充	329
DIAMETER的认证是由扩展协议来管理的	329

DIAMETER的授权可以同认证请求绑定在一起,也可以独立进行	329
DIAMETER增加了事件检测、定期报告、实时记录传输的功能,因此记账功能比RADIUS和TACACS都有了明显的增强	329
对高强度安全性的支持是DIAMETER基本协议的一个标准部分	329
12. 2 通信和网络安全	329
根据无线互联网用户的反映,无线互联网主要具有以下局限性	330
下面主要对无线设备的相关安全特征进行讨论	330
1.认证性	330
2.机密性	330
3.恶意代码以及病毒	331
12.3 安全管理实施	331
12. 3.1 安全策略以及标准	331
1.安全模型	331
Bell-LaPadula (BLP)模型是基于机密性的访问模型	331
Biba模型是基于完整性的访问模型的最初尝试	332
Clark-Wilson模型也是基于完整性的访问模型	332
2.安全策略的必要性	332
3.安全策略的制定过程	332
(1)初始与评估阶段	332
(3)核准阶段	333
(4)发布阶段	333
(5)执行阶段	333
(6)维护阶段	333
12. 3. 2 风险管理与分析	333
而风险常常可以描述为一个数学公式: 风险=威胁×脆弱性×资产价值	333
· 风险	333
· 威胁	333
· 脆弱性	333
· 资产	333
1. 风险分析	333
风险分析主要包含三个重要因素: 知识、观察力以及敏锐性	333
2.风险管理	334
12. 4 应用和系统开发安全	334
12.4. 1 Web应用安全	334
Web应用的攻击弱点主要在于以下几个方面	335
· 已知的脆弱性和错误配置	335
· 隐藏区域	335
· 后门以及调试选项	335
· 参数篡改	335
· Cookie攻击	335
· 缓冲溢出	335
· 直接访问浏览	335
但现有的一些方案能够尽量减小这种风险	335
1.预防	335
2. 一些可用的技术工具	336
12.4. 2 XML的安全性	336
但是随着网络以及Web应用的快速发展,HTML不再能够满足人们的需要,其局限性主要体现在	336
1. XML的优点	336
一个基本的XMI. 文件主要包括	337

为了增强XML文件结构化要求，必须利用XML的辅助技术—文件类型定义(Document Type Definition, DTD)	337
XML主要有三个要素：模式(Schema可扩展样式语言(eXtensible Stylesheet Language, XSL) ffl XLL(eXtensible Link Language, 可扩展链接语言)	337
2. XML的安全问题	337
而XML的签名需求是由XML密钥管理规范(XML Key Management Specification., XKMS)	337
12.5 密码术与安全观念的发展	338
值得指出的是，当今密码体制是建立在三个基本假设的基础上的	338
·随机性假设	338
·计算假设	338
·物理假设	338
数字世界的知识安全问题研究，重大而迫切，知识安全研究内容至少包括以下 5 个方面	339
·知识的表达	339
12.6 安全体系结构和模型	339
12.6.1 UNIX系统的安全性	339
操作系统通常可以提供以下的安全服务	339
·身份识别及认证	339
·访问控制	339
·可用性和完整性	339
·审计功能	339
·用户可用的安全设施	339
下面对于在UNIX系列操作系统中上述服务的实施情况进行详细的讨论	339
1.身份识别及认证	339
2. 访问控制	340
3.可用性和完整性	340
4. 审计日志	340
通常UNIX系统在下列日志文件中记录并存放与安全相关的事件	340
5.用户可用的安全役施 ‘	341
12.6 . 2 数据库的完整性	341
下面对于数据库管理以及维护数据库完整性的方法进行简单的讨论	341
·整体化	341
·惟一的所有者进程	341
·冗余	341
·动态错误检测与纠正	341
·复制	341
·镜像	341
·备份	341
·重构	342
·分割	342
·隔离与独立	342
·封装	342
·隐藏	342
·原子刷新(Atomic Update)	342
·锁定	342
·访问控制	342
·特权控制	342
·复原	342
至少下列因素尹于维护数据库的完整性是必要的	342
12.7 计算机操作安全	342

12.7.1 安全威胁	342
1.对威胁的评估	343
• 查阅	343
• 实验	343
• 调查	343
2.陷阱的好处以及陷阱的特性	343
显然，一个好的陷阱应该是能够捕获到猎物的陷阱，而且一个好的陷阱还应该具备以下的特性	343
• 良好的隐蔽性	343
• 有吸引力的诱饵	343
• 准确的触发机关	343
• 强有力的圈套	343
12.7.2 入侵检测	344
1.入侵检测的历史简单回顾	344
2.入侵检测概述	344
3.检测技术	344
4.存在的问题	345
5.系统的有效性	345
6.性能	345
7.基于整个网络范围的分析	345
12. 8 业务持续和灾难恢复规划	346
12.8.1 业务持续性规划	346
对于恢复规划人员来说，需要进行下列工作.....	346
1.持续性规划的方法	347
• 灾难恢复规划(DRP)	347
• 业务恢复规划(business resumption planning, BRP)	347
• 危机管理规划(crisis management planning, CMP)	347
2.持续性规划流程的衡量手段	347
这些手段通常包括	347
而现在焦点应该放在测量持续性规划流程对企业的整体目标所作的贡献上，这样做有以下好处	347
12. 8. 2 灾难恢复规划	347
灾难恢复规划的一个主要用途是尽量减少灾难发生的可能性.....	348
一旦灾难发生，业务组的首要任务是尽快恢复关键系统并尽可能小地减少对关键系统的影响，同时灾难恢复规划开始实施	348
信息系统的危机处理及灾难恢复主要可以分成下列几种.....	348
12.9 物理安全	348
1.分层的防御体制	349
2.多方面防御机制	349
物理安全的实施通常包括以下几个方面.....	349
• 确认	349
• 标注	349
• 安全	349
• 跟踪	349
• 技能.....	349
3.物理安全存在的缺陷	350
当实施物理安全时，必须认识到一些普遍存在的局限性以及缺点.....	350
• 社会工程学	350
• 密码的泄漏	350
• 尾随	350

• 环境因素	350
• 装置可靠性	350
• 信任度	350
• 用户接受度	350

第 1 章概述

1. 1 信息系统与信息系统工程

信息系统一般泛指收集、存储、处理和传播各种信息的具有完整功能的集合体。人们常说的信息系统大多指支持各部门和机构管理和决策的信息系统。随着社会的进步和技术的发展,信息系统的内容和形式都在不断发生着巨大的变化,当前信息系统重要的特征是计算机和互联网技术的介入。

现代信息系统是以计算机为信息处理工具,以网络为信息传输手段的;它最大限度地屏蔽了时间和空间限制,使人们能以最快捷的方式获取所需信息并加以利用。要想了解现代信息系统的现状,就应追溯近 50 年来计算机信息系统的发展,这有利于正确认识和评价现代信息系统在社会整体发展中所处的地位和所产生的作用。实际上,社会的信息化正是在现代信息系统的逐步建设中,亦即信息系统工程的实施中逐渐形成的,它有一个从局部到整体、从初级到高级、从简单到复杂的发展过程,是社会和技术发展的一种必然。

计算机应用于企业是从最基础的数据处理开始的。早期的计算机程序设计人员的重要贡献是,将计算机从单纯的从事科学计算而拓展到能进行数据处理,从而开辟了一个计算机应用最为广阔的应用领域。最早的计算机在数据处理中的应用,仅着眼于减轻人们在计算方面的劳动强度,如用于计算工资、统计账目、管理雇员等,属于一类所谓电子数据处理,即 EDP 业务,对企业单项业务进行处理,它较少涉及管理的内容。随着企业业务需求的增长和技术条件的发展,人们逐步将计算机应用于企业局部业务的管理,如财会管理、销售管理、物资管理、生产管理等,即计算机应用发展到对企业的局部事务的管理,形成了所谓事务处理系统,即 TPS 但它并未形成对企业全局的、整体业务的管理。形成对企业全局性的、整体性的计算机应用是后来基于单项应用基础上发展并形成的管理信息系统(MIS)的任务。管理信息系统强调以企业管理系统为背景,以基层业务系统为基础,强调企业各业务系统间的信息联系,以完成企业总体任务为目标,它能提供企业各级领导从事管理需要的信息,但其收集信息的范围还更多地侧重于企业的内部。当前,计算机信息系统已经从管理信息系统发展成更强调支持企业高层领导决策的决策支持系统,即 DSS 阶段。互联网技术的发展和运用,在很大程度上拓展和提升了信息系统的功能和作用,其最大的特点是通过互联网将众多的孤立的信息系统(即所谓信息孤岛)加以联系起来,形成在更大程度上实现信息共享的、大范围的基于网络互联的信息系统。互联网技术应用于企业内部信息系统,可促进企业内综合 MIS、DSS 功能,并以办公自动化技术为支撑的办公信息系统的实施。企业信息系统的目标为:借助于自动化和互联网技术,综合企业的经营、管理、决策和服务于一体,以求达到企业和系统的效率、效能和效益的统一,使计算机和互联网技术在企业管理决策和服务中能发挥更显著的作用。

目前,EDP 已成为企业实现信息管理的基础性工作,对提高企业的工作效率和质量有明显的作用,是众多高层系统的基础。MIS 是计算机在企业管理领域中应用的重要组成,它对提高企业管理的总体效率和质量有明显的作用。而 DSS 在企业对重大问题的决策上将产生积极的作用,能最大限度发挥企业的效能,为企业带来总体效益。计算机技术、网络技术和管理科学的发展对企业的信息化过程影响深远。如当前流行的 MRP (Manufacturing Resource Planning, 制造企业资源计划) 技术 ERP (Enterprise Resource Planning, 企业资源计划) 技术和 CRM (Customer Relations Management, 客户关系管理) 技术等都在企业中有不同程度的实践和应用。不论未来会出现什么样的新技术、新产品,它们都将是包含着对信息的基本处理,对企业的科学管理,对重大问题的有效决策,都还是沿着处理、管理和决策的划分在演变。

回顾历史,企业计算机信息系统的建设,其发展轨迹应该是一个从 EDP 到 TPS,再到 MIS, DSS 和 OIS 的发展过程,同时,在发展过程中,密切结合了业务领域,实现了业务内容和信息技术的融合,如 ERP 就是这样的例子。在信息化的过程中企业高层领导和技术人员的职责应该是:如何根据企业的实际业务需求,全面考虑它的战略目标和约束条件,以正确的策略和方法来制定一个适合企业需要的业务和技术发展规划,再从易到难,从部分到整体,用逐渐拓展的方式来分步实现这一规划,在实施中积累经验,优化系统,不断追求具有实际效益、实用和经济的系统。

信息系统工程是以系统的方法来实现信息系统建设的过程。实现信息系统工程的途径有很多,美国学者 James Martin 所命名的信息工程是其中很有影响的一种。它提供了信息系统工程实现的整体方法论,集方法、工具、环境和经验于一体,为信息系统的建设提供了全面的解决方案。信息工程方法从企业开发

信息系统的实际需求出发,提供了结构化的开发方法,并强调系统开发必须从数据规划开始,从而形成以数据为中心的系统开发方法论。信息工程是在方法论指导下,在与方法论相配合的开发工具的支持下去实施系统开发的,它强调了自动化的信息系统必须用自动化的手段来实现,并在实现中有基于信息库的开发环境的支持。信息工程方法不仅在方法论以及技术手段上支持了信息系统的开发,而且也吸收了有效的系统开发经验,从而极大地提高了系统开发的成功率。随着社会信息化的进程加快,社会各行各业都基于自身的需求在加快本行业、本部门、本领域的信息化进程。当今,电子政务、电子商务等领域都投入大量的资金和技术来建立相应的信息系统,因此提高系统建设的成功率就是一件十分迫切的问题,这也是信息工程要解决的问题。

1.2 建立信息系统所涉及的问题

计算机信息系统的建立是企业的一项重大的社会技术工程,James Martin 将建立企业的信息系统称为信息工程。建立企业信息系统是社会发展、企业发展的需要,好的企业信息系统将极大地加强企业的市场竞争力,但要建立一个好的企业信息系统也会受到多方面条件的约束。无疑,研究这些条件将有助于系统的建设。

1.2.1 系统建设前企业应具备的条件

企业信息系统的建设目标、功能需求和规模大小,须服从于企业的环境和需求,而不单纯取决于企业领导的主观意愿;系统建设应该有其先决条件,如果条件不具备,则需先做些基础性的准备工作,否则系统难以成功。

1.企业高层领导应具有对企业信息系统建设规律性的认识

社会信息化是社会和技术发展的方向,它遵循社会和技术发展的客观规律。现代计算机信息系统是一类涉及业务面广、技术难度大的系统工程。如果企业高层领导对信息化工作的规律尚不甚了解,由于受到某种潮流的影响萌发出在企业建立系统的意愿,则会产生消极的影响。因此,企业高层决策人应该对企业是否建立、在什么时候、建立什么样规模的信息系统有比较正确的决策,而正确的决策来源于高层决策人对信息化建设规律性的了解。这种规律性认识来源于学习、调查和咨询。高层决策人应提出恰当的系统目标,给出准确的业务需求,制定合适的开发策略,提供必要的资金保证,配备精干的管理人员,以保证从系统的功能需求分析,到设计,到实施等环节能顺利地实现。否则效果会相反,信息系统失败的例子不论在国际上或国内都是常见的,当引以为戒。

2.企业必须认真分析建立计算机信息系统的实际需求

信息化的舆论将众多的企业推向信息化的浪潮,而企业对信息化的实际需求才是建立企业计算机系统的原动力。这种原动力来自企业内部也来自企业的外部,但归根结蒂是来自企业的内部。任何人的主观意愿或人为因素都不能作为建立系统的依据,都不足以使系统的开发顺利完成并产生实效。因此从某种意义上讲,在系统建设的可行性分析中,系统建设的必要性分析比可能性分析更为基本,更具有现实意义。

3.管理的科学化是企业信息系统建立的基础和保证

科学的管理是企业信息化的基础,没有科学管理的基础,企业无法建成有效的计算机管理系统。对于战略目标模糊、管理理念落后、规章制度不健全、基础数据残缺的企业,首先必须完善管理,使其科学化和规范化,为实现计算机管理奠定良好的基础。

计算机作为信息加工的工具,对企业业务活动所产生的数据进行加工的过程,都是在管理人员的“授意”下进行的。这种“授意”的科学性和正确性都来自管理人员的管理理念和水平,即其管理科学化和规范化的程度在很大程度上约束着信息系统的作用。

4.企业文化和管理人员的组织结构应能满足系统建设的需要

计算机信息系统建设的成败,不但取决于系统的开发人员的素质,更大程度上取决于企业人员的组织结构和文化素养。良好的组织结构和企业文化能保证人们更具有科学的工作态度、良好的敬业精神、善于合作和勤于务实。知识结构和文化素养决定了人对新事物、新技术的敏感和追求。合作精神能促进业主和开发者之间在系统调查、设计和实施过程中的相互配合和相互支持。工作态度直接影响系统的质量,它对资料的准确性、完整性、对业务过程分析和定义的准确性都将产生较大的影响。务实精神将促进系统实际效益的产生,杜绝追求华而不实的“效果”,负责任的企业领导人所追求的是企业的实际效益,包括社会效益,但更重要的是经济效益,因为无实际效益的系统在任何情况下都是无法持久的。

5.规范和齐全的数据是建立企业计算机信息系统的必要条件

数据是系统加工的对象,正像原料是企业生产时所必备的一样,数据是信息加工的依据和来源。数据的完整、齐全和准确直接决定着信息的质量。“进来是一堆垃圾,出去还是一堆垃圾”是数据处理业务中对数据质量低劣的后果描述的名言。如果在建立计算机信息系统之前,发现建立系统所必需的数据有缺损、

不规范，那将无法对企业实现计算机管理。人们将对数据收集的规范化和制度化称为数据工程，并将其作为信息工程的基础性工程，是信息工程的有机组成成分。

6.企业有必要的开发期和维护期的资金保证

企业信息系统的建设是一项投资大、工期长的工程。当前在国内，设备的投资是主要的资金消耗，它包括主要的机器设备，网络设备，辅助设备和应用软件的开发。除了开发期，的资金投入，还必须考虑运行期的资金投入，即所谓维护经费。计算机信息系统是一类高科技的产品，包含较复杂的科技内容，且具有很高的设备更新速度、很短的技术更新周期。因此，系统投入运行后要发挥其作用，还必须有一支维护队伍，并需要不断地对系统全面的维护，包括对硬设备的维护和对软件(特别是对应用软件)的维护。维护费用是一项长期的支出，主要应用于对维护人员的投入。必要的资金保证也是企业计算机信息系统开发的基本条件。

如果将上述条件作为建设企业信息系统前应满足的条件，无疑将会极大增加系统的成功率，并取得较好的效果。

1.2.2 系统建设中企业应具备的条件

1.企业高层领导介入系统建设

计算机信息系统建设的经验表明，企业高层领导对信息系统建设介入的程度，对系统的成功与否有直接的影响和决定作用。当然，这一结论是建立在企业领导对企业信息化有基本了解的前提下的。

企业领导介入系统建设的必要性在于：

- 企业高层领导最了解本企业的战略目标和企业最本质的信息需求。
- 企业高层领导介入系统，能有效地在人力、财力和物力上组织系统的开发，并有效地解决一切在开发中可能出现的各种问题。一般认为，只有企业的最高领导者才有权力在全企业发布进行信息系统建设的宣言，并落实组织机构、动员全企业员工支持系统建设。IBM 公司提供的企业系统规划方法(即 BSP 方法)中特别指出：“BSP 的经验说明，除非得到了最高领导者和某些最高管理部门参与研究的承诺，否则不要贸然开始 BSP 的研究，因为研究必须反映最高领导者关于企业的观点，而研究成果取决于管理部门能否向研究组提供企业的现状、管理部门对企业的理解和对信息的需求。”

经验证明，企业领导仅仅停留在对信息系统开发的一般支持是不够的，必须实际介入。当然，企业领导者过多干预一些纯技术上的问题也是不可取的。取得领导者对工程支持的有效做法是，加强和组织对领导者进行多种形式的宣传。

2.吸收相关企业信息系统建设中的经验和教训

信息系统建设存在较高的失败率，因此借鉴成功的经验，吸取失败的教训是争取成功的好办法。如果能选择业务性质和规模相近的企业信息系统作为本企业信息系统的开发原型，有可能减少系统开发中存在的弯路‘一般来讲，经营业务相近的企业往往就是竞争对手，但在现今的中国这种借鉴的方式还是有可能的，特别是在政府机关、学校等部门是完全可能的。可以通过参观、座谈、类比和分析来获得其需要的信息或知识，乃至某些可支持系统开发的材料。好的建议有可能会降低系统成本、减少投资、缩短开发周期。

3.选择适合本企业实际情况的开发方式

企业信息系统的开发可以有多种方式的选择。传统的方式有：

• **委托开发方式**，即将企业信息系统的开发任务委托给某一个或几个具有系统开发能力的组织来承担，现今大多是通过招标的方式来确定开发单位。这些单位一般包括如系统工程公司、系统集成公司、计算机公司和软件公司等。也有某些具有系统开发能力的科学研究单位和高等学校参与这类竞争。

• **合作开发方式**，当企业有一定开发能力时，企业的技术力量与某一系统开发单位联合起来，或分工合作或混合组织对系统进行开发，这种方式的前提是企业有一支从事开发工作的队伍，如有的企业有自己独立的计算中心或信息中心。

• **自行开发方式**，依靠本企业的技术力量来开发本企业的信息系统，一般这种方式都会在一些有经验的专家的指导下来进行。

经验证明，不论采用哪种开发方式，聘请有经验的专家在项目开始或进行中作必要的咨询是十分有益的。专家可以从更高的视点、更广的视野来审视企业系统开发中的多方面的问题；专家们可以从多方面分阶段来审视，如系统的目标是否合适，系统规模是否适当，对开发方法和开发工具的选择以及开发阶段的划分等提出意见，并发现问题和给出改进建议。

哪一种是最优的开发方式可能难以给出绝对的标准，应服从于本企业的具体情况和条件。但随着社会分工的细化和技术条件的发展，在信息技术发达的国家又出现了另一种企业信息系统的开发和维护的新形式，即所谓“外包”方式。企业将所有与信息系统建设和维护的工作完全承包给专业的信息系统公司，企

业只负责提出自己的信息需求，而一切的系统开发、运行和维护工作都由专业的信息服务公司来承担，企业付给承包公司必要的费用。这种方式有其优越性，当然也有相当的技术和管理难度，它要求企业有相当程度的规范管理，同时也要求承包公司有承包相应业务的经验和必要的物质条件。虽然目前在国内实施系统外包可能尚有困难，但它很可能成为企业信息系统建设的发展方向。

4. 建立系统开发组织机构和选择成员

当系统开发方式基本确定以后，必须建立与开发方式相适应的组织机构，并选拔和安排必要的人选，以负责整个系统的开发工作。

不同的开发方式可以有不同的组织机构，但不论采用哪一种方式，对人员都会有一些基本的要求，如对企业的基本业务应当有所了解，能掌握信息系统开发的基本规律。通常将企业信息系统的开发组织命名为研制组或规划组等，无疑，研制组或规划组的组长是一个重要的、关键的人选。他有承上启下的作用。他应该是一个系统分析员级别的人选，因为他必须组织全组的人员从事系统的需求定义，建立业务模型和进行系统实施，并最后实现系统的有效运行。更重要的是他必须妥善解决在系统开发全过程中，可能出现的多种矛盾，因此研制或规划组的负责人不仅应是一位具有广博知识的技术专家，同时还应该是一位具有组织、领导才能的组织者和领导者。

5. 系统开发策略的制定和开发方法、开发工具的选择

信息系统的开发是一项社会、技术工程，其实现有诸多的制约因素，因此从企业自身的实际情况以及周边的环境出发制定出有利于系统开发成功的策略是十分必要的。制定正确的系统开发策略将涉及下列的诸多因素：确定恰当的系统目标；采用正确的开发方式、方法；选择恰当的技术；利用企业的各种资源；动员本企业的技术力量和处理好企业业务人员与技术人员的关系等。经验证明，系统开发策略制定是否恰当，在很大程度上会影响到系统开发的效率甚至成败。

技术的发展使选择和利用系统开发的方法论日渐成熟，并在很多开发商的手里积累了相当多的开发实例。在选择方法论支持系统的开发中，有两方面的问题应该着重考虑：一种是要了解可选择的方法对开发对象的适应性；另一种是开发者对可选择开发方法的熟练程度。最好是能选择一种既合适而又熟练的方法。诸多的案例说明了一个事实：即在一种成功的方法后面，都会有一套与方法相适的、配套的开发工具。当今，软件工程也好，信息工程也好，其发展的成熟性就表现在把方法论和与方法论相适应的工具集密切地结合起来，并形成一整套的诸如需求定义工具、模型建造工具、数据库设计工具和编程工具，甚至还有维护工具等。这样就大大地提高了方法论的可执行、可操作性，并较大程度规范了设计者、实施者的开发行为。工具与方法的统一，以及工具可提供的开发过程中不同开发内容的相互协调性，将提高系统的开发效率，同时也能对提高开发内容的正确性和科学性提供较大的保证。但能否成功运用好在一定的方法指导下的工具集，开发者必须有投入，包括资金的投入和时间的预投入，即要有预先培训和实际应用。

6. 组织基础数据的收集和预处理，实施数据工程

数据是企业日常运行的记录，它抽象于企业现实运行的物理模型，深刻而形象地描述了企业的一切运行状态。现代企业开发信息系统的目的就是要通过它来收集、存储、处理和传播并充分利用这些数据来为企业和客户服务。实际上数据才是系统和企业的灵魂和财富，因此，衡量一个信息系统的成功标准，很大程度上是评价其所收集到的数据的质和量。由此，就可以理解数据工程是信息工程的基础性工程的含义。

实施数据工程会涉及 3 方面的工作：

- **确定收集数据的范围和数量并提出质量要求。**企业所收集的数据的范围应能支持应用的需求，这与企业所经营的业务内容相关；同样，在界定的范围内数据的数量必须支持企业运作的需要，数量往往与所采集数的时间间隔相关，特别是一些与生产行情有关的数据，或与地理位置有关的数据。数据质量保证是数据收集和数据库工程的永恒主题，必须有数据质量保证措施，应对所收集的数据进行必要的评估。

- **有规范的标准的数据格式。**应尽可能遵从某种标准，以便于系统间的交流和利用。对数据格式进行规范应遵循标准，数据标准应是工业化的产物。实施数据规范的原则是：有国际标准应遵循国际标准，没有国际标准应遵循国家标准，没有国家标准应遵循行业标准，如果尚无标准可以遵循，则应制定企业自己的标准，这一部分工作同样应该先行并在制定数据字典时加以体现。

- **完善对主题数据库的设计。**James Martin 提出，设计信息系统应体现以“数据为中心”的设计思想，并以系统的主题数据库设计为重要体现。主题数据库是与组织机构的业务主题有关而不是与传统的计算机应用项目有关的数据库，它使许多应用项目可以共同使用统一主题数据库，而且可以做到大体上无冗余。主题数据库的设计可以加快应用项目的开发，使程序员要使用的数据已经存在于有关的数据库中。

7. 设计并确定系统目标，进行投资估算

系统目标的确定是系统设计的出发点，合适的系统目标有利于提高系统开发的成功率。目标设定可能

对系统产生影响，因此目标设定以前还需进行必要的调研，应在调研的基础上，以需求和约束两方面的材料为依据，合理地确定系统的目标。系统的目标可分为战略目标和战术目标，战略目标是与企业长期的经营战略密切相关的系统目标，它需要通过较持久的、全企业的努力才能实现，它是一个将企业的战略转化到信息系统战略的过程。战术目标则是指某些近期的、局部的、明确的、必须达到的系统目标。科学的系统目标应是既能满足企业的长期发展要求，又是立足于现实的具有可实现性的目标。

系统目标的设定会涉及多种因素，但较关键的因素是投资的限制。由于系统所需的设备、开发费用和维护都与资金有关，特别是系统的规模、技术水平、设备的档次都会直接和间接地制约系统的目标设定，因此在进行目标分析时企业领导应在系统投资额度上有一定倾向性的意见。

较为理想的做法是能根据调查的情况设计几种不同的目标，即对系统提出不同的要求，并给出相应的功能概述和测算出系统的投资；或者根据不同的投资额来设计和落实系统目标的功能，以供企业的领导部门比较和评选。

8.合理设计信息部门在企业机构中的位置

随着信息技术在企业的应用水平和作用的提升，信息部门在企业机构中的位置也在发生着变化，它基本同步于信息处理在企业中所经历的各个阶段。当前在企业中信息部门多采用信息中心的名称，在政府部门也是如此，回顾计算机在企业应用发展的过程，信息部门在企业机构中的安排大致可归纳成3种形式：

- 早期的较原始的形式，将计算机设备和应处理的业务由相应主管部门负责，如将信息处理业务交给财务部门或生产设计部门主管，其他各部门需要计算机处理业务时，可通过主管单位统一安排。这种组织方式的优点是设备集中，投资经济也易于管理。缺点是设备所在单位有使用优先权，对需要使用计算机的任务在安排和调度上必然存在局限性。

- 将信息处理部门与其他业务部门安排在平行的位置。信息处理部门作为一个独立的单位，直接隶属于企业的行政主管。所有的有关企业业务的计算机应用开发、操作和维护等都集中在一个称为计算机中心的职能部门。这种组织方式的优点是集中了计算机技术人才，可以集中规划、集中开发，可以提高开发人员的业务能力，减轻作业成本，并便于统一调度和管理。这种管理方式的最大缺点是信息部门要应付各种不同业务需求，在很多方面很难达到各业务部门对信息处理的要求。

- 信息部门在企业中的位置高于其他业务部门，信息部门负责建立整个企业的信息系统，在组织形式上就充分强调了信息以及信息部门在企业中的重要作用和地位。在这种形式下，必须妥善处理好信息部门与其他业务部门的关系，以保证企业信息渠道的畅通，同时对全企业能起到、控制和调节作用，以保证信息部门对整个企业的管理和决策起到作为最高管理部门的助手作用。信息部门在企业中的作用随着企业对信息技术应用的加强和深化，其重要性也在不断地加强，但不论信息技术在企业经营管理中的作用如何巨大，对于企业来讲其基本业务却是最为核心和关键的。基本的业务工作是企业的本质和目的，而信息技术是一种手段，因此，正确地处理业务和技术的关系，正确地处理业务部门和技术部门的关系，如何做好业务部门对技术部门的支持和交流，如何做好技术部门对业务部门的服务是一个十分重要的问题。当前为了能在企业或单位的最高管理者统一协调下完成企业或单位的信息化过程，在组织机构上又出现了作为一级协调和领导机构的信息化办公室。这类组织的出现，既反映了各级领导对信息化工作的重视，同时也是为了能从全局来控制并引导企业或部门信息化工作的健全发展。

9.应用自动化的手段来开发系统

在传统的系统建设方式上，人们缺少自动化的系统建造手段，在很大程度上还是依靠人的经验、系统的积累和手工方式。随着软件技术的发展，人们在工具软件的研究方面已取得不少成果，并也在系统开发中得到应用。诸如已产生过一些支持特定应用的软件开发工具，包括各种应用生成器(AG)，通用的第四代语言(4GL)；还有一些属于系统开发中的工具，如支持需求定义和分析的软件；以及各种支持绘制系统流程、数据流程、结构图的工具。此外还有支持数据库设计的工具，如自动绘制E-R图的工具。后来又出现了一些计算机辅助软件工具，即CASE。综合起来可以看到，在软件开发领域已经出现一类新的软件，即所谓工具软件，它已成为软件发展中的重要组成。当前，系统开发的整个发展过程的总趋势是，要能以自动化的手段来开发自动化的系统，要将成熟的开发系统的方法论与支持实现方法论的工具严格地结合起来，从而使开发者在经验的基础上更加有效地开发系统。方法和工具只是一种客观存在，只有熟练地掌握和利用这些工具才能称它为生产力，才能在系统开发中发挥真正的作用。

信息系统的生命期包括了系统的开发期和系统的运行期，从理论上讲，一个系统的运行期应该比其开发期要长很多，这说明这个系统得到了应用，也就是说，系统的开发是成功的。如果一个系统的运行期很短，甚至没有得到真正的应用，那就是一个失败的系统。系统建成后从开发转入到运行，即进入维护期，维护是信息系统运行中的重要工作。曾经有人说过“没有维护就没有软件”，同样的道理也可以说，“没有

维护就没有信息系统”。维护的内容是丰富的，维护的工作也是重要的。成功的系统一般都必须经过从基础性建设到优化再到取得效益的过程，其中优化即是通过维护来达到的。

1.认真做好系统的验收工作

验收是在对系统的严格测试基础上进行的。软件和系统都必须在开发之前制定与系统开发同步的测试计划。没有经过严格测试的软件不可能是合格的软件，没有经过严格测试的系统同样也不可能是合格的系统。完备的验收过程应该经过开发者自己的测试和验收，企业技术人员的验收，企业业务人员的验收和最终用户的验收，验收必须有计划、有依据、有结论。软件和系统测试是一件高投入和费时间、费人力的工作，而且也是一件有技术难度的工作，它经常会被人们有意或无意地忽视。有意是回避系统开发中的难题，其行为类似偷工减料；无意是没有意识到测试的意义和作用。因此对于大型工程而言，往往是测试准备和开发工作同步启动，并十分强调对测试资金和人力的投入。只有经过严格的测试才可能实施正常和有效的验收。

2.着力优化系统的功能和性能

从系统开发工作的完成到系统真正地取得效益，其中必然有一个对系统进行优化的过程。经验证明：要想使企业信息系统交付使用即能达到理想的运行效果，并在功能和性能上都能满足企业的需求是不现实的，尚需经过进一步的努力，做到完善已有功能，进一步提高系统运行效率。同时还应使用户在使用中熟悉系统并发挥系统的潜能。系统维护的内容和方面甚多，但其中最重要也是最主要的是对应用软件的维护。在软件工程中，对软件系统的维护性内容，即对软件所进行的维护，包括对软件的完善性维护、适应性维护和校正性维护等，都属于对系统的优化过程，特别是其中的完善性维护。完善性维护是指一个软件投入使用后，根据用户关于增加新的处理功能、修改原有功能以及各种改进的要求或建议，对应用软件系统的功能和性能做进一步的修改和补充，使之更为完善。在软件工程中，将软件的易维护性也定为一个重要的软件质量标准。

从以上论述可见，系统从开发到运行再到取得效益有一个系统优化的过程，不经过对系统的优化就想得到一个使用有效的系统往往是不现实的。因此，那种对现有系统不满意即摒弃而再另建，甚至重复这种做法，是对信息系统建设规律的不了解所导致的，应当避免。

3.重视文档的整理和接收

系统文档是对现有系统的文字和图形的描述，它最准确和生动地刻画了企业交付使用的系统的现状，因此也作为对今后系统维护的依据。实际上，系统文档也是对系统验收的重要组成部分。如果系统的文档不齐全或不准确，则将带来维护的极大困难，因此应该要求系统的开发者严格按照信息工程方法的要求来形成文档并最后提供文档。系统文档不全或不正确的原因有：开发者没有工程化的意识和习惯，管理不严格；开发者缺少支持形成正规文档的手段和工具。而最根本的原因还是人们对文档的重要性认识不足，它在很大程度上反映出，现时的信息系统还没有真正地成为我们企业的不可缺少的部分，它还不能对企业的效益增长起到决定作用，也就是当前我们离企业的真正信息化为时尚远。

4.重视系统维护队伍的建设

系统维护是系统运行的重要保证，除了外包式系统外，都需要一支与企业有密切关系的维护队伍，它会涉及诸多技术方面，如硬件、软件、数据库、网络等，系统管理员、数据库管理员、网络管理员等都是这一队伍中的必要成员。作为一类系统运行保障，往往人们并不十分强调其工作的创造性，因此维护工作并不是技术人员最向往和最热衷的工作，但它们又都是企业信息系统得以正常运行的不可缺少的工作。因此，企业最高领导采取必要的措施，出台必要的政策去组织一支稳定且有效的维护队伍是十分必要的。从某种意义上讲，开发好一个企业的信息系统是重要的，而维护好一个企业的信息系统更为重要。

1.3 信息系统工程所涉及的技术内容

信息系统工程作为一门综合的技术，与多种学科和技术有着深刻的内在联系。从总体上讲，它会涉及到社会和技术两大领域，并综合应用了管理科学、系统科学、数学、计算机科学、行为科学的研究成果，逐渐形成了自己的新的学科体系。信息工程还是一种正在完善中的新事物，人们并没有完全认识和掌握它的内在规律。因此，在实施中仍然可能存在着一定的盲目性，这一点从国内、外实施信息工程时的高失败率的现实即可得到印证。

1.3.1 管理科学的应用

对于企业管理知识的系统研究开始于 19 世纪后期，传统的管理方法将企业视为一个有机的整体，但其主要的研究对象却是一些企业中存在的特殊问题，为的是能以最高的效率达到企业组织的各项目标，重点是通过改进计划工作的组织结构来提高企业的工作效率。随着企业的组织结构和外界环境的变化，与其相关的管理方法也在发生着改变并得到发展。有人认为对企业管理方法的研究方式大体上可以划分为两种

思路，一种是从行为科学出发，而另一种则是应用数学模型化方法。信息技术的发展使人们可以在更广阔的技术领域重新对企业管理进行认识和变革，即如何更紧密地将企业的业务内容和在新技术支持下的运行方式更紧密地结合起来，从而形成一类更新型的企业运作模式，诸如虚拟企业、虚拟工业、电子商务等。但不论如何变化，企业运作的内容是基础，形式则更侧重于手段，这是最为本质的。

对企业组织的工作方式进行心理学和社会学的综合研究，是从行为科学出发的研究方法。一般对于行为科学可以做两种理解：广义的理解，它是指包括部分社会科学和自然科学在内的有关研究人类行为规律的诸科学，它也是由多门学科所组成的综合性边缘学科。这一点由行为科学的英文“behavioral sciences”原意，即“研究行为的诸科学”即可说明。而狭义的行为科学则是指将“研究行为的诸科学”的原理应用于管理，特别是应用于企业管理，而形成的一门新的以人为中心的管理科学。我们所指的行为科学往往是指后一种，它在研究中着重于系统的社会心理学性质，研究企业人员的各种需要，他们进行活动的各种动因和行为方式以及他们之间的相互关系等。

数学方法则是强调企业运作的数字表征以及各种企业运作模型的设计。各种方法向管理人员提供对某一具体问题的解决方案，它可能是从多种方案中找出的最好方案。如运筹学就是数学方法应用于管理的好例子。运筹学应用的广泛领域也多是面向管理的计算机信息系统的应用领域，而运筹学常用的模型也是面向管理的计算机信息系统支持决策常常需要的模型。因而像运筹学这样的数学方法在信息系统开发中对高层次的决策支持功能提供了重要的手段和方法。

1. 3. 2 方法论的发展与应用

现在已经认识到，信息系统的开发一般都经历系统规划、需求定义、系统设计、实施和维护几个阶段，而它们都应该在科学的方法论的指导下来完成。早期的开发，却多是利用和基于开发者自身的经验。长久以来相关领域的专家和工程技术人员都在不断地研究并提供了多种方法，如软件工程方法、信息工程方法等。从历史的发展来看，信息系统开发的方法论的发展经历了以下的阶段。

1. 基于经验的开发

早期，计算机在管理方面的应用只是计算机信息系统的雏形，称为电子数据处理系统。它功能单一，多属于一些针对具体的事物处理和业务控制的应用。当时的系统开发的重点是利用特定的计算机程序设计语言来编写符合业务功能的程序，如利用 Cobol 语言来编写计算账目或人员管理的具体应用程序。在程序实现中更多地依靠程序人员的技巧和经验，细心和认真程度。在 20 世纪 50 年代到 60 年代，程序设计曾被认为是能发挥创造才能的技术领域。当时人们衡量一个程序人员的水平和熟练程度的重要标准之一，是其在计算机上的累计工作时间，正像衡量飞行员的水平和熟练程度是以前机上飞行小时的多少来判断的一样。虽然基于经验的应用开发似乎原始，但它的经验积累，它所经历的困难和所存在的问题却推动着技术向前进步，促使信息系统的建设从经验走向规范。

2. 软件危机与软件工程

软件是由计算机程序演变而成的一种概念。它当前已是信息系统的主要和重要组成。程序是按既定算法，用某种计算机语言规定的指令或语句编写的集合体。软件是程序再加上程序实现和维护时所必需的文档的总体，它是程序和程序设计发展到一定规模和走向商品化后所形成的概念和成果。20 世纪 60 年代以后，随着计算机应用需求的驱动，软件的规模和复杂度都在不断增加，因此仅仅依靠个人的经验来生产软件已经难以满足应用的需要，更为困难的则是软件生产的复杂性和高成本，使软件的生产陷入危机。它表现为：软件的需求在增长，开发者却无法应用的需求；软件价格昂贵，生产成本很高使用户难以接受；软件的需求定义难以准确，很容易偏离用户要求；软件的生产进度无法控制、质量不易保证以及软件的可维护性很差等。归结起来，软件危机主要表现是：一方面是无法满足日益增长的对软件的需求；另一方面是难以满足对已有软件的维护需要。危机的出现，促使人们去寻找产生危机的内在原因，进而发现其原因：一是软件生产本身所存在的复杂性；二是缺乏完善的软件开发方法和技术。基于上述认识，专家们在 20 世纪 60 年代末经过研究提出了克服软件危机的软件工程概念和方法。

软件工程形成了软件生产的工程化思想，它促使软件生产从基于开发者的个人或小集体的经验走上较为正规化和规范化的道路。自从软件工程思想提出以来，专家和开发者们经过多年的实践，将这门重要的科学和技术不断地发展和深化，从而形成了“软件工程学”这样一门计算机科学的重要分支。

当前作为信息工程的重要组成的软件工程已从单纯的软件开发方法发展成为系统的科学，其所包含的内容已涵盖了软件开发技术和软件工程管理两个相互联系而又有不同侧重的技术内容。软件开发技术包括软件开发方法学，软件开发工具和软件工程环境；软件工程管理包括软件和软件工程管理以及软件工程经济学。

软件工程方法将软件，特别是大型软件的开发过程划分为阶段，每一阶段有明确的任务，在完成任务

时又必须产生相应的成果或文档。它强调首先明确软件需求的重要性，并在需求明确的前提下，先进行对软件的总体设计，再进行软件各部分的详细设计，编码则应在详细设计完成后才进行。软件工程强调对软件正确性和性能的测试，即应对软件开发进行全程的质量监控。上述思想和措施都是将软件的生成按工程化的办法来实施，并在实施中强调加强管理的重要性。

3.自底向上和自顶向下

信息系统开发的发展过程经历过所谓“自底向上”方式和“自顶向下”方式。两种方式都曾被人们实践和发展。

早期的对信息系统的分析、设计和开发方法由于规模较小，基本上是采用“自下而上”的，或称“自底向上”的方式。系统的开发是从单项、局部的应用向多项、全面的应用发展。它们从部分现有的应用向外或向上延伸和扩展，这种方法主要用于对早期的事物处理应用。一些系统加上另外一些系统，将它们联系起来使企业的信息系统逐渐扩大，从而支持管理部门的业务控制、管理规划甚至战略决策。它们是从现有的信息系统开始，根据企业需求的变化而不断演化。所以“自底向上”的分析、设计和开发方法也称为演变法。

随着信息系统规模的不断扩大和对传统开发方法论的探讨，另一种系统开发的方法论被提倡和发展，它就是所谓“自顶向下”的系统分析、设计和开发方法，这也是当前大系统开发所常用的方法。它是从企业或部门的经营和管理目标出发，从全局和整体来规划其信息需求。一它从企业或机构的最高层出发并覆盖所有或主要的业务领域。运用这类方法可以为企业或部门信息系统制定中期或长期发展规划奠定基础。自顶向下方法在一定程度上保证了合理的开发顺序和所有应用的最后整体化。

人们从整体上分析和总结了两种方法的优缺点。

自底向上方法的优点有：

- 使信息系统的开发易于适应组织机构的真正需要。
- 有助于发现和理解每个系统的附加需要，并易于判断其费用。
- 每一阶段所获得的经验和教训有助于下一阶段的开发。
- 相对地说，每一阶段的规模较小，易于控制和管理。

自底向上方法的缺点有：

- 由于方法的演变性质，信息系统难以实现其整体性。
- 由于系统未进行全局规划，系统的数据一致性和完整性难以保证。
- 为了达到系统的性能要求，往往不得不重新调整系统，甚至要重新设计系统。
- 由于系统实施的分散性和演变性，因而与企业目标的联系往往是间接的，系统往往难以支持企业的整体战略目标。

自顶向下方法的优点有：

- 可为企业或机构的重要决策和任务实现提供信息。
- 支持企业信息系统的整体性规划，并对系统的各子系统的协调和通信提供保证。
- 方法的实践有利于提高企业人员的整体观察问题的能力，从而有利于寻找到改进企业组织的途径。

自顶向下方法的缺点有：

- 对系统分析和设计人员的要求较高。
- 开发周期长，系统复杂，一般属于一种高成本、大投资的工程。
- 对于大系统而言，自上而下的规划对于下层系统的实施往往缺乏约束力。
- 从经济角度来看：很难说自顶向下的做法在经济上是合算的。

上述在信息系统开发时常见的两种实施方法，是对不同时期、不同对象的信息系统开发方法的归纳，各有其优缺点，但实践证明在工程实施时，两种方法并非是绝对排斥的，往往在事情进一步的发展中，它们都能取长补短、相互补充。有经验的分析和设计人员，他们会首先确定企业的信息需求环境和性质，然后来选择适合于它的分析和设计方法，他们甚至会从方法的基本原理和适应对象出发使用变通的方法来进行对特定系统的开发，如自顶向下的整体规划和自底向上的分步实施。这无疑是一种对方法论的发展和创造。

4.模型化

对于信息系统，特别是其核心部分的软件系统的开发，专家和工程人员已经从不同的角度、用不同的方法对它们进行了模型化，从而将相关的方法从实践上升到理论。到目前为止，已经提出多种软件生存期的模型，软件的生存周期在很大程度上反映了信息系统的生存周期。瀑布模型、螺旋模型和喷泉模型是软件开发中有影响的几类模型，它们针对不同性质和规模的系统分别规定了不同的工程化活动。

(1)瀑布模型

瀑布模型是一类在软件和系统开发中应用广泛、影响深远的模型，它规定了软件工程的各项活动，包括系统规划，需求分析，软件设计，编码，测试和维护，其过程如图 1.1 所示。

瀑布模型为软件的开发和维护提供了一种有效的模式。可根据这一模式制定出开发计划，进行成本预算，组织开发力量，以项目的阶段评审和文档控制为手段有效地对整个开发过程进行指导，从而力求软件产品能及时交付，并达到预期的质量要求。虽然瀑布模型在一定程度上在消除非结构化软件、降低软件的复杂度、促进软件开发工程化方面起到显著作用，但同时在大量的软件开发实践中也暴露出瀑布模型的缺点，其中最严重的是它缺乏灵活性，难以解决软件需求的不明确或不准确的难题。

(2)螺旋模型

为了解决瀑布模型实施时存在的问题，专家们提出了一种基于原型化开发的进化模型。其实施过程是首先做实验开发，以探究其可行性，并明确软件需求，其结果为一个“原型”，再基于原型去开发一成功产品，将进化模型加以实施，并对实施中可能出现的风险进行分析，即构成螺旋模型。它认为风险是软件开发不可忽视的潜在不利因素，因此应及时对风险进行识别、分析并采取对策，从而减低风险。

螺旋模型的实施过程如图 1.2 所示，

它在笛卡儿坐标的 4 个象限上反映出 4 方面的活动。

制定计划：确定软件目标，选定实施方案，分析项目开发约束条件。

风险分析：分析所选择的方案，考虑该方案可能存在的风险以及如何规避风险。

工程实施：软件开发过程的实施。

工程评估：对成果进行评估，并提出修正意见。

螺旋模型自内向外，从坐标的第一象限开始逐步演化，在实施中可采用开发者认为可行的方法。螺旋模型适合大型软件的开发，它采用进化的方法，并对可能出现的风险做出反应。使用该模型需要具有丰富经验的专家，他们应有相当丰富的风险评估经验和系统开发的专门知识。

(3)喷泉模型

喷泉模型是一类支持面向对象的、自底向上的开发模型，在实施中体现了迭代和无间隙的特征，其过程如图 1.3 所示。它在实施中，对系统的某个部分常常重复迭代，相关的功能在迭代中逐步地演化到系统中，并且能体现无间隙特征，即与瀑布模型相比，其分析、设计和编码之间没有明显的界限。

不难看出模型和方法之间存在着深刻的内部联系，它们是随着人们的经验和手段的进步而发展，并不存在着根本的差别。如从瀑布模型过渡到螺旋模型，可以看出前者仍然是后者的基础，后者是前者的改进和添加。技术的进步也促进了模型和方法的前进，如对象技术的发展和实施是促进喷泉模型发展和成熟的动力和原因。在研究和应用信息系统开发方法时，注意探索它们之间的联系对理解方法和模型的本质，进而灵活地应用它们是非常有益的。

1.3.3 从软件工程到信息工程

术语“软件工程”是指用于说明、设计和编制计算机软件的一套规范。术语“信息工程”是指以当今数据系统为基础，建立一个计算机化企业所需要的一套相互关联的原则。信息工程的主要焦点是用计算机来存储和维护数据，而信息则是从这些数据提炼出来的。软件工程的主要焦点是用于计算机化处理过程的逻辑形式。

James Martin 认为，软件工程技术形成于 20 世纪 70 年代，这些技术包括软件开发方法学(如结构化程序编制方法、结构化系统分析方法和结构化系统设计方法等)以及支持这些方法的各种工具，这些都是

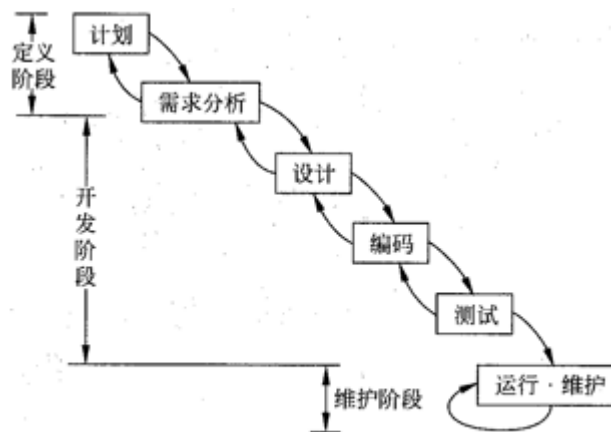


图 1.1 瀑布模型

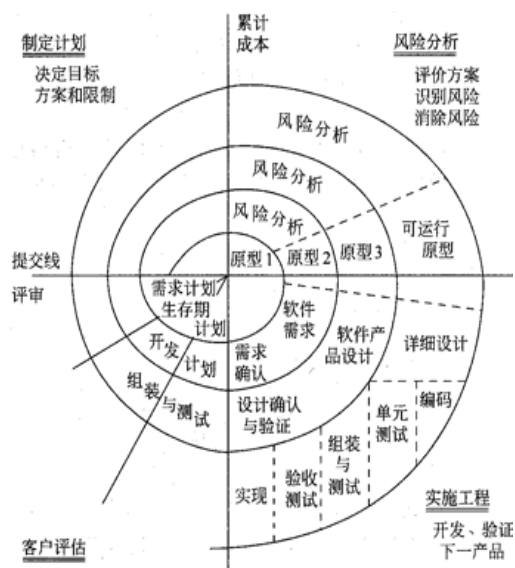


图 1.2 螺旋模型

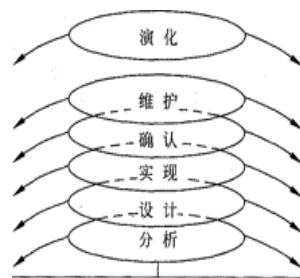


图 1.3 喷泉模型

建立具有复杂逻辑结构的大型软件所绝对不可少的。虽然在 70 年代各种技术得到了迅速的发展，但足够好的技术却不多，而很多已建立的信息系统又不能完全满足企业管理的要求。所以 James Martin 认为，信息工程是把创建企业成功的信息系统所使用的技术加以总结、提高和规范化而形成的，而信息工程中所使用的种种图表和工具，往往是从软件工程中继承来的。

图 1.4 描述了信息工程的前提，在现代数据处理中，要以数据为中心，数据的存储和管理是通过各种数据系统软件来支持的。图 1、4 的左边是建立和修改数据的各项处理，通过适当的、准确的控制方法来收集和输入数据，这些数据往往还要进行周期性的修改。

James Martin 认为，信息工程的第二个基本前提是：一个企业的数据类型变化不能也不会太大。数据是按实体

存储的。除了在极特殊的情况下需要加入新的实体类型外，在项业务活动的生存周期中，实体类型一般是不会变化的，甚至实体的属性类型也很少变化。而经常变化的则是数据类型的值，如机场中候机大厅中的航班信息在不断改变，但其数据格式一般是不变的。如果一开始就对数据进行认真的设计，那么，数据的结构很少会变化。因此，只有数据被正确地标识和结构化时，数据才有生命力，才能被灵活地使用。

由于基本数据类型是稳定的，而数据处理过程是趋于变化的，所以当使用面向处理过程的技术失败时，正确地使用面向数据的技术则有可能会成功。采用面向处理过程的技术所产生的许多系统，实施缓慢且难于变化，而信息工程则着眼于迅速地满足管理者不断变化的信息需求。一旦所需要的数据基础结构建立起来，就可以使用高级数据库语言和应用过程生成器工具很快地得到所要的结果。

无疑，当前的信息工程方法大量地吸收了软件工程的很多技术成果，因为从某种程度上来观察，软件工程实际上可认为是信息工程的一部分，当前的信息工程方法除了突出了其以数据为中心的特征外，还将工程的实施有机地划分为对业务系统的实施和对技术系统的实施。前者包含了软件工程的技术内容，而后者则包含了诸如硬件、网络等工程内容。实施信息工程就是将企业的业务系统与技术系统有机地结合起来。

1.4 系统分析员及其培养

现代信息工程是一项复杂的社会和技术工程。随着社会的发展，企业的业务内容在不断地扩展；随着科学技术的进步，企业所运用的科技手段也有日新月异的变化。因此，当前所开展的企业和企业间的信息系统的内容和形式与过去相比都有了很大的发展。当前，信息系统的建设已呈现出多方面的特点：首先是由于竞争的需要，企业的业务内容和开展业务的形式在不断地发展和变化，除了生产、销售外，还会有研究、服务的业务需求；技术，特别是对信息技术手段的应用，使企业的生产、经营呈现出全新的势态，计算机和网络介入企业的生产和运作环节，使企业产生了革命性的变革，诸如虚拟企业、电子商务等都将对企业的信息系统建设提出高要求和新要求。在对大型、复杂的信息系统建设中，要求有一支训练有素、经验丰富、能适应形势的系统开发队伍和人员，而在这支队伍中的领军人物就是系统分析人员，即系统分析员。system analyst 作为一个专用名词，专指大系统开发中的分析、设计和领导实施的人。一定意义上讲，系统分析员的水平将影响到信息系统开发的质量，甚至成败。当然，在一支完善的信息系统开发队伍中，除了系统分析员外，还需要有业务专家、技术专家和其他辅助人员。

按当前我们国家从事信息系统建设的情况和习惯，系统分析员在系统开发的各个阶段，都将担负着重要的任务，在工程进展中常处于重要的地位，可将其应具有的能力和素质归纳为：

- 必须理解和明确系统建议、企业的经营管理业务和目标以及战略发展方向。
- 要与企业最高领导和管理人员一起设计和确定企业信息系统建设的长期目标，还要对目标进行必要的分解。
- 要在调查企业内部现状和外部环境的基础上对企业信息系统建设的可行性进行分析，并得出必要的结论。
- 要根据企业所处的环境和所具备的条件，按照所确定的目标来制定适合企业信息系统的开发策略。
- 应从现有可供选择的方法和工具中，选择出适合企业信息系统开发所需要的方法和工具，并对开发人员进行培训。
- 必须在充分了解企业业务需求的情况下，建立企业的业务模型，并与企业决策者和业务人员进行交流，达到共识。
- 应根据当前信息技术及产品的发展建立企业信息系统的技术模型，并将它与业务模型结合，建立起完善的企业信息系统的模型。

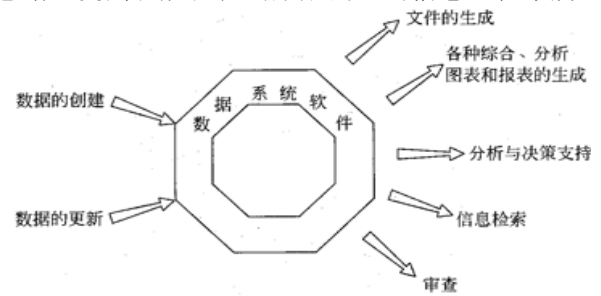


图 1.4 以数据为中心

- 应对企业信息系统开发人员的组织、机构建立、人员安排和实施计划提出意见和建议，并组织好对相关人员的有针对性的培训，以求在开发工作进行中能目标一致，行动一致。

由以上系统分析员所应承担的任务可知，在整个系统开发中，他担负着多方面的任务，并具有以下特点：

- 系统分析员既是企业信息系统建设中系统需求的分析者和系统的设计者，同时又是系统实施的领导者和组织者。他既要对系统进行宏观和总体的把握，也要对系统的局部有较具体的了解，更要掌握好系统开发中的关键点，即里程碑。

- 系统分析员是企业管理人员、系统监理人员和系统开发人员之间的联系人员和信息的沟通者。

- 系统分析员既要根据企业的条件去满足系统的需求，同时也要运用自己的经验和智慧去尽可能地完善和改进新的企业信息系统。

由于系统分析员所承担的任务，他应具有较高的业务素质，只有这样才能满足所承担的任务对他的要求，这些要求包括：

- 系统分析员应具有系统科学的观点，善于运用系统科学的观点和方法来认识和处理企业信息系统开发中出现的各种问题。

- 系统分析员应是既有技术知识同时又有一定社会知识的人才，仅仅是精通某一种技术的专业人员，往往还难以承担系统分析员的任务。

- 系统分析员除了应该具有信息系统开发的理论知识外，还应该是已积累了较丰富经验的工程人员，理论知识能促进和提高系统开发的科学程度，经验能帮助在系统开发中提高效率 and 少走弯路，并提高其实用性。经验的积累在系统开发中有利于知识和成果的运用，以提高系统的成功率。

- 系统分析员必须具有组织和管理才能，并善于处理各种人际关系。只有这样才能协调好复杂的信息系统开发中的多种关系和多个环节。

总结上述有关系统分析人员的论述，人们认为系统分析人员应具有某些特殊的素质，这些素质被总结为：

- 系统分析员应具有深入观察问题的能力，他善于透过现象认识问题的本质。

- 系统分析员应具有丰富的想象力和创造力，他勇于接受新鲜事物，善于从经验的积累中进行创造。

- 系统分析员应具有很强的谈判和协商的能力，善于将自己对系统开发的认识介绍给用户，并说服用户接受自己的主张。

- 系统分析员应具有很强的组织管理能力，在大系统的开发中科学的组织和管理才能产生高的效率和保证质量。

- 系统分析员应具有与人共事合作的精力，他能带领开发组的所有成员，齐心协力、合作共事执行各自承担的任务。

总之，系统分析员应是一类有很强的事业心和使命感、并且能从实际出发解决具体问题、具有务实精神的杰出复合型人才。

系统分析员的选拔和培养一直是社会信息化过程中比较突出且具有难度的问题，而它又是一个关系信息化建设成败的问题，不可想象在社会信息化的过程中没有优秀的领军人物却能获得好的成果。

系统分析员的来源可能存在两种途径：一种是从专业人员进行选拔再进行培训；另一种是从熟悉计算机技术的技术人员中选拔再进行培训。他们除需要较系统地学习和掌握系统分析员的基础知识和应用知识外，更重要的是应该通过案例分析和业务实践来提高实际工作能力。

一般认为，系统分析员培养的困难在于，不同的企业、不同的行业都有自身的业务背景、特点和技术领域。它们各有不同，系统分析员要了解不同企业、行业的业务只有进行行业业务培训或实践，较难组织统一培训。所以通常的系统分析员的培训主要是针对某些具有共同需求的知识进行。这种培训可以采用两种方式，一种是长期的，如可选择的一种方案是用 13~16 周时间；另一种是短期的，如用 3~5 周时间。现举例给出长期训练项目的一种参考计划：

- 信息系统概论(1-2 周)。
- 计算机系统概述(3 周)。
- 系统分析方法论(4~5 周)。
- 实例分析(2~3 周)。
- 系统分析设计工具(3 周)。

上述培养计划是基于学员脱产学习，他们有更多的时间，能较系统地去了解相关的知识内容，这些内容包括实际的和理论的两方面知识。

短期培训是集中进行系统开发中所用到的基本方法和技术的培训。不论是长期的还是短期的培训，其所包含的内容都将包括如下几个方面：

- 管理科学和系统科学的知识。
- 计算机技术和通信技术的新发展。
- 数据库技术与软件工程技术。
- 信息系统开发方法和工具。

我国的现代信息系统的研制和开发，严格地讲，还远未达到成熟的阶段，我们还必须在实践中总结经验，并吸收国外成功的、有效的方法论，借鉴他们的经验，吸取他们的教训，并根据国情来制定企业系统的开发策略，以提高我国企业信息系统开发的成功率。这些任务将在一定程度上依赖于我国系统分析员队伍的建设以及其整体素质。当然，社会信息化是一个历史的进程，它将涉及整个社会的多方面，但愿在这一历史发展的过程中，会有更多的成熟用户，有更多的成熟开发商，当然更希望会有更多的成熟系统分析人员出现。

1.5 系统分析员教程的内容组织

系统分析员教程作为基本覆盖系统分析员考试大纲的教材，被划分为两部分：一部分是介绍系统分析员应掌握的基础知识的，另一部分是介绍系统分析员应掌握的应用知识，本书的内容是指后一种。

系统分析员基础教程按大纲要求应包含如下内容。

- 数学：包括微积分、线性代数、概率统计、离散数学、数值分析和算法复杂性。
- 管理科学：包括管理科学基础、规划论、对策论、决策论和排队论的相关知识。
- 系统科学：包括系统工程原理、系统模型和模拟、系统评价。
- 专业英语：包括英语科技用语和短文阅读范例。

这些内容可参考由清华大学出版社出版的《计算机综合应用知识(第二版)》一书。

本书包含如下内容。

- 信息系统：包括计算机信息系统概念、分析、设计和开发方法，着重介绍了结构化分析、设计方法，企业系统规划方法，战略数据规划方法，信息工程方法和应用原型化方法。
- 软件工程：包括软件生存期过程，软件过程能力评估，软件配置管理，面向对象开发方法和软件复用技术。
- 数据库与数据仓库：包括数据库概念、系统和理论，数据库设计，数据仓库概念、结构和开发。
- 计算机网络：包括网络基本概念、体系结构和协议，局域网和广域网技术，网络管理与网络安全，Internet 与 Intranet，信息服务与网络应用和网络工程。
- 计算机系统与配置：包括计算机系统结构、系统类型，计算机系统评测，计算机系统安全。
- 信息安全技术：包括方法学，通信和网络安全，安全管理实施，应用与系统开发安全，密码术，安全体系结构和模型，计算机操作系统安全，业务持续和灾难恢复，法规和道德规范，物理安全。

本书可以大体上分为三部分：第一部分是有关信息系统的综合知识，它是本书的基础，涵盖了有关计算机信息系统的基础知识，是对相关问题的概括；第二部分是计算机信息系统开发的方法论，着重介绍了流行的结构化分析、设计方法，它应视为掌握系统分析、设计方法的入门和基础，另外还介绍了企业系统规划方法、战略数据规划方法和信息工程方法。介绍这些方法的目的，并非这些方法是目前最新的或最流行的，而是在相应的方法中不仅体现了技术内容，更体现了社会内容，方法适应的是 20 世纪加年代美国的社会现实，在很多方面对我国当前的系统开发都有较大的借鉴意义和价值，而且方法所提供的策略、技术和工具也是相对稳定的。如果一个系统分析人员能很好地掌握和体会到方法的精髓，再去学习和运用其他方法都会事半功倍。第三部分是信息系统开发的支撑技术内容，包括软件工程、数据库、网络、系统和配置以及信息安全。无疑，这些都是计算机信息系统开发时不可或缺的技术内容，掌握好这些技术内容对于系统分析员也是很重要的。

第 2 章信息与系统

2. 1 信息与信息化

2. 1. 1 信息时代与国家信息化

20 世纪是人类社会极其辉煌的世纪，计算机的出现，使人类社会的发展以极快的加速度进行。至 20

世纪末叶，人类社会的发展可以用“突破”或“转折”这些词来形容了。也就是所谓一个新的社会形态——信息时代正向我们走来。作为一个信息系统分析员，必须站在时代的高度，审视这种变化，高瞻远瞩，并以此指导自己的工作。

关于信息社会的特征有很多说法，主要有以下几点：

- 信息技术飞速发展。
- 生产力和经济发展的关键因素是信息和知识。
- 信息产业成为许多发达国家的支柱产业。
- 互联网和电子商务高速增长。

所谓国家信息化是国家意志的一种体现，中国国家信息化的实质是“在国家统一规划和组织下，在农业、工业、科学技术、国防及社会生活各个方面应用现代信息技术，深入开发、广泛利用信息资源，加速实现国家现代化的进程”。这里包含了4层意义，一是国家信息化要有国家统一规划和组织，是国家行为；二是信息化是覆盖现代化全局的，实现国家现代化都离不开信息化；三是各个领域都需要广泛应用信息技术，深入开发、利用信息资源，调整产业结构，以信息化带动工业化，发挥后发优势，努力实现技术的跨越式发展；四是国家信息化是一个不断发展的过程。

国家信息化体系包括6个因素。

- **信息资源：**信息和材料、能源共同构成经济和社会发展的3大战略资源。我国信息资源极其丰富，但开发和利用远远落后于需要，因此该要素既是我国信息化中的关键，又是一个决定性环节。
- **信息网络：**信息网络是信息资源开发、利用的基础设施，信息网络包括计算机网(数字网)、电信网、电视网，在国家信息化的过程中将逐步实现3网融合和最终达到合一。
- **信息技术应用：**信息技术应用是国家信息化中十分重要的要素，它直接反映了效率和效益。
- **信息产业：**信息产业是信息化的物质基础。信息产业包括微电子、计算机、电信等产品和技术的开发、生产、销售，以及软件、信息系统开发和电子商务等。从根本上来说，国家信息化只有在产品和技术方面拥有雄厚的自主知识产权，才能提高综合国力。
- **信息化人才：**人才是信息化的成功之本。不仅要有各个层次的信息化技术人才，还要有精干的信息化管理人才，营销人才，法律、法规和情报人才。
- **信息化政策、法规、标准和规范：**信息化政策和法规、标准、规范是国家信息化快速、有序、健康和持续发展的保障。

2.1.2 信息与数据

随着当代科学技术的发展，管理科学中的新理论、新概念层出不穷，但其中位居显赫者要算是“信息”这个概念了。如今，信息已不再仅仅是“消息”的同义词，它以不断扩展的涵义，渗透到各个科学技术领域和整个社会管理运动过程中，被列入“材料、能源、信息”这3大科技支柱中，构成现代社会文明和人类智力水平的一个重要标志。

在现代化管理中，信息论已成为与系统论、控制论等相并列的现代科学主要方法论之一。信息价值，信息量，信息反馈，信息时效性、真实性，信息处理、传递，以及信息论与信息科学是现代化管理的运动命脉。实际上，现代化管理与信息已融为一体，并形成一种特殊形态的信息运动形式，即管理系统信息流。

2.1.2.1 信息与数据的定义

1948年信息论奠基人美国科学家香农(Shannon)在《通信的数学理论》一文中把信息理解为“用以消除随机不确定的东西”，同年控制论创始人维纳(Wiener)在《控制论》一书中指出“信息就是信息，不是物质也不是能量”。近代信息管理和信息系统学科认为信息是“事物之间相互联系、相互作用的状态的描述”。“是客观世界各种事物变化和特征的反映。”上述定义的描述似乎有些抽象，实际上我们平时采取实用的方式理解信息，如“信息是加工后的数据”，“信息是可以通信的数据和知识”，“信息是管理决策的依据”等。

那么什么是数据呢？从广义上讲，数据是可以记录、通信和能识别的符号，它通过有意义的组合来表达现实世界中某种实体(具体对象、事件、状态或活动)的特征。

在数据的定义中，必须注意两点：一是符号问题，用以表示数据的符号多种多样，它可以是简单的数字，也可以是声音、视频等；二是数据要用具体的载体(也称媒体)来记录和表示，数据的载体可以是多种多样的，例如纸张、磁带、磁盘等。数据只有通过一定的媒体表达后，才能进行存取、加工和传递。当然，数据用什么样的形式表达，也取决于不同的媒体。以多种媒体形式表示的信息成为多媒体信息。

从上面可以看出来，数据和信息的关系可以看作是原料和成品的关系，即信息是经过加工后的数据。

2.1.2.2 信息的属性

信息具有如下基本属性：

- **真伪性**：真实是信息的中心价值，不真实的信息价值可能为负。
- **层次性**：信息一般和管理层一样，可以分为战略层、策略层和执行层 3 个层次。
- **不完全性**：客观真实的全部信息是不可能得到的。我们需要正确滤去不重要的信息、失真的信息，抽象出有用的信息。
- **滞后性**：信息是数据加工的结果，因此信息必然落后于数据，加工需要时间。
- **扩压性**：信息和实物不同，它可以扩散也可以压缩。
- **分享性**：信息可以分享，这和物质不同，并且信息分享具有非零和性。

根据信息的来源，可将信息分为外部信息和内部信息；按照信息的用途又可以分为经营决策信息、管理决策信息和业务信息等；按信息的表示方式，则可以分为数字信息、文字信息、图像信息和语言信息等。

信息的分类还可以有多种方法，因而可以分为许多种类的信息，以下是比较常见的几类信息分类：定量信息和定性信息，文字信息和数字信息，确切信息和模糊信息，自然信息和社会信息，固定信息和变动信息，原始信息和派生信息，重要信息和次要信息，内部信息和外部信息以及格式化信息和非格式化信息等。

2.1.2.3 信息量和信息熵

首先，让我们看看什么是熵。热力学中的熵增原理是这样表述的：存在一个状态函数熵，只有不可逆过程才能使孤立系统的熵增加，而可逆过程不会改变孤立系统的熵。从中可以看出：熵及熵增是系统行为；这个系统是孤立系统；熵是统计性状态量，熵增是统计性过程量。在讨论信息熵表述时，应充分注意这些特征的存在，并且明确给定系统中发生的信息传播是不可逆过程。

我们把信息描述为信息熵，是状态量，其存在是绝对的；信息量是熵增，是过程量，是与信息传播行为有关的量，其存在是相对的。在考虑到系统性、统计性的基础上，我们认为：信息量是因具体信源和具体信宿范围决定的，描述信息潜在可能流动价值的统计量。

本说法符合熵增原理所要求的条件：“具体信源和信宿范围”构成孤立系统，信息量是系统行为而不仅仅是信源或信宿的单独行为，界定了信息量是统计量。此种表述还说明，信息量并不依赖具体的传播行为而存在，是对“具体信源和具体信宿”的某信息潜在的可能流动价值的评价，而不是针对已经实现了的信息流动。

由此，信息量实现了信息的度量。

2.1.3 信息与管理

管理职能主要包括计划、组织、领导、监督和控制。管理者进行这些工作都需要信息。编制计划时需要信息作依据，为了使计划切合实际，需要历史的和当前的各种数据，以便通过分析和研究，预测未来的变化趋势，制定出多种计划方案。组织、领导和监督的根据也是信息。影响计划完成的各种信息反馈到领导那里后，可采取各种办法予以控制，例如人员的行为信息、产品的质量信息、材料的库存信息、财务成本信息、生产进度信息、产量利润信息、收支平衡信息等都是控制的依据。总之，管理离不开信息。

1. 管理信息及其特征

管理是一个复杂有机的动态过程，其中包含的市场需求、生产过程、人员心理、主管意识、技术条件、原料供应等要素之间每时每刻的相互关联、排列顺序和质与量关系，无不变化多端，表达为各种不同的管理信息。所以，管理信息的组织结构千变万化。正是这种千变万化的管理信息所代表的物质运动决定着企业在经营变化、产品换代、技术更新、工艺改进、人员培训、管理体制的调整和市场销售等方面多种多样的管理形态，从而描绘出多维多层次的管理世界的“群体蓝图”显示出管理信息的特有功能。

如上所述，管理信息是整个管理世界物化运动的普遍属性，它表述了它所属的管理系统，在同任何其他管理系统相互作用和联系的过程中，以时间、效益、决策的形式所呈现的管理形态、结构及其运动过程。

根据以信息为依据的管理唯物论的基本原理，可以将管理信息的定义分解为以下几点：

- 管理信息是整个管理世界物化运动的普遍属性。这里的物化运动不限于企业内部，也不限于任何其他管理系统，它应包括管理者自身及其精神活动在内的各个方面。正是这种物化运动的普遍属性，才构成了以人为中心的管理信息系统。
- 管理信息存在于它所属的管理系统同任何其他管理系统的全面的相互作用中。
- 管理信息存在的这种特性是理解管理认识论的关键。
- 管理信息所表述的管理系统的形态、结构及其运动过程是建立在大量可供统计的随机事件的基础之上的。

- 管理信息既然是管理系统的(时间、效益和决策)形态、结构及其运动过程的表现,自然与时间、效益和决策有着确定的关系。实践证明:在复杂的管理中,时间、效益和决策是管理不可分离的属性,不同的时间、效益和决策之间是可以相互转化的。

- 由于管理信息与时间、效益的不可分离性,所以一切信息过程均为不可逆的过程,即在管理信息过程中时间和效益是不可能“倒流”的。

- “运动过程”一词说明管理信息不仅是管理系统在全面相互作用中的产物,而且是该作用过程本身。管理信息本身指的就是一种运动过程。

2.信息与信息收集

信息和其他资源一样也有生命周期。「从信息的获取、传输、加工、存储、维护、使用到退出的整个过程称为信息的生命周期。

信息收集的第一个问题是收集什么信息?这就是信息(数据)识别。

信息识别后进行信息采集。

信息采集方法和信息源有关。信息源有两大类,一是按地域分,一是按时域分。按地域分为内源(系统内),外源(系统外);按时域分为原始信息,加工信息等。

信息收集时注意信息表达方式。信息表达方式包括文字表达、数字表达、图表表达、多媒体表达等。

2.1.4 信息与决策

信息与决策的关系,可以概括为一句话:信息是决策的基础和依据,决策是对信息的判断和运用。

可以说,决策的过程,就是信息的输入、处理、输出的过程。决策的形成需要借助信息进行判断,决策的实施需要借助信息进行控制;决策执行终结后,需要凭借信息进行总结,为新的决策创造条件。决策的每一步骤都离不开信息,如同生产不能没有原料一样,决策离开信息就会成为无源之水,无本之木。

信息的价值在于它能向物质转化。信息、能源、材料是3大资源,它们可以互相转化。利用信息技术可以增加产品,节省能源。如利用信息技术选择先进的材料,开采和生产廉价优质的材料,在生产中合理下料,这些都是信息转化为物质的例子。

决策的类型有多种分法,如按不确定性程度可分为确定性决策,风险(概率)性决策和不确定性决策;若按结构化程度划分,可分为结构化决策、半结构化决策和非结构化决策;若按管理层次划分,可分为战略性决策、管理性决策和业务性决策。不同层次的决策需要的信息是不同的。战略性决策需要的信息是面向市场和产品的信息,不确定性和结构性较差;管理性决策需要的信息以面向企业内部为主,不确定性和结构性较战略性决策要好,而业务性决策需要的信息结构性和确定性都最好。

2.2 系统与系统工程

2.2.1 系统的概念

在日常生活中我们处处使用系统这个概念,如经济领域的工业系统,商业系统,农业系统;自然界的水利系统,气象系统,生态系统;军事领域的作战系统,后勤保障系统;日常生活中的交通系统,文教系统。总之,无论什么人,都处于系统之中,人本身又是系统的一份子。半个多世纪以来,“系统”吸引了许多领域的专家研究和应用,并逐步形成了一门新兴学科—系统科学。那么究竟什么是系统?系统有众多的定义,总的意思是:系统就是由多个元素有机地结合在一起,执行特定的功能以达到特定目标的集合体。说得更详细一些,系统是由各元素或子系统组成的;各元素之间是互相作用或互相制约的;系统是有目标的;系统和环境有关,要适应环境的变化;系统有强烈的整体性,单元要服从整体。系统既然是由元素组成的,则至少应有两个元素,这是最小的系统。大的系统元素很多,有成千上万个元素。如学校系统,其元素有教师、学生、行政人员、教室、图书馆、食堂、实验室……这些元素之间相互联系,相互作用,有条不紊地为培养人才,探索研究而有机地运行着。工厂也是一个系统,其元素由工人、技术人员、管理人员以及厂房、机器设备等组成,为了生产合格的一定数量产品而有条不紊地运行着。城市是一个更大的系统,它由交通系统、商业系统、通讯系统等组成。

1.一般系统论概述

1945年贝塔朗菲(von Bertalanffy)提出一般系统论的概念,他定义系统为“相互作用的多元素的复合体”。该定义有3个基本的重要项,即系统具有多元性、相关性以及整体性。进一步,对系统概括为整体性、关联性、动态性、有序性和终极(目的)性。整体性是系统思想的核心观点,整体特性不等于局部特性之和,即 $W \neq \sum P_i$

其中 W 为整体特性; P_i 为第 i 个元素的特性 W 可能大于 $\sum P_i$, 也可能小于 $\sum P_i$ 。

2.系统的分类

对系统有各种各样的分法,一般的分类方法有5种。

- **按复杂程度分：**可分为 3 类 9 等(见图 2.1)。最简单的是框架，如房屋、桥梁是静态系统；其次是钟表系统，虽然能动，但仍然是静态系统；控制机械能自动调节，如电冰箱，这是控制系统；细胞有生命，能自繁殖，比物理系统要高级和复杂；植物是细胞群体组成的系统，复杂性比细胞要高；动物能自学，比植物更高级；人类能使用语言，能由大脑存储信息、控制行动；人类社会是极为复杂的社会系统；宇宙系统比地球更为复杂。



图 2.1 系统按复杂程度分类

- **按系统抽象程度分：**可分为 3 类：概念系统、逻辑系统、实体系统，如图 2.2 所示。

实体系统是由实际上可见的一些物质组成的系统，也可以称为物理系统，如一个实际存在的计算机系统；逻辑系统只是说明从原理上可行的系统，但并不确定具体的实体性质，例如设计一个管理信息系统的逻辑模型，它只需要提出所需计算机的内存、速度以及性能要求，终端个数等，而没有确定必须选择哪种型号的计算机和终端等，因为从逻辑上认为能满足要求的任何计算机型号都可能实现这样的系统；概念系统是由概念、原理、原则、方法、制度、程序等非物质实体组成的系统，如教育系统，治安系统，管理系统等。

- **按系统的功能分：**可分为经济系统，军事系统，电力系统，铁路运输系统等。

- **按系统和外界关系分：**可分为封闭系统和开放系统，封闭系统是独立于环境的系统，开放系统是指和外界不能分开的系统。

- **按系统内部结构、形态分：**可分为开环系统，闭环系统；静态系统，动态系统；线性系统，非线性系统；确定性系统，随机系统等。开环系统是指系统输出不对输入产生影响，闭环系统是指系统输出反过来作为输入。静态系统是指系统状态变量不随时间而变化的系统，如物理系统中的框架；动态系统是指系统状态变量随时间变化的系统，社会经济系统都是动态系统。确定性系统是指系统状态变量都是确定的，一组惟一的输入可以得到一组惟一的输出；而随机系统的状态变量具有随机的性质，只要有一个变量是随机的，系统就是一个不确定系统，社会经济系统一般都是随机系统。

当然，还可能有其他分类方法，比如按系统的形态还可以分为自然系统和人工系统等。

3.系统的特性

系统具有如下特性：

- **整体性：**组成系统的各元素不是简单地集合在一起，而是有机地组成一个整体，每个元素都要服从整体，追求整体最优。这就是所谓全局观点。

- **层次性：**系统是有层次的。系统中的每个元素仍可以看作是一个系统，例如社会经济系统下属工业系统、农业系统、交通系统等，而工业系统又可分为机械工业系统、冶金工业系统……还可继续分下去。

- **相关性：**系统内各元素(或各子系统)是有联系和相互作用的。

- **目的性：**任何一个系统都有一定的目的或目标。

- **环境适应性：**任何系统都处于特定的环境中。

在系统中，称有意义的元素为实体(Entity)，表示实体特征的称为属性(Attribute)，实体在特定时间内的运动叫活动(Activity)，描述系统在任何时间的必要变量叫状态变量或简称状态(State)，表示状态变化的出现称事件(Events)。



图 2.2 系统按抽象程度分类

2.2.2 系统与环境

一个系统之外一切与其相关联的事物构成的集合，称为系统的环境。系统从环境中产生，又在环境中运行、延续和演化。系统只有涌现出特定的整体性，才能适应环境。如果环境改变，系统必须涌现出新的整体性，才能达成新的依存关系。一般来说，环境中的组成元素之间联系较弱，系统性不够强，这为系统的趋利避害、保护和发展自己提供了可能性。但也要警惕系统的环境也有很强的系统性，这有可能影响系统的生存和发展。

1.系统的能控与能观

系统的能控与能观也可称为可控与可观，这在控制系统中是很重要的概念。能控的意思是，在一个有限的时间间隔内，比如从 t_0 到 t_1 的时间间隔内，可以用一个无约束的控制量(标量或向量)，使得系统的状态由 $x(t_0)$ 转移到 $x(t_1)$ ，则该系统在时间 t_0 是能控的。所谓能观性是指系统状态 $x(t_0)$ 可以通过一个有限的时间间隔，由输出值的观测中确定，那么该系统在 t_0 时刻是能观的。

2.系统的接口与藕合

系统的接口是指系统与环境的结合点或者是子系统之间的联接点，在信息系统中接口的作用十分重

要。系统的藕合就是系统与系统之间联系。若某些系统之间不易藕合，可用缓冲器(中间件)与之联系，如生产子系统与订购子系统之间的仓库、计算机的 CPU 和外部设备之间的缓冲区等。

3. 系统的自组织性

对于一个由大量子系统所组成的系统来说，在一定的条件下，它的子系统之间自发地通过非线性的相互作用就能产生协同现象和相干效应，并形成自己一定的组织功能和时空结构，使系统表现出新的有序状态，常把这个特性叫系统的自组织性。

系统的自组织性在某种意义上意味着自足性、自律性和自我生成性，它强调要从整体系统的相互作用来考察事物。

2. 2. 3 系统工程与系统方法

系统工程的概念源于二战后大型军用系统的开发与建设，如原子弹、航天飞机、卫星通信等。

系统工程是系统科学在工程技术的应用，其核心是组织管理与决策。系统工程来源于社会实践，也随着社会的发展而发展，我国正处于民族复兴时期，经历着空前复杂的社会实践，迫切需要工程学的理论、方法与技术。企业信息化、政府信息化、社会信息化是极其伟大和艰巨的实践，没有一套科学的组织和管理方法与技术，这些复杂的大工程就难以成功。

所谓系统工程，有两层意思：作为学科，它是以研究大规模复杂系统为对象的一门新兴边缘学科；作为工程，它又是一门工程技术，具有和一般工程技术相同的特征，但又具有本身的特点，它并不研究特定的工程物质对象，而是物质系统、概念系统。我国著名科学家钱学森教授指出(1978 年)：“系统工程是组织管理系统的规划、研究、设计、制造、试验和使用的科学方法，是一种对所有系统都具有普遍意义的科学方法。”“系统工程是一门组织管理的技术。”国际学术界还把系统工程和系统分析作为同义词来理解。系统方法或称系统方法论，是研究工程学的思考问题和处理问题的方法论。系统方法的要点是：系统的思想，数学的方法，计算机的技术。系统的思想即把研究对象作为整体来考虑，着眼于整体最优运行；数学的方法就是用定量技术研究系统，通过建立系统的数学模型和运行模型，将得到的结果进行分析，再用到原来的系统中；计算机技术是求解数学模型的工具，在系统的数学模型上进行模拟，以实现系统的优化。美国学者霍尔(H. Hall)最先提出系统方法的“三维结构体系”(见图 2.3)，这是系统工程方法论的基础。

三维结构由时间维、逻辑维和知识维组成一个立体结构。

时间维将系统分为 7 个时间阶段，包括：

(1)规划阶段：对系统进行定义、确定目标、制定开发规划和策略。

(2)制定方案：提出具体方案。

(3)研制阶段：实际系统的研制方案。

(4)试运行阶段：将项目投入试运行。

(5)安装调试阶段：将整个系统安装好，拟定运行维护规范和运行计划。

(6)运行阶段：按预期目标运行系统。

(7)更新阶段：改进旧系统，使之成为新系统。

逻辑维是指系统开发过程中每个阶段所经历的步骤。

(1)问题确定：通过收集数据弄清问题的症结。

(2)目标确定：确定目标及评价标准。

(3)系统综合：研究达到目标的各种策略。

(4)系统分析：通过建模，推断可供选择的各种方案的可能结果。

(5)最优化：求出最优系统方案。

(6)系统决策：选出最优方案。

(7)计划实施：将优选方案付诸实施。

知识维是指完成各阶段、各步骤所需知识。

2.3 信息系统工程

信息系统工程是 1980 年初在我国建立经济信息系统时提出的。信息系统工程属于系统工程范畴。我们一般所指的信息系统工程认为是以计算机、网络、数据库、软件等信息技术与产品为基本构件的系统工程。因此，适用于工程学的规范、方法、经验和管理都可以有选择地用于信息系统工程中。

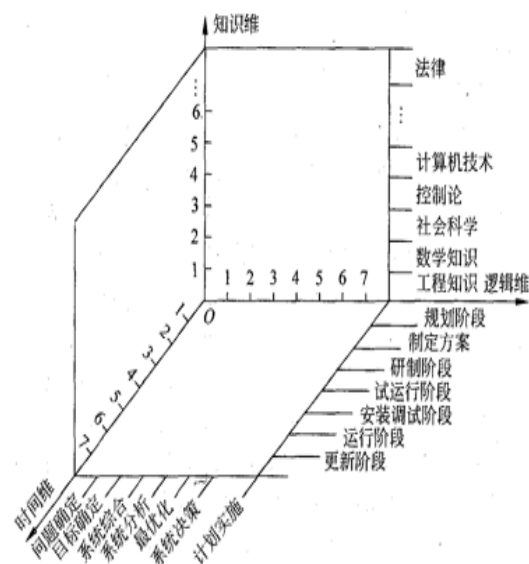


图 2.3 系统的三维结构

1. 信息系统的特征

在信息化的过程中，经常用到的名词为信息系统、系统工程、信息工程(Information Engineering)等，信息系统工程将三者结合起来。信息系统当然是一个系统工程，信息工程更直接地称信息系统为系统工程，所以我们不必在名词上纠缠不清，如果没有特殊说明，本书所讲的信息系统或信息工程就是指信息系统工程。

信息工程和一般的系统工程有所不同，约翰·柯林斯(John Collins)认为“信息工程作为一个学科要比软件工程更为广泛，它包括了为建立基于当代数据库系统的计算机化企业所必需的所有相关的学科”。从该定义可见信息工程的特点为：基于数据库系统；目标是建设计算机化的企业管理系统；有多种技术和学科的综合。时隔 20 多年以后，信息技术和信息系统的发展远非昔日相比了，但信息系统工程的基本特征没有变，即它是一个以生产信息、辅助企业管理和决策的人/机社会系统工程；它不仅以数据库系统为中心，而且已建立在计算机和网络平台上；它不仅包含技术因素，而且还包含了管理、组织和人的要素。

2. 信息系统工程的发展

信息系统工程的发展大致可分 3 个大的阶段。

(1)数据处理时代(20 世纪 60 年代至 80 年代初):在这一阶段中经历了集中式数据处理与操作，以数据库系统为基础的 MIS 和 DSS 以及 20 世纪 70 年代出现的以小型机为中央处理机的信息系统工程。

(2)微机时代(20 世纪 70 年代末 80 年代初至 90 年代中):在这个时代中微机与传统数据处理并存，信息系统工程既有采取分布式(局域网式)的，也有传统方式的。

(3)网络时代(20 世纪 90 年代中至现在):信息系统工程发生了很大的变化，局域网(LAN)，内联网(Intranet)、外联网(Extranet)以及因特网(Internet)等将企业广泛联系起来，信息系统工程无论规模还是复杂性都有了质的飞跃，ERP, SCM, CRM 等名称接踵而至。

3. 信息系统工程的复杂性及解决方案

关于如何衡量信息系统工程的管理水平，目前国内尚没有标准。1991 年美国著名的卡内基-梅隆大学软件工程研究所(SEI)针对软件工程的工程管理能力与水平进行了充分研究，提出了 5 级管理能力的模式，我们认为该模式对评价信息系统工程的管理水平具有参考价值。

SEI 的 5 级管理能力模式如下：

(1)临时凑合阶段：工作无正式计划，作业进度经常被更改，任务计划、预算、功能、质量都不可预测，开发机构的整体组织非常混乱。系统的性能、水平依个人能力而定。

(2)简单模仿阶段：开发方开始采用基本的项目管理方法与原理；项目从规划到运行都有明确的计划；这些计划是通过模仿以前成功的项目开发的例子制定的，有可能通过模仿在本次开发中成功。

(3)完成定义阶段：与项目有关的整体机构的作业进度规格化、标准化，由此达到持续稳定的技术水平与管理能力。这种工程进度管理能力要求把与开发项目有关的活动、作用和责任充分告知所有的开发者，并使之充分理解。

(4)管理阶段：这是理想的项目管理阶段。表现在开发者的工程管理能力不断强化，通过可靠的组织与计划保障，能及早发现可能影响系统功能与性能的缺陷，使系统的性能与可靠性不断改进与提高。

(5)最佳化阶段：这一阶段是理想的项目管理阶段。其特点表现在开发者的工程管理能力不断强化，通过可靠的组织与计划保障，能及早发现项目中可能影响系统功能与性能的缺陷，系统的关键指标在工程的实施过程中得到全面保证与提高。

第 3 章 结构化分析与设计方法

3. 1 方法概述

3.1.1 系统开发生命周期

正如事物有其产生、发展、成熟、消亡的生长过程一样，信息系统也都有其产生发展、消亡的过程。当旧系统不再适应企业发展的需要时，适应企业发展的新系统将代替旧系统，这个周期就被称作系统的生命周期(System Life Cycle, SLC)。

为了有效地进行系统的开发和管理，根据系统生命周期的概念，一般可以将信息系统的开发分成 5 个阶段，即总体规划阶段、系统分析阶段、系统设计阶段、系统实施阶段、系统运行和评价阶段。每个阶段都有其明确的任务，任务完成后都将交付给下一阶段一定规格的文档，作为下一阶段开发的依据。这种开

发过程，在直观上就像一级一级的瀑布，所以系统开发生命周期也称为“瀑布模型”。

如图 3.1 所示，每个阶段完成后都要向下一阶段交付一定的文档。如总体规划阶段向系统分析阶段提交可行性分析报告，系统分析阶段根据可行性分析报告，进一步对系统的功能进行分析和逻辑设计，并提交系统方案说明书。系统设计阶段又称为物理设计阶段。在此阶段，根据系统逻辑方案进行物理设计，并提交系统设计说明书。系统实施阶段是根据系统设计进行程序实现和测试、安装、试运行、系统转化等工作。由于人们对问题的认识有一个深化反复的过程，所以有时会出现一定的反复。

有调查数据显示，系统生命周期中的各个阶段的工作量如图 3.2 所示。可以看出系统实施阶段的工作量约占了总工作量的一半，因为在系统实施阶段，企业不仅仅要进行程序的设计，为新系统准备大量数据等，还要注意新老系统的交替可能给用户带来一系列心理和实际上的变化。图 3.2 还说明，编程的工作量只占系统开发很小的一部分，系统开发不仅仅需要编程人员，还需要管理层、最终用户、系统分析和设计人员互相配合，共同承担。

各个阶段各类人员的投入情况如图 3.3 所示。

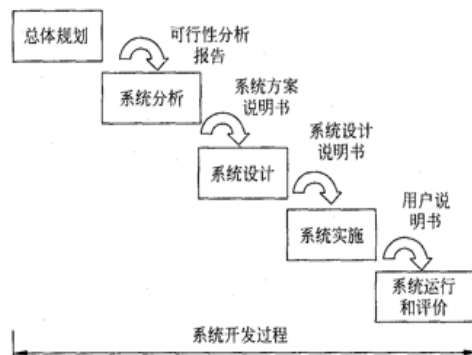


图 3.1 系统开发流程

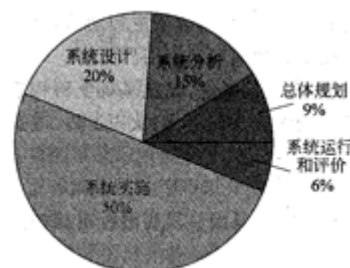


图 3.2 各阶段工作量

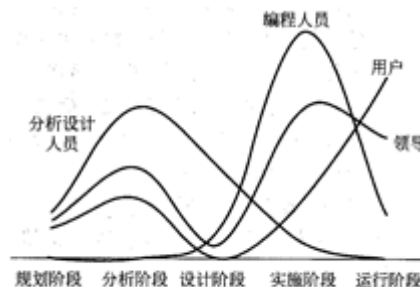


图 3.3 各阶段人员投入

3.1.2 结构化方法的基本思想

结构化方法是“结构化分析” (Structured Analysis, SA) 和“结构化设计” (Structured Design, SD) 的总称，结构化方法是目前最成熟、应用最广泛的信息系统开发方法之一，它的优点是有一套严格的开发程序，各开发阶段都要求有完整的文档记录，国内外已有许多成功开发的例子。结构化开发方法以信息系统生命周期模型为基础，所以也称为“结构化生命周期法”。

3.1.2.1 结构化分析

1. 结构化系统分析思想

结构化分析方法是美国 Yourdon 公司在 20 世纪 70 年代提出的，其基本思想是将系统开发看成工程项目，有计划、有步骤地进行工作，是一种应用很广的开发方法，适用于分析大型信息系统。

结构化分析方法采用“自顶向下，逐层分解”的开发策略。按照这种策略，再复杂的系统也可以有条不紊地进行，只要将复杂的系统适当分层，每层的复杂程度即可降低，这就是结构化分析的特点。

2. 结构化分析方法的内容

结构化分析之后获得的文档是系统分析报告，系统分析报告主要是由下面几个部分组成的：组织结构及其分析，现行业务流程及其分析，现有数据和数据流程及其分析，新系统的初步方案和补充材料，如开发计划等。

3. 结构化分析方法的特点

结构化分析方法有以下特点：

- 结构化分析方法简单，易于掌握和使用。
- 结构化分析方法将分析的结果用图形表示，如业务流程图，数据流程图等，这些图形都有一套标准图符组成，从而将分析结果简明易懂地展示在用户面前。
- 结构化分析的实施步骤是先分析现实环境中已存在的系统，在此基础上再构思即将开发的目标系统，从而大大降低了问题的复杂程度，符合人们认识世界、改造世界的一般规律。

4. 结构化分析方法的局限

结构化分析方法是一种行之有效的方法，但也有一定的局限性。局限性可以概括成以下几方面：

- 结构化分析方法要求对系统有完整确切的需求定义，而实际上这是非常困难的。
- 文档资料数量大。需要书写大量文档，随着分析的深入，这套文档需要及时更新，即使在工具的辅助下，仍有一定的难度。
- 人机界面表达能力差。信息系统是一个人机交互系统。对于许多交互式系统而言，用户十分关心如何使用这个系统，但是结构化分析方法的人机交互表达能力不强。
- 结构化分析方法描述的模型仅仅是书面的，只能供人们阅读和讨论，但不能试用从而及时地获得用户的反馈信息。

3.1.2.2 结构化设计

结构化设计方法也是目前使用最广的一种系统设计和程序设计方法，它是由 IBM 公司提出的，此方法适合于软件系统的总体设计。结构化设计是在结构化分析报告的基础上，进行新系统的设计。结构化设计的主要目的就是为系统实现制定蓝图。

结构化设计方法的基本思想是将系统设计成由相对独立、功能单一的模块组成的结构。

结构化设计方法内容主要包括：

- **系统总体结构：**包括总体结构图、子系统结构图和计算机流程图；
- **系统设备配置：**包括计算机系统配置图，设备在各生产岗位的分布图、主机、网络和终端连接图等；
- **系统分类编码方案：**分类方案、编码和校对方式；
- **数据库结构图：**包括 DB 的结构(主要指表与表之间的结构)，表内部结构(字段，域)和数据字典等；
- **I/O 设计方案；**
- **HIPO 图：**包括层次结构图和 IPO 图；
- **处理逻辑和存储方案。**

3.1.3 系统开发的阶段划分

1.总体规划阶段

总体规划阶段是信息系统的起始阶段。以计算机为主要手段的信息系统是其所在企业的管理系统的组成部分，它必须服务于企业，服从企业的整体目标和管理决策活动的需要。

总体规划的作用可以分成以下几点：

- **指明组织中建立信息系统的范围和目标；**
- **指导信息系统开发；**
- **合理分配和利用各种资源；**
- **通过规划过程找出企业中存在的问题。**

信息系统的建设是投资多、耗时长的一项社会工程，规划作为信息系统建设的基础，如果考虑不周，会使企业的运作受到直接或者间接的损失。

一个比较完整的总体规划，它的内容应该包括信息系统的开发范围和目标；信息系统开发的约束条件；组织及其管理的现状、问题及解决方案；信息系统的总体结构；信息系统建设计划；相关的信息技术发展预测等。

2.系统分析阶段、

系统分析阶段的目标是为系统设计阶段提供系统的逻辑模型，系统设计阶段再根据这个逻辑模型进行物理方案的设计。

系统分析阶段以管理分析为前提，规划未来「信息系统框架，是组织发展与信息系统建设的结合点，是管理人员与技术人员的结合点。系统分析阶段的主要任务就是将在系统详细调查中所得到的文档资料集中在一起，对组织内部整体管理状况和信息处理过程进行分析。系统分析在整个系统开发过程中，是要解决“做什么”的问题，把要解决哪些问题、满足用户哪些具体的信息需求进行调查、分析清楚，从逻辑上，或者说从信息处理的功能需求上提出系统的方案。内容包括组织结构及功能分析、业务流程分析、数据及数据流程分析、用户需求分析、新系统方案等。

3.系统设计阶段

系统设计阶段则是根据系统分析的结果，设计一套与改进后的管理体制及管理手段相适应的新的信息系统，为系统实施阶段的程序设计、调试提供依据。系统设计阶段的主要任务就是在各种技术和实施方法中权衡利弊，精心设计，合理地使用各种资源，最终勾画出新系统的详细设计方案。系统设计阶段的主要内容包括新系统总体结构设计、代码设计、数据库设计、输入/输出设计、处理流程及模块功能设计、安全控制点设计等。系统设计的结果是一系列的系统设计文件，这些文件是物理地实现一个信息系统的重要基础。

4. 系统实施阶段

系统实施阶段是将系统设计阶段的结果在计算机上实现。将原来文字的设计方案转换成实际的可执行的软件系统。系统实施作为系统的物理实现阶段，对于系统的质量、可靠性和可维护性等性能都有着十分重要的影响。

系统实施工作必须在系统分析和系统设计工作完成之后，严格按照系统开发文档进行。系统开发者只有通过系统分析和系统设计文档，对系统目标、系统代码设计、输入/输出设计等有全面的了解，才可能进行系统的实施工作。还应该注意系统开发人员不仅要了解本人所承担的部分，还要了解系统总体结构、彼此接口、数据交换等相互联系的内容，以保证在系统实施工作中局部分散的实施与系统整体一致。

5.系统运行和评价阶段

信息系统在实施阶段结束之后,就进入到系统运行和评价阶段。一般来说,信息系统的寿命短则4—5年,长则10年以上,但是只要系统正式运行,系统的维护工作就将伴随着信息系统的整个生命周期。维护工作主要是对应用系统、数据、代码,以及硬件和网络设备进行维护。针对软件维护的不同性质,系统维护可以划分成4种类型:纠错性维护,适应性维护,完善性维护和预防性维护。统计结果表明一般纠错性维护(诊断,修正系统中遗留的错误)占21%,适应性维护(使系统能够适应环境的变化)占25%,完善性维护(扩充原系统,提高性能等)占50%,而预防性维护仅占4%。可见,完善性维护占据了一半以上的工作。

当系统运行一段时间之后,随着对系统应用的不断深入和应用环境的发展变化,有必要对系统进行评价。系统评价的目的是检查系统是否达到预期目的、是否满足用户要求和系统的各种资源利用效率,提出系统改进和发展方向。系统的评价主要针对两个方面,即系统的性能指标和经济指标,具体地说,就是对系统运行效率、系统运行及维护的费用、系统可靠性、系统的输入输出、系统内信息反馈情况和系统运行平台等进行评价。

3.1.4 系统开发中的管理

3.1.4.1 项目管理

项目管理是通过项目经理和项目组织,运用系统理论和方法对项目及其资源进行计划、组织、协调和控制,旨在实现项目的特定目标的管理方法体系。项目管理的主要任务是制定项目实施计划,对人员进行组织、分工,并按照计划进度和成本管理、风险管理、质量管理的要求,进行系统开发并最终实现预期的目标。

企业信息系统是一项系统工程,因为信息系统的建设符合项目的几个特点:首先信息系统的建设是一次性的任务,有一定的任务范围和质量要求,有时间或进度的要求,有经费或资源的限制;其次信息系统具有生命周期,这与项目具有生命周期一样。所以可以用项目管理的思想和方法来指导信息系统的建设。

在具体实施过程中,项目管理一般包含了以下几个方面的内容。

1.任务划分

任务划分是把整个开发工作定义成一组任务的集合,这组任务又可以进一步划分成若干个子任务,进而形成具有层次结构的任务群。这样做可以将任务责任落实到人,有利于责、权、利相结合的监督和管理方式,另一方面有利于资金的分配,保证资金的有效控制。

序号	任务名	开始时间	结束时间	需要周数	Q2 02												Q3 02												Q4 02												Q1 03											
1	系统规划	2002-03-21	2002-04-24	5																																																
2	系统分析	2002-04-22	2002-06-28	10																																																
3	系统设计	2002-06-20	2002-08-28	10																																																
4	系统实施	2002-08-20	2002-12-02	15																																																
5	系统测试	2002-12-02	2003-01-24	8																																																
6	系统转换	2003-01-24	2003-02-27	5																																																

图 3.4 甘特图

2.计划安排

依据划分完毕的任务即可制定出整个开发及项目管理计划,并产生完成任务的计划表。开发计划可以划分成系统规划,系统分析,系统设计,系统实施,系统测试和系统转换6大活动。当这些计划制定出来之后,就可以画出任务完成的计划表,表明任务的开始时间、结束时间,以及任务之间的相互依赖程度,以此作为实施监控的基础。很多开发公司都使用甘特图来表示计划表,这种表示方法很直观,例如项目经理绘制了如图3.4所示的甘特图(图中Q202代表2002年的第二季度,Q3,Q4依此类推)。

还可以采用一种更为精确的控制方法来对计划进行安排和管理,那就是网络控制(PERT)的方式。感兴趣的读者,可以参考运筹管理书籍中图与网络的内容。

3.经费管理

经费管理在整个开发项目管理中处于重要的地位。项目经理可以运用经济杠杆来控制整个开发工作。经费的有效运用可以起到事半功倍的效果,反之,也许花了很多钱,开发工作却毫无进展。在项目管理中,赋予任务负责人一定职责的同时,还要赋予他一定的经费支配权,同时要对其进行适当的控制。

4.审计控制

按照所采用的开发方法,应针对每一类开发人员制定出工作过程中的责任、义务、完成任务的质量标准等,按照计划对每项任务进行审计。分析执行任务计划表和经费的变化情况,确定需要调整、变化的部分;并根据任务完成计划表和审计结果,掌握项目进展情况,及时处理开发过程中出现的问题,及时修正开发工作中出现的偏差,保证系统开发工作的顺利进行。通过审计和控制,对于系统开发中出现的变化情况,项目经理要及时与用户和主管部门联系,取得他们的理解和支持。

5. 风险管理

任何一个系统开发都存在一定的风险,企业的条件无论多么优越,所做的准备无论多么充分,实施的

风险仍然存在。在系统实施周期中，各种影响因素随时都可能发生变化。如何有效地管理和控制风险是保证系统实施成功的重要环节之一。对于一个项目而言，风险存在于项目的全过程，包括项目规划、项目预准备、实施过程和系统运行。

归纳起来，风险主要有以下几方面：

- 缺乏规划或规划不合理。
- 项目预准备不充分，表现为硬件选型及软件选择错误、咨询合作伙伴力量不足。
- 实施过程控制不严格，阶段成果未达标。
- 设计流程缺乏有效的控制环节。
- 实施效果未做评估或评估不合理。
- 系统安全设计不完善，存在系统被非法入侵的隐患。
- 灾难防范措施不当或不完整，容易造成系统崩溃。

风险管理的主要任务就是对上述的问题采取事先预防的处理方法，以便尽可能地提高系统开发的成功率和开发进程。

6.质量保证

质量管理应贯穿于整个项目始终。在项目规划阶段，就应该建立系统质量的度量模型和相应的机制，对项目质量提出总体的要求；在系统分析和设计阶段应对质量管理不断细化，按自顶向下的方式将总体要求划分成若干易于考核和度量的质量单元。

在系统的各个开发阶段，项目经理或者质量管理小组组长都应该对各种质量保证进行记录，并形成报告，对阶段成果进行技术评审，形成系统文档并对文档进行保管和控制。质量管理小组还要负责制定和执行系统的测试策略和测试计划。

3. 1. 4. 2 人员组成与管理

信息系统项目是智力密集、劳动密集型的项目，受人力资源影响很大，项目成员的结构、责任心、能力和稳定性对信息系统项目的质量以及是否成功有很大的影响。如何将系统开发的人员组织起来，以发挥最大的工作效率，是系统开发的一个重要问题。

1.人员的构成

一个大的项目开发团队可以分为总体组、软件开发组、硬件网络组和测试组等若干个项目小组；更大型的项目可以分成若干子项目小组，子项目小组下面再可分若干分项目小组。

在建立项目小组时应注意以下几个原则：尽早落实责任，以明确每个成员之间的责任分工；知人善任，将每个人的专长尽可能发挥好；减少接口，在开发过程中，人与人之间的联系是必不可少的，因此要有合理的人员分工和良好的组织结构，以减少不必要的生产率的损失。

一般来说，项目小组的规模应该比较小，人数在 2-8 名左右为宜。如果项目小组规模较小且存在时间在一年以内的，项目小组的成员可以是活动负责人制，活动负责人可以负责活动的日常管理工作，也可以负责技术方面的工作。如果项目属于大中型规模，建设时间在一年以上，这时的程序开发项目小组可以是：一个高级系统开发人员带两个中级系统开发人员，每个中级开发人员带两个初级开发人员；而测试项目小组构成是：一名高级系统测试员带两名中级系统测试员，每个中级系统测试人员带两个初级系统测试员。

2.组织形式

可以采取的组织形式有下面 3 种。

- **共同工作小组：**共同工组小组中成员地位平等，成员共同负责信息系统开发，成员互相检查监督。共同工作小组的优点是有民主气氛，容易实现相互合作。缺点是职责不明，无专人负责整体工作。这种组织形式比较适合于小规模的信息系统项目。

- **主管负责制：**在已往的主管负责制中，主管人员主要负责项目的管理工作和关键部分的开发工作，普通的技术人员主要是完成各自的系统开发工作。改进的主管负责制中，主管只负责项目的管理和协调工作，不再担当开发的具体工作。主管负责制的优点是主管可以集中精力组织利用各类资源，缺点是主管必须对项目的开发拥有足够的控制。

- **主管负责下的专业分工制：**主管负责项目的管理和协调工作，普通技术人员按专业分配各自工作。主管负责制下的专业分工制可以充分发挥技术人员的专业特长，但是系统的分析及设计必须十分细致和注重系统的整合性。这种组织形式比较适合于较大规模的信息系统项目。

3.对人员的选择

信息系统开发人员要求有良好的职业道德，丰富的开发经验，知识面广，必要的观察和分析问题的能力，与人合作的能力。选择开发人员时要考虑开发人员对环境的适应能力，由于我国缺乏信息人才，尤其要注意人员流动性。

3.1.4.3 系统开发中全面质量管理

之所以对信息系统采取全面的质量控制，是因为在信息系统生命周期的各个阶段，对上一阶段的理解和本阶段的工作都存在这样那样的问题。并且，根据一些软件公司的统计资料，在后期引入一个变化比在早期引入相同的变动所需付出的代价高两三个数量级。因此要从信息系统开发一开始就进行质量控制，以便尽早发现错误，及时更正。

为了在信息系统的建设过程中实施全面的质量控制，主要采取下述措施：

- **实行工程化的开发方法：**信息系统开发方法一词的广义理解是“探索复杂系统开发过程的秩序”，狭义理解是“一组为信息系统开发起工具作用的规程”。规程由一系列的活动组成，形成方法体系，开发组的每个人都要遵守工程规范。

- **实行阶段性冻结与改动控制：**信息系统具有生命周期，一个大的项目可分成若干阶段，每个阶段都有自己的任务和成果。实行阶段性的冻结，便于管理和控制工程进度，另一方面可以增强开发人员和用户的信心。另外，在每个阶段末，对部分成果进行冻结，可以作为下个阶段开发的基础。冻结之后，不是不能对文档进行修改，而是如果要对文档进行修改的话，要经过一定的审批程序。

- **进行原型演化：**在每个阶段的后期，快速建立反映该阶段成果的原型系统，利用原型系统与用户交互及时得到反馈信息，验证该阶段的成果并及时纠正错误，这个技术就称为“原型演化”。原型演化技术要有先进的 CASE 工具的支持。

3.2 总体规划

3.2.1. 总体规划概述

总体规划是信息系统生命周期的第一阶段，是系统开发过程的第一步。由于信息系统的建设是一项耗资巨大、历时很长、技术复杂且又内外交叉的工程，所以在系统开发的初期就必须做好总体规划。

总体规划阶段的主要目标就是制定出信息系统的长期发展方案，决定信息系统在整个生命周期内的发展方向、规模和发展进程。这样做能为以后的系统分析和设计打好基础。

这个阶段的主要任务是：

- **制定信息系统的发展战略。**主要是使信息系统的战略与整个组织的战略和目标协调一致。
- **确定组织的主要信息需求，形成信息系统的总体结构方案，安排项目开发计划。**
- **制定系统建设的资源分配计划，**即制定为实现开发计划而需要的硬软件资源、数据通信设备、人员、技术、服务和资金等计划，提出整个系统的建设概算。

3.2.1.1 总体规划主要步骤

进行信息系统的总体规划一般包括这样几个阶段：

(1) **对当前系统进行初步的调查。**系统分析员从各级干部、相似的企业和本企业内部收集各种信息，站在“高层”观察组织的现状，分析系统的运行状况。初步调查主要由两部分构成：

- **一般调查。**一般调查包括组织的概括，企业的目标，现行系统运行情况，简单历史，企业的产品，产量，利税，体制及改革情况，人员基本情况，面临的问题，企业的中长期计划以及主要困难等，使系统分析员对企业有一个初步轮廓。

- **信息需求初步调查。**信息需求初步调查是整个初步调查的主要内容。通过调查组织系统的工作职责及活动以了解各职能机构所要处理的数据，估计各机构发生的数据量及频度。信息需求初步调查还应调查环境信息，包括内部环境和外部环境的信息。

(2) **分析和确定系统目标。**这实际上可以由总经理和信息系统开发的领导小组确定，应包括服务的质量和范围、政策、组织以及人员等。它不仅包括信息系统的目标，而且应有整个企业的目标。

(3) **分析子系统的组成以及基本功能。**从上到下对系统进行划分，并且详细说明各个子系统应该实现的功能。

(4) **拟定系统的实施方案。**可以对子系统的优先级进行设定，以便确定子系统的开发顺序。

(5) **进行系统的可行性研究。**

(6) **编写可行性报告。**

3.2.1.2 总体规划方法

用于信息系统规划的方法很多，主要是关键成功因素法(critical success factors, CSF), 战略目标集转化法(strategy set transformation, SST)和企业系统规划法(business system planning, BSP)。现简单介绍前面两种方法，第三种方法在本书的第 5 章中介绍。

1.关键成功因素法(CSF)

关键成功因素法是1970年哈佛大学教授William Zani提出的。主要是通过分析找到影响组织成功的关键因素，围绕关键成功因素确定组织对于信息系统的需求，根据信息系统的需求进行信息系统规划。

它包含以下几个步骤：

- (1) 了解企业目标。
- (2) 识别关键成功因素。
- (3) 识别性能的指标和标准。
- (4) 识别测量性能的数据。

例如，学校有一个目标是成为国际一流大学，可以用树枝图画出影响它的各种因素，以及影响这些因素的子因素，如图3.5所示。

其中，教学成果是影响国际一流大学的因素，而教学环境、课程水平、学生质量是影响教学成果的子因素。同样，学术水平也是关键成功因素，与之有关的科研力量、科研成果是影响学术水平的子因素。也可以用如图3.6的方式来表示关键成功因素。

从图3.6可以看出，国际一流大学是组织目标，而教学成果、学术水平是影响这个组织目标的子目标。改善教学环境、提高课程水平和提高学生质量是教学成果的成功因素。从而可确定各个性能指标，以及各个测量性能的数据，一直到产生数据字典。

关键成功因素法简单易用，且突出重点，从重要需求引发规划，但是容易忽视次要问题；总体规划受成功因素分析结果的制约。

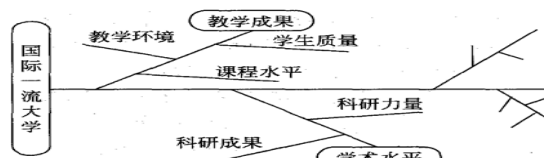


图 3.5 关键成功因素表示方法一

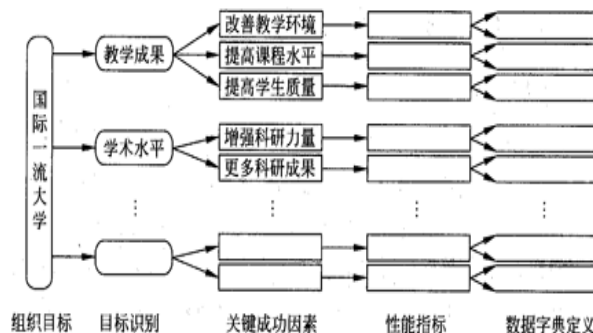


图 3.6 关键成功因素表示方法二

2.战略目标集转化法(SST)

此方法把整个战略目标看成是一个“信息集合”，由使命、目标、战略和其他战略变量(如管理的复杂性、改革习惯以及重要的环境约束)等组成。信息系统的战略规划过程是把组织的战略目标转变成系统的战略目标的过程，如图3.7所示。

具体步骤如下：

(1) 识别组织的战略集。可以先考查一下该组织是否有写成文的战略或长期计划，如果没有，就要去构造这种战略集合。首先描绘出组织各类人员(群体)结构，如卖主、经理、雇员等；然后识别各类人员的目标；最后对于每类人员识别其使命及战略。

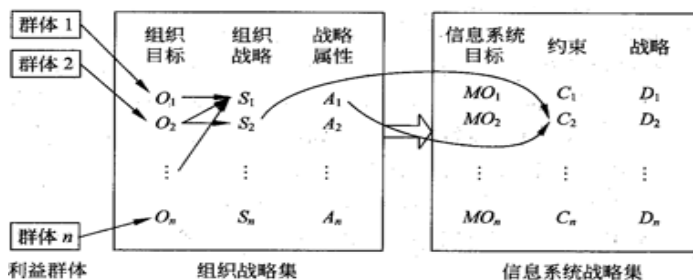


图 3.7 战略目标集转化法

(2) 将组织战略集转化为信息系统战略。如图3.7所示，首先根据组织目标确定信息系统目标；其次对应组织战略集的元素识别相应信息系统战略的约束，最后根据信息系统目标和约束提出信息系统战略。

3. 2. 2 目标系统框架分析

3.2.2.1 管理目标分析

任何一个组织结构都有一个目标，即管理目标，它是系统进行各项经济活动的指南。在开发信息系统时，应该首先弄清什么是组织的管理目标。

进行管理目标分析的步骤是：

- (1) 根据系统调查的结果，进行分析，归纳出现行系统中的关键问题，做出问题表。
- (2) 根据问题表，构造目标的层次结构，即目标树。在目标树中，最上层是总目标，以下各层是分目标或子目标，最下层是为实现目标而采取的具体措施，它是用来衡量目标是否切合实际的标准。
- (3) 对目标树中的各项分目标进行分析。分析各项分目标之间的关系，确定解决目标冲突的方法，指出各项措施的考核指标。
- (4) 将目标树按各层分目标在系统中所起的作用重新绘制。

3.2.2.2 系统目标分析

信息系统的目标分析是指在管理目标分析基础上，确立信息系统应在哪些方面发挥作用以及如何发挥作用。

通常，信息系统应该在下面几个方面发挥作用。

- 信息系统的辅助决策功能。在这方面，计算机可以充分发挥存储、检索、输出信息的能力和迅速、

准确的计算能力，还可以发挥信息系统的人—机系统优点，帮助决策者制定企业的长、短期决策。期望达到的目标有系统预测、计算机辅助决策、系统模拟与控制、利用计算机进行查询和统计。

- 信息系统的辅助管理功能。在这方面，计算机可以代替人的许多工作，如填制报表，生产经营数据统计，财务记账，人事档案的登录等。期望达到的目标是实现办公室自动化。

- 计算机作为计算工具。迅速准确地计算是计算机的基本特点。信息系统中的计算机系统一般要求功能较强，可用于进行系统中各项经济活动的有关计算，如生产经营方案的运算，财务核算，辅助设计中的运算。

在对信息系统目标进行分析后，根据管理目标的轻重缓急，确立信息系统目标实施的步骤，为初步设计打下基础。

3.2.2.3 系统范围及功能

明确系统的范围和功能，可使系统开发成本尽可能低，功能尽可能全。

确定系统范围和功能的原则如下：

- 根据已确定的系统目标和估算出的整个信息系统的信息量，考虑企业现有的客观条件，包括资金情况、设备条件、现场条件又技术水平、管理现状等，合理地确定系统的范围和功能。应注意，既不能超越客观条件的限制，也必须使人、财、物得到充分利用，使系统的功能尽可能完善，保证系统目标的实现。

- 新建的系统，可能要求现行的管理机构在组织上和功能上做某些调整和变动，以适应计算机的管理。在划分系统范围时，应按客观需要选择必要的系统结构和功能，不要受现行系统的限制。因为新系统在管理机制上，性能要优于现行系统，所以不能把现行体制搬到计算机上。只有这样才会使新系统更趋于合理，经济效益和社会效益更明显。

按照上述原则，确定系统的范围和功能应采取的步骤是：

- (1) 绘制出系统的总数据流程图。该图是系统分析阶段的各业务部门的数据流程图，综合绘制在一张图上。

- (2) 根据系统方案的要求、用户的要求和现行系统的环境及确定系统边界的原则，在总信息流程图上圈出系统范围。

- (3) 与用户讨论，协商修改有关内容。

- (4) 确定系统范围，并做分析说明。

3.2.2.4 系统总体结构及投资概算

为了将复杂的信息系统分解成便于理解和实现的部分，一般将信息系统分解为若干相对独立而又相互联系的子系统，即信息系统的主要系统。

1. 系统总体结构

为了方便信息系统的实现，还必须将子系统划分成若干个分系统，原因有三：首先，子系统间的相互关系仍非常复杂，每个系统都可能需要由其他系统产生的信息，分解之后，可以使这种关系更为明确，简单；其次，并不是在一个子系统中的所有过程都需要给予高优先级的支持；最后，给定的子系统往往较大，难以一次同时实现，可能是逐个分系统或几个分系统来实现。

2. 投资概算

投资概算包括以下 4 个方面内容。

- **计算机系统软、硬件设备投资。**将系统软、硬件配置表上列出的软、硬件设备按生产厂家提出的报价单，计算出它们的购置费，并汇总得出这部分的投资。在得出这部分的投资后，系统分析员应根据系统方案的实施步骤，考虑分期投资计划，列出与系统实施步骤相对应的购置计划。

- **系统开发费。**系统开发费指从系统调研到系统全部实现这一过程中花费的研制时间和人力折合的费用，一般按人月或人周计算。系统分析员首先要确定所需的人员和能够投入的人员，再估计出开发周期，将这两个数据折合成“人月或人周”，最后得出开发费用。

- **系统安装和维护费用。**包括 3 方面内容：计算机软、硬件的安装和维护费用，这个费用一般根据计算机软、硬件生产厂家的报价估算；信息系统的安装和维护费用，系统分析员应该参考类似系统的情况，估计出人力和时间，再折合成费用，与所需物力折合的货币费用一起汇总成信息系统的安装和维护费用；基础设施的维修和改造费用，这里的基础设施包括机房、通信系统、供电系统和照明等。

- **人员培训费。**为使系统能够正常地运行和维护，需要对操作和维护人员进行定期培训，这种费用往往占很大的比重。

这 4 种费用也与系统的开发步骤有关，在系统方案报告中，也要与相应的开发步骤一起列出，并标明投资时间。

3.2.3 可行性分析及总体规划报告

可行性分析也称为可行性研究。可行性研究已经成为新产品开发、工程投资等领域中决策的重要手段。信息系统的开发同样也需要进行可行性研究,以便避免盲目投资,减少不必要的损失。

3.2.3.1 可行性分析的内容

可行性分析包括两个方面的内容:建立信息系统的必要性和建立信息系统的可能性。

一般来说建立信息系统的必要性大概有3种情况。

- **“显见”的必要性。**如现在的系统已经不适合或不能满足企业的需要,企业的发展使得数据量越来越多等。这种情况的分析较为容易,结论也比较容易得到。

- **“预见”的必要性。**如企业的发展以及技术的进步,使得企业领导预见到未来不久信息处理手段必须更新,否则不能适应未来信息处理的需要。

- **“隐见”的必要性。**有些系统,如社会服务系统,服务效率很低,明显地影响到社会利益和经济利益。但这种影响不是直接看得见、摸得着的,不是集中的而是分散的,不是突发的而是长期积累的。但是如果这种问题长期积累下去,量变将引起质变。所以应该重视这些“隐见”的系统危害性,建立一个新的高效率的系统。

建立信息系统的可能性主要有以下内容:

- **经济可行性。**经济是开发系统的基础。开发信息系统需要很多钱,不仅仅包括购置硬件设备的费用,还要将购置软件的费用,以及系统的开发和维护的费用包括进去。

- **技术可行性。**信息系统是高技术,缺乏高技术物质基础以及高科技人才是无论如何不能实现的。除了计算机硬件,软件(包括数据库,操作系统,编程语言等)等高技术产品以外,还需要通信网络、汉字技术、数学模型、经济管理等理论和技术,如果一个企业没有足够的技术支持,那是绝对不能成功地开发信息系统的。

- **管理上的可行性。**考虑当前系统的管理体制是否有条件提供新系统所必需的各种数据,以及企业最高层领导及各级管理人员对新系统所提供信息需求的迫切性,即新系统的必要性。此外,对新系统运行后对各方面产生的影响力加以考虑。例如,用计算机处理大批信息,可以代替某些管理人员的工作,于是涉及到他们的工作安排问题。此外,还有当前系统的业务人员对新系统的适应性等。

- **开发环境的可行性。**企业领导意见是否一致,有无资金,能否抽出骨干力量参加系统开发等。

在对上述几个方面的可行性进行分析后,最后应写出新系统开发的可行性报告,并经有关部门审核。如果该可行性报告通过,则可进入系统分析阶段进行开发工作。如果某些条件不成熟,则需改变系统目标,创造条件后再次进行可行性研究。如果可行性研究的结论认为完全不可行,则应放弃系统开发。

3.2.3.2 可行性分析报告

可行性分析报告是可行性分析的最后成果。该报告必须用书面的形式记录下来,作为论证和进一步开发的依据。可行性报告大致由以下内容组成。

1.引言

引言一般包括以下内容:

- 摘要,包括现行系统的名称、目标和功能等;
- 背景,说明系统的用户、开发者,本系统与其他系统或机构的关系;
- 参考资料,包括下达本系统可行性研究的文件、合同或批文;
- 本报告引用的专门术语说明。

2.现行系统调查与分析

主要包含以下内容:

- 现行系统初步调查,包括组织机构层次,任务和范围;
- 主要业务流程及对信息的需求;
- 当前系统的工作量;
- 当前系统运行的各项费用开支、人员和设备;
- 已有计算机的配置、使用效率和存在的问题;
- 现行系统存在的主要问题和薄弱环节;
- 需求调查和分析,包括用户提出的和开发人员分析得出的需求。

3.新系统建设方案

主要包括:

- 新系统的目标和范围;
- 新系统规模以及初步方案(规模、组成和结构等);

- 系统及人员培训实施方案；
- 投资方案(投资金额、来源和时间安排等)。

4. 其他

可行性分析报告还应包括可行性论证，主要从技术可行性、经济可行性、社会可行性等方面进行论述；其他方案以及方案间的比较分析，以及结论。

3. 3 系统分析与建立逻辑模型

3. 3. 1 系统分析概述

3.3.1.1 系统分析的任务和目的

系统分析的主要任务是对现行系统进一步详细调查，将调查中所得到的文档资料集中，对组织内部整体管理状况和信息处理过程进行分析，为系统开发提供所需资料，并提交系统方案说明书。系统分析侧重于从业务全过程的角度进行分析，主要内容有：业务和数据的流程是否通畅，是否合理；数据、业务过程和管理功能之间的关系；原系统管理模式改革和新系统管理方法的实现是否具有可行性等。

确定的分析结果包括开发者对于现有组织管理状况的了解，用户对信息系统功能的需求，数据和业务流程，管理功能和管理数据指标体系以及新系统拟改动和新增的管理模型等。

最后，提出信息系统的各种设想和方案，并对所有的设想和方案进行分析、研究、比较、判断和选择，获得一个最优的新系统的逻辑模型，并在用户理解计算机系统的工作流程和处理方式的情况下，将它明确地表达成书面资料—系统分析报告，即系统方案说明书。

3. 3. 1. 2 系统分析的主要步骤

企业信息系统是一个具有业务复杂性和技术复杂性的大系统，为了使目标系统既能实现当前系统的基本职能，又能改进和提高，系统开发人员首先必须理解并描述出已经实际存在的当前系统，然后进行改进，从而创造出基于当前系统，又高于当前系统的目标系统，即新系统。

系统分析过程一般按如图 3. 8 所示的逻辑进行。

- (1)认识、理解当前的现实环境，获得当前系统的“物理模型”。
- (2)从当前系统的“物理模型”抽象出当前系统的“逻辑模型”。
- (3)对当前系统的“逻辑模型”进行分析和优化，建立目标系统的“逻辑模型”。
- (4)对目标系统的逻辑模型具体化(物理化)，建立目标系统的物理模型。

系统开发的目的是把现有系统的物理模型转化为目标系统的物理模型，即图 3. 8 中双虚线所描述的路径，而系统分析阶段的结果是得到目标系统的逻辑模型。逻辑模型反映了系统的功能和性质，而物理模型反映的是系统的某一种具体实现方案。

按照图 3. 8 所示，可将系统分析阶段的主要工作步骤分为：

- (1)对当前系统进行详细调查，收集数据。
- (2)建立当前系统的逻辑模型。
- (3)对现状进行分析，提出改进意见和新系统应达到的目标。
- (4)建立新系统的逻辑模型。
- (5)编写系统方案说明书。

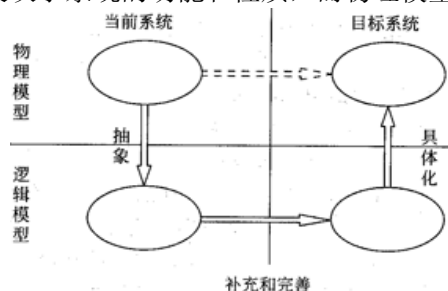


图 3. 8 系统分析过程图

3.3.2 详细调查

系统调查是系统开发过程中的基础工作，通常分为初步调查和详细调查，它们分别在总体规划和系统分析阶段进行。由于新系统一般都是在当前系统的基础上发展而来的，因此在系统分析阶段，通过对当前系统的详细调查，收集整理有关数据，弄清业务现状，查明执行效果，发现薄弱环节，就可以为改进当前系统和开发高质量的新系统提供可靠的资料，为最终建立新系统的逻辑模型打下坚实的基础。

3. 3. 2. 1 详细调查的主要内容

详细调查的范围应该是围绕组织内部数据流所涉及领域的各个方面。但应该注意的是，数据流是通过物流产生的，物流和数据流又都是在组织中流动的。故调查的范围就不能仅仅局限于信息和数据流，还应该包括企业的生产、经营、管理等各个方面。

具体地说，详细调查有如下方面：组织目标和发展战略，组织机构和功能业务，管理模式和管理方法，决策方式和决策过程，业务流程与工作形式，数据、数据处理与数据流程，产品构成及其工艺流程，可用资源和限制条件，现有问题和改进意见等。

下面针对其中几个主要的方面进行介绍。

1.静态信息调查：组织结构的调查

要建立企业信息系统，就必须知道当前系统的组织结构设置情况和它们之间的隶属关系。并关心那些

与计算机管理有关的机构和关系。

通常用组织结构图来描述当前系统组织机构的层次和隶属关系。在绘制组织结构图时应注意，除了后勤等与企业生产、经营、管理环节无直接关系的部门外，组织结构图一定要尽可能全面、准确地反映企业的组织及权力隶属现状。这样做的好处是：一方面通过组织结构图对企业的全貌有一个总体上的认识，便于系统分析工作的展开；另一方面可以按决策层、业务管理层和执行层等将企业管理者进行分层，以便于未来建立的信息系统能有针对性地为不同层次的管理者提供不同细度的信息。

2.静态信息调查：功能体系的调查

系统有一个总目标，为了达到这个目标，必须要完成各子系统的功能，而各子系统功能的完成，又依赖于下面各项更具体的功能执行。功能结构调查的任务，就是要了解或确定系统的这种功能构造。

功能要依靠组织结构来具体实现，因此，在理想情况下，功能和组织应该是一致的。但是由于客观情况的复杂性，在当前系统中，功能体系和组织结构并不能一一对应，这就要求在进行调查时认真分析，加以划分。

3.动态信息调查：业务流程的调查

在对系统的组织结构和功能体系有所了解的基础上，还需要从一个实际业务流程的角度将系统调查中有关该业务的资料串起来，以便于对企业现有的工作过程有一个动态的了解。对业务流程的调查通常可按原有的信息流动过程，逐个调查当前系统中每个环节的处理任务、处理顺序和对时间的要求等情况，弄清每个环节的信息来源和去向。

4.动态信息调查：数据流程调查

实际上在业务流程调查阶段就已经涉及到了数据流程问题，但业务流程调查的工作重点是将组织与功能匹配起来，将功能与功能关联起来由于企业数据是管理信息系统的主要原材料，因此完全有必要对数据流程进行专门、详细的调查。

其中收集资料是数据流程调查阶段的基础和重点工作。在现行系统中存在的大量单据、原始凭证和各种各样的报表，都是信息的载体，对它们的调查、收集和分析，能够对现行系统的数据收集、输入、存储、加工和输出等环节做进一步的研究，为今后系统的详细设计提供依据。

3.3.2.2 详细调查的原则

1.自顶向下全面展开

系统调查工作应严格按照自顶向下的系统化观点全面开展。首先从组织管理工作的最高层开始调查，然后再调查与最、高层管理工作紧密相关的下一层的各项工作，依此类推。

2.存在的不一定是合理的

组织内部的每一个管理部门和每一项管理工作应该根据组织的具体情况和管理需要而设置。调查工作的目的是要搞清这些管理部门存在的理由、环境条件以及工作的详细过程，然后再通过分析其在新的信息系统支持下有无优化的可能性。

3.分工和协作相结合

对于一个大型系统的调查一般都是多个系统分析人员共同完成的，为了提高调查的工作效率，需要按分工和协作相结合的工程化的方法组织调查。工程化就是将工作事先计划，对多个人的工作方法和调查所用的表格、图例做到统一规划，以便能相互沟通，分工协作。

4.点面相结合展开调查

开发信息系统，总是要开展全面调查工作。但如果近期内只需开发组织内某一局部的信息系统，那么就必须坚持全面和重点结合的方法，即在全面调查的基础上重点调查。

5.主动沟通的工作方式

系统调查将涉及组织内部管理工作的各个方面，调查者应主动与被调查者在业务上进行沟通，创造出一种积极、友善的工作环境和人际关系是调查工作顺利进行的重要基础。

3.3.2.3 详细调查的方法

企业信息系统开发中常用的一些调查方法有以下几种。

- **收集资料。**就是将各部门、科室日常业务中所用的各种单据、凭证、报表统统收集起来，对于一些保密性要求不严的资料，采用复印的方式取得，不必得到原始报表。

- **开调查会。**开调查会是一种集中征询意见的方法，适合于对系统的定性调查。调查会有助于大家的见解互相补充，以便形成较为完整的意见。

- **个别访问。**开调查会不能完全反映出每个与会者的意见，因此，在会后根据需要再进行个别访问。

- **书面调查。**根据系统特点设计调查表，进行问卷访问，征求意见和收集数据，这种方法适用于比较复杂的系统。

- **参加业务实践。**在大规模系统调查后，针对仍然存在的问题的工作，系统分析人员可以参与企业的实际工作，发现问题的本质，寻找解决问题的办法。
- **发电子邮件。**如果企业已经具有网络设施，可通过因特网和局域网发电子邮件进行调查，这可大大节省时间、人力、物力和金钱。

在系统调查时，应注意下面的一些问题：

- **事先计划。**系统分析人员要和用户共同制定调查进度的计划，以便事先安排时间、地点和内容，并通知有关人员做好准备。
- **调查态度。**为了取得理想的调查效果，系统分析人员应该始终具有耐心，并掌握一定的调查技巧和具有处理人际关系的能力。
- **调查顺序。**先自上而下初步调查，在了解总体和全局的基础上，再由下而上地进行具体调查。
- **研究分析。**对现行系统的调查过程主要是原始素材的汇集过程。系统分析人员必须对调查结果进行整理、研究，并绘制成描述现行系统的有关图表，以便在较短时间里对现行系统有全面和详细的了解。

3.3.3 需求分析

需求分析就是对处理的对象进行系统调查，在完全弄清用户对新系统的确切要求后，用统一、规范的图表和书面语言表达出来，它是系统开发工作中最重要的环节之一。需求分析工作量很大，所涉及的业务、人、数据、信息都非常多。所以如何科学地组织和适当地着手展开这项工作是非常重要的。

1. 系统范围与目标分析

确定系统范围、定义业务目标和系统需求分析，应在同一阶段完成。只有这些得到了确定，才能确定达到这些目标的方法。这一阶段，主要完成以下 3 个任务：

- **确定系统范围。**把系统范围确定并文档化。然后再确定哪些在系统范围的边缘，即将来系统范围如果发生变动，哪些将被包含进来，哪些将有可能被排除。
- **确定系统需求。**也就是把业务目标、系统目标、项目目标和对系统的关键功能需求文档化。而对系统的关键功能的需求描述，将在以后被用来作为选择解决方案的依据。
- **系统内容说明书。**它融合了系统范围、需求描述和分析中产生的其他信息，这个文档可在以后指导解决方案的选型和实施，同时也是对将来的需求和变化进行控制的一个参考。

2. 系统组织结构与功能分析

系统组织结构与功能分析的目的，是为了调查组织发展目标及其战略规划，了解组织的现状及管理体制，划分组织的各个功能，分清组织内各种流向，如物资流(正向流动)、资金流(反向流动)和信息流(双向流动)。

在系统组织结构与功能分析中，有如下要求：

- 了解组织的目标及其战略规划；
- 了解组织结构及各部分的功能；
- 了解相关部门职能上的各种联系；
- 分析组织结构的合理性；
- 分析组织结构设置的必要性和合理性；
- 发现其中的问题；
- 提出改进的意见。

在系统组织结构与功能分析中，有以下几个主要的工具可以应用。

- **组织结构图**，如图 3. 9 所示。
- **组织/业务关系图**。图 3. 10 描述了业务和部门的关系，包括主要负责单位，协助单位和参与单位。其中 * 表示主要负责单位，X 表示协助单位，/ 表示参加单位。
- **业务功能一览表**。如图 3. 11 所示，描述每一种业务所具有的功能。

3. 系统性能分析

信息系统的性能评价指标是客观评价信息系统性能的依据，一般包括系统平均无故障时间，系统联机响应时间、处理速度和吞吐量，系统操作灵活性和方便性，系统加工数据的准确性，系统的可扩充性和系统的可维护性。

3. 3. 4 业务流程详细调查与分析

根据对组织结构图和业务功能体系图的分析，可决定下一步重点调查的部门，然后对该部门的业务信

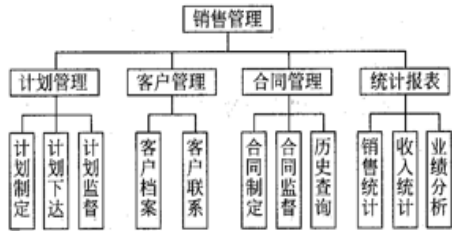


图 3.11 业务功能一览表

业务 \ 组织	部门 A	部门 B	部门 C	部门 D	部门 E	部门 F
业务 1		X	*		X	
业务 2		X	*			/
业务 3			X		X	/
业务 4	*		/	X		/

图 3.10 组织/业务关系图

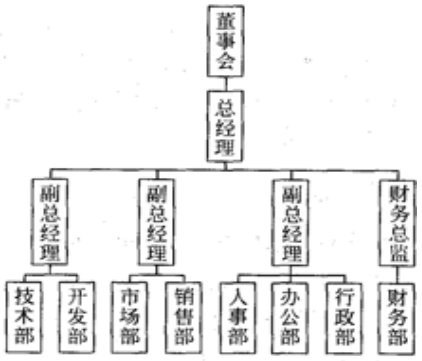


图 3.9 组织结构图

息、业务流程等进行详细调查。流程分析的目的是了解各个业务流程的过程，明确各个部门之间的业务关系，明确每个业务处理的意义，为业务流程的合理化改造提供建议，为系统的数据流程变化提供依据。

业务流程分析的步骤可以总结如下：

- (1)通过调查掌握基本情况。
- (2)描述现有业务流程—绘制业务流程图。
- (3)确认现有业务流程。
- (4)对业务流程进行分析—知识和经验支持。
- (5)发现问题提出解决方案。
- (6)提出优化后的业务流程。

1.组织结构与业务流程详细调查

建立企业信息系统几，就必须知道当前系统的组织机构设置情况和组织机构之间的隶属关系。对组织结构的调查和画出组织结构图是为了分析和了解那些与计算机管理有密切关系的机构和部门，这些部门是数据比较集中的地方。

按现行系统物质、信息或数据流动的过程，逐个调查现行系统中每个环节物质流、信息流或数据流，以弄清每个环节的物质流和信息流的来源和去向，并将现行系统按数据的加工顺序进行描绘。

2.业务流程图和系统概况图

开发人员通过对现行系统业务流程进行详细调查，并对调查结果进行充分认识、深入理解和认真分析，可在详细调查的基础上描述出已经实际存在的现行系统业务流程，即将调查结果用图表和图形描述出来，形成现行系统业务流.程图、系统概况图等。

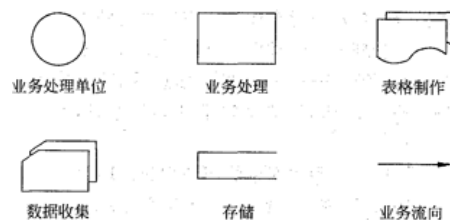


图 3.12 业务流程图符号

业务流程分析主要是为了描述现行系统的物理模型，是具体分析的第一步。为了要对详细调查结果进行整理和分析，然后再经业务人员确认，就必须采用一些简单方便的方法和工具来明确表达它们，使之成为系统分析员和用户之间进行交流的共同语言。业务流程图可以解决此问题。

现行系统概况图是在现行系统业务流程图基础上提取系统的基本要素——输入、输出、处理、存储和外部环境等编制而成的。它是流程图的文字概括，两者配合使用。

业务流程图的基本符号见图 3.12;其中业务处理单位表达某项业务参与的人或事务;业务流向表达业务数据的流动方向;表格制作表明数据的载体;存储表明一种数据载体,数据是作为档案保存的;业务处理表明业务处理功能,一般用一个简单的动词或动宾词组表示。

3.业务流程优化与再造

企业的业务流程直接体现企业的核心能力，是企业完成其使命、实现其目标的基础。传统的企业管理模式下企业的业务流程，非增值环节多，信息传递缓慢，同一流程各个环节之间和不同流程间关系混乱。特别是完整的业务流程被不同职能部门分割，大大降低了流程的效率与效益，难以及时捕获迅速变化的市场机会，致使整个企业效率与效益低下，应变能力差。因此必须应用现代信息技术，对企业流程进行改造与更新。

20 世纪 80 年代以来，国际管理学术界和企业界兴起了管理改革的热潮。首先兴起的是业务流程改善(Business Process Improvement, BPI)，寻求对企业的业务流程的连续、渐进的改善。1990 年，“重新设计”(Reengineering)的思想被引入管理领域，提出了业务流程再造(Business Process Reengineering, BPR)的概念。BPR 是指对企业的业务流程进行根本的再思考和彻底的再设计，从而使企业的关键绩效指标，如成本、质量、服务、效率等，获得巨大的提高。

企业流程再造(BPR)应遵循以下原则：

- **有一个明确的、具有启发性的目标，即共同远景。**把企业的业务流程看作是企业战略的对象，把流程与企业联系起来是流程再造项目成功的必要条件。然而，在一个复杂的企业里，在战略和流程之间往往存在着一条鸿沟。连接企业战略和企业的业务流程的桥梁便是流程远景。因此，流程创新应该从企业的战略开始，所期望的战略定位和流程远景应该是业务流程再造的起点。

- **充分考虑顾客的价值。**在当今以消费为导向的时代，有效地提供顾客满意的产品和服务是 BPR 的另一个驱动力。

- **必须服从统一指挥。**业务流程再造必须是一个自上而下的过程，同时它又是一个跨部门综合性的全新工程，为确保 BPR 的有序贯彻，必须使员工服从统一指挥。同时，要求领导人必须是企业高层的、资深的、有威信的核心人员。

• **充分做好横向及纵向沟通。**一方面，再造从上往下推行，高层管理人员必须讲清楚为什么这样做，如何做，使得全体员工理解再造的方法和目标；另一方面，流程再造势必造成中层管理人员的减少，这也就要求部门流程之间多加沟通。

• **认识流程再造的两大要素—信息技术/信息系统和人员组织管理。**流程再造将对企业流程进行彻底的革新，信息技术和人员组织的管理是企业流程创新的源泉。

• **树立典范、逐步推进，充分利用变革的涟漪效应。**BPR 在实施过程中一般也不可能所有流程并驾齐驱，这就要求精心挑选适当规模实验项目，表明流程再造的有效性，再推广到整个组织，实现涟漪效应。

流程再造方法一般有两类：全新设计法 (Clean Sheet Approach) 和系统改造法 (Systematic Redesign)，前者遵循“推倒重来”的主张，从根本__L 抛弃旧流程，零起点设计新流程；后者继承逐步改善的思想即 BPI 的思想，辨析理解现有流程，在现有流程的基础上，系统渐进地创造新流程。

3.3.5 数据流程分析

数据流贯穿于企业组织的每一个活动中，可以说没有数据流就没有企业的活动。通过对数据流程的分析，一方面可以更准确地了解企业管理活动的全过程，分析出各种管理活动的实质和相互间的关系；另一方面，数据是信息的载体，是正在开发的企业信息系统的主要对象，因此必须对系统调查中所收集的数据和数据处理过程进行分析整理，为以后的新系统逻辑模型、数据库结构和功能模块设计打下基础。

数据流程分析就是把数据在现行系统内部的流动情况抽象出来，舍去了具体组织机构、信息载体、处理工作等物理组成，单纯从数据流动过程来考查实际业务的数据处理模式。数据流程分析主要包括对信息流动、传递、处理、存储等的分析，其目的就是确定合理的数据项，确定合适的数据流向，确认合适的数据处理过程，并发现和解决数据流通中存在的问题。

3.3.5.1 数据流及数据流图

一个系统的基本组件包括输入流、输出流以及处理过程。企业作为一个系统也存在输入流、输出流以及处理过程，企业输入流、输出流的表现形式多种多样，在处理过程中经常要涉及各式各样的输入流、输出流。要想很好地了解一个企业的活动，需具体分析其中所包含的各种流。

• **物资流：**工厂输入原材料与零配件，经过加工制造过程，输出成品；商店进货，经过销售过程，把货卖给顾客。这些输入与输出物品的流动都属物资流。

• **事务流：**事务是指系统与其外部环境或子系统之间发生的交往活动而引起的一系列信息处理活动。例如，工商企业接到订货单，便有开发货单、发票、记账等信息处理活动，它们统称为订单处理，这就是一项事务。再如政府经济行政管理部门接到下级的请示报告，经过调查研究和有关主管人员分析、开会讨论，协调不同意见，做出统一决定，作为对下级的指示，这也是一种事务，可称之为请示报告的处理。这两种事务的流动有一个重要不同之处：工业企业是制造和出售产品的，商业企业是买卖产品的，这是它们的主要业务，事务是随其主要业务而生的，如原材料的购买、产品的购进与售出等，事务流是伴随物资流而产生的；而政府经济行政管理部门只有事务流，没有物资流，这里的主要业务是事务，即情况的处理。

• **货币流：**货币流是指资金的流动，如购买原材料的付款、工资的支付等。货币流一般是伴随物资流和劳务偿付而产生的，但在银行业务中货币流则随存取及信贷业务而产生。

• **人员流：**人员流是指组织内部工作人员的增减和流动。

• **机器及设备流：**机器及设备也是一种物资流，这里指的是机器、设备等的买、卖和流动，它们不同于物质生产的物资流。

• **数据流：**数据流是人们用以记录以上各种流的抽象表达形式。各种流在一个企业内的出现，都各自同时伴随着一个数据流的产生。例如，一个产品制造的物资流，总伴随着生产计划安排，并产生领料单、出料单、生产记录、送货单、入库单等数据的流动。数据流经常贯穿于组织内每个活动中。可以说没有数据流就没有企业的活动。信息的物理表达为数据，票据、凭证、函电、公文等均为数据不同的具体形式。事务流的具体表现也是数据流。因此，数据流和事务流往往会存在于同一渠道内而不易区别。区别的重要标准是，数据流是用以控制其他流的，而事务流则为被控制的对象。数据流在所有各流中有特别重要的意义，即任何其他流的产生与存在总伴随有数据流的产生与存在，数据流是对其他流进行控制的依据。

数据流图或称数据流程图 (Data Flow Diagram, DFD) 是一种便于用户理解、分析系统数据流程的图形工具。它摆脱了系统的物理内容，精确地在逻辑上描述系统的功能、输入、输出和数据存储等，是系统逻辑模型的重要组成部分。

数据流图就是组织中信息运动的抽象，是企业信息系统逻辑模型的主要内容之一。这个模型与系统的物理描述无关，它用一种图形及与此相关的注释来表示系统的逻辑功能，表示所开发的系统在信息处理方面要做什么。由于图形描述简明、清晰，不涉及到技术细节，所描述，的内容是面向用户的，所以数据流

图是系统分析人员与用户进行交流的有效手段，也是系统设计，即建立所开发的系统如物理模型的主要依据之一。

采用数据流图的方式进行数据流程分析一般应遵循以下原则：

- **明确系统边界。**一张数据流图表示某个子系统或某个系统的逻辑模型。系统分析人员要根据调查材料，首先识别出那些不受所描述的系统控制，但又影响系统运行的外部环境，这就是系统的数据输入的来源和输出的去处。把这些因素都作为外部实体确定下来。确定了系统和外部环境的边界，就可集中力量分析和确定系统本身的功能。

- **在总体上遵循自顶向下逐层分解的原则**，即按照结构化方法的思想，采用分层的数据流图，把大问题、复杂的问题分解成若干个小问题，然后分别解决。

- **在局部上遵循由外向里的原则**，即先确定每一层数据流图的边界或范围，再考虑流图的内部，先画加工的输入和输出，再画加工的内部。

分层的数据流图一般由顶层、中间层和底层组成。顶层抽象地描述了整个系统的情况，包括系统的范围、系统与外界实体间的关系(输入输出流)；底层具体画出了系统的细部；中间层则是从抽象到具体的逐步过渡。

3.3.5. 2 数据流图的绘制与检验

对于不同的问题，数据流图可以有不同的画法。具体操作时可按下述步骤进行。

1.识别系统的输入和输出

在系统分析初期，系统的功能需求等还不很明确，为了防止遗漏，不妨先将范围定大一些，系统边界确定后，越过边界的数据流就是系统的输入或输出。

可以首先确定所开发的系统的外部实体，即系统的数据来源和去处。然后再确定整个系统的输出数据流和输入数据流，把系统作为一个加工环节，画出关联图。一般应把数据来源置于图的左侧，数据去处置于图的右侧。

2.绘制系统内部数据流

从系统输入端到输出端(也可反之)，逐步把数据流和加工连接起来，当数据流的组成或数据发生变化时，就在该处画一个“加工”。

首先确定系统的主要信息处理功能，按此将整个系统分解成几个加工环节。确定每个加工的输出与输入数据流以及与这些加工有关的数据存储。根据各加工环节和数据存储环节以及输出与输入数据流的关系，将外部实体、各加工处理、数据存储环节用数据流联结起来，为各数据流、各加工环节和数据存储环节命名、编号，这样就形成了所开发系统的数据流图顶层图(总图)的草图。

然后再补充一些细节，如出错处理等；画数据流图时还应同时画上文件，以反映各种数据的存储位置，并表明数据流是流入还是流出文件；再回过头来检查系统的边界，补上遗漏但有用的输入输出数据流，删去那些没被系统使用的数据流。

3.对复杂加工进行分解

运用“由外向里”、“自顶向下”的方式对加工进行分解。将需要分解的上一层图的加工环节分解成具有明确逻辑功能的数个加工环节，按上一步骤中的作法，对上层需分解的加工环节画出分解数据流草图。

4.对草图进行检查和合理布局

主要是检查分解是否恰当、彻底，DFD 中各成分是否有遗漏、重复、冲突之处，各层 DFD 及同层 DFD 之间关系是否正确及命名、编号是否确切、合理等，对错误与不当之处进行修改。

5.和用户交流

和用户讨论的主要问题是：系统逻辑功能的设置和描述是否合理，能否满足用户的信息需求，数据流和数据存储的内容以及数据来源和去处(外部项)是否符合实际，描述是否准确、合理；用户在了解数据流图的全部内容后对系统逻辑功能有什么进一步的意见与要求。系统分析人员根据与用户讨论的结果对数据流图的草图进行修订。

6.检查、修改、完善

系统分析负责人对数据流图进行复审。检查数据流图是否全面、准确地反映了系统调查以及用户的意见，勾画出现行系统的数据处理逻辑。如果有地方不太明确，应重新调查，并进行修改完善。否则通过复审，数据流图绘制过程结束。

对于一个规模较大且结构复杂的信息系统，它的数据流图可能包括几千个加工，要把它们都画在同一张纸上是不可能的。为了控制复杂性，通常按照“自顶向下，逐层分解”的技术分层处理，因此在多数情况下，这样的数据流图被称作分层数据流图。

分层数据流图便于人们理解和使用，但在绘制时应注意以下事项：

①**自顶向下、逐层分解**。就是由系统外部至系统内部、由总体到局部、由抽象到具体的系统逻辑模型建立的过程。在整个数据流图绘制过程中，始终要把握住对系统总体目标与总体功能的要求，在给定的系统边界范围内进行工作。为了使数据流图简洁、清晰、功能明确、方便交流，分解的层次和每张图的内容要适当。

根据经验，每张图的加工项目以不超过 7-8 个为宜。加工的分解要抓住主要问题，每个分解后的加工环节功能明确，易于理解，一般分解后的加工先确定输出数据流，再确定输入数据流，然后定义加工的内容，进行命名和编号。图上不应有无输入或无输出的加工环节。

在数据流图分解中，要保持各层成分的完整性与一致性。数据流图的逐层分解是以加工的分解为中心的，属于功能分解性质。把上层加工环节称为父加工环节，下层环节为子加工环节。加工的分解可能导致数据流的分解、数据存储的分解甚至外部项的分解。分解时一定要保持父项(被分解项)的内容为对应各子项(即分解后的各项)的内容之和。

下层数据流图不应出现不属于上层图中的数据流子项的新数据流，但可以出现不属于上层图的数据存储环节。子项的新的数据存储环节。因为随着加工的分解，分解后的加工(子加工)之间的界面可能是上层图未定义的数据存储，这就需要在下层图加以定义、命名与编号。

数据流图逐层分解也可能导致某个或某些外部项的分解。因为分解后的各子加工可能与上层图中某个外部项的不同组成部分相联系。当外部项的分解有助于更明确描述系统某些部分的功能与信息需求时，下层图要对分解后的外部项加以定义和命名。下层图不应出现不属于上层图外部项的子项的新外部项。

②**数据流必须经过加工环节，即必须进入加工环节或从加工环节流出**。不经过加工环节的数据流(如外部项之间的数据交换)不在数据流图上表示。因这类数据流与所描述的系统无直接关系。

③**数据存储环节一般作为两个加工环节的界面来安排**。只与一个加工环节有关的数据存储，如果不是公用的或特别重要的，可不在数据流图上画出。直接从外部项来与直接到外部项去的数据流应直接与加工环节相连，不应通过数据存储环节相连。

④**编号**。每个数据加工环节和每张数据流图都要编号。按逐层分解的原则，父图与子图的编号要有一致性，一般子图的图号是父图上对应的加工的编号。如顶层图的图号为 0，其中各加工环节按 1, 2, 3, … 顺序编号，1 号加工环节分解后的子加工按 1.1, 1.2, 1.3, … 编号，2 号加工环节按 2.1, 2.2, 2.3, … 依此类推。加工环节 1.1 分解后的子环节为 1.1.1, 1.1.2, …，依此类推。

数据流与数据存储环节也要进行编号以便于编写、分析与维护。编号方法原则上与加工环节的编号方法相同。

⑤**只绘制所描述的系统稳定工作情况下的数据流图**，不描述系统启动或结束工作时功能和数据流运动规律处于变动状态的情况。

对于一个大型企业信息系统，由于在系统分析初期，开发人员对问题的理解深度不够，数据流图也不可避免会存在某些缺陷或错误，此时就需要进行检查、修改和完善工作。

下面介绍如何从正确性和可读性两方面对数据流图进行改进。

数据流图的正确性可从以下几方面检查：

- **数据守恒**。一个加工环节的输出数据流仅由它的输入数据流确定，这个规则绝不能违背。数据不守恒的错误有两种，一是漏掉某些输入数据流；二是某些输入数据流在加工环节内部没有被使用。

- **文件使用**。在数据流图中，文件与加工环节之间数据流的方向应按规定认真标注，这样有利于对文件使用正确性的检查。例如，如果发现某个文件只有输入流，而没有输出流，要么是画错了，要么是系统分析出现了问题，因为一个不产生任何输出流的文件是没有意义的。

- **子图和父图平衡**。造成子图与父图不平衡的一个常见原因是在增加或删除一个加工环节时，忽视了对父图或子图的修改。在检查数据流图时应特别注意这一点。

- **加工和数据流的命名**。加工和数据流的名字必须体现被命名对象的全部内容，而不是一部分。对于加工的名字，应检查它的含义与被加工的输入输出数据流是否匹配。

如果数据流图的可读性不强，即使正确无误，也不会很好地发挥作用。一般可以从以下几方面提高数据流图的可读性：

- **简化加工之间的联系**。各加工之间的数据流越少，各加工的独立性就越高，因此应当尽量减少加工之间的数据流的数目。加工间的数据流最好控制在 1-2 条，否则就应该考虑对加工进行合并、删除。

- **分解应当均匀**。在同一张数据流图上，应避免出现某些加工已是最小功能单元，而另一些加工却还等待继续分解好几层的情况出现。

- **命名应当恰当**。理想的加工名由一个具体的动词和一个具体的宾语组成。数据流和文件的名字也应

具体、明确。命名应尽量做到使人一目了然。

数据流图从总体上描述了系统的逻辑功能，系统内各部分的信息联系及与系统外各有关事物的联系，反映了系统中信息运动的规律，是系统逻辑模型的主要描述形式。

但数据流图在描述系统逻辑功能和有关信息内容的细节方面仍存在较大的局限性。如：

- 难以在数据流图上标识出数据流、数据存储、加工和外部项的具体内容，如数据流的组成元素、数据存储的数据结构、存取要求、数据量、加工的处理过程与算法等。
- 不能反映系统中的决策与控制过程。
- 难以对系统中人机交互过程以及信息的反馈与循环处理进行描述。

因此，在系统分析中，除了用数据流图描述系统逻辑模型外，还要辅以其他工具，如数据字典、结构化语言、决策表、决策树等。

3.3.5.3 数据流图绘图举例

数据流图由 4 种基本符号组成，见图 3.13。

①**数据流**。数据流由一组确定的数据组成。例如“领料单”数据流由物资编码、物资名称、规格型号、领用数量、出料仓库、领用单位、日期等数据组成。数据流用带有名字的箭头表示，名字表示流经的数据，箭头表示流向。数据流可以从加工流向加工；也可以从加工流向文件，从文件流向加工；还可以从源点流向加工或从加工流向终点。

对数据流的表示通常有以下约定：

- 名字最好能反映出数据流的含义，不同的数据流间不能同名。
- 对流进或流出文件的数据流不需标注名字，因为文件名本身就足以说明数据流了。而其他的数据流则必须标出名字。
- 两个加工之间可以有多个不同的数据流，这是由于它们的用途不同，或它们之间没有联系，或它们的流动时间不同。
- 数据流图描述的是数据流而不是控制流，因此像业务流程图中的一些控制流应从数据流图中删去。

②**数据处理**。加工是对数据进行的操作，它把流入的数据流转换为流出的数据流。每个加工都应取一个名字表示它的含义，并规定一个编号用来标识加工在层次分解中的位置。名字中必须包含一个动词，如“计算”、“打印”、“汇总”等。

加工的作用主要是：

- 改变数据的结构，例如将数组中各数据项重新排序。
- 产生新的数据，例如对原来的数据汇总、求平均值等。

③**数据存储**。数据存储表示数据保存的地方。数据存储名应与它的内容一致，写在开口长方形内。从数据存储流入或流出数据流时，数据流方向是很重要的。如果是读数据存储，则数据流的方向应从数据存储流出，写数据存储时则相反。如果是又读又写，则数据流是双向的。在修改数据存储时，虽然必须首先读数据存储，但其本质是写数据存储，因此数据流应流向数据存储，而不是双向的。

④**外部实体**。外部实体指系统以外又与系统有联系的人或事物，例如顾客、职工、供货单位等。它表达该系统数据的外部来源或去处。外部实体也可以是另外一个信息系统。

为了避免在数据流图上出现线条交叉，同一个外部实体或文件均可在不同位置多次出现，这时要在外部实体符号的右下方画小斜线或在文件符号左边画竖线，以示重复。

从以上内容可以看出，数据流图可通过基本符号直观地表示系统的数据流程和加工、存储等过程。但它不能表达每个数据和加工的具体、详细的含义，这些信息需要在“数据字典”和“加工说明”中表达。

3.3.6 数据字典

数据流图描述了现行系统的总体框架结构，在数据流图的基础上一，还需要对其中的每个数据流、文件和数据项加以描述，将这些定义所组成的集合称为数据字典。

在结构化分析中，数据字典的作用是对数据流上每个成分给以定义和说明，目的是进行数据分析和归档，同时也是数据库/数据文件设计的依据。除此之外，数据词典还要对系统分析中其他需要说明的问题进行定义和说明。

数据字典是系统逻辑模型的详细、具体说明，是系统分析阶段的重要文件，也是内容丰富、篇幅很大的文件，编写数据字典是一项十分重要而繁重的任务。

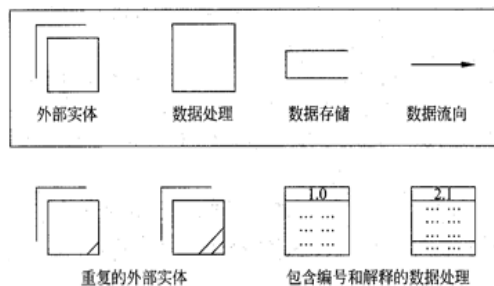


图 3.13 数据流图符号示例图

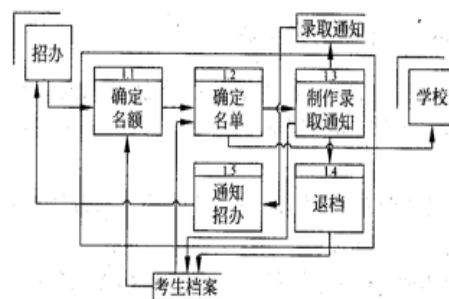


图 3.14 高考招生录取部分数据流图

编写数据字典的基本要求是：

- 对数据流图上各种成分的定义必须明确、易理解、惟一。
- 命名、编号与数据流图一致，必要时(如计算机辅助编写数据字典时)可增加编码，方便查询搜索、维护和统计报表。
- 符合一致性与完整性的要求，对数据流图上的成分定义与说明无遗漏项。数据字典中无内容重复或内容相互矛盾的条目。
- 格式规范、风格统一、文字精炼，数字与符号正确。

数据字典可以用人工方式建立，事先印好表格，填好后按一定顺序排列，就是一本字典；也可以建立在计算机内，数据字典实际上是关于数据的数据库。

3.3.6.1 数据字典项目描述内容举例

数据字典中有 6 类条目：数据元素、数据结构、数据流、数据存储、外部实体和处理。不同类型的条目有不同的属性，现分别说明如下。

1. 数据元素

数据元素是最小的数据组成单位，也就是不可再分的数据单位，如学号、姓名等。对每个数据元素，需要描述以下属性：

- **名称：**数据元素的名称要尽量反映该元素的含义，便于理解和记忆。
- **别名：**一个数据元素可能名称不止一个。若有多个名称，则需加以说明。
- **类型：**说明取值是字符型还是数字型等。
- **取值范围和取值的含义：**指数据元素可能取什么值或每一个值代表的意思。
- **长度：**指出该数据元素由几个数字或字母组成。如学号，按某校现在的编法由 7 个数字组成，其长度就是 7 个字节。

2. 数据结构

数据结构的描述重点是数据之间的组合关系，即说明这个数据结构包括哪些成分。一个数据结构可以包括若干个数据元素或(和)数据结构。这些成分中有三种特殊情况：

- **任选项：**这是可以出现也可以省略的项，用“[]”表示，如[曾用名]是任选项。
- **必选项：**在两个或多个数据项中，必须出现其中的一个称为必选项。例如，任何一门课程是必修课或选修课，二者必居其一。必选项的表示办法，是将候选的多个数据项用“{ }”括起来。
- **重复项：**即可以多次出现的数据项。例如一张订单可订多种零件，每种零件有品名、规格、数量，这些属性用“零件细节”表示。在订单中，“零件细节”可重复多次。

3. 数据流

关于数据流，在数据字典中描述以下属性：

- **数据流的来源。**数据流可以来自某个外部实体、数据存储或某个处理。
- **数据流的去处。**某些数据流的去处可能不止一个，如果有多个，则每个去处都要说明。
- **数据流的组成。**一个数据流可包含一个或多个数据结构。若只含一个数据结构，注意名称的统一，以免产生二义性。
- **数据流的流量。**指单位时间(每日、每小时等)里的传输次数。可以估计平均数或最高、最低流量各是多少。
- **高峰时的流量。**

4. 数据存储

数据存储的条目主要描写该数据存储的结构及有关的数据流、查询要求。有些数据存储的结构可能很复杂，如“学籍表”，包括学生的基本情况、学生动态、奖惩记录、学习成绩、毕业论文成绩等，其中每一项又是数据结构。这些数据结构有各自的条目分别加以说明，因此在“学籍表”的条目中只需列出这些数据结构，而不要列出这些数据结构的内部构成。数据流图是分层的，下层图是上层图的具体化。同一个数据存储可能在不同层次的图中出现。描述这样的数据存储，应列出最低层图中的数据流。

5. 外部实体

外部实体是数据的来源或去向。因此，在数据字典中关于外部实体的条目，主要说明外部实体产生的数据流和传给该外部实体的数据流，以及该外部实体的数量。外部实体的数量对于估计本系统的业务量有参考作用，尤其是关系密切的主要外部实体。

6. 处理

需要在数据字典中描述处理框的编号、名称、功能的简要说明及有关的输入、输出。关于功能的描述，使人能有一个较明确的概念，知道这一框的主要功能。功能的详细描述，还要用“小说明”进一步描述。

3.3.6.2 数据量统计及分析

企业信息化的过程中，国内外许多企业在其发展过程中逐渐形成了多种独立的应用(子)系统；另外，一些公司由地点上分布的多个子公司或部门组成，子公司或部门独立使用着各自的业务处理系统，而这些子系统往往是异质的。当企业或公司需要企业范围内的全局应用时，直接在繁杂的多个子系统上实施是很困难或不可能的。所以，日常的数据量统计和分析是必做的工作。

企业中除了对业务数据量进行增、删、改等事务处理操作和简单的统计汇总以外，高层管理者还要使用数据(历史的、现在的)进行各种复杂分析以支持决策。从大量的数据量中进行统计分析，获取信息，要求系统在保存大量的历史数据量的基础上，还要进行复杂的分析处理(每次处理涉及大量数据)。这些功能对于频繁操作性处理的信息系统而言，将成为沉重的负担。目前该项工作，一般由专门的数据仓库软件负责。

3.3.7 基本加工处理描述

1. 基本加工处理概述

如何对数据流图中的基本加工进行描述，是结构化分析的关键部分。我们把对基本加工的描述称为编写“加工说明”。

这里讲的“加工说明”是指对数据流图中功能单元(不能再作分解的加工)的描述，而对数据流图中其他加工则可以没有加工说明。

编写加工说明的时候，有如下要求：

- 对数据流图中的每个功能单元，必须有一个加工说明。
- 加工说明必须描述功能单元把输入数据转换为输出数据流的转换规则。
- 每个加工说明必须描述转换的策略，而不是转换的实现细节，即主要描述一个加工“做什么”，而不是用程序设计语言来描述具体的加工过程。
- “加工说明应力求完整、严密、易于理解。”

2. 结构化语言

人们常用自然语言描述各种问题。自然语言语义丰富、语法灵活，可描述十分广泛而复杂的问题，表达人们丰富的感情和智慧。但自然语言没有严格的规范，理解上容易产生歧义。在信息处理中人们广泛使用的计算机语言，是一种形式化语言，各种词汇均有严格定义，语法也很严格、规范。但使用的词汇限制在很小的范围内，叙述方式繁琐，难以清晰地描述复杂问题。结构化语言的特点介于两者之间，没有严格的语法规定，使用的词汇也比形式化的计算机语言广泛，但使用的语句类型很少，结构规范，表达的内容清晰、准确，易理解，不易产生歧义，适于表达数据加工的处理功能和处理过程。

结构化语言使用的语句类型只有祈使语句、条件语句和循环语句三种。

3. 决策树

如果一个加工中决策或判断的步骤较多，则使用结构化语言时，语句的嵌套层次太多，不便于基本加工的逻辑功能的清晰描述。决策树(Decision Tree)又称判断树，是一种图形工具，适合于描述加工中具有多个决策、每个策略和若干条件有关的逻辑功能，如图 3. 15 所示。

4. 决策表

在基本加工中，如果判断的条件较多，各条件又相互组合、相应的决策方案较多那么用决策树来描述，树的结构就比较复杂，图中各项注释也比较繁琐。决策表又称为判断表，为描述这类加工逻辑提供了表达清晰、简洁的手段。决策表也是一种图形工具，呈表格形。

编制决策表时，首先要明确加工的功能与目标，然后要识别影响决策的各项因素，列出这些因素可能出现的状态，并制定出决策的原则。

具体例子参照表 3. 1 和表 3. 2。

3.3.8 建立新系统逻辑模型

新系统逻辑模型是指经分析和优化后，新系统拟采用的管理模型和信息处理方法。因它不同于计算机配置方案和软件结构方案等实体方案，故称为逻辑方案或逻辑模型。

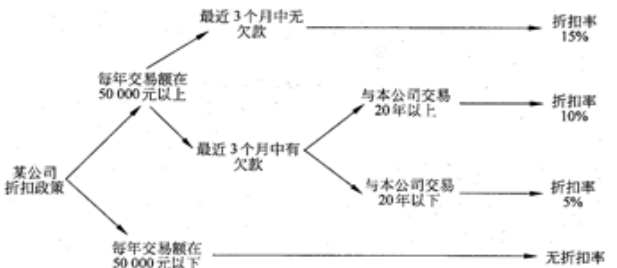


图 3.15 决策树

表 3.2 合并整理后的决策表

不同条件组合	1 (1/2)	2 (3)	3 (4)	4 (5/6/7/8)
C ₁ :交易额在 50000 元以上	Y	Y	Y	N
C ₂ :最近 3 个月无欠款单据	Y	N	N	—
C ₃ :与本公司交易 20 年以上	—	Y	N	—
A ₁ :折扣率 15%	X			
A ₂ :折扣率 10%		X		
A ₃ :折扣率 5%			X	
A ₄ :无折扣率				X

表 3.1 公司的折扣政策

不同条件组合	1	2	3	4	5	6	7	8
C ₁ :交易额在 50000 元以上	Y	Y	Y	Y	N	N	N	N
C ₂ :最近 3 个月无欠款单据	Y	Y	N	N	Y	Y	N	N
C ₃ :与本公司交易 20 年以上	Y	N	Y	N	Y	N	Y	N
A ₁ :折扣率 15%	X	X						
A ₂ :折扣率 10%			X					
A ₃ :折扣率 5%				X				
A ₄ :无折扣率					X	X	X	X

注：Y—满足条件，N—不满足条件，X—选中的

详细调查、收集资料，对现行系统进行业务流程分析、数据流程分析都是为最终确立新系统的逻辑模型做准备。所以说新系统逻辑模型的建立是系统分析阶段的最终结果，是系统设计和系统实施的依据。

从现行系统的逻辑模型到新系统逻辑模型的转换，一般可分为两步，首先分析新系统与现行系统在逻辑上的差别，在此基础上建立起新系统的初步逻辑模型；然后对该模型进行补充和完善，建立最终模型。

对新系统的信息处理方案的确定包括如下几部分。

1. 新系统组织机构及业务流程

首先要做的，就是确定新系统组织机构和业务流程。主要工作有：

- 删除或合并那些多余的或重复的处理过程。
- 说明哪些业务处理过程进行了优化和改动，以及改动的原因是什么，改动后将带来哪些好处等问题。
- 给出最后确定的业务流程图。
- 指出在业务流程图中哪些部分新系统可以完成，哪些部分需要用户完成。

2. 新系统目标及范围

新系统目标及范围的确立，服从于组织的整体目标和管理决策活动的需要。根据组织的整体业务目标和发展战略，确定新系统的发展战略，明确组织总的信息需求，制定信息建设总计划，其中包括确定新系统的总体发展目标、功能、大致规模和粗略估计所需资源，并根据需求的轻、重、缓、急程度及资源和应用环境的约束，确定新系统的范围，以及实现新系统的步骤。

3. 新系统逻辑结构及数据分布

逻辑结构及数据分布的方案有以下几种：从信息资源管理的集中程度考虑有集中式系统(Centralized Systems)和分布式系统(Distributed Systems)；从信息处理的方式考虑主要有批处理方式(Batch Processing)和联机处理方式(Online Processing)。

信息系统应采用的逻辑结构和数据分布方式，应视系统的需求、技术条件来设定，详细内容可参见本书第10章和第11章的相关内容。

4. 新系统数据流图及数据字典

应该确定合理的数据和数据流程。主要工作有：

- 确认最终的数据指标体系和数据字典，如指标体系是否全面合理，数据精度是否满足要求等。
- 删去或合并多余的或重复的数据处理过程。
- 说明哪些数据处理过程进行了优化和改动，以及改动的原因是什么，有哪些好处等；给出最后确定的数据流图，指出在数据流图中哪些部分新系统可以完成，改动后哪些部分需要用户完成。

5. 新系统数据分析及数据量统计

在新系统中，数据分析时要分清数据的种类。有企业的内部数据，也有外部数据；有固定数据，也有随机数据；有结构化数据，也有非结构化数据。

新系统中的数据，还可以按照职能来判断是属于生产管理类数据、销售管理类数据、物资管理类数据、财务管理类数据还是人事管理类数据。

生产管理的职能是计划、调度和统计。根据市场预测的信息、企业目标和生产能力，对企业的人力、材料、设备、资金等进行全面合理的安排，按产品的品种和规格制定生产进度计划和物资、人力、资金需求计划。生产管理类的数据主要来源于每日的生产统计，如产品产值、质量物耗指标、设备运行情况的统计等。

销售管理的职能主要是根据市场需要组织生产经营活动。销售管理类的数据主要来源于国家的政策、市场需求、用户订货情况、企业利润指标要求和企业生产情况等，另外，还有定期的销售分析等。

物资管理职能包括物资需求计划的制定、订货和库存管理，企业内空缺物资分配的管理等。物资管理类的数据主要来源于物资需求计划，库存及物资供应状况，物资净需求量，订货批量，定期的各种库存、费用、材料消耗报表等。

财务管理职能主要是以资金形式来反映企业的财务状况；对各类资金进行管理和控制，以加速流动资金周转，提高资金的使用效率；进行利润管理，为企业提供更准确的利润信息；制定合理的财务收支计划，为企业资金的筹集和运用提供可靠依据。财务管理类数据主要来源于企业内部的各种单据、票证。

人事管理把各种人员按照某种关系组织在一起从事生产经营活动，为社会提供产品或服务。现代的人事管理职能，不仅进行人事档案和劳动工资定额的日常管理，还应把人力资源的合理使用和合理发挥作为目标来实现。而人事类数据则主要来源于职工的各种基本信息，以及经营活动中的各种数据。

6. 新系统实施策略及计划

系统实施是系统开发的最后阶段，是将系统设计的结果进行物理实现。在此期间，将投入大量的人力、物力及占用较长的时间进行硬件购置与安装、程序设计、程序和系统调试、人员培训、系统转换、系统管

理等一系列工作，这个过程即系统实施阶段。

在系统分析与系统设计的阶段中，开发人员为新系统设计了逻辑模型和物理模型。系统实施阶段的目标，就是把系统设计的物理模型转换成可实际运行的新系统。

在企业信息化中，有些企业可利用成熟的管理软件系统，并在咨询公司的帮助下，实施新系统的策略。在这种情况下，新系统实施的内容，主要是实现企业业务内容与管理软件系统之间的整合。

具体实施步骤，可以参照表 3. 3。

7.新系统投资预算及策略

对新系统进行投资，需要预先估算新系统开发、实施和运行所需的费用以及新系统的效益，将投资和预期效益进行比较，以说明在经济上是否合算。

新系统所需要的投资包括：计算机硬件设备费用、软件费用、人员费用(开发费，操作人员、维护人员的工资及培训费用)；材料费用(打印纸、软盘、水、电等的费用)和其他费用(外部设备费用、电源设备费用、机房费用)。

投资策略，实际上就是要利用有限的资金，在这些需投资的方面进行合理的组合。例如在投资有限的情况下，如果是采用商用软件，则是一次购进所有模块，还是选择急需的模块(如仓库管理、财务管理等)分步实施，这些就是投资决策需要考虑斟酌的地方。

表 3.3 新系统实施步骤

阶段	步骤内容
第一阶段：组织及概念设计	<p>目的：确定项目工作组织，培训项目小组，设定测试系统及测试用户，开发概念设计</p> <ul style="list-style-type: none">• 项目准备：项目组织结构，项目初始计划，项目示意图，项目小组成员培训计划• 系统环境设置• 项目小组培训• 定义功能及流程：概念设计，系统设置需求• 概念设计质量检验：有效性报告
第二阶段：详细设计及系统设置	<p>目的：按您的需求在投产准备用户中设置系统，用您的设置测试单独的商业流程，系统与您的其他应用系统的合成，展示经设置的系统以备讨论通过</p> <ul style="list-style-type: none">• 建立总体设置：总体系统设置• 建立公司结构：公司结构的系统设置• 建立主数据库：主数据的系统设置，主数据抽样• 建立功能与流程：流程/功能设置，流程/功能数据抽样• 建立报告：系统报告• 建立备份管理：设计备份管理，备份的系统设置• 建立权限管理：权限设计，权限，轮廓，用户主记录• 终结测试：终结测试报告• 应用系统质量检测：有效性报告
第三阶段：投入运行准备	<p>目的：计划投入运行，安装所需硬件及软件，产生用户文件，培训用户，建立您的系统管理，数据及系统设置传输入运行系统</p> <ul style="list-style-type: none">• 产生投入运行计划• 产生用户文件• 设置运行环境• 培训用户：培训资料，培训记录• 建立系统管理：系统管理材料• 运行系统质量检测
第四阶段：投入运行	<ul style="list-style-type: none">• 投入运行支持：解疑组织• 系统使用优化：用户文件，系统设置，项目终结报告

3. 3.9 系统分析报告

完成整个系统分析阶段的工作后，作为工作成果，应提交一份完整的系统分析报告。系统分析报告一经确认，就成为具有约束力的指导性文件，成为下一阶段系统设计工作的依据和今后验收目标系统的检验标准。系统分析报告形成后必须组织各方面的人员(包括组织的领导、管理人员、专业技术人员、系统分析人员等)一起对已经形成的方案进行论证，尽可能发现其中的问题和不足。对于有争论的问题要重新核实当初的原始调查资料或进一步地深入调查研究，对于重大的问题甚至可能需要调整或修改系统目标，重

新进行系统分析。

在系统分析报告中，数据流图、数据字典和加工说明这 3 部分是主体，是系统分析报告中必不可少的组成部分。而其他各部分内容，则应根据所开发目标系统的规模、性质等具体情况酌情选用，不必生搬硬套。总之，系统分析报告必须简明扼要、抓住本质，反映出目标系统的全貌和开发人员的设想。

系统分析报告主要有以下 3 个作用：

- 描述了目标系统的逻辑模型，作为开发人员进行系统设计和实施的基础。
- 作为用户和开发人员之间的协议或合同，为双方的交流和监督提供基础。
- 作为目标系统验收和评价的依据。

因此，系统分析报告是系统开发过程中的一份重要文档，必须完整、一致、精确且简明易懂。

一份完整的系统分析报告应该包括下述内容。

①组织情况概述。

• 对分析对象的基本情况作概括性的描述，包括组织的结构、组织的目标、组织的工作过程和性质、业务功能。

- 系统与外部实体(其他系统或机构)间有哪些物质以及信息的交换关系和联系。
- 参考资料和专门术语说明。

②现行系统概述。

• 现行系统现状调查说明。通过现行系统的组织结构图、数据流图、概况表等，说明现行系统的目标、规模、主要功能、组织机构、业务流程、数据存储和数据流，以及存在的薄弱环节。

- 系统需求说明。用户要求以及现行系统主要存在的问题等。

③系统逻辑模型。

• 新系统拟定的业务流程及业务处理工作方式。提出明确的功能目标、并与现行系统进行比较分析，重点要突出计算机处理的优越性。

• 新系统拟定的数据指标体系和分析优化后的数据流程，各个层次的数据流图、数据字典和加工说明，以及计算机系统将完成的工作部分。

- 出错处理要求。
- 其他特性要求。例如系统的输入输出格式、启动和退出等。
- 遗留问题。根据目前条件，暂时不能满足的一些用户要求或设想，并提出今后解决的措施和途径。

④新系统在各个业务处理环节拟采用的管理方法、算法或模型。

⑤与新的系统相配套的管理制度和运行体制的建立。

⑥系统设计与实施的初步计划。

• 工作任务的分解。根据资源及其他条件确定各子系统开发的先后次序，在此基础上分解工作任务，落实到具体组织或个人。

- 根据系统开发资源与时间进度估计，制定进度安排计划。
- 预算。对开发费用的进一步估计。

⑦用户领导审批意见。

3.4 系统设计

3.4.1 系统设计概述

系统设计是信息系统开发过程中另一个重要阶段。这一阶段中，要根据前一阶段系统分析的结果，在已经获得批准的系统分析报告的基础上，进行新系统设计。

系统设计的主要目的就是为系统制定蓝图，在各种技术和实施方法中权衡利弊，精心设计，合理使用各种资源，最终勾画出新系统的详细设计方案。但是，实际情况往往与主观设定存在差距，

项目开发过程并不总是能按总体计划分阶段顺利推进，甚至造成反复，究其原因有：

①传统方法认为“系统设计之前，用户的所有的需求都能被预先定义”。事实上，在实际开发中，并非所有的需求都能预先加以定义，虽然通过全面系统调查，对原系统业务流程做了仔细研究，能导出新系统的逻辑模型，但这些模型中用户需求定义并不完全确定，也并不完善，其原因包括两个方面：

• 开发人员缺少足够的专业知识。在调查过程中，由于开发者对业务并不熟悉，以至于在系统分析中往往会忽略某些应该重视的用户需求，用户与开发人员之间彼此信息交流上存在障碍。

• 用户尚缺少足够的技术知识。难以实现业务与技术的有机结合，因此较难准确表达在特定技术条件下的业务活动的实现方法和途径，由此导致以需求定义为基础的系统静态逻辑模型在功能上存在较大的片面性。

②在生命周期法中，系统分析通常用数据流图、数据字典、判断表等工具来描述目的系统的逻辑模型，这些文字和图形工具被认为可以充分反映新系统的逻辑功能。实践证明，在系统开发过程中，由于开发人员和用户双方在认识上的差距，对共同的约定往往会有理解上的差异。因此在系统设计阶段之前建立起来的逻辑模型并不总能充分描述新系统功能，而通过各种静态工具与用户进行信息交流，其效果却十分有限，还达不到修正需求的目的。

③生命周期法将开发工作严格划分为几个不同阶段，并严格分离，即后一阶段工作必须在前一阶段结束才能进行，把各个阶段工作的变化幅度限制在一个特定的范围内。由于各阶段工作间可能存在反复，因此阶段的严格划分实际上很难做到质量的保证。这就造成了理论与实践间的矛盾，这也是传统方法所存在的不足。

3.4.1.1 系统设计的内容和步骤

系统设计阶段的主要依据是系统分析报告和开发者的经验。系统设计的主要内容包括新系统总体结构设计、代码设计、输出设计、输入设计、处理过程设计、数据存储设计、用户界面设计和安全控制设计等。

系统设计的基本任务大体上可以分为两个步骤。

(1)把总任务分解成许多基本的、具体的任务。这些具体任务合理地组织起来构成总任务，称为总体结构设计(architectural design)，又称为概要结构设计(preliminary design)。其基本任务是：将系统划分成模块，决定每个模块的功能，决定模块的调用关系，决定模块的界面，即模块间信息的传递。

总体结构设计是系统开发过程中很关键的一步。系统的质量及一些整体特性基本上是由这一步决定的。系统越大，总体结构设计的影响越大。认为各个局部都很好，组合起来就一定好的想法是不实际的，因为部分最优，并不能保证总体最优。

(2)为各个具体任务选择适当的技术手段和处理方法，即详细设计。内容包括代码设计，输出设计，输入设计，处理过程设计，数据存储设计，用户界面设计，安全控制设计。

系统设计的结果是一系列的系统设计文件，这些文件是物理实现一个信息系统(包括安装硬件设备和编制软件程序)的重要基础。

3.4.1.2 系统结构设计的原则

为保证总体结构设计的顺利完成，主要应遵循以下几条原则：

①分解—协调原则。整个系统是一个整体，具有整体目的和功能。但这些目的和功能的实现又是由相互联系各个组成部分共同工作的结果。解决复杂问题的一个很重要的原则就是把它分解成多个小问题分别处理，在处理过程中根据系统总体要求协调各部门的关系。

在系统中，应按以下要求分解：

- 按系统的功能进行分解。
- 按管理活动和信息运动的客观规律分解。
- 按信息处理的方式和手段分解。
- 按系统的工作规程分解。
- 按用户工作的特殊需要分解(如按保密的要求)。
- 按开发、维护和修改的方便性分解。

协调的依据主要是：

- 目的协调。
- 工作进程协调。
- 工作规范和技术规范协调。
- 信息协调(指信息的提供和收回)。
- 业务内容协调(如某些业务指标的控制)。

②自顶向下的原则。首先抓住系统总的功能目的，然后逐层分解，即先确定上层模块的功能，再确定下层模块的功能。—

③信息隐蔽、抽象的原则。上层模块只规定下层模块做什么和所属模块间的协调关系，但不规定怎么做，以保证各模块的相对独立性和内部结构的合理性，使得模块与模块之间层次分明，易于理解，易于实施，易于维护。

④一致性原则。要保证整个软件设计过程中具有统一的规范、统一的标准和统一的文件模式等。

⑤明确性原则。每个模块必须功能明确，接口明确，消除多重功能和无用接口。

⑥模块之间的藕合尽可能小，模块内部组合要尽可能紧凑。

⑦模块的扇入系数和扇出系数要合理。一个模块直接调用其他模块的个数称为该模块的扇出系数；反

之，一个模块被其他模块调用时，直接调用它的模块个数称模块的扇入系数。模块的扇入、扇出系数必须适当。经验表明，一个设计的好的系统的平均扇入、扇出系数通常是 3 或 4，一般不应超过 7，否则会引起出错概率的增大。但菜单调用型模块扇入与扇出系数可以大一些，公用模块扇入系数可以大一些。

⑧**模块的规模适当**。过大的模块常常使系统分解得不充分，其内部可能包含了若干部分的功能，因此有必要进一步把原有的模块分解成若干功能尽可能单一的模块。但分解也必须适度，因为过小的模块有可能降低模块的独立性，造成系统接口的复杂。一条有益的经验是，一个模块的规模最好是程序系数限制在 1-2 页纸内，这样的模块易于编制、维护、修改。

3.4.2 系统总体结构设计

系统总体结构设计是要根据系统分析的要求和组织的实际情况来对新系统的总体结构形式和可利用的资源进行大致设计，这是一种宏观、总体上的设计和规划。下面介绍系统总体结构设计的主要内容。

3.4.2.1 子系统划分

1. 子系统划分的原则

为了便于今后系统开发和系统运行，子系统的划分应遵循如下几点原则：

- **子系统要具有相对独立性**。子系统的划分，必须使得子系统内部功能、信息等各方面的凝聚性较好。子系统独立可以减少子系统间的相互影响，有利于多人分工开发不同的模块，从而提高软件产品的生产率，保证软件产品的质量，同时也增强了系统的可维护性和适应性。

- **子系统之间数据的依赖性尽量小**。子系统之间的联系要尽量减少，接口要简单明确。一个内部联系强的子系统对外部的联系必然很少，所以划分的时候，应将联系较多者列入子系统内部，剩余的一些分散、跨度比较大的联系，就成为这些子系统之间的联系和接口。这样划分的子系统，将来调试、维护和运行都是非常方便的。

- **子系统划分的结果应使数据冗余较小**。如果把相关的功能数据分布到各个不同的子系统中，则会有大量的原始数据需要调用，大量的中间结果需要保存和传递，大量计算工作将要重复进行。从而使得程序结构紊乱，数据冗余，不但给软件编制工作带来很大的困难，而且系统的工作效率也大大降低了。

- **子系统的设置应考虑今后管理发展的需要**。子系统的设置光靠上述系统分析的结果是不够的，因为现存的系统由于各种原因，很可能没有考虑到一些高层次管理决策的要求。

- **子系统的划分应便于系统分阶段实现**。信息系统的开发是一项较大的工程，它的实现一般都要分期分批进行，所以子系统的划分应能适应这种分期分批的实施。另外，子系统的划分还必须兼顾组织结构的要求。

- **子系统的划分应考虑到各类资源的充分利用**。一个适当的子系统划分应该既考虑有利于各种设备资源在开发过程中的搭配使用，又考虑到各类信息资源的合理分布和充分使用，以减少系统对网络资源的过分依赖，减少输入、输出、通信等设备压力。

2. 系统划分方法的分类

目前有关子系统划分方法如表 3.4 所示。

按功能划分是目前最常用的一种划分方法。按业务处理顺序划分要依据业务流程分析的结果，这种划分方式在一些时间和处理过程顺序特别强的系统中常常采用。按数据拟合程度划分是指按数据而不是按内部集中来划分，这种划分方式的子系统内部聚合力强，外部通信压力小。严格地说，按业务处理过程划分子系统，不是一种很好的方法，但在某些系统开发资源限制较大的场合，特别是要分段实现开发工作时，不得已被采用。最后两种划分方法指的是按业务处理的时间关系或业务展开的环境条件来对系统进行划分，严格地说这也不是太合理的划分方法，但在某些特定的场合也有这种情况。

表 3.4 子系统划分方法

方法分类	划分方式	连接形式	可修改性	可读性	紧凑性
功能划分	按业务处理功能划分	好	好	好	非常好
顺序划分	按业务先后顺序划分	好	好	好	非常好
数据拟合	按数据拟合的程度来划分	好	好	较好	较好
过程划分	按业务处理过程划分	中	中	较差	一般
时间划分	按业务处理时间划分	较差	较差	较差	一般
环境划分	按实际环境和网络分布划分	较差	较差	较差	较差

3.4.2.2 子系统结构设计

子系统结构设计任务是确定划分后的子系统的模块结构，并画出模块结构图。这个过程中必须考虑以下几个问题：

- 每个子系统如何划分成多个模块。
- 如何确定子系统之间、模块之间传送的数据及其调用关系。
- 如何评价并改进模块结构的质量。
- 如何从数据流图导出模块结构图。

具体内容将在 3.4.3 节进行说明。

3.4.2.3 网络设计

如何将规划中的各个子系统从内部用局域网连接起来，以及今后系统如何与外部系统相连接，这就是网络设计要解决的问题。这里所谓的设计是根据实际系统的需求，考虑如何配置和选用网络产品。

网络设计首先要根据系统的要求选择网络的结构。然后根据系统结构划分的结果,安排网络和设备分布,再根据物理位置来考虑联网布线和配件,最后就是根据实际业务的要求划定网络各结点的级别、管理方式、数据读写的权限、选择相应的软件系统等。有关网络的相关知识,可参见本书第 10 章。

3. 4. 2. 4 硬件设备及配置

在确定了系统的划分后,就可以考虑各子系统的设备,即计算机和网络设备的配置问题,以及如何将这些分布的设备和任务、功能、数据资源等集中统一管理。当前流行的系统有集中式和分布式两大类型,采用哪种方式应视系统的功能需求和特点而定,它和系统的性质和业务类型有关,是一类专业性很强的技术,可参见本书第 10 章和第 11 章有关内容。

总之,在满足实际业务需要的前提下,只要资金许可,应购置技术上成熟、性能价格比较高的系统设备。一般可根据如下几个方面进行考虑:技术上是否可靠;维修是否很方便;纵向,新老系统能否兼容;横向,本系统外系统能否兼容;非标准的系列不宜选取,选用用户熟悉的软件、硬件产品;可扩充性如何;对工作环境的要求(如温度、湿度、防尘度等)是否很高;性能价格比如何。

3. 4. 3 系统模块结构设计

3.4.3.1 模块的概念

模块是组成系统的基本单位,它的特点是可以组合、分解和更换。系统中任何一个处理功能都可以看成是一个模块。根据模块功能具体化程度的不同,可以分为逻辑模块和物理模块。在系统逻辑模型中定义的处理功能可视为逻辑模块。物理模块是逻辑模块的具体化,可以是一个计算机程序、子程序或若干条程序语句,也可以是人工过程的某项具体工作。

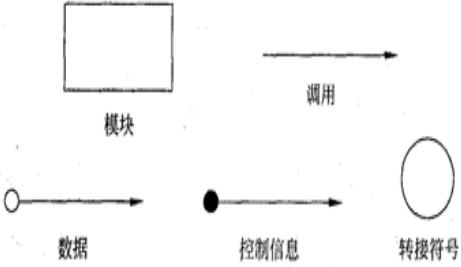


图 3.16 模块结构图的基本符号

一个模块应具备以下 4 个要素:

- **输入和输出。**模块的输入来源和输出去向都是同一个调用者,即一个模块从调用者那里取得输入,进行加工后再把输出返回给调用者。
- **处理功能,**指模块把输入转换成输出所做的工作。
- **内部数据,**指仅供该模块本身引用的数据。
- **程序代码,**指用来实现模块功能的程序。

前两个要素是模块的外部特性,即反映了模块的外貌。后两个要素是模块的内部特性。在结构化设计中,主要考虑的是模块的外部特性,其内部特性只做必要了解,具体的实现将在系统实施阶段完成。

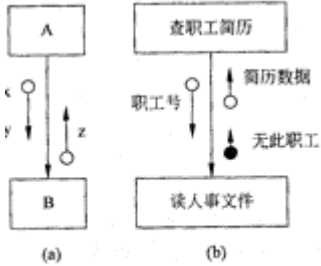


图 3.18 模块间的数据传递

3. 4. 3.2 模块结构图

为了确保系统设计工作的顺利进行,结构设计应遵循如下原则:

- 所划分的模块其内部的凝聚性要强,模块之间的联系要少,即模块具有较强的独立性。
- 模块之间的连接只能存在上下级之间的调用关系,不能有同级之间的横向联系。
- 整个系统呈树状结构,不允许有网状结构或交叉调用关系出现。
- 所有模块(包括后继 IPO 图)都必须严格地分类编码并建立归档文件。

模块结构图主要关心的是模块的外部属性,即上下级模块、同级模块之间的数据传递和调用关系,并不关心模块的内部。

模块结构图是结构化设计中描述系统结构的图形工具。作为一种文档,它必须严格地定义模块的名字、功能和接口,同时还应当在模块结构图上反映出结构化设计思想。模块结构图由模块、调用、数据、控制和转接等 5 种基本符号组成,如图 3.16 所示,现说明如下。

• **模块。**这里所说的模块通常是指用一个名字就可以调用的一段程序语句。长方形中间标上能反映模块处理功能的模块名字。

• **调用。**箭头总是由调用模块指向被调用模块,但是应该理解成被调用模块执行后又返回到调用模块。

如果一个模块是否调用一个从属模块,决定于调用模块内部的判断条件,则该调用称为模块间的判断调用,采用菱形符号表示。如果一个模块通过其内部的循环功能来循环调用一个或多个从属模块,则该调用称为循环调用,用弧形箭头表示。判断调用和循环调用的表示方法如图 3.17 所示。

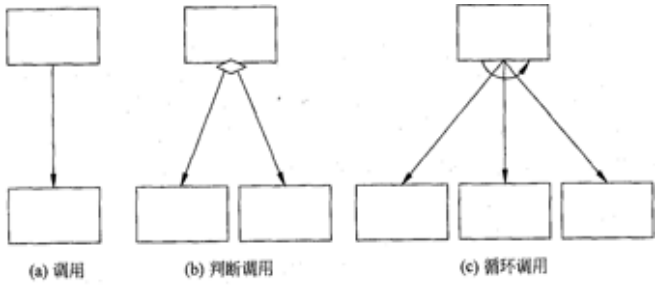


图 3.17 模块调用示例

• **数据**。当一个模块调用另一个模块时，调用模块可以把数据传送到被调用模块供处理，而被调用模块又可以将处理的结果数据送回到调用模块。在模块之间传送的数据，使用与调用箭头平行的带空心圆的箭头表示，并在旁边标上数据名。图 3. 18(a) 表示模块 A 调用模块 B 时, A 将数据 x、y 传送给 B, B 将处理结果数据 z 返回给 A。

• **控制信息**。模块间有时还必须传送某些控制信息。例如，数据输入完成后给出的结束标志，文件读到末尾所产生的文件结束标志等。控制信息与数据的主要区别是前者只反映数据的某种状态，不必进行处理。图 3. 18(b) 中“无此职工”就是用来表示送来的职工号有误的控制信息。

• **转接符号**。当模块结构图在一张纸上画不下，需要转接到另一张纸上，或为了避免图上线条交叉时，都可使用转接符号，圆圈内加上标号，如图 3. 19 所示。

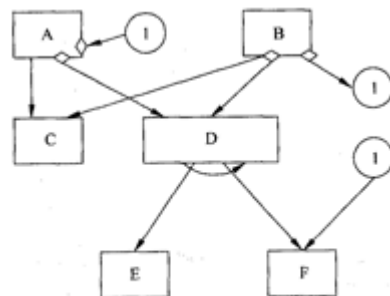


图 3. 19 转接符号的使用

3. 4.3.3 模块的变换型分析与事务型分析

一个系统的模块结构图一般有两种标准形式，变换型模块结构和事务型模块结构。

变换型模块结构描述的是变换型系统。变换型系统由 3 部分组成：输入、数据加工(中心变换)和输出，它的功能是将输入的数据经过加工后输出，如图 3. 20 所示，变换型系统工作时，首先主模块受到控制，然后控制沿着结构逐层达到底层的输入模块，当底层模块输入数据 A 后, A 由下至上逐层传送，逐步由“物理输入”变成“逻辑输入”C，接着在主控模块控制下, C 经中心变换模块转换成逻辑输出 D, D 再由上至下逐层传送，逐步把“逻辑输出”变成“物理输出”E。这里的“逻辑输入”和“逻辑输出”分别为系统主处理的输入数据流和输出数据流，而“物理输入”和“物理输出”是指系统输入端和系统输出端的数据。

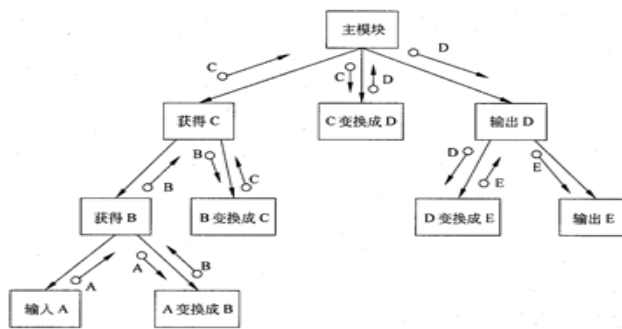


图 3. 20 变换型模块结构

事务型系统由 3 层组成：事务层、操作层和细节层。它的功能是对接收的事务按其类型选择某一类事务处理，如图 3. 21 所示。

事务型系统在工作时，主模块将按事务的类型选择调用某一事务处理模块，事务处理模块又调用若干个操作模块，而每个操作模块又调用若干个细节模块。在实际系统中，由于不同的事务可能有共同的操作，而不同的操作有可能有共同的细节，因此事务型系统的操作模块和细节模块可以达到一定程度的共享。

这两种典型的结构分别可通过“变换分析”和“事务分析”技术，导出“变换型”和“事务型”初始的模块结构图。这两种方法的思想是首先设计顶层模块，然后自顶向下，逐步细化，最后得到一个满足数据流图所表示的用户要求的系统模块结构图，即系统的物理模型。

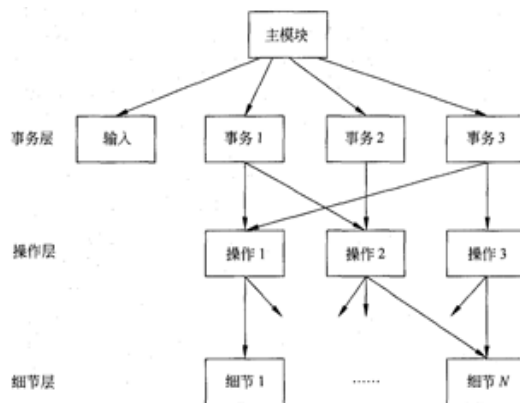


图 3. 21 事务型模块结构

下面分别讨论通过“变换分析”和“事务分析”，导出“变换型”和“事务型”初始结构图的技术。在实际应用中，这两种分析往往交替使用。

1. 变换型分析

变换型分析过程可以分为 3 步。

(1) 找出系统底层逻辑输入、主加工和逻辑输出。

如果设计人员经验丰富，又熟悉系统分析说明书，则容易确定系统的主加工。例如，几股数据流的汇合处往往就是系统的主加工。若一时不能确定哪儿是主加工，可以用下面的方法先确定哪些数据流是逻辑输入，哪些数据流是逻辑输出：

- 从物理输入端开始，一步步向系统的中间移动，直至这样一个数据流：它已不能再被看作系统的输入，则它的前一个数据流就是系统的逻辑输入。
- 同理，从物理输出端开始，逆数据流方向向中间移动，可以确定系统的逻辑输出。
- 介于逻辑输入与逻辑输出之间的加工就是主加工。

当然，实际的数据流图往往比这个例子复杂，输入、输出数据流都可能多个。这时，需要对每个输入、输出数据流逐个进行分析，确定相应的逻辑输入、逻辑输出。处于这些逻辑输入、逻辑输出之间的处

理框架就是主加工。主加工可能包括数据流图中的多个处理框。

(2)设计顶层模块和第一层模块。

找到主加工之后，遵照“自顶向下、逐步细化”的原则，设计各层的模块。每创建一个模块必须确定该模块的外部特征：模块的功能及与其他模块的界面(调用时传送的信息)。为每个模块起一个名字，这个名字应当恰如其分地反映出这个模块的功能。

系统的主加工就是系统的顶层模块，其功能就是整个系统的功能。

第一层模块按输入、变换、输出等分支来处理：为每一个逻辑输入设计一个输入模块，其功能是为顶层模块提供相应的数据；为每一个逻辑输出设计一个输出模块，它的功能即是输出顶层模块的输出信息；为主加工设计一个变换模块，它的功能就是将逻辑输入变换成逻辑输出。第一层模块与顶层模块之间传送的数据应该同数据流图相对应。

(3)设计顶层模块和第一层模块。

对输入、变换、输出模块逐个分解，便可得到初始结构图。

输入模块是为系统提供逻辑输入，一般要进行变换，先确定实现最后变换的变换模块。这个变换模块显然又需要某些输入，对每个这样的输入，对应一个新的输入模块。用类似的方法依次分解下去，直到最终的物理输入为止。对输出模块的分解与上面的办法相似。

2.事务型分析

事务型分析也是按“自顶向下，逐步细化”的原则进行。先设计主模块，其功能就是整个系统的功能。下面有一个“分析模块”和“调度模块”。前者分析事务的类型，后者根据不同的类型调用相应的下层模块。

3.4.3.4 模块的耦合与内聚

一个合理的模块划分，应该是内部联系强，模块间尽可能独立，接口明确、简单，有适当的共用性，要满足“耦合小，内聚大”的原则。

在结构化设计当中，采用“自顶向下，逐步细化”的方法将系统分解成为一些相对独立、功能单一的模块。如何度量模块之间的独立性呢？由于模块内的互相联系越多，模块的独立性就越强，因此，这里再引入模块耦合和内聚的概念。

耦合就是表示模块之间联系的程度。紧密耦合表示模块之间联系非常强，松散耦合表示模块之间联系比较弱，非耦合则表示模块之间无任何联系，是完全独立的。

内聚则用来表示模块内部各组成成分之间的联系程度。一般说来，在系统中各模块的内聚越大，则模块间的耦合越小。但这种关系并不是绝对的。耦合小使得模块间尽可能相对独立，从而各模块可以单独开发和维护。内聚大使得模块的可理解性和维护性大大增强。因此，在模块的分解中应尽量减少模块的耦合度，力求增加模块的内聚度。

模块的耦合方式有 3 种，如图 3.22 所示。

• **数据耦合**。如果两个模块之间的通信信息是若干数据项，则这种耦合方式称为数据耦合。例如，为了计算实发工资，“计算工资”模块必须把工资总额和扣款数传输给“计算实发工资”模块，而“计算实发工资”模块在计算出实发工资后又将实发工资送回到“计算工资”模块。

• **控制耦合**。如果两个模块之间传输的信息是控制信息，则该耦合称为控制耦合。传送的控制信息可分成两类：一类是判定参数，调用模块通过该判定参数的一个值控制被调用模块的工作方式，若判定参数出错的值不同于前者则导致被调用模块按另一种方式工作；另一种是地址参数，调用模块直接转向被调用模块内部的某一地址，这时若改动一个模块则必将影响另一模块。因为控制耦合方式的耦合程度较高，应尽量避免采用地址参数的方式。

• **非法耦合**。两个模块之间，不经过调用关系，彼此直接使用或修改对方的数据这是最差的耦合方式，在结构化设计时决不允许出现这种情况。此外，在程序设计中，应做到各模块只使用自身的局部变量，尽量不使用全局变量，模块之间必不可少的数据联系都必须以参数形式正确指定。

模块的内聚方式有如下几种，其性能比较如图 3.23 所示。

• **巧合内聚**。巧合内聚或称偶然内聚是指模块内各组成部分之间毫无内在联系，整个模块是一种偶然结合，不易修改或维护。

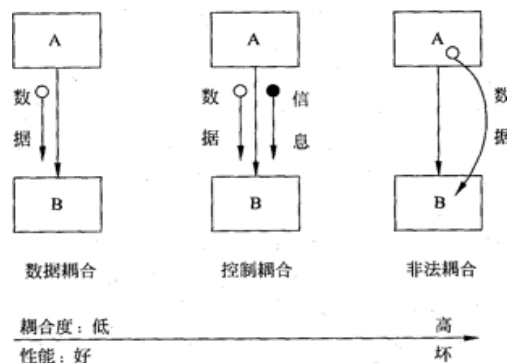


图 3.22 模块的耦合方式

• **逻辑内聚。**逻辑内聚是指模块各组成部分的处理动作在逻辑功能上是相似的。例如，把系统中与“输出”有关的操作抽取出来组成一个模块，包括将数据在屏幕上显示、从打印机上打印、拷贝到磁盘上等，则该模块就是逻辑内聚。逻辑内聚的内聚程度稍强于巧合内聚，但仍不利于修改和维护。

规定的总账科目，即一级科目；中间两位是部或行业规定的二级科目，最后两位是企业可以自定的三级科目。这种编码优点是易于校对，易于处理，缺点是不便于记忆。

- **字符码。**即以纯字符形式编码(英文、汉语拼音等)。这类编码常见的有程序设计中的字段名、变量名编码。例如在开发一个商业经贸性公司的信息系统时，在数据库中需要分别存储商品的进、存、销3个环节的价格、成本、资金占用等信息。为了区别起见，这时可以规定：字段的前两位分别用 J_、C_、X_ 表示进、存、销，后5位数来代表价格、成本、费用、资金占用等等。例如 J_price 表示进价。这就是一个典型的纯字符码。这种编码优点是可辅助记忆，缺点是校对不易，不易反映分类的结构。

- **混合码。**即以数字和字符混合形式编码。混合码是在各类管理中最常用的另一类编码形式。例如 GB. XXX 表示国家标准的某类编码，IEEE 802. X 表示某类网络协议标准名称的编码。

编码问题的关键在于分类。有了一个科学的分类，系统要建立编码就很容易了。准确的分类是工作的标准化、系列化、合理化的基础和保证。一个好的分类既要保证处理问题的需要，又要保证科学管理的需要。

在实际分类时必须遵循如下几点：

- 必须保证有足够的容量，以包括规定范围内的所有对象。如果容量不够，不便于今后变化和扩充，随着环境的变化这种分类很快就失去了生命力。
- 按属性系统化。分类不能是无原则的，必须遵循一定的规律。分类应按照处理对象的各种具体属性系统地进行。如在线分类方法中，哪一层次是按照什么属性来分类，哪一层次是标识一个什么类型的对象集合等都必须系统地进行，只有这样的分类才比较容易建立，比较容易为别人所接受。
- 分类要有一定的柔性，不至于在出现变更时破坏分类的结构。所谓柔性是指在一定情况下分类结构对于增设或变更处理对象的可容纳程度。柔性好的系统在一般的情况下增加分类不会破坏其结构。但是柔性往往还会带来别的一些问题，如冗余度大等，这都是设计分类时必须考虑的问题。
- 注意本分类系统与外系统、已有系统的协调。任何一项工作都是从原有的基础上发展起来的，故分类时一定要注意新老分类的协调性，以便于系统的联系、移植、协作以及新老系统的平稳过渡。

目前最常用的分类方法概括起来有两种，一种是线分类方法，一种是面分类方法。线分类方法的主要出发点是：首先给定母项，下分若干子项，由对象的母项分大集合，由大集合确定小集合……最后落实到具体对象。线分类划分时要掌握两个原则，惟一性和不交叉性。否则分类后如果出现有二义性，将会给后继工作带来诸多不便。线分类法容易识别和记忆，但结构不灵活，柔性较差。

面分类方法与线分类法不同，主要从面角度来考虑分类。面分类方法柔性好，面的增、删、修改都很容易，对机器处理有良好的适应性，缺点是不易直观识别，不便于记忆。

3.4.4.2 输出设计

从系统开发的角度看，输出决定输入，即输入信息只有根据输出要求才能确定。

输出设计包括以下几方面的内容。

- **确定输出内容。**输出内容的设计首先要确定用户在使用信息方面的要求。根据用户要求，设计输出信息的内容，包括信息形式(表格、图形、文字)、输出项目及数据结构、数据类型、位数及取值范围、数据的生成途径、完整性及一致性的考虑等。

- **选择输出设备与介质。**常用的输出设备有显示终端、打印机、磁带机、绘图仪、缩微胶卷输出器、多媒体设备。输出介质有纸张、磁带、磁盘、缩微胶卷、光盘、多媒体介质等。这些设备和介质各有特点，应根据用户对输出信息的要求，结合现有设备和资金条件选择。

- **确定输出格式。**输出格式要满足使用者的要求和习惯，做到格式清晰、美观、易于阅读和理解。

下面着重讨论最终输出方式的设计问题。

最终输出方式常用的只有两种：一种是报表输出，另一种是图形输出。采用哪种输出形式为宜，应根据系统分析和业务管理的要求而定。一般来说，对于基层和具体事务的管理者，应用报表方式给出详细的记录数据为宜；而对于高层领导或宏观、综合管理部门，则应该使用图形方式用以显示综合数据或发展趋势等信息。

3.4.4.3 输入设计

输入设计的目的是保证向系统输入正确的数据。在此前提下，做到输入方法简单、迅速、经济、方便。为此，输入设计应遵循以下原则：

- **最小量原则。**这就是保证满足处理要求的前提下使输入量最小。输入量越小，出错机会越小，花费时间越小，数据一致性越好。

- **简单性原则。**输入的准备、输入过程应尽量容易，以减少错误的发生。

- **早检验原则。**对输入数据的检验尽量接近原数据发生点，使错误能及时得到改正。
- **少转换原则。**输入数据尽量用其处理所需的形式记录，以免数据转换介质时发生错误。

输入设计的内容包括：

- **确定输入数据内容。**包括确定输入数据项名称、数据内容、精度、数值范围。

• **输入方式设计。**主要是根据总体设计和数据库设计的要求来确定数据输入的具体形式。常用的输入方式有键盘输入，模/数、数/模输入，网络数据传送，磁/光盘读入等。通常在设计新系统的输入方式时，应尽量利用已有的设备和资源，避免大批量的数据通过键盘输入。

• **输入格式设计。**实际设计数据输入时(特别是大批量的数据统计报表输入时)，常常遇到统计报表(或文件)结构与数据库文件结构不完全一致的情况。如有可能，应尽量改变统计报表或数据库关系表二者之一的结构，使其一致，以减少输入格式设计的难度。现在还可采用智能输入方式，由计算机自动将输入送至不同表格。

• **校对方式设计。**特别是针对数字、金额数等字段，没有适当的校对措施作保证是很危险的。所以对一些重要的报表，输入设计一定要考虑适当的校对措施，以减少出错的可能性。但应指出的是绝对保证不出错的校对方式是没有的。

常用的校对方式有人工校对，二次键入校对(同一批数据两次键入)和数据平衡校对。

数据平衡校对方法常用在对财务报表和统计报表等完全数字型报表的输入校对中。具体做法是，在原始报表每行每列中增加一个字段，算出每行每列的和。在设计新系统的输入时再另设一个累加值，先让计算机将输入数据的行列数累加起来，然后再将累加的结果与原始报表中计算好的和进行比较。如果一致，则可认为输入正确，反之则拒绝接受该数据记录。

目前，物流系统一般采用条码技术进行输入。条码技术是在计算机的应用实践中产生和发展起来的一种自动识别技术。它是为实现对商品信息的自动扫描而设计的，是快速、准确采集数据的有效手段。条码技术的应用解决了数据采集和数据录入的“瓶颈”问题，为供应链管理提供了有力的技术支持。

3.4.4.4 处理过程设计

总体结构设计将系统分解成许多模块，并决定了每个模块的外部特征：功能和界面。计算机处理过程的设计则要确定每个模块的内部特征，即内部的执行过程，包括局部的数据组织、控制流、每一步的具体加工要求及种种实施细节。通过这样的设计，为编写程序制定一个周密的计划。

处理过程设计的关键是用一种合适的表达方法来描述每个模块的执行过程。这种表示方法应该简明、精确，并由此能直接导出用编程语言表示的程序。常用的描述方式有图形、语言和表格等3类，如传统的框图、各种程序语言和判定表等。

1. 程序流程图

流程图(flow chart)即程序框图，是历史最久、流行最广的一种图形表示方法。程序流程图包括三种基本成分：加工步骤，用方框表示；逻辑条件，用菱形表示；控制流，用箭头表示。

图形表示的优点是直观、形象、容易理解。但从结构化程序设计的角度看，流程图不是理想的表达工具。缺点之一是表示控制的箭头过于灵活。使用得当，流程图简单易懂；使用不当，流程图可能非常难懂，而且无法维护。流程图的另一个缺点是只描述执行过程而不能描述有关数据。

2. 盒图(NS图)

盒图是结构化程序设计出现之后，为支持这种设计方法而产生的一种描述工具。在NS图中，每个处理步骤用一个盒子表示。盒子可以嵌套。盒子只能从上头进入，从下头走出，除此之外别无其他出入口，所以盒图限制了随意的控制转移，保证了程序的良好结构。

3. 形式语言

形式语言是用来描述模块具体算法的非正式的比较灵活的语言。其外层语法是确定的，而内层语法不确定。外层语法描述控制结构用类似一般编程语言的保留字，所以是确定的。内层语法故意不确定，可以按系统的具体情况和不同层次灵活选用，实际上可用自然语言来描述具体操作。

可以看出形式语言同结构性语言的想法是一致的。形式语言的优点是接近自然语言(英语)，所以易于理解；其次，它可以作为注释嵌套在程序中成为内部文档，提高程序的自我描述性；第三，因为是语言形式，易于被计算机处理，可用行编辑程序或字处理系统对形式语言进行编辑修改。

4. 决策树

如果一个加工决策或判断的步骤较多，则使用形式语言时，语句的嵌套层次太多，不便于基本加工的逻辑功能的清晰描述。决策树是一种图形工具，适合于描述加工中具有多个策略、每个策略和若干条件有关的逻辑功能。

5.决策表

在基本加工中,如果判断的条件较多、各条件又相互组合、相应的决策方案较多的情形下用决策树来描述,树的结构比较复杂,图中各项注释比较繁琐。决策表也是一种图形工具,呈表格形。决策表将比较复杂的决策问题简洁、明确、一目了然地描述出来。

3.4.4.5 数据存储设计

信息系统的主要任务是通过大量的数据获得管理所需要的信息,这就必须存储和管理大量的数据。因此建立一个良好的数据组织结构和数据库,使整个系统都可以迅速、方便、准确地调用和管理所需的数据,是衡量信息系统开发工作好坏的主要指标之一。

数据结构组织和数据库或文件设计,就是要根据数据的不同用途、使用要求、统计渠道、安全保密性等,来决定数据的整体组织形式、表或文件的形式,以及决定数据的结构、类别、载体、组织方式、保密等级等一系列的问题。

一个好的数据结构和数据库应该充分反映物流发展变化的状况,充分满足组织的各级管理要求。同时还应该使得后继系统开发工作方便、快捷,系统开销(如占用空间、网络传输频度、磁盘或光盘读写次数等)小,易于管理和维护等特点。有关数据库及数据库设计的相关内容可参见本书第9章。

在建立了数据的整体关系结构之后,剩下的就是要确定数据资源分布和安全保密属性了。其中数据资源的分布是针对分布数据库系统而言的,而安全保密属性的定义则是针对某些特殊信息,如财务数据等而言的。

- **数据资源分布:**如果所规划和设计的系统是在网络环境之下,那么数据库设计必须考虑整个数据资源在网络各结点(包括网络服务器)上的分配问题。

- **数据的安全保密:**一般数据库软件都提供定义数据安全保密性的基本功能。系统所提供的安全保密功能一般有8个等级(0-7级),4种不同方式(只读、只写、删除、修改),而且允许用户利用这8个等级的4种方式对每一个表自由地进行定义。有关安全保密的相关内容可参见本书第9章和第12章。

3.4.4.6 用户界面设计

用户界面是系统与用户之间的接口,也是控制和选择信息输入输出的主要途径。用户界面设计应坚持友好、简便、实用、易于操作的原则。例如,在设计菜单时应尽量避免菜单嵌套层次过多和每选择一次还需确认一次的设计方式。菜单又如在设计大批数据输入屏幕界面时,应避免颜色过于鲜艳和多变。

界面设计包括菜单方式、会话方式、操作提示方式,以及操作权限管理方式等。

1.菜单方式

菜单是信息系统功能选择操作的最常用方式。按目前软件所提出的菜单设计工具,菜单的形式可以是下拉式、弹出式的,也可以是按钮选择方式的。

2.会话管理方式

在所有的用户界面中,几乎毫无例外地会遇到人机会话问题。最为常见的有:当用户操作错误时,系统向用户发出提示和警告性的信息;当系统执行用户操作命令遇到多种可能时,系统会要求用户进一步说明;系统定量分析的结果通过屏幕向用户发出控制型的信息等。这类会话的处理方式是让系统开发人员根据实际系统操作过程将会话语句写在程序中。

一般会话系统是面向企业领导的,会话系统设计必须满足会话的基本要求,如画面清晰、直观形象,明了、简洁,具有容错和纠错能力,提供信息汉字化、图形化、表格化等。

因此,会话设计重点解决会话方式、容错能力和系统的模块结构。

在语音会话方式还没有广泛使用的今天,会话的基本工具是键盘、屏幕和打印机,常用的方式是回答式、菜单式、表格式和图形式。

纠错、容错的目的是保证会话的正确性,提高会话的效率,在系统中可采用如下方法。

- **提示法:**分简单提示和重复提示法。
- **确认回答法:**为用户误操作提供改错机会。
- **无效处理法:**系统拒绝接收错误操作。
- **返回处理法:**拒绝不熟悉系统的用户使用操作。
- **延时处理法:**让用户有足够的时间理解系统的提问内容,防止错误回答。
- **帮助处理法:**给用户提供帮助信息,并给予重新操作的机会。

3.提示方式与权限管理

为了操作使用方便,在系统设计时,常常把操作提示和要点同时显示在屏幕的旁边,以使用户操作方便,这是当前比较流行的用户界面设计方式。另一种操作提示设计方式则是将整个系统操作说明书全送入系统文件中,并设置系统运行状态指针。当系统运行操作时,指针随着系统运行状态来改变,当用户按

“求助”键时，系统则立刻根据当前指针调出相应的操作说明。

与操作方式有关的另一个内容就是对数据操作权限的管理。权限管理一般都是通过人网口令和建网时定义该结点的级别，这两点相结合来实现的。

3. 4. 4. 7 安全控制设计

从数据环境和数据处理两方面看，影响系统安全的因素有：

• **环境性因素**。是指管理机构的组织、硬件和系统软件、系统开发、自然环境等方面的因素。例如组织方面职责不分，没有监督机构等；硬件软件方面，硬件失灵，系统软件失灵，逻辑线路错误等；系统开发方面，没有按科学的方法开发系统和设计程序、系统未经测试和调试等；自然环境方面，火灾、水灾、风灾、地震等；安全管理方面，数据处理资源的接触是随意的，无必要的限制等。

• **数据处理因素**。是指数据处理行为引起的各种情况。例如输入环节录入错误信息，使用无效代码，击错功能键，丢失数据，重复输入，没有将数据存盘等；处理环节使用了错误程序，使用了错误的文件，处理不及时，丢失数据文件和程序等；输出环节错误地发送报表或不及时发送，报告中的错误未加更正等。

要进行系统的安全控制，应针对影响系统安全的两方面因素入手，有的放矢，相应地进行环境和数据处理两方面的有效控制，以保证系统安全有效地运行。

3. 4. 5 系统设计报告

系统设计阶段的最终结果是系统设计报告。系统设计报告是下一步系统实施的基础。

从系统调查、系统分析到系统设计是信息系统开发的主要工作，这3个阶段的工作量几乎占到了总开发工作量的70%，而且这3个阶段所用的工作图表较多，涉及面广，较为复杂。

下面附上系统设计说明书内容目录，供大家参考。

系统设计说明书

一、引言

1. 摘要：摘要说明所设计的系统的名称、目的和功能。

2. 背景

(1) 项目的承担者

(2) 用户

(3) 本项目和其他系统或机构的关系和联系

3. 工作条件/限制：说明本项目开发中所具备的工作条件和受到的限制。

(1) 硬件/软件/运行环境方面的限制

(2) 保密和安全的限制

(3) 有关部门的业务人员提供确切的数据及其定义

(4) 有关系统软件文本

(5) 网络协议标准文本

(6) 国家安全保密条例

4. 参考和引用资料

(1) 本项目的已核准的计划任务书或合同和上级机关的批文

(2) 属于本项目的其他已发表的文件

(3) 本文件中引用的文件资料：列出文件资料的标题、编号、发表日期和制定单位，说明这些文件资料的来源

5. 专门术语定义：例如本文件所用到的术语。

二、系统总体技术方案

1. 模块设计：模块设计阶段中，在系统内部划分成各个基础部分—模块结构，确定系统的总体结构。总体结构与各个分层模块结构的关系是程序实施的重要依据。模块结构采用模块结构图来表示。模块结构图是采用HIPO图(即分层输入—处理—输出图)形式绘制而成的框图。

(1) 名称：列出系统中各主要功能的结构图名称和它们之间的关系

(2) 功能：用文字简单说明主要模块结构图应具有的功能

(3) 功能说明：说明是用伪码形式或是用结构英语形式，或者其他自然语言形式描述模块结构图

(4) 评价

(5) 验收：指设计人员验收的决定和处理情况

2. 代码设计：代码设计是信息系统所必需的前提条件，是不可缺少的重要内容。它是进行信息分类、

校对、总计和检查的关键，它也用于指定数据的处理方法，区别数据类型，并制定计算机处理的内容。

(1)代码的方式和种类：简单说明代码的方式和种类

(2)功能：从编码的原则要求(如单义性、可读性等)去简单说明代码所体现的功能

(3)评价：从识别信息、信息标准化、节省存储单元、提高运算速度、节省计算机的处理费用以及代码的特性进行评价

(4)验收

3. 输入设计：输入设计担负着将系统外的数据以一定的格式送入计算机的任务。它直接影响到人工系统和机器系统的工作质量。输入设计的基点是确保向信息系统提供正确的信息。输入必须有必要的介质和设备。

(1)输入项目：说明对本系统的主要的输入项目

(2)输入的承担者：说明对数据输入工作的承担者的安排，并指出操作人员、维护人员的教育水平和技术专长。如果输入数据同某一接口软件有关，应说明该接口软件的来源

(3)主要功能要求：从满足正确、迅速、简单、经济、方便使用者方面的要求去说明

(4)输入要求：简单说明各主要输入数据类型和来源及所用的设备、介质、格式、数值范围、精度等

(5)输入校验：简述所用的数据校验法和效果

(6)评价

(7)验收

4. 输出设计：输出的含义是把由计算机对输入的原始数据进行处理加工的结果按一定的格式提供给用户。输出不仅有一定的格式要求，而且还必须有必要的介质和设备。

(1)输出项目：说明对本系统的主要输出项目

(2)输出接受者：说明输出的主要项目的数据的接受者

(3)主要功能

(4)输出要求：说明输出数据类型及所用的设备介质、格式、数值范围、精度等

(5)评价

(6)验收

5. 数据库设计说明：数据库设计是指数据库应用系统的设计。编制数据库设计说明书的目的是对设计中的数据结构的标识、逻辑结构和物理结构做出具体的设计规定。编写提纲和内容要求如下：

(1)概述

a. 目的：说明开发的意图、应用目的、作用范围以及有关数据库开发的背景资料

b. 主要功能：简要说明数据库系统的主要功能

c. 用户的安排：指最终用户，说明操作人员、数据管理人员、维护人员的水平

(2)需求、性能规定

a. 精度：简述对数据精度的要求

b. 有效性：说明对数据库存取数据的有效性的要求

c. 时间要求：如响应时间、数据的转换和传送时间等

d. 其他专门要求

(3)运行环境要求

a. 设备：简述运行数据库系统的硬设备及其专门功能

b. 支撑软件：列出支撑软件并说明测试用的软件

c. 安全保密：说明在安全保密方面的全部要求

d. 其他要求

(4)设计考虑

a. 逻辑结构设计：简要说明本系统(或子系统)内所使用的数据结构中，有关数据项、记录、文件的标识、定义、长度及它们之间的相互关系

b. 物理结构设计：简要说明本系统内所使用的数据结构中有关数据库的存储要求、访问方法、存取单位、存取的物理关系(索引、设备、存储区域)、设计考虑和保密处理。

(5)评价：简要说明对时间、空间效率、维护代价和各种用户要求进行权衡所产生的方案性能情况

(6)验收

6. 模型库及方法库设计。

7. 网络设计：系统的网络结构，功能的设计。

8. 安全保密设计。

9. 实施方案说明书：系统设计完成以后就要确定系统实施方案，书写实施方案说明书。信息系统的研制工作就从系统设计阶段转入实施阶段。实施方案说明书就作为系统实施阶段的依据和出发点。

(1) 实施方案说明

- a. 项目的说明：指对系统名称、子系统名称、程序名称、程序语言、使用的设备等逐项说明
- b. 数据项目的说明：指对数据长度、文件名称和形式编号、构成记录的各项名称和内容等逐项说明
- c. 处理内容的说明：指对进行程序设计的处理内容进行详细说明

(2) 实施的总计划

a. 工作任务的分解：对于项目开发中须完成的各项工作，包括文件编制、审批、打印、用户培训工作、使用设备的安排工作等，按层次进行分解，指明每项任务的要求

b. 进度：给出每项工作任务(包括文件编制)的预定开始日期和完成日期，规定各项工作任务完成的先后顺序以及每项工作任务完成的标志

c. 预算：逐项列出本开发项目所需的劳务费(包括办公费、差旅费、机时费、资料费)以及通信设备和专用设备的租金

(3) 实施方案的审批

- a. 参与审议人员：除用户、系统研制人员、程序员、操作员等以外，还包括邀请的专家、管理人员等
- b. 审批的实施方案：说明经审批的实施方案的概况和审批人员名单

3. 5 系统实施

3. 5. 1 系统实施概述

1. 系统实施的目的和任务

系统实施是新系统开发工作，的最后一个阶段。所谓实施指的是将系统设计阶段的结果在计算机上实现，将原来纸面上的、类似于设计图式的新系统方案转换成可执行的应用软件系统。

系统实施阶段的主要任务是：

- **按总体设计方案购置和安装计算机网络系统。**硬件准备包括计算机主机、输入输出设备、存储设备、辅助设备(稳压电源、空调设备等)、通信设备等。购置、安装和调试这些设备要花费大量的人力、物力，并且持续相当长的时间。

- **软件准备。**软件准备包括系统软件、数据库管理系统以及一些应用程序。这些软件有些需要购买，有些需要组织人力编写。编写程序是系统实施阶段的重要任务之一。

- **人力培训。**主要指用户的培训，包括主管人员和业务人员。这些人多数来自现行系统，精通业务，但往往缺乏计算机知识。为保证系统调试和运行顺利进行，应根据他们的基础，提前进行培训，使他们适应、逐步熟悉新的操作方法。

- **数据准备。**数据的收集、整理、录入是一项既繁重、劳动量又大的工作。而没有一定基础数据的准备，系统调试就不可能很好地进行。一般说来，确定数据库模型之后，就应进行数据的整理、录入。这样既分散了工作量，又可以为系统调试提供真实的数据。

- **投入切换和试运行。**

在系统实施过程中，还有若干非技术因素的影响。信息系统的最终受益人是企业的最高领导层，信息系统建设涉及到企业机构、权限的重组，只有具备进行变革权利的人才能真正地推进企业信息化。

企业在推行管理信息化时，总经理首先要了解企业一些公众的心理，如企业的各级员工的习惯心理，对信息系统使用持不信任态度的怀疑性排斥心理；此外，信息系统的使用将传统的金字塔管理变为扁平管理，使以前无法暴露的灰色行为，将被一览无遗；素质较低或年龄较大的员工对操作电脑系统具有畏惧心理。如果没有妥善的培训或疏导，这些都将作为系统应用的极大障碍。

2. 系统实施的步骤

系统开发工作沿着信息系统的生命周期逐渐推进，经过详细设计阶段后，便进入系统实施阶段，下面对工作步骤进行介绍。

(1) **按总体设计方案购置和安装计算机网络系统。**购置和安装硬件是比较简单的事情，只需按总体设计的要求和可行性报告中财力资源的分析，选择好价格性能比高的设备，通知供货厂家按要求供货并安装即可。

(2) **建立数据库系统。**如果前面数据与数据流程分析以及数据库设计工作进行得比较规范，而且开发者又对数据库技术比较熟悉的话，按照数据库设计的要求只需 1-2 个人一天即可建立起一个大型数据库结构。

(3) **程序设计。**

(4) 收集有关数据并进行录入工作，然后进行系统测试。

(5) 人员培训、系统转换和试运行。

3.5.2 程序设计

程序设计的主要依据是系统设计阶段的 HIPO 图以及数据库结构和编码设计。

1. 程序设计的方法

目前程序设计的方法大多按照结构化方法、原型方法、面向对象的方法进行。

编程的目的是为了实现开发者在系统分析和系统设计中提出管理方法和处理构想。所以在编程和实现中，建议应尽量借用已有的程序和开发工具，尽快尽好地实现系统，而不要在具体的编程和调试工作中花费过多的精力和时间。

• **结构化程序设计方法。**若遇到某些开发过程不规范，模块划分不细，或者是因特殊业务处理的需要模块程序量较大时，结构化程序设计方法是一种非常有效的方法。结构化的程序设计方法主要强调 3 点：模块内部程序各部分要自顶向下的结构化划分；各程序部分应按功能组合；各程序部分的联系尽量使用调用子程序 (CALL-RETURN) 方式，不用或少用 GOTO 方式。

• **快速原型式的程序开发方法。**具体实施方法是，首先将 HIPO 图中类似带有普遍性的功能模块集中，如菜单模块、报表模块、查询模块，统计分析和图形模块等，这些模块几乎是每个子系统都必不可少的；然后再去寻找有无相应、可用的软件工具，如果没有则可以考虑开发一个能够适合各子系统情况的通用模块；最后用这些工具生成这些程序模块原型。如果 HIPO 图中有一些特定的处理、功能和模块，而这些功能和模块又是现有工具不可能生成出来的，则再考虑编制一段程序加进去。利用现有的工具和原型方法可以很快地开发出所要的程序。

• **面向对象程序设计方法。**面向对象程序设计方法一般应与 OOD 所设计的内容相对应。它是一个简单直接的映射过程。即将 OOD 中所定义的范式直接用面向对象程序 (OOP) 如 C++，Smalltalk, Visual C 等来取代即可。例如，用 C++ 中的对象类型来取代 OOD 范式中的类-&-对象，用 C++ 中的函数和计算功能来取代 OOD 范式中的处理功能等。在系统实现阶段，OOP 的优势是巨大的，是其他方法所无法比拟的。

2. 程序设计基本模块

一个信息系统的应用软件由很多程序模块组成，这些程序模块可以归纳成为几种基本类型，其结构如图 3.24 所示。

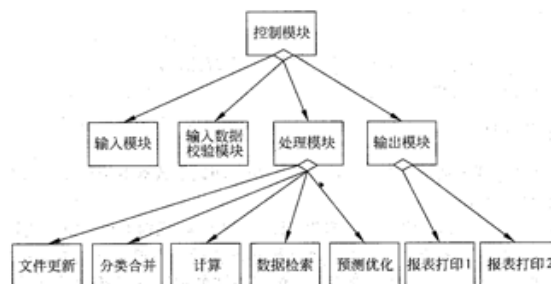


图 3.24 基本程序模块结构

①**控制模块。**控制模块包括主控制模块和各级控制模块。控制模块的主要功能是根据用户要求信息，由用户确定处理顺序，然后控制转向各处理模块的入口。

②**输入模块。**主要用来输入数据，输入方式有键盘输入和软盘输入两种。

③**输入数据校验模块。**该模块对已经输入计算机中的数据进行校验，以保证原始数据的正确性。校验的方法通常有重复输入校验和程序校验两种。

④**输出模块。**输出模块用来将计算机的运行结果通过屏幕、打印机或磁盘、磁带等设备输出给用户。在信息系统中，一般都有大量的表格、图表需要输出，因此输出模块的质量直接关系到整个系统的性能。

⑤**处理模块。**根据信息系统的不同应用要求，有不同的处理功能，通常有以下几种类型：

• **文件更新模块。**当系统应用的数据发生变化时，需要修改数据文件。例如，增加新的记录、修改数据项或记录、删除某些不需要的记录等。一般来说，文件更新模块应该具有下述功能：对记录中关键字的控制功能，通过关键字查找相应记录；控制总记录数的功能，以便控制追加、插入记录的位置；具有记录地址或字节位置的控制功能，以使确定修改数据的位置，控制插入或者追加的数据位置。

• **分类合并模块。**分类合并模块的主要功能是对已经建立的文件，按某关键字进行分类合并。例如，在材料核算系统中耗用材料要按照材料类型合并处理。分类合并程序应该具有下述功能：控制记录总数的功能；字符串比较的功能；排序、统计和计数功能。

• **计算模块。**该模块的主要功能是进行计算处理，包括同类记录中各数据项的运算。

• **数据检索模块。**该模块的主要功能是为用户提供查询的有关信息，它包括输入查询要求和输出特定的查询结果。它是管理信息系统的人机接口，对于人机交互的友好程序以及查询响应时间等均有较高要求。

• **预测或优化模块。**该模块的主要功能是使用预测或优化的数学模型，利用信息系统所提供的有关数据，进行计算和分析并输出结果，用来辅助企业或部门的管理人员进行决策。例如库存管理中的 ABC 分类、最佳订货量计算和财务管理中的资金分析等。

3.5.3 系统测试与调试

3.5.3.1 系统测试的意义及目的

系统测试是为了发现错误而执行程序的过程，成功的测试是发现了至今尚未发现的错误的测试。

测试的目的就是希望能以最少的人力和时间发现潜在的各种错误和缺陷。应根据开发各阶段的需求、设计等文档或程序的内部结构精心设计测试实例，并利用这些实例来运行程序，以便发现错误的过程。信息系统测试应包括软件测试、硬件测试和网络测试。硬件测试、网络测试可以根据具体的性能指标来进行，此处所说的测试更多是指软件测试。

系统测试是保证系统质量和可靠性的关键步骤，是对系统开发过程中的系统分析、系统设计和实施的最后复查。

根据测试的概念和目的，在进行信息系统测试时应遵循以下基本原则：

- **应尽早并不断地进行测试。**测试不是在应用系统开发完之后才进行的。由于原始问题的复杂性、开发各阶段的多样性以及参加人员之间的协调等因素，使得在开发各个阶段都有可能出现错误。因此，测试应贯穿在开发的各个阶段，尽早纠正错误，消除隐患。

- **测试工作应该避免由原开发软件的人或小组承担。**一方面，开发人员往往不愿否认自己的工作，总认为自己开发的软件没有错误；另一方面，开发人员的错误很难由本人测试出来，很容易根据自己编程的思路来定制测试思路，具有局限性。测试工作应由专门人员进行，会更客观，更有效。

- **设计测试方案的时候，不仅要确定输入数据，而且要根据系统功能确定预期输出结果。**将实际输出结果与预期结果相比较就能发现测试对象是否正确。

- **在设计测试实例时，不仅要设计有效合理的输入条件，也要包含不合理、失效的输入条件。**测试的时候，人们往往习惯按照合理的、正常的情况进行测试，而忽略了对异常、不合理、意想不到的情况进行测试，而这些可能就是隐患。

- **在测试程序时，不仅要检验程序是否做了该做的事，还要检测程序是否做了不该做的事。**多余的工作会带来副作用，影响程序的效率，有时会带来潜在的危害或错误。

- **严格按照测试计划来进行，避免测试的随意性。**测试计划应包括测试内容、进度安排、人员安排、测试环境、测试工具、测试资料等。严格地按照测试计划可以保证进度，使各方面都得以协调进行。

- **妥善保存测试计划、测试例子，作为软件文档的组成部分，为维护提供方便。**

测试例子都是精心设计出来的，可以为重新测试或追加测试提供方便。当纠正错误、系统功能扩充后，都需要重新开始测试，而这些工作重复性很高，可以利用以前的测试例子，或在其基础上修改，然后进行测试。

3.5.3.2 测试过程

测试是开发过程中一个独立且非常重要的阶段，测试过程基本上与开发过程平行进行。

一个规范化的测试过程通常包括以下基本的测试活动。

(1)**拟定测试计划。**在制定测试计划时，要充分考虑整个项目的开发时间和开发进度以及一些人为因素和客观条件等，使得测试计划是可行的。测试计划的内容主要有：测试的内容、进度安排、测试所需的环境和条件、测试培训安排等。

(2)**编制测试大纲。**测试大纲是测试的依据。它明确详尽地规定了在测试中针对系统的每一项功能或特性所必须完成的基本测试项目和测试完成的标准。

(3)**根据测试大纲设计和生成测试例子。**在设计测试例子的时候，可综合利用前面介绍的测试例子和设计技术，产生测试设计说明文档，其内容主要有被测项目、输入数据、测试过程、预期输出结果等。

(4)**实施测试。**测试的实施阶段是由一系列的测试周期组成的。在每个测试周期中，测试人员和开发人员将依据预先编制好的测试大纲和准备好的测试例子，对被测软件或设备进行完整的测试。

(5)**生成测试报告。**测试完成后，要形成相应的测试报告，主要对测试进行概要说明，列出测试的结论，指出缺陷和错误，另外，给出一些建议，如可采用的修改方法，各项修改预计的工作量及修改的负责人员。

3.5.3.3 测试策略与测试方法

软件测试方法分人工测试和机器测试。

1.人工测试

人工测试指的是采用人工方式进行测试，目的是通过对程序静态结构的检查，找出编译时不能发现的错误。人工测试一旦发现错误，就能确定问题的位置及是什么错误，而且能一次发现多处错误。经验表明，组织良好的人工测试可以发现程序中 30%~70% 的编码和逻辑设计错误。

人工测试又称为代码审查。其内容包括检查代码和设计是否一致，检查代码逻辑表达是否正确和完整，

检查代码结构是否合理等。主要有 3 种方法：

- **个人复查：**指程序员本人对程序进行检查。由于心理上的原因和思维惯性的影响，对自己的错误一般不容易发现，对功能理解的错误更不可能纠正。因此，这种方法主要针对小规模程序，效率不高。

- **抽查：**通常由 3-5 人组成测试小组，测试人员应是没有参加该项目开发的有经验的程序设计人员。在抽查之前，应先阅读相关的软件资料和源程序，然后测试人员扮演计算机的角色，将一批有代表性的测试数据沿程序的逻辑走一遍，监视程序的执行情况。人工检测程序很慢，只能选择少量简单的例子。

- **会审：**测试人员的构成与抽查类似。在会审之前，测试人员应该充分阅读相关资料，比如系统分析说明书、系统设计说明书、源程序等。有经验的测试人员列出尽可能多的典型错误。在会审时，由编程人员逐句讲解程序，测试人员逐个审查、提问。通过这种方式，往往可能使编程人员发现自己以前没有意识到的错误，使问题暴露。会审后，要将发现的错误登记、分析、归类。

代码复审应该在被测软件编译成功之后，编译都不通过的软件，当然谈不上复审；在复审期间，应保证有足够的时间，让测试小组对问题进行充分的讨论，这样才能有效提高测试效率，避免出错。

2. 机器测试

机器测试是把设计好的测试例子作用于被测程序，比较测试结果和预期结果是否一致，如果不一致，就说明可能存在错误。机器测试只能发现错误的症状，但无法对问题进行定位。

机器测试分为黑盒测试和白盒测试两种。

①**黑盒测试也称为功能测试。**将软件看成黑盒子，在完全不考虑软件的内部结构和特性的情况下，测试软件的外部特性。进行黑盒测试主要是为了发现以下几类错误：

- 是否有错误的功能或遗漏的功能？
- 界面是否有误？输入是否能够正确接收？输出是否正确？
- 是否有数据结构或外部数据库访问错误？
- 性能是否能够接受？
- 是否有初始化或终止性错误？

②**白盒测试也称为结构测试。**将软件看成透明的白盒。根据程序的内部结构和逻辑来设计测试例子，对程序的路径和过程进行测试，检查是否满足设计的需要。其原则是：

- 程序模块中的所有独立路径至少执行一次。
- 在所有的逻辑判断中，取“真”和取“假”的两种情况至少都能执行一次。
- 每个循环都应在边界条件和一般条件下各执行一次。
- 测试程序内部数据结构的有效性等。

3. 软件测试步骤

软件测试实际上分成四步进行。

(1) 单元测试。

单元测试也称为模块测试，在模块编写完成且无编译错误后就可以进行。如果选用机器测试，一般用白盒测试法，多个模块可以同时进行。

单元测试主要从模块的以下五个特征进行检查。

①**模块接口。**模块的接口保证了测试模块的数据流可以正确地流入、流出。在测试中应检查以下要点：

- 测试模块的输入参数和形式参数在个数、属性、单位上是否一致。
- 调用其他模块时所给出的实际参数和被调用模块的形式参数在个数、属性、单位上是否一致。
- 调用标准函数时所用的参数在属性、数目和顺序上是否正确。
- 全局变量在各模块中的定义和用法是否一致。
- 输入是否仅改变了形式参数。
- 开/关的语句是否正确。
- 规定的 I/O 格式是否与输入输出语句一致。
- 在使用文件之前是否已经打开文件或是用文件之后是否已经关闭文件。—

②**局部数据结构。**在单元测试中，局部数据结构出错是比较常见的错误，在测试时应重点考虑以下因素：

- 变量的说明是否合适。
- 是否使用了尚未赋值或尚未初始化的变量。
- 变量的初始值或默认值是否正确。

- 变量名是否有错(例如:拼写错)。
- 是否出现上溢、下溢或地址异常的错误。

③**重要的执行路径**。在单元测试中,对路径的测试是最基本的任务。由于不能进行穷举测试,需要精心设计测试例子来发现是否有计算、比较或控制流等方面的错误。

• 计算方面的错误:算术运算的优先次序不正确或理解错误;精度不够;运算对象的类型彼此不相容;算法错;表达式的符号表示不正确等。

• 比较和控制流的错误:本应相等的量由于精度造成不相等;不同类型进行比较;逻辑运算符不正确或优先次序错误;循环终止不正确(如多循环一次或少循环一次)、死循环;不恰当地修改循环变量;当遇到分支循环时,出口错误等。

④**出错处理**。好的设计应该能预测到出错的条件并且有对出错处理的路径。虽然计算机可以显示出错信息的内容,但仍需要程序员对出错进行处理,保证其逻辑的正确性,便于用户维护。—

⑤**边界条件**。边界条件的测试是单元测试的最后工作,也是非常重要的工作。软件容易在边界出现错误。

由于模块不是独立运行的程序,各模块之间存在调用与被调用的关系。

在对每个模块进行测试时,需要开发两种模块:

- **驱动模块**。相当于一个主程序,接收测试例子的数据,将这些数据送到测试模块,输出测试结果。
- **桩模块**,也称为存根模块。桩模块用来代替测试模块中所调用的子模块,其内可进行少量的数据处理,目的是为了检验入口,输出调用和返回的信息。

提高模块的内聚度可以简化单元测试。如果每个模块只完成一种功能,对于具体模块来讲,所需的测试方案数据就会显著减少,而且更容易发现和预测模块中的错误。

(2)组装测试。

组装测试也称为集成测试,就是把模块按系统设计说明书的要求组合起来进行测试。即使所有模块都通过了测试,但在组装之后,仍可能会出现:穿过模块的数据被丢失;一个模块的功能对其他模块造成有害的影响;各个模块组合起来后没有达到预期功能;全局数据结构出现问题;另外单个模块的误差可以接受,但模块组合后,可能会出现误差累积,最后到不能接受的程度,所以需要组装测试。

通常组装测试有两种方法:一种是分别测试各个模块,再把这些模块组合起来进行整体测试,即非增量式集成。另一种是把下一个要测试的模块组合到已测试好的模块中,测试完后再将下一个需要测试的模块组合起来,进行测试,逐步把所有模块组合在一起,并完成测试,即增量式集成。非增量式集成可以对模块进行并行测试,能充分利用人力,并加快工程进度。但这种方法容易混乱,出现错误不容易查找和定位。增量式测试的范围一步步扩大,错误容易定位,而且已测试的模块可在新的条件下再测试,测试更彻底。

(3)确认测试。

经过组装测试以后,软件就被集成起来,接口方面的问题已经解决,将进入软件测试的最后一个环节—确认测试。确认测试的任务就是进一步检查软件的功能和性能是否与用户要求的一样。系统方案说明书描述了用户对软件的要求,所以是软件有效性验证的标准,也是确认测试的基础。

确认测试,首先要进行有效性测试以及软件配置审查,然后进行验收测试和安装测试,经过管理部门的认可和专家的鉴定后,软件即可以交给用户使用。

• **有效性测试**,就是在模拟环境下,通过黑盒测试检验所开发的软件是否与需求规格说明书一致。在设计测试例子时,除了检测软件的功能和性能之外,还需要对软件的容错性、维护性等其他方面进行检测。测试人员可由开发者的内部人员组成,但最好是没有参加该项目的有经验的软件设计人员。在所有测试例子完成之后,若发现测试结果与预期的不符,这时要列出缺陷清单。在这个阶段才发现的严重错误,一般很难在预定的时间内纠正,需要与用户协商,寻找妥善解决问题的办法。

• **软件配置审查**,主要是检查软件(源程序、目标程序)和文档(包括面向开发和用户的文档)是否齐全以及分类是否有序。确保文档、资料的正确和完善,以便维护阶段使用。

• **验收测试**,是以用户为主的测试。软件开发人员和质量保证人员也应该参加。在验收测试之前,需要对用户进行培训,以便熟悉该系统。验收测试的测试例子由用户参与设计,主要验证软件的功能、性能、可移植性、兼容性、容错性等,测试时一般采用实际数据。

(4)系统测试。

系统测试是将已经确认的软件、计算机硬件、外设和网络等其他因素结合在一起,进行信息系统的各种组装测试和确认测试,其目的是通过与系统的需求相比较,发现所开发的系统与用户需求不符或矛盾的

地方。系统测试是根据系统方案说明书来设计测试例子的，常见的系统测试主要有以下内容：

- **恢复测试：**恢复测试检测系统的容错能力。检测方法是采用各种方法让系统出现故障，检验系统是否按照要求能从故障中恢复过来，并在预定的时间内开始事务处理，而且不对系统造成任何伤害。如果系统的恢复是自动的(由系统自动完成)，需要验证重新初始化、检查点、数据恢复等是否正确。如果恢复需要人工干预，就要对恢复的平均时间进行评估并判断它是否在允许的范围内。

- **安全性测试：**系统的安全性测试是检测系统的安全机制、保密措施是否完善，主要是为了检验系统的防范能力。测试的方法是测试人员模拟非法入侵者，采用各种方法冲破防线。系统安全性设计准则是使非法入侵者所花费的代价比进入系统后所得到的好处要大，此时非法入侵已无利可图。

- **强度测试：**是对系统在异常情况下的承受能力的测试，是检查系统在极限状态下运行时，性能下降的幅度是否在允许的范围内。因此，强度测试要求系统在非正常数量、频率或容量的情况下运行。强度测试主要是为了发现在有效的输入数据中可能引起不稳定或不正确的数据组合。例如，运行使系统处理超过设计能力的最大允许值的测试例子；使系统传输超过设计最大能力的数据，包括内存的写入和读出等。

- **性能测试：**检查系统是否满足系统方案说明书对性能的要求。性能测试覆盖了软件测试的各阶段，而不是等到系统的各部分所有都组装之后，才确定系统的真正性能。通常与强度测试结合起来进行，并同时软件、硬件进行测试。软件方面主要从响应时间、处理速度、吞吐量、处理精度等方面来检测。

- **可靠性测试：**通常使用以下两个指标来衡量系统的可靠性：平均失效间隔时间 MTBF(mean time between failures)是否超过了规定的时限，因故障而停机时间 MTTR(mean time to repairs)在一年中不应超过多少时间。

- **安装测试：**在安装软件系统时，会有多种选择。安装测试就是为了检测在安装过程中是否有误、是否容易操作等。主要检测系统的每一个部分是否齐全，硬件的配置是否合理，安装中需要产生的文件和数据库是否已产生，其内容是否正确等。

3.5.3. 4 调试

调试的任务就是根据测试时所发现的错误，找出原因和具体的位置，进行改正。调试工作主要由程序开发人员来进行，谁开发的程序就由谁来进行调试。

目前常用的调试方法有如下几种：

- **试探法。**调试人员分析错误的症状，猜测问题的所在位置，利用在程序中设置输出语句，分析寄存器、存储器的内容等手段来获得错误的线索，一步步地试探和分析出错误所在。这种方法效率很低，适合于结构比较简单的程序。

- **回溯法。**调试人员从发现错误症状的位置开始，人工沿着程序的控制流程往回跟踪程序代码，直到找出错误根源为止。这种方法适合于小型程序，对于大规模程序，由于其需要回溯的路径太多而变得不可操作。

- **对分查找法。**这种方法主要用来缩小错误的范围。如果已经知道程序中的变量在若干位置的正确取值，可以在这些位置上给这些变量以正确值，运行程序观察输出结果，如果没有发现问题，则说明从赋予变量一个正确值开始到输出结果之间的程序没有出错，问题可能在除此之外的程序中，否则错误就在所考察的这部分程序中。对含有错误的程序段再使用这种方法，直到把故障范围缩小到比较容易诊断为止。

- **归纳法。**归纳法就是从测试所暴露的问题出发，收集所有正确或不正确的数据，分析它们之间的关系，提出假想的错误原因，用这些数据来证明或反驳，从而查出错误所在。

- **演绎法。**根据测试结果，列出所有可能的错误原因。分析已有的数据，排除不可能和彼此矛盾的原因。对余下的原因，选择可能性最大的，利用已有的数据完善该假设，使假设更具体。用假设来解释所有的原始测试结果，如果能解释这一切，则假设得以证实，也就找出错误；否则，要么是假设不完备或不成立，要么有多个错误同时存在，需要重新分析，提出新的假设，直到发现错误为止。

3. 5. 4 系统文档

信息系统的文档，是系统建设过程的“痕迹”，是系统维护人员的指南，是开发人员与用户交流的工具。规范的文档意味着系统是按照工程化开发的，意味着信息系统的质量有了形式上的保障。文档的欠缺、文档的随意性和文档的不规范，极有可能导致原来的开发人员流动以后，系统不可维护、不可以升级，变成一个没有扩展性、没有生命力的系统。信息系统的文档，不但包括应用软件开发过程中产生的文档，还包括硬件采购和网络设计中形成的文档；不但包括上述有一定格式要求的规范文档，也包括系统建设过程中的各种来往文件、会议纪要、会计单据等资料形成的不规范文档，后者是建设各方谈判甚至索赔的重要依据；不但包括系统实施记录，也包括程序资料 and 培训教程等。

文档在系统开发人员、项目管理人员、系统维护人员、系统评价人员以及用户之间的多种作用总结如

下:

- **用户与系统分析人员在系统规划和系统分析阶段通过文档进行沟通。**这里的文档主要包括可行性研究报告、总体规划报告、系统开发合同、系统方案说明书等。有了文档,用户就能依次对系统分析员是否正确理解了系统的需求进行评价,如不正确,可以在已有文档基础上进行修正。

- **系统开发人员与项目管理人员通过文档在项目期内进行沟通。**这里的文档主要有系统开发计划(包括工作任务分解表、网络图、甘特图、预算分配表等)、系统开发月报以及系统开发总结报告等项目管理文件。有了这些文档,每个项目成员就会明白自己的目的及可用的资源和约束,项目管理人员也有了考评的依据。

- **前期开发人员与后期开发人员通过书面文档进行沟通。**这里的文档主要有系统开发各阶段的文档浓口系统方案说明书、系统设计说明书等。有了这些文档,不同阶段之间的开发人员就可以进行工作的顺利衔接,同时还能降低因为人员流动带来的风险,因为接替人员可以根据文档理解前面人员的设计思路或开发思路。

- **系统测试人员与系统开发人员通过文档进行沟通。**系统测试人员可以根据系统方案说明书、系统开发合同、系统设计说明书、测试计划等文档对系统开发人员所开发的系统进行测试。__系统测试人员再将评估结果撰写成系统测试报告。

- **系统开发人员与用户在系统运行期间进行沟通。**用户通过,系统开发人员撰写的文档运行系统。这里的文档主要是用户手册和操作指南。

- **系统开发人员与系统维护人员通过文档进行沟通。**这里的文档主要有系统设计说明书和系统开发总结报告。一有的开发总结报告写得很详细,分为研制报告、技术报告和技术手册3个文档,其中的技术手册记录了系统开发过程中的各种主要技术细节。这样,即使系统维护人员不是原来的开发人员,也可以在这些文档的基础上进行系统的维护与升级。

- **用户与维护人员在运行维护期间进行沟通。**用户在使用信息系统过程中,将运行过程中的问题进行记载,形成系统运行报告和维修修改建议。系统维护人员根据维护修改建议以及系统开发人员留下的技术手册等文档,对系统进行维护和升级。

3.5.5 系统转换

在进行新老系统转换以前,首先要进行新系统的试运行。在系统测试、调试中,我们使用的是系统测试数据,有些实际运行中可能出现的问题,很难通过这些数据被发现。所以,一个系统开发后,让它实际运行一段时间,是对系统最好的检验和测试方法。

系统试运行阶段的工作主要有:

- 对系统进行初始化、输入各原始数据记录。
- 记录系统运行的数据和状况。
- 核对新系统输出和老系统(人工或计算机系统)输出的结果。
- 对实际系统的输入方式进行考察(是否方便、效率如何、安全可靠、误操作保护等)。
- 对系统实际运行、响应速度(包括运算速度、传递速度、查询速度、输出速度等)进行实际测试。

新系统试运行成功之后,就可以在新系统和老系统之间互相转换。

新旧系统之间的转换方式有直接转换、并行转换和分段转换。

- **直接转换。**直接切换就是在确定新系统运行准确无误时,立刻启用新系统,终止老系统运行。这种方式对人员、设备费用很节省。这种方式一般适用于一些处理过程不太复杂,数据不很重要的场合。

- **并行转换。**这种切换方式是新老系统并行工作一段时间,经过一段时间的考验以后,新系统正式替代老系统。对于较复杂的大型系统,它提供了一个与旧系统运行结果进行比较的机会,可以对新旧两个系统的时间要求、出错次数和工作效率给以公正的评价。当然由于与旧系统并行工作,消除了尚未认识新系统之前的紧张和不安。在银行、财务和一些企业的核心系统中,这是一种经常使用的切换方式。它的主要特点是安全、可靠,但费用和工作量都很大,因为在相当长时间内系统要两套班子并行工作。

- **分段转换。**分段转换又称逐步转换、向导转换、试点过渡法等。这种切换方式实际上是以上两种切换方式的结合。在新系统全部正式运行前,一部分一部分地代替老系统。那些在转换过程中还没有正式运行的部分,可以在一个模拟环境中继续试运行。这种方式既保证了可靠性,又不至于费用太大。但是这种分段转换要求子系统之间有一定__的独立性,对系统的设计和实现都有一定的要求,否则就天法实现这种分段转换的设想。

在实际工作中,切换方法较为灵活。一个信息系统从使用到成熟再到提高,是一个较长的过程。只有遵循数据处理的阶段性,信息系统才能健康发展。现以一个连锁企业开始实施新系统为例。

(1) 初始阶段：企业首先为应用系统做基本资料的准备，进行总部、“配送”核心系统的实施。

(2) 推广阶段：总部、“配送”系统稳定后，先从 1-2 家门店试点开始，以门店核心模块为主，完成门店与总部的信息交换、物流过程。然后再逐步推广门店系统，直至完成所有门店的联网工作。

(3) 控制阶段：所有门店联网完成后，进行准确、及时的基本数据采集和调整工作。

(4) 集成阶段：再考虑自动补货、自动配货、财务接口等高级模块应用的工作。

(5) 管理阶段：进入数据的全面启用和介入管理决策。最后，系统步入成熟阶段。

实际上每个企业在不同阶段的发展过程中，对具体问题的解决方法也是不同的，随时都会进行以上阶段的周期重复，随着每次重复时起点的不断升高，整个企业的数据应用水平也就随之逐步提高了。

3.6 系统维护与评价

3.6.1 系统维护概述

3.6.1.1 系统可维护性概念

系统的可维护性可以定性的定义为：维护人员理解、改正、改动和改进这个软件的难易程度。提高可维护性是开发管理信息系统所有步骤的关键目的，系统是否能被很好地维护，可用系统的可维护性这一指标来衡量。

1. 系统的可维护性的评价指标

- **可理解性**。指别人能理解系统的结构、界面功能和内部过程的难易程度。模块化、详细设计文档、结构化设计和良好的高级程序设计语言等，都有助于提高可理解性。

- **可测试性**。诊断和测试的容易程度取决于易理解的程度。好的文档资料有利于诊断和测试，同时，程序的结构、高性能的测试工具以及周密计划的测试工序也是至关重要的。为此，开发人员在系统设计和编程阶段就应尽力把程序设计成易诊断和测试的。此外，在系统维护时，应该充分利用在系统调试阶段保存下来的调试用例。

- **可修改性**。诊断和测试的容易程度与系统设计所制定的设计原则有直接关系。模块的耦合、内聚、作用范围与控制范围的关系等，都对可修改性有影响。

2. 维护与软件文档

文档是软件可维护性的决定因素。——由于长期使用的大型软件系统在使用过程中必然会经受多次修改，所以文档显得非常重要。

软件系统的文档可以分为用户文档和系统文档两类。用户文档主要描述系统功能和使用方法，并不关心这些功能是怎样实现的；系统文档描述系统设计、实现和测试等各方面的内容。

可维护性是所有软件都应具有的基本特点，必须在开发阶段保证软件具有可维护的特点。在软件工程的每一个阶段都应考虑并提高软件的可维护性，在每个阶段结束前的技术审查和管理复查中，应该着重对可维护性进行复审。

在系统分析阶段的复审过程中，应该对将来要改进的部分和可能会修改的部分加以注解并指明，并且指出软件的可移植性问题以及可能影响软件维护的系统界面；在系统设计阶段的复审期间，应该从容易修改、模块化和功能独立的目的出发，评价软件的结构和过程；在系统实施阶段的复审期间，代码复审应该强调编码风格和内部说明文档这两个影响可维护性的因素。在完成了每项维护工作之后，都应该对软件维护本身进行认真的复审。

3. 软件文档的修改

维护应该针对整个软件配置，不应该只修改源程序代码。如果对源程序代码的修改没有反映在设计文档或用户手册中，可能会产生严重的后果。每当对数据、软件结构、模块过程或任何其他有关的软件特点做了改动时，必须立即修改相应的技术文档。不能准确反映软件当前状态的设计文档可能比完全没有文档更坏。在以后的维护工作中很可能因文档不完全符合实际而不能正确理解软件，从而在维护中引入过多的错误。

3.6.1.2 系统维护的内容及类型

系统维护主要包括硬件设备的维护、应用软件的维护和数据的维护。

1. 硬件维护

硬件的维护应有专职的硬件维护人员来负责，主要有两种类型的维护活动，一种是定期的设备保养性维护，保养周期可以是一周或一个月不等，维护的主要内容是进行例行的设备检查与保养，易耗品的更换与安装等；另一种是突发性的故障维护，即当设备出现突发性故障时，由专职的维修人员或请厂方的技术人员来排除故障，这种维修活动所花时间不能过长，以免影响系统的正常运行。

2. 软件维护

软件维护主要是指根据需求变化或硬件环境的变化对应用程序进行部分或全部的修改。修改时应充分

利用源程序，修改后要填写程序修改登记表，并在程序变更通知书上写明新老程序的不同之处。

软件维护的内容一般有以下几个方面：

- **正确性维护**，是指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误。这方面的维护工作量要占整个维护工作量的 17%-21%。所发现的错误有的不太重要，不影响系统正常运行，其维护工作可随时进行；而有的错误非常重要，甚至影响整个系统的正常运行，其维护工作必须制定计划，进行修改，并且要进行复查和控制。

- **适应性维护**，是指使应用软件适应信息技术变化和管理需求变化而进行的修改。这方面的维护工作量占整个维护工作量的 18%-25%。由于目前计算机硬件价格的不断下降，各类系统软件层出不穷，人们常常为改善系统硬件环境和运行环境而产生系统更新换代的需求；企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。这些因素都将导致适应性维护工作的产生。进行这方面的维护工作也要像系统开发一样，有计划、有步骤地进行。

- **完善性维护**，这是为扩充功能和改善性能而进行的修改，主要是指对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。这些功能对完善系统功能是非常必要的。另外还包括对处理效率和编写程序的改进，这方面的维护占整个维护工作的 50%-60%，比重较大，也是关系到系统开发质量的重要方面。这方面的维护除了要有计划、有步骤地完成外，还要注意将相关的文档资料加入到前面相应的文档中去。

- **预防性维护**，为了改进应用软件的可靠性和可维护性，为了适应未来的软硬件环境的变化，应主动增加预防性的新的功能，以使应用系统适应各类变化而不被淘汰。比如将专用报表功能改成通用报表生成功能，以适应将来报表格式的变化。这方面的维护工作量占整个维护工作量的 4% 左右。

3.数据维护

数据维护工作主要是由数据库管理员来负责，主要负责数据库的安全性和完整性以及进行并发性控制。数据库管理员还要负责维护数据库中的数据，当数据库中的数据类型、长度等发生变化时，或者需要添加某个数据项、数据库时，要负责修改相关的数据库、数据字典，并通知有关人员。另外数据库管理员还要负责定期出版数据字典文件及一些其他数据管理文件，以保留系统运行和修改的轨迹。当系统出现硬件故障并得到排斥后要负责数据库的恢复工作。

数据维护中还有一项很重要的内容，那就是代码维护。不过代码维护发生的频率相对较小。代码的维护应由代码管理小组进行。变更代码应经过详细讨论，确定之后要用书面形式贯彻。代码维护的困难往往不在于代码本身的变更，而在于新代码的贯彻。为此，除了成立专门的代码管理小组外，各业务部门要指定专人进行代码管理，通过他们贯彻使用新代码。这样做的目的是要明确管理职责，有助于防止和更正错误。

3.6. 1. 3 系统维护的管理和步骤

要强调的是，系统的修改往往会“牵一发而动全身”。程序、文件、代码的局部修改都可能影响系统的其他部分。因此，系统的维护工作应有计划有步骤地统筹安排，按照维护任务的工作范围、严重程度等诸多因素确定优先顺序，制定出合理的维护计划，然后通过一定的批准手续实施对系统的修改和维护。

通常对系统的维护应执行以下步骤：

- (1) **提出维护或修改要求**。操作人员或业务领导用书面形式向系统维护工作的主管人员提出对某项工作的修改要求。这种修改要求一般不能直接向程序员提出。

- (2) **领导审查并做出答复，如同意修改则列入维护计划**。系统主管人员进行一定的调查后，根据系统的情况和工作人员的情况，考虑这种修改是否必要、是否可行，做出是否修改、何时修改的答复。如果需要修改，则根据优先程度的不同列入系统维护计划。计划的内容应包括维护工作的范围、所需资源、确认的需求、维护费用、维护进度安排以及验收标准等。

- (3) **领导分配任务，维护人员执行修改**。系统主管人员按照计划向有关的维护人员下达任务，说明修改的内容、要求、期限。维护人员在仔细了解原系统的设计和开发思路的情况下对系统进行修改。

- (4) **验收维护成果并登记修改信息**。系统主管人员组织技术人员对修改部分进行测试和验收。验收通过后，将修改的部分嵌入系统，取代旧的部分。维护人员登记所做的修改，更新相关的文档，并将新系统作为新的版本通报用户和操作人员，指明新的功能和修改的地方。

在进行系统维护过程中，还要注意维护的副作用。维护的副作用包括两个方面，一是修改程序代码有时会发生灾难性的错误，造成原来运行比较正常的系统变得不能正常运行。为了避免这类错误，要在修改工作完成后进行测试，直至确认和复查无错为止；二是修改数据库中数据的副作用，当一些数据库中的数据发生变化时可能导致某些应用软件不再适应这些已经变化了的数据而产生错误。为了避免这类错误，一

是要有严密的数据描述文件，即数据字典系统；二是要严格记录这些修改并进行修改后的测试工作。

总之，系统维护工作是信息系统运行阶段的重要工作内容，必须予以充分的重视。维护工作做得越好，信息系统的作用才能够得以充分发挥，信息系统的寿命也就越长。

3.6.2 系统评价

3.6.2.1 系统评价的目的和任务

信息系统的评价分为广义和狭义两种。广义的信息系统评价是指从系统开发的一开始到结束的每一阶段都需要进行评价。狭义的信息系统评价则是指在系统建成并投入运行之后所进行的全面、综合的评价。

按评价的时间与信息系统所处的阶段的关系，又可从总体上把广义的信息系统评价分成立项评价、中期评价和结项评价。

- **立项评价。**指信息系统方案在系统开发前的预评价，即系统规划阶段中的可行性研究。评价的目的是决定是否立项进行开发，评价的内容是分析当前开发新系统的条件是否具备，明确新系统目标实现的重要性和可能性，主要包括技术上的可行性、经济上的可行性、管理上的可行性和开发环境的可行性等方面。由于事前评价所用的参数大都是不确定的，所以评价的结论具有一定的风险性。

- **中期评价。**项目中期评价包含两种含义，一是指项目方案在实施过程中，因外部环境出较大变化，比如市场需求变化、竞争性技术或更完美的替代系统的出现，或者发现原先设计有重大失误等，需要对项目的方案进行重新评价，以决定是继续执行还是中止该方案；另一种含义也可称为阶段评价，是指在信息系统开发正常的情况下，对系统设计、系统分析、系统实施阶段的阶段性成果进行评价，由于一般都将阶段性成果的提交视为信息系统建设的里程碑，所以，阶段评估又可叫里程碑式评价。

- **结项评价。**信息系统的建设是一个项目，是项目就需要有终结时间。结项评价是指项目准备结束时对系统的评价，一般是指在信息系统投入正式运行以后，为了了解系统是否达到预期的目的和要求而对系统运行的实际效果进行的综合评价。所以，结项评价又是狭义的信息系统评价。信息系统项目的鉴定是结项评价的一正规的形式。结项评价的主要内容包括系统性能评价、系统的经济效益评价以及企业管理效率提高、管理水平改善、管理人员劳动强度减轻等间接效果。通过结项评价，用户可以了解系统的质量和效果，检查系统是否符合预期的目的和要求；开发人员可以总结开发工作的经验、教训，这对今后的工作十分有益。

在对信息系统进行评价考核的时候，应该注意以下几个问题。

首先，信息系统通过基本资料录入、进货、订货、盘点、零售等各个环节采集进来，其中任意一个环节的数据录入出现问题，都将导致最终报表的不准确，而报表不准确就意味着企业决策者无法根据报表决定企业的运作，更谈不上数据分析和决策支持了。这也是我们目前大部分使用了信息系统的企业普遍存在的问题。究竟是什么原因导致了数据采集的不准确呢？一些企业错误地将数据准确性作为考核信息部的一个指标，其实数据不准确更多源于管理上存在的问题，正因为数据源头非常多，所造成的数据不一致的问题不是信息部都能解决的，最终数据采集的成败将由最高管理层对数据采集各环节的管理力度所决定。而数据采集的成败又将最终决定整个信息系统应用的成败。

此外，信息系统并不是万能系统。系统应用的过程中，有些问题是信息系统擅长解决的，如大量的、重复的、规范的事务处理；而有些问题是信息系统不擅长解决的，如特殊的、偶然的、不规范的经营管理内容。让信息系统做不擅长的工作，势必在应用的过程中，投入的管理成本远远大于它所产生的效益。对这种灵活、多变的情况，不妨采用人工处理或通过制度的限制，尽量避免不规范的行为频繁发生，从而真正实现企业简单复制、快速扩张、规模效益的目的。

3.6.2.2 系统评价的指标

我们从以下几方面综合考虑，建立起一套指标体系理论框架：

- 从信息系统的组成部分出发，信息系统是一个由人机共同组成的系统，所以可以按照运行效果和用户需求（人）、系统质量和技术条件（机）这两条线索构造指标。

- 从信息系统的评价对象出发，对于开发方来说，他们所关心的是系统质量和技术水平；对于用户方而言，关心的是用户需求与运行质量；系统外部环境则主要通过社会效益指标来反映。

- 从经济学角度出发，分别按系统成本、系统效益和财务指标等 3 条线索建立指标。

各项指标列出如下。

一、系统质量

(1) 执行准确性，响应速度，信息存储量，界面质量

(2) 安全性，可靠性，文档齐全

(3) 数据共享性，易维护性，容错性

二、技术水平

- (1) 技术先进性
 - 软硬件先进性一
 - 开发技术先进性
 - 软件可重用性
- (2) 技术首创性
- (3) 开发效率

三、运行质量

- (1) 直接应用人员的结构, 素质
- (2) 系统运行率.
- (3) 系统维护率

四、用户需求

- (1) 领导重视程度
- (2) 功能需求满足程度(适用程度)
- (3) 人一机交互的友善程度
- (4) 系统价格可接受程度(性能/价格比)

五、系统成本

- (1) 开发成本
 - 硬件成本(购置, 基建, 安装, 调试等)
 - 软件成本(开发, 培训, 系统切换等)
- (2) 运行成本(人员费用, 消耗材料, 技术资料, 折旧等)
- (3) 管理成本(监理, 审计, 服务, 行政等)
- (4) 维护成本(硬件, 软件, 纠错, 适应, 完善等)

六、系统效益

- (1) 经济效益,
 - 按系统功能(如生产管理, 财务管理等)
 - 按服务对象(如企业, 政府等)
 - 按效益类型(如直接/间接, 有形/无形等)
 - 按技术特征(如 EDPS, MIS, DSS 等)
- (2) 社会效益
 - 对社会的影响程度
 - 对本企业的影响程度
 - 福利, 就业, 伦理道德

七、财务评价

- (1) 投资指标(如企业管理费, 非生产人员工资等)
- (2) 收益指标(如销售额, 利润等)
- (3) 综合指标(如净现值, 净现值率, 内部收益率等)

3.6.3 系统运行管理

3.6.3.1 运行管理制度

规范管理的企业, 每一项具体的业务都有一套科学的运行制度。信息系统也不例外, 同样需要一套管理制度, 以确保信息系统的正常和安全的运行。

1. 各类机房安全运行管理制度

信息系统的运行制度, 首先表现为物理意义上的机房必须处于监控之中。机房安全运行制度应该包括如下主要内容:

- 身份登记与验证出人。
- 带人带出物品检查。
- 参观中心机房必须经过审查。
- 专人负责启动、关闭计算机系统。
- 对系统运行状况进行监视, 跟踪并详细记录运行信息。
- 对系统进行定期保养和维护。
- 操作人员在指定的计算机或终端上操作, 对操作内容按规定进行登记。

- 不做与工作无关的操作，不运行来历不明的软件。
- 不越权运行程序，不查阅无关参数。
- 操作异常，立即报告。『

2.信息系统的其他管理制度

信息系统的运行制度，还表现为软件、数据、信息等其他要素必须处于监控之中。信息系统的其他管理制度主要包括如下内容：

- 必须有重要的系统软件、应用软件管理制度。如系统软件的更新维护，应用软件的源程序与目标程序分离等。
- 必须有数据管理制度。例如重要输入数据、输出数据的管理。
- 必须有密码口令管理制度，做到口令专管专用一，定期更改并在失密后立即报告。
- 必须有网络通信安全管理制度，实行网络电子公告系统的用户登记和对外信息交流的管理制度。
- 必须有病毒的防治管理制度。及时检测、清除计算机病毒，并备有检测、清除的记录。
- 必须有人员调离的安全管理制度。例如，人员调离的同时马上收回钥匙、移交工作、更换口令、取消账号，并向被调离的工作人员申明其保密义务。
- 建立安全培训制度，进行计算机安全法律教育、职业道德教育和计算机安全技术教育。对关键岗位的人员进行定期考核。
- 建立合作制度。加强与相关单位的合作，及时获得必要的信息和技术支持。

除此之外，任何信息系统的运行都必须遵守国家的有关法律和规定，特别是关于计算机信息系统安全的法律规定。

3.6.3.2 日常运行管理内容

信息系统的日常运行管理是为了保证系统能长期有效地正常运转，具体有系统运行情况的记录、系统运行的日常维护等工作。

对系统运行情况的记录应事先制定登记格式和登记要点，人工记录的系统运行情况和系统自动记录的运行信息，都应作为基本的系统文档按照规定的期限保管。这些文档既可以在系统出现问题时查清原因和责任，还能作为系统维护的依据和参考。

1. 系统运行情况的记录

原则上讲，从每天计算机的打开、应用系统的进入、功能项的选择与执行，到下班前的数据备份、存档、关机等，都要对系统软硬件及数据等的运作情况做记录。运行情况有正常、不正常与无法运行三种情况。可对正常情况不予记录，对于不正常情况和无法运行情况则应将所见的现象、发生的时间及可能的原因做尽量详细的记录。因为这些信息对系统问题的分析与解决有重要的参考价值。

2. 审计踪迹

审计踪迹(audit trail)就是指系统中设置了自动记录功能，能通过自动记录的信息发现或判明系统的问题和原因。这里的审计有两个特点，一是每日都进行，二是主要是技术方面的审查。

在审计踪迹系统中，建立审计日志是一种基本的方法。通过日志，系统管理员可以了解到有哪些用户在什么时间、以什么样的身份登录到系统，也可以查到对特定文件和数据所进行的改动。

现在大多数的操作系统和数据库都提供了跟踪并自动记录的功能，一些数据库系统中还提供审计踪迹数据字典，使用者可以用预先定义的审计踪迹数据字典视图来观察审计踪迹数据。对于审计内容可以在 3 个层次上设定：

- **语句审计：**语句审计是对于特定的数据库语句所进行的审计。例如在一个系统文件中记录所有使用了 Create 命令的信息。
- **特权审计：**指对于特定的权限使用所进行的审计。
- **对象审计：**规定对特定的对象审计特定的语句。例如可以审计在某个文件上进行了修改其内容的语句。

3.审查应急措施的落实

为了减少意外事件引起的对信息系统的损害，首先要制定应付突发性事件的应急计划，然后每日要审查应急措施的落实情况。

应急计划主要针对一些突发性的、灾害性的事件，例如火灾、水害等。因此，机房值班员每日都应仔细审查相应器材和设备是否良好，相应资源是否做好了备份。

资源备份包括两个方面的工作，即数据备份和设备备份，数据备份是必须要做的，在关键的领域，还必须进行设备备份。应将备份文件拷贝到远离主机或文件中心的其他主机或者存储库中，保证备份文件是一次灾害和事件影响不到的地方。

在维护信息系统正常运行过程中还应对计算机的使用及打印机、墨粉的消耗等制定合理的管理方法。

3. 6. 3. 3 系统软件及文档管理

由于计算机科学技术的迅速发展,新的硬、软件不断推出,使系统的外部环境发生变化。这里的外部环节不仅包括计算机硬件软件的配置,还包括数据库、数据存储方式在内的数据环境。为了适应企业的发展,版本更新和升级必不可少。

信息系统的文档与其他类型的文档一样，也具有它自身的生命周期，分为创建期、处理期、存储期、使用期、销毁期。每种文档都处于生命周期中的某一时期。当然周期的划分也不是绝对的，各周期有时是不能截然分开的。信息系统文档的生命周期普遍要比信息系统的生命周期长。也就是说，绝大多数信息系统的文档要在相应的信息系统淘汰 3-5 年后才能销毁。

• **文档管理的制度化。**必须形成一整套的文档管理制度，其内容可以包含文档的标准、修改文档和出档的条件、开发人员在系统建设不同时期就其文档建立工作应承担的责任和任务。根据这一套完善的来最终协调、控制系统开发工作，并依次对每一个开发成员的工作进行评价。

• **文档要标准化、规范化。**在系统开发前必须首先选择或制定文档标准。在统一标准制约下，开发人员负责建立所承担任务的文档资料。对于已有参考格式和内容的文档，应尽量按相应的规范撰写文档。对于没有参考格式的文档，如需求变更申请书，应该在项目组内部出台相应的规范和格式。

• **文档管理的人员保证。**项目小组应设文档组或至少一位文档保管人员，负责集中保管本项目已有文档的两套主文本。两套文本内容应完全一致。其中的一套可按一定手续，办理借阅。

- **维护文档的一致性。**信息系统开发建设过程是一个不断变化的动态过程，一旦需要对某一文档进行修改时，要及时、准确地修改与之相关联的文档；否则将会引起系统开发工作的混乱。而这一过程又必须有相应的制度来保证。

- **维持文档的可追踪性。**由于信息系统开发的动态性，系统的某种修改是否最终有效，要经过一段时间的检验，因此文档要分版本来实现。而各版本的出版时机及要求也要有相应的制度。

信息系统的文档一般是按文件级管理，即以文件作为管理对象的基本单位。反映一份文件各个方面属性的集合就构成一条记录，一般应包括文档名、责任者、事件、密级、保管期限、分类号、关键词等项目。

4.1 概述

企业系统规划(Business Systems Planning, BSP)法,是由 IBM 公司研制的指导企业信息系统规划的法,虽然研制始于 20 世纪 70 年代,但其方法和思至今仍有指导意义。它着重于帮助企业做出信息系的规划,来满足其近期和长期的信息需求。



实行 BSP 研究的前提是,在企业内有改善计算机信息系统的要求,并且有为建设这一系统而建立总的战略的需要。因而 BSP 的基本概念与组织内的信息系统的长期目标有关。

可以将 BSP 看成是一个转化过程,即将企业的战略转化成信息系统的战略(见图 4.1)。

因此,重要的是,一个企业要能表达出其长期的目标。对于一些企业,可以从计划中找到原则性的回答。对于另一些企业,可能还没有计划,了解企业的战略就成了BSP研究内容的一部分。

2. 一个信息系统的战略应当表达出企业的各个管理层次的需求

对于不同层次的管理活动有着不同特点的信息需求，有必要建立一个合理的框架，并据此来定义信息系统。首先，信息系统应强调对管理决策的支持。一般认为，在任一企业内同时存在着 3 个不同的计划、控制层。

- **战略计划层：**是决定组织的目标，决定达到这些目标所需用的资源以及获取、使用、分配这些资源的策略的过程。

- **管理控制层：**通过这一过程，管理者确认资源的获取及组织的目标是否有效地使用了这些资源。

- **操作控制层：**保证有效率地完成具体的任务。

3. 一个信息系统应该向整个企业提供一致的信息

信息的不一致性，源于“自下而上”的开发数据处理系统的做法。在企业的各部门中信息在形式上、定义上和时间上有差异。为了强调数据的一致性，有必要把数据作为一种资源来统一管理，它不应由一个局部的组织来控制，而应由一个中央部门来协调，使数据对企业有全面性的价值，被企业各单位共享。管理部门要负责制定数据的一致性定义、技术实现，以及使用和数据安全性的策略和规程。

4. 一个信息系统应该适应组织机构和管理体制的改变

信息系统应具有适应性。在一个发展的企业中，数据处理系统决不要削弱或妨碍管理部门的应变能力，而应当有能力在企业的长期的组织机构和管理体制的变化中发展自己，而不受到大的冲击。

为了实现上述目的，要有适当的关于信息系统的设计技术，BSP 采用了企业过程的概念，这种技术要独立于组织机构和各种因素。对于任一类型的企业可以从逻辑上定义出一组过程，只要企业的产品或服务基本不变，则过程改变会极小。

5. 一个信息系统的战略规划，应当由总体信息系统结构中的子系统开始实现

支持整个企业需求的总信息系统一般规模都较大，因而有必要建立信息系统的长期目标和规划，从而形成了 BSP 对大型信息系统而言是“自上而下”的系统规划、“自下而上”的分步实现。

应用 BSP 这种实现战略，信息系统就能按部就班以模块化方式进行建设，并照顾到企业的重点、资金情况和其他考虑。

总结起来，BSP 方法建立了信息系统建设的若干原则。方法本身是可以灵活运用，即方法中的某些步骤和技巧可根据具体情况变化，但这些基本原则不能违背，它们是 BSP 灵魂。

4.1.2 BSP 的目标

BSP 的主要目标是提供一个信息系统规划，用以支持企业短期的和长期的信息需要。其具体目标可归纳如下：

- 为管理者提供一种形式化的、客观的方法，明确建立信息系统的优先顺序，而不考虑部门的狭隘利益，并避免主观性。

- 为具有较长生命周期系统的建设和投资提供保障。由于系统是基于业务活动过程的，因而不因机构变化而失效。

- 为了以最高效率支持企业目标，BSP 提供数据处理资源的管理。

- 增加负责人的信心，坚信收效高的主要的信息系统能够被实施。

- 通过提供响应用户需求和用户优先的系统，改善信息系统管理部门和用户之间的关系。

应将数据作为一种企业资源加以确定，为使每个用户更有效地使用这些数据，要对这些数据进行统一规划、管理和控制。

由 BSP 研究所得到的规划不应当看成是一成不变的，它只是在某一阶段对事物的最好认识。BSP 方法的真正价值在于提供了下面的机会：

- 创造一种环境和提出初步行动计划，使企业能依此对未来的系统和优先次序的改变做出反应，不至造成设计的重大失误。

- 定义信息系统的职能，继续规划过程。

4.2 BSP 方法的研究步骤

4.2.1 研究项目的确立

BSP 的经验说明，除非得到了最高领导者和某些最高管理部门参与研究的承诺，不要贸然开始 BSP 的研究。因为研究必须反映最高领导者关于企业的观点，研究的成果取决于管理部门能否向研究组提供企业的现状，他们对于企业的理解和对信息的需求。因此在一开始时就要对研究的范围和目标、应交付的成果取得一致意见，避免事后的分歧，这是至关重要的。

4. 2. 2 研究准备工作

在取得领导赞同以后，最重要的是选择研究组组长，要有一位企业领导用全部时间参加研究工作并指导研究组的活动。要确认参与研究的其他层次领导是否合适，并能正确地解释由他们所在部门得到的材料。

对研究组和参与研究的管理人员要有适当的培训和辅导，管理人员能较好地提供材料，使研究组能充分地利用这些材料。

要尽快地选好调查对象，并让他们事先准备，安排会面的日程以及向研究组提供信息。准备工作阶段的主要成果应当是研究计划的制定，内容包括一个研究计划、一个会谈日程、一个同主持单位一起做复查的时间表和一个研究报告大纲。

4.2.3 研究的主要活动

如图 4.2 所示，除了项目确立和准备工作外，BSP 研究还包含 11 个主要活动。

1. 研究开始阶段

BSP 研究首项活动是企业情况介绍，全体研究组成员要参加。介绍内容有 3 方面：

- 由管理部门负责人再次重申研究的目标，期望的成果和研究的远景，以及与企业活动和目标的关系。
- 由研究组长介绍情况使研究组成员熟悉有关资料，并讨论有关企业的决策过程、组织职能、关键人物、存在问题、开发策略、敏感问题、计划中的或正在进行着的变化、数据处理部门的形象以及用户对数据处理工作的支持等。研究组长应对有关问题提出自己的评论和看法。
- 由信息系统负责人和管理人员做数据处理部门的情况介绍，介绍数据处理部门的历史和现状、目前的主要活动、计划中的变化和主要存在的问题。

通过上面介绍，使研究组成员加深对企业及其目前存在的和计划中的数据处理业务的全面理解。

2. 定义企业过程

企业过程被定义为在企业资源管理中所需要的、逻辑上相关的一组决策和活动。这些活动将作为安排同管理人员面谈、确定信息总体结构、分析问题、识别数据类以及随后许多研究项目的基础。

3. 定义数据类

企业过程被定义后，即要识别和定义由这些过程产生、控制和使用的数据。数据类是指支持企业所必要的逻辑上相关的数据，即数据按逻辑相关性归成类，这样有助于企业的数据库的长期开发。

4. 分析现存系统支持

弄清目前的数据处理如何支持企业，进而对将来的行动提出建议。对目前存在的组织、企业过程、数据处理和数据文件进行分析，发现不足和冗余，明确责任，并进一步增进对企业过程的理解。

5. 确定管理部门对系统的要求

BSP 方法必须考虑管理人员对系统的要求，并通过与高层管理人员的对话来确认研究组已做的工作，明确目标、问题、信息需求和它们的价值，并建立同最高管理部门的联系，争取他们的参与，使 BSP 研究和管理部门间建立新的、更密切的关系。

6. 提出判断和结论

通过与管理部的会谈对所收集的材料作出确认、解释和补充。要对问题进行分析并联系到企业过程，以便指导安排项目的优先顺序，并指明信息的改进将有助于解决问题。

7. 定义信息总体结构

定义信息总体的结构是由对目前情况的研究转向对将来计划的综合的主要步骤。信息总体结构刻画出将来的信息系统和相应的数据，使系统和它们产生的数据结构化和条理化。由于此项工作是描绘将来信息系统的蓝图，因此全体研究组成员都要加以重视。

8. 确定总体结构中的优先顺序

研究组要确定系统和数据库开发优先顺序。对信息总体结构中的子系统的项目进行排列，然后根据确定的准则评定项目的重要性，从而确定开发顺序。

9. 评价信息资源管理工作

为了使信息总体结构能高效率地开发、实施和运行，必须建立一个可控的环境，信息系统的过程必须加以优化，使其不断地随着技术和业务战略的变化而改变。

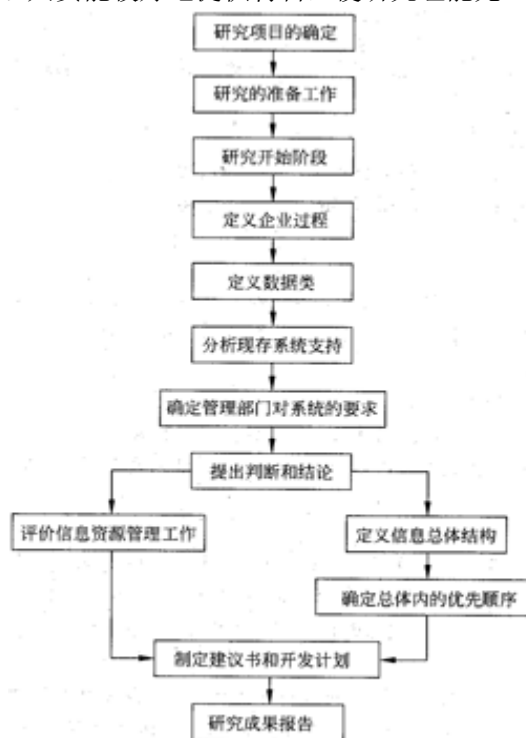


图 4.2 BSP 研究方法的流程

10.制定建议书和开发计划

开发计划是帮助管理部门对所建议的项目作出决策，这些项目是由总体结构优先顺序和信息管理部门的建议来决定的。开发计划要确定具体的资源、日程和其他项目间的关系，并需估计工作规模，以便管理部门进行调度。

11. 研究成果报告

最后在汇报会上向最高管理部门提交格式预先定好的研究报告。报告内容、实施建议要征得最高管理部门的同意并参与意见。

4.3 定义企业过程

4.3.1 过程定义的目的和条件

企业过程为企业资源管理所需要的、逻辑相关的一组决策和活动。它们的分析和识别无需顾及与组织机构的联系。

定义企业过程的目的和作用可归纳为：

- 使信息系统尽量独立于组织机构。
- 帮助理解企业如何完成使命和目标。
- 为从操作控制过程中分离出战略计划和管理控制提供依据。
- 为定义所需要的信息结构和为确定研究的范围、模块的分解和排列、开发的优先顺序提供依据。
- 为定义关键的数据需求提供帮助。

过程定义以前，下列几点是研究的成功必要条件：

- 所有的研究成员必须参与整个活动，且在活动前对期望的成果应有一致的意见。
- 所有提供或调查的材料要记录、整理完好，以免在以后的决策和工程设计时误解或遗忘。
- 研究组成员必须建立和理解资源及资源生命周期的概念。
- 研究前收集的信息必须能对产品和资源进行说明和估计。

4.3.2 产品和资源的生命周期

管理人员的职责是在其负责的领域内有效地管理和利用资源，以支持企业的目标。产品/服务可以定义为关键的资源，它在企业过程定义中起着重要的作用。

产品/服务和其他支持性资源的四个阶段的生命周期，常常被用来逻辑地识别和组合过程。

生命周期的各个阶段可描述如下。

- (1) 需求、计划、度量和控制：决定需要多少产品和资源，获取它们的计划，以及执行计划要求的度量和控制。
- (2) 获取和实现：开发一种产品或一项服务，或者去获得开发中所需要的资源。
- (3) 经营和管理：组织、加工、修改或维护那些支持性资源，对产品/服务进行存储或服务。
- (4) 回收或分配：意味着中止企业对产品或服务的职责，标志着资源使用的结束。

生命周期的概念将有助于研究人员能结构化地、逻辑地、全面地识别过程。

4.3.3 定义过程的基本步骤

图 4.3 给出定义过程的基本步骤，它指出了定义企业过程的 3 类主要资源：计划和控制，产品/服务，支持性资源。对后两类资源的生命周期分析，能够给出它们相应的企业过程的定义，而战略计划和管理控制不是面向孤立的产品或资源，因而应作为一种独立的资源来分析，从而更好地识别全部企业过程。

4.3.3.1 计划和控制过程

准备工作阶段收集到的有关计划、关键成功因素和它们的度量标准等信息，一般可被组合成战略计划类和管理控制类。

战略计划是长远计划或发展规划，管理控制是操作计划、管理计划、资源计划。表 4.1 给出了计划和控制过程的例子。

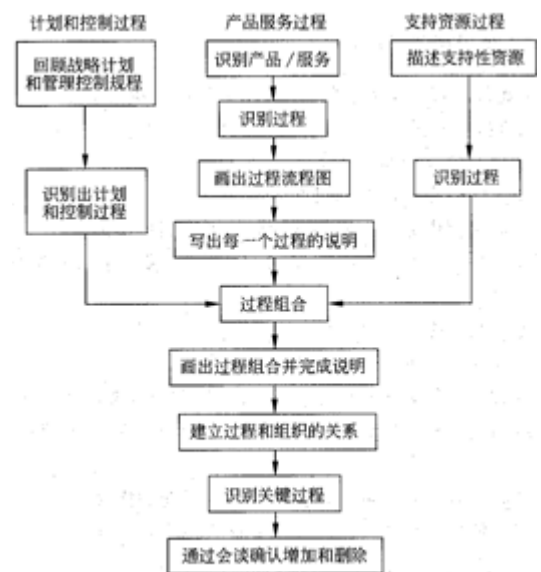


图 4.3 企业过程定义

表 4.1 计划和控制过程

战略计划	管理控制
经济预测	市场预测
组织计划	产品预测
策略制定	资金计划
目标开发	操作计划
产品系列设计	预算

4.3.3.2 产品/服务过程

定义产品/服务过程如下：首先识别企业的产品/服务，再按产品/服务的生命周期的各阶段识别过程，然后画出过程的总流程，再对每一过程定出其说明。

1.识别企业的产品/服务

对生产企业产品/服务过程的识别是易于明确的，但对于有多组或一系列产品和服务则情况较为复杂，由于产品、服务的多样化，很难有公共的过程，那么在过程识别之前必须进行分组考虑。当所有过程被识别后，再寻找可能有的公共信息需求的过程。而对于公众机构和一些服务组织，先弄清他们的目标将有助于更好地说明产品或服务。

2.按产品/服务的生命周期的各个阶段识别过程

识别过程的一般做法是从需求阶段开始，然后逐个阶段进行。注意保持各阶段所识别的过程在层次上的一致性。每一阶段的过程数是无法统一规定的，但 BSP 的研究结果表明，大多数企业过程数在 20-60 之间。一般都会识别出比实际过程多一些的过程，然后做必要的组合。

BSP 经验认为，过程定义的合理性在一定程度上依赖于研究成员对业务流程的熟悉程度和实际的工作经验。表 4. 2 给出某电子元件厂的产品/服务过程的例子。

表 4.2 产品/服务过程例子

需求阶段	获取阶段	经营管理阶段	回收或分配阶段
市场计划	工程设计和开发	订单处理和控制在	销售
市场研究	产品说明	接收和存储	订货服务
预测	工程记录	控制产品质量	运输
定价	生产安排表	检验、包装	
材料需求	生产操作	库存管理	
能力计划	采购(购置)		

3.画出产品/服务过程总流程图

产品/服务过程总流程图是企业与产品/服务有关的过程的总体描述，它可以检查与产品/服务有关的企业过程识别是否完全，有助于识别涉及到管理支持资源的过程，并作为今后定义信息结构的模型。

4. 写出每一过程的说明

写出每一过程的较详细的说明是必要的，过程说明可以用表格形式，也可以是文字形式。过程说明示例如下。

(1)生产计划：为生产满足预测需求的产品，而对材料、人员和生产设备所进行的计划活动，包括

- 资金与生产能力计划：生产需要和生产能力的协调过程。
- 原材料需求：在最优库存及节省订购量情况下，对原材料进行计算，使其满足生产进度。
- 日程安排：满足生产需要和运输需要的劳动力、设备和材料安排。
- 成本核算：确定以生产和管理的成本因素为基础的标准原材料成本和产品成本。

(2)采购：按时以最好的价格获取所需要的材料、设备和指定质量的供应品的活动，包括

- 供货者的选择：选择评价满足材料要求、包装要求和送货要求，并在价格上有竞争力的供货者。
- 订货活动：从选定的供应者处签订货单，购买生产计划规定数量和质量的原材料，购买由管理部门

批准的设备。

- 接收检查：检查数量和质量，接收或退回所购的材料、机器和供应品，并记录这些流动。

4.3.3.3 支持资源过程

1.支持资源的描述

BSP 把支持资源描述成企业为实现其目标时的消耗和使用物。基本资源有材料、资金、设备和人员 4 类。还有一些可供考虑的辅助性资源，如市场、厂商、资料等。

2. 对每一个支持资源，按生命周期各阶段进行识别过程。

表 4.3 是一个例子。

4. 3. 3.4 过程的归并和分析

1.过程的归并

前面已从计划和管理、产品/服务和支持资源 3 方面识别过程，研究组现应按照下列线索对其进行归并：

- 减少过程在层次上的不一致性，如上例中，财政计划过程和人员计划过程应属于不同层次。
- 归并有共性的过程，如获取阶段中，可将材料采购和设备采购等采购过程加以归并。

BSP 认为，正常情况下是 4-12 个过程组，而便于研究的过程的最大数目是 60 个。当然，这一数目同导出它们的资源有关。表 4. 4 是经过组合的例子。

表 4.3 支持性资源过程

资源	生命周期 4 个阶段			
	需求阶段	获取阶段	经营管理阶段	回收或分配阶段
资金	财政计划 成本控制	资金获取 应收款项	证券管理 银行业务 普通会计	付账
人员	人员计划 工资管理	招聘 调动	报酬福利 专业开发	解聘和退休
设备	资金设备 计划	设备采购 建筑物管理	机器维护 设备管理	设备处理和 安排

表 4.4 经过组合的过程表

市场	产品	日常管理	高层管理
市场计划 市场研究 市场预测	安排生产 能力计划 材料需求 操作控制	一般会计业务 成本计划 预算记账	企业计划 组织分析 程序控制 风险管理
销售操作	材料管理	财政	
区域管理 销售 行政管理 订货服务	采购 接收 库存控制 运输	财政计划 资金的获取 经费管理	
工程	设备管理	人事资源	
产品设计和开发 产品规范维护 信息控制	工程流程图 维护 设备性能	人事计划 招聘、培训和解雇 报酬	

2.画出过程组合表和完成过程说明

将每一个过程组合和它的过程都列在一张表上。对每一过程应在的过程组位置以及命名应统一意见。

3.建立企业过程与组织的联系

BSP 用建立过程/组织矩阵的方法,把企业组织机构与企业过程联系起来,它说明了每一过程与机构的联系和其决策人。

由于 BSP 研究是提供企业概况,并不详细到每一个组织实体,因而有时可将公共的类似组织表示为一个组织,例如,所有的销售办公室可列为一个组织单位。

过程/组织矩阵如表 4. 5 所示。在过程/组织矩阵中,把它们的相关程度按不同级别用不同符号标识出来。

过程/组织矩阵有助于弄清调查对象、决定对过程负责人提出的问题,并可将此矩阵作为企业管理系统手册的一个索引。

4.识别企业成功的关键过程

识别关键过程是为了决定要对企业的哪些部门做更详细的研究,了解已知问题的重要性,并提出在同管理部门面谈时要强调的项目。一般战略计划和管理控制就是这样的一些过程。

4.3.3.5 结果和应用

一般从定义企业过程中,应获得以下结果和资料:

- 过程组和它们所含过程的目录。
- 各个过程的说明。
- 关键过程名。
- 产品/服务流程图。
- 研究组对整个企业的理解和分析。

企业过程是下面研究活动的基础。企业过程的最根本的作用是了解使用信息系统来支持企业的需求和机会,这也是 BSP 研究目标。

4. 4 定义数据类

4. 4. 1 识别数据类

企业过程被识别以后,下一步就要识别和分类由这些过程所产生、控制和使用的数据。数据类是指支持企业所必要的逻辑上相关的数据。

识别数据类是为了解决下列问题:

- 了解目前支持企业过程的数据的准确度、及时性和可得性。
- 识别在建立信息总体结构中要使用的数据类。
- 发现企业过程间目前的和潜在的数据共享。
- 各个过程产生和使用了什么样的数据。
- 缺少哪些数据。
- 发现需要改进的系统。
- 确定企业的数据政策。

以企业资源为基础,通过其数据的类型识别出数据类。信息生命周期用图 4.4 表示,则数据类型就和被定义的生命周期的各阶段有关。存档类数据记录资源的状况,支持经营管理活动,仅和一个资源直接有关;事务类数据反映由于获取或分配活动引起的存档数据的变更;计划类数据包括战略规划、预测、操作日程、预算和模型,可以是数据,也可以是文本要统计类数据是历史的和综合的数据,用作对企业度量和控制。

为了识别和这些企业资源有关的数据类,可以使用如表

4. 6 所示的企业资源/数据类型矩阵。列表示主要的数据类型,行表示企业资源,对每一种企业资源,相

表 4.5 过程/组织矩阵																		
过程 \ 组织		总裁	财政副 总裁	控制 员	人事 部长	销售副 总裁	订货 控制 经理	电子 品销 售经 理	工程 副总 裁	生 产副 总 裁	工 厂 厂 长	生 产 计 划 主 任	设 备 经 理	材 料 控 制 经 理	采 购 经 理	部 门 律 师	计 划 主 任	
高层 管理	企业计划	×	×	×		/							×				×	×
	组织分析		×	×		/											×	×
	审查和控制		×	×		/												×
	风险管理	×	×	×		/											×	×
市场	计划	×	×	×		×			×	/	/	×	×					×
	研究	×				×			×	/	/							
	预测	×	×	×		×			/	×						×		
销售 操作	区域管理					×	/	×	×									
	销售	×	×			×	/	×	×									
	行政管理		/			×	/	×	×									
	订货服务	/	/	/		×	/	×	×					×	×		/	
工程	设计和开发	×				×		×	×	×	/	×	×					/
	产品说明维护					×		×	×	×	/	×	×		/			
	信息控制								×	/	/	/		/	×			
	日程安排					×	/	/	/	×	×	×		/	×			
生产	能力计划	×	×	/		×			×	×	×	×	×	×	/	/		×
	材料需求		×							×	×	×		×				
	操作	×	/	/						×	×	×	×	×	×	/	/	
材料 管理	采购		/	/						×	×	×	×	×	×	×		
	接收	/		/						×	×	×	×	×	×	×		
	库存量控制	/	×	/		×				×	×	×	×	×	×	×		
	运输	/								×	×	×	×	×	×	×		
设备 管理	工程流程图								×	×	×	×						
	维护								×	×	×	×						
	设备性能	/							×	×	×	×				×		
日常 管理	普通会计	×	×	×									×					
	成本计划	×	×	×		×	/	/	×		×	×	×	×	×	×		×
	预算	×	×	×		×	×	×	×	×	×	×	×	×	×	×		×
财政	财政计划	×	×	×														×
	资金获取	×	×	×												×	×	×
	经费管理	×	×	×													×	×
人力 资源	人员计划	×	×	×	×													×
	招聘培训				×													×
	赔偿	×	×	×	×	/			/									

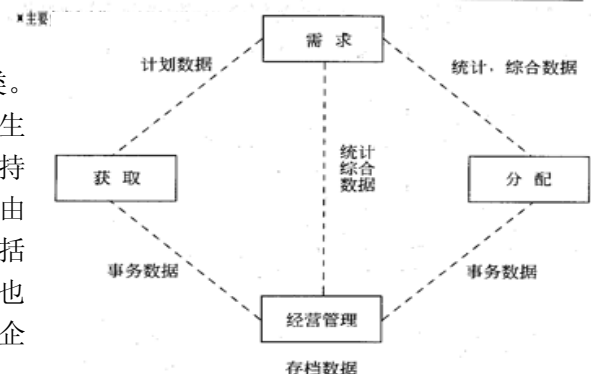


图 4.4 信息生命周期

表 4.6 识别数据类的企业资源/数据类型矩阵

企业资源	产品	顾客	设备	材料	厂商	资金	人事
数据类型	产品	顾客	设备	材料	厂商	资金	人事
存档	完成的商品 零部件	客户	设备机器负荷 工艺规程	原材料费用 材料付款单	厂家	财政 普通分类 会计	雇员 工资 技能
事务	订购	运输		采购 订购	材料 接收	收款 付款	
计划/规模	产品计划	销售区域 市场计划	设备计划 能力计划	材料需求 生产安排表		预算	人员 计划
统计/综合	产品要求	销售历史	工作进程 设备 利用率	外部需求	厂家 品性	财政 统计	生产力 效益 历史

对一个数据类型填上相应的数据类。具体处理时可以从存档类型开始识别，再到事务类型、计划和综合类型。最后经综合或分解后，可得出 30-60 个数据类。识别中并应辅以对企企业关键成功因素的调查，并提出度量和控制的数据类。

数据类的最后确定应分析各个过程使用或产生了什么数据类，确定工作包括按产品/服务生命周期顺序，构造一系列的输入—处理—输出数据类图。每个输入和输出都是待定的数据类。要识别的是输入或输出的信息实体。确定数据类的一种替代方法是选定一些企业过程作样本来制定输入—过程—输出图，图 4.5 是一个例子。

4.4.2 给出数据类定义

最后要定出每一个数据类的定义，并说明它包含什么数据，供讨论和定义数据结构用。一般来说，数据类写得越详细，在以后的研究中越不易失误，建立系统总体结构越方便，对 BSP 的后续研究越有帮助。

4.4.3 建立数据类与过程的关系

BSP 认为数据类和过程是定义企业信息系统总体结构的基础，应该建立它们之间的内在联系，并可清除在考虑定义和内容时所产生的问题。

过程/数据类矩阵是建立二者联系的工具，如表 4.7 所示。其中行表示数据类，列表示过程，并以字母 C 和 U 来表示过程对数据类产生和使用。在矩阵中，首先按关键资源的生命周期顺序放置过程，开始是计划过程，然后是度量和控制过程以及直接涉及产品的过程，最后是管理支持资源的过程；其次是根据过程产生数据的顺序来安排数据，开始是由计划过程产生的数据，接着把它所有数据列入矩阵，并在适当的行列交叉处填上 C 和 U。

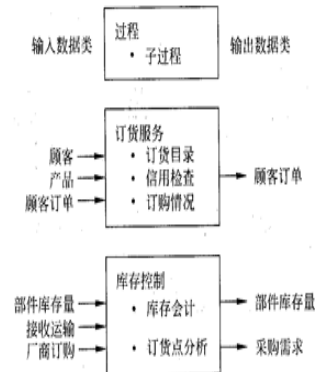


图 4.5 输入—过程—输出图例

表 4.7 过程/数据类矩阵(局部)

数据类	计划	财政	产品	部件目录	材料单	供应商	原料库存	成品库存	设备	在进行的 工作	机器负荷	待购材料	顾客	贸易地区	订货单	价格	职工
过程																	
企业计划	C	U	U						U				U			U	U
机构分析	U																
评论及控制	U	U															
财政计划	C	U							U								U
资本获取		C															
研究			U												U		
预测	U		U										U	U			
设计开发			C	C	U								U				
产品说明书维护			U	C	C	U											
采购						C											U
接收						U	U										
库存控制							C	C	U								
工序设计			U						C			U					
调度			U			U			U	C	U				U		
容量计划						U			U		C	U	U				
材料需求			U		U	U					C				U		
操作									U	U	U	C					
贸易地区管理			U										C		U		
销售			U										U	C	U		

4.5 分析当前业务与系统的关系

4.5.1 分析现行系统支持

当对企业过程和数据类有清晰的了解后，还必须对当前的数据处理工作是如何支持企业的问题有必要的了解。这样，才能为未来信息系统建设提出建议。

1.考察信息系统对过程的支持

为了在全企业范围内了解现存信息系统对各企业过程的支持，可以利用组织/过程矩阵，在其上以符号注明和过程有关的哪些组织正在接受应用系统的支持。这样可以了解哪些没有得到当前系统支持的过程，只得到部分支持的过程和有重复的系统。

具体做法见表 4.8。可给每一个当前的应用系统编号，然后将编号填入得到支持的相应单元中。

2.识别当前的数据使用情况

在现有数据类中哪些部分已由计算机处理和哪个系统使用了哪些数据类，可以通过调查用系统/数据类矩阵(见表 4.9)表达。在每一个适当的单元中用特定的符号(如 X)表示哪个系统类支持着相应的系统。

由系统/数据类矩阵可看出，有多少

表 4.8 过程/组织/当前系统矩阵

过程		销 售				生 产			
组织单元		区域管理	销售	行政管理	订货服务	日程安排	能力计划	材料需求	操作控制
总裁			19		19		11		11 14
财政副总裁					7				14
控制员					7				14
人事部长									
销售副总裁		1 13 19	19		2 7 19	1	1 8		
订货控制经理					12 7	12	8		
电器品销售经理		1 19	19		1 2 7	12			
电子产品销售经理		1 19	19		1 2 7	12			
工程副总裁							9 11		
生产副总裁						6 8 1	1 8 9 1	2 8 9 10 11	
工厂厂长						1 6 9 10 11		9 10 11 14	
生产计划主任						1 6 8 9 11		9 10 11	
设备经理							10 11		
材料控制经理				2		2			14
采购经理				2		2			14
部门律师									
计划主任		19			19		11		11

※ 负主要责任和主要决策者 × 过程的主要参与者 / 某种程度上同过程有关

数据类为不同的系统所共享，进而指出利用数据库技术时保持数据一致性的必要。而且所收集的信息在以后制定实施的优先级别时也是很有用的。

4.5.2 确定管理部门对系统的要求

管理部门对系统的要求是 BSP 设计的出发点。这种要求是通过对 10-20 位高层管理人员进行 2-4 小时面谈来得到的。

面谈的目的有以下几方面：

- 核实已得材料，如有关职责、目标、关键成功因素以及其他一些结论。
- 弄清企业未来的发展方向，信息需求，主要障碍和机会。
- 确定企业存在的问题，并将其与过程、数据类联系。
- 提出解决问题可能的办法和确定潜在的效益。

表 4.9 现有系统/数据类矩阵

数据类	顾客数据	订单	厂商	产品	规格	材料单	成本	零件单	原材料存储	成品存储	雇员信息	销售区域	财政	计划	生产安排表	设备	需求单	机器负荷
顾客订单登记	×	×	×	×			×			×			×					
顾客订单控制	×	×	×	×	×	×	×			×			×		×			
开发票	×	×	×				×						×					
工程控制				×	×	×	×	×	×	×							×	
成品库存管理		×		×				×	×	×			×		×			
材料单系统		×	×	×	×	×		×	×	×			×		×		×	
部件存储			×	×				×	×				×		×			
采购单控制		×	×	×				×					×		×			
生产规程		×	×	×	×	×		×	×	×					×			×
店面控制		×	×	×	×	×			×	×					×			×
能力计划		×	×	×	×	×		×	×	×			×		×	×	×	×
分户总账		×		×					×	×	×			×	×			
开销账目													×					
生产成本计划		×		×				×					×		×	×		
操作记录													×	×		×		
可收账	×	×	×										×					
可付账											×		×				×	
固定资产账													×	×	×	×		
市场分析	×	×		×									×	×				
工资											×							

BSP 方法很重视面谈调查，其规范中对面谈有详细的阐述，它将面谈过程划分为 4 个主要阶段：

- (1) 面谈的一般准备。
- (2) 针对面谈对象的特别准备。
- (3) 进行面谈。
- (4) 总结和分析每次谈话结果。

图 4.6 表达了从面谈中得出信息的重要性以及这些信息与在研究中得出的其他信息的关系。

从图 4.6 可见，在面谈开始前要收集有关企业经营、环境影响、目标、关键成功因素、计划方法、控制方法和企业资源等方面的资料。在面谈以前，研究人员已经通过对环境影响、目标和关键成功因素的考虑，得到了有关整个企业范围内的问题和机会。通过对计划方法、控制方法和企业资源的了解，又识别和描述出企业的过程和数据类。在面谈过程中，通过机构职责和相对于每一过程的信息需求的深入分析，研究人员能够对问题和机会有进一步的理解，建立起问题和过程间的关系。识别出对应于过程的信息需求，并把它们包括在前面定义的数据类中。这时，即可得到图 4.6 中表示的输出部分，即问题所影响的过程、产生这些问题的过程、可能解决的办法、潜在的效益以及解决每一个问题所需要的数据类。

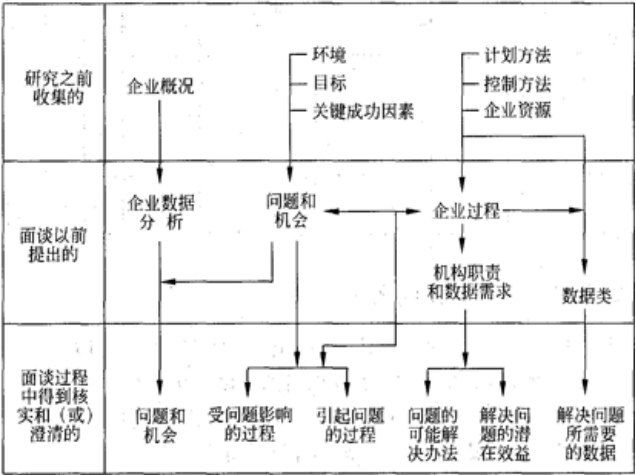


图 4.6 问题和机会分析过程中的信息流

通过机构职责和相对于每一过程的信息需求的深入分析，研究人员能够对问题和机会有进一步的理解，建立起问题和过程间的关系。识别出对应于过程的信息需求，并把它们包括在前面定义的数据类中。这时，即可得到图 4.6 中表示的输出部分，即问题所影响的过程、产生这些问题的过程、可能解决的办法、潜在的效益以及解决每一个问题所需要的数据类。

BSP 建议在进行一般性准备工作中要完成的任务有：明确从面谈中要得出结果，确定面谈时间安排，准备研究组负责人给被访者邀请书，建立记录制度，画出谈话中要用的图表，落实谈话用的办公室，准备一般要提问的问题，落实后勤安排，完成一个试点面谈，计划好可并行的工作。

BSP 还要求及时总结面谈结果，其内容包括下面几方面：

- (1) 回顾面谈，使全体研究人员及时获悉所得到的结果。
- (2) 写出总结，并应用结构化的格式，将打印结果送被访者审阅。
- (3) 分析问题。利用如表 4.10 的问题分析表对问题进行分析。
- (4) 修正图表。在面谈过程中对问题深入了解后，可能需要对已有图表做出修正。例如过程的描述、数据类的描述和过程/数据类矩阵的描述等的修正。

表 4.10 问题分析表

面谈序号	内容	问题	影响过程	潜在效益	产生问题的过程	建议的解决办法	所需数据类
12	计划	对于企业计划没有足够的选择方案	财务	增加资本 减少浪费 改善利润增长	管理	建议财务模型具有“what if”能力	计划
12	管理控制	无法根据产品度量出利润组成	企业计划	提高生产率 增加利润 10%	会计	分产品确定成本分配	成本
12	作业	订单处理中的问题，如批量小、次数多、辅助时间多	订货服务	更有效的成本控制，减少办公人员的辅助工作 50%	销售作业	对订货趋势的分析	销售区域
12	机构	无法识别和提高优秀的人才	人才资源管理	保持人才 提高事业心	人才资源管理	记录雇员的技能和经验	雇员
12	系统	由于过多的生产线停产导致部件库存记录不好	作业	减少停产 改善生产线上的工作条件	库存控制	建立流动库存管理，在库存控制程序中加入编辑子程序	零部件管理过程工作记录

4.5.3 提出判断和结论

收集情况的工作已基本结束，现在的任务是要对得到的事实加以罗列、分析并得出必要的结论，形成报告，在报告中提出判断和结论。

总结目的有如下几方面：

• **与管理人员进行交流。**表明他们在面谈过程中提出的问题已经被理解和接受，并成为整个企业分析的一部分。

- 为提出实施计划提供依据。
- 为建立总体结构优先次序提供依据。
- 为信息结构中的子系统描述提供基础材料。

提出判断和结论的依据来自两个方面：一是在研究初期收集到的材料，特别是用于面谈的图表；另一些就是面谈提供的两个结果，即由被访者认可的面谈总结和列入问题分析表中的主要问题。

下面介绍提出判断和结论的步骤。

(1) 检查前期工作完成情况。

检查所有面谈总结确已完成；确定每次面谈的问题的分析表完整并可用；过程和数据类得到足够的补充和修改。

(2) 确定判断和结论的范畴。

BSP 研究需要有研究范畴和任务之间的一致性，这样将提高工作效率和减少混淆。以下给出研究中每一范畴的内涵。

①目标。

- 对整个企业和每个主要职能部门的目标是否有适当的定义？
- 总目标和部门目标是否一致，能为信息系统规划指示方向否？
- 每一项目标和信息系统间存在何种联系？
- 是否存在支持全量化目标的信息？
- 有没有信息来度量目标达到与否？

②机构。

- 对管理原则是否充分理解？
- 职能和责任是否明确？
- 是否明确指出了正在进行中和计划进行中的机构改革，以及是否了解它对信息系统的影响？
- 职能部门对新信息系统的责任是否容易确定？
- 组织机构关于信息资源管理机构的方针与企业的方针是否相容？

③计划。

- 正规计划的程度如何？
- 长期、短期计划与运营计划的关系怎么样？
- 企业计划是否为信息系统的计划提供了适当的基础？
- 对信息系统的依赖是否在职能计划中得到解释？
- 计划工作是否用上了计算机？
- 计划过程本身适宜于自动化吗？
- 有多少，什么类型的应变计划要做？
- 使用计算机数学模型没有？

④度量和控制。

- 现在用于控制企业的度量方法是否有效？
- 还有哪些度量方法？
- 有什么样的度量和控制关键成功因素？
- 对有效地控制关键企业部门还需要哪些其他数据？
- 为确定目标的完成要做哪些度量和控制？
- 如何将预算用于度量和控制？

⑤运营。

- 在执行操作过程(销售、制造、分配等)中遇到了哪些主要困难？
- 发现了什么有关生产率低、收入低、生产周期长、成本高、生产不能按期完成方面的问题？

⑥现行信息系统支持。

- 目前不能满足的主要信息需求是什么？
- 已有的和用户得到的信息或数据和一般状态是什么？
- 现有的应用系统条件是怎样？
- 环境的影响如何？
- 现有信息系统结构的适应性如何？
- 现有信息系统的设计怎样？
- 现有信息系统的操作有效性如何？
- 有什么样的信息系统计划？

(3)根据以上范畴将问题分类。

将每个问题根据以上范畴分类，把各类问题列入问题分析表中，当出现某些问题可能同属于多种范畴时，可将问题列在一主要范畴中，而把其他范畴记在下边，这时，这个问题可能用来支持多个判断和结论。

(4)将判断和结论写成报告。

可按不同范畴来将问题逻辑地归类，亦可用类似的方法来划分面谈结果。

对于每个逻辑相关的问题组可写出总的结论，而利用这一组内的具体问题和访问材料，形成判断和结论的详细论述。写结论时，应引导出某些建议，但它并不是结论的一部分。对不能形成结论的问题，或者放弃或者只一般性地提及。

对这些判断和结论，应该提出一种框架，同时要求高层管理部门参与对信息处理工作的计划和度量。

(5)将问题分类以确定总体结构优先次序。

如果按产生或引起问题的过程来将问题分类，则可在问题和信息结构的子系统之间提供一种直接联系。因为子系统的设立是为了支持一定的过程，能通过问题分析表确定出解决这些问题的价值，而且知道什么子系统有助于解决什么样的问题，这样就能把建立结构优先级的工作做得更好。

4. 6 定义系统总体结构

4. 6. 1 企业的信息结构图

当企业过程和数据类确定后，应研究如何组织管理这些数据，即将已经识别的数据类按逻辑关系组织数据库，从而形成管理信息来支持企业过程。

为识别要开发的信息系统及其子系统，可用表达数据对系统所支持的过程之间的关系图来定义信息结构。结构图勾画出每一系统的范围，产生、控制和使用的数据，系统与系统的关系，对给定过程的支持，以及子系统间的数据共享。信息结构图是企业长期数据资源规划的图形表示，是现在和将来信息系统开发和运行的蓝图。

4. 6. 2 确定主要系统

为了将复杂的大信息系统分解成便于理解和实现的部分，一般将信息系统分解为若干个相对独立而又相互联系的分系统，即信息系统

的主要系统。通过将过程和由它们产生的数据类分组、归并，进而形成主要系统。其做法是从过程/数据类矩阵入手，并注意到过程是按生命周期顺序排列的。表 4. 7 中，从“计划”开始，字母“C”排列在对角线上，必要时可能移动某些行和列，得到 C 符号的适当排列，从而将企业过程和数据类依据其管理和资源而划分成若干组，并用方框框起来。这些方框表示若干逻辑子系统的组合以及表明维护某些特定的、相关的数据类的责任，如表 4. 11 所示。

4. 6. 3 数据流向表示

落在方框以外的那些 U 表示对数据流的应用，用箭头表示数据从一个系统流向另一个系统。如表 4. 12

表 4. 11 过程数据类组合

数据类 \ 过程	计划	财政	产品	部件目录	材料单	供应商	原材料库	成品库存	设备	在进行的工作	机器负荷	待购材料	工序	顾客	贸易地区	订货单	价格	职工
企业计划	C	U	U						U					U			U	U
机构分析	U																	
评论及控制	U	U																
财政计划	C	U							U									U
资本获取		C													U			
研究			U												U			
预测	U		U											U				
设计开发			C	C	U									U				
产品说明书维护			U	C	C	U											U	
采购						C											U	
接收						U	U											
库存控制							C	C	U									
工序设计			U						C				U					
调度			U			U			U	C	U					U		
容量计划						U			U		C	U	U					
材料需求			U			U					C					U		
操作									U	U	U	C						
贸易地区管理			U											C		U		
销售			U											U	C	U		
贸易管理															U	U		
订货服务			U											U		C		
装运			U													U		
总会计		U												U			U	U
价格计划																U	C	
预算会计	U	U							U								U	U
人事计划		U																C
招聘/发展																		U
补偿		U																U

表 4.12 数据流图

数据类	计划	财政	产品	部件目录	材料单	供应商	原料库存	成品库存	设备	在进行的 工作	机器负荷	待购材料	工序	顾客	贸易地区	订货单	价格	职工
过程																		
企业计划	C	U	U						U					U			U	U
机构分析	U																	
评论及控制	U	U																
财政计划	C	U							U									U
资本获取			C															
研究			U													U		
预测	U		U											U	U			
设计开发			C	C	U									U				
产品说明书维护			U	C	C	U												
采购						C												U
接收						U	U											
库存控制							C	C		U								
工序设计			U						C				U					
调度			U			U			U	C	U					U		
容量计划						U			U		C	U	U					
材料需求			U		U	U						C				U		
操作										U	U	U	C					
贸易地区管理			U											C		U		
销售			U											U	C	U		
贸易管理															U	U		
订货服务			U											U		C		
装运			U					U								U		
总会计		U				U								U			U	U
价格计划						U										U	C	
预算会计	U	U							U								U	U
人事计划		U																C
招聘/发展																		U
补偿		U																U

所示,第3个系统使用了由第2个系统产生的数据“产品”和“原材料单”,第2个系统使用了由第4个系统产生的数据“顾客”和“销售地区”等,并用箭头表示数据流。图中,用一个箭头来表示由一个系统产生和被另一个系统使用的所有数据流。用方框和箭头表示数据的产生和使用后,可以去掉C和U,并对每个分系统命名。表4.13表示了完整的信息结构,最后还要重新安排坐标轴和使用双向箭头,使对构图得以简化,表4.14表示了简化了的结构图。

4.6.4 识别子系统

为了方便信息结构的实现,还必须将分系统划分为更好的子系统。其划分的必要性有:第一,分系统间的相互关系仍非常复杂,每个分系统都可能需要由其他系统产生的信息,分解后能使关系明确、简单;第二,并不是在一个分系统中的所有过程都有需要给与高优先级的支持,第三,给定的系统往往太大,难以一次同时实现,它可能是逐个子系统或几个子系统来实现的。

BSP 给出子系统的以下有关概念:

- 过程提供了合理的子系统边界,因为过程是按企业活动的逻辑关系来划分的。
- 子系统通常仅由一个过程组成,但对其他过程提供支持。
- 一个过程可以由两个或多个子系统来支持。
- 已有的应用系统不应对新系统的规模和边界产生影响。

根据其对数据类的产生和使用特点可将子系统分类如下:

- 产生数据类但不使用其他数据类的子系统。为了确定这类子系统(参照表4.11),把每个C考虑为一个子系统。这类子系统具有独立性。
- 使用其他数据类来产生一个数据类的子系统。产生这些数据类的子系统,要使用其他的数据类。因此所有非第1类所属的C均属第2类。
- 使用数据类但不产生数据类的子系统。它一般是支持度量和控制过程的子系统。从表4.12看,方框中在行上没有C和U就是这类子系统,这些就是只使用数据类而不产生数据类的子系统。

方框以外的那些U,通常不单独构成子系统,但它们是子系统的一部分。

当所有子系统都识别出来以后,应写出每一子系统的功能描述。

4.6.5 先决条件的分析

确定子系统的轮廓以后,则是先决条件的分析,即哪些子系统必须在其他子系统之前开发。通过对企业的理解和已经得到的信息结构,可以分析出子系统间的相互依赖性,从而确定其开发顺序。

表 4.13 信息结构

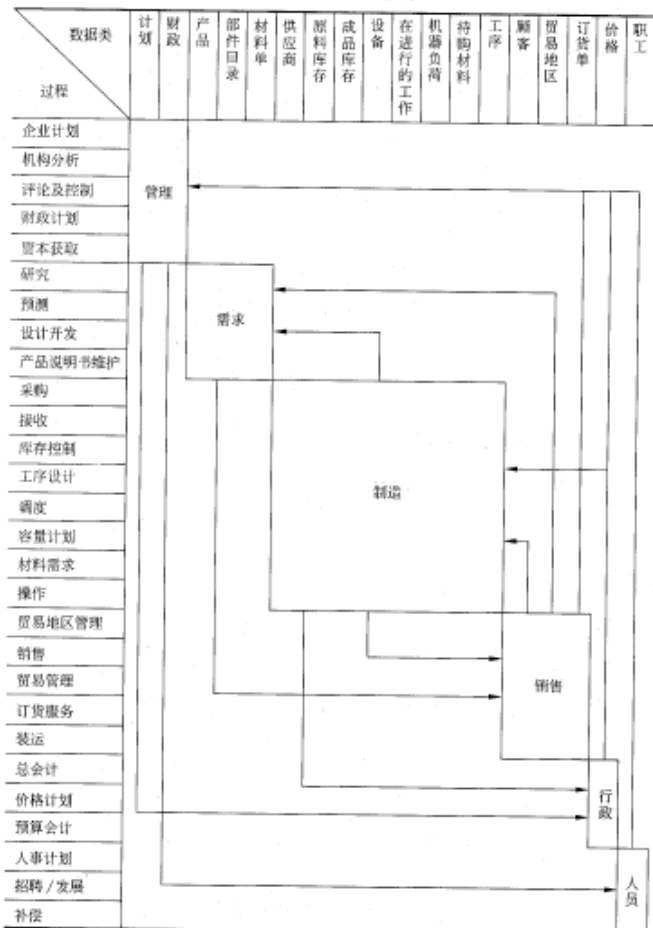


表 4.14 信息结构图重排

4.6.6 信息结构的使用计划

信息结构确定出分系统和子系统，根据它们产生、控制和使用的数据类以及它们支持的企业过程，提供了企业将来信息支持的概貌。子系统的作用是提供了指定优先级别的对象，可以开始对其定义、设计和开发，并且结合企业过程/组织矩阵的使用，信息结构能帮助企业明确产生数据的过程和机构，以确定管理责任和数据政策。

信息结构还能帮助数据管理部门进行有效的数据库逻辑结构设计和对分布式数据处理提供帮助。

4.7 确定系统的优先顺序

4.7.1 确定选择的标准

为了尽早开始实施方案，研究人员应选择出首先要实施的信息结构部分，并向管理部门推荐。

确定子系统优先顺序应考虑下述问题：

- 该子系统是否具有近期投资节省而长期效益好的优点？
- 它会影响哪些方面的人员，有多少人？
- 它是否为初期的数据库结构提供基础性工作？

而确定逻辑优先顺序的主要判断标准可归结成 4 方面：

• **潜在的利益分析。**要确定每个潜在子系统的相对价值，在可计效益、不可计效益和投资回收几方面做出估计。一般 BSP 并不做出投资回收准确计算，而是从面谈中得到有益信息。

• **对企业的影响。**要考虑企业中受影响的组织和人员数目以及定性的影响。如被推荐的子系统将会怎样使企业内的现行状况或问题向好的方面转变，是否有明显的好处，有多少部门，多少雇员将会受到系统变更的影响，以及受到哪种形式的影响。

• **成功的可能性。**应考虑企业接受的程度、实施的可能性、技术的复杂性、实施的先决条件、实施的时间、可能的风险以及可用的资源等。

• **需求。**决策者在考虑投资之前，应清楚地认识来自组织内部的需求、推荐子系统对总目标的支持程度、与其他子系统的关系以及社会的政治意义。

4.7.2 子系统的排序

对构成信息结构的子系统进行分析。BSP 建议可对上面提到的 4 个方面中的每一个做 1-10 个等级的划分，确定实施次序，可绘制图形，以强调最迫切需要的子系统。

4.7.3 优先子系统的描述

最后，应对优先子系统建立详细资料，以便管理人员对其进行评价。一般认为，这些资料应包括企业过程和数据类字典、问题分析表、过程/组织/系统矩阵以及研究判断和结论。它们概述了系统的功能、主要目标和所支持的过程、要解决的问题以及预期效益等。提供的文件应有标准规范。

对优先子系统的基本描述应包括以下几项：

• **一般性描述和目标。**它提供了系统的总体框架结构。包括系统基本功能说明、期望达到的服务级别、系统运行条件或参数、系统包含的应用系统、输出和保留的文件、具体计算方法和必需的控制等。

• **主要问题。**包括从问题分析表中概括出来的有关问题，面谈中管理部门提及并期望解决的问题，研究人员对企业环境进行研究时发现的问题，研究人员对这些问题的判断和结论。

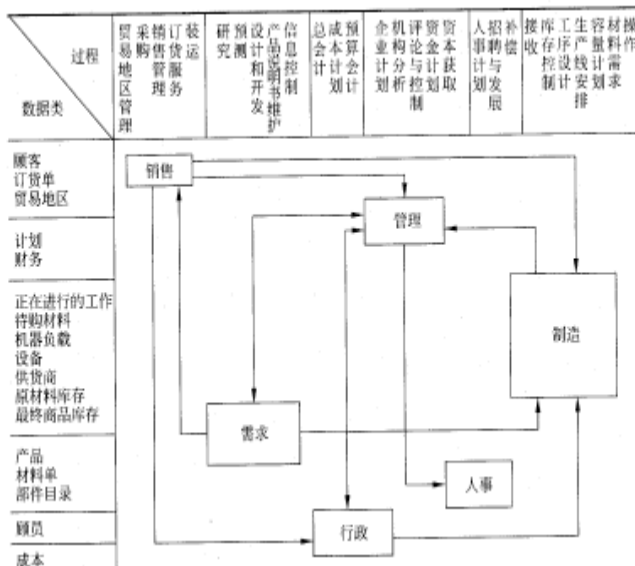
• **潜在的效益。**尽可能地记录下面谈时期管理部门提出的效益，可计的效益最重要，应尽可能详细记录，不可计效益也应列出。

• **受影响的企业过程。**说明系统支持哪些具体过程，对其他企业过程以及其他信息系统的关系，这些可以直接从总体结构中得到。

• **输入和输出。**对系统必要的输入列表说明，以确定使用的数据和信息，它们可以从系统支持的过程的输入数据类查得。对系统期望的输出应做书面描述，包括产生的数据和信息以及报表。它们可以从系统支持的过程及其产生数据得到。

• **影响的组织层次。**利用过程/组织矩阵，查看哪些组织机构要受到这一系统的具体影响和支持，为以后实施提供基础。

• **先决条件。**在信息结构建立过程中，对系统开发的先行项目已有识别，它对制定开发计划有实际意



义。对先决条件的清楚描述，并明确它们与选择好的优先系统的关系，这些都是十分必要的。

4. 7. 4 实施方法的选择

研究人员应对系统的实施提出具体建议。建议是在调查研究的基础上产生的，一般可能是：

- 实行购买的决策，即购买现成的程序，做少许修改，这样最快。
- 实行开发的决策，即系统从头开发，当然也包括项目或系统的移植。
- 开发与购买相结合的决策，即部分开发部分购买，应着重考虑系统间的相容和协调。

实施建议必须明确、具体和详细，以保证执行时不致失误。

4. 8 信息资源管理

BSP 认为应该将数据作为一种资源来进行管理，即信息资源管理 (IRM)，信息资源管理的基本内容包含 3 个主题：

• **资源管理的方向和控制**。要从整个企业管理的角度来分析资源的管理。其指导方针应该是数据可共享、数据处理组织提出应用项目以及资源的有效性。

• **建立企业信息资源指导委员会**。其负责制定方针政策，控制和监督信息资源功能的实施。

• **建立信息资源的组织机构**。其主要功能是从事数据的计划和控制、数据获取以及数据的经营管理，并包括企业应用系统的开发。信息资源组织应由企业的一位副总裁来担任领导，并包括数据处理管理人员和数据管理人员。

4.9 制定建议书和开发计划

通过 BSP 研究而提出的具体建议有或可能有下面 4 方面。

• **信息结构**。包括对目前正在开发的系统所需要的修改，对作为未来方向和未来信息系统规划基础的信息结构的认可，对现行系统的过渡性改进。

• **信息系统管理**。包括加强数据管理以控制组织机构内的数据资源；改进信息系统的规划过程，使得更有效地支持企业和更有效地使用信息资源；提供一个测控系统，以保证未来实施工作能顺利完成。

• **分布信息系统规划**。包括分布信息系统的硬件配置，数据的组织和程序的开发。

• **总体结构优先顺序**。包括提出将被实行的优先级的系统，实行高优先级系统的先行系统的确定。

上述 4 个方面是 BSP 研究的后续活动内容，对于每一方面的建议，都有与之相联系的项目。对于每一项目，又应制定一个开发计划，明确关键的决策和活动。计划应提出费用、潜在效益，并给出项目的详细进度，以便利领导人员做出决策。

每个开发计划应包括下列内容：

- **项目的范围、主题和目标**。
- **预期成果**。给出项目期望的输出或结果。
- **进度**。给出开始和结束的时间。
- **潜在的效益**。它是项目设立的理由。
- **人员和职能**。要求的管理人员和职能。
- **工具和技术**。
- **人员培训**。
- **通信**。协调、联络，确定接口点功能。
- **后勤**。确定材料和设备支持。
- **控制**。确定项目控制方法、责任以及评价或审批的要求。

4. 10 成果报告和后续活动

写出 BSP 研究报告的目的，是为了得到管理部门的支持和参与，并向管理部门介绍研究工作作出的判断，提出建议及通过开发计划。

以下是成果报告一种可选择的形式。

一、引言

背景及综述

系统目标

系统范围

研究组织

二、研究方法

企业及其信息系统的评价

企业过程

企业/信息系统关系

三、主要问题的识别

四、结论及建议

信息结构及优先顺序

信, 息资源管理要求

五、对后续项目的开发计划

描述

可行性

资源要求

效益估计

进度安排

BSP 研究的后续活动是指当 BSP 研究完成后, 进一步开发时应考虑和从事的活动, 它是 BSP 研究主要活动的继续发展。它更偏重于确定细节和做出实现项目的计划。

顺利地开发后续活动将涉及诸多的管理和技术内容, 诸如成立管理委员会、培训即将参与开发的人员、信息系统结构和配置设计、数据库建设和信息资源管理等, 这些都是必须详细研究的内容。

后续活动中重要的步骤之一是第 1 个系统的开发。前边已提到可能的开发途径。有关应用项目的开发, BSP 建议应用开发中心的辅助形式。开发中心提供专用资源和特殊工具, 可加速应用程序的开发, 提高程序的效率。有关后续活动的具体内容可参考 IBM 公司提供的若干资料。

4.11 结论

本章概括地描述了 BSP 方法的基本概念和基本内容。一般认为它适合较大型的信息系统的规划。方法本身是建立企业信息系统的蓝图, 而不是详细设计, 因此在 BSP 研究结束后, 尚存在很多要完成的后续活动。在国内有关 BSP 方法及其变形已有应用, 并取得若干经验, 在技术上有所改进和发展, 如对过程/数据库矩阵可通过计算机按一定算法来自动调整等。

当然, BSP 方法仍有缺陷, 这方面在 James Martin 的著作中有评价。这里把它作为一种方法介绍, 可以在实验中加以灵活运用和改进。

第 5 章 战略数据规划方法

5.1 概述

5.1.1 方法的来源

James Martin 是美国著名的学者, 他的著作甚多, 在世界范围内广泛传播并有较大影响。与信息系统/软件系统开发直接有关的著作就有《信息系统宣言》《战略数据规划方法学》、《数据库环境的管理》和《没有程序员的开发》等。它们结合系统开发实例, 完整、系统地论述了信息系统的开发策略和方法学, 书中列举了较多的国外信息系统开发的成功经验和失败的教训, 由于这些内容对于我国从事信息系统建设的人们都有借鉴价值, 所以有必要加以系统地介绍。

5.1.2 内容概述

James Martin 在其著作前言中的描述, 可以概括地反映本章材料的意图和内容。他说: “在 20 世纪 70 年代, 人们就已经清楚地认识到, 计算机化的信息对企业和其他组织来讲, 都是具有极高价值的资源。人们同时也认识到, 对这个资源的开发, 需要进行总体规划, 这个规划的实施又迫切需要一套形式化的、更易于计算机化的、与数据库设计有关的方法学。”

虽然许多企业早已认识到了对信息资源规划的必要性, 但是很少有人知道如何实现这一规划。一些咨询公司过分强调进行规划的必要性, 但往往又缺少一套过硬的方法来设计所需要的信息资源。本章所描述的方法可用来达到这一目的, 同时还讨论了实际使用这些方法学的实践和经验。

由此可见, 本章较完整地提供了一套进行信息资源开发的方法和策略。

5.1.3 系统开发策略

在 James Martin 著作中, 信息系统开发的战略和策略考虑应贯彻始终,

考虑系统开发战略和策略的根本出发点在于:

- 计算机化的大型企业信息系统的建设是一项企业的重大建设，同时也是一项投资大、开发周期长、具有较高复杂程度的建设项目。
- 计算机化的信息系统不仅是一项技术性的工程，同时也是一项社会性的工程。
- 计算机化的信息系统建设，涉及企业高层管理人员、管理人员、专业技术人员、计算机技术人员和其他用户。
- 计算机化的信息系统建设，涉及管理科学、决策科学、计算机科学和数学等多学科。
- 计算机化的信息系统建设，密切依赖于企业的信息需求、企业环境、企业内部机制、企业人员水平等条件。
- 从长远观点看，计算机化信息系统应该注意和强调投资效益，特别是可见效益、直接经济效益。无经济效益的系统建设难以持久。

由此可见，计算机化信息系统的建设是一项具有技术复杂度和社会复杂度的工程。它的建设必须从实际出发，采用正确的开发策略。

如何保障信息系统开发在企业中有成功的机会，正确的方法论和正确的开发策略是必不可少的。

从普遍原理的角度，必须考虑下列几方面的问题。

1. 企业建立信息系统总体规划的必要性

James Martin 认为，建造一艘战舰，不可能在没有总体设计的情况下，就着手各个零部件的设计和制造。一项完整的信息工程，其复杂程度丝毫不亚于建造一艘战舰。然而，过去由于历史的原因，在绝大多数企业里，信息系统的设计和实施，都是在缺少一个足以能把其各部分配合成一个整体的总体规划的情况下进行的。在没有总体规划的情况下，各子系统独立实施的结果是难以组成协调的大系统；而当需要协调时，则需要对这些子系统加以转换，完成这种转换的代价是昂贵的。不兼容的子系统的存在，将非常难于甚至完全不可能把数据统一、协调起来以满足管理者的需要。好的系统设计，总应避免过分复杂，一个完整的信息系统应该由许多分离的模块组成，每个模块都应该简单到能被有效地设计出来，使设计者能够完全理解，具有较低的维护费用以及高效的开发方法。如果没有一个总体规划作指导，要把这些分散的模块组合起来，构成一个有效的大系统厂，那将是不可能的。因此，设计一个大系统必须要有最高层的规划作为指导，以避免各子系统间的矛盾和冲突，并使用适当的设计工具以协调各项活动。

2. 自顶向下规划与局部设计相结合

建立大型的计算机化的企业信息系统，应注重自顶向下的数据规划和对不同用户领域的系统进行局部设计，两者必须结合，即局部设计是在自顶向下系统规划所建立的框架内进行，而对框架的每一部分，则采用逐步求精的设计方法来完善。

因此，自顶向下的信息系统资源的规划和详细的数据库设计，是建立计算机化的信息系统整套方法的两个重要组成部分，这两个部分应互相兼容并可相互补充地加以运用。首先，按详细程度不同的初始要求，甚至是一个粗略的资源需求的概况，进行自顶向下的规划。要想得到一个实用的系统，还必须在此基础上进行详细设计。

自顶向下规划的主要目标是达到信息一致性，如应保证在数据字段的定义和结构、记录的结构、更新的时间和更新的规划等方面的一致。

3. 高层管理人员的参与

企业信息系统的研究开发工作能否成功，主要取决于管理者对本企业活动的看法以及对信息系统的需求程度，取决于他们能否使研究人员理解企业的业务活动。而且，今后信息系统的多数输入信息都直接或间接地来源于这些管理者。当然，信息系统的多数输出信息也都直接或间接地为这些管理者服务。

如果企业信息系统总体规划的要求只来自数据处理部门自身，这种要求往往是很难实现的。有两个原因：一是，数据处理部门的管理者没有足够的权力来规定统一的数据定义方式；二是，数据处理专家无法对企业的业务活动有充分的理解。这些都要求企业的最高层管理人员参与制定信息资源规划的工作，并领导这一工作的进行。

因此，最高管理者的参与成为系统成功的头等重要因素。信息系统开发中必须有最高层管理人员参与的理由还可列举很多，究其根本，则是企业信息系统的规划和实施，必须能对企业业务活动有深刻的理解，必须有最高管理人员对各部门工作的协调。

4. 处理部门与管理者之间有交流与联系

加强数据处理部门和管理者之间必要的交流和联系是企业信息系统开发的保证。在很多企业里，数据处理部门和管理者之间缺少必要的交流和联系的原因是多方面的。例如数据处理人员经常使用数据处理的行话；数据处理部门没能很好实现自己早期的诺言以及最高管理者未理解要他们参与总体规划的必要性和

迫切性；有时，上级管理者把数据处理人员视为一般下属基层单位的技术人员。

James Martin 认为，如果在数据处理部门和最高管理者之间存在着隔阂，下面的措施将会起到沟通作用：

- 聘请一个专门从事战略数据规划的咨询公司，为企业信息系统规划提出意见。
- 为最高管理者放映录像，介绍信息系统开发的基本问题，详细解释与企业的具体情况密切相关的各个问题。
- 建议最高管理者读一本介绍信息系统建设方面的书。最好有成功的事例介绍。
- 安排最高管理者到有关项目的培训班中学习，James Martin 就开设过高级管理人员短训班。
- 为最高级管理人员举办短期高效的业余讨论班。最好有第一流的专家进行指导。
- 向最高管理者强调，自顶向下的总体数据规划常常可能导致企业机构的调整和重组。

加强数据处理人员和最高层管理者的通信。沟通双方的认识、正确安排数据处理人员在企业中的地位是系统建设中不可少的内容和环节。

5.提高数据处理生产率的途径

数据处理生产率是许多企业关心的问题，随着计算机价格的下降，它变得日趋明显。计算机价格比维持其运行的程序员和分析员的开销要少得多(在国外表现较明显)。因为在程序员和分析员的工资增长的同时，计算机价格却在下降，高级管理者所关心的数据处理生产率，常常是根据开发一个新的急需的应用项目所花费的时间来衡量的。

如何能提高数据处理的生产率？James Martin 指出：许多专家提议，解决数据生产率问题应着眼于结构化程序设计和结构化分析，实质上这样做收效甚微。数据处理生产率低的主要原因是明显的，而且不可能通过对某一过程结构的改变予以纠正。主要包括下列因素：

• **应用的微小变化，可能导致程序的系列变化。**随着系统的扩大，这种连锁反应随之加大，而且这种恶性增长变化所产生的后果有时难以预料；因而数据处理人员一般不愿对系统进行改动。为了解决这个问题，完全需要采用自顶向下的主题数据库规划和熟练的逻辑数据库设计，使用高级的数据库语言。

• **数据格式的不一致、工作文件的不同表示形式，导致数据的共享性差，需要不同的应用程序来适应不同的数据格式并加以维护。**只有采取自顶向下的数据库需求分析，重新规划统一形式的格式，才能应用共同的处理程序。

• **企业的应用程序中存在着许多重复的逻辑结构，而其中有很多功能是相同的，本应由统一的程序来处理。**

• **高级数据库语言比大多数商用程序设计语言有更高的生产率。**在提供适当的数据库工具后，终端用户可以建立自己的应用程序。

• **已存在的适当的数据库系统，对于一定类型的事务，可以缩短和简化系统分析过程；直接应用高效的软件开发工具(如应用第四代语言)可加快开发。**

因此，良好的数据库设计可在上述诸方面为提高数据处理的生产率做出贡献。

6.选择快速收回投资的应用项目

使用高级数据库语言，快速、灵活地建立起能快速收回投资的应用项目，从而使管理者能够感到所投资资金能及时得到报偿。但亦应注意到对管理者所做的各种许诺不要野心勃勃。除了考虑直接经济效益外，决定实施步骤时，还应考虑企业当前最需要解决的迫切问题，亦应优先实施。

7. 数据库费用的支付

James Martin 指出，建立企业的数据库系统，其开发期大约需要 3-4 年的时间，而系统建成后对企业的经营会有很大的效益。但开发一个数据库的经费却是庞大的，对于初次进行这项工作的用户，开发费用之高是他们难以支付的。处理这一问题的正确途径是：数据库管理人员必须拿出一个好的数据库投资规划，并提交各种研究和执行这种开发工作的预算，向最高管理者提交建议书，向他们展示未来系统的总体框架和概貌，做出有关费用和今后两三年中要完成各项重大任务的详细计划。还必须向最高管理者说明，从降低当前和管理信息系统高昂的维护费用和加速应用开发的意义讲，为新的数据库系统付出的费用是合算的。

8. 信息工程

目前一些企业已经建立了相当成功的信息系统，把创建这些成功的信息系统所使用的技术加以总结、提高和形式化，形成了信息工程的主要内容。信息工程的主要焦点是用计算机来存储和维护数据，并从数据提炼出来信息。信息工程是要为企业建立起具有稳定数据型的数据处理中心，并着眼于迅速地满足管理者不断变化的信息需求。James Martin 指出：企业的数据应是相对稳定的，即数据的类型和结构要相对稳

定,而使用数据的处理过程的变化应是频繁和快速的,数据管理人员需要最大的灵活性,以保证数据处理过程能适应管理者快速多变的信息需求。即当必要的数据的最基本结构已经建立时,就可以使用高级数据库语言和应用生成器,很快地建立企业的数据处理过程。

James Martin 给出了企业信息工作的技术模型,图 5.1 是以现代化方法对企业数据处理系统进行建设的一种模式。它的各个技术步骤在 James Martin 的著作中都有论述,而本章只涉及底层的两个模块。

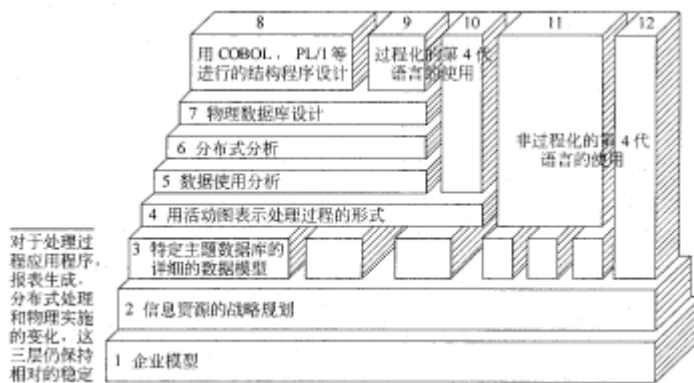


图 5.1 信息工程数据处理管理的现代化方法

5.2 自顶向下规划的组织

5.2.1 规划工作的组织

为了规划工作的顺利开展,应该建立合适的工作班子,其中必须有懂得如何进行规划并有较丰富实际工作经验的成员。当企业没有这类人才时,应考虑从外面咨询机构得到必要的帮助。当然,外来顾问必须是能提供一套成熟的、得到验证的科学方法的专家。聘请外来顾问的好处是,他们不会受本企业过去历史的限制和影响,而能主动地考虑未来的工作方式。当然,把全部的规划工作都由外来顾问处理是不合适的,战略数据规划的领导者应由本企业的人员担当。而他们必须受过严格的培训,以便能自如地掌握规划工作所采用的技术和方法,并在规划过程中,有能力对于初始建立和验收过的规划进行调整和更新。一般称这些领导者为信息资源规划者。

全部规划工作应由核心设计小组来领导,一般它由 4 个人组成,他们将得到企业内各个用户部门的帮助,并从用户部门选取一些主要人员参加到设计小组中。一般称这些部门用户参加者为用户分析员。

James Martin 认为,对一个中等规模的企业,要完成一个自顶向下的规划设计,核心设计小组应包括数据处理管理人员,系统分析领导者,资源管理人员,财务总监,企业的业务经理,客户服务经理等。核心设计小组成员应由外来顾问进行培训指导。

由于设计小组中包括了不同层次、不同业务类型的人员,如既有高级管理者,又有最终用户,因此,必须使用非技术语言来讲解有关技术问题和方法学。这样才能有效地推动整个项目,并使有关技术和方法为所有人员理解。

5.2.2 信息资源规划

自顶向下的全面信息资源规划需要有专门组织来领导,如有的企业设立一个数据或信息资源委员会,负责审查和反馈所需要的信息及规划工作。

在某些情况下,这一组织只负责数据资源规划工作。在另一些情况下,还全面地负责整个信息系统的规划工作。信息资源规划者必须听取高级管理者的意见,而高级管理者必须认可信息规划者所做的规划,这一点必需做到,否则,规划将不能全部实现。

图 5.2 说明了信息资源规划工作中,信息资源规划者自顶向下的规划和数据管理员自底向上进行详细设计工作之间的相互关系。自顶向下的规划者应着眼于全企业,决定企业需要的数据库和其他数据资源。数据管理员则对收集的数据进行分析并综合成所要建立的每个数据库。

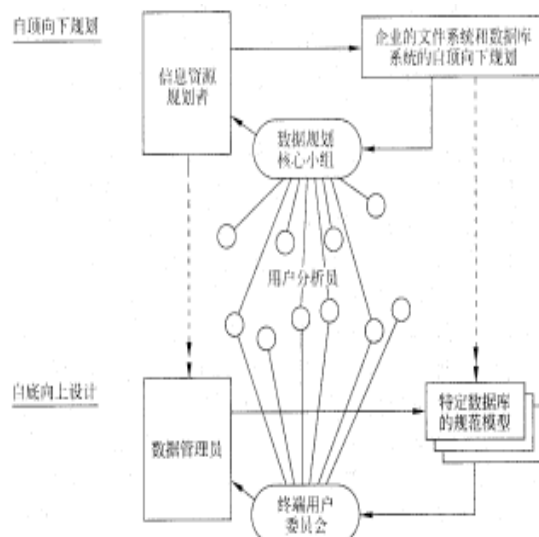


图 5.2 自顶向下规划与自底向上设计

信息资源规划者与数据管理员都需要终端用户的帮助,但是他们所需要的内容不尽相同。信息资源规划者需要各个职能部门用户的帮助,但所有信息和数据不必太详细;数据管理员则需要由终端用户组织的用户委员会在一段时间内对每一个主题数据库进行详细的、精确的审查,要力图使这些主题数据库保持尽可能的稳定。

自顶向下规划与自底向上设计都需要计算机辅助工具,这些工具间应保持一致性和相互支持,它们可用于对彼此设计工作的检查。当需要修改时,两种设计都必须是可修改的。

5.2.3 数据规划的基本步骤

数据规划的步骤可粗略规划如下。

(1) 企业模型的建立。它大致分为三个阶段:

- ①开发一个表示企业各职能范围的模型。
- ②扩展上述模型，使它们表示企业各处理过程。
- ③继续扩展上述模型，使它能表示企业各处理过程。

(2) **确定研究的边界。**当一个企业的各个职能被表示出来后，就必须确定合适的研究范围或边界。James Martin 给出一般原则是：

- **在一个小型企业或密集型的一体化企业中，研究的范围应包括整个企业。**在一个联合企业中，应先在一个公司内进行规划，并把所取得的结果用以指导其他公司的规划工作。在一个复杂的企业，即多部门的企业内，可先在一个部门进行规划，然后推开。

- **假若自顶向下规划的范围太广且涉及到几个独立的单位，那么及时控制和实现数据库的开发是困难的。**假若涉及范围太窄，那么用自顶向下规划工作所得到的好处就会部分丧失。假若信息资源大量涉及及相互交叉作用的几个部门，并且自顶向下规划的工作又被限制在一个部门内，那么这个规划将不可能为建立所需求的信息系统提供一个稳固的基础。

- **战略规划的研究范围与企业的管理方式有联系。**在一些企业中，部门间是相对独立和自治的；在另一些企业中，情况就相反。即使在相同的工业领域，规模相似可在几个不同部门各自独立进行。而采取集团管理方式的企业，自顶向下的全局规划需要同时涉及到所有部门。全局规划的边界划定要适当，不要太大。

(3) **建立业务活动过程。**当以职能为基本单位的企业模型建立后，就应着手建立每个职能范围所包含的业务过程。通过对每个职能范围有代表性活动过程的分析，可以确定各个业务活动过程，它由核心小组先提出业务活动过程组，然后交由各职能范围的业务人员进行审查和精确化，直到把各职能范围内全部的、准确的、一致同意的业务活动过程建立起来。

James Martin 指出，自顶向下的全局规划可以分为粗略的方式和精细的方式。前一种方式一般只描述职能范围和业务活动过程，而不描述活动。它们只描述主题数据库而不去描述组成这些数据库的实体。而通过对一些实例考察，在大多数情况下表明，如果使用计算机化的辅助工具和用户在用户部门的得力帮助下，采用精细规划方法所花费的总时间不会比采用粗略规划方法所用的时间长很多。精细的全局规划能对数据库系统的设计与实施提供充分的指导和帮助。尽管如此，必须指出，自顶向下的全局规划应着眼于全局并尽可能快地完成，而不能陷入对数据模型定义等耗费时间的细节中去。详细数据模型的讨论应在以后的较小范围内讨论。

(4) **实体和活动的确定。**一旦职能范围和业务活动过程的确定取得了一致意见，则应着手进行实体和活动确定，有关实体的数据以及与之有关的活动形成了一个更详细的企业模型，它们由核心小组来汇集。

完成确定实体和活动任务的方法，是培训企业中各个不同职能范围内有兴趣的用户来完成。而在前一阶段确定业务活动过程的系统分析员可以继续同用户一起来指导这些工作的进行。

每个职能范围的工作模式应交给各职能部门的业务人员进行审查。

(5) **对所得规划结果进行审查。**一般在信息系统规划中大量采用直观的图形或图表表示方法。而采用不同的方法学所建立的全局规划图在形式上「是不相同的。不论采用哪种图来表示，最后都必须由各不同职能范围的管理人员和参加分析的用户分析人员进行审查。

每个审查人员可以先查看规划图的全部职能范围，然后对自己所熟悉的职能范围的内容进行深入细致的审查。应该鼓励审查人员对本职能范围以外的其他职能范围内容进行审查，但要求每一个审查者对全局规划图的所有职能都详细了解是不现实的。

对全局规划图的审查和调整的探索性过程是非常必要的，规划图不应过早“冻结”，而且需要充分地精确化，以便正确地、精确地指导数据库设计。由于这些规划图需要经过反复调整，因而应尽量使用计算机作为辅助工具来绘制图形。

James Martin 对规划过程提出了时间要求：自顶向下的全局规划工作应该在 6 个月内完成。按照一般经验，只要有切实可行的规划方法，在 6 个月内 90% 的规划工作都可完成，剩下的 10% 只是一些细微的枝节问题或不确定的问题。这些枝节和剩余问题不应该影响规划工作的执行。为了保证在合适的时限内完成整体规划，而又能保证在实施过程中有充分的依据，因此系统的分解不能过分追求细节，但又不能太粗略，必须适度。

5. 3 企业模型的建立

5.3.1 企业职能范围

企业模型表示了该企业在经营管理中具有的功能。不同的企业模型对企业活动表示的详细程度各异。当数据需求反映到企业模型上时，是该企业模型面向数据的一种变换，可把这个变换分解成需要实现的多

表 5.1 一个企业的职能范围和业务活动过程

职能范围	业务活动过程
业务计划	市场分析, 产品范围评审, 销售预测
资金	财务计划, 资本获得, 资金管理
产品规划	产品设计, 产品定价, 产品规范维护
材料	材料需求, 材料订货, 验收进货, 库存控制, 质量控制
生产计划	生产能力计划, 厂家安排, 工作流程安排
生产	材料控制, 测量和下料, 机器运转
推销	销售区域管理, 销售管理, 客户联系
分销	成品控制, 订货服务, 包装, 分发
财会	债权人债务, 现金流, 工资, 成本核算, 预算计划, 利润分析
人事	人事计划, 人员招聘, 劳保政策

个数据库。

企业职能范围指的是一个企业中的主要业务领域。一个中型企业的职能范围可能有业务计划, 资金, 产品规划, 材料, 生产计划, 生产, 推销, 分销, 财会和人事等。

在信息资源规划中, 第一阶段是确定企业的各个职能范围, 以便能够了解企业整个概貌。

5.3.2 业务活动过程

每个职能范围都要实现一定数量的业务活动过程, 表 5.1 指出了 37 个过程, 绝大多数企业会包含更多的过程。

James Martin 指出, 在一个大型企业中, 可以有大约 30 个职能和 150-300 个可执行的过程。职能范围及其业务活动过程的确定应独立于企业当前的组织机构。应该正确地理解本企业的职能和业务活动过程。它们应该是企业的最基本的决策范围和最基本的活动, 它们应该独立于任何组织机构和管理职责, 以求尽量避免当企业的组织机构被调整或管理层次发生变动时, 信息系统所提供的信息不再适合需要。

业务活动过程可以用简明的文字加以定义, 例如库存管理可以定义成“仓库保管和原材料、零件和部件的收发控制过程和估计库存量的过程”, 这个过程可以由一个独立的部门来完成, 也可以由若干个部门共同完成。不论仓库的分开或合并, 库存管理的业务活动过程总是独立于机构而存在的。

图 5.3 说明在一个生命周期每个阶段的一些业务活动类型。

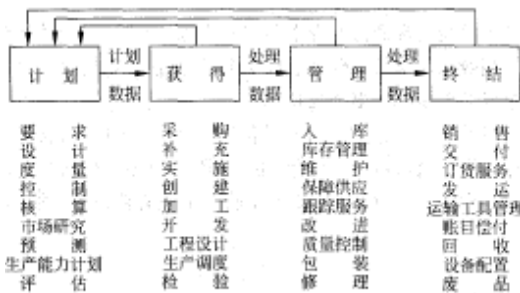


图 5.3 产品、服务及资源的生命周期的 4 个阶段

所确定的各个业务活动过程必然与该机构中各个负责人之间有着对应的关系, 应该建立如表 5.2 所示的图表, 它构成了现行组织的负责人与业务活动的关系图, 有助于帮助了解应访问的对象, 并确认不会有被遗漏的过程。核心设计小组应努力确定该企业或有关部门的所有业务活动过程, 并从这些过程中删除重复过程。

在每个业务活动过程中, 都含有一定数据量的业务活动。如在表 5.1 中有一个“材料订货”的业务活动过程, 它包括如下活动: 提出购货申请、选择供应商、提出购货订单、依据订单、执行交货条款、处理例外情况、记录供应商执行合同情况和分析供应商执行合同情况。

就典型的业务活动而言, 每个过程一般包含 5-30 个活动, 在一个小型企业中可能有几百个活动, 在大型企业中可能有几千个活动。

James Martin 认为, 最好把企业的业务职能范围分解成多个功能, 每个功能又被分解成更低层的功能, 这样逐级向下分解, 直到产生最基本的活动为止, 如图 5.4 所表示的企业模型图就是一个功能逐级分解的过程, 最低层的功能称之为活动, 活动具体执行某种操作。

5.3.3 企业模型图

在一个企业中, 需要一张表明该企业

表 5.2 业务职能范围和业务过程对应于现行组织的结构图

过程	组织	计划	获得	管理	终结
市场	市场调研	×	×	×	×
销售管理	销售预测	×	×	×	×
工程	设计开发	×	×	×	×
生产管理	生产计划	×	×	×	×
材料管理	材料需求	×	×	×	×
设备管理	设备维护	×	×	×	×
资金管理	资金管理	×	×	×	×
人事	人事管理	×	×	×	×
企业管理	企业管理	×	×	×	×

* 主要职能和决策者 / 业务过程的主要参与者

职能和活动的企业模型图，这张图可能很大，因为在一个企业中有几百个活动，有时多于上千个活动。在新系统的研制过程中，这张图表经常被充实和修改，因而应使用计算机来辅助这项工作。图 5.4 给出的是一个企业模型图的示例。这张图可以分解成一些规模更小的图，如企业的一个部门或分解到职能范围。

James Martin 指出，企业模型应具有如下特性：

- **完整性。**模型应提供组成企业的各个职能范围、各种业务活动过程和各种活动的一个完整的图表。

- **适用性。**该模型应是人们合理有效地去理解企业的途径，在分析的每一层上所确定的过程和活动对所涉及的管理工作应是自然和确切的。

- **持久性。**只要企业的目标保持不变，该模型就应该保持正确和有效。只要企业执行的职能相同，企业模型就依然适用。

图 5.4 说明了企业功能的分层分解，当功能分解成一个可执行的单元时，即一个可实现的物理过程、一个特定的计算机过程、一个非计算机化的过程或一个在没有预先指定的终端上进行交互的活动，这种分解即可停止。

企业模型化的过程经常揭示出企业组织机构中的一些冗长的异常情况，对于这些情况，最高管理者往往是不清楚的，因而战略规划工作可能导致整个企业组织的重组，而不仅仅是数据处理部门的重新调整。

建立企业模型的好处是明显的，即使企业信息系统并未付诸实施，亦可随着规划项目的进展，使人们对企业有一个更好的了解，可以克服现行运行系统存在的不足，使得企业的组织变得能够支持尚未实施的职能。

5.3.4 战略业务规划

James Martin 指出，如果可能的话，应该结合数据的战略规划进行必要的业务规划讨论，因为业务规划影响着业务信息需求。要讨论的问题包括：企业未来的变化如何？将会出现与当前哪些活动不同的业务活动？

一些业务规划设计者首先考察一个企业或机构的目的和任务，然后将它们分解成可执行的目标，而目标必须是可度量的。高层次所确定的目标必须分解成部门的目标，这些目标需要清楚地加以阐述，并变成向下级人员下达的明确指示以及保证这些指示完成的行动。

5.3.5 关键成功因素

在大多数企业中，都存在着对该组织成功起关键性作用的因素，一般称为企业经营关键成功因素。通常有 3-6 个决定企业成功与否的因素，为使企业获得成功，这些关键性的任务必须特别认真地完成。

在一个企业的业务活动中，关键成功因素总是与那些能确保企业具有竞争能力的方面相关，在不同类型的业务活动中，关键成功因素也会有很大的不同，即使在同一类型的业务活动中，在不同时间内，其关键成功因素也不同，甚至受外部环境的影响。

下面是 James Martin 给出的各类企业的关键成功因素的示例部分。

- 汽车工业：燃料的节约，汽车的式样，高效供货组织，生产成本的严格控制。
- 软件公司：产品的革新，销售和用户资料的质量，国际市场和服务，产品的易用性。

关键成功因素与企业阐述的目的和使命通常是不同的，后者表示企业的远景规划或最终期望目标，而关键成功因素则与当前业务的处理以及所需要的高性能的关键领域有关。这些关键成功因素提供了最高管理者掌握的控制系统中所需要的测量标准。一个企业要获得成功，就需要对关键的成功因素进行认真的和不断的度量，并且时刻注意对这些因素的调整。

由于企业关键成功因素的变化，一个企业应定期地由该企业的最高管理人员开会对其进行审查。

一般认为，确定关键成功因素所需要的数据往往需要进行特殊采集，有的来自系统以外。很多关键成功因素所需要的数据是从多种逻辑文件中获取的，它们可能分布很广，少量的关键成功因素需要主观的估计，而不是以简单的数量形式确定。

如何度量关键成功因素，在讨论中往往出现许多不同的度量方法，例如某个企业度量方案是根据所鉴定合同的盈利率，当然还有其他一些影响因素，如积极推销等。绝大多数的软度量的方法是由人与人之间的

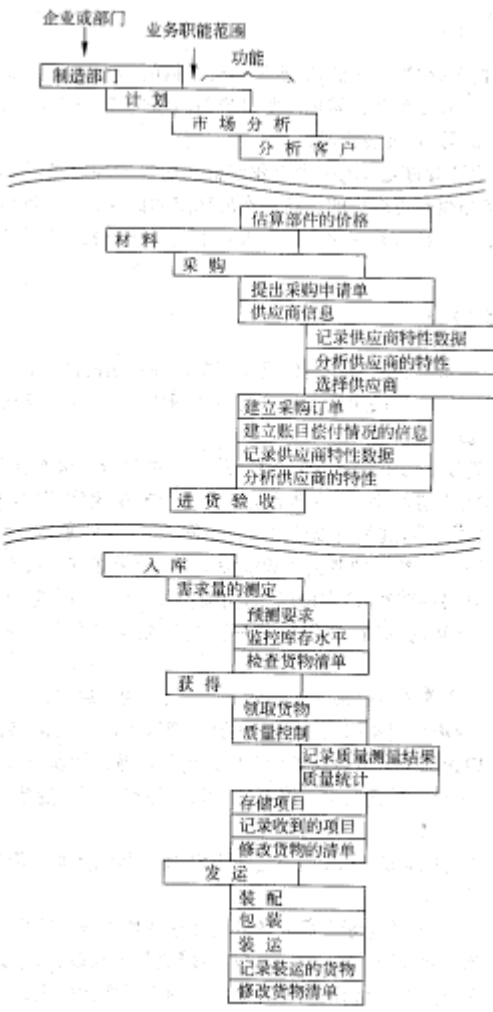


图 5.4 一个企业模型图

讨论完成的。有时几种不同度量方法用于同一因素，如有时一定要安排企业的某位最高层管理人员与客户会谈，这是一个重要的关键成功因素。

在一个信息系统的建立过程中，一开始可以让一个数据处理的负责人或设计者与总经理会谈，确定出他所认为的关键成功因素。然后，要找出测量这些因素的方法。有一些因素只能用软测量的方法，而通常总存在些数量化的测量方法。其次，必须报告这些讨论结果。

关键成功因素分析报告应作为战略数据规划的部分内容加以认真考虑。

5.4 主题数据库及其组合

5.4.1 主题数据库的概念

James Martin 的战略数据规划的重要内容之一就是企业主题数据库的确定。回顾数据库的应用发展史，有两类数据库，即应用数据库和主题数据库。主题数据库与企业经营主题有关而不是与一般的应用项目有关。例如，对于一个工厂来说，其主题数据库应是一个产品数据库，而不是与产品有关的各个独立的存货、订货、进货和质量控制等应用数据库。许多应用项目可以共同使用同一个主题数据库。当然，有些应用项目也可以使用多个主题数据库。

自顶向下的全局规划的目标之一，应是确定所需要的主题数据库。在一个企业中，可建立数据库的典型主题可能有产品、客户、零件、顾客、订货、账目、人事、档案、工程规范。

主题数据库的设计目的是为了加速应用项目的开发。程序人员使用的数据应已存在于有关主题数据库中。它把企业的全部数据划分成一些可管理的单位——主题数据库。主题数据库应设计得尽可能稳定，能在较长时间内为企业的信息资源提供稳定的服务。稳定并非限制主题数据库永不发生变化，而是要求在变化后不会影响已有的应用项目的工作。主题数据库的逻辑结构应独立于当前的计算机硬件和软件的物理实现过程，这样能保持在技术不断进步的情况下，主题数据库的逻辑结构仍然有效。

5.4.2 主题数据库的选择

在大多数的情况下，主题数据库内容的选取和确定还没有一套形式化的方法。在实施过程中往往对于一个主题数据库等问题存在着许多争论，James Martin 推荐两种方法来选择和确定主题数据库。

首先，列出企业所涉及的产品和机构的组成内容，如产品、设备、原材料、建筑物、零部件、现金、供应商、账目、客户、股东等。对每一项都可以有基本记录、特殊记录、事务处理、摘要或统计、计划或设计数据。这些数据类型可以写到相应的项目中，这些数据类型也可以写到由图 5.3 所示的生命周期序列中，例如对于原材料有原材料计划、单据、费用、购货单据、存货清单、回收和用途统计。

若每一种基本产品或机构都划分好生命周期，则所有的数据就被归并成一些相关的数据类。因此有时主题数据库又称为数据类(BSP 用语)。

其次，可以考察应用表 5.1 所示的业务活动过程，然后记录下每一个过程的输入和输出数据属于哪个数据类，这样得到一个数据分类表。

可以将前后两种途径得到的数据分类表相互对照来建立一个联合的数据分类表，从而形成主题数据库的基础。

为了规划企业的主题数据库，对该企业有一个高层次的全面分析是必要的。一个企业的总体规划不仅要考虑主题数据库，而且要考虑已存在或新建立的文件以及为特定的应用项目所单独建立的数据库。高层次的分析是战略规划的重要组成部分。

James Martin 给出了银行的数据库规划示例，如图 5.5 包含 21 个主题数据库，它可以支持一个银行中大部分业务活动过程。应该画出这些数据库与主要的银行活动对照的映象，作为数据库开发的主要规划内容。这方面的材料可以参考 James Martin 的著作。

5.4.3 主题数据库的组合

James Martin 指出，主题数据库与 BSP 方法中的数据类是相当的概念。当给出许多主题数据库及业务活动过程后，在实现企业信息系统时，必须把这些主题数据库组合或划分成若干可以实现的子系统。从方法论的角度 James Martin 引用了 BSP 方法中子系统划分的过程来实现将主题数据库到子系统的组合。有关这方面的内容已在本书的第 4 章中介绍，亦可参阅 James Martin 的原著。

5.4.4 4 类数据环境

James Martin 清晰地区分了计算机的 4 类数据环境，并指出，一个高效率和高效能的企业应该基本上具有 3 类或 4 类数据环境作为基础。下面列举 4 类数据环境。



图 5.5 银行业务的主题数据库规划示例

1.文件环境

不使用数据库管理系统。当建立一个应用项目时，由系统分析员或程序员来设计一些独立的数据文件。对于大多数应用项目，都使用这类独立文件。

特点如下：

- 简单，实现起来相对地容易。
- 随着有高冗余度的大量文件激增，这类环境将导致维护的成本提高。
- 对应用项目表面上的微小改变，都可能引起一系列的其他改变和反应，这就使得改变迟缓、困难和昂贵。

2.应用数据库环境

使用数据库管理系统，其数据共享程度高于文件环境但低于主题数据库环境。各独立的数据库是为各独立的应用项目而设计的。

特点如下：

- 较主题数据库环境容易实现。
- 如同文件环境一样，随着有冗余的数据库的大量激增，维护成本有时比文件还高。

当然也未达到数据库操作的主要优点。

3.主题数据库环境

数据库的建立基本上独立于具体应用，数据的设计和存储独立于它们的应用功能。有关业务主题的数据间的联系，由共享数据库来表示。

特点如下：

- 需要详尽的数据分析和模式化，具有较低的维护成本。
- 这将逐步地导致应用开发效率的提高和用户同数据库的直接的交互式对话。
- 需要改变传统的系统分析方法和全部数据处理方式；
- 如果管理不善，会退化成前两类环境。

4.信息检索系统环境

这一类是为自动信息检索、决策支持系统和办公室自动化而设计的，而不是为专用的计算和大量生产性运行的数据而设计的。新的数据项可以动态地加入到数据库中，软件是围绕着倒排表和其他的数据检索技术设计的，提供了良好的终端用户语言，使用这些语言能灵活地创建自己的逻辑数据文件。

特点如下：

- 较传统的数据库系统更灵活，并能动态地进行变化。
- 通常与主题数据库环境共存。

James Martin 指出，把信息检索系统从生产性的数据系统中分离出来的主要原因是考虑效率，信息检索系统需要把它的数据库按照不同于大容量的生产性系统中的数据组织方式进行组织，它通常仅包含一个数据子集。当一个信息系统包含而且必须去检索存放在一个日常生产性系统中为数庞大的一片数据时，这个信息系统的效率可能很低。另一方面，日常的处理工作也将会被很多由终端用户输入的有关查询操作所干扰。

在讨论数据库管理过程中，区分后两类环境是十分重要的，两者具有不同的问题，有不同的处理方法，但两者都必须符合于企业的数据资源规划。

随着互联网技术的发展和与数据库技术的结合，数据环境也在发生着变化和更新，这也是在系统建设中所出现的新问题，他将引起系统结构的变化，使用工具的变化等。

5.5 战略数据规划的执行过程

5.5.1 企业的实体分析

自顶向下规划的一个主要目标是消除数据冗余和不一致，因为这些冗余和不一致数据导致应用程序和维护的重复，造成数据处理资源的极大浪费。但从以上分析所提供的主题数据库或 BSP 方法的数据类出发来实现上述目标又显得过于粗糙和缺乏上下的有机联系，从而造成实施上的困难。本节介绍 James Martin 给出的一种更为精细的自顶向下的、建立在企业实体层上的补充方法，它要求分析和绘制企业的实体图，从而能详细说明一个主题数据库包含的内容，以及对使用实体的系统确定边界。

自顶向下的规划可以进行几层求精，图 5.6 表示 3 层求精的情况。前面已描述主题数据库层，本节将描述实体层，5.5.2 节将描述实体活动层。

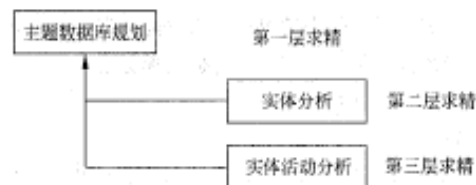


图 5.6 有关对自顶向下规划的 3 层求精

1.企业的实体

实体是数据的载体。实体分析是自顶向下确定企业实体的过程。实体可以是具体的，也可以是抽象的，如顾客、部件或产品规格、财政预算等。

一个典型的中等规模的企业有几百个实体，一个大型但业务单一的企业实体数量并不会比中型企业多许多倍，而多种业务的企业实体数会更多。一般实体集合并不因时间推移而有大变化，除非企业改变经营业务。

用记录表示实体的属性，一个大公司常有数千种类型的记录。如果自顶向下分析做得好，会减少大量的实体重复，重复导致冗余的应用程序和增加维护的难度。

2.实体的确定

在企业中实体常由了解业务的用户分析员来确宁，当然仗些用户分析员必须接受识别实体的训练。

不同的用户分析员常用不同的名字命名同一实体，因此往往用建立一个实体同义词字典来对实体作出一致性的约定，从而消除重复的实体。

经验证明，实体分析过程需要非数据处理的高级管理人员参加，实体分析结果质量和参加的高级管理人员的素质密切相关。



需要高级管理人员参加实体分析的一个重要原因是为了控制实体选择的能力，这些实体应适用于整个信息系统。

3.实体间的联系

下面给出实体和实体间联系的几种表示。用方框表示实体，用方框间的连线和其他辅助符号表示它们之间的关系。

一对一的联系，用一条单箭头线表示，如  是指实体 A 的一个值，实体 B 有且仅有一个值与之对应。

一对多的联系，用一条双箭头线表示，如  是指实体 A 的一个值，实体 B 有多个值与之对应。

用同一条表示 A, B 间的相互联系，如  是指实体 A 的一个值有实体 B 的多个值与之对应，而实体 B 的一个值只对应实体 A 的一个值，如  表示一个分店有多个推销员，而一个推销员只属于一个分店。


在线上划一个圈表示有一个实体值或没有实体值与之相对应，如  是指实体 D 有一个值或没有值与实体 C 相对应。

图 5.7 描述了一个局部范围内的实体联系图，其符号的表示遵从上面的约定。其含义是说明产品/材料

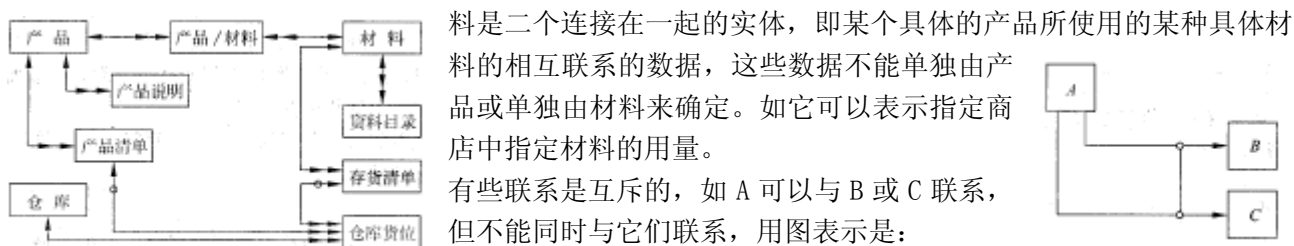
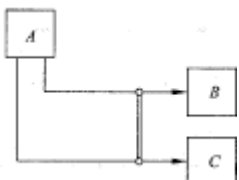


图 5.7 一个局部范围内的实体图

有些联系是互相伴随的，如 A 与 B 有联系，A 也必然与 C 有联系，用图表示是有一条双竖线连接两个联系：



4.实体图和数据模型

如图 5.7 形式的图称为实体图，它是对企业实体的概括。数据模型是对企业更精确的描述，它是对终端用户观点的一个全面的综合。得到完全的数据模型比确定企业实体要花费更多的时间，因而保持最高层管理者对它的兴趣很重要。在做实体分析到对数据模型的建立过程中，一定要反映企业真正的信息需求。

James Martin 指出，为了进行数据库设计，普通的快速而艰苦的方法是必须确定作为数据载体的实体，并且记下与实体相关的属性。在某些情况下，每个属性表都要转换成第三范式。这种方法适合于理论讨论的简单环境，而对于现实生活中的一个企业或政府机构的复杂情况，这种方法有时难以给出满意的结果。良好的数据库设计既需要进行实体分析，又需要进行数据模型化，两种方法起着相互促进的作用。

5.自顶向下的规划和自底向上的设计

自顶向下的规划有别于自底向上的设计，自顶向下的规划是本章所描述的，也是 James Martin 在其“战略数据规划方法学”一书中描述的一个完整过程，这个过程由图 5.8 的左侧表示。自底向上的设计是数据模型的细化过程，该过程导致了物理数据库设计和子程序的建立，这个进程由图 5.8 的右侧表示。

自顶向下的规划和自底向上的设计不是独立的和相反的过程，自底向上设计是自顶向下设计的延伸。数据模型是实体图的延伸和详尽，而更详细的阶段所产生的反馈信息有时又会引起对自顶向下全局观点的调整。

James Martin 认为，不同的方法论因规划的详尽程度要求会不同，但规划应该是完善的，当然不必太详细以免阻碍数据处理的开发。

建立实体图应很快完成，实体的确定无需非常精确，实体的关键字和复合的关键字无需最后确定，这些内容在做数据模型时，详细的数据模型又会导致实体的改变。

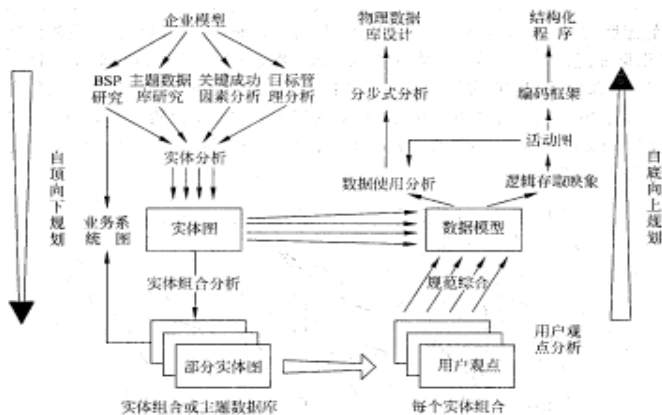


图 5.8 自顶向下规划和自底向上设计

6.结构化实体图

手工绘制实体图只有当实体数在 20-30 个时才有可能，而通常实体数都大大多于这一数目，因此有必要自动绘制，即用计算机辅助绘制实体图。James Martin 指出，一张实体图必须用一种更为清晰的结构化方法画出，应该使用计算机来绘制和维护。

用计算机绘制和维护实体图的过程如下：

首先，清除任何冗余的联系。在实体图中一些实体是根实体，一个根实体是实体图中不发出单箭头的，我们把根实体定义为深度为 1 的实体。

深度为 2 的实体是，它有一个单箭头指向一个深度为 1 的实体。深度为 3 的实体是，它有一个单箭头指向一个深度为 2 的实体。但没有箭头指向一个深度为 1 的实体。

深度为 N 的实体(N>1)是，它有一个单箭头指向深度为 N-1 的实体，但没有箭头指向更低深度的实体。图 5. 9 有 16 个实体，其结点处的数字表示每个结点的深度。

如果把深度为 1 的实体画在实体图的左边，深度为 2 的实体画在向左偏移一段距离的分支位置，深度为 N 的实体画在向右偏移 N-1 段距离的分支位置，并且深度为 N(N>1)的实体是在它指向的深度为 N-1 的实体下方分支位置上。在每一个深度为 1 的实体下面的实体形成一个簇，连接这些簇的箭头画在实体图的左边。图 5. 10 是根据上述方法重新绘制图 5. 9。

有时，一个非根实体会有选择双亲的问题，如图 5. 9 中实体 A 是深度为 2 的实体，它可以连接深度为 1 的任一实体 B, G 和 L 实体。哪一个实体是双亲的最佳选择？应是最强的或最常使用的联系。如假定 $A \longleftrightarrow B$ 比 $A \longleftrightarrow G$ 或 $A \longleftrightarrow L$ 联系更强，则 A 画在 B 下面。类似地，如 $J \longleftrightarrow H$ 比 $K \longleftrightarrow H$ 更强或更经常使用，则 H 画在 J 下面。

对于循环联系，为画实体图，最弱的联系要暂时分开，图 5. 10 中认为 $C \longleftrightarrow N$ 是最弱的联系。

如果联系的强度是未知的，或是具有相同的强度，为了生成一个能够重新绘制的实体图，则可做任意的选择。

7.把实体聚集成超级组

图 5. 10 中由层次簇组成了 4 个实体组，实体又以实体间的联系路径的使用频度为依据聚集成超级组或称实体超级组。它是实体的集合，这些实体实现在一个主题数据库中。一个超级组内的联系路径应有较高的使用频度。不同的超级组之间的联系路径的使用频度是较低的。为了确定实体间的路径，可以像执行实体分析那样，用下述分类进行标记。

实体图联系强度的五种划分如下：

- 非常弱的联系，必然在不同的超级组，用 1 表示。
- 较弱或不常用的联系，必然在不同的超级组，用 2 表示。
- 平均的或无强度判断，用 3 表示。

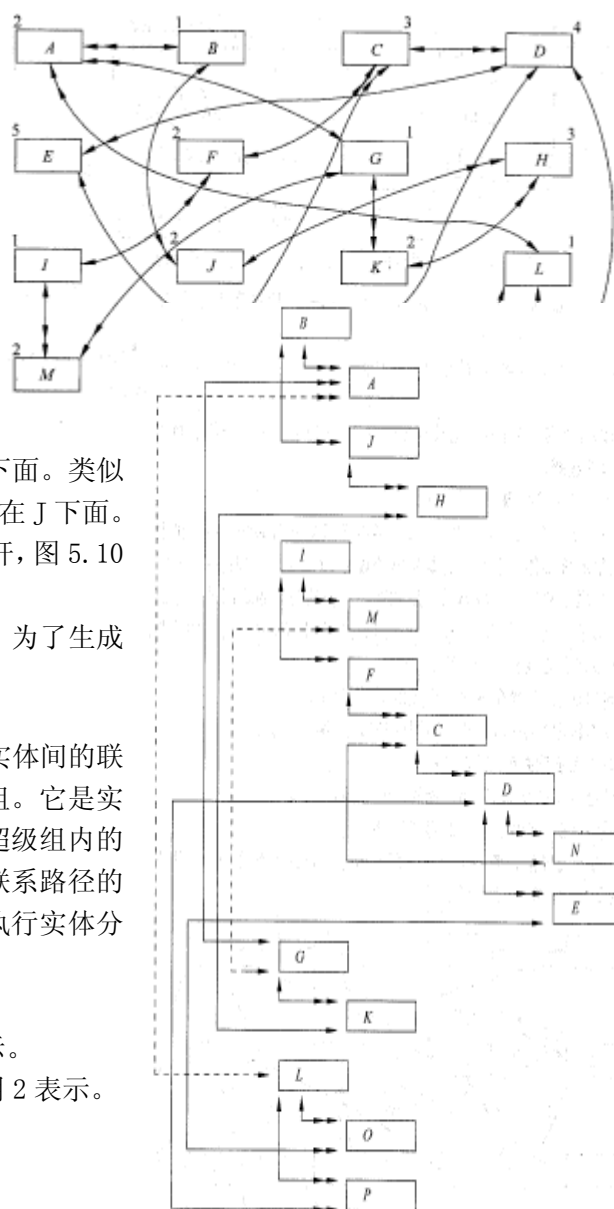


图 5.10 按照结构化的方式重新绘制图 5.9

- 较强或经常用的联系，可能在相同的超级组，用 4 表示。
- 非常强的联系，必然在相同的超级组，用 5 表示。

这种简单的分类可作为把一个难于掌握的实体图划分成可管理的实体图或将实体聚集成超级组的基础，如图 5.11 所示。

图 5.12 表示了联系分类基础上图 5.10 被划分成实体超级组。用虚线表示是第一类联系，表明它们不应在相同的主题数据库中，图 5.10 中位于中间的一个实体组的从 C 到 F 的联系，就属于第二类联系，这就使得它所在的实体组分解成两个独立的实体组。图 5.10 中所有实体组位组成三个超级组，如图 5.12。

这些联系类可以用来决定一个实体属于哪一个实体组，在何处。如图 5.12 中实体 A 画在 B 下，而不是在 G 或 L 下。类似地，在循环图中 C、D 和 N 之间的弱联系是 $C \longleftrightarrow N$ 。在一组内深度相同的实体项中，位置在上的实体表示它与双亲关系最密切，即它和双亲的联系较强，因此，图 5.12 中 A 在 J 上。

James Martin 指出，在自顶向下的阶段，可以依据类别划分实体图成超级组。但在一次处理的基础上还不能把一个实体图完全划归成主题数据库，上面给出的联系类可用于表达有关实体组成可实现的数据库的直观要求，为数据库的详细设计提供依据。

有关实体分析的更详细的例子和细节，可见 James Martin 原著。

5.5.2 实体活动分析

上面讲述的数据规划使用的是实体而不是活动，现在讨论实体以及与之对应的活动，从而对企业使用的数据库给出一个精细的分解图，并导出一种更正规地把实体聚集成主题数据库方法。图 5.4 是以活动为基础的企业模型，在粗略分解的规划中，可以得到对应于职能范围和业务活动过程的主题数据库的轮廓，在精细分解的规划中，相应的职能范围被分解成活动以及与之对应的实体。

1. 企业功能的分解

一个功能是企业要完成的一项任务，每个功能可细分成更低一级的功能，如此细分直至基本活动，它成为一个可以被计算机(或人工)处理的过程。

在高层，一项业务即认为是一个功能，可细分为由若干个任务组成的任务组。图 5.4 中每个方块是一个功能，在最底层的方块是活动。图 5.13 表现了功能的分解过程，每个功能分解成子功能，直至达到一个活动。

包括最底层功能在内的所有功能都是逻辑的。它们没有指出任务实际上是如何完成的。一个可执行的处理过程是物理的，它明确指出了任务是怎样进行的。

2. 绘制层次结构图

James Martin 建议，把自上而下竖着绘制的树结构形式的层次结构图，改成自左向右横着绘制。理由是，在实际情况中，在每一水平层上有很多项，很难画在一张纸上，即竖着画很难办，但改变成图 5.14 那样把它们表示出来却很容易。

图 5.14 使用计算机很容易进行绘制和更新，其中星号 * 标识基本活动，即树结构的叶结点。

3. 基本活动

一个基本活动对应着一个计算机处理过程或人工处理过程，当这个过程自动处理又要使用数据库时，可以用数据库活动图描述。当把功能细分为活动时，要知道细分过程的终止。通常的标准是，如果可以用一句话说明一个活动的目的，则它可用为基本活动；如果需要几句话才能说明，那么这个活动也许应该继续细分。按上述特征确定基本活动，就没有必要长期争论什么是基本活动以及何时终止分解。James Martin 认为，顺其自然比机械地依附于规律要重要，如果把功能进一步分解能够得到所希望的结果，则继续分解，否则就停止分解。决不要把分解完全确定，总会有某些活动没有发现或以后要加入新的活动，模型应

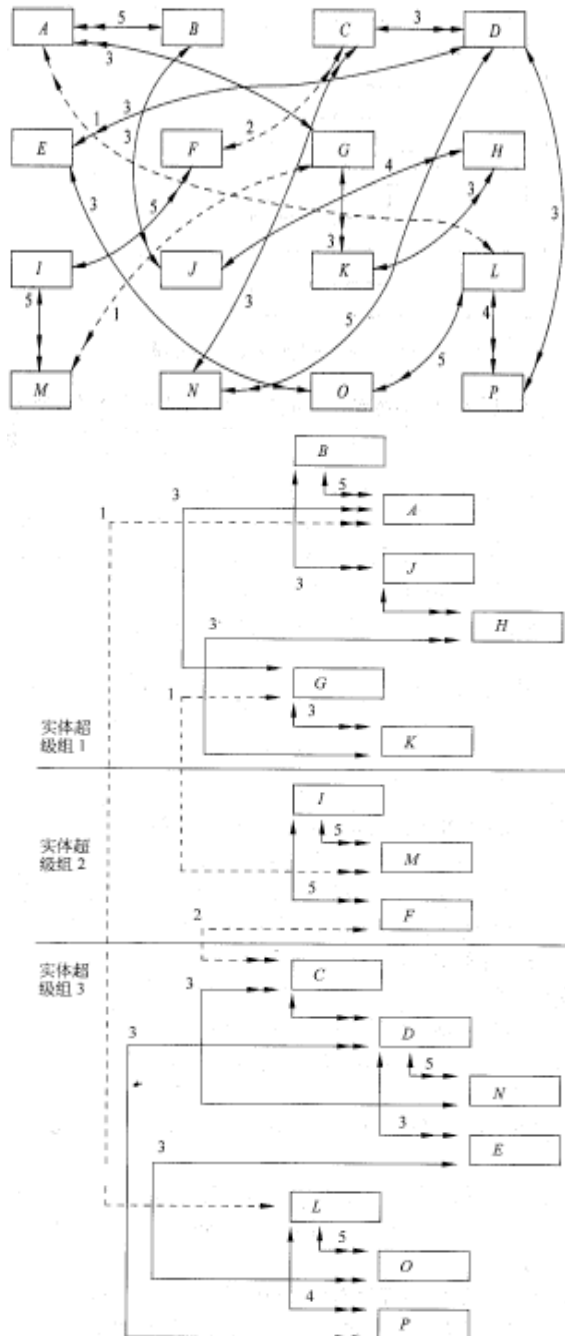


图 5.12 图 5.10 按照图 5.11 中所表示的联系被分成的实体超级组

可以修改。

当一个功能向下分解成一些低层功能时，重要的是要保证这些功能是充分的，并且它们每个功能对全局目标来说都是必要的。不论组织重组与否或自动化处理是否实现，模型建立者应该确信，他们所建立的实体活动模型比机构变化生存的时间要长，但这只在活动是必要的而且是充分的情况下才是正确的。有时能发现当前某些活动是多余的或是不必要的，实体活动分析有时会导致计算机过程的重组，导致企业部门或管理结构的重组。

4. 相关活动的特征

在形成活动模型时，探索活动间的相关性非常有意义，以下给出一些期望的具有良好组织的相关活动所具有的一些特性，它们是试探性的，而非绝对的要求。

- 一组相关的活动应产生某些明确的结果，这是活动的目的。该结果可以是市场对路的一种产品或一种决策，一次销售或一组可供选择的方案等。相反，一次组织粗劣的活动总是产生不可确定的结果或产生一些无关的结果。

- 一组相关活动有明确的边界，任何时刻都能指出活动上从事工作的人、工作的内容和起止时间。相关活动之间的转换具有明确的标记和界限，不相关的活动则相互混杂和重叠，难以确定它们在何时何处进行。

- 一组相关活动作为一个处理单位，它们可以由一个人或一个小组去完成并产生相应结果。活动的管理职责具有明确的规定，由一个人或一些人负责。相反，一个缺乏明确定义的活动也许是由一些不明确职责的人去执行，彼此缺乏配合，或分散于企业的不同岗位，无法使他们相互接触。

- 一组相关的活动，一旦运转，它们应该是自包含的，其行动很大程度上独立于其他活动。如果一个活动按照某种方式与另一活动需要频繁的相互作用时，那应该把它们作为一个活动来对待。与在不同活动上工作的人相比，执行相关活动的一组人间的往来要频繁得多。

5. 实体活动的映象

一般每个活动都与多个实体相关。心理学家的研究表明，大多数人一般能同时处理和掌握 7 个或 7 个以下的事件或概念，超过 7 个将会遇到很大困难，因此一个典型的活动最好使用 7 个实体，如果超过 7 个，那么就把它分解成多个活动。

高级终端用户在数据规划过程中起着辅助作用，他们应画出所管辖范围内的功能活动图，相互对照所使用的实体，如果对于一个功能需要哪些实体还不清楚，那么说明功能分解进行得不彻底。

上一节讨论建立全局的实体图，现在要参照这个实体图去建立业务模型和实体活动。

实体图和功能分解两者可能都不完全，随着时间的推移，应该修改和充实，开始时，希望得到一个全局概况，因而比较粗糙。随后，详细的数据模型化和计算机过程设计都要花费很长的时间，在进行仔细工作后，就可以更新功能分解图、实体图以及它们相互参照的数据库。

功能分解、画实体图和数据模型化都必须应用计算机化的工具，这些图都大且复杂，难以手工维护。James Martin 认为，缺乏自动化工具一直是不适当的数据管理的一个重要原因。

6. 可更新的自顶向下规划

自顶向下规划不应该只做一次，不再重复，这是大多数 BSP 研究的结果。

一般认为，在大型企业中，BSP 研究很难进行重复。然而，大多数企业都经常地发生变化，自顶向下的方法学必须与这些变化保持同步，并能及时更新，本章所研究的方法都具有这种特征，如果需要的话，它们可以和 BSP 研究联系起来，而对它进行改进。如果应用这些方法，并以计算机为辅助工具，自顶向下规划的更新是可能的。

5.5.3 企业的重组

在有管理人员参与活动分析时，常会提出如何来定义活动？当前实施的活动也许发现是多余的，而另一些活动可能以一种控制不住的方式散布在企业中，对出现的决策模型中的活动也可能有其他的选择。

而实体分析不仅是把现行组织机构转换成数据结构，而且提供给高层管理人员一种手段，通过实体分析而提出询问：根据分析的结果，企业或部门应该怎样改变？或按照外部环境该组织大致怎样改变？

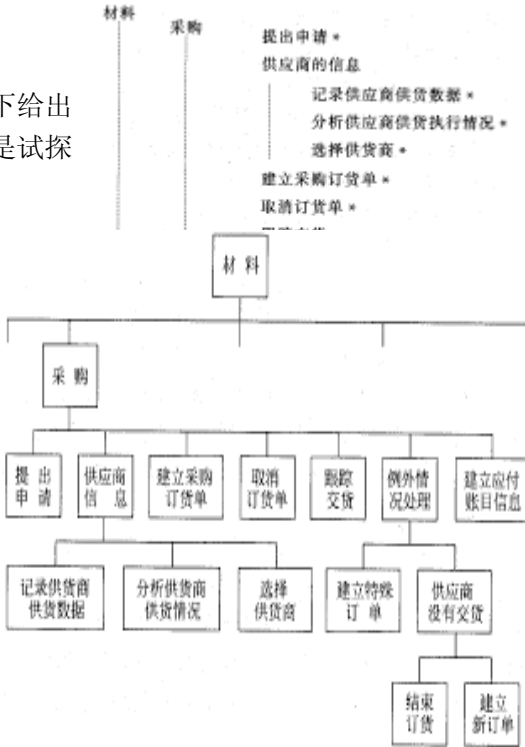


图 5.13 功能分解

因此，实体活动分析导致了过程的重新考虑，常会提出部门或企业的重组问题。James Martin 认为，为企业过程和结构方式的改善进行不断的研究是完全必要的。

这项研究的最好人员配备是长期地让一小部分核心人员做该研究，以保持研究的持续性。

当企业对现有管理过程的功能分析完成以后，更有价值的问题是考虑应该有什么样的管理过程和不应有什么样的管理过程。James Martin 认为，在一个组织机构以及它的信息资源中，数据库技术提供了实质性地改变其管理过程的机会。人们将逐渐认识到，数据库和屏幕技术必将导致一种与现在完全不同的、更加完善的组织工作方式。

5.5.4 亲合性分析

当实体用于描述活动时，把实体聚合成主题数据库的另外一种技术就成为可行的了。如图 5.15 中的一种矩阵可以出来表明一个实体与其他实体的亲合程度。

首先假设有两个实体 E_1 和 E_2 ，如果它们从来没有被相同的活动使用，那么它们的亲合度为零；如果它们总是同时被每一个活动所使用，那么它们的亲合度为 1；许多实体仅被某些活动一起使用，其亲合度在 (0, 1) 区间内。

计算机能够检验每个活动并计算出：

(E_1) = 使用实体 E_1 的活动数目

(E_1, E_2) = 同时使用实体 E_1, E_2 的活动数目

用这些数字可以计算出两个实体的亲合因子，求亲合因子的一种方法是： E_1 与 E_2 亲合度 = $a(E_1, E_2) / a(E_1)$

亲合因子可以用一个矩阵形式表示出来，图 5.15 所示。

图 5.15 表示两个不同实体之间所计算的亲合度矩阵，该矩阵可用于把实体聚合成主题数据库。

如果两个实体的亲合度比较高，则它们应该在同一个主题数据库中；如果它们的亲合度为 0，则不能放在同一个主题数据库中。然而分界线在什么地方呢？

计算机可以把实体按亲合度分类，如果把亲合度为 0 的实体分在相同的组，那么就只有一组；如果把亲合度为 1 的实体分在相同的组，则分组的个数可能等于实体个数。也可以把亲合因子分成能够产生 20, 30 或者设计要求的那么多组，这些组用作主题数据库。

亲合因子没有考虑到每一个活动的使用情况，另一种不同的计算亲合度的方法也许能考虑到活动的使用情况。在使用中，当自动地把实体组合成主题数据库时，这种方法给出了较理想的结果。

假若想要按图 5.15 把实体组合成数据库，那么对实体按照亲合度的大小存放，从高亲合度开始。每个具有最高亲合度的实体对形成该聚合的核心，于是：

E_1, E_4 (亲合度=0.92)

E_{11}, E_8 (亲合度=0.90)

E_6, E_7 (亲合度=0.88)

E_{10}, E_{12} (亲合度=0.87)

终于得到一实体对，其中一个实体已经在聚合中；假设碰到的下一个实体对是 E_2, E_8 (亲合度=0.85)， E_8 已经分配给了聚合 E_{11}, E_8 。那么应该把 E_2 与哪一聚合联系起来呢？为了确定这一点，必须计算

$$E_2 \text{ 到 } E_{11}, E_8 \text{ 的加权亲合度} = \frac{[(E_2 \text{ 到 } E_{11} \text{ 的亲合度}) \times a(E_{11})] + [(E_2 \text{ 到 } E_8 \text{ 的亲合度}) \times a(E_8)]}{[a(E_{11}) + a(E_8)]}$$

设 E_{11} 用于 3 个活动中， E_8 用于 48 个活动中， E_2 与 E_{11}, E_8 的复合亲合度为：

$$(0.34 \times 3 + 0.85 \times 48) / (3 + 48) = 0.82$$

这个值大于图 5.15 中余下的任何亲合值，所以 E_8, E_{11}, E_2 形成一个聚合。

今后，当我们遇到新的 E_8, E_{11}, E_2 亲合的实体时，我们就计算它们的复合亲合度，用这个方法构成的具有高亲合度的聚合的实体在稳定地增加。

图 5.15 中下一个具有最大亲合度的实体对是 E_7, E_4 (亲合度为 0.76)，然而 E_7 和 E_4 都分别分派到了一个聚合中。那么，这些聚合也应该被组合起来吗？为了确定这一点，先计算 E_7 与现存的聚合 E_1, E_4 的复合加权亲合度以及 E_7 与 E_1, E_4, E_6 的复合加权亲合度。假设各自的亲合度分别为 0.55 和 0.37，它们都低于表中下一个实体对的亲合度。即 E_8, E_{10} 的亲合度为 0.74，首先应该处理这一对实体。最后，每个聚合的决定都按亲合度数的大小顺序排列。

处于亲合数序列末尾的某些实体与其他任何实体的亲合性都是微乎其微的，这些实体可以作为文件系统或独立的数据库来实现，设计者应该认真观察所余下的低亲合度的实体，看它们是否属于任何现存的数

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}
E_1		0	0	0.92	0	0	0.01	0	0	0	0	0
E_2	0		0	0	0	0	0	0.85	0	0	0.34	0.17
E_3	0.01	0		0	0	0.12	0	0	0.07	0	0.38	0
E_4	0.01	0	0		0.20	0.11	0	0	0.43	0.01	0	0
E_5	0	0.02	0	0		0	0.21	0	0.08	0	0	0
E_6	0.21	0	0	0.73	0		0.88	0	0	0.12	0.08	0.01
E_7	0.35	0	0	0.76	0	0.30		0	0.01	0	0	0
E_8	0	0	0	0	0	0	0		0	0.74	0	0
E_9	0	0	0.21	0	0.09	0	0	0		0	0	0
E_{10}	0	0	0	0.06	0	0.17	0	0	0		0	0.87
E_{11}	0.01	0	0	0	0	0	0	0.90	0	0		0
E_{12}	0	0.01	0	0	0	0	0	0	0	0.21	0	

图 5.15 亲合度矩阵

数据库。

5.5.5 分布数据规划

随着计算机价格的下降，具有分散数据的分布处理方式日益广泛地发展。当数据分散、设备价格低廉时，分布数据处理比集中处理更为合适，从而对自顶向下的设计和控制也提出了更高的要求。如果自顶向下的设计没有进行，那么系统分析员或用户很可能要对自己地区的数据进行独立设计。

微型计算机价格低廉，如果能提供分布式数据库的支持，则会有更多的优点。人们总希望获得这些优点，使系统具有分布数据管理功能和严格控制功能。

如果技术成本允许，使数据存储于使用它们的地方是完全合理的。事实表明，数据的分散处理，使数据的录入和存储由原来的集中转变为分散，这样在用户部门直接控制下，数据的精确度和完整性大大提高。

分散管理的数据主要特性是数据在特定地区产生和使用，而在其他地区很少或根本不使用，数据的这些特性必然导致数据的分散管理方式的产生。下面介绍一些概念。

1. 分布式数据的 6 种形式

分布式数据存在 6 种不同的基本形式。

• **复制的数据**：相同数据在不同地方存储几个副本，从而避免系统之间的数据传输。当然，只有当查阅数据的频率大大高于更新数据的频率时，这样组织才有意义。大量复制的数据往往是较少变化的数据。

• **子集数据**：外围计算机储存的数据常是大型计算机的数据子集。子集数据是复制数据的一种形式，但它通常没有完整的模式，一般数据主拷贝是保存在较高层机器中，较低层机器的数据变化应立即或定期对高层机器中的数据进行改变。应该极力保障两层机器间数据的相容性。如果低层机中的数据与高层机中的数据不相容，就限制了两层机器之间的数据交换。如果要进行交换，必须付出代价来进行数据转换。

• **重组数据**：在信息系统或决策支持系统中为了使信息检索更易实现，必须利用特定的技术手段，如数据用倒排表、次索引，或用多个关键字将数据从同一机器或多个机器的数据库或文件中选取并进行编辑和重新组织。为了能满意地完成这些工作，其数据库类型必须相同，应有相同的数据表示，从同一数据模型和字典表示中导出。否则将付出数据转换的代价。

• **划分数据**：同一模型用于两个或更多的机器中，而每台机器储存不同的数据，每台机器具有不同的记录，但其构造形式使用的程序是相同的。

• **独立模式数据**：不同的计算机含有不同模式的数据和使用不同的程序，它们由不同的组织安装和使用。它们模式不同，但这些独立的数据系统应是一个公共的自顶向下规划的一部分。

• **不相容数据**：在不同机构建立的独立的计算机系统，数据没有统一设计和规划。一个用户有时从终端通过计算机网络去访问多个独立开发的系统，因此，他们必须熟悉每一台计算机如何储存数据和如何访问、使用这些数据。

2. 同步数据与不同步数据

复制数据、子集数据和重组数据中数据，相同内容可以存放在两个或多个机器中，这样要保持多个副本的同步就是设计中的重要问题，即当一个副本中的某个数据值发生改变时在其他副本中相应值是否能同步改变。

在大多数情况下，两个远程数据副本之间没有必要保持同步直到“分钟”，副本可以在一小时或一天脱离同步，只有一份副本被更新，这些更新以成批方式传送到另外的副本。

根据同步与否的数据副本的差别，可以划分出 9 个数据分布类型，即同步复制数据、不同步复制数据、同步子集数据、不同步子集数据、同步重组数据、不同步重组数据、划分数据、独立模式数据和不相容数据。

表 5.3 部门或场所对每个业务活动过程介入的情况

部门或场所	工	工	工	分	区	仓	总	部门或场所	工	工	工	分	区	仓	总
业务活动	厂	厂	厂	办	域	库	经	业务活动	厂	厂	厂	办	域	库	理
	A	B	C	处	办	处	室		A	B	C	处	办	处	室
市场分析						×	×	测量和下料	×	×	×				
产品范围评审						×	×	机器运转	×	×	×				
销售预测						×	×	销售区域管理				×	×		
财政计划							×	销售				×	×		
资本的获得							×	销售管理				×	×		
资金管理							×	客户关系				×	×		
产品设计	\	\	\	\			×	成品控制							×
产品定价							×	订货服务				×			×
产品规范维护	\	\	\				×	包装							×
材料需求	×	×	×					发货							×
材料订购	×	×	×					贷方和借方	×	×	×				
验收进货	×	×	×			\		现金流通	×	×	×	\	\	\	×
库存控制	×	×	×			\		工资	×	×	×	\	\	\	×
质量控制	×	×	×					成本核算	×	×	×				×
生产能力计划	×	×	×			\		预算计划	×	×	×				×
工厂调度	×	×	×					利润分析	×	×	×				×
工序设计	×	×	×					人事计划	×	×	×				×
材料控制	×	×	×					人员招聘	\	\	\	\	\	\	×
								劳保政策	\	\	\				×

注：×表示主要参与 \表示次要参与

3.没有地理位置的规划

初期给出的分布规划，没有表明数据存在的地理位置。由于分布数据所考虑的复杂因素，如网络的可用性、小型机和存储设备的费用、传输的费用、分布数据库软件和办公室的位置等，战略规划要花很长时间才能实现，在这段时间内分布参数实际可能会改变。因此，数据战略规划最初应该建立一个逻辑的、不考虑地理位置的数据图表示。

4.分布矩阵

一个组织中的业务活动过程可能发生在很多地方，对于地理位置的分布可以借助于矩阵表示。可以构造部门或场所对应于业务活动过程的矩阵，如表 5.3。可以构造部门或场所对应于所使用的主题数据库的矩阵，如表 5.4。可以构造部门或场所使用主题数据库的数据类型的矩阵，如表 5.5。还可以绘制业务活动过程、主题数据库和场所相应关系的三维矩阵，由于图形比较复杂这里未提供，可参看 James Martin 原著。

5.分布数据规划的相关内容

有关分布数据规划，应讨论的内容在 James Martin 的著作中还包括数据分布的定性和定量分析。定性分析即讨论分布处理系统中，如何设计各种应用程序的运行位置；数据分布的定量分析，即以某种方式去安排数据和加工处理位置，使得任意两点间的流通量和相互作用尽量的小。有关这些方面的技术和实例，请参阅 James Martin 原著。

6.分布数据规划过程

现在描述性地归纳有关分布数据规划的基本过程。

(1)按 5.4.3 提出的，用 BSP 方法过程，开发一个过程/主题数据库矩阵，在 BSP 方法中称为过程/数据类矩阵，即通过分析和规划把主题数据库与业务活动过程逻辑地对应起来，先不考虑数据的物理位置。

(2)把用户所在地点与其所使用的业务活动过程列在一起。设计者把用户所在位置和业务活动过程及其对应的主题数据库绘制出来，如表 5.4。

(3)估计过程所在的位置和数据间的事务处理量，并指明事务处理是交互式的还是批量式的。

(4)决定哪些数据是复制的、子集的、重组的、划分的或独立模式的，并决定哪些重复的数据用于哪些位置。这一步考虑到现有的文件和数据库尤为重要。

(5)给出表示数据的地理位置和分布类型图或矩阵，以及给出表示事务处理量的估计和处理方式(交互或批量)的确定。要对数据做某些移动或细化，以达到最小传输量。表 5.6 对以上步骤做示意说明，而对于数据战略

表 5.4 部门式场所对应于主题数据库的矩阵

主题数据库 部门及场所	计	预	财	产	产	部	材	公	供	采	材	机	正	设	销	客	销	成	订	支	成	雇	工
	划	算	务	品	计	录	单	要	商	购	存	载	工	作	例	户	售	本	存	付	本	员	资
总经理室	C	C	C	C	C				C					C	U	U	U	U	C	C	C	C	C
仓库																		C	U			C	C
区域办事处	U	U	U													U	U	U		U		C	C
分办事处																C	C	C		C	U	C	C
工厂 A		U	U	U	C	C	C	C	C	C	C	C	C	C	C	U		U	U		C	C	C
工厂 B		U	U	U	C	C	C	C	C	C	C	C	C	C	C	U		U	U		C	C	C
工厂 C		U	U	U	C	C	C	C	C	C	C	C	C	C	C	U		U	U		C	C	C

注：U—使用 C=建立和使用

表 5.5 在第一部门或场所的主题数据库

主题数据库	计	预	财	产	产	部	材	公	供	采	材	机	正	设	销	客	销	成	订	支	成	雇	工
	划	算	务	品	计	录	单	要	商	购	存	载	工	作	例	户	售	本	存	付	本	员	资
公司	订货(子集)																					15I	110I
分公司	顾客(子集)																					110I	110I
产品(复制)																						20I	20I
工厂 B	供货商(划分)																						
采购(划分)																							
劳动力(划分)																							
产品(复制)																							
工作进度(独立模式)																							
机器工具(独立模式)																							
工厂 A	供货商(划分)																						
采购(划分)																							
劳动力(划分)																							
产品(复制)																							
工作进度(独立模式)																							
机器工具(独立模式)																							
总公司	订货(主数据)																						
顾客(主数据)																							
成品库存																							
产品(主数据)																							
成本(重新组织)																							
采购(子集)																							
供货商(重新组织)																							
主题数据库	总成本																						
生产计划																							
应付账目																							
工厂 A	工作报表																						
机器调度																							
采购																							
生产计划																							
工厂 B	工作报表																						
机器调度																							
采购																							
生产计划																							
仓库	产品库存																						
订货																							
发货																							
12个分公司																							
销售服务																							
订货服务																							
顾客系																							

表 5.6 主题数据库/过程/位置/信息流/传输方式/信息流/估计

规划的其他有关内容，可以用粗或细的方式按表 5.6 画出，当规划进行到物理设计时，对分布的细节就越来越紧迫，这时在活动和实体层应该做一个精细的设计。

表 5.6 是一个主题数据库与过程/位置矩阵，其中 I 表示交互式使用方式，B 表示批量使用方式，数字表示每天事务处理流通量，方框外的数字是传输的流量，表中将不同位置的数据细划分成不同的分布数据类型，被放置的数据使得不同位置间的流通量或潜在的有害的相互作用减小到最小。

(6) 细化分布数据规划，确定数据结构，哪些应该是数据库结构，哪些应该是文件结构，与现有文件、数据库和应用程序的关系都要纳入设计中，最后画出地理数据结构矩阵，它是表 5.6 的扩充，它可能很大而且有很多页，可能经常修改，从而希望由计算机来维护。

(7) 在数据定位中，有一些实际约束检查，包括恢复、安全、重新启动，以及可能引起以物理数据结构矩阵为基础的调整。

(8) 检查应用程序，以决定哪些事务处理被使用以及需要哪些应用程序来处理它们，并建立表示每个程序所使用的数据结构的矩阵。为了检验应用分布的效果，可将程序分为 4 类。

- 0 类：它与它所服务的业务过程和使用的数据在同一地点。
- 1 类：它与它所服务的业务过程在不同的地点，但与所使用的数据所在地点相同。
- 2 类：它与它所服务的业务过程在同一地点，但与它所使用的部分数据或全部数据所在地点不同。
- 3 类：它与它所服务的业务过程和所使用的部分数据都相距甚远。它也可能与 2 类程序相似，只不过是接近数据而与过程分离，而不是所有数据都在同一地点。

0 类程序是最需要的，而 3 类程序应该避免。集中式系统中，常用的是 1 类程序。为了避免模型间的相互作用，应该调整地理数据结构矩阵，使得。类程序占最大比例；避免 3 类程序，同时使 2 类程序越少越好。

分布数据规划无疑是比集中数据规划更为复杂的过程，但往往又是需要的过程。这里概述了有关概念和应讨论的问题，要深一步的实践还会有无数的问题要细化和精化，战略数据规划方法也仅提供了粗的框架。

5.6 战略数据规划过程提要

James Martin 在其战略数据规划方法学中的最后给出了自顶向下战略规划的一个全过程，并说明这一过程集中反映了使用各种方法论的经验，综合了各种方法的优点，以便体现他 (James Martin) 感受到的最好结果。

(1) 得到企业最高管理层的委托。

(2) 选择一套合适使用的方法，并实施其方法。

(3) 定义业务职能范围：

- 绘制一张表示企业、部门和职能范围的图表。
- 由最高管理者 (指导委员会) 进行复审。
- 确定研究范围。
- 准备一份研究的时间进度表。

(4) 拟定职能、活动和实体样本。

• 建立研究小组 (要包括两名数据处理人员，其中一名数据库专业人员和两名对企业情况很熟悉的用户工作人员)，小组必须具有一位权威的领导人——他是一位善良而果断的领导者。

- 确定各职能范围的负责人。
- 选择用户分析员。
- 培训用户分析员。

(5) 把每一职能范围划分成一些业务活动过程，这些业务活动过程可通过各个用户小组进行审查。

- 给出每个业务活动过程的定义。
- 用户分析员复审这些业务活动过程。
- 整理业务过程表，并由用户分析员对它再次核实。
- 建立机构负责人与业务活动的联系。

(6) 把所有业务过程分配给不同的用户分析员以便复审。

(7) 把每个业务过程分解成功能和活动。

- 确定每个业务活动过程所需要的实体。
- 为所确定的实体命名。
- 逐步建立实体图，该图描述了与现存的业务过程、活动、组织结构及与地理位置无关的企业经营中

所需要的数据信息，并将实体分成组或超级组。

- 根据建立相应的主题数据库的需要，对超级组进行调整。
- 为每个超级组命名。

(8)可选择项:

- 把活动映射到实体上。
- 为了在活动基础上聚集活动所支持的实体，使用相关分析算法。
- 改进实体超级组，并用实体图中已产生的超级组对它们作交叉检查。
- 必要时，调整实体图中的超级组。

(9)建立超级组(主题数据库)与业务过程的对应关系。

- 调整图。
- 产生功能聚集的簇，以便形成逻辑功能范围，它们将成为各个系统的基础。

- 确定任务繁忙的日常系统。
- 确定信息系统和决策支持系统的需求。

(10)绘制出现存的数据处理系统。

- 确定所设计的数据库系统与旧系统的联系。
- 计划一个转换策略和一个时间进度表。
- 设计出新旧系统间联系的桥梁。

(11)选择需要采访的高级管理人员。

- 列出采访的问题。
- 进行采访，使业务活动过程与主题数据库对应关系得到确认。

- 确定这一阶段中对当前和将来需要信息应做的考虑。
- 确定数据处理中还存在的问题或被采访人的不同观点。
- 分析采访内容，以决定是否需要改变系统结构。
- 分析采访内容，设置系统实现的优先级别。

(12)画出每个业务活动过程出现的位置。

画出业务活动过程出现位置与主题数据库的对应关系图。

- 检查可分布与不可分布的理由。
- 给出规划分布过程。
- 画出一张数据分布图表，表示6种类型的分布式数据。

(13)向用户分析员送交主题数据库、信息系统和分布系统产生的各种图表。

- 对图表和报表作必要调整。
- 呈交同样报表、图表给管理者。

(14)设置实施过程的优先级别。

- 为实施过程制定时间进度表和职责。
- 建立与自顶向下规划工具相连接的自底向上的设计工具。

(15)建立职责，确保对自顶向下的规划进行不断的更新。

(16)准备和呈交一份结束报告。

以上提供了战略数据规划的参考过程，要全部按条实施还可能遇到本章中未详细介绍的技术内容，可以参阅 James Martin 的有关著作。

这里所提供的规划技术是建立整个企业的数据系统的一整套方法中的第一阶段。这个阶段在图 5. 16 中的最上一层。在图 5. 16 中其他各层在 James Martin 的另外的著作“数据库环境的管理”和“无程序员的开发”两本书中讨论。

如果顶层被充分实施，那么底层的各级将更容易完成。在某些工具的情况下，甚至不需要程序员而自动的产生各种应用过程。

5.7 结论

关于 James Martin 的数据规划方法学介绍到这里，它包含着丰富的内容，并已形成整套的理论、技

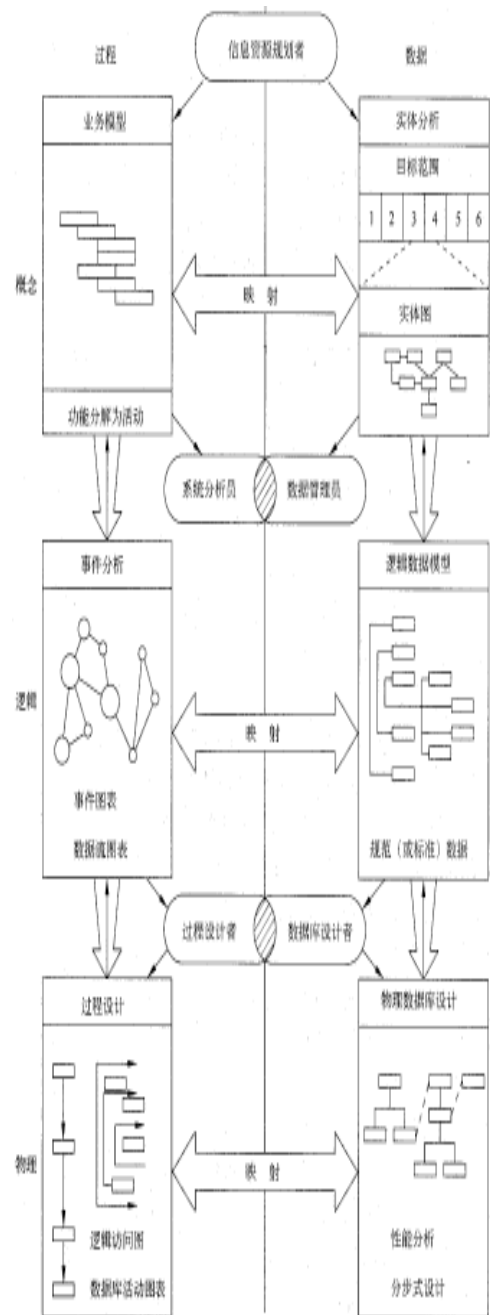


图 5.16 对于信息管理的的方法论

术和方法，其内容与本书第 4 章介绍的 BSP 方法是相容和一致的，它们之间有内在的联系。需要深入研讨的读者可以系统学习有关的参考书和有关资料，James Martin 强调用自动化的手段来实现自动化的工程，如果能将方法中所提供的技术工作以计算机辅助完成，那将是最完美的信息工程环境。

从某种意义上讲，这里介绍的内容只是 James Martin 方法的导引，将这些方法付诸于工程实践还有待更说细的参考资料。

第 6 章信息工程方法

6.1 信息工程基本概念

6.1.1 信息工程发展过程

信息工程 (Information Engineering, IE) 20 世纪 70 年代初期，由美国学者 James Martin 创建的、面向企业计算机信息系统建设的方法和实践。在本书的第 4 章和第 5 章中对其基本思想和内容已有体现。本章更全面地介绍其相关内容，并更着重其整体原则和方法。

20 世纪 80 年代初期，James Martin 完成了一系列有关信息系统开发的著作，其中包括《战略数据规划方法学》和《数据库环境的管理》，集中地阐明了两点：

- 计算机化的信息对企业和机构的重要性，它是企业和机构最宝贵的资源。
- 计算机化的企业和机构，需要对其全部数据加以全面的战略规划和管理。

并认为，缺乏上述两方面的认识，将很难建设成一个现代化的企业，因此 James Martin 将信息工程的理念向前推进到实用。20 世纪 90 年代初期，他又完成了《信息工程》三卷著作，系统地给出了信息系统的开发策略和方法，并且给出了支持信息工程的自动化工具和工程化流程；指出信息工程是在一个企业或企业的主要部门，关于建设信息系统的规划、分析、设计和构成的一整套相互关联的正规化、自动化的技术应用，

强调了实施信息工程的关键因素是：

- 信息工程是在企业范围内应用结构化技术来实现企业信息系统的建设。
- 信息工程应自顶向下分步骤来实施和创建企业的信息系统。
- 信息工程要建立企业的全局模型，即企业的业务模型、数据模型、业务与数据的交互作用模型，从而构建起开发一个计算机化企业的框架，并可在框架内独立开发各个应用系统。
- 信息工程应使用自动化的工具和设施，快速建立和修改各种应用系统，以适应变化的需求。
- 信息工程在系统开发的各个阶段充分使用各种规范化的开发技术，如重用技术、原形技术来实现系统应用。
- 信息工程需要调动各类人员，从最高管理者、业务人员、技术人员到最终用户，他们将参与系统建设的各个阶段，并发挥积极性。

有评论认为，在 20 世纪 90 年代后期，不应用信息工程的方法，就可能无法建立计算机化的企业。James Martin 现已被公认为是管理领域和信息技术领域的最有影响的人士。多年来，他不断地把信息技术的最新成果创造性地引入到现代企业的经营管理之中，对提高企业竞争力发挥了重要作用，它已产生并将继续产生深远的影响。

6.1.2 信息工程概念

信息工程是计算机信息系统发展的产物，它不仅为大型信息系统的开发给出了方法和技术，而重要的是它立足于实践对大型信息系统的开发提出了相应的开发策略和原则，而这些策略和原则是从长期的实践中获得的，遵循这些原则，采用相应策略将在很大程度为系统的成功提供保证。如果人们普遍认为信息系统的建设是一项复杂的社会和技术工程、而其复杂的一面又丰要表现在其社会性上的认识是正确的话，那么信息工程方法就不仅在系统建设的技术性方面，而且在社会性方面都给出了相应的解决思路和途径。

因此可以认为：信息工程虽然是在 20 世纪 80 年代末期发展起来的，但却是在 20 世纪 70-80 年代对大量经验和教训的积累和总结的基础上，进行分析、总结、规范而形成的一整套方法论。它随着信息技术的发展和信息系统建设的实践经验的不断积累以及企业间的竞争加剧所提出的信息需求的多样化和复杂化而不断丰富、完善和发展。

要全面、准确给信息工程下定义可能是困难的，但可以认为：信息工程是建设企业计算机化的信息系统工程的方法和实践，它从业务和技术两个方面为系统的建设提供规范和完备的社会和技术手段，对企业

的信息系统的建设进行规划、分析、设计和构成。

上述企业是泛指，它可以理解为工商企业，也可以是学校、医院、科研、交通等部门，甚至是政府机构。虽然他们业务内容可能各几异，但相应的信息系统的开发会遵从信息工程方法所归纳的普遍性原则，可以运用相同或相似的方法来实现。当然信息工程方法应该强调其实践原理，并可运用创造性的方法来结合实践需求加以实现并发展。

然而，信息工程方法无论如何变化，它都应该具有以下特征或保持以下关键成分：

- 信息工程从企业的整体着眼并应用结构化技术来创建企业的信息系统。
- 信息工程运用自顶向下方式，通过信息战略规划、业务领域分析、系统设计和系统构成等步骤来实现企业的信息系统建设。
- 信息工程需要建立用于存储企业数据模型、过程模型、各种设计信息的信息库。
- 信息工程构筑起一个计算机化企业的框架，并可在这个框架内独立地开发各个系统。
- 信息工程的各种阶段的实现都应使用计算机化的工具，可快速创建和修改各个系统。
- 信息工程是整个企业范围的开发方法，能使各分别建立的系统协调一致，可最大限度地使用可重用技术。
- 信息工程要求企业最高管理者亲自参加和领导信息战略规划的制作，要求最终用户参与到系统建设并发挥起业务专长。
- 信息工程应促进信息系统的长期发展，应能确定有助于企业战略目标实现的计算环境，使系统能适应信息需求的变化。

6.1.3 信息工程的组成

信息工程与传统的信息系统开发方法相比更注重对信息系统开发过程的整体支持，它提供了系统的方法论，强调应有与方法论相配合的自动化工具以及开发环境，并十分强调开发中对以往成功经验的吸收和利用。因此可以认为：信息工程不仅是方法，而是方法、工具、环境和经验的集成而经验又是在实践中产生、积累并经过分析、提升和总结而形成的。

归纳上述，可以将信息工程的组成总结如下：

- **系统的方法论。**信息工程形成了“以数据为中心”，而不是“以应用为中心”的开发方法，并在方法中强调以数据为战略资源，以数据规划为基础的信息工程方法，它以主题数据库的组织和实施来实现，并提供直到系统完成的各阶段的实施方法。
- **完备的工具集。**信息工程形成与方法论配合的完整的工具集。没有完备的适应于开发系统各阶段的工具，将难以保证方法论的有效实现，工具集能将各阶段的工作任务有效加以实施，且规范和完整。各阶段有明确的输入信息要求和输出信息要求，便于较机械地执行。
- **信息工程环境。**信息工程形成以信息库为核心的信息工程环境。信息库积累了信息系统的规划、分析、设计以及系统维护信息，并完整地辅有相应的对信息的处理工具。
- **成熟经验总结。**信息工程总结了美国 20 世纪 80 年代信息化建设中的系统经验和教训，对我国当前的信息化建设有很大的借鉴。

6.2 信息工程方法

6.2.1 信息工程金字塔表示

信息工程方法认为，与企业的信息系统密切相关的三要素是：企业的各种信息、企业的业务过程和企业采用的信息技术，即信息、过程和技术构成企业信息系统的三要素，正如一座 3 个面的金字塔。信息工程则自上而下的将整个信息系统的开发过程划分为 4 个实施阶段：信息战略规划 (Information Strategy Planning, ISP) 阶段、业务领域分析 (Business Area Analysis, BAA) 阶段、系统设计 (System Design, SD) 阶段和系统构成 (System Construction, SC) 阶段，如图 6.1 所示。

在实际实施过程中，金字塔的 4 个阶段又会划分为若干个步骤来完成。

6.2.2 信息工程步骤：

信息工程的 4 个阶段在实施中，根据其任务和性质一般可再划分为 7 个步骤，如图 6.2 所示，步骤的内容分别是：

(1)信息战略规划(Information Strategy plan, ISP)。分析并建立起企业信息需求的全面视图，并产生企业的信息战略规划。

(2)业务领域分析(Business Area Analysis, BAA)。对特定业务领域进行详细分析，并产生业务领域说明 (Business Area Description)。



图 6.2 信息工程的 7 个步骤

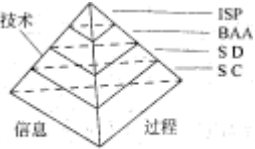


图 6.1 信息工程金字塔

(3)业务系统设计(Business System Design; BSD)。详细描述支持特定业务领域的应用系统,并产生企业系统说明书(Business System Specification)。

(4)技术系统设计(Technical System Design, TSD)。将业务系统设计的结果定制在特定的目标计算环境,并产生技术系统说明书(Technical System Specification)。

(5)系统构成(System Construction, SC)。建立系统的全部可执行部分,完成用户认可的系统功能(Accepted System),并完成相应的用户规程(User Procedures)和操作系统(Operational Procedures),实现系统构成(System Construction)。

(6)系统转换(System Transition, ST)。将新构成的应用系统安装到生产环境中,以代替原有的系统,实现系统转换(System Transition)。

(7)系统运行(System Production, SP)。系统投入生产,并充分认识应用系统所带来的效益完成系统运行(System Production)。

6.3 信息战略规划

6.3.1 信息战略规划的任务

信息战略规划是信息工程实施的起点,也是信息工程的基础。

信息战略规划的目的是要评估实施信息工程的企业的信息需求,并在评估信息需求的基础上去建立这些信息需求的信息结构,进而建立支持相应信息结构的业务系统结构,并确定支持业务系统结构所必需的技术结构,最后提交信息战略规划的结果。其流程如图 6.3 所示。



图 6.3 信息战略规划流程

根据上述流程,信息战略规划的具体任务应包括如下内容:

(1)在实施信息战略规划前,必须制定信息战略规划项目的计划,即要确定信息战略规划项目的范围和内容,建立信息战略规划项目相应任务的进度表,并选择和培训项目组成员。

(2)初始评估。除应对企业信息需求进行评估,即为了实现企业的战略目标,保持企业的竞争优势,要对企业所需的信息进行评估外,还必须对企业的组织机构和信息环境作出必要的评估,并对其进行必要的分析和审查,从所得结果产生规划工作框架。

(3)定义信息结构。根据评估的相应结果,定义出企业运营所进行的业务活动所需要的数据和信息,进行精化后给出高层次的企业模型和相应的图表。

(4)评估当前的环境。对企业当前的业务环境和技术环境进行较全面的调查和评估,为确定业务和技术系统结构提供支持。

(5)确定业务系统结构。确定支持信息结构所需要的业务系统和相应的软件、硬件和业务系统的运行、传输环境。

(6)完成信息战略规划项目,提交信息战略规划报告。

上述扼要地介绍了信息工程方法的基础工作,即信息战略规划所包含的任务,以下各项将更为具体地陈述各个环节所包含的内容。

6.3.2 信息战略规划的实施

收集和整理企业的原始数据和资料是制定企业信息战略规划的首要工作。通过对所收集数据和资料的分析,从而来确定企业现有的信息技术环境是否能支持企业应完成的任务并做出规划。

6.3.2.1 数据和资料的收集

为了制定企业信息战略规划将涉及到以下多方面的资料。

1.有关制定企业计划的资料

制定企业计划的资料包括企业的任务、目标、战略、计划和关键成功因素(Critical Success factor, CSF)等,它们是企业业务规划将涉及的内容。

任务是对企业的目的和性质的总体描述,通常会用一个总的任务来描述一个企业。例如,在信息工程的相关资料中曾举出两个美国的例子:一个是“联合农具公司”,它的任务是通过制造和销售农业机具,为投资者获得最大效益;另一个是非赢利的“野生动物保护组织”,它的任务是募集资金,用于识别和保护濒临灭绝的动物物种。下面我们也借用这两个例子,并结合我国的情况来说明问题。由于任务的不同和企业性质的差异,决定其收集资料的类型和性质将会有所不同。

目标是企业为完成其任务,在一个长时间内需要获得的结果。而时间的长短则取决于企业的任务。例如,联合农具公司的目标会是“成为某一地区农业机具的市场占有者”,而野生动物保护组织的目标之一会是“为保护某种濒临灭绝动物,如为熊猫而在某地区建立一个保护区”。从以上叙述可见,目标可能会有总目标和具体目标的差别。

战略或称策略是实现企业目标时所应采取的正确方法，没有正确可行的方法就不能或难以达到企业预定的目标，企业目标必须有实现目标的战略或策略支持。如农机公司实现目标应采取的策略是“针对地区需求，开展市场研究，确定相应地区最需要且价格适中的农机产品”；野生动物保护组织要实现预定目标所应采取的策略则应是“与可能提供资金支持的企业和私人保持和建立密切的联系”。

关键成功因素(CSF)是指影响企业目标实现的特定因素。如联合农机具公司的关键成功因素可能是其产品“能满足地区客户的特殊需求”；野生动物保护组织的关键成功因素是“向人们宣传保护野生动物的意义”。当然，对于企业而言，CSF可能不是惟一的，如果如此则需要首先去确定关键的CSF。

计划是用以实现相关战略和处理关键成功因素的活动时间表。而计划中一个活动的进度是用性能来监测的，它表明所度量目标的实现程度，特定目标的实现又会受到关键成功因素的影响。如农机公司可用每季度其销售量所占市场份额来监测其所制定的目标；野生动物保护组织可用每年与捐款人建立了多少联系来监测其所要达到的目标。

2.有关组织结构的资料

组织结构是指所定义的组织单元间的关系；而组织单元是指由若干组织角色所构成的组成企业的实体或与企业进行交易的外部实体；而组织角色则是指根据所承担的工作，分配给个人或组织单元的职位类型，如负责人、业务专家、技术权威或工作小组等。

3.有关业务活动的资料

企业都具有自身的职能范围，它确定了一个企业的主要业务领域。企业的职能范围又都包含一定数量的业务活动过程，在每个业务活动过程中，都有一定数量的业务活动。企业往往都被划分为若干部门，部门又包含着若干业务职能范围，还可把企业的业务职能范围分解成多个功能，它对应于业务活动过程，每个功能还可被分解成更基本的低层功能。这样逐级向下分解，直到最基本的业务活动。

在多数企业中，往往未能利用结构图把业务活动表示出来。当人们在结构图上列出相关活动，并标明它们与其所用的数据之间的关系时，可能会发现某些重复性的活动，这是由于不知道同样的活动已在其他领域发生。因此要尽量控制所用数据的冗余度。

当战略规划列出了所涉及的活动以及这些活动所使用的数据时，应该尽量减少重复性的活动，并通过企业业务活动、结构图，把业务活动同其所用的数据对应起来，从而揭示出相互重复的业务活动，并重组企业的业务活动。

以下按层次来说明与业务活动或称活动有关的概念。

• **业务功能：**有时也称为业务活动过程，它是能完成企业某一方面业务的高层次业务活动。在每一个业务功能中，都有一定数量的业务活动，如在一项订购物资的业务功能中，它会包含如下一系列的业务活动：提出采购申请、选择供应商、提出采购订单、依据订单，执行交货条款、处理例外情况、准备付帐信息、记录供应商执行合同情况和分析供应商执行合同情况。

事实上，业务功能描述了企业独立于组织结构所应完成的工作。通常应该用名词来描述这些功能，例如生产、销售、装运等。

• **业务活动：**是最基层的业务功能，它的实现往往是通过特定类型的实体的输入、处理和输出来体现的。一般可由动词来描述这些活动，以表示该活动是要执行的操作，例如接收订货、发送货物、采购材料等。

• **主题域：**它通常是指与企业业务密切相关的事物和对象，例如重要资源、产品、供应商等。

• **实体类型：**它是企业需要保存的数据的载体，是具有相同定义的实体的集合，例如雇员、客户等。

• **实体关系：**企业中相同或不同实体类，型间的联系，例如客户实体类型和订单实体类型，通过“发出”来建立关系，即客户发出订单。

4.现有系统环境资料

为了评价现有的计算机系统，对于硬件、软件、通讯、数据存储和分布是否能支持企业的当前和长远的业务需求，以及对业务的覆盖率等情况，需要获取相应资料。

现有系统环境是指支持当前企业各方面业务活动的计算机系统，应获取每个系统的名称、类型和功能以及其覆盖的业务范围、采用的产品或技术、对现行业务的支持程度。

如果将计算机系统按其性质划分为 4 类，则可区分现有系统的性质和功能。

• **决策性系统：**它辅助企业对重大的、难以预测的战略性问题进行决策，其过程具有不确定性。

• **规划性系统：**它支持对企业特定问题在一个较大的框架内，运用分析和运筹手段求取某些满意的答案。

• **控制性系统：**它支持对企业的可执行系统进行监测和管理，并提供常规的分析报告。

• **操作性系统**：它支持对企业日常事物的处理，包括各种预定义处理、批处理和日常的办公事物处理，是一类大容量、可操作的实时处理系统。

按照系统性质划分，可确定企业现行系统的功能，在对现行系统了解的基础上对新系统进行规划。

现行系统的数据存储，包括存储内容和分布情况，它应是一些支持企业管理和决策的永久性数据，可以是文件系统或数据库。可对它们的命名、内容，功能和分布位置做简要记录，例如产品数据库、客户数据库等。

5.当前技术环境的资料

当前企业已存在的技术环境的相关资料应包括：

- **硬件产品**。包括主计算机、服务器以及各种辅助设备。
- **软件产品**。包括操作系统、数据库管理系统或其他，如某些支撑软件、网络软件。
- **网络产品**。包括路由器、集线器、调制解调器等。
- **应用系统**。包括为各种业务所编写的应用程序、所采用的、编程方法和编程工具。

6.3.2.2 审查文档资料

审查企业业务和技术文档，从中获取有关企业主要业务和技术环境的信息和知识，可以作为信息战略规划的原始依据，并从中提出需要深入调查和了解的有关问题，这些文档资料应该包括以下几类。

• **业务文档**。从业务角度来审查企业的相关文档，诸如公司的业务计划和预测、组织图表和手册、年度报表、公司账目表、公司指南、业务实践备忘录以及广告文献等。通过业务文档可以从宏观和微观两个方面来了解企业的业务活动。

• **技术文档**。从技术角度来审查企业的相关文档。诸如信息系统开发文档、信息系统运行平台结构、数据库结构以及在系统开发中所采用的方法、技术和工具。通过技术文档可以了解现行系统所采用的技术对现行业务系统运行的支持程度和不足。

• **系统文档**。对信息系统的现行系统所提供的文档的分析。诸如对硬件系统、系统软件系统、数据库系统、应用系统等分析可以评估现行系统的功能和性能。

经审查后将包含重要信息的文档进行编目以备在进一步规划时查阅和利用。

6.3. 2. 3 通过采访获取资料

通过采访获取企业有关重要的资料是信息工程方法获得资料的重要方法，并形成所谓一套称为“结构化的采访技术”。

1.采访的准备

采访是在获取和分析了企业的有关资料以后进行的，只有在初步掌握企业的相关资料以后，再进行有针对性的采访才能有高效率的收获。

2. 采访的对象

在制定信息战略规划时一般应采访三类人员。

• **最高管理者**。通过对企业最高管理者的采访，一方面可以加深企业高层管理者对项目应承担义务的认识，同时也能加深项目人员对企业业务和高层管理者对企业几发展战略认识的理解。

• **中层管理者**。通过对中层管理者的采访，可以进一步加深对各部门业务内容的理解，澄清在资料中某些存在的问题。

• **其他相关人员**。他们能提供某些关键信息或提供更进一步的资料，或对某些问题有某些独特的见解。例如某些退休的管理人员、技术人员等。

3. 采访的技术

信息工程方法根据经验形成了所谓结构化采访技术，其内容可归纳为：

• **采访人员**。采访人员一般由两人组成，其中一人负责提出问题并与被访人交流，另一人负责记录采访中得到的信息。一般在采访中不要使用录音等电子设备，这样有利于被访者能较坦诚和轻松地对待采访，并保持平和的气氛。

• **采访准备**。通过对书面文档的审查，应研究被采访者的职责，所在组织的结构和职能，并对被采访者有初步了解。准备好一份与被采访者讨论的主要问题的表格或提纲，并预先提交给受访者，用以引导采访的正常进行。

• **进行采访**。准时按约定时间和地点进行采访；采访开始前，采访组成员作自我介绍，陈述采访意图、目的和要求。访问中应首先讨论主要和重要的问题，并争取能覆盖所有的问题以及事先并未提出的问题。对所有讨论问题应及时记录，力争准确和完整。

• **整理结果**。结果的整理应按规划要求的专业化形式进行，可以采用简化、列表方式并将整理结果反馈给被采访人。如可对收集、到的数据进行分类归总，归纳成如下项目的表格：问题、解决办法、价值说

明、信息系统需求、影响到的业务、活动过程、引起的业务活动过程。

• **时间和次数。**经验表明，一次采访约两小时：对一般企业而言，对整个企业的信息战略规划，大约需要访问 40 次；对企业的部门制定信息战略规划，约需访问 15-20 次。

6.4 建立企业模型

建立一个初始的高层企业模型是信息战略规划的任务之一。高层企业模型应包括企业的高层业务功能，业务处理的主要数据和这些数据之间的关系。建立企业模型，首先要从识别企业的组织机构入手，并进一步确定企业的任务、目标和关键成功因素及其对信息的需求。

6.4.1 识别企业的组织机构

建立企业模型的方法是通过审查有关组织机构的书面文档来获得资料，并把所获得的原始资料作为信息源，再利用有关软件工具，如信息工程设施(Information Engineering Facility, IEF)中的规划工具箱来建立组织层次图。IEF 是由美国德州仪器公司(Texas Instruments)提供的计算机辅助规划、分析和设计的工具，其中规划工具箱是 IEF 的组成部分，它将识别企业组织机构和产生组织层次图规范成以下过程。

- **信息来源：**包括组织机构文档
- **输入信息：**包括组织机构图，组织手册，组织文件等。
- **输出信息：**包括组织层次图和组织单元的信息记录表。

IEF 的规划工具箱分两步完成上述过程：

(1) 利用组织层次图表来表示层次形成的组织结构，如图 6.4 所示，图中组织单元方框下的竖线表示方框可继续分解。

(2) 建立组织单元信息记录，包括名称、负责人姓名和职务、组织单元的任务和组织单元间的关系。

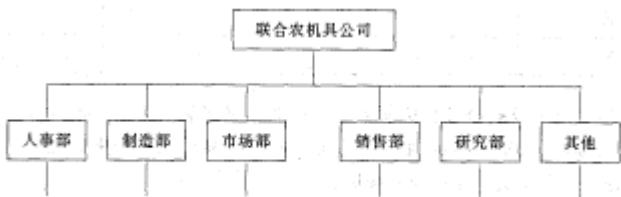


图 6.4 组织层次图示例

可见，识别企业的组织机构和建立组织层次图，是从企业所提供的组织机构文档中获取相应信息、并以此为数据输入信息、通过 IEF 规划工具箱形成并输出组织层次图和相关信息的过程。

6.4.2 企业的任务、目标和关键成功因素

为了获取企业的信息需求，应在企业组织机构的基础上，再利用审查时所获得的书面文档中的相关材料，进一步识别和分类企业的任务、目标和关键成功因素(CSF)。

信息工程方法将企业任务、目标和 CSF 的识别规范为以下过程。

- **信息来源：**包括审查的书面文档及相关材料。
- **输入信息：**包括组织层次图、业务计划、年终报告、备忘录等。
- **输出信息：**包括企业任务说明，组织单元目标和关键成功因素表，企业目标/组织单元目标矩阵等。

可分 5 步完成上述过程。

- (1) 考察和分析全部的可利用文件，从中确定并列出企业的任务、目标和关键成功因素。
- (2) 将任务、目标和 CSF 组成任务说明，任务说明书详见举例。
- (3) 记录组织层次图中与企业直接相关的组织单元的目标和关键成功因素。
- (4) 建立企业目标/组织单元目标矩阵，如表 6.1 所示，其中 D 表示单元目标和企业目标直接一致，I 表示单元目标和企业目标见解一致。

通过目标矩阵可反映企业目标与单元目标联系的紧密程度，且每一单元目标至少应与一个企业目标有关，反之亦然；如一个企业目标与任何单元目标无关或一个单元目标与企业目标无关，都会反映有异常的业务情况。

(5) 产生一份初始的目标层次表，其形式同如图 6.5 所示。

以下给出关键成功因素的例子。

CSF 是确保企业竞争能力的因素。不同类型的业务活动会有不同的关键成功因素。

汽车工业关键成功因素有节省燃料、减少排放有毒气体、汽车样式、高效供货和成本控制等。而软件企业关键成功因素为产品革新、产品质量、资料质量、市场国际化、强化服务、产品多用、兼容性等。

在相同行业中，会有相同的关键成功因素，如同为软件行业，创新产品可能是共同的因素，但不同的创新方向和内容又可能有不同的成功因素。

同样，在不同的时间段，也会有不同的关键成功因素，这种因素与时间密切相关，如外部环境变化对产品数量或型号的特殊要求。

一般认为，关键成功因素是由企业的关键信息、关键假设和关键决策组成。企业应利用关键的信息，在特定的假设前提下做出正确且关键的决策而保持企业的竞争能力。

关键成功因素可根据不同的管理层次向下分解成部门的关键成功因素，即应根据企业的关键成功因素来制定部门的关键成功因素，一并使部门的关键成功因素支持企业的关键成功因素。

图 6. 5 是一家连锁公司和部门层次图。

以下举例说明公司与部门间的关键成功因素间的联系，第一栏为公司 CSP，以后为部门 CSF。

连锁公司 CSF:

- 收入的全面增长
- 提高梢售和市场效率
- 降低成本提高生产率
- 提高管理水平

...

销售部 CSF:

- 测试价格的伸缩度•判定最优价格
- 增加商品进货渠道和样式
- 减少商店中的偷盗行为

人事部 CSF:

- 雇佣和留住合格人才
- 调动每个人积极性
- 时奖金数量的度量
- 改进和加深培训项目

...

市场部 CSF:

- 扩大基本客户
- 改进运营方式和重点
- 建立战略区域
- 增进和改善客户服务

...

不动产部 CSF:

- 得到较好位置•最好是坐落在拐角处
- 关闭经营不善的商店
- 长期不动产投资的管理
- 完善公司综合预算

...

管理部 CSF,

- 低成本高生产率
- 降低供应和管理费用
- 实现条码扫描和分析系统
- 通过调整达到较高效率
- 减少一线管理开支

...

规划部 CSF:

- 用于资本投资基金的可用性
- 规划与计划的跟踪
- 商品和商店的利润跟踪
- 投资效益的跟踪和管理

...

以下是任务说明书举例。

农机具公司任务说明书

任务：通过农机具的制造和销售•为投资者谋取最大回报

目标:

- 在南方各省占 35%的市场份额

表 6.1 企业目标/组织单元目标矩阵

单元目标	单元 1	...	单元 n
企业目标
XXX	D	...	I
XXX	I	...	D
XXX	D	...	D

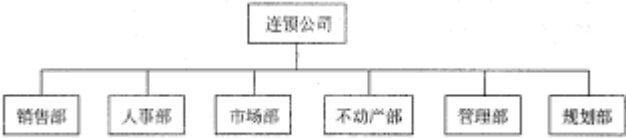


图 6.5 连锁公司部门层次

- 每年销售额至少增长 8%
- 每年利润增长 12%
- 发展和保持小型农机具业务

占总销售额的 20%

关键成功因素：

- 有效管理产品发送渠道
- 提高供货质量
- 提供有效的信息
- 提高客户满意度
- 发挥雇员积极性
- 完成自动生产线

6.4.3 信息需求分析

在了解企业组织机构、任务、目

标和关键成功因素信息输出的基础上，需要识别支持它们所需要的信息，即进行信息需求分析。

在信息工程方法中，将信息需求的识别规范成以下过程。

- **输入信息：**包括书面文件，组织层次图和输出信息。
- **输出信息：**包括信息需求列表，信息需求/组织矩阵，性能度量/组织矩阵。

可分 3 步完成上述过程。

(1) **识别和记录信息需求特征。**其中信息特征包括需求描述、信息使用、所支持的目标和关键成功因素、重要性因素、满意度和需求权值等。信息需求表举例见表 6.2 所示。

(2) **建立信息需求/组织单元矩阵，**其元素√表示组织单元的信息需求，见表 6.3。

(3) **给出评价每个目标完成的方法，**即为性能度量，建立性能度量/组织单元矩阵，其元素表示一个组织单元通过一个或多个性能度量来监测它的目标。

表 6.3 中，为了说明信息的重要性引入了“重要性因素”、“满意度”和“需求权值”3 个概念。前已说明，信息需求是描述一个组建单元为实现其目标和支持它的功能对所需信息的说明。它包括需求描述、信息使用、支持的业务对象、当前环境是否支持需求等。为此设定若干评价指标。

• **满意度：**为每一信息需求分配其满意度(0~3)，其中 0 表示当前环境完全支持该信息需求；3 表示当前环境完全不支持该信息需求。

• **重要性因素：**为每一信息需求分配相对的重要性因素(0~5)，其中 5 表示该信息支持一个关键成功因素；4 表示该信息对实现目标是必要的；3 表示该信息对执行业务活动是必要的；2 表示该信息对目标的实现是有用的；1 表示该信息对其他目的是有用的。

• **需求权值：**需求权值定义为重要因素与满意度相乘的积，作为优先考虑需求的最初因素。

根据以上对满意度、重要性因素和需求权值的定义可见，表 6.2 中“有效的成品数”为最需要的信息。

6.4.4 企业模型的建立

企业模型是企业信息结构的基础，是企业所具有的业务功能和所涉及的主要数据，又称为主题域 (Subject Areas) 的宏观表示。

信息工程方法将建立企业模型的过程规范为以下过程。

- **输入信息：**包括组织层次图、企业年度报告、建立过的任何企业模型等。
- **输出信息：**包括主题域列表、主题域图表、功能层次图等共同组成初始企业模型。

可分 4 步完成上述过程。

(1) **确定业务处理的主题域。**

主题域是企业感兴趣的事物的概括，是业务处理的主要数据，每一主题域可以被分解成与该主题有关的基本数据对象。在建立企业模型的进程中，规划者只尝试确定大范围的概念，并记录每一主题域的名称和简要描述。表 6.4 是一个示例。

(2) **建立主题域图表。**

使用 IEF 中的数据建模工具的主题域图表，可描述主题域及主题域之间的关系。两个主题域之间的联系代表了它们的一个或多个组成部分之间的一种业务关系。主题域图表用包含主题域名的双层方框代表命名的主题域，主题域间的连线代表它们之间的联系。

表 6.2 信息需求列表

信息需求	使用	支持的目标	支持需求的系统	重要性因素	满意度	需求权值
产品的返回率	质量控制	客户满意度提高	库存系统	4	2	8
产品销售点统计	市场调查	确定新市场	无	2	3	6
有效的成品数	预订货核对	客户不满次数	无	5	3	15
区域日销售量	计划进度	提高 3% 销售量	订货系统	4	1	4

表 6.3 信息需求/组织单元矩阵

组织单元 信息需求	组织单元 1	组织单元 2	组织单元 3	...
XXXX	√			...
XXXX			√	...

表 6.4 农机具公司的部分主题域及简要描述列表

主题域	描述
客 户	关于购置产品的所有个人和组织的信息和交付产品及付款方式
产 品	所有已制成和出售的成品
原材料	用于制造产品的部件,包括原材料和预制部件
供应商	原材料供应商
采购员	公司中负责从供应商处采购原材料的人



图 6.6 基于主题域表的部分主题域图

主题域示例见图 6. 6。

(3) 确定高层次的业务功能。

确定高层次的业务功能，记录有关信息，可使用 IEF 中的规划工具箱的功能层次图表工具来建立企业的功能层次图。

图 6. 7 是功能层次图示例，其中每一个圆角框表示一个功能，框中记有功能名。它是两层功能层次图，第一层为企业，第二层为企业的高层功能。

一般的企业业务可划分为 5-10 个高层业务功能，对每一个高层业务功能记录的信息应包括功能名称、内容、有关主题域等。

图 6. 8 是功能描述举例。

(4) 分解成业务过程。

分解高层业务功能为较低层的业务功能，即业务过程，从而产生 3 层的企业功能层次图。按一般规律，每个高层业务功能可分解为 2~7 个较低层次的业务功能。分解一个高层业务功能的基本依据是它对主题域的使用和生命周期。如高层业务功能“原材料管理”可分解为原材料的获取、部件的加工、装配和质检，它反映出原材料

主题域的生命周期。将所分解出的子业务功能加到所产生的企业功能层次图中，即构成 3 层的企业功能层次图，如图 6. 9 所示。

图 6. 9 是以农机具公司为例，以 IEF 工具所控制的 3 层功能层次图。在功能分解时应反映上层的主体要求，不应注重过多细节。

6.5 确定企业信息结构

确定企业信息结构是信息战略规划的任务之一。企业信息结构确定了执行企业业务活动所需要的信息。在确定信息结构时，首先要进行功能分解，确定企业的业务活动，并建立功能分解图；进行实体分析并建立实体关系图；同时应分析和评估企业现有的系统环境，分析现有系统对企业信息结构和信息需求的支持。

6.5.1 企业业务功能的确定

在建立企业模型过程中，确定了企业高层业务功能后，还可将高层业务功能进一步分解为业务过程。一般认为，在进行功能分解中，规划者将业务功能再分解成业务过程，这项任务大约将产生 50-100 个功能和过程。

信息工程方法将功能分解过程规范成以下过程。

- **输入信息：**包括初始的功能层次图、收集相关业务活动信息、企业的组织层次图。

- **输出信息：**包括功能层次图、功能依赖图、业务功能/组织单元矩阵。

可分 3 步进行。

(1) 利用 IEF 中的规划工具箱的活动层次图表表示工具，把功能层次图中所表示的功能继续分解成为更低层的功能或业务过程。

业务功能和业务过程都是处在不同层次上的企业业务活动的集合。其中过程是在较低层上的、具有开始和终结的一系列步骤、有实际意义的业务活动，而功能执行的意义则是体现在一个更高的抽象层上。如“采购”是一业务功能，以名词表示，而“订购原材料”是对“采购”功能分解而产生的一个业务过程，以动词和名词组合表示。

在功能分解的过程中应参考下列分解原则：

- 每个功能至少可分解成两子功能或过程。
- 在同一层次上的成分应属于同种类型，要么都是功能，要么都是过程。
- 同种功能或过程在分解中不能重复出现。
- 组成较高层次的功能的子功能或过程，必须反映较高层功能的所有方面。

(2) 利用 IEF 中规划工具箱的活动依赖图工具构造功能依赖图。

功能依赖图是表示该功能的子成分之间的依赖关系，它指一个业务活动所提供的信息必须作为另一个活动的前提。如功能 A 被分解为 W、X、Y、Z 等成分，则 A 的依赖图则应反映 W、X、Y、Z 之间的依赖关系。

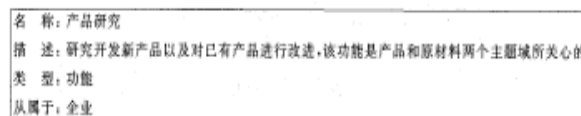
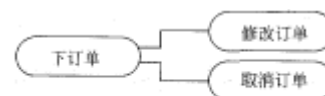
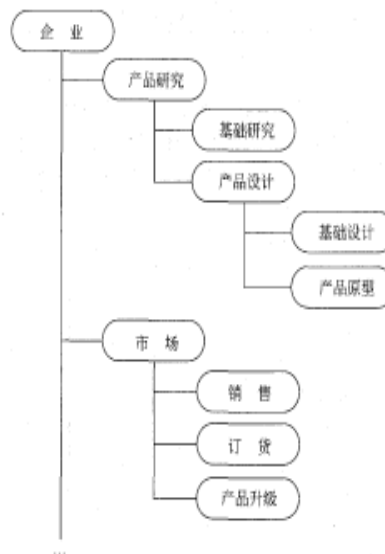


图 6.8 产品研究功能描述



由于依赖分析较为耗时，因此往往加以简化其分析内容，如仅对最底层或大项目中的较复杂的功能进行依赖分析。

通常会用连接两个业务过程的连线表示其依赖关系，如“订货处理”功能被分解为“下订单”、“修改订单”和“取消订单”3个业务过程。则“下订单”就是另外两个过程的前提。“订单处理”功能的依赖图见图 6. 10 所示。

(3) 将业务功能映射到组织单元上，建立业务功能/组织单元矩阵。

业务功能/组织单元矩阵的元素能反映每一个组织单元所参与的业务功能。为了表示组织单元参与业务活动的不同程度，还可以给矩阵元素附以标记值，如 3 表示负主要责任，2 表示负次要责任，1 表示一般参与。

6.5.2 实体分析与实体关系

实体是与企业业务相关的数据的载体，实体分析将产生与企业相关的实体类型以及实体间的相互关系，并建立起实体、业务功能和信息需求间的关系。

信息工程方法将实体分析规范成以下过程。

- **输入信息：**包括主题域图、书面文档、采访记录、信息需求表、功能表、实体类表。
- **输出信息：**包括功能层次图，实体类/信息需求矩阵，业务功能/实体类矩阵。

可分 5 步实现。

(1) 确定实体类型。

实体是数据的载体，实体类型是具有相同实体的集合。在图形表示中一般用长方框表示实体类，而确定实体的方法是：细化初始主题域表从而得到实体类。如对于客户主题域，可细化为相关的实体类型：“客户”、“送货点”、“记账”等项可通过采访进一步获得和确定实体类型，并记录实体类型的名称和定义。

实体类型名称是对企业内部人员的有特定含义的名词，如产品、订单、订单行等有特定意义的实体名，而实体类型的定义通常是使用 1 或 2 个句子来描述。

(2) 定义实体类关系。

实体类间的联系反映实体类之间的关系，由于实体类之间存在着各种联系，因此必须将这些联系加以确定。如客户与订单之间的联系可以是“客户发出订单”，它描述了订单实体类的活动，这两种活动都可视为“关系的成员”，即实体类间的每一个关系都是由两个关系成员组成的。

项目规划中，应确定一个关系的如下信息：

- **关系的名字。**一个关系成员的名字是由一个动词或动词短词构成，由该关系成员的名字连接两个相关的实体类的名，就构成这个关系的名字，其形式为(第一个实体类)(关系成员)(第二个实体类)，例如：“客户”发出“订单”。
- **关系的基数。**基数是关系的一个属性，它说明参与一个关系成员中的一个实体类型的配对数目。规划中，一个关系成员的基数考虑“仅有一个”或“一个或多个”。

例如，从“客户”的观点来看，每个客户发出一个或多个订单；从“订单”的观点来看，每个订单只由一个客户发出。

实体关系的基数可由图 6. 11 表示，其中单线方框表示实体类，而实体间的关系用连线连接。



图 6.11 实体关系的基数表示

(3) 可利用 IEF 的规划工具箱的数据建模工具建立实体关系图，示例如图 6. 12。

(4) 可利用 IEF 的规划工具箱的矩阵处理器，建立实体类/信息需求矩阵，其矩阵元素表示对应的信息需求所要求的实体类。

(5) 记录业务功能所使用的实体类，建立实体类/业务功能矩阵，其元素表示对应的业务功能与实体类的作用，并以指示符区分其作用；用 C 表示创建实体类、D 表示删除实体类、U 表示更新实体类、R 表示读取实体类，并规定其优先执行顺序是 CDUR。

该矩阵应遵循如下规定：

- 每一业务功能至少与一实体类有关。
- 每一实体类必被一业务功能创建。
- 每一实体类至少有两个功能与之相关，其中一个为创建，另一个是删除/读取/更新。

由规定可知：如果一个业务功能与任何实体类都无关，则该业务功能无效；如果一个实体类由多个功能创建，则说明功能分解是按组织结构进行而非按

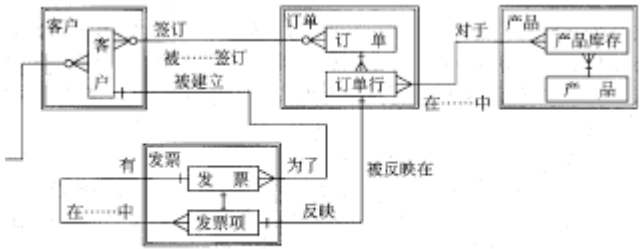


图 6.12 实体关系(ER)图局部示例

业务功能进行。当上述规定被违反，则可能是业务功能不全，或实体类不全，或指示符值不正确，应进行补充和修改。

当输出完成后，应对功能层次图和实体关系图进行审查，以确保能正确反映企业的业务功能和实体类，还需对完成功能分解和进行实体分析任务的其他输出结果进行审核。

6. 5. 3 企业环境评估

规划和创建良好的信息系统是企业构造良好的信息环境。为此，评价当前企业的环境，以确定当前企业的信息系统的地位和作用，并将其纳入系统规划是十分必要的。

信息工程在评价企业当前环境时，将包含以下内容。

1. 现有系统和数据存储清单

将企业现有的计算机应用系统和数据库及文件系统经调查核实后列出清单，规范成以下过程。

- 输入信息：书面文档，包括：系统描述、程序手册和用户手册。
- 输出信息：包括当前系统清单、当前数据库和文件清单、当前系统/数据存储矩阵。

可分 3 步实现。

(1) 确定和列出当前系统清单，包括系统名、说明和状态，其中状态为可运行和仅规划两类，如表 6. 5 所示。

(2) 列出当前数据库和文件清单，确定已使用和仅规划的，记录其名称、状态信息，类似于表 6. 5。

(3) 用 IEF 中的规划工具箱的矩阵处理器工具，建立当前系统/数据存储矩阵。其矩阵元素取指示符值 C, D, U 或 R，表示对应的当前系统对数据存储的作用。检查当前系统/数据存储矩阵，是否发现异常，如是否有数据库或文件不被任何系统所作用，或系统不作用于任何数据库或文件。

2. 信息结构范围

利用矩阵分析确定当前系统和数据存储对企业信息结构的支持，对没有当前系统和数据存储支持的业务功能和实体类应确定其信息结构需求。

确定信息结构的范围可规范成以下过程。

• 输入信息：包括业务功能、实体类、当前系统、当前数据库和文件清单、当前系统/数据存储矩阵。

• 输出信息：包括业务功能/当前系统矩阵，实体类/当前数据存储矩阵，结果评估。

可分 3 步实现。

(1) 建立业务功能/当前系统矩阵，矩阵元素为指示符 X，它表示当前系统对企业的业务功能的支持程度，如 X=3 为强功能支持，X=2 为中功能支持，X=1 为弱功能支持，区分不同的支持度更有利于确定信息结构的范围，表 6. 6 所示矩阵是一示例。

(2) 建立实体类/当前存储矩阵，矩阵元素为指示符 X，它表示当前数据存储含有所对应的实体类的数据。其图示类似表 6. 6。

(3) 分析业务功能/当前系统矩阵和业务功能/当前数据存储矩阵。

在业务功能/当前系统矩阵中，不支持任何业务功能的一个当前系统，即在矩阵中有一空列，它将表示：

- 一项业务功能可能被遗漏。
- 未理解该系统的作用。

在业务功能/当前系统矩阵中，不被当前系统支持的一个业务功能，即在矩阵中有一空列，它将表示：

- 该业务功能在当前信息环境中不被支持。
- 未理解不同系统是如何支持业务功能的。

以上 4 种情况中，除第 3 种情况能给出确定的结论并真实地反映了实际情况外，其余三种都有待于进一步的分析。

表 6. 5 应用系统列表

系 统	描 述	状 态
人事	职工记录和雇佣史	可运行
分发	库存产品选择、包装和分发	可运行
工程信息系统	维护产品材料清单，……	可运行
市场信息系统	提供市场调查资料和客户介绍等	规划
产品清单	仓库存放成品清单	可运行

表 6. 6 业务功能当前系统矩阵(示意)

当前系统 \ 业务功能	实验室管理系统	产品工程化方法	销售和票据	材料需求规划系统	采购接收系统	库存控制系统	运输管理系统	人事记录系统	工资	总账	成本管理	法规系统	设备	不动产	职工培训系统	飞行部件管理	政府合同	食品供应
基础研究	X	X																
产品设计		X																
工程化方法		X																
销售市场			X	X														
订货处理				X								X						
采 购					X	X	X	X				X						
接 收						X	X	X				X						
库 存						X						X						
废品处理						X						X						
分发中心操作						X	X											
包 装						X												
运 输						X	X											
运输管理								X										
总 账				X				X		X	X							
成本核算				X	X	X	X	X	X	X	X	X	X		X			
法律服务														X		X	X	

类似地，对实体类/当前数据存储矩阵可以进行同样分析。对没有当前系统和数据存储支持的业务功能和实体类，应确定其信息结构的需求。

3.信息需求列表

对于所确定的信息需求，应给出每一项的需求程度，并以此作为规划实施系统的优先级依据。

其规范的过程如下。

- **输入信息：**包括信息需求表，当前系统清单。

- **输出信息：**包括新的信息需求表。

可分 3 步实现。

(1)确定信息需求表中的每一项信息需求的满意度，其值从 0-3，其含义如下：

- 0 表示信息需求在当前环境中完全支持。
- 1 表示信息需求在当前环境中适度支持。
- 2 表示信息需求在当前环境中弱度支持。
- 3 表示信息需求在当前环境中全不支持。

(2)定义需求权值。为满意度乘以重要性因素，重要性因素取值范围为 1-5。需求权值反映对信息的需求程度，其值域是 0-15，值越大反映了重要性大而满意度差，如值 15 代表了一个非常重要的信息需求，而当前环境完全不支持它。

(3)产生新的信息需求表，其中记录每一项信息需求的需求权值。

4.信息系统组织评估

评估当前信息系统组织是否适合企业信息资源的管理。为执行信息结构的新规定而给出企业信息系统组织的机构和功能设置的依据。

其规范的过程如下。

- **输入信息：**包括组织层次图、功能层次图、描述信息系统组织和任务的文档。

- **输出信息：**包括当前信息系统组织的 RAEW 矩阵，建议要增加的信息系统角色，建议的信息系统组织结构及目标信息系统组织的 RAEW 矩阵。

可分 5 步实现。

(1)扩展企业的组织层次图和功能层次图，从而确定信息系统的组织单元和业务功能。

(2)建立与信息系统组织的责任(R 人权力(A)、知识(E)和工作(W)相关的 RAEW 矩阵，即信息系统的业务功能/组织单元矩阵，它的元素由 4 部分组成，即其详细含义如下：

- R 表示责任，该组织单元对业务功能的执行责任。
- A 表示权力，该组织单元有执行业务功能的权力。
- E 表示知识，该组织单元为执行业务功能提供的知识。
- W 表示工作，该组织单元实际执行业务功能。

A	R
E	W

深入分析 RAEW 矩阵将可能发现规划中的某些问题，如执行某项业务功能的组织单元，却缺少相应的专业知识，这将反映出系统建设中的存在问题，必须加以修改。表 6. 7 是一个示例。

(3)考虑信息系统组织中需要新增加的角色，如信息管理、数据管理、开发支持、信息中心、通信管理等，它们可以是部门，也可以是人员或技术。

(4)定义一个新的组织机构，并为每一个新的或变化的组织单元定义其职责。

(5)定义新的 RAEW 矩阵，它是在(2)给出的 RAEW 矩阵基础上，加入新的组织单元对信息系统各功能的参与情况。新的 RAEW 矩阵被称为“目标 RAE W 矩阵”。它是一个描述了企业完整的信息系统角色，并能反映出各组织单元的具体业务功能。

6.5.4 现有技术环境分析

分析企业现有技术环境，其目的是检查和评估当前企业已具有的与建设信息系统有关的软、硬件设备的基本情况，并为确定企业的技术结构作准备。其输出结果则给出了对企业当前技术环境的评价。

实现技术环境分析的规范过程如下。

表 6.7 信息系统初始的 RAEW 矩阵

组织单元 业务功能	总裁	副总裁	部门主管	生产部	市场部	内部 顾问组	维护组	网络组	运行部
长期规划	R	A	R						
战略规划			E	W	E	W	E	W	
系统开发				R	A	R			
开发协调				E	W	E	W		
系统组装					R	A			E
市场营销			R		E	W			W
网络管理								R	A
文件/办公管理				A	R			E	W
用户培训					W	W		W	E
运行/网络					R	A			
分布支持					E	W	E	W	
设备管理									R
用户支持									A

- **输入信息：**描述企业内部使用的计算机软、硬件产品的书面文档，描述可能影响技术环境的业务政策的书面文档。

- **输出信息：**技术清单表，硬件设备/组织单元使用矩阵，设备/分布位置矩阵，技术类别分布矩阵，非技术因素约束表，技术环境评价。

可分 4 步实现。

(1) 列出以下企业使用的硬件设备和软件产品的技术清单，并为每一类软、硬件记录名字、位置、每一位置使用数量、获得日期、拥有或租用、有关性能和用法的注释等信息。

- 处理工具(如计算机、外部设备、OS、支持软件)。
- 工作站、服务器和终端。
- 通信工具(CICS 或 IMS/DC, 集线器和调制解调器等)。
- 数据库管理软件(如 DBMS, 数据字典)。
- 软件开发工具(如 CASE 工具、编辑器、代码生成器和动画制作软件)。
- 办公软件(如字处理软件、电子邮件)。
- 决策支持软件(如电子表格、统计软件)。
- 外部资源(如服务台、工具管理)。

(2) **建立硬件设备/组织单元使用矩阵。**在具有高度分散的技术环境中，为了表明硬件设备与使用组织的关系，应建立终端、工作站/分布位置矩阵。反映在不同位置上设备的类型和数量。还应建立一个技术类型矩阵，其元素表示不同位置上的设备类型。

(3) **确定技术环境中的非技术因素的约束，**如政府规定的特指软件，或本财政年度不允许增加新设备等。非技术因素的约束为建立技术结构提供了有用的信息。

(4) **评价企业的技术地位。**评价企业的技术地位有不同的标准，如以下两种标准是可供选择的：

- 按当前信息技术的整体状况衡量。以项目组的经验来判断企业所应用的技术是否最大限度地利用了当前的技术条件并达到了应有的广度和深度。

- 按企业主要业务领域中技术应用的水平来衡量。项目组应尽可能地利用所获得的同行业中其他企业的技术应用水平。

对上述输出信息应由专门的机构和人员进行审查，同时还应征求企业管理人员、信息技术人员的意见，力求客观和准确地对企业的技术环境做出评估。

6.6 确定业务系统结构

确定业务系统结构，给出要设计的应用系统的高层框架是信息战略规划的任务之一。业务系统结构描述支持信息结构所要求的业务系统和数据存储，即数据库和文件。信息工程方法通过对实体类间的亲合度分析和业务功能之间的亲和度分析，识别和确定预期的数据存储和预期的业务系统，确定企业的业务领域。

本节所述有关亲合度分析在本书的第 5 章中有些已有叙述，现仅做扼要归纳，可参看第 5 章的相应内容。

6.6.1 业务领域划分与数据存储确定

通过对实体类/业务功能的 C(建立)U(使用)矩阵的分析、聚合和调整，从而得到企业业务领域的划分；通过对实体类之间亲合关系的分析，组成实体类超级组，并对实体类超级组涉及的业务主题进行适当调整，从而可确定系统的数据存储。

将上述内容的获取规范为以下过程。

- **输入信息：**实体类/业务功能 CU 矩阵。
- **输出信息：**超级实体类组/实体类矩阵。

CU 矩阵中的 C 表示 Create, 即表示由某业务功能建立的实体类；U 表示 Use, 即表示某实体类被某业务功能使用。U 是表示对实体类的读(获)、修改(U), 删除(D)的统一符号。

可分 3 步实现。

(1) 利用 IEF 规划工具箱中的自动聚合软件，自动调整业务功能/实体类的 CU 矩阵，其过程类似于第 4 章所述的 CU 矩阵的建立。从而从初始的实体类/业务功能 CU 矩阵，最后获得一个经过判断和调整后的、业务功能和实体类组合成的初步业务领域划分图，图中给出了业务领域划分和数据流，并可对业务领域适当命名。

(2) 依据业务功能/实体类 CU 矩阵，通过对实体类之间的亲合度分析来确定实体类的聚合，聚合后在一起的实体类组即为超级实体类组。亲合度表示一个实体类与其他实体类的亲合程度。在 IEF 规划工具箱中，提供了计算实体类之间亲合度的算法，可以直接由它来建立实体类/实体类亲合度矩阵，其内容可参考第 5

章有关亲合度的叙述。所有实体类之间的亲合度可形成亲合度矩阵，其元素的值表示对应的两实体类之间的亲合度。若两个实体类之间的亲合度比较高，它们应属于同一预期的数据库；若它们的亲合度很低，则它们不应放在同一预期的数据库中。所有实体类之间的亲合度可用亲合度矩阵表示，其元素的值表示对应的两实体类之间的亲合度，其取值在 0-1 之间。而实体亲合度矩阵可用于把实体聚合成数据库。

(3)利用 IEF 中的规划工具箱，建立实体类组/实体类矩阵，其横行为实体类，其纵列为命名的聚合的实体类组。其元素指明每一实体类所属的聚合实体类组。这些聚合实体类组就是预期的数据库。

6.6.2 业务系统的识别和确定

识别信息结构的业务需求是通过分析业务功能之间的亲合度，从而把业务功能聚合成自然的业务功能组合而实现的。

可将其过程规范如下。

- 输入信息：业务功能/实体类 CU 矩阵。
- 输出信息：聚合业务功能组/业务功能矩阵，业务功能/业务功能亲合度矩阵。

可分两步实现。

(1)对业务功能之间的亲合度进行分析，从而确定业务功能组，即聚合的业务功能。若业务功能之间没有共同的实体类引用，则它们之间的亲合度为 0；若所引用实体类完全相同，则它们的亲合度为 1；其余的情况则在 0 与 1 之间，利用其取值可构成业务功能/业务功能亲合度矩阵。

如果功能之间亲合度较高，则说明它们引用的实体类大多相同，它们应同属于一个预期的系统；如果它们的亲合度极低，则说明它们引用的实体极不相同，则不应在同一预期的系统中。因此，可按亲合度的大小对业务功能分组，分成适当的个数，一般可分成 25-50 个预期的业务功能组。

类似地，利用聚合算法，业务功能按亲合度大小存放，形成聚合的核心。

(2)利用 IEF 工具箱中的工具，建立聚合业务功能组/业务功能矩阵，矩阵的元素指出各业务功能所属的聚合业务功能组。但还能根据规划者对业务的理解去调整聚合业务功能组中的业务功能，使功能的分布符合实际业务需要。所得到的聚合的业务功能组即应为企业预期的业务系统。

6.6.3 业务系统结构图的建立

业务系统结构图用以反映业务系统之间的优先顺序。可根据业务系统处理业务的性质来对预期的业务系统进行分类，以达到修正预期业务系统的目的，并建立业务系统的信息流矩阵，用以标识系统之间的信息流，反映系统之间的关系。

上述过程可规范成以下过程。

• 输入信息：包括业务功能/实体类 CRUD 矩阵，聚合功能组/业务功能矩阵。

• 输出信息：包括预期系统/预期系统信息流矩阵，修正后的预期业务系统。

可分 3 步实现。

(1)根据处理特征，对预期的业务系统进行分类。一般可将预期的业务系统分成诸如战略性类型、规划性类型、控制性类型和操作性类型。建立起系统分类/预期系统矩阵，如表 6.8 所示。矩阵元素表示预期的业务系统所属的系统类型。

(2)建立预期系统之间的信息流。当两个系统所包含的业务功能使用了相同的实体类，就可能存在系统间的信息流。因此可建立预期系统/预期系统的信息流矩阵，矩阵元素即为所在列的系统给所在行的系统提供信息，在矩阵中，横行的系统表示信息提供者，纵列的系统表示信息的接收者(见表 6.9)。例如：订单输入系统的订单也是账目系统开发票的依据，因此账目系统使用了订单输入系统的信息。

(3)人工调整不规则情况，使预期系统成为实际系统。根据经验和系统出现的非正常状况对系统进行人为调整和修正。

6.6.4 确定和组成业务领域

确定和组成业务领域是战略规划中，确定业务系统结构的最后一项任务。它通过建立和聚合预期业务系统/预期数据存储 CU 矩阵，将一些预期业务系统与预期数据存储系统组成业务领域，并与已获得的初步

表 6.8 系统分类/预期系统矩阵

预期系统 \ 系统分类	S ₁	S ₂	S ₃	S ₄	...	S ₁₀	S ₁₁	S ₁₂
战略性	X	...						
规划性	X	X	...			X		
控制性		X	X	...		X	X	
操作性		X		X	...	X		

表 6.9 系统分类/预期系统矩阵

预期系统(接收) \ 预期系统(提供)	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆
S ₁		X			X	
S ₂	X		X			
S ₃				X	X	
S ₄					X	X
S ₅			X	X		
S ₆	X				X	

业务领域相互对照，实施人工调整后，使企业业务领域获得正确划分。

其规范过程如下。

- **输入信息：**包括超级实体组/实体类，预期业务系统/业务功能，实体类/业务功能 CU 矩阵。
- **输出信息：**包括企业的业务领域划分，业务领域/业务功能矩阵，业务领域/实体类矩阵。

可分 4 步实现。

(1) 从识别和确定预期的数据存储中，产生企业初步业务领域划分。

(2) 建立预期业务系统/预期数据存储 CU 矩阵，对该矩阵用识别和确定预期的数据存储方法，确定 8 至 15 个业务领域，并与已产生的业务领域进行相互对照，实施人工调整，从而得到企业业务领域的正确划分。

(3) 建立业务领域/预期系统矩阵，其元素表示预期业务系统所属的业务领域，或该业务领域所包含的预期业务系统，再进行人工调整，使每一个预期业务系统仅属于一个业务领域。建立业务领域/预期数据存储矩阵，其元素表示预期数据库和文件所属的业务领域，或者该业务领域使用了哪些预期数据和文件，并确保属于一个业务领域，从而使业务领域与预期业务系统、预期数据库和文件联系起来。

(4) 建立业务领域/业务功能和业务领域/实体类矩阵，从而得到每一业务领域所包含的业务功能以及所包含的实体类。

对上述输出信息应由专门的机构和人员进行审查，对领域的划分应进行深入讨论，必要时应根据所得信息对所确定的业务系统结构进行修改，使结论尽可能地与审查人员的意见协调后达成一致。

6.7 确定系统的技术结构

确定技术结构，即确定支持企业的信息结构和系统结构所需要的软、硬件及通信的主要技术和产品配置，它是信息战略规划的另一类重要任务。

6.7.1 数据分布与数据分布矩阵

当前企业信息系统的管理方式可能是分散管理和集中管理，它们分别具有以下特征。

1. 分散管理的数据具有的特征

由于企业所具有的某些特殊情况，常会采用分散管理数据，特征如下：

- 企业各部门的地理位置分散。
- 某地区所用数据，其他地区很少或根本不使用。
- 由当地部门负责输入的数据能保证准确、保密和安全。
- 当数据更新频率太高时，采用分散式管理有利。
- 有利于终端用户对特定数据的高效查询。

2. 集中管理的数据具有的特征

特征如下：

- 企业的某些数据有集中管理的必要，如企业的工资单、购货单、总账目等。
- 企业易于保证数据的一致性和完整性，可避免对多个副本进行更新时引起的实时同步问题。
- 数据集中管理有利于提高用户对多种查询时的效率和速度。
- 集中管理有利于提高数据的安全性。
- 大数据量可存放在便宜的外存储设备中，有较好经济效益。

事实上会在同一系统中使用不同的方式对数据加以存储，它们是根据需要来确定的。在实标的存储方式中，第 5 章有关分布数据存储中已经介绍，可能存在 6 种数据的分布存储方式，这里不做详细介绍。

6.7.2 分布矩阵与业务系统分布矩阵

通过建立数据分布矩阵来分析企业数据分布状况，其规范过程如下。

- **输入信息：**组织层次缩排表和支持文本，预期数据存储/实体类矩阵，实体类/组织单元矩阵
- **输出信息：**组织单元/地点矩阵，预期数据存储/地点矩阵和数据分布决策矩阵。

可分两步实现。

(1) 确定每个预期的数据库和文件的地点要求。

需查清哪一地点有哪一组织单元，并记录在组织单元/地点矩阵中。查清哪一地点建立和使用哪些实体类，根据实体类/组织单元矩阵和组织单元/地点矩阵，建立实体类/地点矩阵，表示在各地点所建立和使用的实体类。根据预期数据存储/实体类矩阵

表 6.10 各地点的预期数据存储

预期数据存储 部门及场所	计	预	财	产	产	部	材	公	供	采	材	机	现	设	车	客	销	成	订	支	成	雇	工
	划	算	务	品	品	件	料	共	货	购	料	器	行	工	间	户	售	品	付	付	本	员	资
总公司	C	C	C	C	C			C					C	U	U	U		U	C	C	C	C	
仓库																		C	U			C	C
区域办事处	U	U	U													U	U	U		U		C	C
分办事处																C	C	C		C	U		C
工厂 A		U	U	U	U	C	C	C	C	C	C	C	C	C	C	U			U	U		C	C
工厂 B		U	U	U	U	C	C	C	C	C	C	C	C	C	C	U			U	U		C	C
工厂 C		U	U	U	U	C	C	C	C	C	C	C	C	C	C	U			U	U		C	C

和实体类/地点矩阵，建立预期数据存储/地点矩阵，它的元素表示了各地点是建立还是使用预期的数据库或文件，并以值 C 和值 U 来表示其元素凉口表 6.10 所示。

(2)在预期的数据存储/地点矩阵上，给出数据的分布决策。如表 6.11 所示，其元素表示在各地理位置上的数据类型，M 表示主数据，P 表示分区数据，RG 表示重组数据，T 表示远程处理数据,V 表示变形数据，R 表示复制数据，S 表示子集数据。

6.7.3 业务系统分布矩阵的确定

当企业有多个分布在不同地理位置的场所和部门时，要通过建立业务系统分布矩阵来了解企业的业务系统分布情况。这需要通过建立预期业务系统/地点矩阵来实现。完成上述任务可通过以下操作来完成。

输入信息有组织单元/地点矩阵，预期业务系统/业务功能矩阵，业务功能/实体类矩阵。

可分两步实现。

(1)确定每一地理位置的业务功能，建立业务功能/地点矩阵，其元素指出相应地点所具有的业务功能。

(2)确定每一地理位置上对预期业务系统的要求。利用预期业务系统/业务功能矩阵和业务功能/地点矩阵，确定出预期业务系、统/地点矩阵，其元素反映出对应位置参与业务活动的程度，当取值为 X 时表示主要参与，取值为 \ 时表示次要参与，如表 6.12 所示。

规划者还应对每个参与的每个预期的业务系统进行性能需求分析，为利于确定整个企业所需要的技术支持。

在此任务中，要求前面任务的一些输出信息作为本任务的信息输入。

具体过程可规范如下。

- 输入信息：现有系统评价，技术清单，业务功能/当前系统矩阵，实体类/当前数据存储矩阵，业务系统/业务功能矩阵，性能要求书面文档。

- 输出信息：技术需求报告。

可分两步实现。

(1)对每个预期的业务系统，进一步确认已收集到性能度量时收集的可利用信息，包括：

- 对现有系统的评价。
- 信息技术潜在影响评估。
- 现有技术环境的分析。

其性能度量应包括：几

- 现有和规划项目中的性能需求，如交易数量的平均值与峰值。
- 应用性需求，如联机处理或批处理时间要求。
- 响应时间的约束。
- 安全性要求。

(2)提供与每个业务系统有关的性能需求技术说明，完成技术需求说明书，如包括：

- 联机系统每天的有效时间数。
- 订货处理业务的响应时间等。

6.7.4 技术分配要求的确定

技术分配要求是在对数据分布进行分析后确定的。对数据可以进行定性分析，也可以进行定量分析。

表 6.11 各地点的数据分布决策

预期数据存储	计	预	财	产	部	材	公	供	采	机	现	设	客	销	成	订	支	成	雇	工	
部门及场所	划	算	务	品	件	料	共	货	购	器	行	备	户	售	品	货	付	本	员	资	
总公司	M	M	M	M	M			M				V	M	T	T	M	M	M	P	P	
仓库																M	R		P	P	
区域办事处	T	T	T										R	M	M			T	P	P	
分办事处													P	P	P		R	T	P	P	
工厂 A		T	T	S	S	P	V	V	S	P	P	V	V	V	R		T	T	V	P	P
工厂 B		T	T	S	S	P	V	V	S	P	P	V	V	V	R		T	T	V	P	P
工厂 C		T	T	S	S	P	V	V	S	P	P	V	V	V	R		T	T	V	P	P

表 6.12 部门或场所对预期业务系统的参与程度

部门或场所	工	工	工	分	区	仓	总
预期业务系统	厂	厂	厂	办	域	库	公
	A	B	C	事	办		司
市场分析					X		X
产品范围评审					X		X
销售预测					X		X
财务计划							X
资本的获得							X
资金管理							X
产品设计	\	\	\	\			X
产品定价							X
产品规范维护	\	\	\				X
材料需求	X	X	X				
材料订购	X	X	X				
验收进货	X	X	X			\	
库存控制	X	X	X			\	
质量控制	X	X	X				
生产能力计划	X	X	X				\
工厂调度	X	X	X				
工序设计	X	X	X				
材料控制	X	X	X				
测量和下料	X	X	X				
机器运转	X	X	X				
销售区域管理				X	X		
销售				X	X		
销售管理				X	X		
客户联系				X	X		
成品控制						X	
订货服务				X		X	
包装						X	
发货						X	
贷方和借方	X	X	X				X
现金流通	X	X	X	\	\	\	X
工资	X	X	X	\	\	\	X
成本核算	X	X	X				X
预算计划	X	X	X				X
利润分析	X	X	X				X
人事计划	X	X	X				X
人员招聘	\	\	\	\	\	\	X
劳动保障	\	\	\	\	\	\	X

注：X 表示主要参与 \ 表示次要参与

定性分析是从处理、开发、管理等多方面的因素来对数据分布进行分析，从而确定实现预期业务系统是采用集中方式还是分布方式。定量分析是通过计算放置在不同地点的机器之间的信息流通量，来考虑数据和程序的分布，也称为对数据分布进行定量分析。

确定技术分配要求可规范成以下过程。

- **输入信息：**技术需求说明，业务系统/地点矩阵，预期数据存储/地点矩阵，技术信息文档。
- **输出信息：**有关位置的系统/数据存储矩阵，各地点的业务系统与数据存储间的交互关系矩阵，企业整体网络规划。—

可分 6 步实现。

(1)建立因素矩阵，对数据分布进行定性分析。

因素矩阵的行表示选择集中式或非集中式处理应考虑的因素；因素矩阵的列表示系统开发、系统操作和系统管理所设计的应用项目和数据。不同的企业可以用不同的方法选择其相关的因素，这些因素与系统的功能相关。

如对具有多分支机构的银行的事务处理系统，有的业务要求集中式处理，而有的业务要求分布式处理，因此系统常会采用集中式处理和分布式处理的混合设计，从而系统配置最后的选择不是根据一个业务功能，而是根据多个业务功能进行综合平衡。

(2)对数据分布进行定量分析，合理安排数据和应用程序的位置。

定量分析可更具体地了解各用户地点和数据存放地点之间的数据流通量，通过计算机数据的流通量，可以提供数据和业务过程应如何分布的依据和见解，使用户地点和数据存放地点之间的流通量最小的设计将导致数据存储的分布式；使实现业务过程的应用程序和它们的使用的数据之间的流通量最小的设计将导致在数据存储的集中式。

(3)建立有关地点的系统/数据存储矩阵。其元素表示所在地点的业务系统对相应数据库或文件的使用和创建的情况。

(4)建立有关地点的业务系统和地点的数据库或文件之间交互关系矩阵。其元素标明其使用方式和数量、以符号 I 表示系统交互式使用数据，其数值为交互式使用量、以符号 B 表示系统批量使用数据，其数值为批量使用量，如 15I 或 50B, 15 和 50 即为使用量, I 或 B 为使用方式。

(5)根据上述分析，绘制成各地点的计算机、文件、数据库的组成，反映各地点系统配置情况。

(6)制定出各计算机(主机、客户机、服务器)、各地理位置的业务系统连接成的企业整体网络规划。

企业的整体网络规划是技术结构的主要内容之一。

6. 7. 5 方案的确定与评估

经过分析，规划者应给出一个推荐的技术结构，而最好是给出一个以上的方案供选择，并对方案在技术上、经济上和操作上进行必要的可行性分析。分析应从下述内容进行考虑：

- 给出可选技术方案的成本估算，评价是否符合预算要求。
- 方案对组织或技术变化的相对适应能力。即当企业的业务结构或信息技术发生变化时，技术方案能否适应其变化，以便于系统的更新和升级。
- 方案对企业成功运营的有利影响。
- 方案对增强企业竞争优势和提供新的业务能力的分析。
- 方案存在的风险和规避风险能力的分析等。

应提供所选技术方案的相关文档，它包括技术文档、用户手册、行业评论等。

6.8 信息战略规划报告

信息战略规划报告的形式和提交是信息战略规划阶段的最后任务，是汇总前面 6 项任务所形成的成果。编写出的信息战略规划报告，将完成对企业信息系统的全面规划，并将呈交企业最高管理者，如果被最高管理者所接受，则将成为企业信息系统建设的依据。因此报告是所有前期工作的最后体现。

6. 8. 1 报告的组成和内容

信息战略规划报告的读者首先应是企业的高层管理者，因此规划人员应以他们为对象来编写报告，不能将报告写成一份纯技术性的文件。

一般认为，信息战略规划报告应由 3 个主要部分组成：

- **摘要，**它简要地综述项目的结果。
- **规划，**完整地展示整个规划和基本原理。
- **附录，**包括主要的支持信息。

摘要是从信息战略规划的主体抽取形成的，其目的是回答高层管理所最关心和有兴趣的问题。

摘要通常不要多于 5 页，其内容应涉及下列主题：

- 信息战略规划所涉及的范围。
- 企业的业务目标和战略重点。
- 信息技术对企业业务的影响。
- 对现有信息环境的评价。
- 推荐的系统战略。
- 推荐的技术战略。
- 推荐的组织战略。
- 推荐的行动计划。

其中，系统战略是关于信息结构规划和业务系统结构规划的总结；技术战略是关于技术结构的总结；组织战略是关于信息系统组织进行机构改革的建议；行动计划是指要执行的主要项目，项目的持续时间，硬件设备获得的时间。

规划，即信息战略规划是组成报告的主体内容，它详细说明执行摘要中相关的要点、所使用的表格、图形和插图表达的重要信息。其篇幅约在 40-70 页，不宜过长。

规划其主要内容包括：

- 阐述总体内容。包括规划的范围，规划委托人，规划组成员。
- 业务环境描述。包括企业的任务、目标、关键成功因素、信息需求及组织结构。
- 评价现有信息环境，确定在满足业务环境需求方面存在的问题。
- 通过可选方案和推荐的信息结构、业务系统结构、技术结构，阐明其优点，确定问题的解决方案。
- 最后给出推荐的行动计划。

大部分规划的详细内容包含在附录中，并可考虑是否形成一个用于存放技术信息的信息战略规划技术报告。

6.8.2 规划成果展示

规划成果不仅应形成充实的报告，而且应该利用必要的场合和运用各种手段来展示规划成果，充分体现规划的意义和作用。

通常规划者应该通过准备好的图片、动画等形式，当前最好是利用 Power Point 软件来在演示会上展示其成果，演示会大约一小时，其目的是使最高管理者从思想上认识到信息战略规划成果的价值和意义。

如果最高管理者认可报告，则应明确表示并采取措施将规划成果发布到整个企业；若认为尚存在问题和意见，则规划组应进一步听取意见，并进一步讨论和修正。

6. 9 信息工程方法和环境

6.9.1 方法与工具的结合

由上述各节所介绍的内容可见，不论哪一个环节都是在信息工程方法指导下，并利用与方法论相互配合的、计算机化的工具来实现的。信息工程方法学将方法和工具的发展相互交融在一起，互相支持共同发展。支持信息工程方法各阶段的工具，按照结构化方法集成为所谓工具箱，这些工具箱不仅支持开发生命周期，而且支持规划、分析、设计和构成各阶段的每项活动，设计工具箱与代码生成器的严密集成，可直接生成可执行代码。而按信息工程方法所实现的所有工具箱和用于存储、协调开发信息的信息库，即构成所谓信息工程设施(Information Engineering Facility, IEF)。IEF 是一整套支持 James martin 方法实施的计算机辅助信息系统的工具，它不完全相同于传统的软件工程环境的是，它更强调对数据的规划，或者说它更重视数据资源的规划和开发。IEF 是德州仪器公司的注册产品。

因而，可以认为方法和工具结合所构成的信息工程环境是现代信息工程发展的鲜明特征。工具体现了方法思想和步骤，并规范了人们开发系统的行为，从而可以保证人们能按照既定的步骤和要求来体现方法的规矩，从而减少人们在开发中可能产生的任意性和盲目性。同时能提高开发效率和保证开发质量。

6.9.2 信息工程设施

信息工程设施是为支持信息工程而设计的，它的所有工具和为支持其工具完成相关任务的设施都是依据规范化和结构化的要求实现的。由于 IEF 都是按实现信息工程的理念来设计的，它成为信息工程方法学和支持该方法学的工具紧密结合的典范。

以下介绍信息工程设施的不同结构的设计。

1.知识件工具集(关 Knowledge Ware Toolset)

知识件工具集是由 James Martin 的知识件公司在 20 世纪 80 年代研制的支持信息工程理念的集成化计算机辅助系统工程(CASE)工具，其结构如图 6.13 所示。

知识件工具集具有集成化 CASE 工具的特征，集成化 CASE 由集成化工具箱构成，依据信息工程中执行的功能，按照结构化方式集成，它们包括规划工具箱、分析工具箱、设计工具箱、构成工具箱等，它们支持企业范围的规划、数据模型化、业务过程模型化、系统设计和构成，并能直接生成文档等。这类工具是为信息工程设计的，而不仅是为软件工程师设计的。

在信息工程工具集中，一般都具有存储开发信息和进行协调控制功能的计算机化的信息库。信息库中积累了信息系统的规划、分析、设计、构成各个阶段的相关开发信息，以及系统维护的有关信息，并提供综合信息的工具，是信息工程工具的核心部分 James Martin 在其著作中曾将信息库比喻为 Encyclopedia，即百科全书。

信息库的基本内容包括：

- 数据字典的内容，包括数据字典所描述的数据项的名称、描述及处理过程、变量等信息。
- 信息工程各阶段所产生的各种规划、模型、设计的编码表达式。信息库使用一些工具对其进行检查、协调分析和确认。
- 信息库存储各种图表所表示的含义，以及能够驱动图表或对其修改和维护。

• 信息库存储和使用大量与知识相关的规则，可利用规则来进行推理，并尽量保证所产生的各种规则、模型和设计的正确、完整和安全。

- 信息库能驱动一个代码生成器，并产生代码生成所需要的一些信息。

信息库的结构如图 6.14 所示。

2.Composer

Composer 是德州仪器公司提供的—个基于客户机/服务器结构的信息工程设施，其结构如图 6.15 所示。

Composer 主要由信息库和 5 个工具箱及通信设施组成。

• **信息库。**用于存放所有设计规范说明，这些说明用来灵活地组成业务应用模型。

• **规划工具箱。**用于支持项目规划、软件规划，包括数据模型浏览器、数据建模和数据建模列表以及实体类生命周期图、活动依赖图、功能层次图、机构层次图和矩阵处理器的实现。

• **分析工具箱。**用于支持对规划工作的细化，包括的工具具有数据模型浏览器、支持数据建模和数据模型列表、实体生命周期图、矩阵处理器、活动分解层次图、活动依赖图、活动图、结构图的形成以及一致性检查等。

• **设计工具箱。**用于实现业务领域内的业务系统设计，包括的工具具有会话设计、屏幕设计、窗口设计、原形化工具等。

• **构成工具箱。**根据各阶段形成的信息，构成应用系统，在工作站和信息库的支持下，实现装载模块打包、自标环境选择、生成和安装代码和数据库、交互式图表检查和预定义报表等功能。

• **实现工具箱。**实现远程文件安装，实现平台、数据库管理系统和事物处理器的组合。提供安装工具，在目标平台上执行代码和安装数据库；支持交互式图表检测和在目标平台上执行的程序。

其次 Composer 尚提供实现客户机管理以及客户机管理器与目标服务器之间通信连接的功能。由上可见，信息工程设施的产品是信息工程方法和支持该方法的工具密切结合的产物。它有力地推动了企业信息

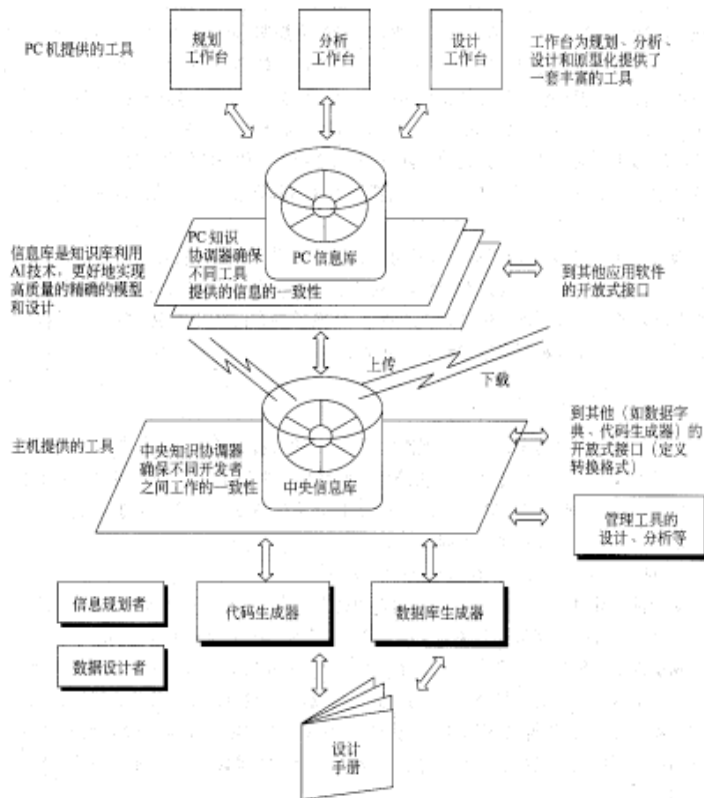


图 6.13 知识件工具集



图 6.14 信息库的结构

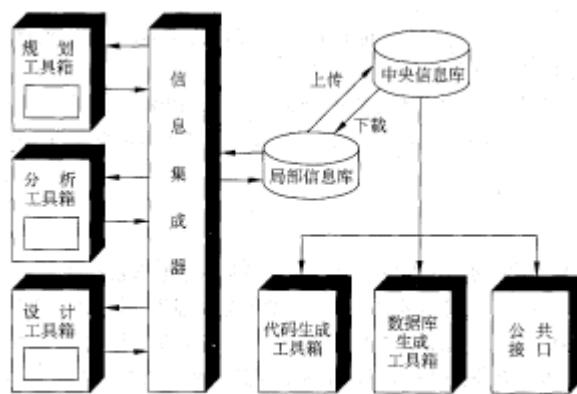


图 6.15 Composer 的结构示意

系统的建立，为企业信息化奠定提供了极好的基础和支持。

除了国外产品外，在国内也有相应的产品开发，如 IRP2000, 它是由大连圣达计算机发展有限公司开发的。它在一定程度上体现了 James Martin 的设计思想和方法。

6.10 小结

信息工程是 James Martin 倡导和实行的二种从事信息系统开发的方法论、工具、环境及理念的集成，它提出了信息系统建设应以数据为中心的基本原理，并指出信息系统的开发必须首先做好规划，并辅以自动化的手段，运用成套、完整的工具，在规范的步骤下去完成开发工作。本章就是按这一思路，介绍了信息工程方法的基本原理和步骤，它是对 James Martin 方法的具体化，但这里并没有十分具体地去介绍如知识件工具集或信息工具设施 IEF，这不是本书应完成的任务。

第 7 章应用原型化方法

7.1 概述

7.1.1 原型化的概念

计算机信息系统的开发通常采用结构化方法，本书的前几章介绍的都是这类方法。结构化方法要求严格定义或预先明确说明用户需求。这类方法试图在系统设计以前，就对应用需求建立一套完备的、一致的和正确的说明，即在系统建立以前，要对系统的功能进行严格的定义或确切的说明。但系统开发的实践表明，尽管在许多情况下用了这种或那种严格定义或预先说明的方法，但当系统建成以后，用户仍然会觉得建立的系统或者是不完全正确的，或者是不完备的。因此，经常要进行反复的修补，而更坏的情况则是推倒重来，这当然是代价昂贵且又令人沮丧的事情。

另一方面，随着计算机应用的普及，应用领域的扩展，计算机价格的下降，用户在不断扩展，不断提出新的应用需求，为了满足这种需求，必须大大提高应用开发生产率，能快速地建立与用户需求相吻合的应用系统，从而防止大量应用问题的堆积。

回顾各种系统开发方法并结合实践经验可知，在开发过程中提高生产率很大程度上依赖于解决需求定义问题。如果用户需求没有分析清楚，系统提供的服务将会受到很大的限制，那样就根本谈不上应用好的设计、测试、复审等技术以及它们能给系统开发带来的效益。

需求定义的一种变通的方法，是获得一组基本的需求后，快速地加以“实现”。随着用户或开发人员对系统理解的加深而不断地对这些需求进行补充和细化系统的定义是在逐步发展的过程中进行的，而不是一开始就预见一切，这就是原型化方法。因此可以认为原型化方法是确定需求的策略，对用户的需求进行抽取、描述和求精。它快速的迭代并建立最终系统的工作模型，它对问题的定义采用启发的方式，并由用户作出响应，是一种动态定义技术。

可以认为，在一定意义上讲，是由于预先需求规格说明的某些缺陷而导致了动态定义技术的形成与发展。

7.1.2 原型化的内容

原型化方法认为，对于大多数企业的业务处理来说，需求定义几乎总能通过建立目标系统的工作模型来很好地完成，而且认为这种方法和严格的定义方法比较起来，成功的可能性更大。严格的定义方法试图仅仅使用描述性的语言和图形文档技术来建立一个最终是完备的需求规格说明，实际上这将会遇到困难。

本章的目的是从概念和使用的角度来介绍原型化方法，首先给出概念性的分析，说明原型化方法的正确性，然后对如何在具体的业务环境中进行原型化提出指导性的建议。

在叙述有关内容以前，先给出常在本章中出现的两个名词的含义。

• **严格定义/预先定义：**指的是一种确定应用系统业务需求的策略，在任何的设计、实现或使用系统之前，预先指出所有的要求。

例如，通过会面、观察、对现行系统和过程进行审查，有关业务方针的研究、发表创造性的意见都可以导出目标系统的一个逻辑模型，而物理模型的提出在分析阶段一般认为是不合适的，应加以避免。目前的一些工具，如结构化分析和定义语言，通常都可用来辅助分析、做文档和表示需求。

• **应用原型化：**指的是完成需求定义的策略。用户的要求被提取、表示，并快速地构造一个最终系统的工作模型并发展此模型。原型法最大的特点在于，只要有一个初步的理解，就快速地加以实现，第一个模型就作为以后各方之间通信的一个基础，从而加深有意义的对话。随着项目参加者对问题及可能答案的

理解程度的加深，模型被逐步细化和扩充，直至系统建成投入运行。

7. 2 原型定义策略

7. 2. 1 需求定义的重要性

过去大多数的数据处理部门都是在一个结构化的开发生命周期环境之下开发系统的，它表述了随着特定开发阶段的展开，逐步实现系统的开发过程，已完成的阶段可以进行项目的审查、控制、管理等。实际的开发步骤、阶段的标志以及交付的文档，因具体实现技术不同可能有差别，但其指导思想和总体目标是一致的。

不管方法的来源如何，所有结构化生命周期法都强调需求定义对于系统或项目的成功是绝对重要的。如果一开始对问题就没有一个清晰的理解，那么就谈不上会有什么效果、效率和好处。

为了进行需求定义，有必要知道下述情况。

- **约束：**业务环境对应用系统施加的某些限制，即预先已确定的接口和像政府这样非公司实体的政策。
- **系统输出：**每个系统输出的定义及其特征。例如媒介、频数、数据元素的内容和保留时间等。
- **系统输入：**每个系统输入的定义及其特征。例如数据元素的内容、来源、数量、频数、保密性考虑等。
- **系统数据需求：**系统中的数据定义以及数据间的关系。
- **数据元素：**数据元素的特征和属性定义，例如格式、名字、同义词、编辑标准和保密等。
- **转换：**旧系统如何向新系统的转换；新系统如何运转起来？如何普及新系统？
- **功能：**系统必须完成的逻辑转换，转换对象和时间，指定系统应完成的确切操作。
- **控制/审计/保密：**系统如何确保性能、数据完整性和操作的正确性、

审计跟踪和保密性。几如何控制系统错误？

- **性能/可靠性：**系统的性能特征是什么？耐故障能力的强弱。

以上提出了需求定义的基本内容，并未包含一切，但也能说明需求定义在本质上是一件严肃而艰巨的工作。

从实用上讲，一般认为，需求定义必须有下列的一些属性。

- **完备的：**所有需求都必须加以适当说明。
- **一致的：**需求之间应该没有逻辑上的矛盾。
- **非冗余：**不应有多余的、含混不清的需求说明。
- **可理解：**参加的各方应能以一种共同的方式来解释和理解需求，需求应是明确可辨的。
- **可测试：**需求必须能够验证。
- **可维护：**文档的组织应该是可灵活修改和易读的。
- **正确的：**所规定的需求必须是用户所需要的。
- **必要的：**需求应是准确和完整的。

如果需求是不完全、不合乎逻辑、不贴切或使人易于发生误解的，那么不论以后各步的工作质量如何，都必然导致一场灾难。因而可见，系统开发中，需求定义是系统成功的关键一步，必须得到足够的重视，并且应提供保障需求定义质量的技术手段。

许多成本分析表明，随着开发生命周期的进展，改正错误或在改正错误时引入的附加错误的代价是按指数增长的。图 7. 1 给出的是一个典型的改正费用曲线的例子。研究表明 60%~80% 的错误来源于定义，见图 7. 2。因此，开发面临的问题是，随着生命周期的展开，不仅发现修改费用越来越高，而且发现绝大多数的错误起源于早期的定义阶段。

由于上述原因，人们对保证提供高质量的定义技术发生了很大的兴趣。开发人员试图用具有完整的方法论的“高效”预先定义技术来确保生成的规格说明是完备的、一致的和正确的。

7.2.2 严格定义的策略

当前，较多的应用韵 1 采用的需求定义方法是一种严格的或称预先定义的方法。从概念上讲，一个负责定义的小组试图完全彻底地预先指出对应用来说是合理的业务需求，并期待用户进行审查、评价、认可，并在此基础上顺利地开展工作。但是这一切都是在尚无使用经验的情况下进行的，而提出的建议又是以图形和叙述性的文字形式表达的。

很多结构化的分析技术，在对未来的工作建立了逻辑上的视图之后，就希望能按图 7. 3 所示的生命周期稳定前进。但实际情况往往与此愿望相反，有很多的项目不像预定的顺序向前推进，还会遇到和经历很多反复，有时在系统测试时，用户才发现与自己的意愿相违背。实践证明，在很多情况下预先定义尚不

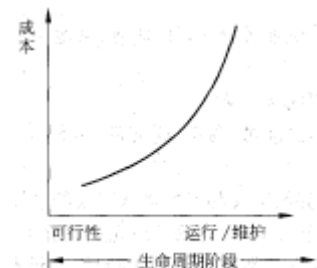


图 7.1 改正-费用曲线



图 7.2 系统错误的来源

能顺利地表达一个有序的生命周期。

严格定义的方法是在以下几个假设的前提下形成的。

1. 所有的需求都能被预先定义

- 个人对系统的认识往往与实际不完全吻合。
- 实地观察和使用系统会刺激用户对系统提出新的需求。
- 观察和经历往往会取消对系统的事先需求。

2. 修改定义不完备的系统代价昂贵且实施困难

上述假设是基于已经说明过的“改正—费用”曲线和“系统错误源”。很明显，它们立足于生产经验和职业实践者的共同意见。然而，现在应该借助于新的软件技术的发展重新评价。新的软件技术使得快速建立和修改应用系统的可能性成为现实。

因此，虽然从历史情况看，软件或系统的修改是困难的、昂贵的、费工费时的，而在当前的软件技术条件下，情况会发生很大的改变，假设的有效性必须重新评价现在的软件技术允许建造软件的“结构玩具”，并允许有效地构造软件或系统模型。

3. 项目参加者之间能够清晰而准确地进行通信

严格定义方法的又一项重要假设是：在系统开发的进程中，项目组、项目经理、分析人员、用户开发人员、审计人员、保密分析员、数据管理员、人际关系专家等都能够清晰而有效地进行通信。

而实际情况往往是复杂的，对于共同的约定，每个人往往会有自己的解释和理解，对规格说明上应该有而尚未有的规定和说明，会有各种意见或加进个人的看法。而文字叙述，如英语或汉语及其他文字描述，并非一种准确的通信工具，即使提供了结构化的文字语言，如结构化英语以及判定表、树等较严格的通信的高级方式，虽然减少了模糊性，但它仍然缺乏“严密性”、“专业性”和“行业感”。

因此，在多学科、多行业人员之间架起通信的桥梁是一件很困难的事。相互间通信的有效性的损失是开发过程失败的主要原因之一。虽然每个参与开发的人都遵从定义报告，但在实际时他们常常会有意或无意地带有个人的不同理解而自行其事。

4. 静态描述或图形模型对应用系统的反映是充分的

使用预先定义技术时，主要的通信工具是定义报告，包括工作报告和最终报告，虽然具体的形式因各自的技术有所不同，但它们的作用是相似的，主要包括以下的内容和形式。

文字叙述：包括应用系统的目__标、对象和其他需求的传统文字叙述和解释的内容。

图形模型：主要适用流程图技术。它表明外部实体、过程和文件之间数据的流动。

逻辑规则：它包含不含模糊性的若干逻辑准则，如判定表等。

数据字典：其内容是系统实体的定义、属性及实体间关系的定义和描述。

所有技术工具的共同特点是，它们都、是被动的通信工具和静止的通信工具，不能表演，因而无法体现所建议的应用系统的动态特性。而要求用户根据一些静态的信息和静止的画面来认可系统似乎近于苛求。

因此，严格定义技术本质上是一种静止、被动的技术。因此要它们来描述一个有“生命”的系统是困难的。理解和评价一个应用系统的最好方式，应该是去体验它，而不仅是去阅读和讨论它。

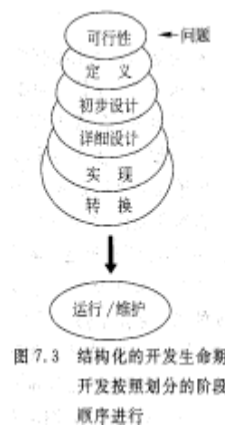
5. 严格方法的生命周期的各阶段的划分都是正确的

许多数据处理人员曾认为，相对于工程学科来说，应用系统和软件的开发更像是一种技术。对于发展“软件工程”原理，把软件开发和应用开发提高到与其他工程原理相同的水平，人们一直在做出巨大的努力。因为部分的工程技术依靠严密的方法论和严格的纪律，人们因而认为软件开发的各阶段也应该是这样。从理论上讲，对实现定义来说“严格”是对的。但可惜的是，在用户的认识上，需求却常常是模糊的。硬性地去坚持要一个只有初步设想的人对需求做出准确无误的说明是不切实际的，由于生命周期的要求，他们可能被迫做出允诺。如果决策是在缺乏充分依据的情况下做出的，则迟早会提出修改的要求。因而在这种情况下，需要以一种灵活的方式来处理不完备的需求，而非一味追求事先的严格定义。

但当你已经知道了要建设系统的需求，就完全有必要进行严格的构造、设计、编码、测试、修改和控制系统的开发。

因此，严格方法的假设可能是不正确的。为合理起见，它必须和其他方法结合起来并加以完善，而试探法常在其他学科中用来验证设想的合理性。预先定义方法应该首先赢得合理性，而不应认为它总是正确的。

综合上述各点可见，严格定义的合理性在许多情况下并不满足，因此建立在脆弱基础上的开发策略在实施中一旦导致系统的失败，决非意外之事。为了更好地处理由于缺乏支持严格方法的假设而给项目带来的风险，需要利用人们多思、向上和喜欢根据经验行事的本性，探求一种变通的方法。



7.2.3 原型定义的策略

应用原型化方法为预先定义技术提供了一种很好的选择和补充。人们对物理模型的理解要比对逻辑模型的理解来得准确。原型化方法就是在人们这种天性的基础上建立起来的，它考虑到用户有时也难免有判断错误，不可能在系统开发过程中提出更多、更好的要求。原型法以一种与预先定义完全不同的观点来看待定义问题。

与预先定义技术完全不同，原型化方法开发策略有以假设。

1.并非所有的需求在系统开发以前都能准确地说明

要想详细而精确地定义任何事情都是有困难的。实际上，用户很善于叙述其目标、对象以及他们想要前进的大致方面，但对于他们要如何实现那些事情的细节却不甚清楚和难以确定。对于所有参加者，建造一个系统都是一个持续不断的学习和实践的过程。当人们仅有局部经验的时候，怎么可能要求人们对全局需求进行叙述呢？

人们认为，数据处理是几个很少的能依靠讨论和灵感来提供模型的学科之一。人们需要在他们做出决策以前提供帮助，最好的帮助就是现实世界的实例，对例子进行研究，然后进行评价。

2.有快速的系统建造工具

直到最近，大系统的原型化才成为可能。当前必要的软件技术产品正在进入市场，使得应用系统得以快速模型化，而且能快速地进行修改。如果生命周期可以浓缩到定义阶段，而且能用易适应的软件加以实现和控制的话，人们就不会被改正费用曲线所支配。

用于完成原型化的较好的工具应包含以下几个部分：

- **集成数据字典。**用于存储所有系统实体的定义和控制信息。
- **高适应性的数据库管理系统。**提供了设计上和存取上的方便，允许直接进行数据的模型化和简化程序开发。

- **非过程的报告书写器。**与字典融为一体，具有非过程化、自由格式和大量的默认值的特征。
- **非过程查询语言。**可提出特殊要求，且能将查询结果保留，并和字典融为一体。
- **屏幕生成器。**描述屏幕的交互机制，自动完成输入编辑，如数据检查、表格检查等。
- **超高级语言。**适用于应用开发的高功能/高默认过程语言。
- **自动文档编排。**提供与数据字典相联系的自动文档化功能。
- **原型人员工作台。**具有交互功能，使用方便，并能产生反馈信息的工作站。

原型技术今天存在于各种形式的开发活动中。如果“原型”可以快速地构造，那么就可以测试一个“好的设想”，如果设想有错，那么就把它丢掉，而不致遭受大的损失。如果设想是对的，就可以进一步求精；而对于想法、概念、观点和要求的正确性，都可以在原型实验室中加以验证，而这一切都需借助于快速生成工具的支持。目前所谓应用生成器(AG)和第四代生成语言(4GL)，都是原型化方法的有力支持工具。

3.项目参加者之间通常都存在通信上的障碍

即使定义很完善的规格说明，不同的项目参加者也会存在或多或少的理解上的差异。而文字性的描述更是缺乏一般工程说明语言所具有的精确性。

而另一种形式是，用户和原型人员基于一组屏幕进行对话和讨论，其方式简单、明确。所有的项目参加人员也可以以一种简明的方式同原型进行通信，从他们自身的理解出发来测试原型。原型提供了一种沟通所有项目参加者的生动活泼的实际系统模型。

因此，排除开发人员通信上的障碍，不是试图将每一个项目参加者都培养成职业的系统定义人员，而是让每个人以一种易于接受的方式去理解规格说明。

4.需要实际的、可供用户参与的系统模型

文字和静态图形是一种比较好的通信工具，然而其最大的缺点是缺乏直观的、感性的特征，因而往往不易理解对象的全部含义。交互式系统能够提供生动活泼的规格说明，用户见到的是一个“活”的、运行着的系统。理解纸面上的系统，操作在机器上运行的系统，其差别是十分显著的。因此当提供一个生动的规格说明成为可能的话，人们就不会满足于一个静止的、被动的规格说明。

因此，当能提供一个活生生的系统模型时，人们对它的了解将比说明性的材料好得多。

5.需求一旦确定，就可以遵从严格的方法

原型化方法的采纳，并不排除和放弃严格方法的运用，一旦通过建立原型并在演示中得到明确的需求定义后，即可运用行之有效的结构化方法来完成系统的开发。

6.大盘的反复是不可避免的，必要的，应该加以鼓励

应该鼓励用户改进他们的系统，改进建议的产生是来自于经验的发展。应该意识到，当把模型展示在面前，由你积极思考去改进一个现有的系统，应该是一件令人兴奋的而不是一件让人厌恶的事情。

在开发最终的需求时，反复是完全需要和值得提倡的，只有做必要的改变后，才可能达到用户和系统间的良好匹配。

综合上述各点可见，原型化方法的假设比预先定义方法能提供更为开明的策略。如果能把原型作为对现实的一个近似的解答而接受，那么就能通过进一步的完善，使得生命周期的费用、实现的进度以及项目的风险达到较为满意的程度。图 7.4 详述了加入原型化策略的结构化开发生命周期方法。它把定义阶段进行了放大，让用户通过一个在定义阶段的小生命周期进行实际的体会。而这种体会对于发现最终的需求是很有帮助和非常必要的。通过正常的迭代而避免非正常的反复；而当定义结束时，产品应该被满足地接受，因为在定义阶段中，系统有关的各方都直接感受和改善了产品。

7.2.4 原型化的优点及其意义

应用原型化是一种系统开发的高级策略，优点如下：

- 原型化方法加强了开发过程中用户的参与和决策。
- 原型化提供了一个验证用户需求的环境。
- 原型化允许生命周期的早期进行人/机结合测试。
- 原型化提供了生动的文档。
- 原型化具有对开发人员和用户的吸引力。
- 原型化提高了人们对系统的安全感。
- 原型化可以接受需求的不确定性和风险。
- 原型化可以缓和通信的困难。
- 原型化可以从个体(树木)和全局(森林)两个方面来观察问题。

- 原型化可以提供很好的项目说明和示范。
- 原型化会简化项目管理。
- 原型化有利于获得开发经验。
- 原型化有利于应用实例来建设系统。
- 原型化以用户为中心来建设系统。
- 原型化提供了建立最终系统的中期训练工具。
- 原型化用有意识的迭代取代了无计划的重复和反复。

以上所述各点反映了原型化方法的内在特性，它

是在计算机技术发展到了某一阶段，用户应用需求高涨的情况下，发展出来的一种方法论，但它同时又是对开发人员有较高要求的一种方法论。

7.2.5 原型化与预先定义的比较

选择预先定义还是原型化方法的决策，不是简单地比较就能确定的，而是要对其基础假设进行评价。表 7.1 给出了每种方法的假设，你认为哪一种更富有吸引力？

总的来讲，支持“严格”的假设是更多的追求理想化，支持原型化的假设偏重于直觉和经验，实际情况是否果真如此呢？也可凭借读者的经验来判断。一般来讲，这常常是对的。图 7.5 说明了一个典型的工程项目生命周期。应该注意到原型化对各阶段的重要性。大部分工程部门都在投入大量资源之前，使用原型化方法来控制风险和检验某种设想的正确性和可行性。原型化把有争论的问题通过实际检验来加以澄清，从而消除系统开发中固有的通信障碍。

7.3 原型生命周期

7.3.1 原型生命周期划分

图 7.6 是一个原型生命周期的示意图，这是一个完整的生命周期模型。

应当指出，当把原型化作为需求定义策略时，它受到结构化开发步骤的限制，根据一般规律，可行性研究的完成应该安排在需求说明以前，在可行性研究中，大多数典型的问题已经被说明，例如当前环境的检查，当前操作问题的分析，业务目标、对象及

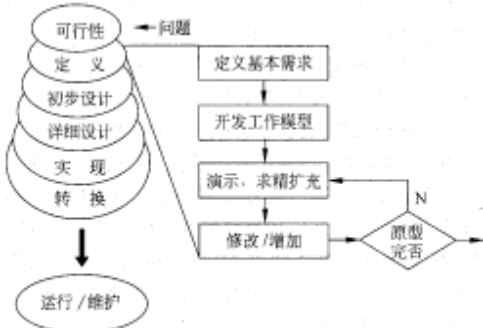


图 7.4 结构化方法和原型法的结合

表 7.1 假设/定义策略对比

假 设	定义策略	
	预先定义	原型化
最终完备的预先定义是可能的	X	
预先定义极其困难		X
修改系统代价极其昂贵	X	
有现行的快速建造工具		X
存在好的项目通信	X	
通信缺陷的固有性		X
静态模型就够了	X	
需要动态模型		X
严格的反复	X	
一旦需求明确即可严格		X
迭代说明定义失效	X	
迭代是不可避免的，必要的和希望的		X

注：X表示选择。

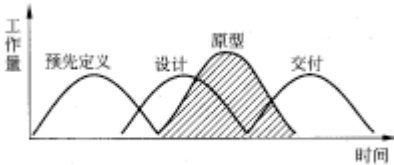


图 7.5 工程项目生命周期，原型的构造和评价是中间的一步

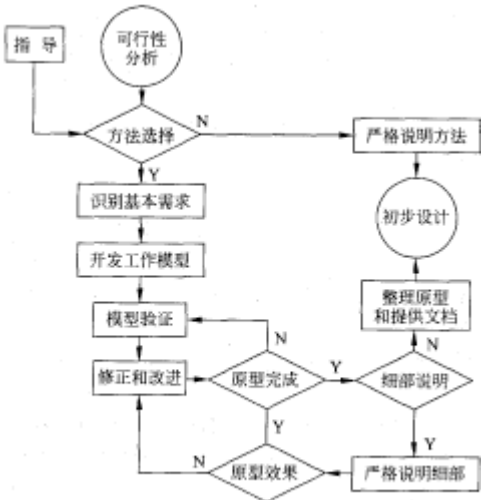


图 7.6 定义应用需求的原型生命周期

通过应用访问的机会，应用方面的主要约束，系统边界及交互点，用户组织及项目代表，成本及利润的目标，系统规划中应用的地位等。

这些内容对原型开发人员逐步适应用户环境是有价值的。即使可行性分析已经完成，大部分更详细的信息也要在用户需求分析中细化、具体化。

原型生命周期的出口端是初步设计，在预先说明的环境中，初步设计起始于一个物理解决过程。而对于原型法，初步设计提供分析和解剖原型的功能。一般不能期望直接完成原型，因为一个实际生产的系统还有很多需要未被满足，诸如系统性能、恢复机制、操作文档、辅助变换和质量控制等。可以把原型扩展成实际应用，也可以把原型作为一个需求文件为新系统服务。

原型生命周期由 10 个步骤组成。这个过程的目的是提供一个既与用户需要也与开发者需要相联系的需求说明。生命周期把建立大的系统模型作为目标。对于过程的每一步，将陈述它的目的、描述其活动并给出出口标准。

1.合适的(好的)选择

判定应用系统是适合于原型方法还是预先说明方法。可以从系统结构、逻辑结构、用户特征、应用约束、项目管理和项目环境等多方面来选择。当确定原型方法是最合适的开发方法时，即可退出。

- **系统结构**：联机事务处理系统、相互联系的应用系统适合于原型化；而批结构，如批处理、批编辑和批修改等结构不适合于原型化。

- **逻辑结构**：结构化系统，如操作系统、文件管理系统和管理信息系统等适合于原型化，而基于大量算法的问题不适合原型化。

- **用户特征**：不满足于预先说明方法的经验；愿意为定义和修改模型投资；难于肯定详细需求；愿意接受制定决策的职责和能够并准备积极参与的用户是适合于使用原型化方法的用户，相反则是不适合的。

- **应用约束**：对已经运行的系统的补充不适合于原型化。

- **项目管理**：项目管理者愿意使用这种方法的才适合于使用原型化方法。

- **项目环境**：需求说明技术应该根据每个项目的实际环境来选择。

理论上讲，对上述所有因素都应该考虑并得出结论，并对复杂情况做出权衡。

2.识别基本需求

识别基本需求，以便能够设计和建立初始模型。如果在这一步之前进行了可行性分析，则可马上获得大量有价值的信息。

发现需求没有捷径，必须对当前系统进行调查，与用户交互、做业务性研究等，尽管发现需求是一件困难的工作，但还是逐步形成了一些有效的方法。传统的需求调查方法和本章讨论的方法都是可供选择的方法。

原型化方法与传统分析方法的主要不同是：它既不必是完整的也不必是完善的，而只是一种“好设想”。需求分析的目标是为初始模型搜集大量信息，只有充分积累，才能建立第一步的模型，即系统的简化原型。

用户需求的初始确定对生命周期的成功是至关重要的。一般认为，低于 60% 准确性的初始模型是令人失望的，也会打击用户对这种方法的热情。迭代只是用来改善和完成修改。应用原型化不能作为一个建立初始模型前而逃避有意义的分析的借口。

3.开发工作模型一

目的是建立原型的初始方案。必须提交一个有一定深度和广度的工作模型，以便进行有意义的讨论，并从它开始迭代。一般认为，提交一个有多种功能的简单屏幕比一个只有很少功能的完善工程文本更有意义。首先必须证明对应用问题已有一个比较完整的理解。在迭代中屏幕和报告是系统改进的基本动力。

初始模型的质量对生命周期的其他步骤的成败至关重要。如果它存在明显的缺陷，就是一种不好的设想。如果为了追求完整而做得太大，它会不易反应，而且对其中的一些不好的设想要进行大量的修改；如果模型是应用的核心部分，那么迭代将从一个优秀的初始模型开始工作。

提交一个初始模型需要的时间随其规模的大小、复杂性、完整程度而不同。3-6 周提交一个系统模型应该是可能的，这样既有足够的时间开发富有意义的功能，又能保持用户的兴趣。最大限度不能多于两个月，两个月后提交的应该是一个系统而不是一个模型。

原型工作人员应由 2 人(可增加处理补充支持功能 1 人)组成，甚目的是减少通信障碍。

4.模型验证

目的是验证系统模型的正确程度，进而开发新的并修改原有的需求。它必须通过所有有关人员的检查、评价和测试。

为改进和验证模型，开发者应积极地鼓励所有的评论者，充分解释所完成模型的合理性，但不要为它

辩护。它应该在交互中达到完善。

在迭代的初期:

- 模型通过用户进行验收。
- 总体检查,找出隐含错误。
- 在操作模型时,使用户感到熟悉和愉快。

在迭代的后期:

- 应发现丢失和不正确的功能。
- 测试思路和提出建议。
- 改善用户/系统界面。

提交了完整的模型并不意味着系统已成功。即使开发过程完全正确,但实际上用户还可以提出一些有意义的修改要求。这不能看成是对开发者的批评,而应看成是开发过程中一种自然的现象,原型化的目标是鼓励改进和创造,而不仅是保持某种设想。

5.修正和改进

要使原型与用户的修改愿望协调一致。作为前一步的结果,大部分修改功能是所要求的。当发现严重的理解错误,使正常操作的应用系统与用户愿望相违背时,产生废品的可能性也是存在的。但大多数原型(并非全部)不合适的部分都是可以修正或作为新模型的基础。如果发现是废品应该立即放弃,而不能继续凑合。

更多的情况是在现有的模型基础上做进一步的改进,这就要求控制随之可能引起的积极和消极的影响。必须有一个字典,它不仅用以定义应用,而且必须记录系统成分之间的所有关系。对于原型化软件提供管理开发过程有效的集成化字典是一项关键的软件需求。

在一般情况下,特别是用户积极参与的情况下,应保留改进前后的两个模型,这样做的好处是,不仅当用户需要时易于退回,而且并存地演示两个可供选择的对象是帮助决策的良好方式。

6.判定原型完成

判断有关应用的实质是否已被掌握,这个重复周期是否可以结束?

对于模型来说,每一个成功的改进都会促进模型的进一步完善。实际上模型就是描述功能和对最终系统的展示。

判定结果可能有不同的转向,继续验证或进行详细说明。

7. 判别细部说明

对原型组成成分的说明。组成原型的细部是否需要严格地加以说明?原型方法不排除对系统必要成分进行严格和详细的说明,如将需求转化为报表、给出统计数字。那些不能通过模型进行说明的成分,如果有必要的话,必须提供说明,并借助于屏幕来进行讨论和确定,当各种成分都被说明后,即可退出。

8.严格说明细部

对已提交的需求说明定义它所有严格说明的成分。

不能通过模型说明的所有项目,仍需通过文件说明,这些较明显的项目有系统的输入人、系统的输出、系统的转化、系统的逻辑功能、数据库组织、系统的可靠性、用户地位等。原型化有助于完成严格的需求说明。如输入、输出记录都可以通过屏幕进行统计和讨论。

严格说明成分要作为原型法的模型编入字典,这样将给开发过程提供一个统一的连贯的需求说明。

9.判定原型效果

考察由严格说明成分附加的信息是否会使模型失效。

如果新加入的成分导致模型部分失效,则不应使模型进入初步设计。

如果模型存在问题,应对附加成分进行修改,建立一个新的模型,使其满足用户需要。

除非处理一个特别大的或极复杂的应用系统,一般严格说明成分不会从根本上影响模型的有效性。

10. 整理原型和提供文档

整理原型和提供文档是把原型整理编号,为下一步的开发服务。原型化方法像其他的任何软件系统一样必须有文档。当然原型软件的初期需求模型就是一个自动的文档。

原型化方法生命周期提供了一种完整的、灵活的、近于动态的需求定义技术,它具有以下特征:

- 综合了所有提出的必要的需求,建立原型就近似于预先需求规格说明。
- 模型能进行必要的裁剪和组织,以接近目标系统。
- 综合用户、项目经理、原型开发人员的各方需求。
- 原型化方法也是有序的和可以控制的。

通过原型生命周期所提供的技术和方法,能使业务需求定义合理化,原型化方法通过动态演示可使以

用户为中心的需求得到检验和认可。

7.3.2 原型化的准则与策略

原型化是对应用开发的一个挑战，如果每个原型都单独建立，那将需要做大量的、重复的工作。幸运的是已经总结出一套基本的准则和策略，它们有助于整个原型开发过程。系统地运用这些准则和策略，对于大多数原型化过程只需分析应用的一些特殊部分，而多数功能、结构和用户界面能从其他模型得到借鉴和重用。

原型化的准则提供一套原型开发的思想方法。原型化策略提供一系列原型开发的有效方式。它们较系统地阐述了原型的建立方法和操作指导。

7.3.2.1 原型化的准则

下列准则能应用于原型化过程。

1. 大多数的应用系统都能从一个小的系统结构集合导出

大多数的传统的业务应用系统是从几个基本系统结构中导出的。虽然外部特征易于掩盖其共性，但许多常规的系统已被构造出来无须从头开始去完成。

熟悉系统流程图的开发人员会发现，经常有若干功能相同或类似的部分在图中重复出现，而这些具有共同性的内容，对于原型开发者将是一笔重要的财富。它们是构成系统的基本要素。

可归纳成以下 8 个基本的模型结构。

- **成批编辑/修改**：把用户输入汇集成批，定期输入给系统。
- **成批生成报表**：定期以成批方式从数据库中产生标准的和(或)非标准的报表。
- **成批转换**：一批程序定期应用多种转换逻辑去修改指定的数据库。
- **成批对接**：定期产生于系统之间的一个或多个输入/输出的成批对接。

• **联机结构化的修改/查询**：定期产生于用户和系统之间的事务。

• **联机特殊查询**：系统处理的随机性的特殊查询。

• **联机界面**：定期产生于实时情况下的应用之间的一个或多个系统对接。

• **联机报表生成**：在对事务的响应中，一个报表或者立即打印，或者推迟成批打印。

图 7.7 综合上述基本结构为原型化提供了一个示范的选择。

2. 多数系统使用一个常用和熟悉的功能集合

有一个基本功能的共同集合，作为一个规格化的子集，经常出现在大多数传统应用系统业务中。参考一系列成功的模型功能后，发现问题可归纳为：

- **确定应用系统需要的基本功能**，再分析应用系统的个别差异。
- **分析应用系统中不常用的功能**，当在初始模型中为一热点时，这些功能可放在迭代阶段来完成。

大多数应用系统包括的基本功能有：对数据库记录的增加、删除和修改；对文件的显示、浏览和查找。

以上功能都是实现应用系统的基础，尽管应用系统间就上述基本功能而言还会有细小的差别，但这些差别就是要等待讨论的和处理的部分。

先提交一个系统的核心或框架，尽管它还不成熟，但仍然有利于用户需求的讨论。

3. 大多数的输入编辑能从一个小的编辑模型集中导出

有一个熟悉的和可重复使用的公共的编辑集合，它用于确保系统输入的合法性，因此上述问题可简化为：

- **识别每个输入所需要的一般编辑的子集。**
- **识别应用需要的特殊编辑。**它可以留待以后的迭代中来完成。

所有应用系统中数据的输入需要恰当的编辑，可以归纳出一部分常用的一般编辑的作用和功能，但应用系统中对编辑的特殊要求就可能很复杂。大量的布尔逻辑和算法对于特殊情况的应用系统可能需要提供特殊的环境。

建立参考编辑模型应具备以下条件：用户的不含混的提问，用户对编辑的推动，用最少的说谦明去初始模型建立工作外壳的能力。

4. 基于一个 4 步的报表模型生成应用系统的报表

从数据库生成报表的 4 步过程为：

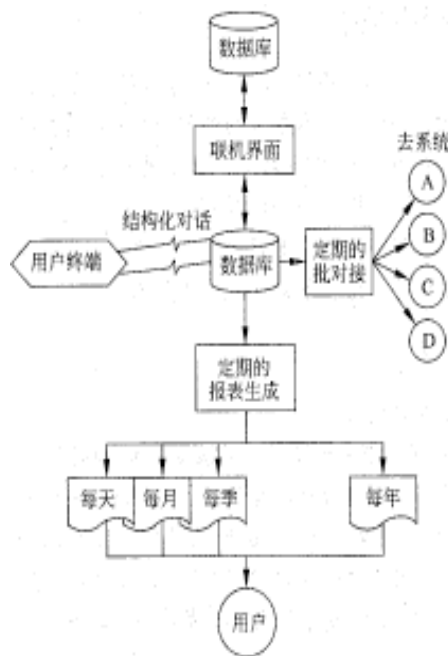


图 7.7 有助于原型化的系统结构

- (1)从数据库选择和拆卸数据。
- (2)按说明分类每个报告。
- (3)为了打印定格式和编辑数据。
- (4)打印该报告。

数据库中的数据，可以通过一个非过程化的报表生成器，由用户参与来生成批处理报表。如果提供了查询语言和对用户友好的报告书写器，大多数的报表生成工作可以由用户来完成。

原型化的报表生成可划分为两类：生成报表是数据库推动的；生成报表可延迟到以后的迭代中。

5. 有一个“正确”的设计结构集合，对原型将会产生积累作用

实践证明，正确的设计实践对一个广泛的应用范围都会是有效的，因此从这种意义上讲原型化是“增值的”，而不论其需求是否明确，随着原型化进程原型将得到积累。

以上开发原则提供了一种基本想法，即快速原型化过程是有依据的和合理的。如果每个项目或系统都是以个别的方式从头调查，而又要快速地建造一些不熟悉的系统，这几乎是不可能的。上述事实总结并提供了这样一种看法——在原型开发中，原型工作者只是在剪裁和粘接那些已经多次建立过的东西。虽然从用户的观点看，那些系统似乎都是各异的，较少共性，但这是从表面的功能观察，而有经验的模型开发人员都能找出系统的基本功能和共性，从而利用他们以前多次开发过的模型进行剪裁和粘接，并进行必要的增补，从而快速地建立一个新的模型。

7.3.2.2 原型化的策略

下列策略能用于快速建立原型及原型改进。

1.用第三范式规范数据，建立应用系统的数据模型

由于许多应用系统是由数据驱动而不是由过程驱动的，一旦了解了数据，也就可能了解应用系统需要的功能。而组织、分析和模型化应用系统的数据的最好理论依据是关系数据理论，特别是有关第三范式(3NF)理论，它不仅能保证数据库建立的合理性，而且从规范化的结果能推导出很多需要的应用系统。

2.大多数富有成效的建立模型的途径是利用组合工程

导出或得到一个系统需要的实体的大多数的有效途径是：

- (1)利用一个已存在的实体。
- (2)全部从已存在的系统实体中装配此实体。

建立、管理和重组各个成分的能力是检验产品生产部门生产水平的标准。在原型开发中一个重要的指标是：对已开发的原型成分的利用率；建立原型的新的部分时，其中可复用成分的多少。

通过装配而不是新建提交一个现实的模型是一个原型开发策略。建立各组成成分时，不仅要考虑速度，而且要考虑减轻测试、文件说明、变化控制等因素所给与的负担。

原型化涉及系统速度、灵活性和变化。为了完成需要的高效率，最有效的途径是组合而不是建立，组合工程是建立成功模型的核心。

3.最有成效的建立模型的途径是“剪裁和粘贴”

如果通过组合还不能得到一个新的成分，那么最好的策略是利用一些基本成分，通过对它们的“剪裁和粘贴”而形成新的成分。

“剪裁和粘贴”对大多数数据处理实体是适用的。新记录通过删去现存记录的元素而建立起来。新模块通过有选择地从现有的模块拷贝某些代码来建立。通过拷贝建立一个有标准题头、结尾和项目格式的模式屏幕，对列表项目系统，要建立两个模型屏幕：一个关于菜单的和一个关于“干活人”的屏幕。

实施“剪裁和粘贴”应有两个必备的条件，首先，组合工程的基本原理必须实施，大的相互牵扯的结构对“剪裁”是不利的；其次，一个好的编辑器能允许开发人员并行操纵多个工作空间。

4.用系统举例

通过演示实际系统，让用户浏览实例所提供的功能后再来决定需求。最好是在分析阶段让用户有机会去浏览所想看到的原型文件。通过与用户对话的方式来确定系统功能的取舍。这种做法不但用户易于接受，而且在初始模型建立之前，开发者就有机会体会用户的需求倾向。

数据处理是不易展示产品功能的行业之一。它不像其他产品可以通过商店来陈列产品，通过橱窗进行展示。

实例显示是沟通产品和用户的一种好的通信方式。如果能通过已有实例，即已经开发好的原型来向用户展示，就不必再依靠用户的想象力去建立一些初始的简单的模型。而且要邀请用户参与对实例的讨论。

5.字典驱动的软件结构

使用基于集成的和灵活的数据字典的开发结构，是保证系统开发速度和适应性的主要途径。字典驱动的结构不仅提供了非冗余的全部系统的成分说明，而且字典对应用的内部模型化，自动化地为大多数的原

型提供文件和材料。

高适应性“结构玩具”软件的有效性是执行需求定义并使原型化成为正确策略的一个前提条件。如果你是去建立高功能 / 高复用的“结构玩具”，就必须有一个用以定义、维护和定位它们的数据字典。

一个数据字典是定义了一个系统的所有成分的仓库。它保存应用系统并将其模型化。软件字典的一种特殊类型是灵活的和被集成的数据字典。集成性的含义是指建立系统元件的所有开发工具都是以数据字典为来源而得到其他构件/部分的。灵活性的含义是指当字典建立时，它能记录系统所有部分之间的关系。

一个基于既灵活而又是集成的数据字典的软件结构为原型人员提供了一个完整的记录管理系统。所有的系统实体和系统间的联系都被存储在一起。高度的可重用性和文档的自动生成就是自然而然的事了。

6.文档的自动化

虽然原型没有事先的文档化，但如果它是可变化的和对实际开发人员有价值时，亦应给与完整、一致和准确的说明。提供这些特征的惟一方式是使用原型来编排文档，以自身作为文档的来源。

原型人员要在建造模型的同时维护所有系统构件之间各种联系的记录是不可能的。然而这些文档对于迭代过程却又有重要作用。为了解决这个矛盾，可以将原型视为一个应用数据库，从它机械地导出完全和准确的文档。

7.小的原型化队伍

原型不能由一支大的队伍来建立。不论应用规模的大小，最多 3 个人组成一个原型化小组，而更可取的却是由两个人来组成。当开发组超过 3 个人时，在快速开发环境中，要做到快速、目标和观点一致、保持良好的通信并且低管理费用是很困难的。

随着人员的增多，则需要正式的辅助文档资料，确定检查点，进行研讨以及确定对管理大型项目的必要的控制技术。两个人是最理想的编制，便于通信、便于统一目标，可使用非正式文档说明，而且事先无需进行精确的项目管理。

8.交互式原型开发者工作台

原型开发人员应能在一个交互的和综合的工作台上建立模型。所有的软件成分必须在一台交互式 CRT 上完成和执行。这样的装置提供了快速反馈和继续活动的功能，二者对于快速开发无疑都是重要的。

9.陈述性规格说明

软件功能需求提供的两种方式如下：

过程性说明。给出过程码，它精确地说明如何去做这些功能。特点是符号多、耗时间、易出错，为了改正要做广泛的检验。陈述性说明仅要求去说明需求。如果出错，则只需声明改正。

陈述性说明。一个语句说明做什么，软件会决定怎样做。陈述性说明为厚型化人员提供了工作上的方便，有较高的开发效率，无需为某种需要去书写代码，只需做一定说明，所有的逻辑需要都能自动完成。从开发效率上看，陈述性的说明比过程性说明好。

10.终端用户报表生成器

原型的大多数编辑/修改部分需要开发人员亲自组织，但对报表生成却并非如此。随着报表生成工具的完善和友好性的提高，终端用户对亲自完成报表的可能性和兴趣在提高。

愿意并有兴趣的终端用户能自己建立报表，这样就减少了开发人员的工作量，也增加了用户对模型化过程的参与和通信。而最重要的是省去了用户与原型化人员间的翻译工作。

11.专业原型化人员

为了快速、保质地开发原型，获得与“好的设想”吻合的模型，需要谨慎地挑选和发现专业原型化开发人员。他们应该对一个完整的生命周期有了解，受过训练，胜任原型化结构，是为提交高质量产品有责任心的人。不能在原型化过程中采取毫不在意的做法，这样在碰巧的情况下，也许能通过模型；而在糟糕的情况下，可能对开发组织的信誉和用户的信任等方面造成很坏的影响。

12.开发人员参加原型化

提倡将要完成实际系统的人员参与模型的开发。他们现在了解得越多，对他们工作中所用到的文件就会越清楚。这也是建立实际系统之前让开发人员浏览整个过程的好机会。理想的情况是开发人员作为原型开发人员的一部分参与系统的开发。

以上 12 条策略提供了指导原型开发的特殊方式。快速而精确地建立模型决不是偶然的奇迹。它是遵从减少错误、提高生产率的一系列特殊策略的结果。在明确地应用了原则的思想后，就能探索出成功建立模型的途径。所以在完成一个大系统的原型时，这些策略一定会被用到和被实施。

在研究了原型化的准则和策略后，我们知道建立模型是一个物理过程，而不是一个逻辑过程。带着这些思想进入原型生命周期，可以很好地处理物理建造问题，把知识和技术集中在分析例外和个别特性上。以这种方式指导问题的解决，方有可能提供高质量的初始模型。

7.3.3 混合原型化策略

原型生命周期提供了一种用原型化完成需求定义的完整的方法。但对于一些有特殊要求或特殊情况的应用，如规模较小、完整性要求较弱的应用，为了获得较大的效益，可以采取灵活的做法，以适应实际目标。结构化方法的整体性要求往往对解决一些待定的问题缺乏灵活性。因此原型化方法应该既保持其特点又具有相应的灵活性。

已介绍的原型生命周期意味着对自身的以下若干约束：

建立一个完整的模型。

原型人员要建立初始模型。

原型化要从定义阶段开始。

实际系统将用自家的资源来建立。

下面是一些可供选择的方法，它们改变了上述某些约束。

1. 仅对屏幕的原型化

提供原型生命周期的目的是提交一个有内容的工作模型。这项工作当然包括修改、恢复数据库中的记录。实际上应用文件的定义、建立、修改都需要时间和工作量。

一种可选取的方法是仅用模仿系统屏幕的办法来约束模型。屏幕程序为演示每个屏幕以及每个屏幕间的交流做好准备。

如果关心的重点是用户/系统的交互界面，原型法可以形成出色的表演。如果用户的基本考虑不是系统的逻辑操作，而是系统的外部友好性，使用屏幕的方法比做一个完整的模型更灵活、更有效。

当然，要考虑这种局部原型化方法的局限，当只是一部分问题通过原型化被检验时，其余的部分将做适当的考虑。

2. 使用购买到的应用系统作为初始模型

提供的原型生命周期要求原型人员建造初始模型。而对许多应用系统，好的模型已经存在。软件商提供了广泛的商业应用系统，它们完全可以视为内容丰富的模型系统，从它们来扩展符合实际需求的系统。

应用软件的成功直接依赖于它的模型特性，而用户在购买之前应先检验和实践其功能。虽然这些应用软件可能对现行执行系统不是最好的，但它们作为一个初始模型。迭代法不仅仅可基于自家开发的模型，也可以基于软件商的产品。

3. 可行性分析中的原型化

定义含混的问题不仅发生在需求分析阶段，在可行性分析阶段同样可能出现。由于原型化需求依赖于可行性分析阶段的有关信息和文件，因此在这个阶段做有限的模型化可能是有用的。

在这个阶段运用模型，不仅可以提供一些基本情况，而且也有助于部门领导决定是否通过此项应用。大多数企业有指导部门，他们必须决定哪个被推荐的系统可以在给定的有限资源下进行开发，识别每个参加竞争的应用系统的潜在利益。

4. 子系统原型化

不同的问题有其特性。如有的问题规模很大，要 80-90 个屏幕。而有的问题其中某些部分带有风险性。预先说明的方法考虑了如何通过分解技术去和一个大问题打交道，把分解后的局部/子系统单独解决。

原型化方法自然也能用同样的策略。如果问题太大，以致于开发人员在一定时期内解决问题太困难，则可将问题分解为多个原型。如果只有一个特殊的子系统是带有风险性的，也可以只限定在那个子系统上，而对其他逻辑清晰、人/机界面友好的子系统可做屏幕原型化开发。

分解技术适合于原型开发策略，也适用于每一个分散的单位。

5. 原型与需求建议

很多公司使用非标准的开发资源作为他们开发过程的一部分。原型能在建立需求建议中给所有的参加者很大的帮助。

原型一旦建立，即是一份说明文档。目标系统的建立还没有完成，需要进行内部的开发。一个需求建议要包括对于原型的调查、检验的成本和时间。另外，可考虑购买商品软件，把它与原型进行比较，从而确定它是否可以作为工作模型提交给用户，一般工作模型可以给用户比文件更多更好的建议。

6. 最终用户进行原型化

对于某些应用，最终用户在得到必要的帮助时可以自己开发原型。对于原型开发人员，大多数的困难是随着系统的扩大，结构变得复杂。然而，很多应用系统基本上是由结构化的特定报表组成，如果给出强有力的报表工具，那么最终用户自己来开发模型是可能的。这样能减少用户和开发者之间的转换工作，并且把责任落实到他应有的位置上。当然，这个过程需要用户愿意并能够承担。对于具有复杂内部结构、可应用多种工具或通过多种途径去完成的模型开发，最好是由专业人员来做。

如果结构合适,由最终用户来开发系统效果会非常好。如果用户乐意且能够做这工作,让用户亲自完成模型对开发一个真正用户满意的系统是很有效的和有益的。

综上所述,可以认为原型化方法应该是一类具有灵活性和适应性的方法,能适应不同条件和情况的需要。一种不能适应环境的方法可能对开发不是一种帮助,而是一种负担。

7.3.4 原型的实施

只有在数据处理的领域中,才会提出原型是否可能并应该成为产品的问题。而大多数领域都认为原型化所做的只是产品模型化的工作。对于一个复杂系统而言,原型直接成为实际系统的设想是不现实的。除非原型化的模型是一个常规的应用系统或它只是对某种应用系统的小规模的扩充。

模型化阶段的重要目的是开发用户需求,实际系统的其他许多需求和限制是在模型制作中不应考虑的,它们不会在模型化中完成,而只有在应用系统产品中才需要,例如打印格式、操作运行书、转换过程、用户文档、生产恢复/启动过程、质量控制检查、数据库规模、人/机错误处理、测试规划、硬件/通信资源配置、应急回退过程、训练过程。

因此,原型化方法并不能省去必要的考虑和步骤,从而建立起一个能在生产环境中工作的系统。原型化方法不应作为一个可压缩开发生命周期(见图 7.8a)的方法来提供。原型化方法能够以最佳方式描述数据、功能和人机界面等问题,但不能表达其他许多必须解决的问题。

对目标系统所需要的结果,不排除用原型作为一部分或不完整的基础。如果目标系统需要的结构与模型一致,无论是改进模型为目标系统,或是购买软件商提供的系统,或是通过组合建立系统,都是一个经济与技术问题,应该细致地研究和决定。

不论能否直接完成原型,完整的结果是开发真实系统的基础和条件。一种原型的设计和考虑并不能直接提供一个可操作的系统,应该通过较全面的补充和完善,才有可能将通过原型化所得到的结果过渡到一个真实的、可操作和可运行的系统。

开发高质量的、规模较大的系统没有捷径,更无魔法。它必须依靠一支有经验的开发队伍,按照原型化的原则和策略,在较完善的开发环境中,在用户支持和配合下,动态地调整开发的诸多环节,以最大限度地满足用户对系统的需求。

虽然原型作为一个产品系统是否能被直接完成是有争议的,但这并不意味着它就不能成为产品系统。如何解决从原型过渡到可运行的产品系统,不能再回到传统的结构化开发生命周期的方法。人们提出了一个修改了的生命周期(见图 7.8b),它能接受原型法开发过程的结果。这个生命周期可达到一个较完善的结果,它补充和调整了设计过程,加入了优化/补充步骤,以最终协调原型使适应任何操作约束。由于原型化和模型字典驱动的软件已经很成熟,所以这种生命周期已经变为一种标准的方法。

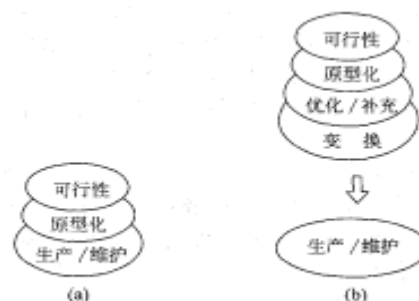


图 7.8 压缩的生命周期和修改后的生命周期

7.4 原型化中心

7.4.1 原型化中心的组织

在一个典型的企业内部,通常会有 3 个主要的数据处理活动中心(见图 7.9)。

- **开发中心**: 是开发新的系统和维护现有系统所配置的资源组合。

- **生产中心**: 是支持和完成产品系统所配置的资源组合。

- **信息中心**: 是为最终用户直接地存取、造表、分析数据所配置的资源组合。

虽然中心之间会有重叠的需求,但是每个中心有自己常规的和特别的资源,以支持它们专门的职能。

作为一种数据处理服务,原型的出现需要建立第 4 个数据处理中心—原型化中心(图 7.10)。

虽然其规模和资源耗费都会小于其他 3 个中心,但为了有效地满足原型制作的需要,必须建立一个可以提取和组合的资源集合。原型化的目标和目的不同于其他中心,但它同样为保证原型化的顺利进行,需要以最佳方式去组合,诸如原型开发人员、计算机硬件和软件等各种资源的组合以及建立一个支持原型开发计划和高效率的工作环境。

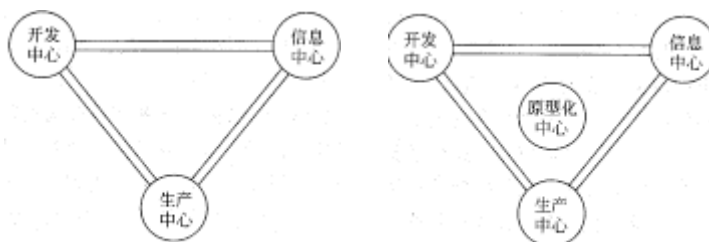


图 7.9 数据处理中心

图 7.10 原型化中心

7.4.2 原型化中心的人员配备

一个原型化人员是一位建筑师，他对系统的总体概念和系统各部分目标的一致性负有责任。建立一个大的系统的模型是一项创新的和高技能的活动，而建立计算机信息系统又是人们所做的最复杂的活动之一。

原型化人员要在一个缩小的模型上进行完整的开发加工。其整个生命周期要压缩到需求阶段内。所有必须的工作，如分析、设计、编程、屏幕生成、报表生成以及人为因素都要由原型化人员来完成。因此原型化人员应该是一个训练有素，具有技能、技巧、经验和才干的专业人员。他还能对用户提出的模糊需求给出指导，但他并不一定是某一领域的专家。

原型不能由多人的大组织来完成。原型化小组最佳规模是由两个人组成，或者再外加一位多为项目并行处理做补充的、提供支持功能的第三者。

现从以下几个方面说明组小的必要性：

- 通信。系统开发中人际通信应保持在最小限度。数据处理文献表明，随着小组规模的扩大，引起的通信问题呈指数形式增长，生产率将随着小组规模扩大而下降。

- 当小组规模超过 2 人时，哪怕只增加到 5 人，组内成员的对话总数可能会从 1 很快地增加到 22，两人小组甚至不需要写出说明文件就能进行个人通信和交流学识，进行系统的开发。

- 观念一致性。一项应用应该是一个系统而不是 N 个系统。为了保持概念的一致性和维护在系统实现中的集成性，原型开发者的人数应该尽可能少。两个人工作几乎能自然地维护着产品的一致性。他们会不断地把将完成的工作集成为一个系统。

- 生产率。组合工程和“剪裁与粘贴”是获得小组最好生产率的策略。当你把整个事情当做一个部件去完成时，你能预见到未来的用途，而且当你需要一个部件时，你有系统的存在于头脑中的索引供搜索和查找。这种优势并不总存在，它会随着小组的规模扩大而消失。

可以降低对原型化人员技能要求的深度。原型化人员应能完成许多任务，但他们并不必也不能都达到专家的水平。原型化人员应该是一个系统构筑的老手，但他只是解决普遍性的问题，而特殊的问题应由相应领域中的专家来解决。

如何获得这些具有广泛技能水平的原型开发人员？最好是选用本系统内优秀的系统分析员和项目主持人，他们除了在传统的开发环境中正在完成类似的工作外，还在不断地积累有关的情况和经验。

在原型化的初期，被选择的人员可能不一定是具有全面技能的多面手，可能没有几个人是自分析到调试报告的每一过程都被训练和经历过的。然而单位内部有许多优秀的教练员和职业专家，他们可以在广泛的学科原理和主要概念方面训练基本上理解开发过程的人员。“80/20”规则对于原型化人员的训练仍然是有效的：80%的知识和问题可以很快地被掌握和被解释清楚，而有 20%的技术要花几年的时间来学习和体会。对原型化人员来说，80%就可能足够了。

培养新的原型化人员的最好方式是将具有基本技能的技术人员放到一个连续接触快速制作的环境中。和有经验的原型化人员结合并参加其工作是一种学习技术和知识的良好方式。

7.4.3 硬件需求

原型化中心必须有硬件的支持。但不论原型化中心设置的计算机是什么类型，对于原型开发人员和最终用户来讲，他们接触的只是一台终端，终端是他们的直接操作对象。

1. 终端

原型化中心要求的终端可以概括为 3 种类型：

- 用户终端。原型化硬件结构应该提供与最终用户通常使用类型相同的终端。
- 原型软件终端。原型化硬件结构应该包括最有效地使用原型化软件所需要的终端类型。
- 打印终端。原型硬件结构应该包括能够快速访问批打印输出的高速打印终端。

2. 个人计算机

原型化中心应该考虑带有终端仿真软件的个人计算机的可利用性。个人计算机具有便携的优点。由于终端仿真软件的利用，可使一台个人计算机能像其他任何一种终端那样使用。这给原型化人员带来了两方面的益处：

- 个人计算机允许原型化人员在他认为合适的地方，如在家中以舒适的方式进行工作。
 - 更重要的是，个人计算机可带到用户工作现场去演示、修改和验证。用户可以现场进行操作测试。
- 个人计算机为原型化的研究提供了好的、灵活的条件。

7.4.4 软件需求

原型化软件需求内容可简要的归纳为：

- 数据字典驱动。
- 有结构地支持组合工程。
- 从现有组件“剪裁和粘贴”出新的组件。
- 提供交互原型化工作台。
- 使用描述性文档而非过程化文档。
- 自动生成应用文档。



图 7.11 非集成化环境, 常规软件结构

而原型化处理中集成化的概念是很重要的。图 7.11 描述了一个常规的软件开发环境，对于这种环境，无论每一个部件的功能和优点多么完善和突出，但这个环境是非集成化的。应该在部件之间架起桥梁以允许相互间的通信。在非集成化的环境中，实现快速原型化是不可能的。图 7.12 描述了一个集成化的软件开发环境，一个活动的和集成的数据字典将各种资源加以联系和定义。它为原型化人员提供了一个记录管理系统，形成一体化的开发环境。

集成是原型件的首要的软件需求。在几年前还不存在这种软件，因而高效率的原型化是不可能的。

大多数常规软件产品强调其功效，要求利用最少资源和在大多数情况下能提供有效的运行。而对原型软件来说，其执行特性是次要的。这种软件的着眼点在于主要考虑原型化人员的最佳生产率。原型只是建造中的模型，不必要要求达到运行中的最佳状态。

生成完整应用所必需的所有软件成分应包含在原型化结构中(见图 7.12)。应该包含下列软件成分：

- 一个灵活和集成的数据字典。
- 一个强功能的数据库管理系统。
- 一个能生成批处理和联机处理程序的高水平的过程化语言。
- 一个过程处理监督器。
- 一个批处理和联机问题的查询语言。
- 一个非过程化的报告生成器。
- 一个文档资料生成器。
- 一个文本编辑器。

这种需求与前述的集成体系的需求是相同的。如果原型化软件能生成整个模型，则它提供给原型化人员的软件成分必须是完整的。如果它是可集成的，则所有软件成分必须处在字典控制之下。

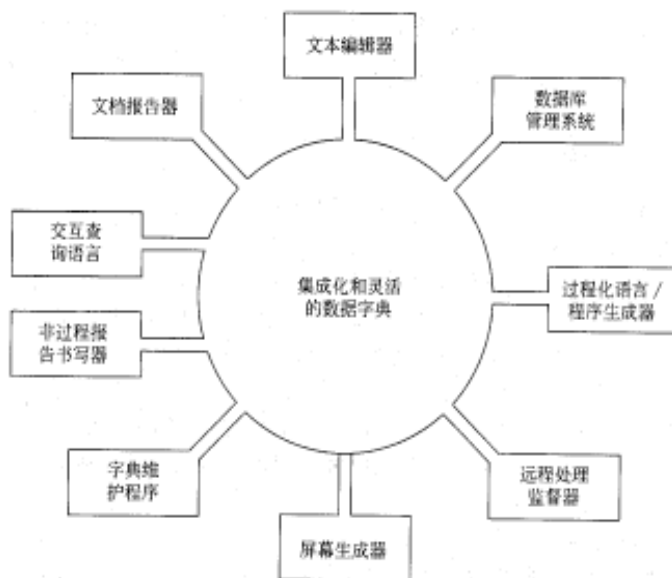


图 7.12 集成化软件结构, 所有的软件都通过数据字典通信

7.4.5 原型工作环境

开发原型需要创造一个工作环境，这将有助于提高生产率。

1. 项目工作室的建立

用于开发原型的自封闭式工作环境将有利于促进合作、减少约会、保证资源和资料完整、齐全。除了常规的设施外，项目工作室应有交互式终端、批量打印终端、软件参考文档和帮助人们进行回忆的有关软件或原型的视觉信息。

2. 快速响应的工作环境

快速响应对原型演示过程很重要，用户在屏幕前等待过长，将会丧失兴趣和信心，因而提供一个合适的环境是必要的。最一般的时间要求是：交互式最多不得超过 5 秒，批处理方式 15 分钟左右。

3. 规范的原型构造过程

必要的标准和规范能加快原型向生产系统转换。作为需求说明的原型应可读、可分析、可解释，因此也必须是规范的。必须制定一套制度，以组织和管理开发人员。

4. 文档资源

原型开发要得到技术性和业务方面的文档资料，文档资料不仅对直接构造原型有用，而且会提供大量的附加知识，如吸收别人的经验，改善设计方法。特别是与设计和求精应用原型有关的技术和过程，都有必要吸取别人的新思想和新方法。

5. 演示/展示设施

审查和评论原型的重要手段是演示。要使反复审查得以通过，应鼓励审查者踊跃地参加演示过程，因而必须使审查者感到舒适。实现这一目标的有效手段是将显示器与大屏幕投影机相连，审查人员可坐在工作台旁作详细记录；只要必要，就能对任何屏幕展开讨论。

6.集中式/分散式原型开发中心

在组织内部将原型开发部门集中还是分散，应根据组织的规模和业务复杂程度而定。大的组织可设置原型开发中心作为所有部门决策中心，项目负责人不仅需要而且必须与开发人员进行协商。这样安排的优点在于提高工作效率。可以为主要的开发领域设置开发中心，他们作为各自决策机构，例如销售部门、库存部门、财务部门等。这样，每个开发人员也可集中精力服务于一个特定的职能部门，从而精通业务。

集中与分散的结合，可能是一种兼顾效率和效果的方式。职能集中化有利于解决问题的有效性。而职能的分散化则更强调用户要求的响应和服务效果。

分散式的开发人员通过负责开发的组织接近用户具有以下明显的优点：

- 开发人员将成为业务领域的行家，从而在解决新问题时可少走弯路。
- 开发人员与用户所在组织形成了一种一致的工作关系。
- 不必考虑两个不同组织的开发者之间的合作，而职能分散

对大多数开发机构并不

- 需要设置分散的物理设施。

7.零件部门

零件部门的职能是为所有原型开发人员提供通用的零件。其职责是挑选/设计具有高度通用能力零件，对零件进行分类编目以便查找，保证零件的正确性。

零件可以是设计好的软件成分，也可以是分解的框架，它们的目标都是面向工程化和可重用化的。

原型开发中心做出的每个决策都是对它的开发过程的贡献。正是由于所有贡献的合成作用，才使得原型开发中心起到一个比各种零件的作用大得多的重要作用。

人员安排是最重要的因素。应将活动的高度可见性和高度的责任心相互结合起来。分配的工作应赋有难度，而不是只需稍加努力即可轻易完成的十分平常的工作。

如果所使用的软件/硬件结构不合适，则原型人员在开发过程中将会受到限制。虽然，为了实现特殊问题的原型化，可能要对软件做某些修改，但一般情况下，所列出的软件要求是不应改变的。软件是自封闭和集成化的。开发过程中，软件的挑选非常重要，如果不能灵活和重复地使用已有的零件或尽快地从现有零件开发出新的零件，就很难完成大量的开发工作，进而实现应用的原型化。

工作环境应满足开发者和审查者的要求。项目工作室为设计活动提供了一个高强度的工作场所。用于演示的大屏幕将加强所有审查者的观察和认真思考。采用集中还是分散原型开发方式，提供了对开发效率和效果之间进行选择的灵活性。零件提供部门为理想化的原型环境提供了可能，一些完整的原型几乎全部是由已有零件构成的。图 7. 13 概括了一个原型开发中心的各种资源。



图 7.13 组合原型化中心

7.5 原型化与项目管理

7.5.1 项目管理的必要性

原型化并不是孤立出现的事件，它是一个很活跃的过程，受控于项目管理。项目的功能包括 5 个方面：质量、资源、成本、时间和技术。计划、控制和活动组织是由项目管理机制控制的。

原型引入之后，需要对项目的过程加以适当修正，许多已被现行项目管理所接受的开发项目，还需要对同一问题再给出精确的定义。通过对计划的认真执行、组织机构的进一步合理化和对计划执行的全面的监督，那么一些管理中的疏忽将可能避免。

原型化并不会改变整个项目实施和项目管理的有效性和合理性，而是作一些适当的调整。对一个原始的开发项目，项目管理试图给出一个令人满意的控制，原型的分阶段执行需要有适当的约束和一定的灵活性。

因此，和所有开发方法一样，原型化方法也需要项目管理，但呆板的、拘于形式的管理会抑制灵活的原型开发方式，因而需要对项目的传统方式和过程加以适当的修正，使其既有灵活性又有章可循。

7.5.2 项目管理的内容

由于原型化的影响，项目管理有以下 4 方面的内容。

1.估计过程

这就是估计原型的时间、成本和系统目标的方法。对项目估计成本和时间是很困难的工作，实际情况常常不同于估算。但既然是简单的估计也会对原型制作有利。

对简单的估算规则有如下情况：

(1)对建立初始原型的估计：对于允许的富有实际意义的开发项目时间上的估计要充分，另外还要保证用户始终对此感兴趣，但也不能冒浪费大量时间的风险。

(2)对原型的修改的估计：为保证项目发挥它的潜力，时间上要作充分的估计，要允许有意义的变化。还要保证符合质量要求。

(3)对建立初始原型和修改原型的估计：对于具有各种规模、多种复杂程序的原型，时间上的估计要灵活机动。经验告诉我们：影响估计的首要因素是问题的规模大小和复杂度；在用户面前修正原型的早期重复和大范围的功能追加要占用很多时间，而后期的重复仅仅是在显示屏上的重新定位。

原型化的过程可以帮助提高实际系统成本估计的可靠性。原型本身就是一个可估计的实际系统的样本资料。

原型化的成本估计就是指由项目管理所要求的实行系统的建立和修改成本的估计。首先，用户为满足其要求而支付的时间和设备，这些成本是显而易见的，它取决于每次重复周期的进展状况。其次，在原型被接受后，一个静态的成本估计立即可以做出。

2.费用重新分配

模型开发需要的所有费用都要记在用户的账单上，原型的制作也带来了占用机器的费用。

费用分配对控制重复周期是最有效的，因为重复会多花钱。用户要比较追加功能后的费用。如果原型化的费用不是分摊在用户身上，那么，这步主要的控制作用在重复周期中将失去意义。

3.变化控制

由谁来决定原型的改变是一个复杂的问题。鉴于目前的用户大多数不是单一个体，而每个用户的要求又不尽相同，探索这一问题的最佳解决方案是，对项目管理机制来说，做一个交互式的控制板，一个小型设计组根据目前掌握的资料作出变化或不变化的决定。一些用户应该明确地做出自己的选择。

4. 活动停止

在传统的定义讨论中，用户的活动停止是以叙述/图解模型为基础的。在原型化环境中活动停止就相当于已允许原型作为理想的系统。用户已同意这些可靠的原始资料。用户已看到和体会到这就是他们将来所要得到的。

由上可见，项目管理需要有严格的变化控制过程，以保证向用户交付可接受的原型；用户要对任何变化负责，这些变化会影响系统的交付；开发者要得到关于原型内部操作系统的外部表示的复制品。

如果不坚持上述原则，整个原型化的目标将会丢失。

7.6 结论

原型化方法的介绍即将结束，可以得出 6 个结论：

- 原型化从用户角度考虑是非常适当的。
- 原型化从开发者角度考虑也是合适的。
- 原型化可用于大规模的项目开发。
- 原型化是可行的。
- 原型的制作者相当于一个建筑师。
- 原型制作的核心策略为处理过程提供了方便的工作环境。

原型化对解决现实世界的事务处理系统是高效率的，它的成功就在于起点低，可以追加功能和扩充，还可以交付富有生气的原型，这是原型化的基本特征。所定义的需求问题最后从原型那里得到圆满的解答。

原型化是可能的，但在国内需要实践，提供好的原型化环境，并运用原型化原理来建立原型中心，积累原型系统和软件，将是又一种减少重复工作、提高系统/软件复用率、高效率开发系统和培养人员的途径。

未来的开发，用户将是一支生力军，而原型化方法将是一种最适合于用户运用的方法。

原型化的基本原理并非全新，但它在计算机辅助下，却能发挥出极大的优越性和高效率。而且其应用面完全可以超出软件的开发，而发展到整个计算机信息系统/应用系统领域，甚至更大。

第 8 章 软件工程

8.1 软件生存期过程

软件生存期概念帮助人们较为全面地认识了软件开发(包括软件的运行与维护)。在 1988 年制定和公

布的国家标准《GB 8566-88 计算机软件开发规范》中，将软件生存期划分为 8 个阶段，即可行性研究和计划、需求计划、概要设计、详细设计、实现、组装测试、确认测试、使用和维护。

该标准为每个阶段规定了任务、实施步骤、实施要求以及完成的标志。对软件生存期按此方式做 8 个阶段的划分大致符合也适应传统的瀑布模型，反映了 20 世纪 80 年代人们对软件工程的认识。

此后，于 20 世纪 90 年代初形成了软件工程过程的概念。认为软件工程过程是为了获得软件产品或是为了完成软件工程项目需要完成的有关软件工程活动，每一项活动又可分解成一些软件工程任务。每一个软件开发机构都可以规定自己的软件工程过程，针对不同类型的软件产品或是软件工程项目，软件机构可以规定自己适用的软件工程过程，甚至可能使用多个不同的软件工程过程。这使我们对软件生存期有了更为广义的理解。软件生存期中除了开发过程以外还有许多其他的软件工程过程。这一思想集中体现在 1995 年制定和公布的国家标准((GB/T 8566-1995 信息技术—软件生存期过程》中。事实上，它是《GB 8566-88 计算机软件开发规范》的代替标准。

该标准定义了软件生存期 7 个主要过程，它们和管理过程、获取过程、供应过程、开发过程、操作过程、维护过程和支持过程。

支持过程包括了文档开发过程、配置管理过程、合同要求的评审和审计过程、验证和确认过程、软件质量保证过程、改正过程、培训过程和环境建立过程。

1995 年国际标准化组织在此基础上对生存期过程作了调整，公布了新的国际标准，即((ISO/IEC 12207 信息技术—软件生存期过程》。该标准全面、系统地阐述了软件生存期的过程、活动和任务。标准定义的 17 个过程分别属于主要过程，支持过程和组织过程。我们可以通过图 8.1 看出它的结构。

在表 8.1 中给出了 17 个过程的主要活动和任务的描述。以下对该标准提出的软件生存期过程给予简要说明。

1. 主要生存期过程(primary process)

包括 5 个过程，这些过程供各主要当事方(如需求方、供方、开发者、运行者和维护者)在参与或完成软件产品开发、运行或维护时使用。

(1)获取过程：需求方获取系统、软件产品或软件服务的活动。

(2)供应过程：供方向需求方提供系统、软件产品或软件服务的活动 “

(3)开发过程：开发者定义并开发软件产品的活动。

(4)运行过程：运行者在规定的环境中为其用户提供计算机系统服务的活动。

(5)维护过程：维护者提供维护软件产品服务的活动。

2. 支持生存期过程(supporting process)

包括 8 个过程，其每个过程均有明确的目的支持其他过程，帮助软件项目获得成功及良好的产品质量。

(1)文档编制过程：记录生存期过程中产生的信息所需的活动。

(2)配置管理过程：实施配置管理活动。

(3)质量保证过程：为确保软件产品和软件过程符合规定的需求并能坚持既定计划所需的活动。联合评审、审核、验证与确认可作为质量保证技术使用。

(4)验证过程：为确保一个活动的产品满足前一活动对它的要求和条件的活动。

(5)确认过程：为确保最终产品满足预期使用要求的活动。

(6)联合评审过程：评审方与被评审方共同对某一活动的状态和产品进行评审的活动。

(7)审核过程：审核项目是否按要求、计划、合同完成的的活动。

(8)问题解决过程：分析和解决在开发、运行、维护或其他过程中出现的问题(不论其性质和来源如何)的活动。

3.组织生存期过程(organizational process)

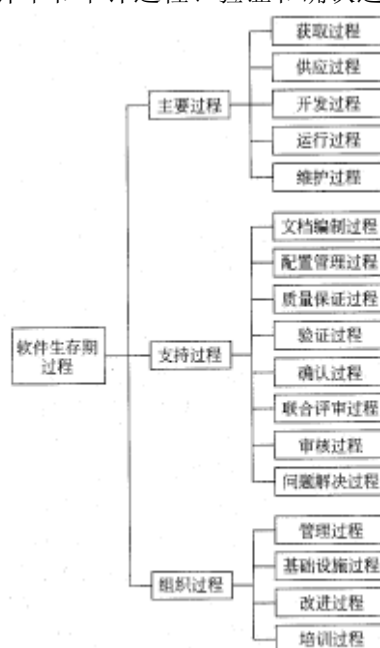


图 8.1 《ISO/IEC 12207 信息技术——软件生存期过程》标准定义的生存期过程

包括 4 个过程，这些过程被某个机构用来建立和实现与生存期过程相关的基础结构，甚至人事制度，并使其不断得到改进。

(1) **管理过程**：规定生存期过程中的基本管理活动，包括项目管理。

(2) **基础设施过程**：建立生存期过程基础结构的基本活动。

(3) **改进过程**：某一机构(需方、供方、开发者、运行者、维护者或其他过程的管理者)为建立、测量、控制和改进其生存期过程需开展的基本活动。

(4) **培训过程**：对人员进行适当培训所需的活动。

8.2 软件过程能力评估

软件产品的质量取决于软件开发过程，具有良好软件过程的软件机构能够开发出高质量的软件产品。这一点虽然早已为人们公认，但切实地在软件过程方面开展工作也只是十多年前的事。1987 年在美国国防部支持下，卡内几—梅隆大学率先推出了软件过程评估项目的研究成果—软件过程能力成熟度模型 CMM，很快就引起了软件界的广泛关注，并在此后引发了一系列反响，导致在其基础上形成了国际标准(ISO/IEC 15504)。

事实上，CMM 给了软件开发机构一把度量软件过程的尺子，这个尺子从低水平到高水平有 5 个等级的刻度，用它去度量便是软件过程评估的工作。另一方面，CMM 同时也是一个指南，它在客观上起到了指导软件机构的作用，它可告诉软件机构，如果要在原有的水平上提高一个等级，应该关注哪些问题，这就是软件过程改进的工作。

本章在概述软件过程评估之后，重点介绍了 CMM，并且在最后把国际标准也作了扼要的描述。

8.2.1 软件过程评估的意义

软件过程评估是软件过程改进和软件能力评价的前提环节。

8.2.1.1 软件过程改进的需要

1. 软件过程不断改进是软件工程的基本原理之一

1983 年美国 TRW 公司 B. W. Boehm 总结了该公司在 12 年内、总共花了 15 000 人年、先后开发五代指挥控制软件的经验，得出如下 7 条原则：

- 按软件生存周期分阶段制定计划并认真实施。
- 逐阶段进行确认。
- 坚持严格的产品控制。
- 使用现代程序设计技术。
- 明确责任。
- 用人少而精。
- 不断改进开发过程。

这就是著名的软件工程七原理。由此可见，不断改进软件开发过程是软件工程的基本原理之一。

2. 软件过程改进是软件生存周期的基本过程之一

软件工程界始终十分重视对软件过程的研究，20 世纪 70 年代中期形成了软件生存周期的概念，1995 年正式发布了一项国际标准，即 ISO/IEC 12207 信息技术—软件生存周期过程，这是软件过程研究的一个重要成果。这项标准科学地定义了软件生存周期的过程，总共 17 个，其中一个就是改进过程。

表 8.1 软件生存期过程的主要活动和任务描述

项目	过程名	主体	主要活动和任务描述
主要过程	获取 acquisition	需方	定义、分析需求或委托供方进行需求分析而后认可；招标准备；合同准备以及验收
	供应 supply	供方	评审需求；准备投标；签订合同；制定并实施项目计划；开展评审及评价；交付产品
	开发 development	开发者	系统需求分析；系统结构设计；软件需求分析；软件结构设计；软件详细设计；软件编码和测试；软件集成；软件合格测试；系统集成；系统合格测试；软件安装及软件验收支持
	运行 operation	运行者	制定并实施运行计划；运行测试；系统运行；对用户提供帮助和咨询
	维护 maintenance	维护者	问题和变更分析；实施变更；维护评审及维护验收；软件移植及软件退役
支持过程	文档编制 documentation		设计文档编制标准；确认文档输入数据的来源和适宜性；文档的评审及编辑；文档发布前的批准；文档的生产与提交、储存和控制；文档的维护
	配置管理 configuration management		配置标识；配置控制；记录配置状态；评价配置；发行管理与交付
	质量保证 quality assurance		软件产品的质量保证；软件过程的质量保证以及按 ISO 9001 标准实施的质量体系保证
	验证 verification		合同、过程、需求、设计、编码、集成和文档等的验证
	确认 validation		为分析测试结果实施特定的测试；确认软件产品的用途；测试软件产品的适用性
	联合评审 joint review		实施项目管理评审(项目计划、进度、标准、指南等的评价)；技术评审(评价软件产品的完整性、符合标准等)
	审核 audit		检验项目是否符合需求、计划、合同以及规格说明和标准
组织过程	问题解决 problem resolution		分析和解决开发、运行、维护或其他过程中出现的问题，提出响应对策，使问题得到解决
	管理 management	管理者	制定计划、监控计划的实施，评价计划实施；涉及到有关过程的产品管理、项目管理和任务管理
	基础设施 infrastructure		为其他过程所需的硬件、软件、工具、技术、标准，以及开发、运行或维护所用的各种基础设施的建立和维护服务
	改进 improvement		对整个软件生存期过程进行评估、度量、控制和改进
培训过程	培训 training		制定培训计划；编写培训资料；培训计划的实施

实践表明，软件过程需要不断完善，首先从非工程化的软件开发方式改变为工程化的软件开发方式，按照软件工程的系统方法进行软件的工程活动和管理活动，进而不断完善各个软件过程，从而不断提高软件过程能力。随着这种能力的提高，一个软件组织完成软件产品时在预算、进度，特别是产品质量方面的风险就逐步降低。显然，软件过程能力的提高需要首先对当前的软件过程状况进行科学的评估。

8.2.1.2 降低软件风险的需要

1.软件采购者的需要

软件产品或软件服务的采购单位进行招标、选择承制者时，为了降低风险，需要对备选单位的软件过程能力进行评价，而这种评价的依据是对该单位的软件过程的评估结果。

2.软件承制者的需要

软件产品研制单位和软件服务单位在响应顾客的需要、进行投标时，为了降低风险，需要对自己的软件过程能力进行评价，避免承担力所不及的任务，而这种评价的依据仍然是根据实际需要，对相应软件过程的评估结果。

8. 2. 2 软件过程评估方法的产生

如何评估软件过程，曾经是软件过程研究者要回答的另一个重要问题。这方面的研究在 20 世纪 80 年代取得了突破性的进展。1987 年美国卡内几—梅隆大学软件工程研究所(Software Engineering Institute, SEI)以 W. S. Humphrey 为首的研究组发表的“承包商软件工程能力的评估方法”是最杰出的代表。这个评估方法给出了软件过程能力成熟度框架，1991 年发展为 SEI CMM(Software Engineering Institute-Capability Maturity Model) 1. 0(能力成熟度模型 1.0 版)，它把软件过程按完善程度分为 5 个等级，描述了不同完善程度的软件过程的不同特点。这个方法本是美军委托研究，用来评估军用软件承包商的软件过程，从而评价其软件开发能力的；但在试用过程中，该方法的另一个更加重要的作用越来越被人们重视，就是它描述了软件过程不断改进的科学途径，使软件开发组织能自我分析，找出尽快提高软件过程能力的策略。这个方法的意义得到国际软件产业界和软件工程界广泛关注和认可，人们认为这是 80 年代软件工程技术最重要的发展之一。

1991 年国际标准化组织采纳了一项动议，开展调查研究，以便确定是否需要编制有关软件过程评估的国际标准，并于 1993 年得出肯定的结论，开始了有关标准的研究制定工作，现已取得重要结果，产生了技术报告 ISO/IEC TR 15504 SPICE(Software Process Improvement and Capability dEtermination)信息技术—软件过程评估，并预计于 2001 年产生正式标准。从该技术报告的内容看来，制定标准的基本目的及其思路均与 SEI CMM 相似。

上述两个研究组织的有关研究结果以不同方式给出了评估软件过程的方法和不断改进软件过程的科学途径，在下面两节分别加以简介。

8.2.3 软件能力成熟度模型 CMM(Capability Maturity Model)简介

8.2.3.1 模型概要

下面以 SEI CMM1.1 版本为依据进行介绍。SEI CMM1. 1 模型概要如表 8.2 所示。

这个模型的制定者有一个基本认识，就是软件开发的危险之所以大，其中最关键的问题在于软件开发组织不能很好地管理其软件过程，从而使一些好的开发方法和技术起不到预期的作用。可是，即使在这种组织中，个别软件项目仍能产生优质产品。这些项目的成功一般是通过工作组的杰出努力，而不是通过重复使用具有成熟软件过程的方法。

表 8. 2 涉及的一些概念说明如下。

• **软件过程：**用于开发和维护软件及其相关产品(例如项目计划、设计文档、代码、测试用例、用户手册等等)的一系列活动，包括软件工程活动和软件管理活动，其中必然涉及有关的方法和技术等。

• **软件过程能力：**描述(开发组织或项目组)通过遵循其软件过程能够实现预期结果的程度。一个软件

表 8.2 SEI CMM1.1 模型概要

过程能力等级	特 点	关键过程域
1. 初始级	软件过程是无序的,有时甚至是混乱的,对过程几乎没有定义,成功取决于个人努力。管理是反应式(消防式)的。	
2. 可重复级	建立了基本的项目管理过程来跟踪费用、进度和功能特性。制定了必要的过程纪律,能重复早先类似应用项目取得的成功。	需求管理 软件项目策划 软件项目跟踪和监督 软件子合同管理 软件质量保证 软件配置管理
3. 已定义级	已将软件管理和工程两方面的过程文档化、标准化,并综合成该组织的标准软件过程。所有项目均使用经批准、剪裁的标准软件过程来开发和维护软件。	组织过程定义 组织过程焦点 培训大纲 集成软件管理 软件产品工程 组际协调 同行专家评审
4. 已定量管理级	收集对软件过程和产品质量的详细度量,对软件过程和产品都有定量的理解与控制	定量的过程管理 软件质量管理
5. 优先级	过程的量化反馈和先进的新思想、新技术促使过程不断改进。	缺陷预防 技术变更管理 过程变更管理

开发组织或项目组的软件过程能力提供一种预测该组织承担下一个软件项目时最可能的预期结果的方法。软件过程能力既可对整个软件开发组织而言，也可对一个软件项目组而言。

- **软件过程性能：**表示(开发组织或项目组)遵循其软件过程所得到的实际结果。同样，软件过程性能既可对整个软件开发组织而言，也可对一个特定软件项目组而言。可见，软件过程性能描述已得到的实际结果，而软件过程能力则描述最可能的预期结果。

- **软件过程成熟度：**一个特定软件过程被明确和有效地定义、管理、测量和控制的程度。成熟度可指明一个软件开发组织软件过程能力的增长潜力。随着软件组织的软件过程成熟度的提高，开发组织通过其方针、标准和组织机构等将其软件过程规范化和具体化。从而使得开发组织明确定义的有关管理和工程的方法、实践和规程等在现有人员离去后仍能继续下去。

- **软件能力成熟度等级：**软件开发组织在走向成熟的过程中几个具有明确定义的、表征软件过程能力成熟度的平台。每一个成熟度等级为过程继续改进、达到下一个等级提供一个基础。每一等级包含一组过程目标，当其中一个目标被达到时，就表明软件过程的一个(或几个)重要成分得到了实现，从而导致组织的软件过程能力增长。

- **关键过程域：**互相关联的若干软件实践活动和有关基础设施的一个集合。每个软件能力成熟度等级包含若干个对该成熟度等级至关重要的过程域，它们的实施对达到该成熟度等级的目标起保证作用，这些过程域就称为该成熟度等级的关键过程域。顾名思义，既然有关键过程域，就可能有非关键过程域。由于非关键过程域对达到相应软件成熟度等级的目标不起关键作用，所以在定义软件成熟度等级时不加以叙述。

- **关键实践：**对关键过程域的实施起关键作用的方针、规程、措施、活动以及相关基础设施的建立。关键实践一般只描述“做什么”，而不强制规定“如何做”。关键过程域的目标是通过其包含的关键实践的实施来达到的。

国际上有一个已取得共识的基本观点是：整个软件过程的改进是基于许多小的、进化的步骤，而不是通过了殊革命性的创新来实现的。这些小的进化步骤就通过一些关键实践来实现。

- **软件能力成熟度模型：**对软件组织进化阶段的描述，随着软件组织定义、实施、测量、控制和改进其软件过程，软件组织的能力经过这些阶段逐步前进。这个能力成熟度模型使软件组织能够较容易地确定其当前过程的成熟度并识别出其软件过程执行中的薄弱环节，确定对软件质量和过程改进最为关键的几个问题，从而形成对其过程的改进策略；软件组织只要关注并认真实施一组有限的关键实践活动，就能稳步地改善其全组织的软件过程，使全组织的软件过程能力持续增长。

8.2.3.2 模型的产生和原理

美国软件工程研究所(SEI)提出的软件能力成熟度模型(CMM)的分层结构基于已有 60 多年历史的产品质量原理。希沃特(Walter Shewart)在 20 世纪 30 年代发表了统计质量控制原理。戴明(W. Edwards Deming)和朱兰(Joseph Juran)的著作又进一步发展和论证了该原理。SEI 将这些原理应用于软件开发，发展成为软件过程成熟度框架，该框架为软件过程定量控制建立了项目管理和项目工程的基本原则，这个框架是软件过程得以不断改进的基础。

实际上，将质量原理改变为成熟度框架的思想是克罗斯比(Philip Crosby)在其著作“Quality is Free”中首先提出的。克罗斯比的质量管理成熟度网络描述了采用质量实践时的 5 个进化阶段。该成熟度框架后来又由 IBM 的拉迪斯(Rom Radice)和他的同事们在汉弗莱(Watts Humphrey)指导进一步改进以适应软件过程的需要。1986 年，汉弗莱将此成熟度框架带到了软件工程研究所并增加了成熟度等级的概念，形成了当前软件产业界正在使用的框架的基础。

汉弗莱的成熟度框架早期版本发表在 1987 年的 SEI 技术报告。该报告中还发表了初步的成熟度提问单，这个提问单作为工具给软件开发组织提供了软件过程评估的一种方法。1987 年又进一步研制出了软件过程评估和软件能力评价两个方法，以便估计软件过程成熟度。自 1990 年以来，在美国政府和工业部门许多人的帮助下，SEI 基于几年来将框架运用到软件过程改进方面的经验，进一步扩展和精炼了该模型，命名为软件工程研究所的能力成熟度模型(SEI CMM)。

8. 2.3.3 不成熟和成熟软件组织的比较

表 8.3 概述了不成熟软件组织与成熟软件组织的差异，这种比较分析不仅是形成软件能力成熟度模型的基础，也有利于理解该模型。

在不成熟的软件单位，软件过程一般由实践者及其管理者在项目进程中临时拼凑而成。通常即使已有规定的软件过程，也不能严格地遵守和贯彻。这样的软件组织的管理一般是反应式(也称消防式)的，通常经理们集中精力于解决即时危机。这样的组织执行合同时，由于制定的计划进度和经费预算不是根据现实

的估计，因而推迟进度和超出预算已成惯例。在硬性规定时限(所谓后墙不倒)的情况下，为满足进度要求，常在产品功能和质量上作出让步。

在不成熟组织中，不存在判断产品质量或者解决产品或过程问题的客观基础。因此，产品质量难以预测。当项目进度滞后时，常缩短或取消像评审和测试这些旨在提高质量的活动。

可是，一个成熟软件组织具有在全组织范围管理软件开发过程和维护过程的能力。规定的软件过程被正确无误地通知到所有职员，工作活动均按照已规划的过程进行。强制式的过程一般比较适用，而且和实际工作方式相一致。这些已定义的过程必要时将会变更，通过可控的先导性试验和费效分析使这些过程得到改进。对已定义过程中的所有岗位及其职责都有清楚的描述，通过文档和培训使全组织有关人员已定义的软件过程都能很好地理解。由于全组织运用统一的软件过程使过程纪律性一致增强，从而使其软件过程所导致的生产率和质量能随时间的推移得到改进。

成熟组织中，经理监控软件产品的质量和顾客的满意程度。在判断产品质量和分析产品及过程问题方面有客观的、定量的基础。进度和预算是基于以前的实施数据，因而是现实的；通常都能达到产品的成本、进度、功能和质量的预期结果。一般讲，成熟组织一致地遵循一个有纪律的过程，因为所有的参加者都了解这样做的价值，而且存在支持该过程的必要基础设施。

表 8.3 不成熟软件组织与成熟软件组织的比较

项目	不成熟的软件组织	成熟的软件组织
软件过程	临时拼凑，不能贯彻	有统一标准，且切实可行，并不断改进；通过培训，全员理解，各司其职，纪律严明
管理方式	反应式(消防式)	主动式，监控产品质量和顾客满意程度
进度、经费估计	无实际根据，硬性限定时限，常在质量上作让步	有历史数据和客观依据，比较准确
质量管理	问题判断无基础，难预测，进度滞后时，常减少或取消评审、测试等保证质量的活动	产品质量有保证，软件过程有纪律，有必要的支持性基础设施

8.2.3.4 软件过程成熟度的 5 个等级

CMM 提供了一个框架，将软件过程改进的进化步骤组织成 5 个成熟度等级，为过程不断改进奠定了循序渐进的基础。这 5 个成熟度等级定义了一个有序的尺度，用来测量一个组织的软件过程成熟度和评价其软件过程能力。这些等级还能帮助组织对其改进工作排出优先次序。成熟度等级是已得到确切定义的、在向成熟软件组织前进途中的平台。每一个成熟度等级为继续改进过程提供一个台基。每一等级包含一组过程目标，通过实施相应的一组关键过程域达到这一组过程目标，当目标满足时，能使软件过程的一个重要成分稳定。每达到成熟度框架的一个等级，就建立起软件过程的一个相应成分，导致组织过程能力一定程度的增长。如图 8.2 所示。

图 8.2 所示的 5 个等级各有其不同的行为特征，现通过以下 3 个方面来描述不同等级组织的行为特征：即一个组织为建立或改进软件过程所进行的活动、对每个项目所进行的活动和所产生的横跨各项目的过程能力。

1.等级 1—初始级

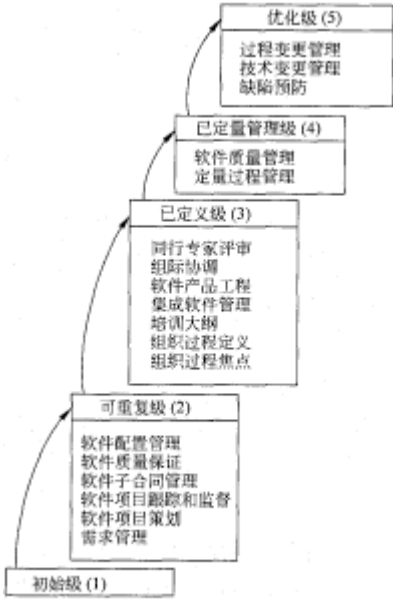
在初始级上，组织一般不能提供开发和维护软件的稳定环境。当组织中缺乏健全的管理实践时，不适当的规划和反应式的驱动体系会降低良好的软件工程实践所带来的效益。在危机时刻，项目一般抛弃预定的规程，回复到仅作编码和测试。项目的成功完全依赖于某位杰出的经理及某个有经验的、战斗力强的软件队伍。偶尔，有能力的、坚强的软件经理能经受住在软件过程中走捷径的压力，但当他们离开项目后，他们能使过程稳定的影响也随之消失。甚至一个纪律性较强的工程过程也不能克服由于缺乏健全的管理实践所造成的不稳定。

等级 1 组织的过程能力是不可预测的，因为随着工作进展，软件过程经常被改变或修订(过程是无序的)。进度、预测、功能性和产品质量一般是不可预测的。实施情况依赖于个人的能力，且随个人固有的技能、知识和动机的不同而变化。等级 1 组织几乎没有明显的稳定的软件过程，只能依靠个人的能力而不是组织的能力去预测实施结果。

2.等级 2—可重复级

在可重复级上，已建立了管理软件项目的方针和实施这些方针的规程，可基于在类似项目中的经验对新项目进行策划和管理。达到等级 2 的目的是使软件项目的有效管理过程制度化，这使得组织能重复在以前类似项目中的成功实践。有效过程具有如下特征：实用、已文档化、已实施、已培训、已测量和能改进。

等级 2 组织中的项目已设置基本的软件管理和控制。项目制定的约定均来自对以前项目的观察结果和当前项目的需求，因而切实可行。项目的软件经理跟踪软件成本、进度和功能性。软件项目标准均已定义，



并且组织能保证准确地执行这些标准。如果有子承包商，软件项目便与他们一起努力建立一种强有力的顾客—供应商关系。

等级 2 组织的过程能力可概括为是有纪律的，因为软件项目的策划和跟踪是稳定的，能重复以前的成功。由于遵循切实可行的计划，项目过程处于项目管理系统的有效控制之下。

3.等级 3—已定义级

在已定义级上，全组织的开发和维护软件的标准过程已文召化，包括软件工程过程和软件管理过程，而且这些过程被集成为一个有机的整体，称为组织的标准软件过程。用这个标准软件过程来帮助软件经理和技术人员，使他们工作得更有效。组织中有一个专门负责软件过程活动的小组，例如软件工程过程组 (SEPG)。该小组制定并实施全组织的培训计划，以保证职员和经理具有履行其职责所必须的知识和技能。

根据项目特征剪裁组织的标准软件过程，从而建立定义的软件过程，称为项目定义软件过程。一个已定义软件过程包含一组协调的、集成的、妥善定义的软件工程过程和管理过程。妥善定义的过程具有如下特征：关于准备就绪的判据、输入、标准、进行工作的规程、验证机制（例如同行专家评审）、输出以及关于完成的判据。因为软件过程已妥善定义，管理者就能洞察所有项目的技术进展情况。

等级 3 组织的软件过程能力可概括为标准的和一致的，因为无论软件工程活动还是管理活动，过程都是稳定的且可重复的。在所建立的产品线内，成本、进度和功能性均受控制，对软件质量也进行跟踪。这种过程能力建立在整个组织范围内对已定义过程中的活动、角色和职责的共同理解之上。

4. 等级 4—已定量管理级

在已定量—管理级上，组织对软件产品和过程都设置定量的质量目标。对所有项目都测量其重要软件过程活动的生产率和质量。利用全组织的软件过程数据库收集和分析从项目定义软件过程中得到的数据。等级 4 上的软件过程均已配备有妥善定义的和一致的度量。这些度量为定量地评价项目的软件过程和产品打下基础。

通过将项目过程实施的变化限制在定量的可接受的范围之内，可实现对产品和过程的控制。对过程实施方面有意义的变化与随机变化（噪声）应能够加以区别。开发新应用领域的软件所带来的风险应是已知的，并得到精心的管理。

等级 4 组织的软件过程能力可概括为可预测的，因为过程是已测量的并在可测的范围内运行。该等级的过程能力使得组织能在定量限制的范围内预测过程 and 产品质量方面的趋势。当超过限制范围时，采取措施予以纠正。软件产品具有可预测的高质量。

5.等级 5—优化级

在优化级，整个组织集中精力进行不断的过程改进。为了预防缺陷出现，组织有办法识别出过程的弱点并预先予以加强。在对新技术和推荐的组织软件过程的变更进行费效分析时，利用有关软件过程有效性的数据，识别出采用了最佳软件工程实践的技术创新，并推广到整个组织。

等级 5 组织对所有软件项目组都分析缺陷，确定其原因，并且认真评价软件过程，以防止已知类型的缺陷再次出现，同时将经验教训告知其他项目。

等级 5 组织的软件过程能力的基本特征是不不断改进，因为这些组织为扩大其过程能力的范围进行着不懈的努力，因而不断改善其项目的过程性能。为了能够不断改进，既采用在现有过程中增量式前进的办法，也采用借助新技术、新方法进行革新的办法。

8. 2.3.5 跳越成熟度等级

这个问题涉及两个方面：一方面跨越等级的现象自然存在，另一方面跳越等级是不可能的。

1. 跨越等级的现象

处于较低等级的组织可以而且往往必须实施较高等级上的某些过程，因为这样作会带来好处。例如，虽然 CMM 中在等级 3 以前不讨论软件产品工程过程的活动—诸如需求分析、设计、编码和测试，但是甚至等级 1 的组织都必须进行这些活动。在有利可图时，等级 1 或等级 2 的组织可以进行同行专家评审（等级 3 的）、巴列托 (Pareto) 分析（即主次分析法，等级 4 的）或者引入新技术（等级 5 的）。在讨论一个组织为了从等级 1 提高到等级 2 应采取何种步骤时，常建议建立一个软件工程过程小组，而建立软件工程过程组是等级 3 组织的一个属性。虽然测量是等级 4 的关注焦点，但它也是较低成熟度等级的必备部分。

不过这些较高等级的过程或活动的潜力只有在建立了适当基础之后才能得到完全的发挥。例如，同行专家评审，要坚持实施，即使对非常紧急的项目也是如此，否则就不可能完全有效。成熟度等级只描述一个等级上占主导地位的问题。等级 1 组织的占主导地位的问题是管理；策划和管理软件项目的困难掩盖了其他问题。

2.跳越等级的错误

跳越等级地前进是不可能的。因为每个等级形成一个必要的基础，从此基础出发才能达到下一个等级，

因此,跳越等级是违反发展规律的,刻意追求跳越等级不可能取得成功。CMM 鉴别 5 个等级,一个组织必须也必然逐步经历这些等级才能建立起优秀的软件工程文化。没有合适基础的过程,在处于压力之下时(也正是最需要这些过程的时刻)不能起作用,也不能提供未来改进的基础。

等级 1 的组织,在尚未建立可重复过程(等级 2)之前,试图去实施已定义的过程,通常不会成功,因为项目经理会被进度和成本的压力压垮。看起来定义和实施工程过程似乎要比定义和实施管理过程容易(特别在技术人员眼中),但是如果没有管理规定,工程过程会成为进度和成本等压力的牺牲品。

一个尚无已定义过程作为基础就试图实施定量管理过程(等级 4)的组织通常是不成功的,因为没有已定义的过程就没有解释度量的共同基础。虽然对于一个个项目能采集数据,但几乎没有什么度量对本项目之外的其他项目有重大意义,也不能显著地增加组织对软件过程的理解。

一个尚无定量管理过程(等级 4)作为基础就试图实施优化过程(等级 5)的组织,由于对过程变更所产生的后果缺乏了解,多半会失败。在不能控制过程、使它处于统计意义上狭窄的范围内(即过程度量中仅有小的变化)的情况下,数据中有太多的噪声会导致不能客观地定义某项具体过程的改进是否有效。因为如果没有定量的依据,决策过程可能退化为主观臆想的争论。

软件能力成熟度等级的提高是一个循序渐进的过程,具有实施较高成熟度等级某过程的能力并不表示可以跳越成熟度等级。

8.2.3.6 关键过程域

图 8.2 表示,除等级 1 外 CMM 对每个成熟度等级都指明几个关键过程域,关键过程域指出为了达到某个成熟度等级所必须着手解决的相应问题,从而指明,组织为了改进软件过程应关注的过程域。

每个关键过程域包含一系列相关活动,当这些活动全部完成时,就能达到对增强过程能力至关重要的一组相应目标。目标表明每个关键过程域的范围、边界和意图。达到关键过程域目标的途径可能因项目而异,这是因为在应用领域或环境上有差异。一个组织要实现某个关键过程域,必须达到该关键过程域的全部目标。当一个组织的所有项目均已达到某个关键过程域的全部目标时,该组织已使以该关键过程域为特征的过程能力规范化了。

尽管其他问题也影响过程性能,但 CMM 只指出关键过程域,这是因为它们在改进组织软件过程能力上最有效。可以认为它们是达到一个成熟度等级的必要条件。图 8.2 结出了每个成熟度等级的关键过程域。为了达到某个成熟度等级,必须实现该等级上的全部关键过程域。

随着组织晋升到过程成熟度的更高等级,在各个关键过程域上应进行的具体实践将有所发展。例如,等级 2 上软件项目策划这个关键过程域所描述的项目估计能力中的许多项必须进化,以便能处理在等级 3,4,5 上可得到的、附加的项目数据。当采用已定义软件过程来管理项目时,等级 2 的软件项目策划及软件项目跟踪和监督就进化为等级 3 上的集成软件管理。

(1) 等级 2 上的关键过程域集中关注软件项目所关心的、与建立基本项目管理和控制有关的事情。

- **需求管理:** 目的是在顾客和软件项目之间建立对顾客需求的共同理解。与顾客的协议是策划和管理软件项目的基础。对与顾客关系的控制遵循有效的更改控制过程。

- **软件项目策划:** 目的是制定进行软件工程和管理软件项目的合理的计划。这些计划是管理软件项目的必要基础。没有切合实际的计划不可能实施有效的项目管理。

- **软件项目跟踪和监督:** 目的是建立适当的对实际进展的可视性,使管理者在软件项目实施情况显著偏离软件计划时能采取有效的措施。

- **软件子合同管理:** 目的是选择合格的软件分包商,并有效地管理它们。把用于基本管理控制的需求管理、软件项目策划以及软件项目跟踪和监督等关键过程域所关注的事情与软件质量保证和软件配置管理等关键过程域中必不可少的要求结合在一起,适当地实施于分包商。

- **软件质量保证:** 目的是给管理者提供对于软件项目正在采用的过程和正式构造的产品的恰当的可视性。软件质量保证是绝大多数软件工程过程和管理过程不可缺少的部分。

- **软件配置管理:** 目的是在项目的整个软件生存周期中建立和维护软件产品的完整性。软件配置管理是绝大多数软件工程过程和管理过程不可缺少的部分。

(2) 等级 3 的关键过程域既涉及项目,又涉及组织,因为组织建立起了对所有项目都有效的、使软件工程过程和管理过程规范化的基础设施。

- **组织过程焦点:** 目的是规定组织在改进其整体软件过程能力的软件过程活动方面的责任。组织过程焦点活动的主要结果是一组软件过程财富。软件过程财富是指在进行过程定义和维护活动方面有用的实体的集合,一般包括组织的标准软件过程、准予使用的软件生存周期的描述、对组织的标准软件过程进行剪裁的指南和准则、组织的软件过程数据库和软件过程有关的文档库。软件过程财富在组织过程定义中加以

描述。正如综合的软件管理中所述，这些财富供软件项目使用。

- **组织过程定义：**目的是开发和保持一组便于使用的软件过程财富，以便使项目的过程实施能得到改进，并且为组织能获得积累性的长期效益奠定基础。这些财富提供了一组稳定的基本原则。

- **培训大纲：**目的是培育个人的技能和知识，使他们能有效地执行任务。尽管培训是组织的责任，但是软件项目应该确定他们所需要的技能，如果项目需求独特，那么该项目应提供所需要的培训。

- **集成软件管理：**目的是将软件工程活动和管理活动集成为一个协调的、已定义的软件过程，该过程是剪裁组织的标准软件过程和组织过程定义中所描述的有关过程财富而得到的。剪裁的根据是项目的经营环境和技术需要。集成软件管理是从等级 2 的软件项目策划和软件项目跟踪和监督进化而来的。

- **软件产品工程：**目的是一致地执行妥善定义的工程过程。为了能有效地生产正确的、一致的软件产品，该工程过程集成了全部软件工程活动。软件产品工程描述项目的技术活动，例如需求分析、设计、编码和测试。

- **组际协调：**目的是为软件工程组积极参与其他工程组工作制定一种方法，使得项目能更有效地满足顾客的需求。组际协调是集成软件管理的一个方面，它涉及多学科，延伸到软件工程之外；不仅应该集成软件过程，而且软件工程组和其他组之间的相互作用也必须加以协调和控制。

- **同行专家评审：**目的是及早且高效地消除软件工程产品中的缺陷。一个重要的必然结果是增强对软件工程产品的了解和对可预防的缺陷的了解。同行专家评审是一种重要而又有效的工程方法，在软件产品工程中，可通过设计评审、结构化审查、或者一些其他的学院式的评审方法来实施同行专家评审。

(3) 等级 4 上的关键过程域的关注焦点是建立起对软件过程和正在构造的软件工程产品的定量了解。该等级上的两个关键过程域一定量过程管理和软件质量管理一是互相紧密依赖的。

- **定量过程管理：**目的是定量地控制软件项目的过程性能。软件过程性能表示遵循一个软件过程所得到的实际结果。焦点是在一个可测的稳定的过程范围内鉴别出变化的特殊原因，并且在适当时改正促使出现瞬时变化的环境。定量过程管理为组织过程定义、集成软件管理、组际协调和同行专家评审等实践附加上内容丰富的测量计划。

- **软件质量管理。**目的是建立对软件产品质量的定量了解和实现特定的质量目标。软件质量管理对软件产品工程中所描述的软件工程产品实施内容丰富的测量计划。

(4) 等级 5 上的关键过程域包括为了实施连续不断的和可测的软件过程改进，组织和项目都必须解决的问题。

- **缺陷预防：**目的是鉴别缺陷的原因并防止它们再次出现。软件项目分析缺陷、鉴别其原因并更改项目定义软件过程。将具有普遍价值的过程变更通知给其他软件项目。

- **技术变更管理：**目的是识别出能带来好处的新技术(即工具、方法和过程)，并以有序的方式引进这些新技术‘技术变更管理的关注焦点是在不断变化的环境中高效率地进行创新。

- **过程变更管理：**目的是为了改进软件质量、提高生产率和缩短产品开发周期，持续不断地改进组织中所采用的软件过程。过程变更管理既采用缺陷预防的增量式改进，又采用技术变更管理的创新式改进，并使得整个组织可以共享这些改进。

上述关键过程域总共 18 个，所包含的过程分为 3 类，见表 8.4。

8.2.3.7 关键实践

前面所述关键过程域的目标概括了该关键过程域的关键实践、，为了达到有关目标，必须实施相应的关键实践。每个关键过程域所包含的关键实践涉及 5 个方面，这 5 个方面是：执行约定、执行能力、执行的活动、测量和分析，以及验证实施，称之为 5 个共同特征，关键过程域所包含的关键实践全部按这 5 个共同特征加以组织，如图 8.3 所示。

共同特征是表明一个关键过程域的实施和规范化是否有效、可重复且持久的一些属性。5 个共同特征的含义如下。

- **执行约定：**描述一个组织在保证将过程建立起来并持续起作用方面所必须采取的行动。执行约定一般包含制定组织的方针和规定高级管理者的支持。

表 8.4 关键过程域的过程分类

等级	过程分类	管理 (软件项目策划等)	组织 (高级管理者评审等)	工程(需求分析、设计、编码、测试等)
5 优化等			技术变更管理	
			过程变更管理	缺陷预防
4 已定量管理级		定理过程管理		软件质量管理
3 已定义级		集成软件管理 组际协调	组织过程焦点 组织过程定义 培训大纲	软件产品工程 同行专家评审
2 可重复级		需求管理 软件项目策划 软件项目跟踪和监督 软件子合同管理 软件质量保证 软件配置管理		
1 初始级		无序过程		

• **执行能力**: 描述为了能实施软件过程, 项目或组织中必须存在的先决条件。执行能力一般包括资源、组织机构和培训。

• **执行的活动**: 描述为实现一个关键过程域所必须的角色和规程(即描述必须由谁作什么)。执行的活动一般包括制定计划和规程, 执行计划, 跟踪执行情况, 必要时采取纠正措施。

• **测量和分析**: 描述对过程进行测量和对测量结果进行分析的需要。测量和分析一般包括一些为了确定所执行活动的状态及有效性而能采用的测量和分析。

• **验证实施**: 描述保证遵照已建立的过程进行活动的措施。验证一般包括管理者和软件质量保证部门所作的评审和审计。

执行的活动这一实践描述为了建立过程能力必须作些什么。而其他实践则作为整体形成一个基础, 使组织能将执行的活动所描述的实践规范化。

关键实践, 无论属于哪个共同特征, 描述的都是对关键过程域的有效实施和规范化贡献最大的基础设施或活动。一般说来, 每个关键过程域的每个共同特征都包含一项到十几项关键实践; 每项关键实践又可能另有若干子实践(或称下级实践), 用来帮助确定关键实践是否得到满意的实施。例如, 等级 2 的一个关键过程域“软件项目策划”所包含的关键实践总共有 25 个, 其中

- 执行约定有 2 个, 例如, 约定 1 是指定项目软件经理负责协商约定和制定项目软件开发计划。
- 执行能力有 4 个, 例如, 能力 1 是对软件项目有文档化的经过批准的工作说明。
- 执行的活动有 15 个, 例如, 活动 5 是识别或确定具有可管理规模的预定阶段的软件生存周期模型。
- 测量和分析有 1 个, 即进行测量, 并用测量结果来确定软件项目策划活动的状态。
- 验证实施有 3 个, 例如验证 1 是高级管理者定期参与评审软件项目策划活动。

通过实施这些关键实践, 就能实现软件项目策划这个关键过程域的下列 3 个目标:

- 对策划和跟踪软件项目所用的软件估计已建立文档。
- 软件项目的活动和约定是有计划的并已建立文档。
- 受影响的组织和个人都同意他们关于软件项目的约定。

8.2.3.8 CMM 的应用

CMM 有两个基本用途: 软件过程评估和软件能力评价。

软件过程评估, 目的是确定一个组织的当前软件过程的状态, 找出组织所面临的急需解决的与软件过程有关问题, 进而有步骤地实施软件过程改进, 使组织的软件过程能力不断提高。

软件能力评价, 目的是识别合格的能完成软件工程项目的承制方, 或者监控承制方现有软件工作中软件过程的状态, 进而提出承制方应改进之处。

由于软件过程评估和软件能力评价是有着不同目的两种应用, 因此所用的具体方法有明显差异。但是两者都以 CMM 模型及其衍生产品为基础, 实施的基本步骤也一致。

1. 软件过程评估和软件能力评价的基本方法和步骤

软件过程评估关注一个组织的软件过程有哪些需改进之处及其轻重缓急。评估组采用 CMM 来指导他们进行调查、分析和排优先次序。组织可利用这些调查结果, 参照 CMM 中的关键实践所提供的指导, 规划本组织软件过程的改进策略。

软件能力评价关注一个特定项目在进度要求和预算限制内构造出高质量软件所面临的风险。在采购过程中可以对投标者进行软件能力评价。评价的结果可用于确定在挑选承制方方面的风险; 也可对现有的合同进行评价以便监控承制方的过程实施, 从而识别出承制方的软件过程中潜在的可改进之处。

CMM 为进行软件过程评估和软件能力评价建立了一个共同的参考框架, 作为评估软件过程成熟度的根据。

图 8.4 概要地描述评估和评价中的共同步骤。

(1) **建立一个小组**。该小组的成员应是具有丰富软件工程专业知识和管理知识的专业人员。对该小组进行 CMM 基本概念和评估或评价方法细节方面的培训。

(2) **填写提问单**。让待评估或评价单位的代表完成成熟度提问单的填写和其他诊断工具的要求。

(3) **进行响应分析**: 评估或评价组对提问单响应进行统计分析, 定义必须作进一步探查的区域。待探查的区域与 CMM 的关键过程域相对应。

(4) **进行现场访问**。访问被评估或评价单位的现场。评估或评价组根据响应分析的结果, 召开座谈会,

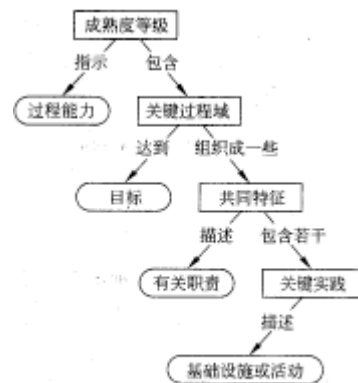


图 8.3 CMM 的结构

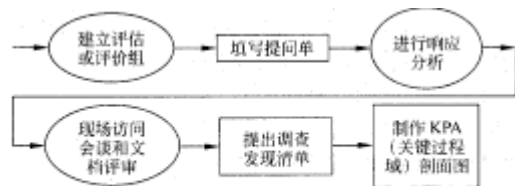


图 8.4 软件过程评估和软件能力评价的共同步骤

进行文档复审，以便了解该现场所遵循、的软件过程。评审或评价组成员在提问、倾听、复审和综合各种信息时应以 CMM 中的关键过程域和关键实践为指导。该组运用专业性的判断确定现场关键过程域的实施是否满足相关的关键过程域的目标。当 CMM 的关键实践与现场的实践间存在明显差异时，必须用文档记下对此关键过程域作出判断的理论依据。

(5) **提出调查发现清单。**在现场工作阶段结束时，评估或评价组产生一个调查发现清单，明确指出该组织软件过程的强项和弱项。在软件过程评估中，该调查发现清单作为提出过程改进建议的基础；在软件能力评价中调查发现清单作为软件采购单位所作风险分析的一部分。

(6) **制作关键过程域 (KPA) 剖面图。**评估或评价组制作一份关键过程域剖面图，指出该组织已满足和尚未满足关键过程域目标的区域。一个关键过程域可能是已满足要求的，但仍存在一些相关的问题，如果未发现或未指出这些问题，就会妨碍实现该关键过程域的某个目标。

总之，软件过程评估和软件能力评价方法两者的共同点如下

- 采用成熟度提问单作为现场访问的出发点。
- 采用 CMM 作为指导现场调查研究的导引图。
- 利用 CMM 中的关键过程域生成明确地指出软件过程强项和弱项的调查发现清单。
- 在对关键过程域目标满足情况进行分析的基础上，衍生出一个剖面。
- 根据调查发现清单和关键过程域剖面，向合适的对象提出结论意见。

2. 软件过程评估和软件能力评价之间的差异

尽管软件过程评估和软件能力评价有上述相似之处，但由于在动机、目的、输出和结果的所有权等方面均不同，导致二者在会谈目的、询问的范围、所采集的信息和结果的表示方式上不同，而且所采用的详细规程和培训要求也不一样。

软件过程评估是在开放、合作的环境中进行的，评估目的在于暴露问题和帮助经理和工程师们改进他们的软件过程，评估的成功取决于管理者和专业人员对改进软件过程的支持。评估过程中虽然提问单是个重要工具，但更重要的是通过各种会谈了解组织的软件过程。评估的结果除了识别组织所面临的软件过程问题外，最有价值的还是明确软件过程的改进途径，制定进一步的行动计划，使全组织关注改进过程，增强改进的动力和热情。

软件能力评价是在更像审计的环境中进行。评价的目的与金钱密切相关，因为评价组的推荐性意见将影响承建方挑选或资金设置。评价过程的重要是复审已文档化的审计记录，这些记录能提示组织实际执行的软件过程。

3. 其他应用

除了上述两个基本用途外，CMM 在过程改进方面还有一些其他用途，主要是组织内负责软件过程改进的机构，例如软件工程过程组 (SEPU)，在策划改进措施、实施措施计划和定义过程时可以充分利用 CMM。

- 在策划改进措施期间，软件工程过程组可将 CMM 中关键过程域的目标与组织的当前实践相比较，仔细分析与公司目标、管理优先级、实践运行的层次、实施每项实践对组织的价值、组织在其文化背景下实施某项实践的能力等方面有关的关键实践。

- 软件工程过程组必须确定需要作哪些过程改进，如何实现变更以及如何获得所需要的支持。CMM 可以给有关过程改进的讨论提供一些初步的议题，帮助揭示与通用软件工程实践完全不同的前提条件。

- 在实施措施计划时，软件工程过程组可用 CMM 和关键实践来构造部分可操作的措施计划和定义软件过程。

8. 2. 3. 9 软件过程成熟度提问单

美国软件工程研究所给出的与 SEI CMM 1. 1 相应的软件过程成熟度提问单有两部分内容，一部分是对填写提问单的人员本身的背景作调查，这种调查有助于理解对提问单的回答；另一部分是关于软件实践的。关于软件实践的提问按关键过程域分组。在每组提问之前有一段文字，描述相关的关键过程域，还有相关的术语定义。限于篇幅，这里仅给出软件项目策划这一个关键过程域的软件实践方面的提问作为示例。

在给出提问单前需要说明，软件项目策划的目的是为进行软件工程活动和管理软件项目制定合理的计划。软件项目策划包括估计待完成的工作，建立必要的约定和制定进行该工作的计划；对每个提问的回答可能是“是”、“否”、“不适用”或“不知道”4 种之一；另外还可以在每个提问的回答下面写一些评论这组提问中有如下几个术语：

- **约定 (commitment)。**自由采用的、可视的、期待各方遵守的协议。
- **事件驱动的评审和活动 (event-driven review/ activity)。**根据项目中某事件的发生而进行的评审或活动 (例如：一个软件生存期阶段的完成时的评审)。

• **方针(policy)**。一种指导原则，一般由高级管理者制定，组织或项目在确定决策时必须遵守。

• **软件计划(software plans)**。如何进行软件开发和(或)维护活动的计划集合，既可是正式计划，也可是非正式计划。这类计划的例子有：软件开发计划、软件质量保证计划、软件配置管理计划、软件测试计划、风险管理计划和过程改进计划。

下面是摘录的软件项目策划这个关键过程域的提问单。

8.2.4 软件过程评估的国际标准概述

8.2.4.1 软件过程评估国际标准的制定

1991年国际标准化组织(ISO)决定调研国际社会对软件过程评估标准的需求，1993年ISO决定组织制定软件过程评估标准，1995年完成了工作草案并开始试用，在此基础上进行了修改，1996年完成了工作草案1.0版，确定标准号为ISO/IEC 15504，名称为软件过程评估(software process assessment, SPA)，并开始第2批试用，1998年10月发表了第2批试用的总结，即ISO/IEC TR 15504 Information technology—software process assessment 1998-08-15(以上称为软件过程评估标准)。

该标准的目的是有3点：

• 帮助软件开发组织了解本组织的过程状态，以便进行改进。

• 帮助软件开发组织确定其过程对满足某特定要求的合适程度。

• 帮助人们确定某个软件开发组织对开发某具体产品的合适程度。

为了达到上述目的，人们要求该标准提供国际公用的过程评估模型，使大家对通过过程评估进行过程改进和能力评定有共同的理解，便于使用和管理，并鼓励对现有的几种软件过程评估模型取长补短。

8.2.4.2 软件过程评估标准的组成

软件过程评估标准包含9个部分，这9部分中第2,3,9部分是标准性的，其他都是参考性的，如图8.5所示。

• **部分1：概念和引导指南(参考件)**。描述该标准各部分的关系，对标准的选用给出指导，解释标准的要求和对进行评估的适用性。

• **部分2：过程和过程能力的参考模型(标准件)**。定义了一个二维参考模型，描述过程评估中所用的过程和过程能力。该参考模型用过程的目的和输出来定义一组过程，并给出借助于过程属性的评估来评价过程能力的框架，这些过程属性构成能力等级。定义了各种评估方法与参考模型的相容性要求。

• **部分3：进行评估(标准件)**。规定了对评估方法的要求，以保证评估输出可重复、可靠且一致。

• **部分4：进行评估的指南(参考件)**。给出了对进行软件过程评估的指导，解释在不同的评估环境下部分2和部分3的要求。包括对文档化的评估过程、相容的评估模型和评估支持工具的选用指南。

• **部分5：评估模型和指示器指导(参考件)**。给出进行过程评估的样本模型，该模型以部分2中的参考模型为依据并直接与之相容。该样本模型包括了广泛的过程性能和能力的指示集。

• **部分6：评估人员资格指南(参考件)**。描述与进行过程评估相关的评估人员资格、教育、训练和经验，说明资格证明、教育、训练和经验的机制。

• **部分7：过程改进指南(参考件)**。说明如何定义评估输入和如何使用评估结果。给出了在各种情况下的过程改进应用示例。

• **部分8：供应者过程能力评定指南(参考件)**。说明为了确定过程能力如何定义评估输入和如何使用评估结果。阐述了简单情况和较复杂情况(例如未来能力)下的过程能力评定。既适用于组织内部自己进行过

提问：	回答：			
	是	否	不适用	不知道
1. 供策划和跟踪软件项目用的估计(例如：规模、成本和进度估计)是否已文档化？	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
评论：				
2. 软件项目计划是否将拟进行的活动和对软件项目所作的约定文档化？	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
评论：				
3. 所有受影响的组和个人对他们有关软件项目的约定是否同意？	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
评论：				
4. 项目是否遵循一个书面的用于策划软件项目的组织方针？	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
评论：				
5. 是否为策划软件项目提供足够的资源(例如，资金和有经验的个人)？	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
评论：				
6. 是否用测量来定义软件项目活动策划的状态(例如，项目策划活动里程碑的完成情况与计划相比较)？	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
评论：				
7. 项目经理是否既定期地又事件驱动地评审软件项目的策划活动？	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
评论：				

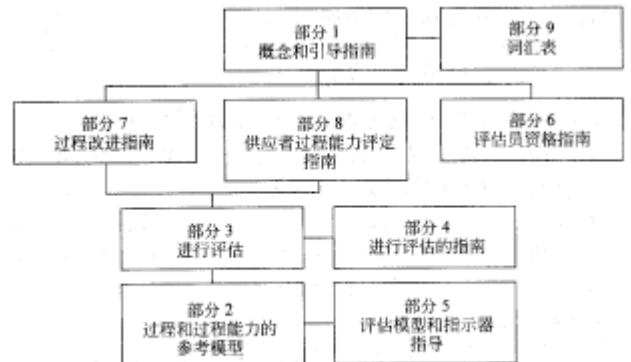


图 8.5 软件过程评估标准的组成部分

程能力评定，也适用于采购者对（潜在）供应者进行过程能力评定。

• **部分 9：词汇表(标准件)**。集中了专为本标准定义的所有术语。

部分 2 一部分 8 这 7 个部分之间的关系如图 8.6 所示。

图 8.6 表明过程评估可用于过程改进或能力评定。对这两种应用的指导分别在部分 7 和部分 8 中。进行评估时要求所用的模型与部分 2 中的参考模型相容；部分 5 给出了一种样本模型。评估过程必须文档化，并应以符合部分 3 规定要求的方法为基础，遵照部分 4 提供的指导。有资格的评估员负责确保评估是一致的；关于评估员的技能和资格在部分 6 中有明确阐述。

8.2 . 4.3 参考模型

参考模型由二维组成，其结构见图 8. 7。一维是过程维，用可测量的主要过程目标来描述，另一维是过程能力维，以适用于任何过程的一系列过程属性来描述，这一系列过程属性表示管理一个过程和改进过程实施能力所必需的可测量的特性。

1.过程维

过程维包含 5 个过程类，共 40 个过程，分别属于 3 个软件生存周期过程组，见表 8. 5。

2.过程能力维

参考模型的过程能力维为任何过程的过程能力定义一个测量标准。过程能力分为 6 级，从不完备(0 级)，经过已实施(1 级)、已管理(2 级)、已建立(3 级)、可预测(4 级)，到优化(5 级)，随着等级的提高，所实施过程的能力也逐步增长。能力的量度以一组过程属性(PA)为基础，过程属性用来确定某个过程是否达到了某个规定的能力。每一个过程属性测量过程能力的一个具体方面。属性都用百分比表示，从 0 到 1 表示该属性所达到的程度，即指示值。过程的能力等级和相应的过程属性见表 8.6。

在 0 级，过程没有得到实施，不能给出过程的输出，几乎或完全没有能证明达到规定属性的证据。

在 1-5 级，所实施的过程给出过程输出。证明各级成绩的属性是从 1 级至该级的所有过程属性。过程属性评定的等级分 4 档，表示对该过程属性的规定能力所达到的等级。

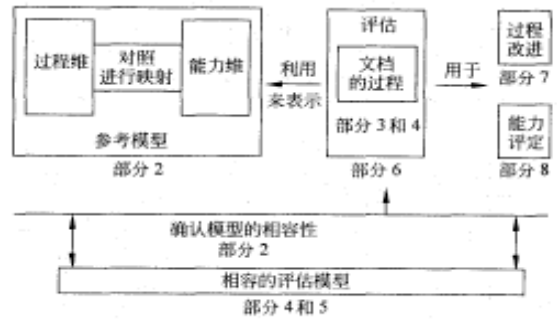


图 8.6 ISO/IEC TR 15504 诸部分的关系概观

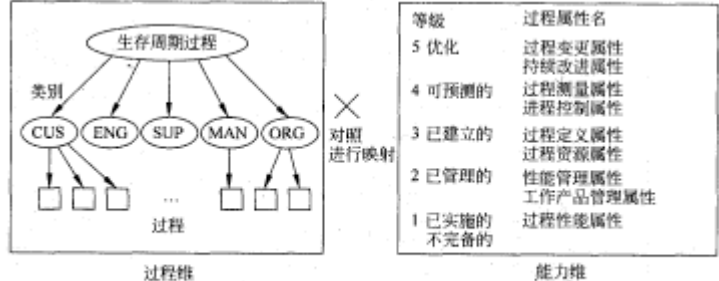


图 8.7 参考模型的二维结构

表 8.5 过程和过程类别

软件生存周期过程组	过程类	过程名*	子过程名**
基本过程组	CUS (顾客供方过程类)	CUS.1 获取(基本的)	CUS.1.1 获取准备 CUS.1.2 对供方的选择 CUS.1.3 对供方的监督 US.1.4 顾客验收
		CUS.2 供应(基本的)	
		CUS.3 需求推导(新的)	
		CUS.4 操作(扩展的)	CUS.4.1 运行使用(扩展的) CUS.4.2 顾客支持(扩展的)
	ENG (工程过程类)	ENG.1 开发(基本的)	ENG.1.1 系统需求分析和设计 ENG.1.2 软件需求分析 ENG.1.3 软件设计 ENG.1.4 软件构造 ENG.1.5 软件集成 ENG.1.6 软件测试 ENG.1.7 系统集成和测试
		ENG.2 系统和软件维护(基本的)	
支持过程组	SUP (支持过程类)	SUP.1 文档编制(扩展的)	
		SUP.2 配置管理(基本的)	
		SUP.3 质量保证(基本的)	
		SUP.4 验证(基本的)	
		SUP.5 确认(基本的)	
		SUP.6 联合评审(基本的)	
		SUP.7 审计(基本的)	
		SUP.8 问题解决(基本的)	
组织过程组	MAN (管理过程类)	MAN.1 管理(基本的)	
		MAN.2 项目管理(新的)	
		MAN.3 质量管理(新的)	
		MAN.4 风险管理(新的)	
组织过程组	ORG (组织过程类)	ORG.1 组织调整(新的)	
		ORG.2 改进过程(基本的)	ORG.2.1 过程建立 ORG.2.2 过程评估 ORG.2.3 过程改进
		ORG.3 人力资源管理(扩展的)	
		ORG.4 基础设施(基本的)	
		ORG.5 测量(新的)	
		ORG.6 重用(新的)	

注：总共分 5 种过程类型，其中在 * 列中有 3 种：基本的，指与 ISO/IEC12207 中的相应过程的含义相同；扩展的，为 ISO/IEC 12207 中相应过程的扩展；新的，是 ISO/IEC 12207 中所没有的。在 ** 列中有两种：子过程是 ISO/IEC 12207 中相同过程的一个或多个活动的组织；扩展的子过程是 ISO/IEC 12207 中相同过程的一个或多个活动，但增加了内容。

•N: 未达到。所评估过程的属性值为 0%-15%，几乎或完全没有证据证明规定属性的成绩。

•P: 部分达到。所评估过程的属性值为 16%-50%，有证据证明有良好的系统化方法来达到规定的属性，成绩的某些侧面也许不可预测。

•L: 大部分达到。所评估过程的属性值为 51%-85%，有证据证明对规定的属性有良好的系统化方法并取得了显著的成绩，过程的性能在某些领域或某些工作单位可能有差别。

•F: 充分达到。所评估过程的属性值为 86%-100%，有证据证明有完备的系统化方法来充分达到规定的属性，在规定的各组织单位不存在明显的弱点。

对被评估的每一个过程属性必须用上述属性等级给一个评分，一个过程的过程属性评分集构成该过程的过程剖面。评估输出包括所有被评估过程的过程剖面集。一个过程所达到的能力等级依据该过程的属性评分值按表 8.7 所示过程能力等级模型评定。

8.2.4.4 评估框架

1.过程评估环境

过程评估环境大致如图 8.8 所示。

过程评估可能是在过程改进期间或作为过程能力评定工作的一部分进行的。无论是哪种情况，正式进入评估过程之前必须有委托者的约定；对于评估输入应考虑图 8.8 所示评估输入的 6 个方面；根据选用的评估模型对选定的若干过程进行评估；选用的评估模型必须与参考模型相容；评估过程至少包含 5 项活动，如图 8.8 所示；评估输出包括被评过程的过程剖面集，也可能还有每一个被评过程的能力等级。

图 8.8 中左下方的指示集包含过程性能指示和过程能力指示，这些指示一般采用被客观证明了的工作产品特性和与被评过程相关的实践特性的形式。完备的过程评估模型包含所用的各种指示的细节。

2.过程改进环境

过程改进环境大致如图 8.9 所示。

在经营环境中要针对组织的特定需要和经营目标，通过对资源、文化等限制的明确说明和理解，来实现软件过程改进。

3.过程能力评定环境

过程能力评定以过程评估为基础，过程能力评定环境如图 8.10 所示。采购者说明规定的要求；规定的要求转换为目标能力(表示要求达到的过程能力)和过程评估输入(指出评估范围)；供应者可以提出一个推荐的能力，即有关组织单位各相关过程的一组能力等级。

4.对评估过程的要求

为使评估结果客观、公正、一致、可重复且有代表性，软件过程评估标准部分 3 规定了进行评估时必须满足的要求，其中对图 8.8 中涉及的评估输入的定义、评估责任、评估过程活动和评估输出的

表 8.6 过程的能力等级和过程属性

能力等级	名称	过程属性
1 级	已实施的过程	PA1.1 过程性能属性
2 级	已管理的过程	PA2.1 性能管理属性
		PA2.2 工作产品管理属性
3 级	已建立的过程	PA3.1 过程定义属性
		PA3.2 过程资源属性
4 级	可预测的过程	PA4.1 过程测量属性
		PA4.2 过程控制属性
5 级	优化过程	PA5.1 过程变更属性
		PA5.2 持续改进属性

表 8.7 能力等级评定

等级	过程属性	评定值
1 级	过程性能	大部分或充分
2 级	过程性能	充分
	性能管理	大部分或充分
	工作产品管理	大部分或充分
3 级	过程性能	充分
	性能管理	充分
	工作产品管理	充分
	过程定义和剪裁	大部分或充分
	过程资源	大部分或充分
4 级	过程性能	充分
	性能管理	充分
	工作产品管理	充分
	过程定义和剪裁	充分
	过程资源	充分
	过程测量	大部分或充分
	过程控制	大部分或充分
5 级	过程性能	充分
	性能管理	充分
	工作产品管理	充分
	过程定义和剪裁	充分
	过程资源	充分
	过程测量	充分
	过程控制	充分
	过程变更	大部分或充分
	持续改进	大部分或充分

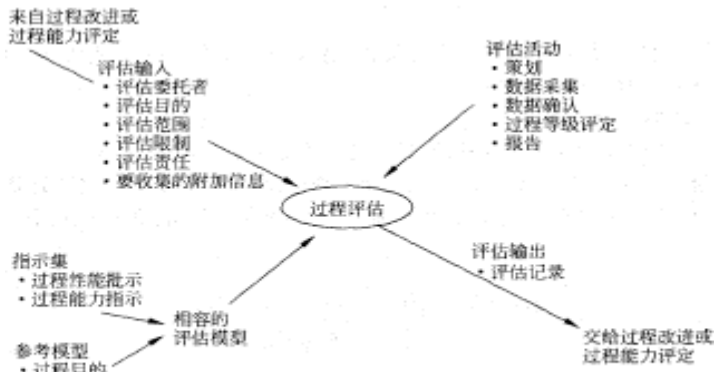


图 8.8 过程评估的环境

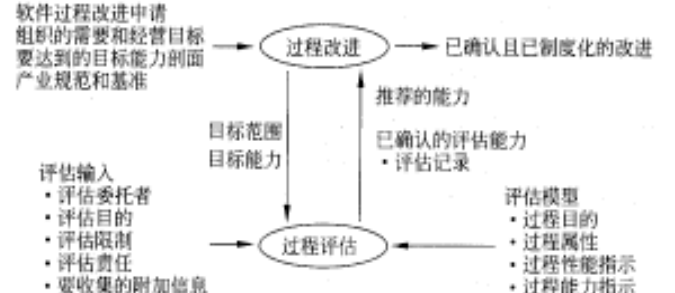


图 8.9 过程改进的环境

记录进行了详细描述。总的说来, 评估应满足以下要求:

- 使用至少要满足部分 3 规定要求的评估过程。
- 以与部分 2 定义的参考模型相容的评估模型为基础。
- 运用全面的关于过程性能和过程能力的指示值集合。
- 利用部分 2 定义的过程属性等级模式产生过程剖面。
- 有客观证据证明上述条件已经满足。

过程性能和过程能力的指示是判断过程属性等级的客

观基础, 也是评估结果可以比较的基础。部分 3 规定了关于指示的 3 条要求:

- 对评估范围内的所有过程一个相容的模型必须包含全面覆盖参考模型两维的指示集。
- 评估期间必须使用指示值来支持评估员在评定过程属性中的判断。
- 必须记录并维护以指示值为基础的证据。

8.2.4.5 软件过程评估标准的特点

与 CMM1.1 相比, 软件过程评估标准参考模型的基本应用目的、基本构成部分、基本原理以及能力等级的含义均与 CMM 相似; 软件过程评估标准不仅吸收了 CMM 的主要思想, 还参考了其他类似工作, 尤其是欧洲的 BOOTSTRAP 项目等的成果, 并注意了克服 CMM1.1 存在的一些缺陷, 因此与 CMM1.1 有一些重要差别, 例如:

- 参考模型与 CMM 不同, 它明确给出过程维, 其中所包括的过程都必须实施, 否则, 就表明未按良好的软件工程开展基本活动, 也就谈不上有什么软件过程能力, 所以在这种情况下软件过程能力是零级; 仅当实施了过程维的各个过程, 才能通过过程能力维的过程属性, 分析评定软件过程能力等级。而 CMM 则没有定义类似过程维的过程。

- ISO/IEC T8 15504 所确定的评估对象是过程维的各个过程, 给出这些过程的能力等级; 而 CMM 1.1 的应用对象是项目或组织, 而不是过程, 给出一个项目或一个组织的整体软件过程成熟度等级。

- 软件过程评估标准有一个明确意图, 就是作为 ISO 9000 族标准的一个支持标准, 因此, 其内容与 ISO 9000 标准相互协调; 而 CMM 1.1 没有这些考虑。

- 软件过程评估标准, 特别是其第二部分, 直接对准 ISO/IEC 12207-1995K 信息技术软件生存周期过程》, 并在此基础上作了必要的补充和扩展, 而 CMM1.1 没有这样考虑。(不过应当指出, 从 CMM2.0 版本的草案看, ISO/IEC 12207-1995 所描述的软件生存周期过程在 CMM2.0 中都有适当的阐述, 见表 8.8)。

8.3 软件配置管理

软件配置管理是软件管理的重要内容。近年来, 软件项目的规模越来越大, 复杂性越来越高, 由于管理上失误给我们的教训也越来越深刻。这都使得人们不得不重视配置管理问题。许多软件工程标准中都对软件配置管理作了明确的规定, 提醒我们必须对配置管理给予充分的重视。

软件开发过程中的变更以及相应的返工会对产品的质量有很大的影响。有统计表明, 变更及其返工可能耗费 50% 的开发工作量, 如果不从配置管理方面加以控制, 必将导致严重的后果。软件配置管理的一个重要内容就是对变更加以控制, 使变更对成本、工期和质量的影响降到最小。

本章在讨论软件配置管理概念的基础上, 介绍了配置管理计划应包括的内容, 接着重点讨论了配置标识、变更管理、配置审核等问题, 另外还涉及到版本管理、配置状态报告以及采用配置管理工具的问题。

8.3.1 软件配置管理的概念

8.3.1.1 软件配置项(software configuration item)

随着软件开发工作的开展, 会得到许多工作产品或阶段产品, 还会用到许多工具软件, 可能是外购软件, 也可能是用户提供的软件。所有这些独立的信息项都要得到妥善的管理, 决不能出现混乱, 以便于在提出某些特定的要求时, 将它们进行约定的组合来满足使用的目的。

这些信息项是配置管理的对象, 称为软件配置项。在表 8.9 中列举了若干类软件配置项及其生成的阶段。

如果说配置项是一个独立存在的信息项, 我们可以把它看成一个元素。单独的一个元素发挥不了什么作用, 但随着工作的进展, 出于不同的要求, 需要将这些元素进行不同的组合。软件配置是一个软件产品在生存期各个阶段的不同形式(记录特定信息的不同媒体)和不同版本的程序、文档及相关数据的集合, 或者说是配置项的集合。

这里以交付给不同用户的某一软件产品为例, 进一步说明软件配置的含意。如果我们开发的软件产品是具有一定功能和性能的初始系统, 那么最终的产品应能满足用户的需求。为此, 必须认真研究用户的真正需求。经调查, 了解到“用户 1”代表了一些用户, 这个用户群使用的计算机为“机型 1”, 所用的操作



图 8.10 过程能力评定

系统是“操作系统 1”；而“用户 2”所代表的用户群使用“机型 2”和“操作系统 2”(参看图 8. 11)。就是说，不同用户有着不同的工作环境，我们的软件产品必须考虑到这些差异，并且充分地使其满足各个用户的使用要求。为做到这一点，产品的设计可能作成如图 8. 12 所示的安排：两类产品分别针对两个用户群，产品内部设计的模块(按上面的说法是配置项)如下。

- 用户 1：采用 A、B、C、D、E 和 F 模块。
- 用户 2：采用 A、B、C、D、E 和 G、H 模块。

两者的差别不仅表现在一个含有 F，另一个含有 G 和 H，而且即使两者的 A 在逻辑上是同一个内容，但在物理上仍然可能因两类用户需求的不同而有差异，例如，两个 A 分别以不同的媒体出现。

为实现这两种不同的软件配置，在实际工作中，我们完全可以将各个配置项分别开发出来，再根据需要，组合成针对用户使用需求的不同产品，正如图 8. 13 所示。

8.3.1.2 软件配置管理

(software configuration management)

1.什么是软件配置管理

软件工程项目的实践表明，软件过程会出现许多软件配置项，表 8. 9 只是从类别上加以区分，列举了一部分配置项，如果再加上版本的不同，就会有更多的配置项，对于它们如果没有一套科学的办法加以管理就会出现各种差错和混乱。

关于什么是软件配置管理，这里引用几种说法。

① 国际标准 ISO 9000-3: 19970

配置管理是一个管理学科，它对配置项(包括软件项)的开发和支持生存期给予技术上和管理上的指导。配置管理的应用取决于项目的规模、复杂程

表 8. 8 ISO 12207, 软件过程评估标准与 CMM 在过程级上的对照

ISO/IEC 12207	ISO/IEC 15504	CMMv1. 1	CMMv2 草案
5 基本生存周期过程			
5.1 获取过程	CUS. 1 获取过程	软件子合同管理	软件获取管理
5.2 供应过程	CUS. 2 供应过程 ¹	(软件项目策划, 软件项目跟踪和监督, 软件产品工程)	(软件项目策划, 软件项目控制, 软件产品工程)
	CUS. 3 需求推导过程		软件产品工程, 活动 2
5.3 开发过程	ENG. 1 开发过程	软件产品工程	软件产品工程
5.3.1 过程实现	ENG. 1 开发过程	软件产品工程	软件产品工程
5.3.2 系统需求分析	ENG. 1. 1 系统需求分析和设计过程		(软件产品工程, 活动 2) ³
5.3.3 系统结构设计	ENG. 1. 1 系统需求分析和设计过程		(软件产品工程, 活动 2) ³
5.3.4 软件需求分析	ENG. 1. 2 软件需求分析过程	软件产品工程, 活动 2	软件产品工程, 活动 3
5.3.5 软件结构设计	ENG. 1. 3 软件设计过程	软件产品工程, 活动 3	软件产品工程, 活动 4
5.3.6 软件详细设计	ENG. 1. 3 软件设计过程	软件产品工程, 活动 3	软件产品工程, 活动 4
5.3.7 软件编码和测试	ENG. 1. 4 软件构造过程	软件产品工程, 活动 4	软件产品工程, 活动 5
5.3.8 软件集成	ENG. 1. 5 软件集成过程	软件产品工程, 活动 6	软件产品工程, 活动 6
5.3.9 软件鉴定测试	ENG. 1. 6 软件测试过程	软件产品工程, 活动 7	软件产品工程, 活动 7 和活动 8
5.3.10 系统集成	ENG. 1. 7 系统集成和测试过程	软件产品工程, 活动 6	软件产品工程, 活动 6
5.3.11 系统鉴定测试	ENG. 1. 7 系统集成和测试过程	软件产品工程, 活动 7	软件产品工程, 活动 6、活动 7 和活动 8
5.3.12 软件安装	CUS. 2 供应过程		软件产品工程, 活动 10
5.3.13 软件验收支持	CUS. 2 供应过程		软件产品工程, 活动 10 和活动 11
5.4 操作过程	CUS. 4 运行使用过程		软件产品工程, 活动 11
5.5 维护过程	ENG. 2 系统和软件维护过程		(软件产品工程, 活动 11) ⁴
6 支持生存周期过程			
6.1 文档编制过程	SUP. 1 文档编制过程	软件产品工程, 活动 8	软件产品工程, 活动 9
6.2 配置管理过程	SUP. 2 配置管理过程	软件配置管理	软件配置管理
6.3 质量保证过程	SUP. 3 质量保证过程	软件质量保证	软件质量保证
6.4 验证过程	SUP. 4 验证过程	同行评审; 软件产品工程, 活动 5 和活动 6	同行评审; 软件产品工程, 活动 6 和活动 7
6.5 确认过程	SUP. 5 确认过程	软件产品工程, 活动 5	软件产品工程, 活动 7 和活动 8
6.6 联合评审过程	SUP. 6 联合评审过程	软件项目跟踪和监督, 活动 13	软件项目控制, 活动 10
6.7 审计过程	SUP. 7 审计过程	(软件质量保证) ⁵	软件质量保证
6.8 问题解决过程	SUP. 8 问题解决过程	软件配置管理, 活动 5	软件配置管理, 活动 5
7 组织的生存周期过程			
7.1 管理过程	MAN. 1 管理过程 ⁶	软件项目策划; 软件项目跟踪和监督; 集成软件管理	软件项目策划; 软件项目控制; 集成软件管理
	MAN. 2 项目管理过程	软件项目跟踪和监督; 集成软件管理	软件项目策划; 软件项目控制; 集成软件管理
	MAN. 3 质量管理过程	软件质量管理	(统计过程管理) ⁷
	MAN. 4 风险管理过程	软件项目策划, 活动 13 软件项目跟踪和监督, 活动 10 集成软件管理, 活动 10	软件项目策划, 活动 11 软件项目跟踪和监督, 活动 8 集成软件管理, 活动 6 和活动 7
	ORG. 1 组织调整过程 ⁸		组织过程焦点; 组织软件财富通用性
7.2 基础设施过程	ORG. 4 基础设施过程	组织过程定义	组织过程定义
7.3 改进过程	ORG. 2 改进过程	组织过程定义	组织过程定义
7.4 培训过程	ORG. 3 人力资源管理过程	培训大纲	组织培训大纲
	ORG. 5 测量过程	测量和分析(公共特征)	测量和分析(公共特征), (组织过程性能)
	ORG. 6 重用过程		组织软件财富通用性
		需求管理	需求管理
		组间协调	项目接口协调
		同行评审	同行评审
		定量过程管理	统计过程管理
			组织过程性能
		缺陷预防	缺陷预防
		技术改变管理	组织过程和技术改革
		过程改变管理	组织改进部署

注：圆括号“()”中所列的关键过程区域与 ISO12207 中有关过程没有明显的直接关系，但经过判断或推理可以看出其间的关系。

1. 供应过程涉及向顾客提供满足协议要求的软件。虽然在 CMM 中没有明确地说明供应过程，但在若干关键过程域(KPA)中阐述了建立合同、开发软件 and 向顾客交付软件，这些都是这个过程的一些活动。
2. 虽然没有明确称为系统需求分析，但 PE. AC. 2(产品工程，活动 2)的实现常常就是如此。
3. 虽然没有明确称为系统需求分析，但 PE. AC. 2(产品工程，活动 2)的实现常常就是如此。
4. 一般说来，CMM 认为维护是一种特定环境，在这个环境中要实现所有适合的 KPA。在 PE. AC. 11(作为修复)的一些子实践中专门阐述了维护，以提供这个支持关键实践的完整描述。
5. SQA(软件质量保证)既包括质量保证，也包括审计。审计比 QA(质量保证)在很大程度上独立性更强。SQA 这个关键过程区域可以按独立的功能实现，也可以不按独立的功能实现。主要要求是客观的验证，而不是独立的验证。因此，在一个具体环境中，SQA 可以包括审计过程，也可以不包括审计过程。
6. 这是通用的策划和管理过程，要应用于所有的过程，而不是专门对项目的。
7. 等级 4 的过程和产品问题在 1. 1 版本中是分开阐述的，但在版本 2 中都综合在 SPM(统计过程管理)中了。
8. 组织调整过程的目的是确保有关人员共享公共的视线、文化和对经营目标的理解。

表 8.9 软件配置项的分类、特征和举例

分 类	特 征	举 例
环境类	软件开发环境或软件维护环境	编译器、操作系统、编辑器、数据库管理系统、开发工具(如测试工具)、项目管理工具、文档编制工具
定义类	需求分析与定义阶段结束后得到的工作产品	需求规格说明、项目开发计划、设计标准或设计准则、验收测试计划
设计类	设计阶段结束后得到的工作产品	系统设计规格说明、程序规格说明、数据库设计、编码标准、用户界面标准、测试标准、系统测试计划、用户手册
编码类	编码及单元测试结束后得到的工作产品	源代码、目标码、单元测试数据及单元测试结果
测试类	系统测试完成后的工作产品	系统测试数据、系统测试结果、操作手册、安装手册
维护类	进入维护阶段以后生成的工作产品	以上任何需要变更的软件配置项

序的风险大小。

②W. Babich 的解释。

软件配置管理能协调软件开发,使得混乱减少到最小。软件配置管理是一种标识、组织和控制修改的技术,目的是最有效地提高生产率。

③GB/T 11457:1995《软件工程术语》国家标准。

标识和确定系统中配置项的过程,在系统整个生存期内控制这些配置项的投放和更动,记录并报告配置的状态和更动要求,验证配置项的完整性和正确性。并对下列工作进行技术和行动指导与监督的一套规范:

- 对配置项的功能特性和物理特性进行标识和文件编制工作。
- 控制这些特性的更动情况。
- 记录并报告这些更动进行的处理和实现的状态。

综合以上几种对软件配置管理的解释,可以把软件配置管理概括为:它是采用技术手段和行政手段进行管理和监督的一套规范化方法;对配置项的功能特性和物理特性加以标识,并将其文件化;控制这些特性的变更;报告变更进行的情况和变更实施的状态以及验证与规定需求的一致性。

总之,软件配置管理主要是对软件生存期过程中的各种阶段产品和最终产品演化和变更的管理,它是软件质量管理的重要组成部分。如果从变更的意义讲,软件配置管理是要解决软件的变更标识、变更控制以及变更发布的问题。

2.软件配置管理的任务

为达到上述软件配置管理的要求,通常认为实施软件配置管理应完成以下几方面的任务:

- 制定软件配置管理计划。
- 确定配置标识规则。
- 实施变更控制。
- 报告配置状态。
- 进行配置审核。
- 进行版本管理和发行管理。

这里提到的几项配置管理任务是和国际标准 ISO/IEC 12207:1995 信息技术—软件生存周期过程》中所规定的软件配置管理过程的活动一致的。表 8.10 给出了该标准关于软件配置管理过程的活动、任务的摘要。

3.软件配置管理与软件开发过程

根据国际标准 ISO/IEC12207 提出的软件生存期过程,开发过程和配置管理是两个不同的、分别独立的过程。前者属于基本软件过程,后者属于支持软件过程,但两者不是完全无关的。

如前所述,软件配置管理的一项重要任务是实施变更管理,但在软件工程项目中往往存在着两类不同的变更。一类是开发阶段内部发生的变更,如程序员在编程时,发现前面已写的代码有问题,需要加以修改。这在开发过程中是最为常见的,可以认为是“常规的”变更,这类变更无需配置管理过问。另一类变更则有许多不同,它是开发过程解决不了的变更,例如,正在编程时出现了需求的变更,或是在系统测试时要对代码作变更等等。这样一类“非常规的”变更也并不少见。如果不对这类变更作特别的处理,就会发生软件产品内部的不一致、不完整等问题。事实上,

这正是配置管理要做的事。尽管这类变更的实施仍然是由开发人员去完成,但变更的评估和批准以及变更实施的控制都要由软件配置管理人员去做。因此,在一定意义上,开发过程应纳入配置管理过程的控制之下,特别是第二类变更是在受控条件下进行的。图 8.14 表明了配置管理与开发过程的关系。

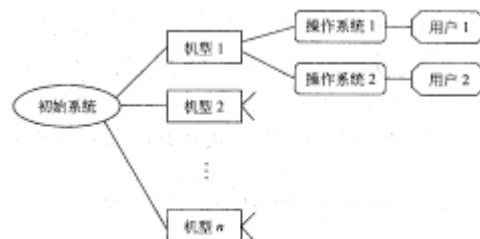


图 8.11 不同用户有自己的工作环境

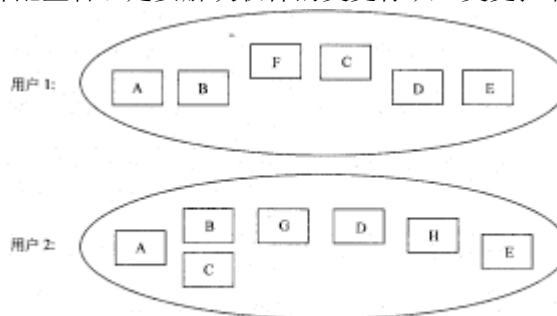


图 8.12 面对不同用户产品的不同配置

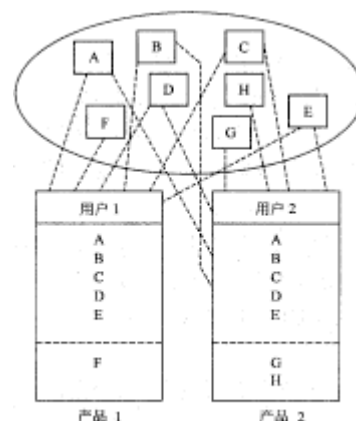


图 8.13 两个产品具有不同的配置

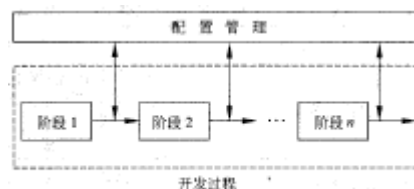


图 8.14 配置管理与开发过程

活 动	任 务	解 释
1. 过程实施	• 开发配置管理计划	• 计划描述：配置活动、这些活动的规程、进度、配置管理组织及与其他组织的关系 • 计划应形成文件
2. 配置标识	• 制定标识规则	• 借助配置标识控制软件项及其版本 • 标识内容包括基线文档、版本基准号等
3. 配置控制	• 标识并记录变更申请 • 分析与评价变更 • 批准(或不批准)申请 • 实现、验证和发行已变更的软件项 • 审核跟踪变更 • 控制并审核受控软件项	• 跟踪变更原因、变更授权 • 保证重要功能的安全或保密
4. 配置状态报告	• 编制管理记录和状态报告	• 表明受控项(包括基线)的状态和历史 • 状态报告应包括变更号、最新版本、发行标识、版本号及各版本比较
5. 配置评价	• 确定和保证软件项的功能完整性、物理完整性	
6. 发行管理和交付	• 有效控制软件产品和文档的发行和交付 • 在产品的生存期内保存代码、文档的主拷贝	包含重要的安全或保密功能的代码和文档应按组织的方针处理、储存、包装和交付

应该注意的是，软件配置管理直接控制的只是开发过程的产品，它只是间接地影响着开发活动。

8.3.1.3 软件配置管理的意义

1. 软件项目的特点

软件工程项目的对象是软件产品，它和传统的制造业产品有着很大的差别，这些差别决定了软件过程必须相应地采取特殊的管理措施，否则将无法达到软件工程项目的目标。

必须重视软件项目以下的特点是：

- 软件产品是逻辑实体，是不可见的、抽象的智力产品。
- 软件项目的规模日益庞大和复杂。
- 参与软件项目的人员数量增加，

相应地人员之间的沟通渠道数量按指数倍增。

若团组中人员数为 n ，则人员间存在着 $n*(n-1)/2$ 个需要互相沟通信息的渠道。

- 软件产品非常容易拷贝。
- 软件时时处在演化和变更状态，这包括技术在快速地发展，业务环境在不断地演变，不同用户各有不同的需求，软件需求往往在软件开发中变更，开发人员对阶段产品的变更相对比较灵活、容易等。
- 开发人员的离去对项目有较大的影响。

2. 忽视软件配置管理可能导致的混乱现象

以上列出的各项软件项目特点都要求采用软件配置管理，认真对待，以防出现各种混乱现象。其实在许多软件项目中一些混乱现象往往是屡见不鲜的，比如：

- 发给用户的软件产品事后发现，提供的用户手册是老的版本，或者是给错了版本。
- 某个软件已开发完成，完全应该正常运行，可是安装后不能工作。__、
- 这个软件在北京能正常工作，怎么拿到上海就出问题了？
- 上个月已经把这个缺陷解决了，现在怎么又出现了？
- 某个软件公司总是发生软件开发人员把开发的产品拿出去个人出售赢利。
- 哪个是最新修改了的源程序？我怎么也找不到，我还得给它打补丁呢。
- 编那个程序的人现在到哪去了？找不到他就找不到这个程序。
- 上个月用户还让我们作这个变更，可现在他为什么又不要了？

所有这些现象都将妨碍软件产品的正常工作，都属于产品的质量问題

3. 几类配置问题及其解决的对策

• **多重维护。**一个软件产品的几个拷贝在不同的地方在使用，或者若干个软件都含有一些共同的模块。在一个用户发现有软件缺陷后并没有通报给其他用户，便作了修正；或是几个用户分别发现了缺陷，在没有互相通报的情况下，各自作了不同的修正，这自然会出现歧异。

为解决这类问题，应该是发现缺陷后设法对所有的拷贝以相同的方式修正。

• **共享数据。**在一个程序中作了变更，但与另一程序的正常运行发生了冲突。例如子程序、数据库管理系统定义等都可能出现这种现象。

解决的办法是要控制变更，并且应保持互相沟通。

• **同时修改。**多个程序员对一个模块操作，就可能出现“怪异”现象：一个程序员对该模块所作的变更消失了。

防止发生这类现象就要更合理地划分模块，避免出现同时操作的情况。

• **丢失版本号或是不知版本号。**要明确规定保留哪个版本，销毁哪个版本；采用一种系统化的方法标识版本，并控制版本的变更；采用统一的备份规程。

原则上讲，认真实施软件配置管理就不会发生以上这些配置问题，或者说，以上的各种对策已经体现在软件配置管理的任务之中了。

8.3.2 软件配置管理计划

原则上，软件配置管理计划是软件开发计划的一个组成部分。一个软件工程项目启动以后，要认真分

析项目的要求和特点，精心地组织策划。在考虑制定进度安排计划、人员投入计划、质量保证计划、风险管理计划、文档编制计划等的同时，必须制定配置管理计划。

配置管理计划通常要涉及到该项目对软件配置管理的要求，实施软件配置管理的责任人、责任组织及其职责，开展的软件配置管理活动、方法和工具等。

这里以 IEEE 的标准为例，介绍配置管理计划应包括的内容。

配置管理计划标准 IEEE 828-1990 Standard for Software Configuration Management Plan

1. 引言

- 配置管理计划的目的、适用范围、使用要求
- 项目概述
- 项目中需特别关注的配置管理问题和风险
- 软件配置管理严格性要求的等级
- 限制和假设
- 术语
- 参考文件

2. 软件配置管理

- 配置管理的组织结构
- 职责和权限
- 指令和方针
- 参照的规程(组织的规程或客户的规程)
- 遵循的标准

3. 软件配置管理活动

- 配置标识
- 变更管理和配置控制
- 配置状态说明
- 配置审核
- 接口和子合同方控制

4. 软件配置管理进度安排__

- 软件配置管理重要事件的顺序
- 软件配置管理各项活动间的依赖关系
- 与其他重要项目里程碑的关系

5. 软件配置管理所需的资源

- 采用的工具
- 使用的设备
- 应用的技术
- 所需的培训
- 对其他人员的要求

6. 软件配置管理计划的维护

- 维护的责任
- 计划更新的条件和审批
- 计划变更的交流和通报

8.3.3 软件配置标识

软件配置标识是软件配置管理的基础性工作，是管理配置的前提。软件配置项没有进行标识便无法进行区分，也就容易出现混乱甚至丢失。

8.3.3.1 确定配置项

大中型软件项目在其开发过程中可能产生数十个、上百个，甚至上千个文档，其中许多是技术性的，也有不少是管理性的。技术性文档随着开发的进程，每个阶段都在演化，它们之间互相衔接，具有继承关系；同时对每一个文档来说，都会出现后期版本对前期的修正和扩展。而管理性文档如计划书、报告书、建议书、备忘录等等也有类似的变化和变更。确定配置项就是要决定究竟哪些需要保存下来，要被管理起来，或是说应该纳入配置管理之下，成为受控的。

Roger S. Pressman 认为至少以下所列配置项应该是受控的。

软件配置项

1. 系统规格说明

2. 软件项目计划一

3. 软件需求规格说明

- 图形分析模型

- 处理规格说明

- 原型

- 数学规格说明

4. 初步用户手册

5. 设计规格说明

- 数据设计描述

- 体系结构设计描述

- 模块设计描述

- 接口设计描述

- 对象描述(采用面向对象技术时)

6. 源代码清单

7. 测试规格说明

- 测试计划和步骤

- 测试用例和记录的结果

8. 操作和安装手册

9•可执行程序

- 模块可执行代码

- 链接的模块

10•数据库描述

- 模式和文件结构

- 初始内容

11. 联机用户手册

12•维护文档

- 软件问题报告

- 维护请求

- 工程变更指令

13. 软件工程标准和规程

8. 3. 3. 2 配置项命名及其相关信息

1.配置项命名

配置项命名是配置标识的重要工作。所谓标识，其实质就是区分，在众多的配置项中合理、科学地命名是最为有效的区分方法。为配置项命名时切忌任意，和随机。命名的基本要求如下。

- 惟一性：在一个项目内不能出现重名，以避免混淆。

- 可追溯性：也是系统性的要求，即名字应能体现相邻配置项之间的关系。

一个典型的实例是采用层次式命名规则来反映树状结构。我们知道树状结构上结点之间存在着层次的继承关系。例如在图 8. 15 中，CODE 是根结点为 PCL_OOLS 的树结构第 6 层结点，对其命名为 PCL_TOOLS/EDIT/FORMS/DISPLAY/AST_INTERFACE/CODE。

2.对象的标识形式

为做好软件配置项的控制和管理，作为配置对象的软件配置项(SCI)应具有自己的名字，并将其放入项目数据库中。一般而言，配置对象除有名字外，还有属性以及关系与其他对象相联接。在图 8. 16 中，配置对象“设计规格说明”“数据模型”、“模块 N”，“源代码”和“测试规格说明”均已给出定义。特别是配置对象间的联接关系也已用箭头表示清楚。例如，图中的两个单向箭头表示的是构成关系，即“数据模型”和“模块 N”两个对象都是对象“设计规格说明”的组成部分、因此可称“设计规格说明”为复合对象。另一方面，图中的双向箭头则指明了相互关系，如果“源代码”这一对象有了变更，就会沿着相互关系找到受影响的配置对象“测试规格说明”。

每个对象用一组特征信息(名字、描述、一组资源、实现)惟一地标识。

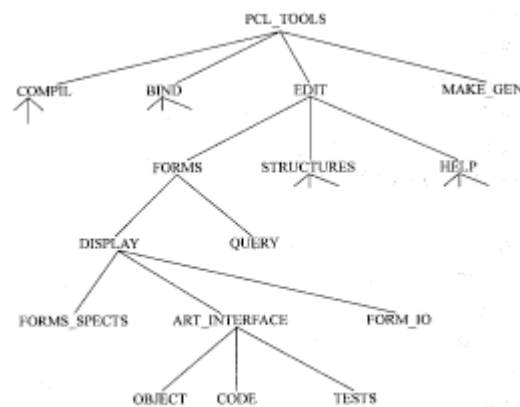


图 8.15 树状结构

- 名字：确切标识对象的字符串。
- 描述：数据项表，应包括对象表示的软件配置项类型(如文档、程序或数据)，项目标识符，变更和/或版本信息。
- 资源：该对象所提供的、处理的、引用的或另外所需要的实体。例如数据类型、特定函数，甚至是变量名。
- 实现：基本对象是指向“文本单元”的指针；复合对象则为“null”(空)。

标识配置对象还必须考虑所命名对象之间的联系。可以用 Gpart-ofd 来标识复合对象。它还能用来表达对象间的层次关系，例如可以写成：

```
E-R diagram 1. 4<part-of> data model;
data model<part-of> design specification
```

如果认为，对象层次结构中的对象间关系是沿着层次树的路径方向，那是不对的。事实上，在许多情况下，一些对象间的关系是跨越对象层次结构分支的。例如，数据模型关联到数据流程图(假定使用了结构化分析)，并且还关联到某个特定等价类的一组测试用例。这些交叉的结构联系可用以下方式表达：

```
data model<interrelated> data flow model;
data model<interrelated> test case class m;
```

前一句表明了构成对象间的关系，后一句表明了复合对象(data model)和基本对象(test case class m)间的关系。

配置对象间的关系可以用 MIL 语言(module interconnection language)表示。MIL 描述的是配置对象间的相互依赖关系，可自动构造系统的任何版本。

3.对象演变图

软件对象的标识还应对整个软件过程中对象的演变具有分辨能力。可以利用演变图来表示对象的演变，它给出了对象的演变历史，正如图 8.17 所示。图中对象以号码区分，其演变关系和顺序是很清楚的。这里对象的演变实际上就是版本的演变。

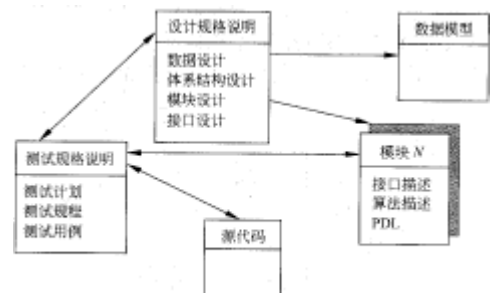


图 8.16 配置对象

8.3.4 变更管理

已经谈过变更是软件项目的一个突出的特点，也许它是软件项目最为普遍的一个特点。软件配置管理的一个重要任务便是对变更加以有效的控制和管理，其目的是对于复杂、无形的软件，防止在其多变的情况下失控，出现混乱，最终造成工作的损失，给用户带来伤害。

8.3.4.1 软件变更

1.软件变更的不可避免性

软件的变更来源有两个方面。一方面是用戶，他们是软件项目需求的提出者。一个十分常见的现象是用戶提出需求以后，在软件开发过程中用戶改变了需求，这只能迫使开发工作返工，丢弃一些无法修正的部分。无疑这会造成一定的损失，但又无法完全避免。要求软件用戶一次性地把需求讲清楚，并且不允许此后需求有任何变更，这是不现实的。我们只能尽力减少需求变更，降低它所造成的影响。开发人员如何解决好自己的工作产品与变更的用戶需求之间的一致性，正是 CMM2 级“需求管理”这个关键过程域的主要目标。

变更来源的另一方面是软件开发人員自身。他们在工作中可能发现前期工作中有些不妥当的地方，便要修改已经确定的设计方案或是设计的细节。也许是项目管理人员提出要修订已经确定的项目方案。由此所导致的返工甚至部分工作产品的报废也是在所难免的。

原则上说，随着工作的进展，无论是用戶还是开发人员都将掌握了更多的信息，对问题本身和设计方案有了更深入的认识，同时也会发现原来的设想有不充分、不完善甚至有不合理、不可行的成分。这时提出修正是完全合理的，是符合人們认识规律的。对于复杂而生疏的问题要求人們一次认识正确，其解决方案也要求一次设计完全无误都是不现实的。

软件变更出现的不可避免性决不意味着软件可以任意修改，也不能以此作为软件产品质量达不到要求的借口。毫无疑问，软件过程中变更管理的责任重大、能否解决好变更管理问题是成熟软件組織的一个明显的检验标志。

2.软件变更的复杂性

软件变更的复杂性不仅表现在由于项目规模大了，软件配置项的数量很大，也不只是因为版本多了，复杂性增加了，还必须看到软件变更的牵延性和项目組内部人員沟通协调的因素。

软件在一处出现了变更，可能要涉及到一些相关的部件和文档，为此要把这一变更通知到受影响的相

关人员。例如，测试引发了需求的修改，那么很可能要涉及到需求规格说明、概要设计、详细设计和代码等相关文档，甚至会使测试计划随之变更。因此，软件的某个部分改动了，就可能关系到参与该项目开发工作的许多人员。

如果是多个开发人员对软件的同一部件作修改，情况会更加复杂、例如，在软件测试时发现了两个故障。先指定甲去为第一个故障定位，并设法消除；同时指定乙去解决第二个故障。尽管最初以为两故障是无关的，但后来两人发现这两个故障引起的根本原因是一个，都是在某一个部件中某一变量初始化错误造成的。可是两人接受任务时还不了解这一情况。于是甲从库中取出该部件，作了修改，又送回库中；此后，乙从库中取出了原始版，做了他的修改，放入库中时代替了甲修改后的版本。显然，甲的工作白做了。在回归测试时，发现甲并没有做他的修正工作，要求他重新再做。他感到冤枉，只好再做，投入了不必要的人力和时间。

由此例可以看出配置管理的重要性，应该对存放在库内的配置项进行监督，实行“检查出库”和“检查入库”的措施。

3.变更管理的任务

变更管理简单地说就是控制修改，使之不出现改错、改乱的现象。变更管理的任务如下。

- **分析变更：**研究变更的必要性，经济可行性(成本一效益比是否合算)和技术可行性(能否实现)。
- **记录和追踪变更。**
- **采取措施保证变更在受控状态下进行。**

因此，IEEE 解释变更管理时说，它是软件配置管理的一个重要组成部分，涉及到在给配置项建立了正式的配置标识后变更的评价、协调、审批与实现诸方面的活动。

8.3.4. 2 配置库

配置库(configuration library)也称配置项库(configuration item repository)，是配置管理的有力工具。

1.配置库的作用

配置库的主要作用表现在：

(1)记录与配置相关的所有信息，其中存放受控的软件配置项是很重要的内容。

(2)利用库中的信息可评价变更的后果，这对变更控制有着重要的意义

(3)从库中可提取各种配置管理过程的管理信息，可利用库中的信息查询回答许多配置管理的问题，例如：

- 哪些客户已提取了某个特定的系统版本？
- 运行一个给定的系统版本需要什么硬件和系统软件？—
- 一个系统到目前已生成了多少个版本，何时生成的？——
- 如果某一特定的构件变更了，会影响到系统的哪些版本？
- 一个特定的版本曾提出过哪几个变更请求？
- 一个特定的版本有多少已报告的错误？

利用配置库实现配置管理是非常有效的。正如同一个大型工厂，生产出的许多零部件以及许多成品需要在仓库里加以集中存放和保管。要依靠仓库的管理机制保证存放在其中的零部件和成品的安全和有序，不致发生混乱(例如，把外形相似或完全一样的两种产品混淆)，也不致发生仓库存放的物品丢失现象。为此要强化仓库的管理，要采取一些有力和有效的措施，例如要严格坚持出入库的检查制度。

与此相似，采用配置库实现软件配置管理，就可以把软件开过程的各种工作产品，包括半成品或阶段产品和最终产品管理得井井有条，使其不致管乱、管混、管丢。上述甲乙二人修改程序时出现的问题，正是要靠对配置库的“入库检查”(check-in)和“出库检查”(check-out)加以解决，同时若配合有访问权限的措施就完全可以做到库内存放的产品什么人可以“看”，什么人可以“取”，什么人可以“改”，可以“存入”等等的控制。在这种控制之下的库中产品，如果甲正对其修改，乙就无法拿到，因为他取出时，这个产品被“锁住”了，所以不可能发生甲乙之间的问题。

2.三类库

配置库有三类。

• **开发库(development library)。**存放开发过程中需要保留的各种信息，供开发人员个人专用。库中的信息可能有较为频繁的修改，只要开发库的使用者认为有必要，无需对其做任何限制。因为这通常不会影响到项目的其他部分。

• **受控库(controlled library)。**在软件开发的某个阶段工作结束时，将工作产品存入或将有关的信息

存入。存入的信息包括计算机可读的以及人工可读的文档资料。应该对库内信息的读写和修改加以控制。

• **产品库(product library)**。在开发的软件产品完成系统测试之后，作为最终产品存入库内，等待交付用户或现场安装。库内的信息也应加以控制。

作为配置管理的重要手段，上述受控库和产品库的规范化运行能够实现对软件配置项的管理。

8.3.4.3 配置基线

1.基线

基线(baseline)是软件生存期各开发阶段末尾的特定点，也称为里程碑(milestone)，在这些特定点上，阶段工作已结束，并且已经形成了正式的阶段产品。

建立基线的概念是为了把各开发阶段的工作划分得更加明确，使得本来连续开展的开发工作在这些点上被分割开，从而更加有利于检验和肯定阶段工作的成果。同时有利于进行变更控制，有了基线的规定就可以禁止跨越里程碑去修改另一开发阶段的工作成果，并且认为建立了里程碑，此时的阶段成果已被“冻结”。

图 8.18 给出了软件配置基线的示意图。图中在每个开发阶段的末尾都标出了该阶段的基线，图的上部则给出了各开发阶段的工作成果。事实上，现在人们已经把这些工作成果称为基线了。例如，设计基线指的就是设计规格说明。

作为阶段工作的正式产品，基线应该是稳定的，如作为设计基线的设计规格说明应该是通过评审的。如果还只是设计草稿，就不能作为基线，不能被“冻结”。

2.基线的种类

如果把软件看作是系统的一个组成部分，以下 3 种基线是最受人们关注的。

• **功能基线**。功能基线是指在系统分析和软件定义阶段结束时，经过正式评审批准的系统设计规格说明中对被开发软件系统的规格说明；或是指经过项目委托单位和项目承办单位双方签字同意的协议书或合同中所规定的对被开发软件系统的规格说明，或是指由下级申请及上级同意或直接由上级下达的项目任务中所规定的对待开发软件系统的规格说明。

• **分配基线**。分配基线是指在软件需求分析阶段结束时，经正式评审和批准的软件需求规格说明。

• **产品基线**。产品基线是指在软件组装与系统测试阶段结束时，经正式评审和批准的有关所开发的软件产品的全部配置项的规格说明。

这 3 种基线如图 8.19 所示。

3.基线与配置项

提出基线的概念本来是为了更好地实现变更控制，但如果把每个基线都当成一个整体来看待会造成麻烦。因为一个变更很可能只涉及到基线的很小部分。例如，假定某个大型软件中的一个模块修改了，如果将这一变更当作整个软件产品基线的变更，就很不方便。

事实上，基线可由多个软件配置项组成，一个软件配置项可以是一个文档，或者是一个可直接放在配置控制之下的工作产品，能够作为一个独立的基本部件加以修改。文档通常已被认为是独立可修改的部件了，但如有必要还可将其再加细分，把文档中的章、节甚至段当作软件配置项来看待。

以产品基线为例，它往往含有多个代码级的配置项。而代码的变更会是频繁的，因为几乎所有的变更最后都要导致某些代码的变更。特别是在多个程序人员参与工作的情况下、每个人负责自己分工的那个模块，责任是清楚的，这就十分有利于变更的控制和追踪。

在定义软件配置项时有两种作法：一个作法是，把每个单独可编译的模块当作一个软件配置项，模块的名字就是软件配置项的名字；另一作法是把每个文件(由若干模块或若干定义构成)当作一个软件配置项，文件名当作配置项名。

很明显，软件配置管理所管的配置项并不都是互相独立的，它们之间可能存在着某种相互依赖关系。如果说，配置项 X 对配置项 Y 依赖，是指假如 Y 作了变更，要求 X 也作变更，使 X 保持正确或者说使两个基线是一致的。不过除非是从软件配置项的性质导出的情况，这种依赖关系很难清晰地文档中表达。例如，体现设计文档的配置项往往依赖于代表需求文档的配置项，那就要在设计文档中说明，每项设计对应了哪些需求的实现。如果一个设计基线是由许多配置项组成，我们可以据此理解设计的某些项和需求的某些项之间有着依赖关系。在代码中的情况也是这样，代表一模块的配置项依赖于另一模块的配置项，这种

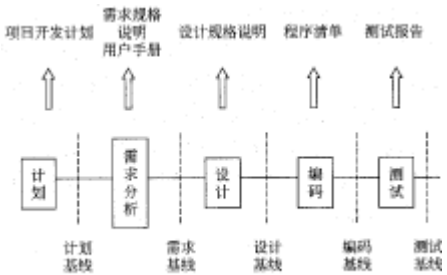


图 8.18 软件配置基线

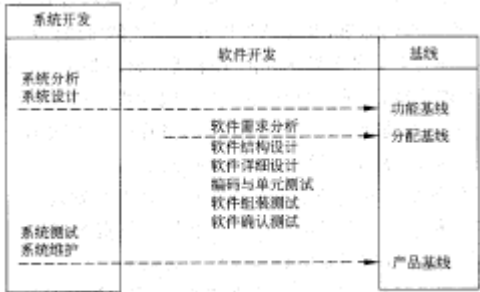


图 8.19 功能、分配和产品基线

依赖关系往往可从设计规格说明中得到，在实施变更控制时，依赖关系应在变更请求中反映出来。这一点在后面讨论变更请求中还会涉及到，就是要明确变更的影响范围。

8.3.4.4 变更控制

1. 变更控制组

变更控制组(change control Board-CCB)也称为配制控制组(Configuration Control Board)，是配制项变更的监管组织。其任务是对建议的配制项变更作出评价，审批以及监督已批准变更的实施。

变更控制组的成员可以包括项目经理、用户代表、软件质量控制人员、配置控制人员。这个组织不必是常设机构，完全可以根据工作的需要组成，例如按变更内容和变更请求的不同，组成不同的 CM 小的软件项目 CCB 可以只有 1 人，甚至只是兼职人员。

如果 CCB 不只是控制变更，而是负有更多的配置管理任务，那就应该包括基线的审定、标识的审定以及产品的审定。并且可能根据工作的实际需要分为项目层、系统层和组织层来组建，使其完成不同层面的配置管理任务。

2.变更请求与变更控制

(1)利用配置库实现变更控制。

一般情况下，开发中的软件配置项尚未稳定下来，对于其他配置项来说是不可见的，是处理工作状态下，或称自由状态下，此时它并未受到配置管理的控制，开发人员的变更并未受到限制。但当开发人员认为工作已告完成，可供其他配置项使用时，它就开始趋于稳定。把它交出评审，就开始进入评审状态；若通过评审可作为基线进入配置库(实施 check-in)，开始“冻结”，此时开发人员不允许对其任意修改，因为它已处于受控状态。通过评审表明它确已达到质量要求；但若未能通过评审，则将其回归到工作状态，重新进行调整。可以通过图 8. 20 看到上述配置项状态变化的过程。

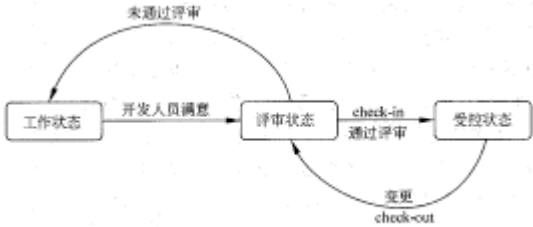


图 8.20 配置项的状态变化

处于受控状态下的配置项原则上不允许修改，但这不是绝对的，如果由于多种原因需要变更，就需要提出“变更请求”(check request). 在变更请求得到批准的情况下，允许配置项从库中检出(实施 check-out)，待变更完成，并经评审后，确认变更无误方可重新入库，使其恢复到受控状态。我们称此过程为库管理，可以借助于工具实现库管理。

(2)变更请求。

变更请求是实施变更控制的起始一步，也是必不可少的一步。最为常见的变更理由可能是消除缺陷、适应运行平台的变更，或是软件扩展提出的要求，例如增加功能、提高性能等。

变更请求的主要内容有 3 个方面：

- **变更描述。**包括变更理由、变更的影响、变更的优先性排序等，就是要申述要做什么变更，为什么要做，以及打算怎么做变更。
- **对变更的审批。**对变更必要性、可行性的审批意见，主要是由配置管理的负责人和 CCB 对此项变更把关。
- **有关变更实施的一些信息。**表 8. 11 提供了变更请求表的实例。

(3)变更控制过程。从上述变更请求表中已能看出变更控制的大致过程，下面以变更请求表 CRF 为基础进一步给出其控制过程，见表 8. 12。

在分析和评估变更请求中主要考虑的是变更对成本、进度和质量等方面的影响。必要时配置管理人员、变更分析人员可能要和变更请求人交谈和商讨。在 CCB 批准后送交变更实施者，应该要求记录变更的情况。实际上变更请求表上不仅记载了变更请求和变更审批的信息，而且还包含有关变更实施的信息，因此，可通过变更请求表了解到变更的实施状态。

关于变更实施情况还可通过状态说明了解，参看 8.3. 7 节。

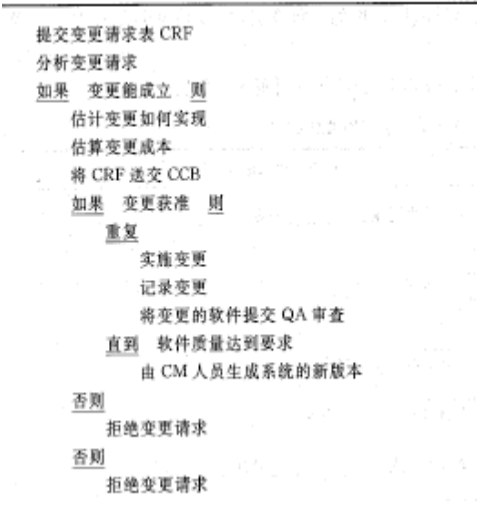
(4)故障报告。

提出变更请求最为常见的情况是已经入库的基线发现了新的

表 8.11 变更请求表 CRF

项目名	变更请求标识
变更请求：	变更请求人 _____ 日期：_____
变更理由 _____	
变更描述 _____	
影响范围 _____	
变更优先性考虑 _____	
估计变更工作量 _____	
分析与评估：	分析者 _____ 日期 _____
分析与评估意见 _____	
审批：	CCB 负责人 _____ 日期 _____
CCB 审查意见 _____	
变更实施：	实施负责人 _____ 日期 _____
变更实施情况 _____	
质量保证审查：	QA 负责人 _____ 日期 _____
审查意见 _____	
配置管理审查：	CM 负责人 _____ 日期 _____
审查意见 _____	

表 8.12 变更控制过程



缺陷，表现为故障，为了更好地实施变更和变更管理，有的软件组织要求在提出变更请求前先提出故障报告(fault report, FR)。

附有故障报告的变更请求，特别在故障较为严重时，常常被当作高优先度的变更请求处理。事实上，故障报告还可用于追踪软件中缺陷清除的状态。

故障报告包含的内容有：

- **FR ID(故障报告标识)**。故障信息。包括故障描述，故障严重程度，怀疑有问题的部位，故障的影响，故障现象和环境信息，估计的故障原因，故障信息提供者等信息。
- **CCB 评估意见**。是否批准，优先性如何，相关说明等。
- **故障修复信息**。要变更的部分和相关说明。

3.变更记录

按上述要求尽管变更已被置于控制之下，但为长期保留变更的相关信息以备后用，需要把这些信息保存起来。

首先应将变更请求表(CRF)作为配置项在配置库中登录。其次，在变更的代码模块或文档内应记录有关变更的信息。以代码变更为例，表 8.13 是变更记录置于模块首部的实例。

8.3.5 版本管理

8.3.5.1 软件版本

所谓软件版本，包括两种不同的含意。一种是为满足不同用户的不同使用要求，如适用于不同运行环境或不同平台的系列产品。如有的需适合于 Unix 用户，有的适用于 Windows 用户，称为 Unix 版和 Windows 版软件产品。它们之间在功能和性能上是相当的，原则上没有差别，或者说，这些是并列的系列产品。对于这类差别很小的不同版本，互相也称为变体(variant)。

另一种软件版本的含意是在软件产品投入使用以后，经过一段运行提出了变更的要求。例如，需要作较大的修正或纠错，需要进一步增加功能和提高性能。这种修正不止进行一次，于是得到的也是系列产品，只不过是顺序演化的系列产品，每次演化出的产品称为一个版本，每个版本都可以说出它是从哪个版本导出的演化过程。

必须注意到，修正后的新版本往往不能完全代替老版本，尽管新版本有某些优越的特性。因为一些用户仍然使用着老版本，并且不容易立刻做到以旧换新，否则可能会打扰老版本原有的工作环境。显然，多个版本被多个用户同时在使用的情况是不可避免的现实。这就要求多个版本在库内共存，于是版本管理成为重要的课题，否则便会出现混乱。

8.3.5.2 版本管理(version management)也称版本控制(version control)

它把用于管理软件过程中生成的各种不同的配置对象的规程和相关管理工具结合起来。按 G. M. Clemm 对版本管理的解释：配置管理使用户借助选择适用的版本来选定软件系统的配置，为此需确定每个软件版本的属性，同时还应考虑到由描述一些预期属性所确定(或所构成)的配置。

版本管理要解决的第一个问题便是版本标识，也就是为区分不同的版本，要给它们以科学的命名。通常有以下几种版本命名的方法。

1. 号码版本标识

以数字表示，如第 1 版，第 2 版等。再进一步加以区分可以给出 1.0, 1.1, 1.2, 2.0 等版号。一般认为 1.0, 2.0 等是基础版本号，1.1 和 1.2 等是对基础版 1.0 的第 1 次修订和第 2 次修订。显然这些修订是对前一版少量的或次要的更动。若有重大更动或因多次修订导致的全局性重要更动，则应提高基础版本号，例如上升到 2.0。

这种顺序号码的命名法被广泛地采用，它的突出优点就是简单直观。但如果版本多了，并且出现了非简单顺序的线型号码，就很难从号码上区分其前后的继承关系，无法体现命名的可追溯性原则。如图 8.21 所示。另外，根据号码也不能看出更多的属性信息。为解决这一问题又有其他的版本命名方法。

2. 符号版本标识

这种标识版本方法是把重要的版本属性有选择地给出。如 V1/VMS/DB Server, 表示一个在 VMS 操作系统上运行的数据库服务器版本。为了从版本标识上看到更多的信息，可能给出更多的属性，如面向的客户群、开发语言、硬件平台、生成日期等。

版本管理工具近年出现了一些实现版本控制的自动工具。例如 RCS, SCCS, PVC5Version Manager 等。它们之间的差别在于用来构造系统特定版本的属性不同，或是构造的机制有所不同。

表 8.13 代码变更记录实例

//PROTEUS Project (ESPRIT 6087)				
//PCL-TOOLS/EDIT/FORMS/DISPLAY/INTERFACE				
//Object: PCL-TOOL-DESC				
//作者:陈**				
//开发日期:2001.12.8				
//版权归属:ASDC				
//变更记录				
//版号	变更负责人	日期	变更概要	变更理由
//1.0	王**	2002.4	*****	*****
//1.1	李**	2002.9	*****	*****
...				
...				

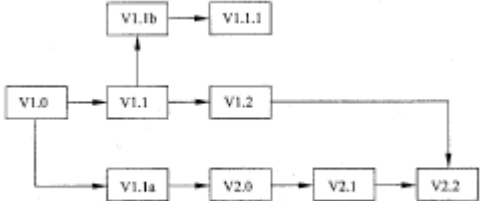


图 8.21 号码版本标识

8.3.6 配置审核

8.3.6.1 什么是配置审核

关于配置标识、配置项的变更控制等方面应该如何按规定实施，前面已经给出了说明，但在具体的项目开发中是否得到了遵循，需要进行检查。配置审核的任务便是验证配置项对配置标识的一致性。软件开发的实践表明，尽管对配置项做了标识，实践了变更控制和版本控制，但如果不做检查或验证仍然会出现混乱。

这种验证包括：

- 对配置项的处理是否有背离初始的规格说明或已批准的变更请求的现象。
- 配置标识的准则是否得到了遵循。
- 变更控制规程是否已遵循，变更记录是否可供使用。
- 在规格说明、软件产品和变更请求之间是否保持了可追溯性。

配置审核工作主要集中在两个方面，一是功能配置审核，即验证配置项的实际功效是与其软件需求一致的；二是物理配置审核，即确定配置项符合预期的物理特性。这里所说的物理特性是指特定的媒体形式。

8.3.6.2 为什么要实施配置审核

配置审核的实施是为了确保软件配置管理的有效性，体现配置管理的最根本要求，不允许出现任何混乱现象，例如：

- 防止出现向用户提交不适合的产品，如交付了用户手册的不正确版本。
- 发现不完善的实现，如开发出不符合初始规格说明或未按变更请求实施变更。
- 找出各配置项间不匹配或不相容的现象。
- 确认配置项已在所要求的质量控制审查之后作为基线入库保存。
- 确认记录和文档保持着可追溯性。

8.3.6.3 如何实施配置审核

1. 实施配置审核的时机

通常选择以下几种情况实施配置审核：

- 软件产品交付或是软件产品正式发行前。
- 软件开发的阶段工作结束之后。
- 在维护工作中，定期地进行。

2. 实施配置审核的责任人

参与实施配置审核的审核人员可以包括项目组人员及非项目组人员，例如其他项目的配置管理人员、软件组织的内部审核员以及软件组织的软件配置管理人员。

3. 配置审核工作的开展

工作步骤如下：

(1) 由项目经理决定何时进行配置审核工作。

(2) 质量保证组或软件组的配置管理组指定该项目的配置审核人员。

(3) 项目经理和配置审核员决定审核范围。

(4) 配置审核员准备配置审核检查单。

(5) 配置审核员安排时间审核文档和记录，审核活动可能涉及到：项目范围，配置项的入库(check-in)及出库(check-out)，评审记录，配置项的变更历史，测试记录，文件的命名，变更请求，版本的编号。

(6) 配置审核员在审核中发现不符合现象，并作记录。

(7) 由项目经理负责消除不符合现象。

(8) 配置审核员验证所有发现的不符合现象确已得到解决。

8.3.7 配置状态报告

8.3.7.1 什么是配置状态报告

配置状态报告(configuration status reporting)也称配置状态说明与报告(configuration status accounting&reporting)，它是配置管理的一个组成部分，其任务是有效地记录和报告管理配置所需要的信息，目的是及时、准确地给出软件配置项的当前状况，供相关人员了解，以加强配置管理工作。

在软件工程过程中，必须注意到它的动态特性。事实上，在软件工程过程中，软件配置项都在不停地演化着。例如，随着开发的工作进展，工作产品不断地扩展，形式也在变化着，从需求规格说明到设计说明到源程序等等。另一方面，由于各种原因(纠错只是其中的一个原因)，设计说明本身也在演变着，版本在更新着。对于这种动态特性如果没有控制手段，其后果是不可想象的。

配置状态报告就是要在某个特定的时刻观察当时的配置状态，也就是要对动态演化着的配置项取个瞬

时的“照片”，以利于在状态报告信息分析的基础上，更好地进行控制。

需要跟踪捕捉的状态报告信息可以是配置项的当前标识，已交付软件的配置，变更请求或问题报告的状态和已获准变更的状态。

8.3.7.2 配置状态报告信息

1. 状态说明的实体关系

可用图来说明配置状态所涉及到的实体之间具有的特定关系如图 8.22 所示。

图中的框表示状态实体，连接各框之间的连线一端有一个头，另一端有 3 个头，表示 1 对多的关系。

2. 状态说明数据词典

状态说明中涉及的信息如下所示。

一、配置项库(repository)

库名
库标识
所有者
范围/描述

二、配置项(configuration item)

库标识
项标识
项名
描述
项类型(源代码、测试计划等)

三、配置项版本(configuration item version)

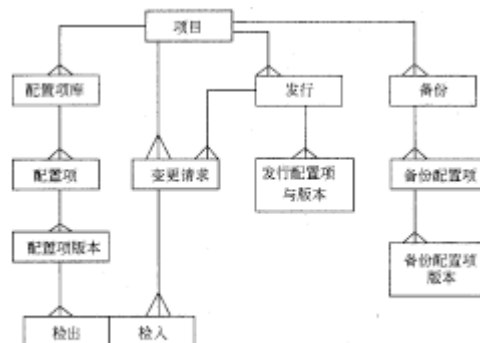
库标识
项标识
版本号
入库日期、时间
与前版差异描述
锁定状态

四、出库与入库(check-out & check-in)

库标识
项标识
出库版本号
出库责任人
出库日期及时间
实施的变更请求号
变更描述
入库版本号
入库责任人
入库日期及时间

五、变更请求(change request)

变更请求号
软件版本号
申请
一 申请人
一 申请日期
一 变更部件
一 变更优先性
一 变更概述
一 变更预期效果
一 附件
分析与审核



- 一受影响工作项
 - 一估计工作量投入
 - 一成本
 - 一其他影响
 - 一假设
 - 一效果
 - 一分析日期
 - 一分析人
 - 一是否批准
 - 一理由
 - 一审批日期
 - 一批准人
 - 一发行版本
 - 实施状态
 - 一受影响的每个工作项
 - ★库标识
 - ★项标识
 - ★变更描述
 - ★出库版本
 - ★出库日期及时间
 - ★变更工作量
 - ★验证工作量
 - ★入库版本号
 - ★入库日期及时间
 - 一说明
 - 一变更结束日期及时间
 - 一变更结束责任人
- 六、发行(release)
 - 发行版本
 - 发行日期
 - 目的
 - 创建说明
- 七、发行配置项及版本号
 - 库标识
 - 项标识
 - 概述
 - 项类型(源代码、测试计划等)
 - 版本号
- 八、备份
 - 备份号
 - 备份日期
 - 备份人
 - 目的
 - 介质
- 九、备份配置项
 - 库标识
 - 项标识
- 十、备份配置项版本
 - 库标识
 - 项标识

版本号

3.定期提交的配置状态报告的内容示例

内容包括以下各项。

• 各份变更请示概要：变更请求号、日期、申请人、状态、估计工作量、实际工作量、发行版本、变更结束日期。

- 基线库状态：库标识、至某日预计库内配置项数、实际配置项数。
- 发行信息：发行版本、计划发行时间、实际发行日期、说明。
- 备份信息：备份日期、介质、备份存放位置。
- 配置管理工具状态。
- 配置管理培训状态。

4.配置状态报告提供信息的利用示例

在配置状态报告中提供了许多有关软件配置的信息，应该充分利用这些信息实现配置的控制。以下给出利用这些信息可以解决一些需要澄清的问题。例如：

- 程序 P13 的 1. 6 版在哪个备份中可以使用？
- 在发行 5.1 和发行 5. 2 之间实现了哪些变更请求？
- 在发行 5.2 中哪些程序更改过了？
- 在变更请求 671 中要对哪些配置项进行更改？在变更前和变更后，这些程序单元的版本是什么？是否所有的变更都完成并入库了？

8.3.7.3 状态说明

在变更请求批准后，实施变更需要一段时间，要设置一种管理手段来反映变更所处的状态，这就是变更状态说明(status accounting). 它可供项目经理和 CCB 追踪变更的情况。

要求状态说明回答的问题可以是：

- 某个变更请求是否已被批准？
- 已批准的变更请求目前处于什么状态？
- 已完成的变更投入了多少时间和工作量？
- 某个软件配置项与哪几个变更请求有关？

状态说明的信息可通过变更请求(CR)和故障报告(FR)得到。

变更状态可分为活动态(正在实施变更)，完成态(已完成变更)和未列入变更 3 种。

8.4 面向对象的开发方法

面向对象方法起源于 20 世纪 60 年代由挪威计算中心开发的 Simula67 语言，它首先引入了类的概念和继承机制, 80 年代美国加州的 Xerox 研究中心推出的 Smalltalk 语言和环境，使面向对象方法得到比较完善的实现，掀起了面向对象方法研究的高潮。80 年代至 90 年代，涌现出大批实用的面向对象语言，如 C++, Object Pascal, Eiffel 等，大大地提高了软件开发的效率以及软件的可复用性和可维护性。90 年代面向对象方法不再局限于编程阶段，向着软件生命期的前期阶段发展，形成了面向对象的分析和设计，发展成一整套软件方法学，成为计算机领域的主流技术之一。

面向对象方法的基本思想是从现实世界中客观存在的事物出发来构造软件系统。软件系统适用的业务范围称作软件的问题领域，把问题领域中事物的特征抽象地描述成类，由类建立的对象作为系统的基本构成单位，它们的内部属性与服务描述了客观存在的事物的静态特征和动态特征。对象类之间的继承关系、聚合关系、消息和关联反映了问题领域中事物之间实际存在的各种关系。

80 年代以来相继出现了多种面向对象分析和设计的方法，较为流行的有 Booch 方法, Coad 和 Yourdon 方法, Jacobson 方法, Rumbaugh 方法, Wirfs-Brock 方法等。各种方法分别提出了一套较为完整的系统模型、表示方法和实施策略、其中 Coad 和 Yourdon 方法建模符号简单，开发的模型直接明了，容易掌握，在软件开发的实践中应用广泛，因此本章选用 Coad 和 Yourdon 方法，讨论面向对象的分析，并且简单地介绍面向对象的设计以及文档的编写。

8.4.1 面向对象分析

面向对象分析(object-oriented analysis, OOA)的目标是建立待开发软件系统的模型 OOA 模型描述了表示某个特定应用领域中的对象、对象间的结构关系和通信关系，反映了现实世界强加给软件系统的各种规则和约束条件。OOA 模型还规定了对象如何协同工作和完成系统的职责。

在 Coad 和 Yourdon 的方法中，OOA 模型是一个类图，由 5 个层次构成，分别为对象-类层、属性层、服务层、结构层和主题层。事件响应对象交互(Event-Response Object-Interaction; EROD 图是 OOA 的辅

助模型，主要用于检查模型描述的系统是否提供了满足需求的对象及服务。下面以电梯控制系统(简称 ECS)为例说明如何建立 OOA 模型。

8.4.1.1 电梯控制系统需求说明

要求设计和实现一个程序，用以调度和控制一幢 40 层高的建筑物一中的 4 部电梯。程序必须高效而合理地调度电梯。例如，如果有人在第 4 层楼按下下行电钮想要召唤电梯下楼，那么下一个正在往下走的电梯到了第 4 层后就该停下接收该乘客。另一方面，如果电梯没有乘客，它应该停在它最后所到达的楼层，直到它再次投入使用为止。另外，在将乘客运送到目的地以前，电梯不能向着相反的方向运行。而且，一个已经满载的电梯不再受理新的召唤请求。

每个电梯中都配备有一个目的地按钮面板，其上的每个按钮对应着一个楼层。这些目的地按钮可以被从计算机传到按钮面板的信号控制而发光。当乘客按下一个未发光的目的地按钮时，按钮面板后的电路将给计算机发送一个中断信号，并且将引起中断的目的地按钮所对应的楼层号存放在输入寄存器中。

当接收到目的地按钮所发来的中断时，系统将发送信号到按钮面板，从而使相应的目的地按钮发光。另外，当控制器命令电梯停在某楼层时，它同时应该发送一个信号给按钮面板，使相应的目的地按钮不再发光。

电梯中对应的每个楼层都配有一个楼层传感器开关。当电梯抵达某个楼层的 8 英寸范围内时，电梯上的一个轮子便关闭该楼层的传感器开关，并向计算机发送一个中断信号。并且将引起中断的楼层传感器开关所对应的楼层号存放在输入寄存器中。

每个电梯内部都配备有一个到达指示灯面板，每个楼层对应着一个指示灯。当电梯到达某个楼层时，其相应的到达指示灯发光；而当电梯离开某个楼层，其相应的到达指示灯熄灭。

建筑物的每个楼层都配备有一个召唤按钮面板，上面装有两个按钮。其中一个按钮标有向上的标记，而另一个则标有向下的标记。底层的按钮面板上只有一个按钮，上面标有向上的标记；顶层的按钮面板上也只有一个按钮，上面标有向下的标记。当乘客按下某个未发光的按钮时，按钮面板后的电路将向计算机发送一个中断信号。并且将引起中断的召唤请求按钮所在的楼层号存放在输入寄存器中。

当计算机接收到召唤按钮所发来的中断信号时，它将发送信号到召唤按钮面板上，从而使相应的请求按钮发光。当控制器命令电梯停在某楼层时，它同时应该发送一个信号给召唤按钮面板，以关闭相应的请求按钮指示灯。

电梯的制造商们使用符合行业规定的开关、中继器和电路可以保证电梯的安全，而无需计算机控制器来担心这个问题。例如，当电梯抵达某个楼层的 8 英寸范围内计算机给电梯下达了停止命令时，电梯则停在与该楼层平齐的位置上，打开电梯门，持续打开一段时间再关上电梯门。如果在此期间计算机向电梯发出向上或向下运行(例如当电梯门处于打开状态时)的命令，制造商们的策略则是忽略该命令，直到电梯运行的条件被满足为止。

8.4.1.2 标识对象和类

对象是系统中用来描述客观事物的一个实体，是构成系统的一个基本单位，由一组属性和一组对属性进行操作的服务组成。

类是某些具有相同属性和服务的对象的模板，抽象地描述了属于该类的全部对象的内部结构，主要包括对象的属性和操作。

对象-类层表示待开发系统的基本构造块，对象-类是现实世界中应用领域概念的抽象，这一层是整个 OOA 模型的基础。

建立对象-类层，首先分析应用领域中的概念以及用户需求描述的系统功能，从中选出候选对象的集合。然后对候选对象逐个进行审查，丢弃无用的对象。通常如果正确地标识了对象，就可以为每一种对象定义一个类。但是从标识对象到定义类是从单个对象到一般概念的抽象过程，需要对特殊情况进行检查，并作必要的修改和调整。最后画出 OOA 模型的对象-类层，然后对每个对象-类给出文字描述。

对象-类的图形符号如图 8.23 所示。

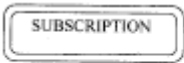


图 8.23 对象-类的图形符号

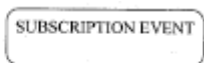


图 8.24 抽象类的图形符号

对象-类用两个矩形框表示，外层矩形框表示实例的边界，内层表示类的边界。对象-类的名字写在矩形框内的上方，如图 8.23 所示。只有一个边框表示没有实例的抽象类，见图 8.24。

1. 如何选择候选对象

对用户需求及相关的资料进行短语频率分析，能够得到大量的现实世界中的概念，可以在这些概念中选择候选对象。也可以根据用户需求从以下几方面考虑建立对象：人员、组织、设备、物品、业务、事件、表格等。

对象的选取与应用领域紧密相关，如在医学研究系统中，可以将女孩阿诗玛抽象成人这个类，由头、躯干、四肢、臂、腿等对象-类组成，而在交通管理系统中女孩阿诗玛很可能属于驾驶员类，由事故历史、违章记录、保险金额等相关的类组成。

选择对象时要考虑系统的边界，确定哪些是待开发系统要建立的对象，而哪些是系统职责之外的实体。商店管理系统中，如果不考虑给顾客发优惠卡，商店不需要记录任何有关顾客的信息，顾客则不应该作为候选对象。

在建立 OOA 模型时不考虑与实现技术相关的对象，只考虑与应用领域相关的对象类，软件开发环境、用户界面等都属于实现相关的技术，相应的对象-类应推迟到建立面向对象设计模型时考虑。

应该严格地审查每个候选对象。对每个候选对象，可以审查以下这些问题：

- 每个对象都具有一定的功能吗？
- 其他对象是否要求这个对象执行某些功能？
- 其他对象是否要求这个对象报告它所存储的信息？
- 是否每个对象都封装了一些对外隐藏的信息？
- 是否每个对象都有一个生存期，是否描述了其产生、生命期中的各种状态及其最后的消亡？
- 假如向应用领域的专家描述这个候选对象，他/她能不能马上明白这个对象的重要性？

对于每一个具体的候选对象，并不要求所有这些问题的回答都是肯定的，但这些答案不能全是否定的。

还需要写下在系统范畴之外所有会发生的事件，再考虑以下问题：

- 哪些你的系统能识别？
- 哪些你的系统必须作出响应。
- 对于所列出的每一个会发生的事件，哪些候选对象能识别这个事件？哪些候选对象产生系统响应？
- 是否所有的候选对象都涉及到了？
- 还需要建立其他的对象吗？
- 是否存在既不能识别所发生的事件又不能产生系统响应的对象？如果存在的话，那么这些对象的功能又是什么呢？

现在我们仍将注意力放在系统上而暂且不去考虑对象，那么

- 你的系统是否需要与其他人或系统相连接？
- 你是否标识了介于接口之间的对象？注意不要将这些对象与实现接口的对象(如网络对象、GUI 图形用户界面对象等)混为一谈。’

去掉这个候选对象可提出问题：“系统会怎么样呢？无法工作成为废物，还是产生其他什么问题？”最后再问：“如果我要复用包括这个对象在内的一大块东西，这个对象是否真正会被复用到呢？它在目前和将来能提供足够的通用功能，因而使它成为目前正在构造的系统以及将要构造的系统的一个组成部分吗？”

请注意，由于这是一个重复进行的过程，可能会在所列出的对象表中加入一些新出现的对象或删除一些已存在的但没有用的对象。这些在详细地标识每个对象的属性、服务及相关消息的过程中肯定会发生。

2. 电梯控制系统的对象类层

根据电梯控制系统的需求说明可以列出一张很长的应用领域概念清单，如电梯、电梯线路，电梯中断、电梯控制系统、电梯门、电梯号、电梯位置、电梯到达、电梯停止运行、乘客、电梯超载、电梯马达、目的地按钮、目的地面板、召唤按钮、召唤指示灯、大楼……

其中一些概念是与实现相关的，与电梯控制系统的功能无关，如电梯线路、电梯中断等。有些概念虽然很重要，但不能成为对象，它们可能是属性或服务，如电梯号、电梯位置。某些是与实现技术相关，如按钮。还应该考虑有些概念在字面上相同，是否真的都有不同的意义。冗余的、矛盾的和模棱两可的概念不应该成为候选对象。最后初步选定的电梯控制系统对象的集合如下。

- 到达事件 (ARRIVAL EVENT)。这个事件对象封装了电梯到达某一楼层时必须执行的所有各种服务。
- 到达按钮面板 (ARRIVAL PANEL)。这个对象是我们原来的到达指示灯 (ARRIVAL LIGHT) 的另一种说法。
- 目的地事件 (DESTINATION EVENT)。这个对象封装了如何得知目的地请求的秘密。
- 目的地面板 (DESTINATION PANEL)。目的地面板对象告诉我们应该将哪部电梯开到目的地。
- 电梯 (ELEVATOR)。假设任何类型的控制对象(如 ELEVATOR CONTROLLER 或 SCHEDULER)都是与实现有关的，将在 OOD 模型中加入这样的对象。

在建立 OOA 模型时假设：电梯控制系统可以没有集中的控制器或调度。因此集中的控制器或调度就不是基本的需求，而是一个实现问题。

请读者注意我们并没有说电梯控制系统可以不要控制或调度，正相反，对电梯进行控制和调度是必须的，只是说集中的控制或集中的调度不是非有不可的。

在分析阶段，电梯这个对象封装了电梯管理和控制所需要的数据和各种用于报告电梯当前状态的服务。

- 电梯马达 (ELEVATOR MOTOR)。电梯马达这个对象包含了各种控制服务。标识出电梯马达这个对象能使人们方便地在其中隐掉马达的技术细节，从而尽可能提高控制系统的可扩充性。

- 楼层 (FLOOR)。决定利用楼层这一对象实现如何派送电梯的功能。在楼层派送电梯方法中，一个重要的概念就是哪一个楼层“占有”电梯，即哪个楼层能控制电梯。如果电梯处在或将要到达某一楼层，就说这个楼层“占有”这部电梯。只有当前占有电梯的楼层才能改变这部电梯的升降状态。当电梯在上行或下行时，其控制权就以接替的方式从一个楼层传到另一个楼层。从面向对象的角度来说，就意味着一部电梯只能由一个楼层的实例进行控制。如一部电梯从第 40 层楼向下移动，它就由 39 层, 38 层……相继控制，直到最后到达目的地楼层。

- 超重传感器 (OVERWEIGHT SENSOR)。不仅仅在电梯对象中封装传感器技术的秘密，而且决定建立一个单独的超重传感器对象。

- 召唤事件 (SUMMONS EVENT)。见前面所讨论的目的地事件。

- 召唤控制面板 (SUMMONS PANEL)。见前面所讨论的目的地按钮面板。

现在讨论以下没有成为模型中对象的条目，为什么它们不符合选择对象的准则。

- 大楼 (Building)。用户认为它完全不属于应用领域范畴。如果需要将 ECS 安装在具有不同楼层、不同数量的电梯的多个建筑物中，那么不同的楼层、不同的电梯数等这些属性包含在大楼中。

- 按钮 (各类按钮) (Button (all kinds))。正如前面所讨论的，这个问题的重要的看法就是认为按钮显然属于 (与实现有关的) 人机界面技术的一部分。事实上，一些电梯的运行是由机器人 (由它来传送目的地请求) 或红外线检测器控制的。按钮属于 OOD 模型。

- 门 (Door)。ECS 不需要知道或了解门。门是由电梯的机械系统管理的，电梯只需要报告它什么时候就绪。

- 电梯调度 (Elevator Schedule)。尽管将电梯调度作为对象是有一定道理的，不过电梯调度不是电梯控制系统的基本成分，而是实现这个系统的一种技术。

- 楼层传感器 (Floor Sensor)。到达事件 (ARRIVAL EVENT) 是楼层传感器的与实现无关的说法。

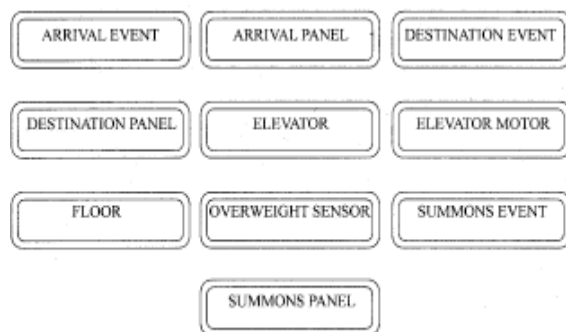


图 8.25 给出了 ECS 初始 OOA 模型的对象-类层图。

图 8.25 ECS 初始 OOA 模型的对象-类层图

对于每个类应给出一个简短的文字描述 (25 个字或更少)，类 DESTINATION PANEL 和 ELEVATOR 的文字描述如下。

- DESTINATION PANEL (目的地面板)：这个对象-类封装了用于表明目的地楼层信息的属性。在收到目的地请求后，该对象-类更新目的地面板，并报告所要去的目的地。

- ELEVATOR (电梯)：对象-类 ELEVATOR 执行电梯的控制和报告功能，封装了控制电梯运动、报告电梯状态和识别电梯是否就绪的各种服务。这个对象-类所封装的属性是关于电梯运行方向、位置和状态方面的信息。

从以上所选择的对象中可以得出以下一些有用的结论：

- 最后得到的对象表不可能被证明为是绝对正确的或绝对错误的。不同的分析员最后选定的对象可能会稍有差异。

- 用户的偏好对模型最后所选择的对象有很大的影响。系统最终将交付给用户使用，因此应该由用户决定模型 (包括所选择的对象) 是否被接受。现在，用户越来越懂得计算机，他们中一些人所坚持要加入的对象可能与他们所熟悉的某种特定实现技术有关。可以劝他们不要那样做，但不能将自己的愿望强加给他们。

- 对候选项所做的选择并不能看成是一个最终结果。在后面还将扩充分析模型，因此可能要不断地对选择结果进行修正。这样做的理由很简单：我们所作出的决策相对来说是基于对各种对象的行为和属性的比较肤浅的理解之上的，仍需要补充大量的附加细节，这就必然要修正目前所建立起来的模型。因此建模过程可以看作是一种反复的、螺旋式上升的方法，而不是一种瀑布式的方法。

• 对象名必须要合适，它所描述的应是一个类，而不仅仅是那个类所执行的一个功能或那个类的一个属性。对象名必须是惟一的，而且必须在应用领域中有意义，而不体现实现技术。对象名应是一个名词或形容词一名词的形式，避免采用名词—动词的形式，“与”“或”等连接词不应出现在名字中。对象—类的描述要清楚，不能有二义性。

8.4.1.3 发现和标识结构

结构层通过建立对象之间的组装及继承关系，标识了对象

的结构体系。在 OOA 模型中，标识结构是很重要的。结构层是用于处理 OOA 模型复杂性的机制之一。

可标识两种类型的结构：一般—特殊结构(又称为泛化—特化结构)和整体—部分结构。前者建立了继承关系，而后者标识了组装关系。

如果父类或泛化类的特征可为其所有的子类特化类共享，就建立起了一般—特殊结构。一般—特殊结构的符号表示如图 8.26 所示。

带有半圆形的连线表示一般—特殊结构，半圆形的圆弧与父类相连。图 8.26 表明子类已发表的文章(PUBLISHED ARTICLE)和已录用的文章(ACCEPTED ARTICLE)继承父类文章(ARTICLE)的属性和服务。

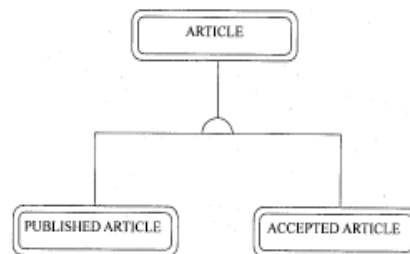


图 8.26 结构层(一般-特殊结构)的一部分

父对象是由若干子对象以某种方式组装而成，这就构成了整体—部分关系。这种关系一般建立在物理组装的基础上，也有可能是其他类型的组装。父对象称为整体对象，子对象又称为部分对象。表示整体—部分结构的符号如图 8.27 所示。

带有三角形的连线表示整体—部分结构，三角形的顶角与整体对象类相连。靠近对象类一端的一对数字表示该对象类的实例需要另一端的对象的数量，称为对象的多重性。逗号之前的数字表示至少有几个对象，逗号之后数字表示的是最多有几个对象。如果两个数字相同则可以只写一个数字。数量不固定可用字母 m 或 n 表示。图 8.27 表示已发表的文章(PUBLISHED ARTICLE)是月刊(MONTHLY ISSUE)的一个组成部分。一期月刊由 1 至多篇已发表的文章组成，1 篇已发表的文章必须是并且只能是一期月刊的组成部分。

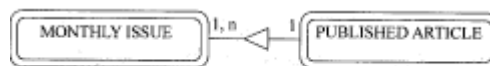


图 8.27 结构层(整体-部分结构)的一部分

继承结构和整体部分结构尽管在概念上不同，实际上都是将可复用的软件成分抽取出来建立单独的具有公共属性或服务的类，继承结构中特殊类通过继承而具有一般类的特征。整体部分结构中整体对象通过组装而具有部分对象的特征。部分对象也可以是另外一个整体对象的组成部分，例如在商店管理系统中，货物类的对象是商店的一个组成部分，也可以是订单类或发货单类的对象的一个组成部分。将组成部分提取出来做为一个可复用构件，在有些情况下，还可以成为多个应用领域的可复用构件。如书这个对象类，在出版社、书店、图书馆等不同领域的管理系统中都可以成为一个部分对象。

1. 如何发现和标识结构

发现和标识一般—特殊结构主要从以下几个方面考虑：

• **考察类的属性和服务。**如果一个类的属性和服务不能适合该类的所有实例，或者不同类中有共同的属性和服务，则应该将共同的属性和服务抽取出来建立一个超类，将不同的属性分别放在子类中。

• **按常识对事物进行分类。**如体育项目有田径、游泳、球类等。超市中的货物可分为食品、日用品、服装、鞋帽等类别，对每类进行细分，可以建立起一般—特殊结构。

• **考虑领域范围内的复用。**分析一个具体系统时，所建立的类应该体现该领域的共同特征，提供在本领域中复用性更强的类。正如以上所讨论的，如果需要将 ECS 安装在具有不同楼层、不同数量电梯的多个建筑物中，就应该建立楼房这个类，它包含楼层、电梯号等属性。

发现和标识整体—部分结构主要从以下几个方面考虑：

• **物理组装关系。**如计算机是由主机、显示器、键盘等部分组成。

• **空间包含关系。**如商场包括了货架、商品、收款台以及售货员和管理人员。

• **组织机构的上下级关系。**如学校由校机关、系、后勤等部门组成，各系又由教研室、实验室、教务科、人事科等部门组成。

• **概念上的组装关系，**指的是抽象事物的整体—部分关系。如书是由前言、章、节等组成。

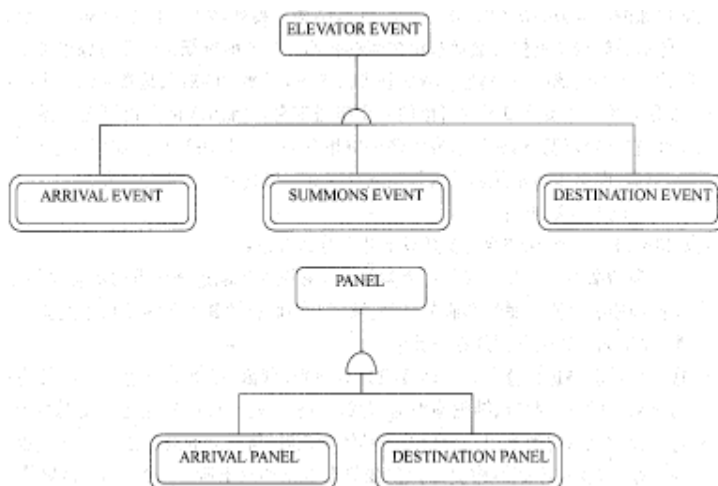


图 8.28 ECS 的一般-特殊结构

标识结构层是个不断反复的过程，在建立 OOA 模型的其他层次时还会修改和调整结构层。

2. 电梯控制系统的结构层

在电梯控制系统中可建立两个一般-特殊结构，如图 8.28 所示。泛化类 ELEVATOR EVENT(电梯事件)用于识别不同类事件中的共同性。SUMMONS EVENT 召唤事件), ARRIVAL EVENT(到达事件)和 DESTINATION EVENT(目的地事件)都共享诸如 event time(事件时间)和 floor-id(楼层号)这样一些属性。类 PANEL(面板)用来处理 ARRIVAL PANEL(到达面板)和 DESTINATION PANEL(目的地面板)中的共同性。

最明显的整体一部分关系就是 ELEVATOR 和它的各个物理部件 OVERWEIGHT SENSOR(超重传感器);ELEVATORMOTOR(电梯马达)、ARRIVAL PANEL(到达面板)和 DESTINATION PANEL(目的地面板)之间的关系，见图 8.29。这些关系都是一对一的。对于 ARRIVAL PANEL(到达面板)也是这样。因为每一楼层上的到达面板与相应的电梯里的到达面板是一致的。

第二种整体一部分关系如图 8.30 所示。这种关系是建立在物理关联的基础上，而不像图 8.29 所示的那种建立在物理包容基础上的整体一部分关系。

8.4.1.4 划分主题层

一个复杂的实际系统中的类可能有几十个甚至几百个，类之间的关系也是错综复杂的，开发和理解这样大规模的系统相当困难，将系统划分成多个主题是解决问题的方法之一。

主题是把一组具有较强联系的类组织在一起得到的类的集合。每个主题可以看作是一个子模型，或一个子系统。把有关的对象-类用一个矩形边框框起来，表示一个主题，每个主题有一个惟一的名称写在主题框内。主题应在应用领域内有意义。

对于中小型系统可以先建立 OOA 模型的对象-类层和结构层以及属性层，然后把联系较强的类，如一个继承结构、一个整体部分结构、或是通过实例连接互相关联的类划分在一个主题中，这是一种自底向上的方式。对于大型系统，可以根据子领域、子系统甚至是组织或地区先将系统划分成几个主题，然后由多个开发小组分别建立 OOA 模型。

对于非常庞大和复杂的系统可以建立多级主题，每级所包含主题的数目不超过 7 ± 2 个。主题之间可能会有重叠，即包含一些共用的对象-类，或者主题之间会有结构连线、实例连线或消息连线，应该适当地调整。将属于多个主题的分类划分到其他类联系最紧密的一个主题中，或将耦合较强的主题合并为一个主题。

图 8.31 是电梯控制系统的 OOA 模型。基本上是根据 OOA 模型中存在的两种结构划分成两个主题。一个是. 电梯管理主题，主要用于控制硬件；另一个主题是电梯调度，它主要检测事件的发生，并对电梯进行调度。需要指出的是，对于比较小的模型，多个主题并不是必须的。

8.4.1.5 定义属性和实例连接

OOA 模型的属性层包括对象属性和对象之间的关系，即实例连接。对象属性是对象内封装的数据；对象只有在其封装的这些数据之上才能工作。实例

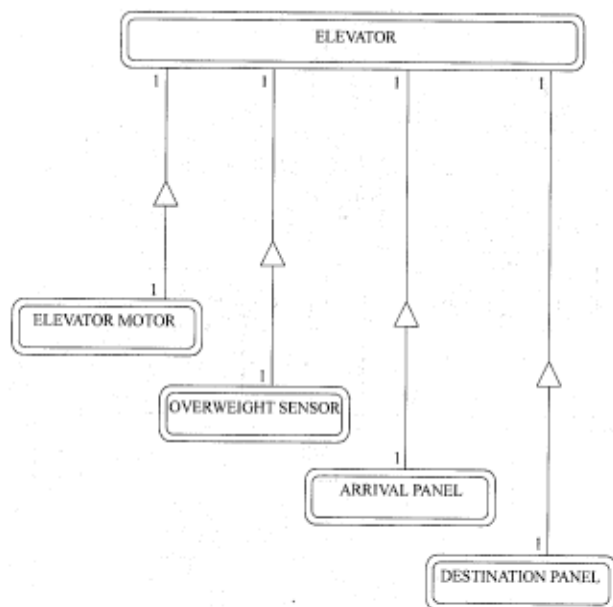


图 8.29 ECS 的整体-部分关系

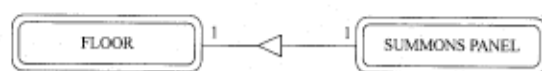


图 8.30 ECS 的整体-部分关系

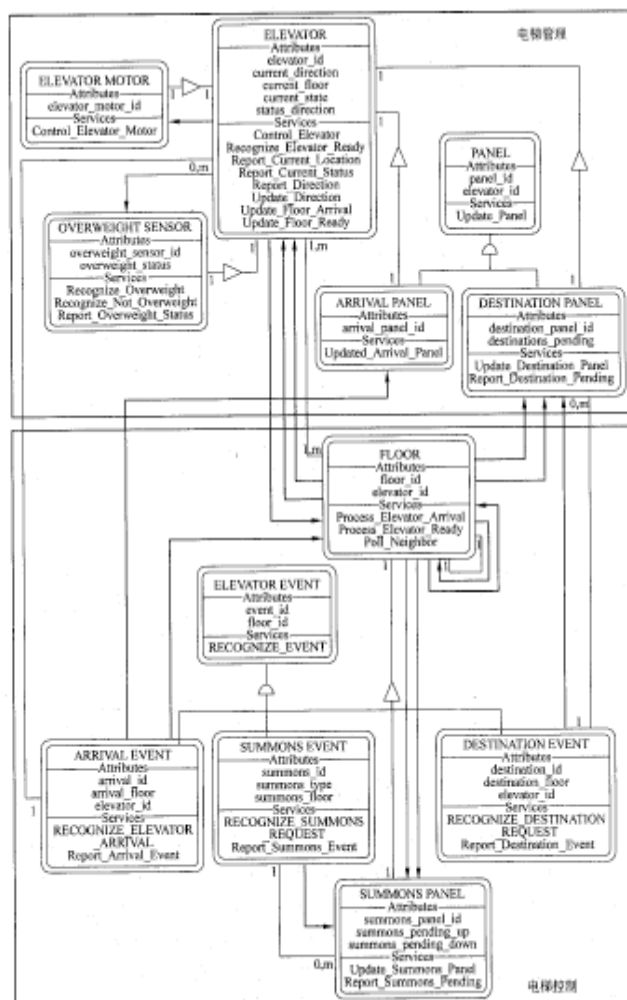


图 8.31 电梯控制系统的 OOA 模型

连接可以看成是一种事务规则或应用领域的限制。实例连接表明一个类中的对象是如何与另一个类中的对象相连的。当实现这些对象-类的时候，这些事务规则指明服务如何执行，以确保与系统的功能相一致。

例如在订购系统中，订户和地址之间有实例连接，建立一个新的订户对象，必然会建立一个新的地址对象，订户对象取消之后相应的地址对象也应该取消，这是必须遵从的应用领域事务规则。符号表示如图 8.32 所示，实例连接两端的数字和整体一部分关系一样表示的是对象的多重性。

整体一部分关系是一种重要的实例连接，两者都是反映了类的实例之间的静态关系，但实例连接的对象之间没有整体和部分的的关系。



图 8.32 类的对象之间的实例连接

实例连接通常用对象指针或对象标识实现，在被连接的两个类中选择多重性数量较小且最好是固定数目的类中设立一个属性，用于表示相连的另一个类中的对象实例。如果多重性的数目不定，则会造成存储空间浪费，而且在实现相应的服务时会产生问题。因此对于多对多的实例连接往往在它们之间增加一个类，将其转换成两个一对多的实例连接。如杂志订阅系统中一篇文章可以有多个作者，一个作者可以写多篇文章。增加一个作者文章跟踪类，如图 8.33 所示。

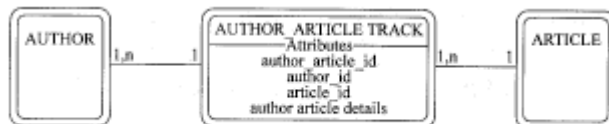


图 8.33 将多对多的实例连接转换成两个一对多的实例连接

1. 如何标识属性

基本的做法是标识属性，修改 OOA 模型的其他层以适应这些属性，如果有必要的话修改对象，然后重复这个过程。

可以从以下几个方面考虑标识属性：

- 所标识的问题领域的概念中有许多都可以表达事物的特性，例如销售系统中，商品的价格、数量、生产日期等。

- 根据系统的功能，确定对象应该有什么属性。如杂志订购系统，需要计算顾客订购杂志应付的金额，则相应地建立杂志名称、份数、订阅的开始时间和结束时间，以便根据单价、份数和时间计算应付的金额。

- 为了惟一地标识一个对象，往往需要建立一个标识数属性。如工作证号、学号、货物号、订单号等，以便于对其他属性值进行查询、修改等操作。

- 为了区别对象的状态，往往需要增加一个属性，如办公系统中，一张表格在由某人填写的过程中，领导不能审批。填写完毕提交部门领导审批时，填表人不能修改。领导要求修改时，才能修改。领导签字批准后不能再修改。为了区别一张表格的不同状态，需要建立一个属性，以便设置成不同的属性值，控制执行不同的系统行为。

- 为了表示实例连接，需要设置相应的属性。如杂志订单类，应设置属性订户编号、收件人的编号给出相应的订户以及收件人对象。整体一部分结构是一种实例连接，也应该建立相应的属性。

- 如果已经建立了实体一关系图，实体的属性表示了存储数据的需求。这些应存储的数据必须出现在 OOA 模型中。实体可能对应于某一对象，这样，实体属性就成为对象属性。也许一个实体不只对应于一个对象，那么这个实体的属性必须分配到 OOA 模型的不同对象之中。

对于初步标识的属性应该进行审查，可以审查以下这些方面的问题：

- 继承结构中父类子类属性的一致性。子类通过继承可能得到的属性不应该重复定义。

- 可以从其他属性中导出的属性应该去掉。例如售出的一种货物的金额可以根据单价和数量计算出来，不应该保留许多冗余的信息，但是需要经过比较复杂的计算才能从许多其它属性中得到的值可以做为一个属性。

- 在标识一个对象时，如果其属性的值“不适用”，就应当对其重新考虑。一个对象的所有属性都必须能应用于该类的所有实例中，即使其值为空。“不适用”的值是不能被接受的。

以市政府信息管理应用领域为例。因为市政府要发放 LICENSE(许可证)，因此首先标识对象 LICENSE，见图 8.34，LICENSE 的属性可以有发放日期、付费、有效期等等。但市政府所发的许可证有很多类型，如婚姻许可证、养狗许可证、钓鱼许可证等等。有效期属性可能对婚姻许可证就不适用，属性配偶姓名对养狗许可证就不适用，出生日期对钓鱼许可证也不适用。因此需要重新考虑对象 LICENSE，通过建立泛化-特化结构将 LICENSE 细分为 MARRIAGE LICENSE(婚姻许可证)、FISHING LICENSE(钓鱼许可证)和 DOG LICENSE(养狗许可证)，图 8.35 给出了修改后的结构。

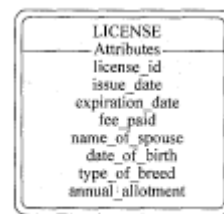


图 8.34 对象的初步选择

2. 如何标识实例连接

首先根据系统的功能描述分析对象之间的静态依赖关系，例如订购系统中订单和订户、发货地址、货物价格等类的对象具有静态关系，应该在 OOA 模型中建立这些类的实例连接，然后标识实例连接的多重性。

应该注意实例连接的一端应给出对应的另一端的对象的数目，这与实体关系图中所标的数目正好相反，在实体关系图中应标出对应另一端的本端的实体数目，如图 8.36 所示。

在某些系统中一个人员可以有多种身份，如杂志订购系统中客户可以是订户、收件人，也可以是作者、编辑等等。可以将这些类的对象共同的特征抽取出来，建立一个身份类作为父类，各种身份的不同特征放在子类中。客户类的实例通过与身份类的实例连接得到不同身份的特性，如图 8.37 所示。

这样做的好处是动态变化的属性和服务可以通过实例连接得到，一个客户具有多种不同身份或者他的身份变化时，所具有的不同属性和服务放在 ROLE 类的子类中定义。

3. 电梯控制系统的属性层

在图 8.31 电梯控制系统的 OOA 模型中给出了 ECS 类的属性以及实例连接。

对于每个属性都应该给出相应的文字描述，以下是 5 个属性描述的例子。

- arrival floor(封装在 ARRIVAL EVENT 对象-类中)：发生到达事件的楼层。
- current-direction(封装在 ELEVATOR 对象-类中)：表示当前电梯的运行方向。如果电梯正停靠在某一楼层上，那么该属性就表示电梯前一次运动的方向。
- current-state(封装在 ELEVATOR 对象-类中)：表示当前电梯的状态，有效值为 busy(忙)、ready(就绪)、open(打开)等。
- elevator id(封装在 FLOOR 对象-类中)：惟一标识 Elevator 的某一实例，在 Floor 的某一实例发送出来的消息中用到。

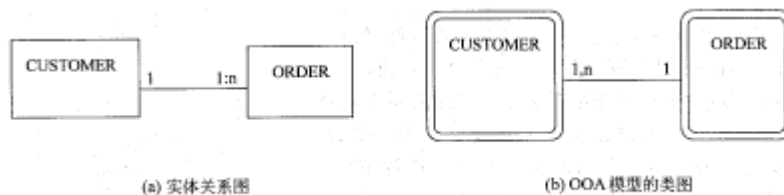


图 8.36 实例连接的标识

- summons-type(封装在 SUMMONS EVENT 对象-类中)：表示所请求的召唤方向，有 UP、DOWN、NONE 3 种。

所建立的一个实例连接就是：ARRIVAL EVENT(到达事件)必须明确地与一个 ELEVATOR(电梯)相关联，而 ELEVATOR 可能与零个或多个 ARRIVAL EVENT 相关联。这是什么意思呢？首先，ARRIVAL EVENT 必须报告某个特定电梯的到达。在应用领域中，如果 ARRIVAL EVENT 只是说明有电梯到达而没有具体说明是哪一部，这是没有意义的。同样，这个实例连接还要求 ELEVATOR 能与零个或多个 ARRIVAL EVENT 相关联。之所以是多个事件，是因为希望一部电梯能多次到达某个地方。零个事件是因为电梯可能哪里都没到达过，例如，当早上刚启动执行 ECS 时。

一般不应该建立冗余的实例连接。例如，ARRIVAL EVENT(到达事件)和 ARRIVAL PANEL(到达面板)有关系吗？事实上是有的。这种关系隐含在 ARRIVAL EVENT 与 ELEVATOR 的关联中，而 ELEVATOR 又与 ARRIVAL PANEL 有关联(记住，整体一部分关系也是一种实例连接)。因此在 OOA 模型中没有必要显式地表示这种实例连接。但如果这种关系对用户非常重要、非常有意义，或者用户想要将它们明显地表示出来，那么就需要显式地建立冗余的实例连接。

在图 8.31 电梯控制系统的 OOA 模型中给出的其他实例连接就不再这里一一讨论了。

8.4.1.6 定义服务和消息

类-对象的服务，加上类的实例之间的消息通信，共同组成了 OOA 模型的服务层。服务层建立了对象之间的动态关系。例如，在图 8.38 中，类 SUBSCRIPTION(订阅)和 SUBSCRIBER(订户)都分别提供一定的服务。另外，它们的对象相互之间通信。对象请求其他对象的服务称为发送消息，消息连接用有向箭头表示。图 8.38 中的消息连接表示 SUBSCRIPTION(订阅)的诸多服务中有一个负责与 SUBSCRIBER(订户)的某一个服

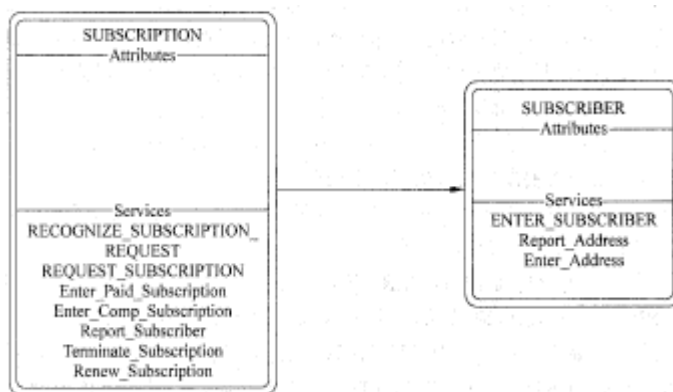


图 8.38 服务层的一部分

务通信。

1.如何发现和标识服务层

标识服务主要从系统的功能出发,逐项审查用户需求,将它们分配到相应的对象中,作为对象的服务,这些服务包括对属性值进行计算、加工处理,输入输出信息或进行控制操作。对于为了遵守严格封装原则所需要的读、写属性的操作,不一定标识在对象的服务中,以便使系统的分析模型简洁、突出表达事物的重要行为。

每种状态下,对象的行为规则不同,应该标识成不同的服务,对象的状态转换是由相应的服务引起的。

标识了每个对象中必须封装的一组服务后,应该将服务与对象的属性相比较,验证其一致性。如果已经标识了对象的属性,那么每个属性就必须关联到某个服务,否则这个属性对于这个对象来说就形同虚设,永远不可能被访问。

然后,画出对象之间的消息通信路径,协调系统的行为。一种方法是访问每个对象,然后问,“好吧,对象先生/女士,……你是如何生成的?你下一步要做什么?然后呢?……”这个生存期要求给出许多从对象生成到消亡的状态。每一状态的改变都关联到对象之间消息的传递。这是一种自底向上的方法,从对象着手,逐渐向上分析。

相反的,可以使用事件-响应方法从系统行为着手。一个对象必须识别系统中发生或出现的每个事件,然后生成发送给其他对象的消息,那些对象最后必须建立响应。因此就能够知道每个服务必须接收、处理以及生成什么消息。这是一个自顶向下的方法。从系统行为着手,然后逐渐分析到对象。

2.电梯控制系统的服务层

在图 8. 31 电梯控制系统的 OOA 模型中给出了 ECS 中每个类的服务以及消息连接。对于每个服务和消息都应该给出相应的文字描述,以下是服务和消息描述的例子。

- 服务: Control Elevator(控制电梯,封装在 ELEVATOR 对象-类中),这个服务控制一个给定电梯(由 elevator-id 指定)的运动。控制过程如下:

第一步,根据接收到的消息:

如果是[UP|DOWN]([上升|下降]),那么将属性 ELEVATOR(elevator_id).current_direction 设置为[UP|DOWN]([上升|下降])。将属性 ELEVATOR(elevator_id). Current_status 设置为 BUSY(忙)。

如果是 STOP(停止),那么将属性 ELEVATOR(elevator_id). Current_status 设置为 STOPPED。

第二步,向类 ELEVATOR MOTOR 的由 elevator_id 所标识的实例发送一个单向消息。

该消息为(elevator_id, [UP|DOWN|STOP])。

第三步,挂起类 ELEVATOR 执行此服务的实例,直到接收到下一个消息。

- 消息:Control Elevator Motor 控制电梯马达),这个消息由 Elevator.Control_Elevator 接收。该 Elevator 实例发送一个单向的消息给由 elevator_id 所标识的 Elevator Motor 实例。这个消息是(elevator_id, STOP)。这个消息由 Elevator Motor. Control_Elevator_Motor 接收,从而使 Elevator Motor(由 elevator_motor_id 标识)停止运转。然后电梯的机械系统把电梯门打开。

8.4.1.7 事件响应对象交互图

建立了 OOA 模型之后,为了检查模型是否体现了用户需求,应该根据系统识别的所有事件,以及如何响应这些事件的描述,画出事件响应对象交互图,简称 EROI 图。EROI 图是标识和描述对象相互通信的极其有用的工具,对于每个事件画出一个对应的图,表明了由哪个对象来识别事件的发生,产生什么消息、其他哪些对象接收这些消息,并产生什么响应。

如图 8. 39 所示 EROI 图中有许多垂直的线条,每一条垂直的线条都代表了 OOA 模型中的一个对象-类。EROI 图显示了由哪个对象-类识别事件的发生,这由符号⊙表示。消息由带方向的箭头→表示,如果还有同步的响应产生,就采用双向的箭头↔表示。消息从识别事件的对象-类发送到目的地对象-类。EROI 图的垂直方向代表时间。也就是说,如果一个消息画在其他消息的下方,就认为该消息晚于其他消息发生。从同一点发送的消息表示消息的次序没有意义。

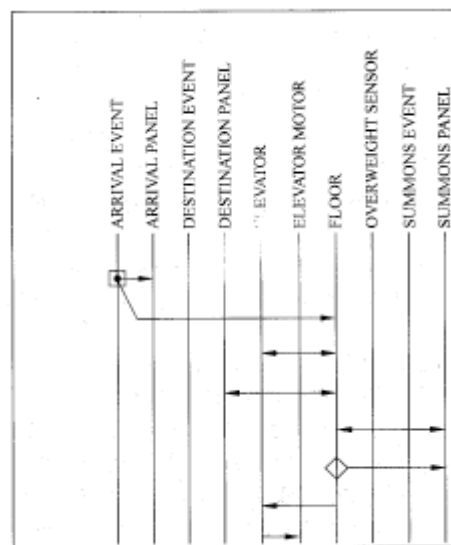


图 8.39 电梯到达调度的楼层

图 8. 39 给出了 ECS 中电梯到达调度的楼层的 EROI 图。每当电梯到达一个楼层时,都要询问相应的 Floor 以确定该楼层是否为调度楼层。Floor 是通过与相应的 Destination Panel、Elevator 和 Summons Panels

通信而作出最后决定的。这些对象都封装了一些关于目的地请求和召唤请求的知识，在这个例子中为一个目的地请求或一个召唤请求。所采取的相应动作就是使电梯停靠在当前楼层，参见图 8. 39。

电梯到达调度的楼层的 EROI 图的说明如下：

- ARRIVAL EVENT. RECOGNIZE_ARRIVAL_EVENT 服务检测是否有电梯到达。

• 建立 Arrival Event 事件。其属性有 arrival_id(一个任意的标识符)、elevator_id(生成到达事件的电梯)以及 arrival_floor(生成该事件的楼层)。Arrival Event. Report_Arrival_Event 将一个消息单向发送给与 elevator_id 有关联的 Arrival Panel。这个消息是 (arrival_floor)。Arrival Panel. Update_Arrival_Panel 接收这个消息并相应地刷新到达面板。Arrival Event. Report_Arrival_Event 服务发送一个消息给与 arrival floor 有关联的 Floor。这个消息为：(arrival_id, report_elevator_id, report_arrival_floor)。Floor. Process_Elevator_Arrival 接收这个消息。这个服务可以发出许多消息。一个双向的消息是发送给与 report elevator_id 有关联的 Elevator。这个消息是：(report_status_direction?, report_current_direction?)。该 Elevator 实例的 Elevator. Report_Status_Direction 服务通过更新相应的 report_status_direction 和 report current_direction, 发送回针对此消息的响应，给出属性 status_direction 和 current_direction 的值。发送给与 report_elevator_id 有关联的 Destination Panel 的消息是双向的。该消息是：(report_arrival_floor, destination_pending- above?, destination_pending-below?)。该 Destination Panel 实例的 Resport_Destination_Pending 服务根据 destination_pending 的属性和 report_current_floor 的值，确定 destination_pending_above 的值[TRUE|FALSE]和 destination_pending_below 的值[TRUE |FALSE], 从而发送回一个响应。发送给与 arrival floor 有关联的 Summons Panel 的消息是双向的。该消息为：(report_summons_pending_up?, report_summons_pending_down?)。相应的 Summons Panel. Report_Summons_Pending 通过更新具有 summons_pending_up 和 summons_pending_down 属性值的参数，响应这个消息。服务 Process_Elevator_Arrival 根据电梯调度算法确定若到达楼层有召唤请求时，电梯是否应当停下来。该算法中处理的数据来自于 Elevator, Summons_Panel 及 Destination_Panel 实例对到达楼层所发出的消息的响应。该算法确定 Updated_Current_Direction, Updated_Summons_Pending_Up、Updated_Summons_Pending_Down 以及 Updated_Status_Direction 的值，并在电梯运行的过程中响应沿途的召唤请求和目的地请求。它将会在第 1 层楼和第 40 层楼发出 STOP 命令。

对于这个事件，要么存在一个有关 Floor 实例的目的地请求，要么存在一个正确的召唤。算法相应地更新 Summons Panel 实例，发送相应的消息给 Elevator 实例，更新 status_direction, current_direction 和 current_status 的属性。Elevator 接收这个消息，并向与这个 Elevator 实例有关联的 Elevator Motor 实例发送消息作为响应。Elevator Motor 实例的服务 Control- Elevator- Motor 接收到这个消息，使得电梯马达停止转动。然后电梯的机械系统使电梯门打开。

- Arrival Event 事件结束。

显然，EROI 图必须与 OOA 模型一致。EROI 图中的消息与服务层的 OOA 模型中的消息有着对应关系。EROI 图中对象-类之间交换的消息表现了模型的动态行为。

8.4.2 面向对象的设计

分析过程是建立基本系统行为的过程，而设计过程则被视为定义系统构造蓝图的过程，依据系统构造蓝图便可以在特定的环境中实现系统。

1. OOD 模型

扩展 OO. 4 模型，就得到面向对象设计(object-oriented design, OOD)模型。这样做有利于将分析转化成设计(有时这种转化工作是很繁重的)。OOD 模型和 OOA 模型一样，包含有 5 个层次，即对象-类层、属性层、服务层、结构层和主题层。但同时 OOD 体系结构中又引进了 4 个“组成部分”，这些部分分别是：问题领域部分、人机交互部分、任务管理部分和数据管理部分。

下面将分别简要地讨论这些部分。首先，问题领域部分是指那些执行基本应用功能的对象。事实上，可以将 OOA 模型复制过来当作问题领域部分的初始版本，然后逐步地细化这个初始版本，使其最终能解决实现限制、性能缺陷等方面的问题。

人机交互部分(HIC)指定了用于系统的某个特定实现的界面技术。采用图形用户界面时 HIC 中有许多窗口对象，通过使用整体一部分结构，将窗口对象进一步分解为各种文本域、选择按钮、图符等。假定使用某种 GUI 构件软件包，如商品化的类库，或者某种能够为屏幕上的显示窗口等创建可运行软件的生成器。在这种情况下，只需要对软件包输入合适的参数，该软件包就能提供所有的文本域、选择按钮、图符等窗口的组成部分。因此，只需要填写一些细节作为窗口对象中的属性就可以了。界面技术的细节与系统所做

的工作被分离开来,这种方法的好处是提高了可复用性。例如,当一个给定的应用系统从 GUI(图形用户界面)升级到语音响应接口时,只需替换其中的人机交互部分,系统的其他部分都不必改动。

OOD 模型的任务管理部分(TMC)则指定了那些创建系统时必须建立的操作系统部分。首先,要标识一些新的类,这些类主要负责处理并发问题、中断、调度(在操作系统一级)以及其他有关特定平台的一些问题。正像在 HIC 中所做的那样,TMC 把有关特定平台的处理机制对系统的其他部分隐藏了起来。这样,如果决定将系统移植到另一个平台上,那么只需替换 TMC 的类就可以了。

最后,数据管理部分(DMC)定义了那些与所用数据库技术接口的对象。基本的策略是为每个关系表建立一个对象,这些关系表将在数据库管理系统中创建。这种对象封装了数据(行)将如何建立、读、写以及删除的秘密。其他有些对象需要访问关系表,因此需要建立这些对象与该对象之间的消息。和人机交互部分一样,数据管理部分也可以看成是事务分离原则应用的又一个例子。在这里,数据库技术的细节与基本的系统功能被分离开来。

2.建立电梯控制系统的 OOD 模型

电梯控制系统的人机交互部分由各种电梯按钮、指示灯以及它们的接口组成。ECS 中不存在需要进行设计的屏幕、窗口或其他所谓的“常规的”用户界面形式,参见图 8.40。

任务管理部分如图 8.41 所示。电梯控制系统没有数据存储的要求,所以不需要建立数据管理部分。所有在对象中存储的数据都驻留在计算机内存中,在系统掉电时,这些数据就全部丢失。

现在以召唤事件为例完整地说明系统功能的执行过程,如图 8.42 所示。

当一个召唤按钮被按下时,在 ECS 之外就产生了一个中断。同时,一个二进制数就存储到输入寄存器中。召唤按钮一共有 78 个(除最底层和最高层各有一个外,其他楼层每层都各有两个)。向下的召唤按钮编码为奇数,向上的召唤按钮编码为偶数。零表示当前没有按钮被按下。因此输入寄存器中存放的二进制数的范围应是 00000000,00000010 到 01001111。

在 ECS 中,召唤按钮被按下的第一个反应是类 SUMMONS INTERRUPT(召唤中断)被唤醒,并报告“我接收到了一个召唤 1”相应的 SUMMONS INTERRUPT 的对象就向 INPUT REGISTER(输入寄存器)对象发送一个消息,询问输入寄存器的当前值。这时底层执行程序就锁住输入寄存器中的值直到该值被读出。中断优先级规定这个值只能由 SUMMONS INTERRUPT 对象获取。然后这个 SUMMONS INTERRUPT 对象就向类 SUMMONS BUTTON(召唤按钮)发送一个消息,告诉它“你里面有一个按钮被按下了!”

对于输入寄存器来说,数字就是数字,任何两个数字之间没有什么不同;但对于类 SUMMONS BUTTON 来说,不同的数字则代表了不同的意义。类 SUMMONS BUTTON 封装了如何将寄存器值映射到按钮号码的机制。它通知相应的 SUMMONS BUTTON 对象“你被按下了,现在你该开始工作了!”。这时 SUMMONS BUTTON 对象就向类 SUMMONS EVENT 发送一个消息,接着开始进行事件的处理。

很明显,这种封装需要做许多工作。事实上,这个开销是我们心甘情愿付出的代价,因为我们希望系统只需要修改一小部分就能适应一个新的环境、而且这些改动不应影响系统的其他部分。例如,可能决定修改按钮编号模式。有些楼层或许需要第 3 个按钮用于召唤运货电梯。在这种情况下,只需改动一个类,即 SUMMONS BUTTON 类就可以了。而系统的其他部分则对这

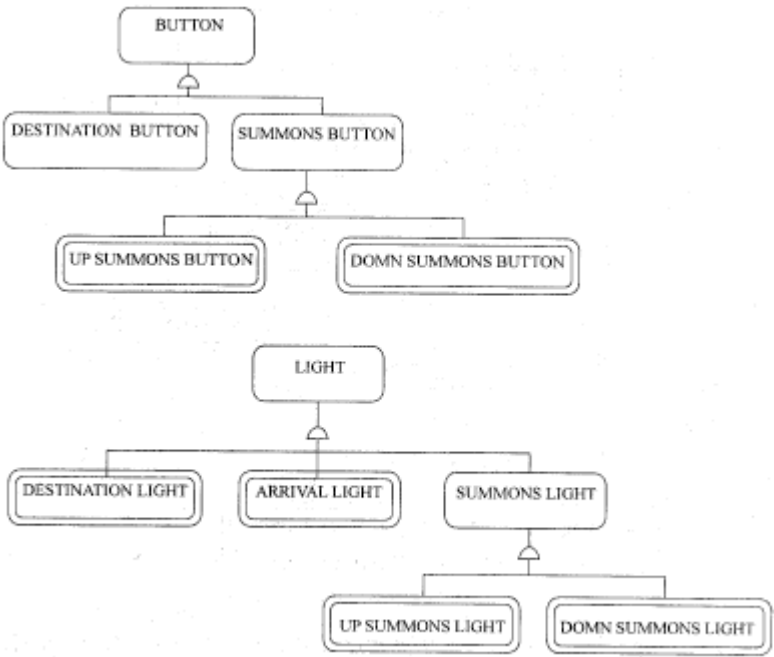


图 8.40 ECS 的人机交互部分(HIC)

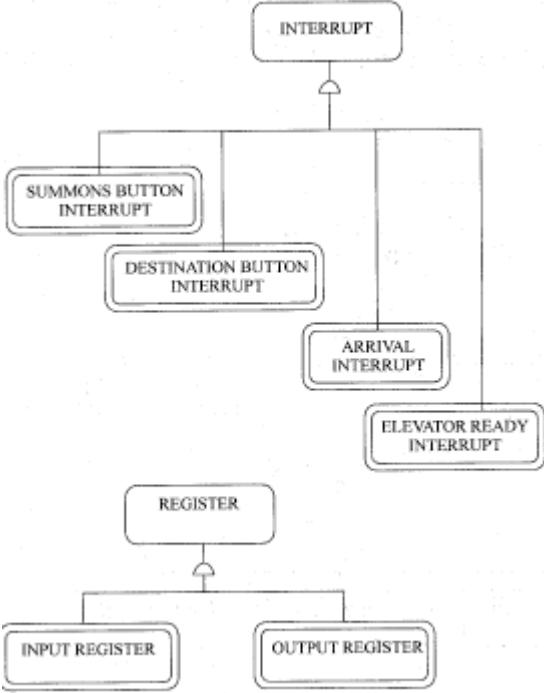


图 8.41 ECS 的任务管理部分(TMC)

种改动置之不理。

8.4.3 OOD 文档的编写

如图 8.43 所示, OOA 文档应该包括 5 个层次 OOA 模型 EROI 图以及 OOA 模型的详细说明。复杂系统的 OOA 模型应表示成一组主题, 每个主题的大小最好能用单张图表示, 详细描述包括类、属性、服务、消息以及用户场景(即事件响应)。

OOD 模型包括 4 个组成部分, 每个组成部分的 OOD 文档都与 OOA 文档相同。

传统的方法分析和设计采用不同的表示法、不同的用词、不同的工具等等。而面向对象方法分析和设计之间的界限是模糊的, 从分析、设计到编码等软件开发的各个阶段中, 建立的模型与程序的主要成分是对应的, 表示方法是一致的。这有利于文档的编写及理解, 而且基本程序框架和许多代码可以用软件工具生成出来, 大大提高了开发的效率与质量。因此, 面向对象方法已经得到了广泛的应用。

8.5 软件复用技术

8.5.1 软件复用的概述

实施软件复用(software reuse)的目的, 是使软件开发工作进行得更快、更好、更省。“更快”是指及时提供软件产品而在市场竞争中赛过竞争对手;“更好”是指软件产品具有更可靠的质量;“更省”是指软件开发和维护成本更低。

换句话说, 实施复用的目的是快速、可靠、低成本地完成客户合同。具体说, 实施复用是将冗余工作减到极小;并提高工作结果的可靠性(因为可复用构件系统的初次开发过程中实施了阶段审查和评审、单元测试和系统测试, 随后还多次接受了现场测试);从而大幅度缩短软件开发周期(从数年减到数月, 从数月减到数周)。

日美的一些大公司资料表明, 软件复用率最高可达到约 90%;而且软件复用使得企业在及时满足市场、提高软件质量、降低开发费用和维护费用等方面, 均有显著改进。

例如, AT&T 的电信操作支持系统软件复用率达 40%-92%; Motorola 公司在为编译器和编译器工具编写测试包时, 复用率达 85%; Eyicson AXE 公司的电信开关系统产品, 复用率达 90%。HP 公司早在 1984 年就开始开发可复用构件, 1987 年建立复用库, 据 20 世纪 80 年代几个方面的统计, 复用率达 25%-50%。HP 公司在 1990 年开始实施一个“宏伟”的复用计划, 收集并研究最好的体系结构、过程、组织结构, 打算将其装备到公司的各部门, 但实践证明此法不通, 后来 HP 采用典型示范先行的系统的过渡方法, 成功地在公司内全面实施复用。

日本的软件大公司在 80 年代中期复用率就达 50%左右; 97 年的一份报告说, Hitachi 的 Eagle 环境复用率达 60%-98%、该环境为软件工程师提供了可复用的程序框架以及函数过程。

除了复用率之外, 在企业的经营管理方面, 也取得了理想的效益。例如, 产品上市时间缩短 2-5 倍; 产品的缺陷密度减少 5-10 倍; 产品的维护费用减少 5-10 倍; 软件开发总费用可减少 15%-75%。

近年来, 新一代软件复用技术是以面向对象“构件”为关键, 甚至复用大粒度的“对象”(object), 快速地开发成应用软件。基于“构件”(component)的软件技术的成熟程度和推广速度在迅猛增长。这些新技术包括微软的 Visual Basic, ActiveX, OLE(Object Linking and Embedding), SUN Java, OMG 的

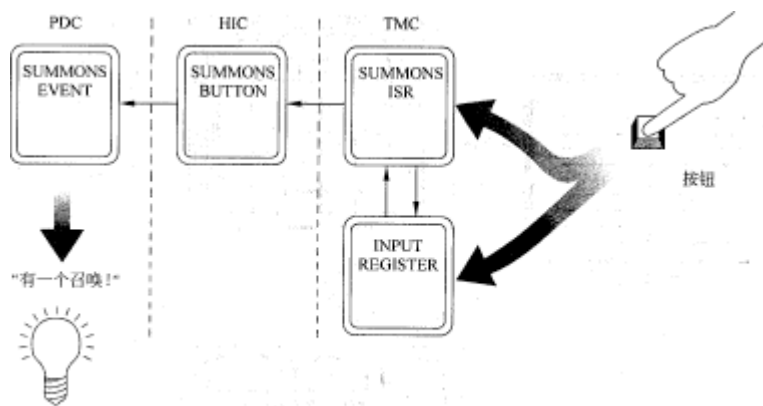


图 8.42 “召唤事件”完整的执行过程

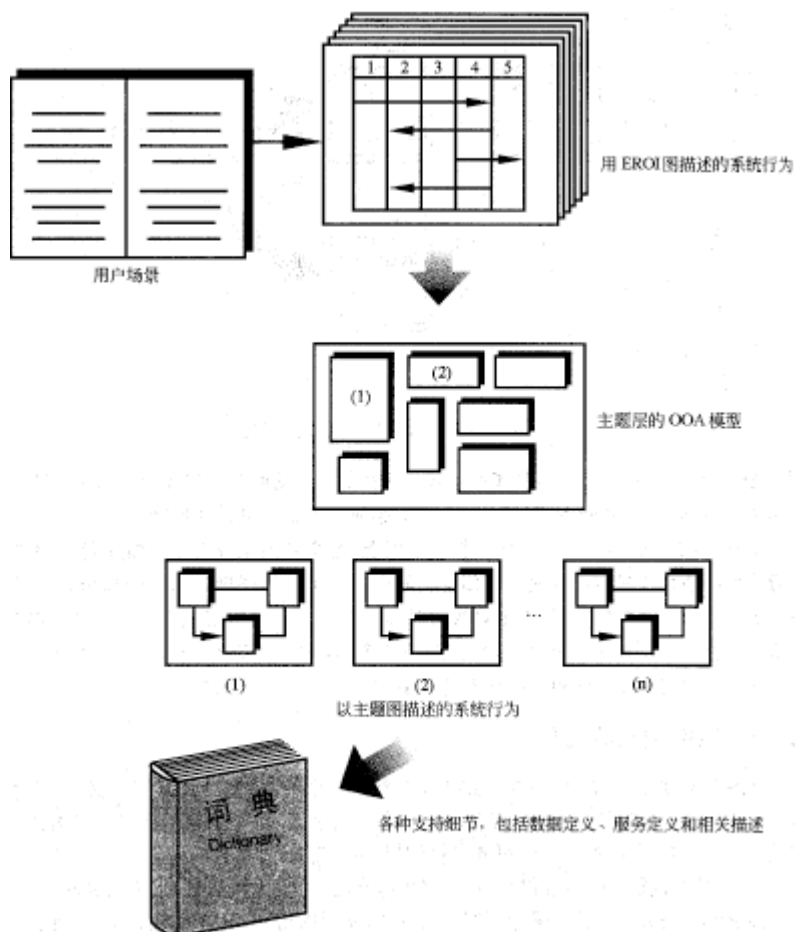


图 8.43 OOA 文档的一般结构

CORBA(Common Object Request Broker Architecture)、IDL(接口定义语言)等等。而非面向对象语言,如 COBOL 和 Fortran,在复用实践中也取得了显著成就。

当日益增多的基于构件的应用软件开发出来的时候,人们认识到,仔细定义体系结构和机制方面的重要性,将会在更大程度上有利于构件的复用。此外,还需要面向构件的建模方法和相应的支撑工具。

应用软件(即客户合同软件)和构件应当由不同群组的人们进行开发,但他们又需要密切协同工作,以满足客户的要求。

不同的作者提到构件,其含义略有所异,本文所说的构件是建立在对象技术基础上。此外,软件复用的实际经验还表明,体系结构(architecture)、过程(process)和组织结构(organizational structure)的有效管理都很重要。下面将逐步讨论。

8.5.2 软件开发过程

8.5.2.1 以往的软件开发技术不能满足复用的需要

本节将以软件复用为目标,从工程、过程、组织管理、经营 4 个方面,评述以往的软件开发(特别是应用软件开发)技术和经营管理方法。

1. 工程、

此处所说的“工程”(engineering)是指“软件开发工程”(software development engineering)。其技术和方法不能很好地满足复用的需要,表现在:

- **缺乏界定“复用”的机制。**为了软件复用,需要沿着软件开发流程的需求分析、设计、实现、测试等阶段,分析它们的描述模型,并明确地界定出潜在复用的部分,被界定之处表示可被复用,或可被可复用构件所替代。而以往的软件工程缺乏这种界定机制。

- **缺乏制作可复用构件的方法。**这反映在许多方面,例如,不能有效地挑选出可复用构件并对之进行强化;缺乏对构件进行打包、文档、分类、界定的技术;缺乏有效方法进行构件库系统的设计和实现;缺乏对构件库进行访问的良好机制。

- **不成熟的体系结构设计致使可复用构件缺乏足够的灵活性。**如果一个构件很死板,它的复用机会就很少。过去的办法只是对构件进行调节,使之满足新需求,而没有认识到分层的体系结构对构件的灵活适应性的重大影响。

- **缺乏实施复用的工具。**为了实施复用,需要一系列新工具,并将它们集成到面向复用的支撑环境中,而过去的软件工程支撑环境缺乏这方面的工具。

2. 过程

此处所说的“过程”(process)是指“软件开发过程”(software development process)。以往的软件开发过程,没有设置“复用”思考点,让开发者思考“可否将过去已做过的某件东西替换成可复用构件系统”、“如何安排构件系统”;在软件的分析、设计、编码阶段后期安排的评审、审查、走查等过程中,也没有关注到软件复用问题;复用体系设计师的潜在作用以及可复用构件开发者的作用均未曾定义。

3. 组织管理

以往的软件开发实践,只关注一个应用项目,而复用实践要关注到覆盖整个应用领域的诸多项目。只关注一个项目的管理与应用领域大包揽式产品开发的管理,两者之间的差异很大,新的管理机构应当能够同时抓住这两种关注点。

此外,还有文化的问题,例如,有人不相信本单位内的其他人的工作成果,不愿依赖他人;有人不愿做复用者,担心丢失自己的创造性;可复用构件开发过程的质量控制措施不力,生产出的可复用构件质量不理想。

这些管理模式、培训教育方面的新问题,需要有相应的新组织结构的支持。

4. 经营方式

实施“领域工程”(domain engineering),开发可复用构件、建立构件库、职工的培训教育等,这些工作都需要经费支持。而此项投入的资金,一直到出现需要复用这些构件的项目才开始回收。只有充分认识到复用的经营性利益,才会下决心认真实施复用。

总之,软件开发单位为实施复用,必须树立实施复用的经营方式,提供经费支持,设置相应的组织结构,在组织上予以支持,还要对职工进行培训教育,使本单位的应用工程师们都渴望采用复用技术,更“快、好、省”地完成应用系统开发任务。

8.5.2.2 软件复用需要改变软件开发过程

根据近几年的经验,复用界(包括复用的研究界和实践界)已有了这样的共识,为了获得系统地复用的效果,须在软件开发的过程方面进行重大的变革。

以往的软件开发过程,每个项目都是从头做起,不同项目之间的共享部分甚微,复用方面至多是每个

开发者复用自己的积蓄。而新的开发方法，则将诸多应用开发项目与界定并开发可复用构件联系在一起。这样做，就必须彻底审视开发单位的经营方式和组织结构，在开发过程方面要做重大变革。要从如何快速、可靠、低成本地完成客户合同任务的角度，重新思考属于软件的每一件事情。

首先，要认识到可复用构件实际上是开发单位的“资产”(asset)，需要投资获得，并用来生产应用软件。也就是说，此项投资在以后的复用过程中将得到回报。为此，需要认真界定出可复用资产，开发它们，并进行打包、编制文档，以方便应用工程师复用。

其次，开发单位必须建立新的系统工程过程，使开发者有机会来思考和确定复用方案，使应用工程师有机会进行挑选所需的可复用构件。

系统的软件复用如图 8.44 所示，由可复用资产的开发、管理、支持和复用 4 个过程组成。工作在可复用资产开发过程中的是构件开发者和领域工程师，工作与应用项目开发过程中的应用工程师。下面将逐一地讨论这 4 个过程。

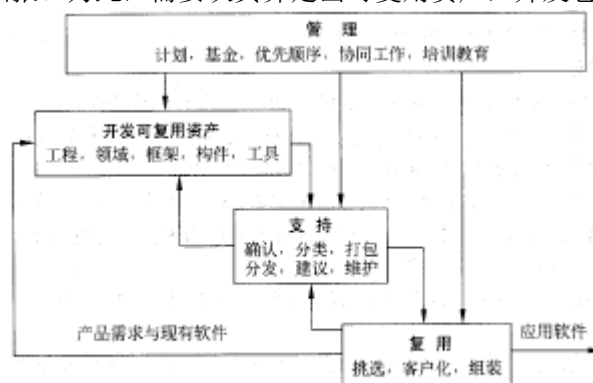


图 8.44 系统的软件复用牵涉到 4 个并行的过程

- **可复用资产的开发。**此开发过程要界定并提供可复用资产，以满足应用工程师的需要。可复用资产的来源可以是新开发的、再建设的、购置的。可复用资产有多种类型，如代码、接口、体系结构、测试、工具、规范等。此开发过程的活动包括：清理现有的应用软件和资产，列出其详细清单，并进行分析；进行领域分析；体系结构定义；评估应用工程师的需求；进行技术改革；可复用资产的设计、实现、测试和打包等。

- **复用。**复用过程是使用可复用资产来生产客户合同应用软件的过程。此过程的活动包括：检验领域模型；收集和分析最终用户的需求；从可复用资产中挑选合适的构件，并进行必要的客户化调节；设计和实现可复用资产未覆盖到的部分；组装出完整的应用软件，对之进行测试。

- **支持。**支持过程的任务是全面支持可复用资产的获取、管理、维护工作。此过程的活动包括，对所提供的可复用资产进行确认；对构件库进行分类编目；通告和分发可复用资产；提供必要的文档；从应用工程师收集反馈信息和缺陷报告。

- **管理。**管理过程从事计划、启动、资源、跟踪，并协调这几个过程。管理过程的活动包括：对新资产的获取工作进行优先性排队；安排其施工日程；分析其影响；解决有关的矛盾；进行培训；进行指挥。

8.5.2.3 领域工程和应用系统工程

当今大多数的软件复用中，涉及到“领域工程”(domain engineering)这项重要活动。在此活动中需要一整套方法，界定潜在的可复用资产，并确保可复用资产能被频繁复用的体系结构。而针对一个应用系统的开发过程，称为“应用系统工程”(application system engineering)。

1. 领域工程

虽然可以借用应用系统工程中的一些术语，通俗地描绘领域工程，诸如，结构分析/结构设计、面向对象分析/面向对象设计等，但二者有很大差异。主要的差别是，领域工程要适用于一族应用系统(一个领域中的诸多系统)，而不只是一个应用系统。而且领域工程比多个应用系统工程简单相加更复杂。人们往往没有充分理解它的难度，不能事先计划好，从而无法进行有效管理。

领域工程要在选定的领域中界定出事物的共性与可变性，要为诸多的应用和构件定义一个体系结构，要开发一系列的可适度扩展的构件。总之，领域工程企图寻求可复用资产，以支持后继的一系列的应用系统工程。

领域工程的主要活动有：针对某领域中的主要“特性”(feature)进行建模；界定该领域中各项特性的共性与可变性(可以采用适当的可变性机制，如继承性或模板，来描述可变性)，对领域模型中的共性体和变体进行分组和组群；设计该领域的体系结构，如依赖机制、特性、子系统、变体；开发可复用资产集；推出经过确认的包装精良的可复用构件系统。

开发可复用资产的成本比开发普通的应用系统昂贵，开发单位须选择合适的构件作为可复用资产进行开发。能被频繁复用的构件才具备复用的经营性价值，值得投资开发。构件的选择当然也与领域的选择有关。

开发可复用构件应当按照规范化的软件过程进行，否则将生产出一堆不能搭配在一起运作的构件，也就不能有效地支持以后的应用工程。

目前已出现多种实施领域工程的方法，有的方法关注的是如何利用现有的领域、体系结构和系统专业技能，有效地界定出“领域”；有的方法关注如何从领域中挑选实例，如何分析需求和趋向；有的方法关

注如何收集各项特性，如何表述它们，如何对它们进行分组和组群；有的方法的资料，只用很少的篇幅描述其核心技术，而用大量篇幅的文档(诸如指南、手册、角色定义、控制点等)说明如何复用；有的方法则说明他们如何采用面向对象技术；有的方法说明如何将领域分析集成到完整的软件工程生命周期中。

本文的方法是将类似的领域工程的若干步骤，集成到面向复用过程中，而不描述明显的领域工程活动。本文采用面向对象方法，将读者导向到一个期望的应用、体系结构和可复用构件。

体系结构和构件的开发者们需要一套系统的方法来优化安排未来的需求。至于这套方法所包括的诸项活动细节以及如何与面向对象软件工程 DOSE (object-oriented software engineering) 进行集成的细节，因篇幅所限不能尽述。

2. 应用系统工程的变化

对于人们熟知的应用系统工程，每次生产应用系统总是从头开始，没有复用问题，至多利用少量的“积蓄”代码块。

而对于现在复用式的应用系统工程，应用工程师应当检验领域模型，收集和分析客户需求，设法将已有的可复用构件汇集在一起，生产出应用系统。

理想情况下，应用系统由若干可复用构件组成。应用工程师利用可复用资产提供的可变性机制，对可复用构件进行客户化调节，就可组装成应用系统。如果现有的可复用构件还不足以完全满足客户所有的需求，就需要另外编程。此类编程工作通常由应用工程师完成，并集成到应用系统中。在需要另外编程的模块中，有可能被界定为新的可复用资产，由构件开发者按照严格的软件开发规范生产出来，再由应用工程师汇集到应用系统中。

从软件复用出发，领域工程和应用系统工程之间的关系如图 8.45 所示。



图 8.45 领域工程为应用系统工程提供可复用资产

8.5.3 构件技术

1. 应用系统和应用系统族

一个“应用系统”(application system)是软件开发单位向其外部世界提供的软件系统。而可复用资产主要是提供给本单位的应用工程师使用，不一定提供给外部世界。

一个“应用系统族”(application system family)是具有共同特性的一系列应用系统。根据这些共同特性开发出的公用构件，用于支持开发该应用族中的各个应用系统。

下面列举两种不同类型的应用系统族。

- 一套应用系统：这是一组不同的应用系统，它们要配合在一起才能正常工作。例如微软公司的办公软件 MS-Office，它包括文字编辑器 Word、数据库管理系统 Access、图文编辑器 PowerPoint，它们在 Windows 下协同工作，提供一套完整的办公环境。
- 应用系统变体：需要利用同一个应用系统，为不同的用户进行配置、打包、安装到不同的地方。例如，电信开关系统的 Ericsson 的 AXE 族。

有时，可以把若干相对独立的应用系统，处理成一个应用族的若干系统，办法是采用同样的可复用构件作为它们的底层。例如，若干应用系统的底层是类似的窗口行为，则采用微软的基础类。

2. 应用系统与构件

当要开发若干相关的应用系统(一个应用族)时，不主张采用每个应用系统都从头开发的老方法，而主张先按照复用的要求，界定这一组应用系统的共同“特性”(feature)，根据这些共同特性，建立模型(可能有多个模型)，并按照复用的要求，将模型分解成恰当规模和结构的构件，仔细地进行设计、实现、打包；编写文档，形成方便使用的可复用构件。在对可复用构件进行设计时，要特别注意尽量降低可复用构件之间的依赖性。

这批可复用构件将用于支持该应用族的各个应用系统的开发工作。

有许多种类的“构件”(component)，譬如使用案例、分析、设计、实现、接口规格说明、子系统、属性类型，还可以是其他形式的“工作成品”，例如模板、文档、测试案例说明、OCX/Active 构件、基于 CORBA 的构件等等。

顺便解释一下此处提到“工作成品”。它的种类很多，可以是一段代码、文档、一件软件模型(在软件开发单位中可以进行独立管理的软件模型)，可以是类型、类及其附属文档，可以是完整的模型、子系统、测试模型，也可以是模型中的模型元素。有的工作成品是抽象的，面向管理的，如配置文件、工作成品名称与版本清单等。开发单位不仅要认真管理好单位的每件产品(例如应用系统、可复用构件库)，而且应当管理好工作成品，每件工作产品都有其标识及责任人，并有相应的文档，说明它的目的和使用方法。

对于构件，应当按可复用的要求进行设计、实现、打包、编写文档。构件应当是内聚的，并具有相当稳定的公开的接口。有的构件具有广泛的可复用性，可复用到众多种类的应用系统中。有的构件则只在有限的特定范围内被复用。

构件有不同的含义。有人采用大型的定义，即一个构件是相关工作成品的一个集合；Barnes 等人(1991)主张广谱复用的含义，即把所有种类的工作成品(如文档、指南、计划、测试、代码)都看成是可复用构件；MS, UML, OMC 使用“构件”一词指称一个封装的代码模块或大粒度的运行时的模块，本文采用的技术与其兼容。

本文采用基于对象技术的构件。对象技术中的封装、继承等特征，可简化构件的开发工作。但对象的继承机制也存在副作用，它使一个类(或类型)对它的上级类(或类型)有很强的依赖性，这使得构件的维护复杂了，甚至影响到基于可复用构件的应用系统开发工作。

3. 构件系统

单个构件的用处不大，若干个构件联合起来，用处就大了。所以要将相关的构件组织在一起，形成构件系统。实施复用的软件开发单位通常拥有多个构件系统，有的是购置的，有的是自己开发的。

有多种形式不同档次的构件系统。小规模构件系统只有少数几个构件及其文档；有的构件系统是相对独立的许多类的一个集合(实际上是类库)；有的是若干相互关联类的一些框架；有的是 Java 类和 OCXs 的集合；有的是能够用于生成完整的应用系统的较复杂的构件系统。

纯粹类库中诸多类可以是相对独立的，而构件系统的内容一般说来多于一个类库，它包括相互关联的构件，这些构件协同工作可以生产出一群相互关联的对象。应用工程师可以将这些构件组合起来向最终用户提供使用案例，即应用系统。

应用系统和构件系统都是系统产品(而不是工作成品)。它们都可以采用模型和结构的类型定义出来。这两类系统的主要差别在于如何实施工程、如何管理以及如何使用。一般情况下，构件系统只在单位内部使用，不提供给外界，而应用系统提供给外界的客户。与应用系统相比，构件系统具有通用性，可复用，这就要求构件系统的开发过程应当实施更为严格的工程规范。

总之，一个构件系统是能提供一系列可复用特性的系统产品。将这些特性实现成相互依赖相互连接的众多构件，包括众多的类型、软件包、文档。

构件系统中的构件应当是高内聚低耦合的，但构件之间应当有若干种关系，例如继承关系(即一个构件可从其他构件那里继承其功能)；可以发送消息给其他构件；可以与其他构件联合，支持协同工作。

一个好的构件系统使得应用工程师确能又快、又好、又省地开发应用系统。对构件系统中的每个构件，都要精心地进行设计和实现，使得它具有适当的灵活性，能够与其他构件(甚至与其他构件系统)协同工作，向应用工程师提供适当层次的功能。构件系统应当是易于理解和易于使用的。每个构件类型、类以及与其他构件的相互作用，均应当有良好的文档，文档中用的术语应当前后一致。对构件应当是仔细地进行建模、实现、制作文档、测试，便于以后的有效维护和改进。—

一个构件系统可以辅以相关的过程和工具，用于支持构件的复用。为此，需要开发相应的工具箱，把构件、工具、客户化语言的问题描述、应用开发过程都包装在该工具箱内。

4. 构件系统的门面

为便于叙述，下面用“复用者”指使用可复用构件的人们。“复用者”不仅包括应用工程师，也包括构件开发者，因为在开发可复用构件时，经常用到已有的构件。换句话说，软件开发单位内的软件工程师们几乎都是复用者。

所以，构件系统应当为复用者提供简便灵活的使用“门面”(facade)。门面是一种特殊的软件包，它的作用是简化复用者的工作。它是构件系统的用户视图，让复用者了解该系统诸构件的功能与用法，它还可以“输出”(export)可复用构件。

构件系统通过门面向复用者表达和输出可复用构件。这样做既让复用者不用过问构件系统内幕细节，又使构件系统内部的改动不会影响复用者。

一个构件系统，可以根据其复用者的群组和构件系统自身的演变，设置若干门面。门面的内容应当经过精心设计实现，符合适当的体系-结构和工业标准，满足复用者的需求。门面的设计要与体系结构设计师、构件开发者、复用者进行多方协商。

门面可以是通用建模语言 UML 的一种特殊软件包，门面软件包里可以包括一些直接定义的构件，还可以包括从 OOSE 系统内幕输入的一些构件。

构件系统的输出实际上就是被挑选出来的类型、类、关系，以及附属的文档，通过门面提供给复用者。类型和类包括：各种的结构(如角色、使用案例、分析、设计对象和模板)、子系统和服务包、接口、实现

类、属性类型等。

前面提到“输出”，是从构件系统角度说明它如何提供构件。而从复用的角度(譬如应用系统的角度)，则用“输入”(import)说明它如何获得构件。图 8.46 表示应用系统与被复用的构件系统之间的关系，图中的“输入”箭头表示复用。

5.可变性和客户化

为了使构件系统更切合实际、更有效地被复用，构件应当具备“可变性”(variability)，以提高其通用性。针对不同的应用系统，只需对其可变部分进行适当的调节，即进行“专化”(specialize)，对于应用系统来说，就是进行“客户化”工作。

过去，面向对象系统的开发者最初只使用继承性对付可变性，而单纯使用继承性会使一些复杂的系统变得脆弱，易出错，为此不得不限制其灵活性。

为了应付复用时遇到的各种不同情况，构件系统应当提供若干不同的可变性机制。现在已有多种客户化技术，本文只讲述一般性的客户化。

可变性不仅可以提高构件系统的通用性，而且还会显著减少构件系统中的构件数目，因为一个通用性好的构件可以顶替数目众多的相似的非通用构件。

需要进行“客户化”才能真正被复用的构件，又叫做抽象构件，而可以被直接复用的构件则叫做具体构件。要复用一个具体的构件，要做的事只是输入该构件以及它所依赖的所有构件。而抽象构件是通用的，也是不完备的，仅仅输入构件还不够，还需先进行客户化。超类型、超类、带参数的模板等就是典型的抽象构件。

抽象构件一方面向复用者提供了一些公共特性(或职能)，另一方面还提供可变的特性(职能)。复用者要根据复用的具体需要，用合适的“变体”(variant)改造可变特性，这就是客户化工作。构件系统可以预制一些变体让复用者选择，也可提供一种机制让复用者提供“变体”(variant)，也就是让复用者可扩展其特性。

所谓一个特性，可以是一个使用案例、使用案例的一个部分或者使用案例的一个职能。本文采用特性一词，可以比较方便地说明某些实现细节或操作限制，例如，目标操作系统选择为窗口系统、在规模和性能方面的限制等。所以说，一个特性就是构件(或构件系统，或应用系统)的任何一个突出的特征，而且特性可作为选择项，让复用者或客户进行挑选。

如果每个特性都实现成一组关系密切的使用案例和对象类型或类的构件，就会便于使用。出自不同模型而又支持一组公共特性的若干相关构件，被跟踪(trace)链和参与(participate)链接在一起。

6.打包和编写文档

为了方便复用，必须仔细地为构件系统打包，并用该系统的输出构件、变体和门面等术语编写文档。门面应当使复用者明白如何使用各种构件和预制的变体，如何进行客户化，要说明诸构件是如何相互依赖的，在联合使用这些相互依赖的构件时会有哪些限制。

8.5.4 分层式体系结构

1.软件体系结构

所谓“软件体系结构”(software architecture)，是在高层次上定义软件的组织，并处理如何将系统分解为若干单元，这些单元又如何相互作用。良好的体系结构应当容忍变更，可理解，并使系统功能的设计更具适应性。

在以后的叙述中，有时简称软件体系结构为“体系结构”(architecture)。

现代的大型信息系统是十分复杂的，而且受到不断变化的标准、组合分布计算技术、系统平台等各方面的影响。这种复杂性是内在的，而且是无法回避的，但可以设法用良好的软件体系结构来处理。

软件体系结构是一个易于产生幻感的术语。软件工程师们觉得对它已经理解了，却又发现难于给出它的确切定义。此术语背后的概念很重要，它影响着软件的设计和结构，因而影响软件的特征。本节将使用简化的方法，给出此术语的定义。

在一个面向对象的系统中，实现类是组织在子系统中，软件体系结构定义了软件按子系统组织的静态结构(子系统之间通过接口相互连接)，并在一定程度上定义了诸结点(执行那些子系统的诸结点)之间是如何相互作用的。

还有人采用别的方法给出软件体系结构的定义，例如，采用可计算构件和构件间的连接等术语，定义一个软件系统的体系结构。本文将不再展开讨论。

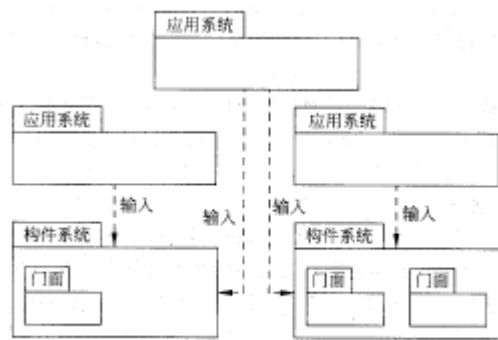


图 8.46 应用系统通过构件系统的门面复用其中的可复用构件

2.良好的软件体系结构的重要作用

应用系统是提供给最终用户使用的；而构件系统中的诸多构件，则往往是提供给本单位的软件工程师复用的。两者的开发过程及其具体规定有所差异，但不论是应用系统还是构件系统，通常均由一个开发团队按照特定的软件开发过程进行开发。

选择合适的体系结构，对于一个软件开发单位来说，是最重要的决策之一。有了良好的体系结构，各个开发团队的软件工程师们则可更有效、更有预见地进行系统的设计和实现工作。工程师们可以在定义好的接口界面上进行工作。在开发应用系统时，良好的体系结构的作用相当于选择构件的指南。如果缺乏清晰定义的体系结构和接口，诸多构件很难协同工作，工程师们就难以复用构件。

为了维护软件系统的完整性，使得开发和维护工作不致于杂乱无章，体系结构是很重要的。良好的软件体系结构还是简化软件系统复杂性的关键，让大规模的开发单位能以并行方式开展工作。

此外，修改需求和新增需求是常出现的事，不但应用系统如此，构件系统也是如此。良好的体系结构使构件系统和应用系统均能有序地随时进行修改。体系结构的定义方式和描述方式，也应当使人们易于对系统进行修改和改进。为了建立允许变更的体系结构，辨清软件的哪些部分是很可能变更的、哪些部分是稳定不变的十分重要。体系结构中最为稳定的部分，应当对软件的子系统和接口组织起着最具影响的作用。同时，体系结构又要预见到可能的变更，与之相应的子系统和接口应当设计成可变更的。这如同盖房屋，有些部分(地基、外墙)不常变动，而有的部分(内墙、内部分割)常变动，还有的部分(每间房内的家具)变动更为频繁。如果把房屋的外墙造得易于更换，而把房内家具粘牢或焊死在地板上，那真是不得要领。

开发单位的规模越大(特别是跨地域分布的情形)，通信联系的开支就越大，因为软件开发者们需要经常协调他们的工作。具备显式接口的良好的体系结构能降低通信开支，因为体系结构为开发者们、提供了需要了解的大部分信息(特别是其他部分能做什么的信息)。

定义软件体系结构的工作比开发一个应用系统或一个构件系统更困难。如果软件开发单位遇到新的领域，或者采用了新的技术，那么制定良好的软件体系结构，对于该单位就是最重要的事情之一。当然，由于面对新领域或新技术，缺乏经验的体系设计师们可能觉得困难，正因为如此，此事更显得重要，因为开发者们在建立新应用系统或新构件系统时，也会因缺乏经验而更需要指导。

采用什么样的体系结构可以满足上述各方面的要求呢？没有简单的答案。凭经验说，采用分层式体系结构并正确运用，才是正确的起步。

3.分层式的体系结构

粗略地说，所谓分层式体系结构(layered architecture)是按层组织软件的一种软件体系结构，其中每层的软件建立在低一层的软件层上。位于同一层上的诸多软件系统或子系统，具有同等一的通用度，低层的软件比高层的软件更具通用性。一个层次可视为同等通用档次的一组(子)系统。

所以，在分层式体系结构中，最高层是应用系统层，可包容诸多应用系统。次高层是构件系统层，可包括多个构件系统，用于建立应用系统。应用系统建立在构件系统层之上，而这个构件层中的诸多构件系统又可建立在更低层次的构件系统之上。

这里所说的“软件组织”，是指软件的静态分层组织，就像在编译连接时软件诸模块之间的分层依赖关系那样，是一种静态的关系，而并非指软件在运行时的组织和动态的结构。一个系统的动态特征是由使用案例、协作、过程和结点模型来定义的，这些动态模型要与软件的静态分层组织联合起来使用。

即使按照上述原则，人们仍可以定义出诸多形式的分层式体系结构，层的数目、层的名称、层的内容均可随情况而定。本文介绍一种较为典型的4层次的分层式体系结构，如图8.47所示。下面将逐层介绍各层的内容。

• **第一层是最顶层或最高层，对于每种软件体系结构来说，最顶层总是应用系统层**，此层应当包括诸多应用系统，每个应用系统向最终用户提供一组使用案例。有的应用系统还可具有不同版本和若干变体。应用系统可以通过其接口直接与其他系统交互操作，还可以通过低层软件提供的一些服务或对象(例如ORBs、操作系统、业务专业化服务)间接地与其他系统交互操作。

• **第二层是次顶层或次高层，它应当是“业务专业化”(business-specific)**，此层应当包括专门针对不同业务类型的一系列构件系统。这些的构件系统提供的使用案例和对象构件，均是可复用的，用于开发应用系统。业务专业化层的构件建立在中件层上。

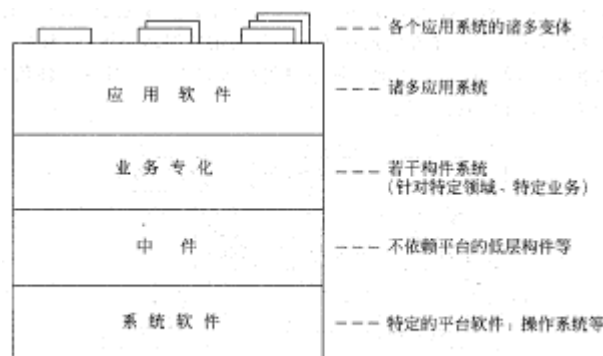


图 8.47 4 层的分层式体系结构

• **第三层是“中件层”(middleware-layer)**, 位于次顶层(业务专业化层)下面, 此层为次顶层的诸构件系统提供实用软件类以及不依赖平台的服务, 例如在异种机型环境下的分布式对象计算等。此层常包括图形用户界面 GUI(graphical user interface)构筑者使用的构件系统, 与数据库管理系统 DBMS 的接口, 不依赖平台的操作系统服务, 对象请求代理 ORBs(object request broker), OLE 构件(object linking and embedding component), 如电子表格(spreadsheet)和框图编辑器。此层的构件是提供给应用系统和构件系统的开发者们使用的, 使得他们能专注于业务构件和应用-系统的构筑。

• **第四层是最低层, 它是系统软件层**, 此层包括计算和网络基础设施的软件, 如操作系统、专用硬件接口软件等。

但是, 目前已出现了一些专用操作系统, 其本身就提供了不依赖平台服务, 致使第三与第四层之间显得模糊不清。一般说来, 在这两层之间很难精确划清。例如可以从两个视角审视 Java: 其一, Java 是一种语言, 故它位于系统软件层, 即第四层, 更有意思的是, 可以把 Java 看成是组织分布对象的一个重要部分, 通过 Java 可将对象移到不同的机器上, 从而改变客户机-服务器系统的应用划分; 其二, Java 的一个重要部分又属于中件层, 至少中件层的许多软件是用 Java 写成的。因此, 在我们的例子中, 已将 Java 安置在中件层。

为了确保分层式系统可管理, 规定在一个系统内不能从低层复用高层的构件。一个分层式系统有两维, 水平方向是在同层次内的相互引用的诸多系统, 而垂直方向表达了跨层的静态的依赖关系, 如图 8.48 所示。图中虚线单向箭头表示垂直方向依赖关系, 实线双向箭头表示水平方向的相互引用。

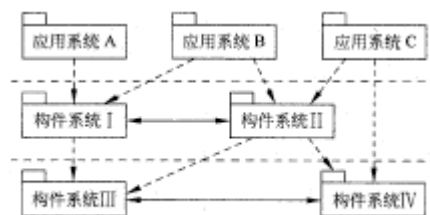


图 8.48 一个分层式系统中的依赖关系和相互引用关系

8.5.5 渐进地实施复用和复用单位的组织结构

系统的复用不可能自发实现。为在开发单位内实施系统的复用, 不仅需要采用相应的新技术, 而且必须在复用的管理和组织方面投入努力。8.5.2.2 节描述的是已经进入软件复用轨道的开发单位应当采用的开发和复用过程。本节将讨论软件开发单位如何从现有的状态进入复用轨道。

8.5.5.1 软件复用需要改变开发单位的组织结构

传统的软件开发单位的组织结构, 通常是一个高层经理下面有若干个应用工程项目经理, 由高层经理分配资源, 包括物力人力资源, 譬如在人力资源方面, 将已经完成的项目中的人员分配到需要进行开发的项目中, 而每个项目经理只负责他所掌管的项目。在这样的开发单位中没有可复用资产方面的资源。

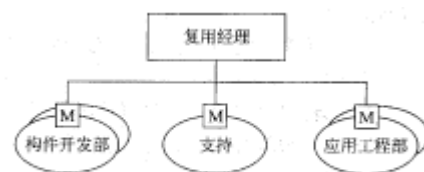


图 8.49 利于实施系统复用的一种软件复用组织结构

实施软件复用的开发单位的组织结构就不同了。这样的开发单位至少有两个基本职能, 而且由两种类型的部门分别承担之, 其一是开发可复用资产, 相应的部门是构件系统开发部; 其二是完成诸多的应用工程项目(即复用), 相应的部门是应用工程部。

具备系统复用经验的开发单位往往还需要第三个职能, 即支持职能, 相应的部门是支持部; 而且在构件开发、应用开发、支持 3 个平行的部门上面还需要一个高层经理, 如图 8.49 所示。高层经理关注的是总目标; 支持部门要对构件开发部所提供的可复用资产进行确认、对构件库进行分类编目、向本单位的工程师们发通告和分发可复用资产、提供必要的文档、从复用者收集反馈信息和缺陷报告。

这样的组织结构是根据以往的经验教训逐步摸索总结出来的。例如, 有的开发单位曾把构件开发者安排在应用工程部, 在项目经理的领导下工作, 由于项目经理的注意力集中于从外面获取的应用项目上, 因此可复用资产的开发目标往往被延误或被遗忘。这种情况在开始时很常见, 应当说这是教训。然而, 如果将构件开发者与应用开发者在职能上完全隔离开, 甚至在地理位置上也分割开, 那么构件开发者就如同在真空中工作, 所生产出来的可复用构件往往不能充分满足应用开发者的实际需要, 或者不及时, 所以说这也是不可取的。

一方面, 构件开发者应当尽量接近应用开发者, 以使其开发出的构件能尽量符合实际需要; 另一方面, 构件开发者与应用开发者应当是两个并列的部门, 使构件开发者能摆脱开应用项目的日常压力, 保证可复用资产的开发和持续改进。构件开发部的职责是生产高质量的可复用资产, 以满足复用者(特别是应用工程师们)在数年内的工作需要; 应用工程部的职责是尽量利用可复用构件, 又快又省地完成应用工程项目的开发任务。

即便采用了构件开发部与应用工程部并列的组织结构, 有时仍然会遇到压力, 例如, 项目经理面临紧迫期限的挑战, 出于过大的压力可能要求停掉构件开发工作, 这就会影响到可复用资产发展的长期目标。出现这种情况, 说明这个单位尚未充分认识到复用的重要地位, 也没有认识到可复用库的巨大工作量和难

度。所以说，实施软件复用的单位需要一个高层经理，即图中位于3个职能部门之上的高层经理。他关注的是总目标，应当在构件开发和应用工程利益之间进行权衡。有的单位称呼这样的高层经理为复用经理。

跨地域的大规模公司有时采用复用管理委员会替代高层经理，委员会包括体系设计师和经理，他们力图在跨地域的各个分单位之间进行复用。各分单位的矛盾提交到委员会上，进行讨论和裁决。不过，这样的委员会的解决矛盾周期可能较长。为此，各个分单位要权衡一下是否要把矛盾提交到委员会上去解决。

8.5.5.2 渐进地系统地采用复用技术

前面重点讨论实施系统复用的开发单位最好采用什么样的组织结构。本节将讨论如何从现有的状态逐步过渡到系统复用状态。

所谓系统地采用复用技术(简称系统复用)，就是全面地采用复用技术。所谓渐进地采用复用技术，通俗说，就是逐步递增复用技术。

一个软件开发单位要想采用复用技术，通常面临两种压力：其一，必须保持现有的传统机制继续运转，它包括获取开发单位维持经费的各项活动；其二，要对现有的传统机制进行变革，使之逐步过渡到全面复用状态。

在线的经理们通常熟知如何运作现有的机制，而不熟悉新的复用机制，更不知如何在保持老机制运转的同时，开创出新的复用机制。所以，从传统机制到复用机制的过渡并非易事，往往需要数年的时间，而且以逐步过渡方式为宜，可以典型示范先行，摸索经验教训，再逐步地扩展，逐步增加复用覆盖面，以至覆盖整个开发单位。

1. 采用系统的复用技术

要采用系统的复用技术，就需要在亚冬人员、过程、组织结构、体系结构、工具、技术等多方面同时进行变革。对于大规模的软件开发单位，尤其需要系统地采用复用。而同时面对这么多方面的变革，如果缺少一种系统的变革方法，很容易被许多枝节问题所困扰，风险较大。

现在已有多种不同的变革方法，适用于大规模的软件开发单位。有的方法注重建立过程和组织结构的模型，有的方法注重处理变革过程中的人员问题，有的方法则注重复用成功因素。本文建议组合这3种方法，形成渐进的系统的过渡方法。

- 用业务工程提供系统过渡的框架。业务工程 BE(business engineering)提供了一种以过程为核心的组织结构和系统设计的视图，它包括对复用业务的设想、建模、逆向工程和正向工程各个步骤。其关键的思想是要确定一系列的功能交叉的过程，这些都是开发单位应当有效地执行的过程，然后进行组织结构的优化，并制定和建立贯穿整个开发单位的政策和信息系统，目的是消除组织结构方面的隔阂。为了实施过渡，我们将采用面向对象业务工程方法。

- 组织结构变革涉及到人员问题。组织结构的变革涉及到人员的安排、人们的思想忧虑、政策、组织方面的压力、知识的缺乏等。进行变革的过程中需要进行细致的工作，包括为人们建立信心、鼓励、支持、领导、过程权限分享等。

- 渐进地实施注重实效的复用进程。渐进地实施复用技术，包括一组重实效的指南、模型、里程碑，还包括复用单位如何逐步地计划安排其复用的演进。目前，复用界(包括复用研究界和实践界)已经得到这样的共识：成功的复用计划应当是通过一系列步骤逐渐成熟的。

2. 一个实例

Hewlett-Packard 公司出于业务经营的需要实施复用。经过多年的实践，总结出渐进的过渡的方法，每前进一步，再进入下一步，增加一些复用的新技术和活动，各步的复用技术包括：

- (1)黑盒代码的复用。
- (2)库和工作成品的管理。
- (3)体系结构和系统。
- (4)应用工程和构件工程技巧。
- (5)面向复用的过程和组织的组织的管理。
- (6)新工具和技术。

他们在充分认识到需要过渡到下一步的复用时，就下决心进行投资和准备，引入更高档次的复用技术。而高档次的复用又需要相应的组织结构来支持，而且需要增加体系结构的层次，过程的实施也更为严格。这是一个成熟的复用模型的实例，图 8. 50 展示了该公司是如何渐进采用复用技术的，其中每个步骤均为实际的业务需要所驱动。

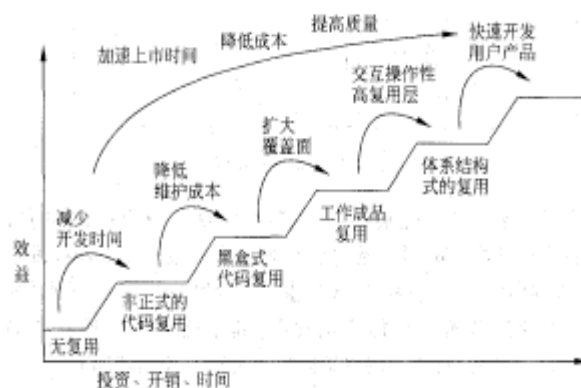


图 8.50 渐进实施系统复用技术的一个实例(HP 公司)

多数的复用实施方法是从小规模典型示范开始，在证实其成功之后，再在本开发单位中逐步扩大范围进行推广，并逐步提高复用档次。我们将称之为“典型示范驱动式渐进过渡方法”。一般说来，采用此法见效较快，也可以较快发现问题，风险小，开始的投资也较小。

实施渐进过渡要注意稳步前进，一旦掌握了新技巧，就要及时采取措施加以巩固，使其成为持久的制度化的实践，本步内掌握的技术要多经历考验，再进入下一步。随着经验的增长，还可以跨单位扩大复用的范围，并提高复用的档次。

3. 渐进地采用复用技术

如果将 8.5. 5. 1 节所说的 3 个方面技术中的关键部分组合到面向对象业务工程的框架中，便可得到如图 8. 51 所示的高层视角的复用过渡过程。与图 8. 50 相比，此图所示的方法更激进，它允许一些步骤并行地进行。下面将逐一说明渐进过渡中各个步骤的工作内容和涉及到的有关人员。

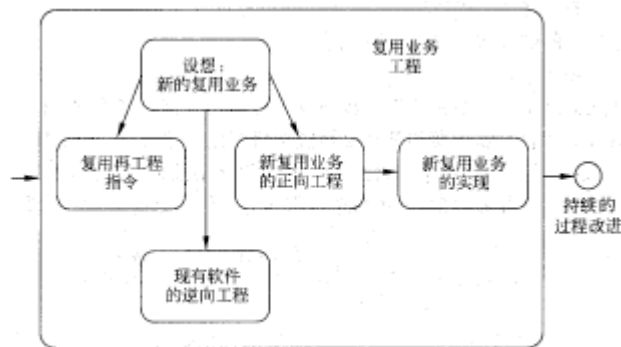


图 8.51 渐进地过渡

为便于叙述，我们给每个步骤赋予一个标签名 TRA1、…、TRA6，并在此先交代一下过渡中涉及到的 3 层领导人员：其一是过程的业主，负责发布指令和提供足够的资金，负责发布第一步 TRA1 中的再工程指令，还要参与以后的各步，以利控制进度，特别要参与第二步 TRA2，以利理解所设想的复用业务；其二是软件业务 SEB(software engineering business) 经理，负责复用驱动式软件业务 RSEB(reuse-driven software engineering business) 的队伍组织，并向复用业务业主报告 SEB 经理应当确保体系结构、相应的组织结构安排，并负责应用系统和构件系统计划；其三是过渡团队负责人，具体领导过渡团队进行过渡工作，逐步建立复用业务、制定计划、实施计划。

• TRA1：发布软件业务再工程的指令。过程的业主(即软件开发单位的管理层)制定并发布再工程的指令，郑重申明开发单位高层复用业务的目标及其主要理由。指令中明确规定了初始的业务、过程、体系、组织结构、复用目标、变革的范围，并设立过渡团队，确定其各方面的责任，而且管理层还授权给过渡团队仔细设想复用业务。

• TRA2：设想新的复用业务。基于业务需要和初始应用族工程的工作，SEB 经理领导 RSEB 过渡团队制定出高层的新的体系结构、软件业务过程、组织结构的设想，写成文档，并向过程业主报告。计划中要设置中间过渡点，要重视过渡中的通信问题。

关于业务过程再工程 BPR(business process reengineering)的指南，包括使用典型示范和管理领导的重要性，可在 Hammer 和 Stanton(1995)的手册中查到。

• TRA3 逆向工程。过渡团队要对现有的体系结构、软件资产、软件过程、组织结构、工具、基线测量等，进行认真的调查研究，其目的是理解现有的软件过程实践、弄清现有资产和现有的复用状态、理解现有组织结构方面的问题。

• TRA4 新的复用业务的正向工程。制定出新的复用业务所需的软件工程过程 SEP(software engineering process)、组织结构、软件过程环境和工具。

• TRA5 实现新的复用业务。建立新的模型，进行人员培训，使开发过程、组织结构、体系结构、系统各方面逐步地实现以新代旧。

• TRA6：持续地改进过程。随着新的业务模型开始运作，不断地收集并分析复用过程和产品评估，以测定过程、界定关键问题，以利改进，逐步实现变革。此步实际包含从 TRA1 到 TRA5 的反复迭代过程，下面将专门讨论此迭代过程。

• TRA6：持续地改进过程。随着新的业务模型开始运作，不断地收集并分析复用过程和产品评估，以测定过程、界定关键问题，以利改进，逐步实现变革。此步实际包含从 TRA1 到 TRA5 的反复迭代过程，下面将专门讨论此迭代过程。

4. 迭代式过渡

渐进复用的实践经验是，需要迭代数轮，才能达到较为满意状态。有两方面的原因：第一，体系结构应当是稳步变革，一般需要 2-3 轮迭代；其次，组织结构的变革不能靠一次大跳跃就完成，而应当逐步地边学习边变革。体系结构的稳步变革，迭代过渡，可通过体系结构的接口界面得到控制，还可让越来越多的人逐步地参与过渡。

从 TRA1 到 TRA5 的每一轮迭代，大约需要 3-12 个月，每一轮迭代，都有明确的目标。通常，第一轮迭代是要得到分层式体系结构的大图，开始认识复用规划，同时还要关注重要的软件工程过程。一般说来，最好先从一支核心团队开始，然后逐步扩大队伍和范围。具体扩到多大，要根据团队的经验以及现有的咨询力量而定。随着经验的积累，应当对其他的过程开展更为仔细的研究，而且应当有更多人员参与推广成功的经验。下面通过一个例子说明，此例包括 4 轮迭代。

(1)第一轮迭代的目的是，初步理解应用族，重点关注体系结构，开发单位开始认识到复用。此外还要完成下列事项：

- 复用业务的高层设想(即概要式的设想)。
- 初步的市场分析。
- 目标组织结构的业务模型，其重点在于理解应当开发出什么样的应用系统和什么样的构件系统，作为第一批产品推出。
- 约定关键的客户。
- 建立过渡团队，过渡团队要启动应用族工程的过程并且要物色合适的人员，参与过渡工作。

(2)第二轮迭代的目的是建立体系结构，尤其要设计好接口界面和门面，并让更多的团队逐步参与新的组织结构。本轮迭代的工作还包括：复用业务的进一步设想(即仔细的设想)，进一步的市场分析，与关键客户签订合同，建立合适的团队，开始实施应用族工程。

构件工程师们更多的时间是开发构件系统，此项工作是与需要使用该构件系统的应用系统工程 ASE(application system engineering)过程并行地进行，前提是 ASE 过程较少，因此有可能并行地进行。这种情况在头几轮迭代经常出现，那时体系结构尚未稳定，参与复用过渡的人员也少。然而，随着复用队伍人数增多，依赖构件系统的应用系统数目增多，构件系统就应当先行开发, ASE 过程应当使用已有的(即使是不尽人意的)构件系统，对构件系统的改进则放在下一版本。

(3)第三轮迭代的目的是，组织更多的人员参与构件系统工程，开发并改进构件系统。ASE 过程则源于客户要求，并建立适当的 ASE 专业团队。]

(4)第四轮迭代的目的是，建成稳定状态的复用组织结构，并掌握若干源于客户合同作为考验新的构件系统和(前一轮迭代开发的)构件系统新版的 ASE 过程。建立一个独立的构件支持团队(在第二轮中，构件的开发者是应用系统开发团队中的兼职者)。

经过这 4 轮的迭代，以后的迭代便可将注意力集中在可复用资产的积累上，可扩大其广度和深度。例如：

表 8.14 过渡计划

内 容	第一轮迭代	第二轮迭代	第三轮迭代	第四轮迭代
业务需求和机遇	发布再工程指令	产品计划	客户订单	最终用户反馈
应用族和体系结构	体系结构粗框	体系结构基线	构件系统	应用系统
工作团队	体系结构	构件工程师	应用工程师	构件支持者
采用的过程	AFE	CSE	AFE	客户 ASE&CSE 过程

- 扩充或改进体系结构，使其覆盖更多的应用族。
- 增加构件系统的数目或种类。

- 支持持续递增的构件工程或应用工程项目。
- 支持多版本的构件系统和应用系统。
- 进一步扩大组织，并进一步改进过程。
- 组织跨地域分布的团队。

5.过渡计划实例

过渡计划应当是一个既具长期(数年)目标又有阶段里程碑的计划。它应当定义下面的内容：

- 体系结构如何演变。
- 要开发什么应用系统和构件系统。
- 要定义哪些过程。
- 要建立和培训哪些团队。
- 如何安排复用的认识和实施进度。

表 8.14 所示的是一个过渡计划的实例。

表中的 AFE, ASE, CSE 分别是应用族工程 (Application Family Engineering), 应用系统工程 (Application System Engineering), 构件系统工程 (Component System Engineering)缩写。

8.5.5.3 充分利用可共享复用成果

有若干复用计划、复用过程和技术的发展，得到了政府、企业、国际财团的资助。已经制定出各种的复用指导原则、过程、认证复用的度量方法，还开发成一系列复用工具。这些成果都是可共享的成果，应当充分利用。

REBOOT 国际财团 (Reuse Based on Object-Oriented Techniques)国际财团于 1990 年由西欧的 9 个公司发起。他们基于面向对象技术的复用，编写了一部优秀的书(Software Reuse)，开发了支持复用的两种过程模型：为复用开发和利用复用进行开发，还开发了一系列工具，称为 REBOOT 环境。

他们强调的一个原则是：未来复用者的需求，就是对可复用构件的信心。开发者的倾向是抵制复用，因为他们缺乏这种信心。为克服这种状态，REBOOT 推荐一种文档结构，它包括测试信息和复用者的经验。

他们还强调：复用并不意味着仅仅复用其代码。构件也可以是分析阶段和设计阶段的成果，代码类的构件也不仅仅是些诸如“类型”、“类”这样“小的”和“简单的”构件，可以大到是一整个系统，例如人员管理系统。

STARS 是美国国防部的一项长期的项目，它是可适应、可靠软件的软件技术的缩写 (Software Technology for Adaptable, Reliable Software)。它关注过程、体系结构、复用三者的集成。它认为软件生产线开发的软件周期应当包括过程驱动、软件体系结构、领域工程、可复用构件库这 4 个概念。

STARS 和美国国防部资助了复用技术的开发，有的是直接资助，有的是合同资助，如 Boeing, IBM, Loral, Unisys 等公司。技术包括复用过程的概念框架 CERP, 若干有组织的领域工程方法 ODM, 面向特征的领域分析 FODA, 可复用的国防软件广泛方法的若干手册和指南 CARDS、领域特化软件的体系结构 DSSA 以及复用库互操作的一个模型。

此外，还有 Boeing, IBM, Motorola 等公司也都提供一些可共享的复用技术成果。

8.5.5.4 实施系统复用需要遵循的原则

根据已经采用复用技术的许多开发单位的共同经验，如果要系统地实施软件复用，就需要遵循下述 10 条原则：

- 需要顶层管理领导，并需要有长期回收的经费支持。
- 为了渐进地推行系统的复用，需要规划和调节系统的体系结构、开发过程、组织结构，并以小规模的首行项目为典型示范，而后再铺开。
- 为了复用，先规划体系结构及其逐步实施的过程。
- 过渡到明确的复用组织机构，将可复用构件的创建工作与复用工作(即利用可复用构件开发应用系统的工作)分离开，并且提供明确的支持职能。
- 在真实的环境中，进行可复用构件的创建和改进工作。
- 要将应用系统和可重用构件作为一个经济核算的产品整体进行管理，应当注重公用构件在应用系统及其子系统领域中的高盈利作用。
- 要认识到单独的对象技术或者单独的构件技术都是不够的。
- 采用竞赛和更换负责人的办法，进行开发单位的文化建设和演变。
- 对基础设施、复用教育、技巧培训，要投资和持续地改进。
- 要采用度量方法测量复用过程，并要优化复用程序。

第 9 章数据库与数据仓库

传统的数据库技术是以单一的数据资源，即数据库为中心，进行从事务处理、批处理，到辅助决策分析等各种类型的数据处理工作。随着计算机技术的飞速发展和企业界不断提出的利用数据库中积累的大量数据进行分析、决策的新的需求，数据仓库技术应运而生。

本章结合信息系统的开发，介绍关系数据库系统和数据库系统设计，并简单介绍数据仓库、联机分析处理和数据挖掘技术，它们是系统分析员的必备知识。

9.1 关系数据库系统

9.1.1 关系数据库系统概述

关系数据库系统是支持关系模型的数据库系统。

30 多年来，关系数据库系统的研究取得了辉煌的成就。关系方法从实验室走向了社会，涌现出许多性能良好的商品化关系数据库管理系统(RDBMS)，如 DB2, Oracle, Ingres, Sybase, Informix 等。数据库的应用领域迅速扩大。

9.1.1.1 关系数据模型

关系数据模型由关系数据结构、关系操作集合和关系完整性约束 3 大要素组成。

1. 关系数据结构

关系模型的数据结构单一。在关系模型中，现实世界的实体以及实体间的各种联系均用关系来表示。在用户看来，关系模型中数据的逻辑结构是一张二维表。

2. 关系操作集合

关系模型中常用的关系操作包括选择(select), 投影(project), 连接(join), (divide), 并(union), 交(intersection), 差(difference)等，以及查询(query)操作和增(insert)、删(delete), 更新(update)

操作两大部分。查询的表达能力是其中最主要的部分。

关系操作的特点是集合操作方式，即操作的对象和结果都是集合。这种操作方式也称为一次一个集合(set-at-a-time)的方式。相应地，非关系数据模型的数据操作方式则为一次一个记录(record-at-a-time)的方式。

关系模型给出了关系操作的能力和特点，关系操作通过关系语言实现。关系语言的特点(它的优点)是高度非过程化。

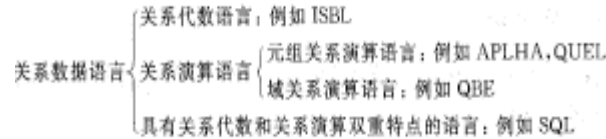


图 9.1 关系数据语言的分类

所谓非过程化是指：

- 用户不必请求 DBA 为他建立特殊的存取路径，存取路径的选择由 DBMS 的优化机制来完成。
- 用户也不必求助于循环、递归来完成数据的重复操作。

关系操作能力可用两种方式来表示：代数方式和逻辑方式。关系代数是使用对关系的运算来表达查询要求的方式。关系演算是用谓词来表达查询要求的方式。几关系演算又可按谓词变元的基本对象是元组变量还是域变量分为元组关系演算和域关系演算。关系代数、元组关系演算和域关系演算 3 种语言在表达能力上是完全等价的。因此，本书只对关系代数进行阐述。

关系代数、元组关系演算和域关系演算均是抽象的查询语言，这些抽象的语言与具体的 DBMS 中实现的实际语言并不完全一样。但它们能用作评估实际系统中查询语言能力的标准或基础。实际的查询语言除了提供关系代数或关系演算的功能外，还提供了许多附加功能，例如集函数、关系赋值、算术运算等。

另外还有一种介于关系代数和关系演算之间的语言 SQL(Structured Query Language)。SQL 不仅具有丰富的查询功能，而且具有数据定义和数据控制功能，是集查询、DDL, DML 和 DCL 于一体的关系数据语言。它充分体现了关系数据语言的特点和优点，是关系数据库的标准语言。

因此，关系数据语言可以分为如图 9. 1 所示 3 类。它们的共同特点是：语言具有完备的表达能力，是非过程化的集合操作语言，功能强，能够嵌入高级语言中使用。

3.关系的完整性约束

数据库的数据完整性是指数据库中数据的正确性和相容性。那是一种语义概念，包括两个方面：

- 与现实世界的应用需求的数据的相容性和正确性。
- 数据库内数据之间的相容性和正确性。

例如，学生的学号必须惟一，性别只能是“男”或“女”，学生所选修的课程必须是已开设的课程，等等。可见，数据库中数据是否具备完整性关系到数据库系统能否真实地反映现实世界，因此数据库数据的完整性是十分重要的。

数据完整性由完整性规则来定义，关系模型的完整性规则是对关系的某种约束条件。关系模型中可以有 3 类完整性约束：实体完整性、参照完整性和用户定义的完整性。其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件，应该由关系数据库管理系统(DBMS)自动支持；而用户定义的完整性是应用领域需要遵循的约束条件，体现了具体领域中的语义约束，一般由关系数据库管理系统或工具提供编写手段，由 DBMS 的完整性检查机制负责检查。

9.1.1.2 关系模型的数据结构和基本术语

在关系数据模型(Relation Model)中，数据结构用单一的二维表结构来表示实体及实体间的联系，如图 9. 2 所示。

(1)关系(relation)：一个关系对应一个二维表，二维表的名称就是关系的名称。

例如，图 9. 2 包含两个表，也即两个关系：学生登记表关系和系信息表关系。

(2)属性(attribute)和值域(domain)：在二维表中的列(字段)称为属性。属性的个数称为关系的元数。列的值称为属性值；属性值的取值范围称为值域。

例如，图 9. 2 中学生登记表关系的属性有学号、姓名、性别、年龄、系号、原单位共 6 个属性，所以元数是 6。年龄属性的值域是大于等于 15 岁，小于等于 40 岁。系信息表关系的属性有系号、系名、办公室、主任、电话共 5 个属性，所以元数是 5。

关系名 → 学生登记表
属性(列)和属性名

学号	姓名	性别	年龄	系号	原单位
010101	张 力	女	22	01	数学所
010302	林宏业	男	23	02	物理所
011008	王 萌	男	24	04	化学所
⋮	⋮	⋮	⋮	⋮	⋮
990129	陈婷婷	女	23	01	计算所
990116	李一鸣	男	30	16	首钢

关系模式 (关系定义)

元组集合

↑

主码

↑

$15 \leq \text{年龄} \leq 40$

← 年龄属性的值域

系信息表				
系号	系名	办公室	主任	电话
01	计算机	教 209	张 立	301
02	物理	教 501	李记欣	276
03	数学	教 410	王鸣利	346
04	化学	教 306	高 明	417
⋮	⋮	⋮	⋮	⋮
16	外语	教 701	陈 钢	628

图 9.2 关系数据模型的数据结构示例

(3) **关系模式(relation schema)**: 在二维表中的行定义(记录的型), 即对关系的描述称为关系模式。一般表示为: 关系名(属性 1, 属性 2, ..., 属性 n)

例如, 图 9. 2 中有两个关系模式, 分别表示为:

学生登记表(学号, 姓名, 性别, 年龄, 系号, 原单位)

系信息表(系号, 系名, 办公室, 主任, 电话)

(4) **元组(tuple)**: 在二维表中的一行(记录的值), 称为一个元组。关系模式和元组的集合通称为关系。例如, 在学生登记表关系中的元组有(010101, 张力, 女, 22, 01; 数学所), (010302, 林宏业, 男, 23, 02, 物理所), (011008, 王朝, 男, 24, 04, 化学所)等。

(5) **分量(component)**: 元组中的一个属性值。

例如, 在学生登记表关系中元组(010101, 张力, 女, 22, 01, 数学所)的每一个属性值 010101, 张力, 女, 22, 01, 数学所, 都是它的分量。

(6) **候选码(candidate key)或候选键**: 如果在一个关系中, 存在多个属性(或属性组合)都能用来惟一标识该关系的元组, 这些属性(或属性组合)都称为该关系的候选码或候选键。

例如, 在学生登记表关系中, 如果姓名不允许重名时, 学号和姓名都是候选码。

(7) **主码(primary key)或主键**: 在一个关系的若干个候选码中指定一个用来惟一标识该关系的元组, 这个被指定的候选码称为该关系的主码或主键。

例如, 在学生登记表关系中, 学号一般都是惟一的, 如果姓名不允许重名时, 存在两个候选码: 学号和姓名, 若选中学号作为惟一标识, 那么, 学号就是学生登记表关系的主码或主键。

(8) **主属性(primary attribute)和非主属性(nonprimary attribute)**: 关系中包含在任何一个候选码中的属性称为主属性或码属性, 不包含在任何一个候选码中的属性称为非主属性或非码属性。

例如, 在学生登记表关系中, 如果姓名不允许重名时, 学号和姓名是主属性, 其他属性是非主属性。

(9) **外码(foreign key)或外键**: 当关系中的某个属性(或属性组)虽然不是该关系的主码或只是主码的一部分, 但却是另一个关系的主码时, 称该属性(或属性组)为这个关系的外码。例如, 如果图 9. 2 中的系信息表关系的主码是系号, 那么, 在学生登记表关系中的系号就是外码, 因为它是另一个关系系信息表的主码。

(10) **参照关系(referencing relation)与被参照关系(referenced relation)**: 参照关系也称从关系, 被参照关系也称主关系, 它们是指以外码相关联的两个关系。以外码作为主码的关系称为被参照关系; 外码所在的关系称为参照关系。由此可见, 被参照关系与参照关系是通过外码相联系的, 这种联系通常是 1: n 的联系。例如, 图 9. 2 中的系信息表关系是被参照关系, 而学生登记表关系是参照关系。它们通过外码“系号”相联系。

(11) **关系的形式定义**: 从数学的观点定义关系称为关系的形式定义。有两种定义方法。

- **用集合论的观点定义关系**: 关系是一个元数为 K 的元组集合, 即这个关系有若干个元组, 每个元组有 K 个属性值(把关系看成一个集合, 集合中的元素是元组)。

- **用值域的概念来定义关系**: 关系是属性值域笛卡儿积的一个子集。设一个关系的属性是 A_1, \dots, A_n , 其对应的值域为 D_1, \dots, D_n (也可以有相同的), 定义 D_1, \dots, D_n 的笛卡儿积 $D = D_1 \times \dots \times D_n = \{ (d_1, \dots, d_n) \mid d_i \in D_i, 1 \leq i \leq n \}$ 。D 中的每一个子集 D' 称为关系。这里 D 的元素 (d_1, \dots, d_n) 就是一个 n 元元组 (n-tuple), 元素中的每一个值 d_i 称为元组的一个分量。

若 D_i ($i=1, 2, \dots, n$) 为有限集, 其基数(cardinal number)为 m_i ($i=1, 2, \dots, n$), 则 $D_1 \times D_2 \times \dots$

$\times D_n$ 的基数 M 为:
$$M = \prod_{i=1}^n m_i$$

笛卡儿积可表示为一个二维表。表中的每行对应一个元组, 表中的每列对应一个域。例如, 我们给出 3 个域:

D_1 =教师集合={李鸣, 王立刚}

D_2 =课程集合={数据库技术, 信息系统}

D_3 =学生集合={陈列, 刘红, 张明婉}

则 $D_1 \times D_2 \times D_3$ 的笛卡儿积为:

$D_1 \times D_2 \times D_3 = \{ (李鸣, 数据库技术, 陈列), (王立刚, 数据库技术, 陈列),$
 $(李鸣, 数据库技术, 张明婉), (王立刚, 数据库技术, 张明婉),$
 $(李鸣, 数据库技术, 刘红), (王立刚, 数据库技术, 刘红),$
 $(李鸣, 信息系统, 刘红), (王立刚, 信息系统, 刘红),$
 $(李鸣, 信息系统, 陈列), (王立刚, 信息系统, 陈列),$
 $(李鸣, 信息系统, 张明婉), (王立刚, 信息系统, 张明婉) \}$

其中(李鸣, 数据库技术, 陈列), (王立刚, 信息系统, 张明婉)等等都是元组。李鸣、数据库技术、张明婉、陈列等等都是分量。

该笛卡儿积的基数为 $2 \times 2 \times 3 = 12$, 这也就是说, $D_1 \times D_2 \times D_3$ 一共有 $2 \times 2 \times 3 = 12$ 个元组。这 12 个元组可列成一张如表 9.1 所示的二维表。

表 9.1 D_1, D_2, D_3 的笛卡儿积

应当注意:·

· 元组不是 d_i 的集合, 元组的分量是按序排列的, 而集合中的元素是不排序的。例如, 在关系中有 $(a, b, c) \neq (b, a, c) \neq (c, b, a)$, 但在集合中 $\{a, b, c\} = \{b, a, c\} = \{c, b, a\}$ 。

教师	课程	学生	教师	课程	学生
李 鸣	数据库技术	陈 列	王立刚	数据库技术	陈 列
李 鸣	数据库技术	张明婉	王立刚	数据库技术	张明婉
李 鸣	数据库技术	刘 红	王立刚	数据库技术	刘 红
李 鸣	信息系统	陈 列	王立刚	信息系统	陈 列
李 鸣	信息系统	张明婉	王立刚	信息系统	张明婉
李 鸣	信息系统	刘 红	王立刚	信息系统	刘 红

· 无限关系和有限关系: 若一个关系的元组个数是无限的, 则称这个关系为无限关系; 否则称为有限关系。关系数据库系统考虑的是有限关系。

· 关系数据库也存在型和值之分, 关系数据库的型也称为关系数据库模式, 是对关系数据库的描述, 它包括若干域的定义以及在这些域上定义的若干关系模式。关系数据库的值是这些关系模式在某一时刻对应的关系的集合, 通常就为关系数据库。

(12)数据库对关系的限定: 关系模型的数据结构表示为二维表, 但不是任意的一个二维表都能表示一个关系。关系数据库对关系的限定有:

- **每一个属性是不可分解的。**这是关系数据库对关系的最基本的一条限定, 要求关系的每一个分量必须是一个不可分的数据项, 也就是说, 不允许表中还有表, 例如图 9.3 中的两个关系就不符合要求。因为, 在学生基本情况表关系中, 属性成绩被分为英语、数学、数据库等多项, 这相当于大表中还有一张小表(关于成绩的表)。在职工工资表关系中, 属性工资和扣除都是可以再分解的。
- **每一个关系模式中属性的数据类型以及属性的个数是固定的, 并且每个属性必须命名, 在同一个关系模式中, 属性名必须是不同的。**
- **每一个关系仅仅有一种记录类型, 即一种关系模式。**
- **在关系中元组的顺序(即行序)是无关紧要的。**
- **在关系中属性的顺序可任意交换, 交换时应连同属性名一起交换才行。**
- **同一个关系中不允许出现完全相同的元组。**

9. 1. 2 关系模型的完整性约束

数据完整性由完整性规则来定义, 关系模型的完整性规则是对关系的某种约束条件。关系模型中可以有 3 类完整性约束: 实体完整性、参照完整性和用户定义的完整性。

为了维护数据库中数据的完整性, 在对关系数据库执行插入、删除和修改操作时, 必须遵循这 3 类完整性规则。

为了便于理解关系模型的 3 类完整性规则, 考虑一个实例。

[例 1) 在学生选课管理数据库中有如下 4 个关系:

- 学生(学号, 姓名, 性别, 专业号, 年龄), 主码为学号
- 课程(课程号, 课程名, 学分), 主码为课程号
- 选修(学号, 课程号, 成绩), 主码为学号, 课程号

学生基本情况表

学号	姓名	性别	年龄	系别	籍贯	年级	成 绩		
							英语	...	数学
97001	刘红	女	18	管理	江苏	97	83	...	78
...
97500	陈列	男	19	计算机科学	北京	97	80	...	88

职工工资表

职工号	姓名	职称	工 资			扣 除		实发
			基本	补助	职务	房租	水电	
86051	李 鸣	讲师	805	120	50	60	12	903
86052	王立刚	副教授	1000	150	90	100	125	1015
...

图 9.3 不满足关系数据库限定的关系

专业(专业号, 专业名), 主码为专业号

1. 实体完整性规则(entity integrity rule)

若属性 A 是关系 R 的主属性, 则属性 A 不能取空值。实体完整性规则是对关系中的主键属性值的约束, 即: 关系中的元组在组成主键的属性上不能有空值。实体完整性规则规定关系的所有主属性都不能取空值, 而不仅是主码整体不能取空值。例 1 中的关系“学生(学号, 姓名, 性别, 专业号, 年龄)”中, 主码为“学号”, 则“学号”不能取空值。在关系“选修(学号, 课程号, 成绩)”中, “学号、课程号”为主码, 则“学号”和“课程号”两个属性都不能取空值。

对于实体完整性规则说明如下:

• **实体完整性规则是针对关系而言的。**一个关系通常对应现实世界的一个实体集。例如学生关系对应于现实世界中学生的集合。

• **现实世界中的实体是可区分的, 即它们具有某种惟一性标识。**

• **相应地, 关系模型中以主码作为惟一性标识。**

• **主码中的属性即主属性不能取空值。**所谓空值就是“不知道”或“无意义”的值。如果主属性取空值, 就说明存在某个不可标识的实体, 即存在不可区分的实体, 这与第 2 点相矛盾, 因此这个规则称为实体完整性规则。

2. 参照完整性规则(reference integrity rule)

若属性(或属性组)F 是关系 R 的外码, 它与关系 S 的主码 K_s相对应(关系 R 和 S 不一定是不同的关系), 则对于 R 中每个元组在 F 上的值必须为:

• 或者取空值(F 的每个属性值均为空值)。

• 或者等于 S 中某个元组的主码值。

参照完整性规则就是定义外码与主码之间的参照约束, 即外码的值不允许参照不存在的相应表的主码的值, 或者外码为空值。

现实世界中的实体之间往往存在某种联系, 在关系模型中实体及实体间的联系都是用关系来描述的。这样就自然存在着关系与关系之间的参照(引用)。先来看 3 个例子。

[例 2] 学生实体和专业实体可以用例 1 中的学生关系和专业关系表示, 其中主码用下划线标识:

学生(学号, 姓名, 性别, 专业号, 年龄)

专业(专业号, 专业名)

这两个关系之间存在着属性的引用, 即学生关系引用了专业关系的主码“专业号”。显然, 学生关系的“专业号”属性与专业关系的主码“专业号”相对应, 因此“专业号”属性是学生关系的外码, 它的值必须是确实存在的专业号, 即专业关系中有该专业的记录。这里专业关系是被参照关系, 学生关系为参照关系, 这也就是说, 学生关系中的外码属性的取值需要参照专业关系的属性取值。如图 9. 4(a)所示。

[例 3] 学生与课程之间存在多对多的联系, 因为一个学生可以选修多门课程, 一门课程也可以被多个学生选修。对这种情况, 可以用例 1 中的如下 3 个关系表示, 其中主码用下划线标识:

学生(学号, 姓名, 性别, 专业号, 年龄)

课程(课程号, 课程名, 学分)

选修(学号, 课程号, 成绩)

这 3 个关系之间也存在着属性的引用, 即选修关系引用了学生关系的主码“学号”和课程关系的主码“课程号”。显然, 选修关系的“学号”属性与学生关系的主码“学号”相对应, “课程号”属性与课程关系的主码“课程号”相对应, 因此“学号”和“课程号”属性都是选修关系的外码。这里学生和课程关系均为被参照关系, 选修关系为参照关系, 如图 9. 4(b)所示。同样, 选修关系中的“学号”值必须是确实存在的学生的学号, 学生关系中有该学生的记录, 选修关系中“课程号”值也必须是确实存在的课程的课程号, 即课程关系中有该课程的记录。

不仅不同的关系之间可以存在引用关系, 同一关系内部属性间也可能存在引用关系。

[例 4] 如果在例 1 中, 学生关系增加一个属性“班长”, 原学生关系改为:

学生 1(学号, 姓名, 性别, 专业号, 年龄, 班长)

其中, “学号”属性是主码, “班长”属性表示该学生所在班级的班长的学号, 它引用了本关系“学号”属性, 即“班长”必须是确实存在的学生的学号。

在例 4 中, “班长”属性与本关系的主码“学号”属性相对应, 因此“班长”是外码。这里学生 1 关系是参照关系也是被参照关系。

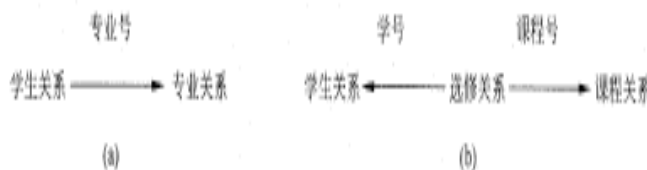


图 9.4 关系的参照图

另外, 需要指出的是, 外码并不一定要与相应的主码同名。不过, 在实际应用中, 为了便于识别, 当外码与相应的主码属于不同关系时, 往往给它们以相同的名字。

例如, 对于例 2, 学生关系中每个元组的“专业号”属性只能取下面两类值:

- 空值, 表示尚未给该学生分配专业。
- 非空值, 这时该值必须是专业关系中某个元组的“专业号”值, 表示该学生不可能分配到一个不存在的专业中。即被参照关系“专业”中一定存在一个元组, 它的主码值等于该参照关系“学生”中的外码值。

对于例 3, 按照参照完整性规则, “学号”和“课程号”属性也可以取两类值, 空值或目标关系中已经存在的值。但由于“学号”和“课程号”是选修关系中的主属性, 按照实体完整性规则, 它们均不能取空值, 所以选修关系中的“学号”, 和“课程号”属性实际上只能取相应被参照关系中已经存在的主码值。

参照完整性规则中, R 与 S 可以是同一个关系, 例如对于例 4, 按照参照完整性规则, “班长”属性值可以取两类值:

- 空值, 表示该学生所在班级尚未选班长。
- 非空值, 这时该值必须是本关系中某个元组的“学号”值。

3. 用户定义的完整性规则

用户定义的完整性规则是用户根据具体应用的语义要求, 利用 DBMS 提供的定义和检验这类完整性规则的机制, 由用户自己来定义的完整性规则。

关系数据库系统根据现实世界中其应用环境的不同, 往往还需、要一些另外的约束条件, 用户定义的完整性就是针对某一具体应用要求来定义的约束条件。它反映某一具体应用所涉及的数据必须满足的语义要求。例如, 某个属性必须取惟一值、某些属性值之间应满足一定的函数关系、某个属性的取值范围在 0-200 之间等。关系模型应提供定义和检验这类完整性的机制, 以便系统用统一的方法处理它们, 而不要由应用程序承担这一功能。

所以, 用户定义的完整性规则通常是定义对关系中除主码与外码属性之外的其他属性取值的约束, 即对其他属性的值域的约束。

对属性的值域的约束也称为域完整性规则(domain integrity rule), 是指对关系中属性取值的正确性限制, 包括数据类型、精度、取值范围、是否允许空值等。取值范围又可分为静态定义和动态定义两种: 静态定义取值范围是指属性的值域范围是固定的, 可从定义值的集合中提取特定值; 动态定义取值范围是指属性的值域范围依赖于另一个或多个其他属性的值。

为了维护数据库中数据的完整性, 在对关系数据库执行插入、删除和修改(更新)操作时, 就要检查是否满足上述 3 类完整性规则。

• **当执行插入操作时:** 首先检查实体完整性规则, 插入行在主码属性上的值是否已经存在, 若不存在, 可以执行插入操作; 否则不可以执行插入操作。再检查参照完整性规则, 如果是向被参照关系插入, 不需要考虑参照完整性规则; 如果是向参照关系插入, 插入行在外码属性上的值是否已经在相应被参照关系的主码属性值中存在, 若存在, 可以执行插入操作; 否则不可以执行插入操作, 或将插入行在外码属性上的值改为空值后再执行插入操作(假定该外码允许取空值)。最后检查用户定义完整性规则, 检查要被插入的关系中是否定义了用户定义完整性规则, 如果定义了, 检查插入行在相应属性上的值是否遵守用户定义完整性规则, 若遵守, 可以执行插入操作; 否则不可以执行插入操作。

• **当执行删除操作时:** 一般只需要检查考虑参照完整性规则。如果是删除被参照关系中的行, 检查被删除行在主码属性上的值是否正在被相应的被参照关系的外码引用, 若不被引用, 可以执行删除操作; 若正在被引用, 有 3 种可能的做法: 不可以执行删除操作(拒绝删除), 或将参照关系中相应行在外码属性上的值改为空值后再执行删除操作(空值删除), 或将参照关系中相应行一起删除(级联删除)。

• **当执行更新操作时:** 因为更新操作可看成是先执行删除操作, 再执行插入操作, 因此是上述两种情况的综合。

9.1.3 关系数据库标准语言 SQL

SQL (Structured Query Language) 称为结构化查询语言。SQL 语言集数据查询(data query)、数据操纵(data manipulation)、数据定义(data definition)和数据控制(data control)功能于一体, 充分体现了关系数据库语言的特点和优点。

其主要特点如下。

• **综合统一。**SQL 语言集数据定义语言(DDL)、数据操纵语言(DML)和数据控制语言(DCL)的功能于一体, 语言风格统一, 可以独立完成数据库生命周期中的全部活动, 包括定义关系模式、录入数据以建立数据库、

查询、更新、维护、数据库重构、数据库安全性控制等一系列操作的要求，这就为数据库应用系统开发提供了良好的环境。例如，用户在数据库投入运行后，还可根据需要随时地逐步修改模式，并不影响数据库的运行，从而使系统具有良好的可扩充性。

- **高度非过程化。**非关系模型的数据库数据操纵语言是面向过程的语言，使用这样的语言进行数据操作，必须指定存取路径。而用 SQL 语言进行数据操作，用户只需提出“做什么”，而不必指明“怎么做”，因此用户无需了解存取路径，存取路径的选择以及 SQL 语句的操作过程由系统自动完成。这不但大大减轻了用户负担，而且有利于提高数据独立性。

- **面向集合的操作方式。**非关系数据模型采用的是面向记录的操作方式，操作的对象都是一条记录(一次一个记录)，需要指明如何用循环结构按照某条路径一条一条地把满足条件的记录读出来。而 SQL 语言采用集合操作方式，不仅查找结果可以是元组的集合，而且一次插入、删除、更新操作的对象也可以是元组的集合(一次一个集合)。

- **以同一种语法结构提供两种使用方式。**SQL 语言既是自含式语言，又是嵌入式语言。作为自含式语言，它能够独立地用于联机交互的使用方式，用户可以直接键入 SQL 命令对数据库进行操作。作为嵌入式语言，SQL 语句能够嵌入到高级语言(例如 C, COBOL, Fortran, PL/1)程序中，供程序员编写程序时使用。而在两种不同的使用方式下 SQL 语言的语法结构基本上是一致的。这种以统一的语法结构提供两种不同的使用方式的作法，为用户使用提供了极大的灵活性与方便性。

- **语言简洁，易学易用。**SQL 语言功能极强，但由于设计巧妙，语言十分简洁，完成 数据定义、数据操纵、数据控制的核心功能只用了 CREATE, DROP, ALTER, SELECT, INSERT, UPDATE, DELETE, GRANT, REVOKE 9 个动词，如表 9. 2 所示。

表 9.2 SQL 语言的动词

SQL 功能	动 词
数据查询	SELECT
数据定义	CREATE, DROP, ALTER
数据操纵	INSERT, UPDATE, DELETE
数据控制	GRANT, REVOKE

9. 1. 3. 1 SQL 数据库的体系结构

SQL 语言支持数据库 3 级模式结构，如图 9. 5 所示。有些术语与传统的关系数据库术语不同。在 SQL 中，模式对应于“基本表(base table)”，内模式对应于“存储文件”，外模式对应于“视图(view)”和部分基本表。元组对应于表中的“行(row)”，属性对应于表中的“列(column)”。

SQL 数据库体系结构特点如下：

- 一个 SQL 数据库是表(Table)的汇集。
- 一个 SQL 表由行集构成，一行是列的序列，每列对应一个数据项。
- 一个表可以带若干索引，索引也存放在存储文件中。
- 存储文件的逻辑结构组成了关系数据库的内模式。存储文件的物理结构是任意的，对用户是透明的。
- 一个表或者是一个基本表，或者是一个视图。

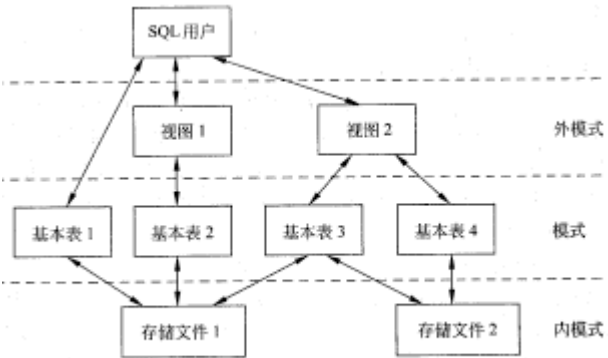


图 9.5 SQL 数据库的体系结构

基本表是实际存储在数据库中的表；视图是从一个或几个基本表或其他视图导出的表。它本身不独立存储在数据库中，即数据库中只存放视图的定义而不存放视图对应的数据，这些数据仍存放在导出视图的基本表中，因此视图是一个虚表。视图在概念上与基本表等同，都是关系，用户可以在视图上再定义视图。

- 一个基本表可以跨一个或多个存储文件存放，一个存储文件可存放一个或多个基本表。每个存储文件与外部存储器上一个物理文件对应。

- SQL 用户可以是应用程序，也可以是终端用户。SQL 的宿主语言有 Fortran, COBOL, Pascal, PL/I, C 和 Ada 等。SQL 也能作为独立的用户接口，供交互环境下的终端用户使用。

9. 1. 3. 2 SQL 的数据定义

关系数据库由模式、外模式和内模式组成，即关系数据库的基本对象是表、视图和索引。因此 SQL 的数据定义功能包括定义表、定义视图和定义索引，如表 9. 3 所示。由于视图是基于基本表的虚表，索引是依附于基本表的，因此 SQL

表 9.3 SQL 的数据定义语句

操作对象	操作方式		
	创 建	删 除	修 改
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	
索引	CREATE INDEX	DROP INDEX	

通常不提供修改视图定义和修改索引定义的操作。用户如果想修改视图定义或索引定义，只能先将它们删除掉，然后再重建。不过有些关系数据库产品如 Oracle 允许直接修改视图定义。

1.基本表

SQL 语言使用 CREATE TABLE 语句创建基本表，其一般格式如下：

```
CREATE TABLE<表名>(<列名><数据类型>[列级完整性约束条件]
    [, <列名><数据类型>[列级完整性约束条件]...]
    [, <表级完整性约束条件>] >
    [其他参数];
```

其中，任选项“其他参数”是与物理存储有关的参数，随具体系统的不同而不同；<表名>是所要创建的基本表的名字，它可以由一个或多个属性(列)组成定义表的各个属性时需要指明其<数据类型>。不同的数据库系统支持的数据类型不完全相同，实际使用时应根据具体数据库系统支持的数据类型声明。

建表的同时通常还可以定义与该表有关的完整性约束条件，这些完整性约束条件被存入系统的数据字典中，当用户对表进行操作时，由 DBMS 自动检查该操作是否违背所定义的完整性约束条件。声明完整性约束条件有两个层次(或称两个级别)：如果完整性约束条件涉及到该表的多个属性时，则必须在表级上定义，称为表级完整性约束条件；否则既可以在列级定义，也可以在表级定义，若在列级定义，称为列级完整性约束条件。

SQL 支持空值的概念，空值是未知值，任何非主属性列可以有空值，除非在 CREATE TABLE 语句列的定义中指定了 NOT NULL。

定义表的各个属性时必须指明其数据类型和长度。不同的数据库管理系统支持的数据类型不完全相同。具体的数据类型定义参照相应的数据库管理系统说明。

随着应用环境和应用需求的变化，有时需要修改已建立好的基本表，包括增加新列、增加新的完整性约束条件、修改原有的列定义或删除已有的完整性约束条件等。SQL 语言用 ALTER TABLE 语句修改基本表，其一般格式为：

```
ALTER TABLE<表名>
    [ADD<新列名><数据类型>[完整性约束]]
    [DROP<完整性约束名>]
    [MODIFY(列名)<数据类型>];
```

其中，<表名>为指定需要修改的基本表名 ADD 子句用于增加新列和新的完整性约束条件，DROP 子句用于删除指定的完整性约束条件，MODIFY 子句用于修改原有的列定义。

不论基本表中原来是否已有数据，新增加的列一律为允许空值。

SQL 没有提供删除属性列的语句，用户只能间接实现这一功能。即先将原表中要保留的列及其值复制到一个新表中，然后删除原表，再将新表重新命名为原表名。

当某个基本表不再需要时，可以使用 SQL 语句 DROP TABLE 进行删除。其一般格式为：

```
DROP TABLE<表名>;
```

基本表定义一旦被删除，表中的数据和在此表上建立的索引都将自动被删除掉，而建立在此表上的视图虽仍然保留，但已无法引用。因此执行删除基本表操作一定要格外小心。

2.索引

创建索引是加快表的查询速度的有效手段。SQL 语言支持用户根据应用的需要，在基本表上建立一个或多个索引，以提供多种存取路径，加快查找速度。一般说来，创建与删除索引由数据库管理员(DBA)或表的属主(即建立表的人)负责完成。系统在存取数据时会自动选择合适的索引作为存取路径，用户不必也不能选择索引。

创建索引的一般格式为：

```
CREATE[UNIQUE][CLUSTER]INDEX<索引名> ON<表名>(<列名>[<顺序>[, <列名>[<顺序>]]...);
```

其中，<表名>指定要建索引的基本表的名字。索引可以建在该表的一列或多列上，多列时各列名之间用逗号分隔。每个<列名>后面还可以用<顺序>指定索引值的排列顺序，包括 ASC(升序)和 DESC(降序)两种，缺省值为 ASC。UNIQUE 表示此索引的每一个索引值只对应惟一的数据记录。CLUSTER 表示要建立的索引是聚簇索引。所谓聚簇索引是指索引项的顺序与表中记录的物理顺序一致的索引组织。例如，执行下面的 CREATE INDEX 语句：CREATE CLUSTER INDEX name_index ON emp(name);

将会在 emp 表的 name 列上建立一个聚簇索引，而且 emp 表中的记录将按照 name 值的升序存放。

用户可以在最频繁查询的列上建立聚簇索引以提高查询效率。显然在一个基本表上最多只能建立一个聚簇索引。建立聚簇索引后，更新索引列数据时，往往导致表中记录的物理顺序的变更，代价较大，因此，对于经常更新的列不宜建立聚簇索引。

索引一经建立,就由系统使用和维护它,不需用户干预。创建索引是为了减少查询操作的时间,但如果数据增删改频繁,系统会花费许多时间来维护索引。这时,可以删除一些不必要的索引。

删除索引的一般格式为: DROP INDEX(索引名);

删除索引时,系统会同时从数据字典中删去有关该索引的描述。

9.1.3.3 SQL 的数据操纵

SQL 语言的数据操纵功能包括 SELECT, INSERT, DELETE 和 UPDATE 4 个语句,即检索查询和修改(包括插入、删除、更新)两部分功能。

1. SQL 的查询语句。

数据库查询是数据库操作的核心。SQL 语言提供了 SELECT 语句对数据库进行查询,该语句的一般格式是: SELECT [ALL|DISTINCT] <目标列表表达式>[, <目标列表表达式>]...

FROM<基本表(或视图)> [, <基本表(或视图)>] ...

[WHERE<条件表达式>]

[GROUP BY<列名 1>[HAVING<内部函数表达式>]]

[ORDER BY<列名 2>[ASC|DESC]];

整个语句的含义是:根据 WHERE 子句的条件表达,从基本表(或视图)中找出满足条件的元组,按 SELECT 子句中的目标列表表达式,选出元组中的属性值形成结果表。如果有 ORDER 子句,则结果表要根据指定的列名 2 按升序或降序排序。GROUP 子句将结果按列名 1 分组,每个组产生结果表中的一个元组。通常在每组中作用集合函数,分组的附加条件用 HAVING 短语给出,只有满足内部函数表达式的组才能输出。

SQL 语言对数据库的操作十分灵活方便,原因在于 SELECT 语句中的成分丰富多样,有许多可选形式,尤其是目标列和条件表达式。如包括简单查询、连接查询、嵌套查询、集合查询等。

2. SQL 的修改语句

SQL 的修改语句包括更新、删除和插入 3 类语句。

更新操作语句的一般格式为:

UPDATE<表名> SET<列名>=<表达式> [, <列名>=<表达式>] ... [WHERE 谓词];

语句的含义是:更新指定表中满足谓词的元组,把这些元组按 SET 子句中的表达式修改相应字段上的值。

删除语句的一般格式为: DELETE FROM 表名 [WHERE 谓词];

语句的含义是:从指定表中删除满足谓词的那些记录,没有 WHERE 子句时表示删去此表中的全部记录,但此表的定义仍在数据字典中。DELETE 语句删除的是表中的数据,而不是关于表的定义。

插入语句的一般格式有两种:

- 插入一个元组

INSERT INTO 表名[(字段名[, 字段名] ...)] VALUES(常量[, 常量] ...);

- 插入子查询结果

INSERT INTO 表名[(字段名[, 字段名] ...)] 子查询;

第一种格式把一个新记录插入指定的表中;第二种格式把子查询的结果插入指定的表中。若表中有些字段在插入语句中没有出现,则这些字段上的值取空值 NULL。当然在表定义中说明了 NOT NULL 的字段在插入时不能取 NULL,若插入语句中没有指出字段名,则新记录必须在每个字段上均有值。

插入和删除与更新操作一样,都会引起完整性被破坏的问题。支持关系模型的系统应该自动地检查,对破坏完整性的插入操作拒绝执行。

9.1.3.4 视图

视图是关系数据库系统提供给用户以多种角度观察数据库中数据的重要机制。

视图是从一个或几个基本表(或其他视图)导出的表,它与基本表不同,是一个虚表。数据库中只存放视图的定义,而不存放视图对应的数据,这些数据仍存放在原来的基本表中。基本表中的数据发生变化,从视图中查询出的数据也就随之改变了。

视图一经定义,就可以和基本表一样被查询、被删除,也可以在一个视图之上再定义新的视图,但对视图的修改(插入、删除、更新)操作则有一定的限制。

1. 创建视图

创建视图语句的一般格式为:

CREATE VIEW<视图名> [(<列名> [, <列名>] ...)] AS<子查询>[WITH CHECK OPTION];

其中,子查询可以是不含有 ORDER BY 子句和 DISTINCT 短语的任意的 SELECT 语句。

可选择项 WITH CHECK OPTION 表示当对视图进行 UPDATE, INSERT 和 DELETE 操作时,保证更新、插入

或删除的行满足视图定义中的谓词条件(即子查询中的条件表达式)。

如果 CREATE VIEW 语句仅指定了视图名, 省略了组成视图的各个属性列名, 则隐含该视图由子查询中 SELECT 子句目标列中的诸字段组成。

但在下列 3 种情况下必须明确指定组成视图的所有列名:

- 其中某个目标列不是单纯的属性名, 而是集合函数或列表表达式。
- 多表连接时选出了几个同名列作为视图的字段。
- 需要在视图中为某个列启用新的更合适的名字(重新命名列名)。

需要说明的是, 组成视图的属性列名必须依照上面的原则, 或者全部省略或者全部指定。

2. 删除视图

视图创建好后, 若导出此视图的基本表被删除了, 该视图将失效, 但一般不会被自动删除。删除视图通常需要显式地使用 DROP VIEW 语句进行。该语句的格式为:

```
DROP VIEW<视图名>;
```

一个视图被删除后, 由该视图导出的其他视图也将失效, 用户应该使用 DROP VIEW 语句将它们一一删除。

3. 查询视图

通过视图进行查询时, 首先要进行有效性检查, 检查查询涉及的表、视图等是否在数据库中存在。如果存在, 则从数据字典中取出查询涉及的视图的定义, 把定义中的子查询和用户对视图的查询结合起来, 转换成对基本表的查询, 然后再执行这个经过修正的查询。

把对视图的查询转换为对基本表的查询的过程称为视图的消解。视图可以和其他基本表一起使用, 实现连接查询或嵌套查询。这也就是说, 在关系数据库的 3 级模式结构中, 外模式不仅包括视图, 而且还可以包括一些基本表。

假定我们已经使用了下列语句建立了信息系学生的视图 IS_S:

```
CREATE VIEW IS_S AS SELECT s#, sname, age FROM S WHILE dept = "IS";
```

查询信息系选修 1 号课的学生的语句为:

```
SELECT s#, sname FROM IS_S WHILE S.s# = SC.s# AND c# = "1"
```

执行这个查询时, 将转化为对基本表的查询:

```
SELECT S#, sname FROM S, SC WHILE S.s# = SC.s# AND dept = "IS" AND c# = "1";
```

4. 修改视图

修改视图包括插入 (INSERT)、删除 (DELETE) 和更新 (UPDATE) 3 类操作。

由于视图是虚表, 因此对视图的更新最终要转换为对基本表的更新。

为防止用户通过视图对数据进行插入、删除和更新时, 无意或故意操作不属于视图范围内的基本表数据, 可在定义视图时加上 WITH CHECK OPTION 子句, 这样在视图上进行修改数据时, DBMS 会进一步检查视图定义中的条件, 若不满足条件, 则拒绝执行该操作。

与查询视图类似, DBMS 执行此语句时, 首先进行有效性检查, 检查所涉及的表、视图等是否在数据库中存在。如果存在, 则从数据字典中取出该语句涉及的视图的定义, 把定义中的子查询和用户对视图的更新操作结合起来, 转换成对基本表的更新(插入和删除亦同), 然后再执行这个经过修正的更新操作。

注意: 在关系数据库中, 并不是所有的视图都是可更新的, 因为有些视图的更新不能惟一地有意义地转换成对相应基本表的更新。

一般对所有行列子集视图都可以执行修改和删除元组的操作, 如果基本表中所有不允许空值的列都出现在视图中, 则也可以对其执行插入操作。除行列子集视图外, 还有些视图理论上可更新的, 但它们的确切特征还是尚待研究的课题。另外还有些视图从理论上是不可更新的。

目前各个关系数据库系统一般都只允许对行列子集视图的更新, 而且各个系统对视图的更新还有更进一步的规定, 由于各系统实现方法上的差异, 这些规定也不尽相同。

应该指出的是, 不可更新的视图与不允许更新的视图是两个不同的概念。前者指理论上已证明其是不可更新的视图。后者指实际系统中不支持其更新, 但它本身有可能是可更新的视图。

5. 视图的作用

视图是关系数据库系统提供给用户以多种角度观察数据库中数据的重要机制。视图最终是定义在基本表之上的, 对视图的一切操作最终也要转换为对基本表的操作。合适地定义和合理地使用视图的优点有:

• **视图能够简化用户的操作。** 视图机制使用户可以将注意力集中在他所关心的数据上。如果这些数据不是直接来自基本表, 则可以通过定义视图, 使用户眼中的数据库结构简单、清晰, 并且可以简化用户的

数据查询操作。例如，对于那些经常要通过计算或要从若干张表连接来获得数据的查询，可将这类查询定义为一个视图，然后就可容易地对该视图进行操作。

- **视图使用户能以多种角度观察同一数据。**视图机制能使不同的用户以不同的方式观察同一数据，当许多不同种类的用户使用同一个数据库时，这种灵活性是非常重要的。

- **视图对重构数据库提供了一定程度的逻辑独立性。**数据的逻辑独立性是指当数据库重构造时，如增加新的关系或对原有关系增加新的字段等，用户和用户程序不会受影响。在关系数据库中，数据库的重构造往往是不可避免的。当然，视图只能在一定程度上提供数据的逻辑独立性，比如由于对视图的更新是有条件的，因此应用程序中修改数据的语句可能仍会因基本表结构的改变而改变。

- **视图能够对机密数据提供安全保护。**有了视图机制，就可以在设计数据库应用系统时，对不同的用户定义不同的视图，使机密数据不出现在不应看到这些数据的用户视图上，这样就由视图的机制自动提供了对机密数据的安全保护功能。

9.1.3.5 SQL 的数据控制语句

这里主要讨论 SQL 语言的安全控制功能。

数据库管理系统保证数据安全的主要措施是进行存取控制，即规定不同用户对于不同数据对象所允许执行的操作，并控制各用户只能存取他有权存取的数据。不同的用户对不同的数据应具有不同的操作权利。

表 9.4 不同对象类型允许的操作权限

对 象	对象类型	操 作 权 限
属性列	TABLE	SELECT,INSERT,UPDATE,DELETE,ALL PRIVILEGES
视图	TABLE	SELECT,INSERT,UPDATE,DELETE,ALL PRIVILEGES
基本表	TABLE	SELECT,INSERT,UPDATE,DELETE,ALTER,INDEX,ALL PRIVILEGES
数据库	DATABASE	CREATE TABLE

1. 授予权限

SQL 语言用 GRANT 语句向用户授予数据访问的权限, GRANT 语句的一般格式为:

GRANT<权限>[, <权限>=...] [ON<对象类型><对象名>=...] TO(用户>[, <用户>=...] [WITH GRANT OPTION];

其语义为：将对指定操作对象的指定操作权限授予指定的用户。

对不同类型的操作对象有不同的操作权限，常见的操作权限如表 9.4 所示。

对属性列和视图的操作权限有查询(SELECT)、插入(INSERT)、更新(UPDATE)、删除(DELETE)以及这 4 种权限的总和(ALL PRIVILEGES)。

对基本表的操作权限有查询(SELECT)、插入(INSERT)、更新(UPDATE)、删除(DELETE)、修改表(ALTER)和建立索引(INDEX)以及这 6 种权限的总和(ALL PRIVILEGES)。

对数据库可以有建立表(CREATE TABLE)的权限，该权限属于 DBA，可由 DBA 授予普通用户，普通用户拥有此权限后可以建立基本表，基本表的属主(Owner)拥有对该表的一切操作权限。

接受权限的用户可以是一个或多个具体用户，也可以是 PUBLIC，即全体用户。

如果指定了 WITH GRANT OPTION 子句，则获得某种权限的用户还可以把这种权限再授予其他用户。如果没有指定 WITH GRANT OPTION 子句，则获得某种权限的用户只能使用该权限，但不能转授该权限。

GRANT 语句可以一次向一个用户授权，也可以一次向多个用户授权，还可以一次授予多个同类对象的权限，甚至一次可以完成对基本表、视图和属性列这些不同对象的授权，但授予关于 DATABASE 的权限必须与授予关于 TABLE 的权限分开，因为对象类型不同。

2. 收回权限

授予的权限可以由 DBA 或其他授权者用 REVOKE 语句收回，REVOKE 语句的一般格式为:

REVOKE<权限>[, <权限>=...] [ON<对象类型><对象名>=...] FROM<用户>[, <用户>=...]

SQL 提供了非常灵活的授权机制。DBA 拥有对数据库中所有对象的所有权限，并可以根据应用的需要将不同的权限授予不同的用户。所有授予出去的权限在必要时又都可以用 REVOKE 语句收回。

9.1.3.6 嵌入式 SQL

SQL 语言可以作为独立语言在终端交互方式下使用，在这种方式下使用的 SQL 语言是面向集合的描述性语言，是非过程性的，即大多数语句都是独立执行，与上下文无关的。

SQL 语言还可以嵌入到某种高级语言中使用，利用高级语言的过程性结构来弥补 SQL 语言实现复杂应用方面的不足。这种方式下使用的 SQL 语言称为嵌入式 SQL(Embedded SQL)，能嵌入 SQL 的高级语言称为主语言或宿主语言。

前面已经讲到 SQL 的特点之一是，在两种不同的使用方式下 SQL 语言的语法结构基本上是一致的。当然细节上会有许多差别，在程序设计的环境下，SQL 语句要做某些必要的扩充。

对宿主型数据库语言 SQL，DBMS 可采用两种方法处理，一种是预编译，另一种是修改和扩充主语言使之能处理 SQL 语句。目前采用较多的是预编译的方法。即由 DBMS 的预处理程序对源程序进行扫描，识别出 SQL 语句，把它们转换成主语言调用语句，以使主语言编译程序能识别它，最后由主语言的编译程序将

整个源程序编译成目标码，如图 9.6 所示。



图 9.6 嵌入式 SQL 的预编译处理

把 SQL 嵌入主语言使用时必须解决以下 3 个问题。

1. 区分 SQL 语句与主语言语句

这是对通过在所有的 SQL 语句前加前缀 EXEC SQL 来解决的(如例 5 程序中④，⑤，⑥，⑦)。SQL 语言的结束标志随主语言不同而不同。例如在 C 中以分号“;”结束：EXEC SQL<SQL 语句>;

2. 数据库工作单元和程序工作单元之间的通信

嵌入式 SQL 语句中可以使用主语言的程序变量来输入或输出数据。把 SQL 语句中使用的主语言程序变量简称为主变量。在 C 语言中用 BEGIN DECLARE SECTION 与 END DECLARE SECTION 语句说明(如例 5 程序中①，②)。

在 SQL 语句中使用这些主变量时，需在主变量名前加冒号“:”作标志，以区别于表中的字段名(如例 5 程序中⑤)。

SQL 语句执行后，系统要反馈给应用程序若干信息，这些信息送到 SQL 的通信区 SQLCA。SQLCA 用语句 EXEC SQL INCLUDE 加以定义(如例 5 程序中③)。SQLCA 是一个数据结构 SQLCA 中有一个存放每次执行 SQL 语句后返回代码的状态指示字段变量 SQLCODE。当 SQLCODE 为零时，表示 SQL 语句执行成功，否则返回一个错误代码(负值)或警告信息(正值)。程序员应该在每个 SQL 语句之后测试 SQLCODE 的值，以便处理各种情况。

3. 协调 SQL 语言和主语言的处理方式

一个 SQL 语句原则上可产生或处理一组记录，而主语言一次只能处理一个记录，为此必须协调两种处理方式。这是用游标(cursor)来解决的。

与游标有关的 SQL 语句有下列 4 个：

(1)游标定义语句(如例 5 程序中④)。游标是与某一查询结果相联系的符号名，游标用 SQL 的 DECLARE 语句定义，它是说明语句，此时游标定义中的 SELECT 语句并不执行。

(2)游标打开语句(如例 5 程序中⑤)。此时执行游标定义中的 SELECT 语句，同时游标处于活动状态。游标指向查询结果中的第一行之前。

(3)游标推进语句(如例 5 程序中⑥)。此时执行游标向前推进一行，并把游标指向的行(称为当前行)中的值取出，放到语句中说明的对应的程序变量中。FETCH 语句常置于主语言程序的循环中，并借助主语言的处理语句逐一处理查询结果中的每一行。

(4)游标关闭语句(如例 5 程序中⑦)。关闭游标，使它不再和原来的查询结果相联系。关闭了游标可以再次打开，与新的查询结果相联系。

在游标处于活动状态时，可以修改和删除游标指向的行。

为了更好地理解上面的概念，下面给出带有嵌入式 SQL 的一小段 C 程序。

(例习在表 S 和 SC 中检索某学生(姓名由主语言变量 GIVENSNAME 给出)的学习成绩信息(S# SNAME, C#, GRADE)。程序可这样编写：

```
EXEC SQL BEGIN DECLARE SECTION;..... ①主变量说明开始
    CHAR givensname(8);
    CHAR sno(5), cno(3), sname(8);
    INT grade;
EXEC SQL END DECLARE SECTION; .....②主变量说明结束
EXEC SQL INCLUDE SQLCA;..... ③定义 SQL 通信区
main()
{
    EXEC SQL DECLARE C1 CURSOR FOR ..... ④嵌入的 SQL 语句定义游标
        SELECT S#, SNAME, C#, GRADE FROM S, SC
            WHERE S# = SC.S# AND SNAME = ' givensname;
    EXEC SQL OPEN C1;..... ⑤打开游标
    for(;; )
    {
        EXEC SQL FETCH C1 INTO: sno,: sname,: cno,: grade;
        ..... ⑥推进游标指针并将当前数据放入主变量
        if(sqlca. sqlcode<>SUCCESS).....利用 SQLCA 中的状态信息决定何时退出循环
```

```

        break;
        printf("sno:%s, sname:%s, cno:%s, grade:%d" ,:sno, :sname,: cno,: grade);
        /*打印查询结果*/
    }
EXEC SQL CLOSE C1; ..... ⑦关闭游标
}

```

9.2 规范化理论与数据库设计

数据库设计是数据库应用领域中的主要研究课题。数据库设计的任务是针对一个给定的应用环境，在给定的(或选择的)硬件环境和操作系统及数据库管理系统等软件环境下，创建一个性能良好的数据库模式、建立数据库及其应用系统，使之能有效地存储和管理数据，满足各类用户的需求。

数据库设计需要理论作为指南。由 E. F. Codd 于 1971 年开始提出，以后又有了很大发展的关系数据库规范化理论就是数据库设计的一种理论指南。规范化理论研究的是关系模式中各属性之间的依赖关系及其对关系模式性能的影响，探讨“好”的关系模式应该具备的性质，以及达到“好”的关系模式的设计算法。规范化理论提供了判断关系模式优劣的理论标准，帮助预测可能出现的问题，提供了自动产生各种模式的算法，因此是设计人员的有力工具，也使数据库设计工作有了严格的理论基础。

规范化理论虽然最初是针对关系模式的设计而提出的，然而它不但对于关系模型数据库的设计，而且对于其他模型数据库的设计也都有重要的指导意义。

下面首先介绍关系数据库规范化理论，讨论关系数据库的逻辑结构设计问题。简单地说，就是如果要把一组数据存放到关系数据库中，应该设计一组什么样的关系模式，使我们既不必存储不必要的冗余信息，又可以方便地对信息进行存取。然后介绍数据库设计的几个阶段，各个阶段的任务，以及如何将关系数据库的规范化理论应用到数据库设计中。

9.2.1 “不好”的关系模式

在讨论“好”的关系模式应该具备什么性质之前，先看一个“不好”的关系模式。设有“供应者”关系模式 SUPPLIER(SNAME, SADDRESS, ITEM, PRICE)，其中各属性分别表示供应者名、供应者地址、货物名称、货物售价。一个供应者供应一种货物则对应到关系中的一个元组。

关系模式 SUPPLIER 有如下“毛病”：

- 数据冗余。一个供应者每供应一种货物，他的地址就要重复一次。
- 更新异常(不一致性的危险)。由于数据冗余，有可能在一个元组中更改了某供应者的地址，而没有更改另一个元组中同一供应者的地址，于是同一供应者有了两个不同地址，与实际情况不符。
- 插入异常。如果某供应者没有供应任何货物，则无法记录他的名称和地址。事实上，SNAME 和 ITEM 构成关系模式 SUPPLIER 的一个码(后面将给出码的严格定义)，码值的一部分为空的元组是不能插入到关系中的。
- 删除异常。如果一个供应者供应的所有货物都被删除，就会无可奈何地丢失了该供应者的名称和地址。

关系模式产生上述“毛病”的原因以及消除这些“毛病”的方法都与数据依赖的概念密切相关数据依赖是可以作为关系模式的取值的任何一个关系所必须满足的一种约束条件，是通过一个关系中数据间值的相等与否体现出来的相互关系。这是现实世界属性间相互联系的抽象，是数据内在的性质，是语义的体现。数据依赖极为普遍地存在于现实世界中。例如上述关系模式 SUPPLIER，由于客观情况是每个供应者只有一个地址，因而当 SNAME 的值确定之后，SADDRESS 的值也就被惟一确定了。关系模式 SUPPLIER 的任何一个关系中都不可能存在两个元组，它们在 SNAME 上的取值相等，而在 SADDRESS 上的取值不等，这就是一种数据依赖。

现在人们已经提出了许多种类型的数据依赖，其中最重要的是函数依赖和多值依赖。

9.2.2 函数依赖

1.函数依赖的定义

设 $R(A_1, A_2, \dots, A_n)$ 是一个关系模式， X 和 Y 是 $\{A_1, A_2, \dots, A_n\}$ 的子集，若只要关系 r 是关系模式 R 的可能取值，则 r 中不可能有两个元组在 X 中的属性值相等，而在 Y 中的属性值不等，则称“ X 函数决定 Y ”，或“ Y 函数依赖于 X ”，记作 $X \rightarrow Y$ 。

注意，函数依赖 $X \rightarrow Y$ 的定义要求关系模式 R 的任何可能的 r 都满足上述条件。因此不能仅考察关系模式 R 在某一时刻的关系 r 就断定某函数依赖成立。例如，关系模式 $R(SNO, NAME, AGE)$ ，可能在某一时刻， R 的关系 r 中每个学生的年龄都不同，也就是说没有两个元组在 AGE 属性上取值相同，而在 SNO 属性

上取值不同,但我们决不可据此就断定 $AGE \rightarrow SNO$ 。很有可能在另一时刻, R 的关系 r 中有两个元组在 AGE 属性上取值相同,而在 SNO 属性上取值不同。

函数依赖是语义范畴的概念,我们只能根据语义来确定函数依赖。例如,在没有同名的情况下, $NAME \rightarrow AGE$,而在有同名的情况下,这个函数依赖就不成立了。

若 $X \rightarrow Y$,但 $Y \not\subseteq X$,则称 $X \rightarrow Y$ 为非平凡的函数依赖。

若 $X \rightarrow Y$,则称 X 为决定因素。

若 $X \rightarrow Y, Y \rightarrow X$,则记作 $X \leftrightarrow Y$ 。

若 Y 函数不依赖于 X,则记作 $X \nrightarrow Y$ 。

在关系模式 R 中,如果 $X \rightarrow Y$,并且对于 X 的任何一个真子集 X' ,都有 $X' \nrightarrow Y$,则称 Y 对 X 完全函数依赖,记作 $X \xrightarrow{f} Y$ 。

若 $X \rightarrow Y$,但 Y 不完全函数依赖于 X,则称 Y 对 X 部分函数依赖,记作 $X \xrightarrow{p} Y$ 。

在关系模式 R 中,如果 $X \rightarrow Y, (Y \not\subseteq X), Y \nrightarrow X, Y \rightarrow Z$,则称 Z 对 X 传递函数依赖。

例如,在关系模式 SC(SNO, CNO, G, CREDIT)中,

$(SNO, CNO) \xrightarrow{f} G$ 成绩完全函数依赖于学号,课号

$CNO \rightarrow CREDIT$ 学分函数依赖于课号

$(SNO, CNO) \xrightarrow{p} CREDIT$ 学分部分函数依赖于学号,课号

在关系模式 S(SNO, NAME, AGE, DNO, DEAN)中,

$SNO \rightarrow NAME$

$SNO \rightarrow AGE$

$SNO \rightarrow DNO, DNO \nrightarrow SNO, DNO \rightarrow DEAN$

$SNO \rightarrow DEAN$ 系主任传递函数依赖于学号

函数依赖(以及后面要介绍的多值依赖)是数据的重要性质,关系模式应能反映这些性质。因此,以下把关系模式表示成 $R(U, F)$,其中 U 是一组属性, F 是属性组 U 上的一组数据依赖。当且仅当 U 上的一个关系 r 满足 F 时, r 称为关系模式 $R(U, F)$ 的一个关系。

2.函数依赖的逻辑蕴含

设 $R(U, F)$ 是一个关系模式, $X; Y$ 是 U 中属性组,若在 $R(U, F)$ 的任何一个满足 F 中函数依赖的关系 r 上,都有函数依赖 $X \rightarrow Y$ 成立,则称 F 逻辑蕴含 $X \rightarrow Y$ 。

在关系模式 $R(U, F)$ 中为 F 所逻辑蕴含的函数依赖的全体称作 F 闭包,记作 F^+ 。

例如,关系模式 S(SNO, NAME, AGE, DNO, DEAN),其属性组上的函数依赖集为 $F = \{SNO \rightarrow NAME, SNO \rightarrow AGE, SNO \rightarrow DNO, DNO \rightarrow DEAN\}$, $SNO \rightarrow DEAN$ 就是 F 所逻辑蕴含的一个函数依赖。

3.码

设 K 为关系模式 $R(U, F)$ 中的属性或属性组,若 $K \rightarrow U$ 在 F^+ 中,而找不到 K 的任何一个真子集 K' ,能使 $K' \rightarrow U$ 在 F^+ 中,则称 K 为关系模式 R 的候选码。当候选码多于一个时,选定其中一个作主码。

包含在任何一个候选码中的属性叫做主属性。不包含在任何一个候选码中的属性叫做非主属性。最简单的情况,单个属性是码。最极端的情况,整个属性组是码,称作全码。

下面举一个关系模式中包括多个候选码的例子。关系模式 CSZ(CITY, ST, ZIP),其属性组上的函数依赖集为 $F = \{(CITY, ST) \rightarrow ZIP, ZIP \rightarrow CITY\}$,即城市、街道决定邮政编码,邮政编码决定城市。容易看出, (CITY, ST) 和 (ST, ZIP) 是两个候选码。CITY, ST, ZIP 都是主属性。

4.函数依赖的公理系统

为了确定一个关系模式的码,为了从一组函数依赖求得蕴含的函数依赖,需要从 F 计算 F^+ ,或者至少需要判断函数依赖 X, Y 是否在 F^+ 中。为此需要一套推理规则。这样的推理规则 1974 年由 Armstrong 首先提出来,称作 Armstrong 公理系统,它包括 3 条推理规则。

设 F 是属性组 U 上的一组函数依赖,于是有如下推理规则:

- 自反律。若 $Y \subseteq X \subseteq U$,则 $X \rightarrow Y$ 为 F 所逻辑蕴含。
- 增广律。若 $X \rightarrow Y$ 为 F 所逻辑蕴含,且 $Z \subseteq U$,则 $XZ \rightarrow YZ$ 为 F 所逻辑蕴含。
- 传递律。若 $X \rightarrow Y$ 及 $Y \rightarrow Z$ 为 F 所逻辑蕴含,则 $X \rightarrow Z$ 为 F 所逻辑蕴含。

注意:由自反律所得到的函数依赖均是平凡的函数依赖,事实上自反律的应用只依赖于 U,不依赖于

F。

作为一个例子，我们看看怎样对上面提到的关系模式 CSZ 证明 $(ST, ZIP) \rightarrow (CITY, ST, ZIP)$ ，即说明 (ST, ZIP) 是一个候选码。证明步骤如下：

- (1) $ZIP \rightarrow CITY$ (F 中已给出)；
- (2) $(ST, ZIP) \rightarrow (CITY, ST)$ (对 (1) 用增广律，加 ST)；
- (3) $(ST, ZIP) \rightarrow (CITY, ST, ZIP)$ (对 (2) 用增广律，加 ZIP)。

严格地说，要证明 (ST, ZIP) 是码，还需要说明 $ST \rightarrow (CITY, ST, ZIP)$ 和 $ZIP \rightarrow (CITY, ST, ZIP)$ 都不在 F^+ 中。

Armstrong 公理系统是正确、完备的。即由 F 出发根据推理规则推导出的函数依赖一定为 F 所逻辑蕴含，而且 F 所逻辑蕴含的每一个函数依赖必定可以由 F 出发根据推理规则推导出来。

根据 Armstrong 公理系统的 3 条推理规则可以得到下面 3 条很有用的推理规则：

- 合并规则。由 $X \rightarrow Y, X \rightarrow Z$ ，有 $X \rightarrow YZ$ 。
- 伪传递规则。由 $X \rightarrow Y, WY \rightarrow Z$ ，有 $XW \rightarrow Z$ 。
- 分解规则。由 $X \rightarrow Y$ 及 $Z \subseteq Y$ ，有 $X \rightarrow Z$ 。

9.2.3 关系模式的规范化

在 9.2.1 中我们考察了一个“不好”的关系模式，以及它的毛病。这一节里我们要讨论“好”的关系模式应该具备的性质，即关系模式的规范化问题。

1. 第一范式(1NF)及进一步规范化.

关系模式需要满足一定的条件，不同程度的条件称作不同的范式。最低要求的条件是元组的每个分量必须是不可分的数据项，这叫做第一范式，简称 1NF，是最基本的规范化。在第一范式的基础上进一步增加一些条件，则为第二范式。以此类推，还有第三范式，Boyce-Codd 范式等等。

本来，所谓“第几范式”是表示关系模式满足的一定条件，所以经常称某一关系模式为第几范式的关系模式。然而，通常我们又把范式这个概念理解为符合某种条件的关系模式的集合，所以 R 为第二范式的关系模式也可以写成 $R \in 2NF$ 。

各种不同的范式都是以对关系模式的属性间允许的数据依赖加以限制的形式表示的。这一节我们在函数依赖的范围内讨论。

函数依赖 $X \rightarrow Y$ 不仅给出了对关系的值的限制，而且给出了数据库中应该存储的某种联系：从 X 的值应该知道与之联系的惟一 Y 值。若 X 不含码，则有麻烦了。码是一个元组区别于其他元组的依据，同时也是元组赖以存在的条件。在一个关系中，不可能存在两个不同的元组在码属性上取值相同，也不可能存在码或码的一部分为空值的元组。若某关系模式的属性间有函数依赖 $X \rightarrow Y$ ，而 X 又不包含码，那么在具有相同 X 值的所有元组中，某个特定的 Y 值就会重复出现，这是数据冗余，随之而来的是更新异常问题；某个 X 值与某个特定的 Y 值相联系，这是数据库中应存储的信息，但由于 X 不含码，这种 X 与 Y 相联系的信息可能因为码或码的一部分为空值而不能作为一个合法的元组在数据库中存在，这是插入异常或删除异常问题。

第二范式、第三范式和 Boyce-Codd 范式就是不同程度地限制关系模式中 X 不包含码的函数依赖 $X \rightarrow Y$ 的存在。

2. 第二范式(2NF)

若关系模式 $R \in 1NF$ ，且每一个非主属性完全函数依赖于码，则 $R \in 2NF$ 。

2NF 就是不允许关系模式的属性之间有这样的函数依赖 $X \rightarrow Y$ ，其中 X 是码的真子集，Y 是非主属性，即不允许有非主属性对码的部分函数依赖。

让我们再看看那个“不好”的关系模式：SUPPLIER(SNAME, SADDRESS, ITEM, PRICE)，其属性组上的函数依赖集是 $F = \{SNAME \rightarrow SADDRESS, (SNAME, ITEM) \rightarrow PRICE\}$ 。显然 (SNAME, ITEM) 是码，SADDRESS, PRICE 是非主属性， $SNAME \rightarrow SADDRESS$ 是非主属性对码的部分函数依赖，所以关系模式 $SUPPLIER \notin 2NF$ 。

9.2.1 节中指出的关系模式 SUPPLIER 的种种“毛病”就是由于 $SNAME \rightarrow SADDRESS$ 这个函数依赖的存在造成的。SNAME 是码的真子集，可能在许多元组中 SNAME 取相同值，于是在这些元组中 SADDRESS 也取相同值，数据的冗余引起了更新异常。SNAME 的值确定了，则 SADDRESS 的值也随之确定，这样的信息应存在数据库中；但由于 SNAME 只是码的真子集，若码的其余部分值为空，则其供应者的地址是什么这样的有用信息无法作为一个元组存在于数据库中，这就是插入异常和删除异常的问题。

可以用分解的方法将一个非 2NF 的关系模式分解为多个 2NF 的关系模式。例如，将 SUPPLIER 分解为两个关系模式 SUPPLIER1(SNAME, SADDRESS) 和 SUPPLY(SNAME, ITEM, PRICE)，就不再有非主属性对码的部

分依赖，都是 2NF 的关系模式了。

3. 第三范式(3NF)

若关系模式 $R \in 2NF$ ，且每一个非主属性都不传递依赖于码，则 $R \in 3NF$ 。

3NF 就是不允许关系模式的属性之间有这样的非平凡函数依赖 $X \rightarrow Y$ ，其中 X 不包含码， Y 是非主属性。 X 不包含码有两种情况，一种情况 X 是码的真子集，这是 2NF 不允许的，另一种情况 X 不是码的真子集，这是 3NF 不允许的。

我们曾考察过关系模式：S(SNO, NAME, AGE, DNO, DEAN)，其属性组上的函数依赖集是 $F = \{SNO \rightarrow NAME, SNO \rightarrow AGE, SNO \rightarrow DNO, DNO \rightarrow DEAN\}$ 。显然 SNO 是码，其余的属性都是非主属性。非主属性 DEAN 传递依赖于码 SNO，所以 $S \notin 3NF$ 。但没有非主属性对码的部分依赖，所以 $S \in 2NF$ 。

关系模式 S 也存在数据冗余、更新异常、插入异常、删除异常等问题。这些问题是由于 $DNO \rightarrow DEAN$ 这个函数依赖的存在造成的。DNO 不包含码，可能在许多元组中 DNO 取相同值，于是在这些元组中 DEAN 也取相同值，数据的冗余引起了更新异常。DNO 的值确定了，则 DEAN 的值也随之确定，这样的信息应存在数据库中，但由于 DNO 不含码，则可能因为码值为空而不能将某系的系主任是谁这样的有用信息作为一个元组存入数据库中，这就是插入异常和删除异常的问题。

可以用分解的方法将一个非 3NF 的关系模式分解为多个 3NF 的关系模式。例如，将 S 分解为两个关系模式 SI(SNO, NAME, AGE, DNO)和 DEPT(DNO, DEAN)，就不再有非属性对码的传递依赖，都是 3NF 的关系模式了。

4. Boyce-Codd 范式(BCNF)

若关系模式 $R \in 1NF$ ，且对于每一个非平凡的函数依赖 $X \rightarrow Y$ ，都有 X 包含码，则 $R \in BCNF$ 。

BCNF 是 3NF 的进一步规范化，即限制条件更严格。3NF 不允许有 X 不包含码， Y 是非主属性的非平凡函数依赖 $X \rightarrow Y$ 。BCNF 则不管 Y 是主属性还是非主属性，只要 X 不包含码，就不允许有 $X \rightarrow Y$ 这样的非平凡函数依赖。因此，若 $R \in BCNF$ ，则必然 $R \in 3NF$ 。然而，BCNF 又是概念上更加简单的一种范式，判断一个关系模式是否属于 BCNF，只要考察每个非平凡函数依赖 $X \rightarrow Y$ 的决定因素 X 是否包含码就行了。

在前面见到过关系模式 CSZ(CITY, ST, ZIP)，其属性组上的函数依赖集是 $F = \{(CITY, ST) \rightarrow ZIP, ZIP \rightarrow CITY\}$ 。(CITY, ST)和(ST, ZIP)是两个候选码，没有非主属性，自然 CSZ $\in 3NF$ 。但函数依赖 $ZIP \rightarrow CITY$ 的决定因素 ZIP 不包含码，所以 $CSZ \notin BCNF$ 。

关系模式 CSZ 也存在种种“毛病”，例如，若无街道信息，则一个邮政编码是哪个城市中的邮政编码这样的信息无法存在于数据库中。若将 CSZ 分解为两个关系模式 ZC(ZIP, CITY)和 SZ(ST, ZIP)，就不再有非平凡的函数依赖的决定因素中不包含码的情况，都是 BCNF 的关系模式了。

1NF, 2NF, 3NF, BCNF 的相互关系是 $BCNF \subset 3NF \subset 2NF \subset 1NF$ 。在函数依赖的范畴内，BCNF 达到了最高的规范化程度。

9.2.4 多值依赖和 4NF

1. 多值依赖

在函数依赖的范畴内，BCNF 达到了最高的规范化程度。BCNF 的关系模式是否很完美呢？让我们看一个例子。

关系模式 WSC(W, S, C)中 W 表示仓库, S 表示保管员, C 表示物品。假设每个仓库有若干个保管员，存放若干种物品，每种物品由存放仓库中的所有保管员负责保管。现有仓库、保管员、物品一组数据如图 9.7 所示。

这一组数据表示成关系模式 WSC 的二维表，如图 9.8 所示。

关系模式 WSC 的属性之间没有任何函数依赖，(W, S, C)是码， $WSC \in BCNF$ 。但关系模式有明显的冗余数据毛病。若仓库 W1 增加一个保管员 S3，则必须插入 W1S3C1, W1S3C2, W1S3C3 三个元组，若仓库 W2 减少一种物品 C4，则必须删除 W2S1C4, W2S3C4 两个元组。造成上述问题的原因是关系模式 WSC 的属性之间存在的一种称为多值依赖的数据依赖。

直观地，关系模式 WSC 中的多值依赖以如下形式表现出来：对于每一对 W, C 值，都有 S 的一组值与之对应，这一组 S 值只依赖于 W 值，不依赖于 C 值。

多值依赖的定义是：设 R 是属性集 U 上的一个关系模式，X, Y 是 U 的子集， $Z = U - X - Y$ 。若在 R 的任一关系 r 中，只要存在元组 t, s，使得 $t[X] = s[X]$ ，就必然存在元组 w, v (w, v 可以与 s, t 相同)，使得

Figure 9.7 is a diagram showing two sets of data, W1 and W2, each associated with a set of S values and a set of C values. W1 is associated with {S1, S2} and {C1, C2, C3}. W2 is associated with {S1, S2} and {C3, C4}.

图 9.7 关于仓库、保管员、物品的一组数据 WSC

W	S	C
W1	S1	C1
W1	S1	C2
W1	S1	C3
W1	S2	C1
W1	S2	C2
W1	S2	C3
W2	S2	C3
W2	S1	C4
W2	S3	C3
W2	S3	C4

图 9.8 图 9.7 数据对应的二维表

$w[X]=v[X]=t[X]=s[X]$, 而 $w[Y]=t[Y]$, $w[Z]=s[Z]$, $v[Y]=s[Y]$, $v[Z]=t[Z]$, 则称 Y 多值依赖于 X , 记作 $X \twoheadrightarrow Y$ 。

若 $X \twoheadrightarrow Y$, 而 $Z = \Phi$, 则称 $X \twoheadrightarrow Y$ 为平凡的多值依赖。

在关系模式 WSC 中, 存在多值依赖 $W \twoheadrightarrow S$ 。

多值依赖具有以下性质

- 若 $X \twoheadrightarrow Y$, 则 $X \twoheadrightarrow Z$, 其中 $Z=U-X-Y$, 即多值依赖具有对称性。从关系模式 WSC 的例子可以看出 $W \twoheadrightarrow S$ 的同时显然有 $W \twoheadrightarrow C$ 。

- 若 $X \rightarrow Y$, 则 $X \twoheadrightarrow Y$, 即函数依赖可以看作多值依赖的特殊情况。因为当 $X \rightarrow Y$ 时, 对于 X 的每一个值 x , 都有 Y 的一个确定值 y 与之对应。这符合对多值依赖 $X \twoheadrightarrow Y$ 的描述, 只是将 Y 的“一组值”换成“一个值”, 因此函数依赖是多值依赖的特殊情况。

- 设属性集之间的包含关系是 $XY \subseteq W \subseteq U$, 那么当 $X \twoheadrightarrow Y$ 在 $R(U)$ 上成立时, $X \twoheadrightarrow Y$ 也在 $R(W)$ 上成立; 反过来当 $X \twoheadrightarrow Y$ 在 $R(W)$ 上成立时, $X \twoheadrightarrow Y$ 在 $R(U)$ 上不一定成立。即多值依赖的有效性与属性集的范围有关。这是因为多值依赖的定义中不仅涉及属性组 X, Y , 而且涉及 U 中其余属性 Z 。一般地, 在 $R(U)$ 上若有 $X \twoheadrightarrow Y$ 在 $R(W)$ ($W \subseteq U$) 上成立, 则称 $X \twoheadrightarrow Y$ 为 $R(U)$ 的嵌入型多值依赖。

比较一下, 函数依赖 $X \rightarrow Y$ 只与属性集 X, Y 有关, 与其他属性无关。只要 $X \rightarrow Y$ 在 $R(XY)$ 上成立, 则 $X \rightarrow Y$ 在任何 $R(W)$ 上成立 ($XY \subseteq W \subseteq U$)。

- 若 $X \twoheadrightarrow Y$ 在 $R(U)$ 上成立, 且 $Y' \subset Y$, 我们不能断言 $X \twoheadrightarrow Y'$ 在 $R(U)$ 上成立。这也是因为多值依赖的定义中涉及了 U 中除 X, Y 之外的其余属性 Z , 考虑 $X \twoheadrightarrow Y'$ 是否成立时涉及的其余属性 $Z' = U - X - Y'$ 比确定 $X \twoheadrightarrow Y$ 成立时涉及的其余属性 $Z = U - X - Y$ 包含的属性列多, 因此 $X \twoheadrightarrow Y'$ 不一定成立。

比较一下, 若函数依赖 $X \rightarrow Y$ 在 $R(U)$ 上成立, 且 $Y' \subset Y$, 那么肯定 $X \rightarrow Y'$ 在 $R(U)$ 上成立。

2. 第四范式(4NF)

若关系模式 $R \in 1NF$, 且对于每一个非平凡的多值依赖 $X \twoheadrightarrow Y$ ($Y \not\subseteq X$), 都有 X 包含码, 则 $R \in 4NF$ 。

4NF 就是限制关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖。因为根据定义, 要求每一个非平凡的多值依赖 $X \twoheadrightarrow Y$ ($Y \not\subseteq X$), 都有 X 包含码, 于是 $X \rightarrow Y$, 所以所允许的非平凡多值依赖实际上是函数依赖。

关系模式 WSC 中, $W \twoheadrightarrow S$, $W \twoheadrightarrow C$ 都是非平凡的多值依赖, 而 W 中又不含码因此 $WSC \notin 4NF$ 。正是由于 $W \twoheadrightarrow S$, $W \twoheadrightarrow C$ 这样的非平凡且非函数依赖的多值依赖的存在, 造成了关系模式 WSC 的数据冗余问题。若将 WSC 分解为两个关系模式 $WS(W, S)$ 和 $WC(W, C)$, 就不再有非平凡且非函数依赖的多值依赖, 都是 4NF 的关系模式了。

虽然 4NF 是基于多值依赖的概念定义的, 但 4NF 是 BCNF 的进一步规范化。容易证明, 若 $R \in 4NF$, 则必然 $R \in BCNF$ 。于是我们有 $4NF \subset BCNF \subset 3NF \subset 2NF \subset 1NF$ 。

9.2.5 关系模式的分解

为提高规范化程度, 我们都是通过把低一级的关系模式分解为若干个高一级的关系模式来实现的。这样的分解使各关系模式达到某种程度的分离, 让一个关系模式描述一类实体或实体间的一种联系, 即采用所谓“一事一地”的设计原则。

然而, 对于同一个关系模式可能有多种分解方案。例如, 关系模式 $S(SNO, DNO, DORMNO)$, 其属性组上的函数依赖集是 $F = \{SNO \rightarrow DNO, DNO \rightarrow DORMNO\}$

显然 $S \notin 3NF$ 。对关系模式 S 至少有 3 种分解方案。

- 分解 1: $S11(SNO, DORMNO)$, $S12(DNO, DORMNO)$ 。
- 分解 2: $S21(SNO, DNO)$, $S22(SNO, DORMNO)$ 。
- 分解 3: $S31(SNO, DNO)$, $S32(DNO, DORMNO)$ 。

每种分解方案得到的两个关系模式都属于 3NF(事实上, 都属于 BCNF 和 4NF)。如何比较这 3 种分解方案的优劣呢? 将一个关系模式分解为多个关系模式时除提高规范化程度外, 还需要有什么别的考虑吗? 回答是肯定的。

1. 模式分解的等价标准

规范化过程中将一个关系模式分解为若干个关系模式, 应该保证分解后产生的模式与原来的模式等价。常用的等价标准有两种: 要求分解是具有无损连接性的和要求分解是保持函数依赖的。

将一个关系模式 $R(U, F)$ 分解为若干个关系模式 $R1(U1, F1)$, $R2(U2, F2)$, \dots , $Rn(Un, Fn)$ (其中 $U=U1 \cup U2 \cup \dots \cup Un$, Fi 为 F 在 Ui 上的投影), 意味着相应地将存储在一个二维表 r 中的数据分散到若干个二维表 $r1, r2, \dots, rn$ 中去(其中 ri 是 r 在属性组 Ui 上的投影)。我们当然希望这样的分解不丢失信息, 也就是说, 希望能通过对关系 $r1, r2, \dots, rn$ 的自然连接运算重新得到关系 r 中的所有信息。

事实上, 将关系 r 投影为 r_1, r_2, \dots, r_n 时并不会丢失信息, 关键是对 r_1, r_2, \dots, r_n 做自然连接时可能产生一些原来 r 中没有的元组, 从而无法区别哪些元组是 r 中原来有的, 即数据库中应该存在的数据, 在这个意义上丢失了信息。

例如, 设关系模式 $S(SNO, DNO, DORMNO)$ 在某一时刻的关系 r 如图 9.9 所示。

SNO	DNO	DORMNO
S_1	D_1	A
S_2	D_2	B
S_3	D_2	B
S_4	D_3	A

图 9.9 关系模式 S 在某一时刻的关系 r

若按分解 1 将关系模式 S 分解为 $S_{11}(SNO, DORMNO)$ 和 $S_{12}(DNO, DORMNO)$, 则将 r 投影到 S_{11} 和 S_{12} 的属性上, 得到关系 r_{11} 和 r_{12} , 如图 9.10 所示。

SNO	DORMNO	DNO	DORMNO
S_1	A	D_1	A
S_2	B	D_2	B
S_3	B	D_2	B
S_4	A	D_3	A

图 9.10 分解 1 所得到的结果

做自然连接 $r_{11} * r_{12}$, 得到 r' , 如图 9.11 所示。

r' 中的元组 S_1D_3A 和 S_4D_1A 不是原来 r 中的元组。就是说, 我们无法知道原来的: 中到底有哪些元组, 这当然是我们所不希望的。

SNO	DNO	DORMNO
S_1	D_1	A
S_1	D_3	A
S_2	D_2	B
S_3	D_2	B
S_4	D_1	A
S_4	D_3	A

图 9.11 自然连接 $r_{11} * r_{12}$ 的结果

设关系模式 $R(U, F)$ 分解为关系模式 $R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_n(U_n, F_n)$, 若对于 R 的任何一个可能的 r , 都有 $r = r_1 * r_2 * \dots * r_n$, 即 r 在 R_1, R_2, \dots, R_n 上的投影的自然连接等于 r , 则称关系模式 R 的这个分解是具有无损连接性的。

分解 1 不具有无损连接性, 这是一个不好的分解方案。

在将一个关系模式分解为 3 个或更多个关系模式的情况下, 要判别分解是否具有无损连接性需要比较复杂的算法。然而若将一个关系模式分解为两个关系模式, 则很容易判别分解是否具有无损连接性。

关系模式 $R(U, F)$ 分解为关系模式 $R_1(U_1, F_1), R_2(U_2, F_2)$ 是具有无损连接性的分解的充分必要条件是 $(U_1 \cap U_2 \rightarrow U_1 - U_2) \in F^+$, 或 $(U_1 \cap U_2 \rightarrow U_2 - U_1) \in F^+$ 。

再考虑一下前述的分解方案 2, 将关系模式 S 分解为 $S_{21}(SNO, DNO), S_{22}(SNO, DORMNO)$

由于 $U_1 \cap U_2 = SNO, U_1 - U_2 = DNO$, 显然 $U_1 \cap U_2 \rightarrow U_1 - U_2$, 所以分解 2 具有无损连接性。然而分解 2 也不是一个很好的分解方案, 将前面例子中的关系 r 投影到 S_{21}, S_{22} 的属性上, 得到关系 r_{21} 和 r_{22} , 如图 9.12 所示。

SNO	DORMNO	SNO	DORMNO
S_1	D_1	S_1	A
S_2	D_2	S_2	B
S_3	D_2	S_3	B
S_4	D_3	S_4	A

图 9.12 分解 2 所得到的结果

假设学生 S_3 从 D_2 系转到 D_3 系, 于是需要在 r_{21} 中将元组 S_3D_2 修改为 S_3D_3 , 同时在 r_{22} 中将元组 S_3B 修改为 S_3A 。如果这两个修改没有同时完成, 数据库中就会存在不一致信息。这是因为分解得到的两个关系模式不是互相独立造成的。

F 中的函数依赖 $DNO \rightarrow DORMNO$ 既没有投影到关系模式 S_{21} 中, 也没有投影到关系模式 S_{22} 中, 而是跨在两个关系模式上。函数依赖是数据库中的完整性约束条件。在 r 中, 若两个元组的 X 值相等, 则 Y 值也必须相等。现在 r 的一个元组中的 X 值和 Y 值跨在两个不同的关系中, 为维护数据库的一致性, 在一个关系中修改 X 值时就需要相应地在另一个关系中修改 Y 值, 这当然是很麻烦而且容易出错的, 于是就有要求模式分解保持函数依赖这条等价标准。

设关系模式 $R(U, F)$ 分解为关系模式 $R_1(U_1, F_1), R_2(U_2, F_2), \dots, R_n(U_n, F_n)$, 若 $F' = (F_1 \cup F_2 \cup \dots \cup F_n)^+$, 即 F 所逻辑蕴含的函数依赖一定也由分解得到的各个关系模式中的函数依赖所逻辑蕴含, 则称关系模式 R 的这个分解是保持函数依赖的。

分解 2 不是保持函数依赖的, 因为分解得到的关系模式中只有函数依赖 $SNO \rightarrow DNO, SNO \rightarrow DORMNO$, 丢失了函数依赖 $DNO \rightarrow DORMNO$ 。分解 3 是保持函数依赖的。

2. 关于模式分解的几个事实

• 分解具有无损连接性和分解保持函数依赖是两个互相独立的标准。具有无损连接性的分解不一定保持函数依赖, 例如分解 2; 保持函数依赖的分解不一定具有无损连接性, 例如, 关系模式 $SC(SNO, DNO, CNO, CREDIT)$, 其属性组上的函数依赖集为 $F = \{SNO \rightarrow DNO, CNO \rightarrow CREDIT\}$ 。分解为两个关系模式 $SC_1(SNO, DNO)$ 和 $SC_2(CNO, CREDIT)$, 这个分解是保持函数依赖的, 但不具有无损连接性。

因此, 关系模式的一个分解可能是具有无损连接性, 可能是保持函数依赖的, 也可能是既具有无损连接性又保持函数依赖的。

• 若要求分解具有无损连接性, 那么模式分解一定可以达到 BCNF。

- 若要求分解保持函数依赖，那么模式分解可以达到 3NF，但不一定能达到 BCNF。
- 若要求分解既具有无损连接性，又保持函数依赖，则模式分解可以达到 3NF，但不一定能达到 BCNF。

在此不再讨论模式分解的算法，有兴趣的读者可参阅有关参考书。

9.2.6 数据库设计过程

数据库设计工作量大而且过程比较复杂，是一项数据库工程也是一项庞大的软件工程。数据库设计包括结构特性的设计和行为特性的设计两方面的内容。结构特性的设计是指确定数据库的数据模型。数据模型反映了现实世界的的数据及数据间的联系，要求在满足应用需求的前提下，尽可能减少冗余，实现数据共享。行为特性的设计是指确定数据库应用的行为和动作，应用的行为体现在应用程序中，所以行为特性的设计主要是应用程序的设计。

可将数据库设计分为 6 个阶段：需求分析，概念结构设计，逻辑结构设计，物理结构设计，数据库实施，数据库运行和维护。数据库设计的各阶段可以和软件工程的各阶段对应起来，软件工程的某些方法和工具同样可以适用于数据库工程。数据库工程和传统的软件工程的区别在于：软件工程中比较强调行为特性的设计；在数据库工程中，由于数据库模型是一个相对稳定的并为所有用户共享的数据基础，所以数据库工程中更强调对于结构特性的设计，并与行为特性的设计结合起来。

下面对数据库设计各个阶段的任务和方法做一个简单介绍。

1. 需求分析

需求分析阶段的任务是：对现实世界要处理的对象(组织、部门、企业等)进行详细调查，在了解现行系统的概况、确定新系统功能的过程中，收集支持系统目标的基础数据及其处理方法。需求分析是在用户调查的基础上，通过分析逐步明确用户对系统的需求，包括数据需求和围绕这些数据的业务处理需求。

调查的重点是“数据”和“处理”。通过调查要从用户中获得对数据库的下列需求：

- **信息需求。**信息需求定义未来信息系统用到的所有信息，弄清用户将向数据库输入什么样的数据，从数据库中要求获得什么样的内容，将要输出什么样的信息。即在数据库中需存储哪些数据、对这些数据将作如何处理等。要能描述数据间本质上和概念上的联系，描述信息的内容和结构，以及信息之间的联系等性质。

- **处理需求。**处理需求定义未来系统数据处理的操作功能，描述操作的优先次序，包括操作执行的频率和场合，操作与数据之间的联系。处理需求还包括弄清用户要完成什么样的处理功能、每种处理的执行频度、用户要求的响应时间，以及处理的方式是联机处理还是批处理等等。同时也定义了安全性和完整性的约束。

在需求分析中，通过自顶向下、逐步分解的方法分析系统。分析的结果用数据流图 DFD(data flow diagram)进行图形化的描述。通过数据流图，可以清晰地表达系统中的功能要求和数据流向。而且，数据流图可以是分层次的，系统的整体功能要求可以分解为系统的若干子功能要求，通过逐步分解的方法，一直可以分解到将系统的工作过程表达清楚为止。在功能分解的同时，每个功能在处理中所用的数据存储也在逐步分解，从而形成若干层次的数据流图。

除数据流图外，还采用一些规范表格对于数据分析的结果描述做补充描述。一般有数据清单(数据元素表)，业务活动清单(事务处理表)，完整性及一致性要求，响应时间要求，预期变化的影响等。它们是数据字典的雏形，具体内容主要包括：

- **数据项。**它是数据的最小单位，包括项名、含义、别名、类型、长度、取值范围及与其他项的逻辑联系等；

- **数据结构。**是若干数据项的有序集合，包括数据结构名、含义、组成的成分等。

- **数据流说明。**可以是数据项，也可以是数据结构。表示某一加工的输入输出数据；包括数据流名、说明、流入的加工名、流出的加工名、组成的成分等。

- **数据存储说明。**加工中需要存储的数据，包括数据存储名、说明、输入数据流、输出数据流、组成的成分、数据量、存储方式、操作方式等。

- **加工过程。**包括加工名、加工的简要说明、输入输出数据流等。

需求分析的阶段成果是产生系统需求说明书，系统需求说明书主要包括数据流图、数据字典的雏形表格、各类数据的统计表格、系统功能结构图等。

2. 概念结构设计

数据库概念结构设计任务是产生反映企业信息需求的数据库概念结构，即概念模型。概念模型是不依赖于计算机系统和具体的 DBMS 的。

概念模型应具备以下特点：

• **有丰富的语义表达能力。**能表达用户的各种需求，包括描述现实世界中各种事物及事物之间的联系，能满足及用户对数据的处理要求。

• **易于交流和理解。**概念模型是 DBA, 应用系统开发人员和用户之间的主要交流工具。

• **易于变动。**概念模型要能灵活地加以改变，以反映用户需求和环境的变化。

• **易于向各种数据模型转换，易于从概念模型导出与 DBMS 有关的逻辑模型。**

设计概念结构的策略有如下几种：

• **自顶向下。**首先定义全局概念结构的框架，再作逐步细化。

• **自底向上。**首先定义每一局部应用的概念结构，然后按一定的规则把它们集成，从而得到全局概念结构。

• **由里向外。**首先定义最重要的那些核心结构，再逐渐向外扩充。

• **混合策略。**把自顶向下和自底向上结合起来。自顶向下设计一个概念结构的框架，然后以它为骨架再自底向上设计局部概念结构，并把它们集成。

最常用的设计策略是自底向上设计策略。

设计数据库概念模型的最著名、最常用的方法是 P. P. S. chen 于 1996 一年提出的“实体—联系方法”(Entity-Relationship Approach)，简称 E-R 方法。它采用 E-R 模型将现实世界的信息结构统一用实体、属性以及实体之间的联系来描述。

实体是客观存在并可互相区分的“事物”。实体必须有一组表征其特征的“属性”来描述。

属性与实体无截然划分的界限，描述另一事物的某一特征的、而且其本身在一定意义上说是不再需要描述的事物一般归为属性。在设计时可以归为属性的事物尽可能归为属性，以简化 E-R 图的处理，但也要根据需求而定。

例如，工种通常为职工的属性，但要涉及劳保部门时，它就成为一个实体，可用一些属性来描述，

工种：工种号，工种名，工种类型，工种补贴，保健费…

联系是指实体之间存在的对应关系，一般可分为一对一的联系(1:1)，一对多的联系(1:n)和多对多的联系(m:n)。

联系也可以有属性，例如学生选修课程的联系，可以有属性“成绩”。

在实体—联系方法中用 E-R 图直观地表示 E-R 模型。在 E-R 图中，用长方形表示实体，用椭圆形表示属性，用菱形表示联系。在图形内标识它们的名字，它们之间用无向线段相连，表示联系的线段上标明是哪种联系。

采用 E-R 方法的数据库概念结构设计可分为三步进行。

(1)设计局部 E-R 模型。局部 E-R 模型的设计内容包括确定局部 E-R 结构的范围、定义属性、定义实体、定义联系等。

(2)设计全局 E-R 模型。这一步是将所有局部的 E-R 图集成为全局的 E-R 图，即全局的概念模型。

把局部 E-R 图集成为全局 E-R 图时，可以采用一次将所有的局部 E-R 图集成在一起的方式，也可以采用逐步集成进行累加的方式，一次只集成两个局部 E-R 图，这样复杂度较低。

当将局部的 E-R 图集成为全局的 E-R 图时，可能存在 3 类冲突。

• **属性冲突：**包括类型、取值范围、取值单位的冲突。

• **结构冲突：**例如同一对象在一个局部 E-R 图中作为实体，而在另一个局部 E-R 图中作为属性，同一实体在不同的 E-R 图中属性个数和类型不同等。

• **命名冲突：**包括实体类型名、联系类型名之间异名同义，或同名异义等。

属性冲突和命名冲突通常用讨论、协商等行政手段解决；结构冲突则要认真分析后用技术手段解决，例如把实体变换为属性或属性变换为实体，使同一对象具有相同的抽象；又如，取同一实体在各局部 E-R 图中属性的并作为集成后该实体的属性集，并对属性的取值类型进行协调统一。

(3)全局 E-R 模型的优化。一个好的全局 ER 模式除能反映用户功能需求外，还应满足下列条件：实体类型个数尽可能少，实体类型所含属性尽可能少，实体类型间联系无冗余。优化就是要达到这 3 个目的，即相关实体类型的合并，一般把具有相同码的实体类型进行合并，还可以考虑将 1:1 联系的两个实体类型合并为一个实体类型；冗余属性的消除；冗余联系的消除。但要注意效率，根据具体情况可存在适当冗余。图 9. 13 是两个局部的 E-R 图集成的全局 E-R 图的举例。

3.逻辑结构设计

逻辑结构设计的目的是从概念模型导出特定的 DBMS 可以处理的数据库的逻辑结构(数据库的模式和外模式)，这些模式在功能、性能、完整性和一致性约束及数据库可扩充性等方面均应满足用户提出的要求。

特定的 DBMS 可以支持的数据模型包括层次模型、网状模型、关系模型、面向对象模型等。下面我们仅对概念模型向关系模型的转换进行讨论。

E-R 模型向关系模型转换的规则是：

- 一个实体类型转换成一个关系模式，实体的属性就是关系的属性，实体的码就是关系的码。

对于实体之间联系的转换则有以下不同的情况 ((2) — (5))：

- 一个 1, 1 联系可以转换为一个独立的关系模式，也可以与联系的任意一端实体所对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，每个实体的码均是该关系的候选码。如果与联系的任意一端实体所对应的关系模式合并，则需要在该关系模式的属性中加入另一个实体的码和联系本身的属性。

- 一个 1:n 联系可以转换为一个独立的关系模式，也可以与联系的 n 端实体所对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而联系的码为 n 端实体的码。如果与联系的 n 端实体所对应的关系模式合并，则需要在该关系模式的属性中加入 1 端实体的码和联系本身的属性。

- 一个 m:n 联系转换为一个关系模式。与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为各实体码的组合。

- 3 个或 3 个以上的实体间的多元联系转换为一个关系模式。与该多元联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为各实体码的组合。

具有相同码的关系模式可合并。

转换得到的关系模式需要根据规范化理论进行规范化处理。规范化过程实际上是一个关系模式分解的过程，对于转换得到的规范化程度较低的关系模式进行分解，以达到所要求的更高的规范化程度。关于数据库设计过程中规范化理论的应用在 9. 2. 7 节还要进行讨论。

4. 物理结构设计

数据库的物理设计是对已确定的逻辑数据库结构，利用 DBMS 所提供的方法、技术，以较优的存储结构和数据存取路径、合理的数据存放位置以及存储分配，设计出一个高效的、可实现的物理数据库结构。

物理设计常常包括某些操作约束，如响应时间与存储要求等。

由于不同的 DBMS 所提供的硬件环境和存储结构、存取方法不同，提供给数据库设计人员的系统参数及其变化范围不同，因此物理结构设计没有一个放之四海而皆准的原则，只能提供一些技术和方法供参考。

(1) 存储记录的格式设计。

对数据项类型特征作分析，对存储记录进行格式化，决定如何进行数据压缩或代码化。使用“记录的垂直分割”方法，对含有较多属性的关系，按其中属性的使用频率不同进行分割；或使用“记录的水平分割”方法，对含有较多记录的关系，按某些条件进行分割。并把它们定义在相同或不同类型的物理设备上，或在同一设备的不同区域上，从而使访问数据库的代价最小，提高数据库的性能。

(2) 存储方法设计。

物理设计中最最重要的一个考虑是把存储记录在全范围内进行物理安排。包括：

- **顺序存放**，平均查询次数为关系的记录个数的二分之一。
- **散列存放**，查询次数由散列算法决定。
- **聚簇(cluster)存放**，“记录聚簇”是指将不同类型的记录分配到相同的物理区域中去，充分利用物理顺序性优点，提高访问速度。使经常在一起使用的记录聚簇在一起，以减少物理 I/O 次数。

(3) 存取方法设计。

存取方法设计为存储在物理设备上的数据提供数据访问的路径。索引是数据库中一种非常重要的数据存取路径。在存取方法设计中要确定建立何种索引，以及在哪些表和属性上建立索引。

通常情况下，对于数据量很大，又需要做频繁的查询的表建立索引，并且选择将索引建立在经常用做

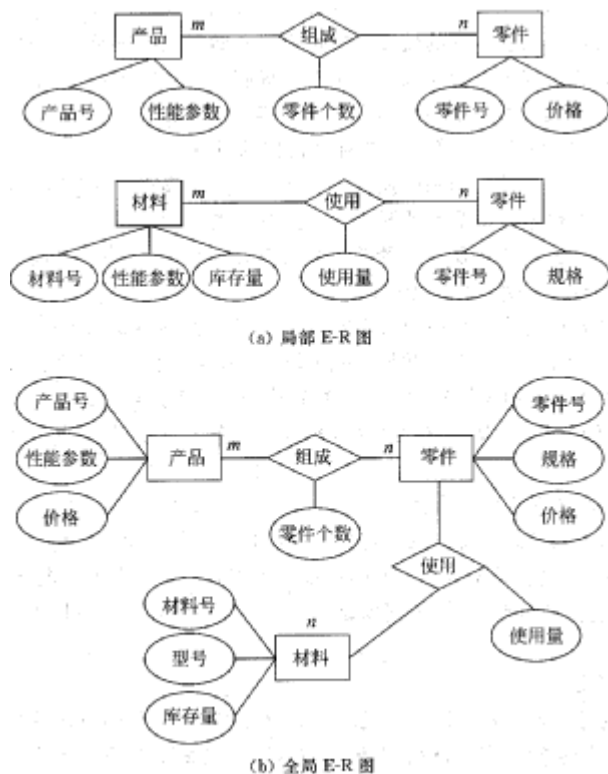


图 9.13 集成局部 E-R 模型为全局 E-R 模型

查询条件的属性或属性组，以及经常用做连接属性的属性或属性组上。

5.数据库实施

数据库实施阶段指的是基于数据库逻辑结构设计和物理结构设计的结果，在计算机上建立起实际数据库结构，装入数据，并进行测试和试运行的过程。该阶段的主要工作如下：

- **建立实际的数据库结构。**
- **装入试验数据对应用程序进行测试，以确认其功能和性能是否满足设计要求，并检查对空间的占有情况。**
- **装入实际数据，即数据库加载，建立起实际的数据库。**

在这个阶段中还要进行一些其他工作，包括加强数据库的安全性、完整性控制，及保证一致性、可恢复性等。这些总是以牺牲效率为代价的，设计人员的任务就是要在实现代价和尽可能多的功能之间进行合理平衡。

6.数据库运行和维护

数据库投入正式运行，标志着数据库设计和应用开发工作的结束和运行维护阶段的开始。在数据库运行阶段，对数据库经常性的维护工作主要由 DBA 负责，主要工作包括：

- **数据库的转储和恢复。**DBA 要针对不同的应用要求制定不同的转储计划，以保证一旦发生故障能尽快将数据库恢复到某种一致状态，并尽可能减少对数据库的破坏。

- **数据库的安全性、完整性控制。**DBA 要根据数据库系统运行过程中实际情况的变化修改安全性控制，及时调整授权和密码等。在数据库系统运行的过程中，数据完整性约束条件也会发生变化，也需要 DBA 不断修正，以满足用户要求。

- **数据库性能的监督、分析和改造。**目前有些 DBMS 产品提供了监测系统性能参数的工具，DBA 可以利用这些工具方便地得到系统运行过程中一系列性能参数的值。通过分析这些数据 DBA 可以判断当前系统运行状况是否最佳，并作出相应的参数调整。

- **数据库的重组织和重构造。**数据库系统运行一段时间之后，由于记录不断增、删、改，会使数据库的物理存储情况变化，数据库性能下降。这时需进行数据库重组织，按数据库的原设计要求重新安排存储位置，回收垃圾，减少指针链等，以提高系统性能。

由于数据库应用环境发生变化、增加了新的应用或新的实体、取消了某些应用、某些实体与实体间的联系发生了变化等原因，使原来的数据库设计不能满足新的需求，这时就需进行数据库重构造，调整数据库的逻辑结构和物理结构，以满足新的应用需求。

9.2.7 规范化理论在数据库设计中的应用

规范化理论是数据库设计的理论基础，它可以应用到数据库设计的不同阶段中。

在概念结构设计阶段，可以以规范化理论为指导“规范化”E-R 图中的实体。当进行概念结构设计，识别出所有的实体时，有可能在一个实体的属性间存在函数依赖。例如，假设“职工”实体中除其他属性外还包含属性“部门号”和“部门地址”，有一个函数依赖：部门号→部门地址。这样的实体如果转换为关系，则是一个未达到 3NF 的关系。可以在概念结构设计时对 E-R 图中的实体进行“规范化”，从“职工”实体中去掉“部门地址”属性；创建一个包含属性“部门号”、“部门地址”的“部门”实体；并建立一个“职工”实体与“部门”实体之间的联系。

函数依赖有助于我们检测到不好的 E-R 模型设计。如果在从 E-R 模型向关系转换时得到的关系不属于希望的范式，问题也许在 E-R 设计中。因此，如果我们在概念结构设计阶段引入规范化理论做指导，那么一般说来，在从 E-R 模型向关系转换时得到的关系将是规范化程度较高的。

当然规范化理论最主要的应用是在数据库逻辑结构设计阶段，当 E-R 模型向关系的转换完成后，逐一检查转换得到的各个关系模式，如果某些关系模式未到达应用所要求的规范化程度，则进行关系模式的分解。

前面总是强调为消除“不好”的关系模式的数据冗余等种种不良特性而对关系模式进行分解，提高关系模式的规范化程度，但有时候数据库设计者会希望要包含冗余信息的模式，即规范化程度较低的模式，目的是提高性能。例如，在银行应用中，规范化的模式是：

account(account-number, depositor-number, balance)

depositor(depositor-number, depositor-name, depositor-address)

其中 account 关系是关于储蓄账户的信息 depositor 关系是关于存款人的信息。假设每次查询储蓄账户的存款余额时，都希望同时查到存款人的姓名和地址，这就需要进行 account 和 depositor 两个关系的连接。

一个避免进行连接计算的方法是保存一个包含 account 和 depositor 的所有属性的关系

account-depositor : account-depositor(account-number , depositor-number , depositor-name , depositor-address, balance)。这使得显示账户信息更快。然而 account-depositor 关系的规范化程度只达到了 2NF。如果一个人持有多个账户, 他的姓名和地址就会重复存储多次, 当他的地址变更时, 就必须更新所有的副本。把一个规范化的模式变成非规范化的过程称为解除规范化(denormalization), 适当地解除规范化可以提高对响应时间要求严格的系统的性能。保持高的规范化程度以避免数据冗余等毛病, 还是降低规范化程度而追求高查询性能, 这是一个设计权衡问题。

9.3 数据仓库与联机分析处理、数据挖掘

数据库应用可以被广义地划分为事务处理和决策支持。事务处理系统现在已得到了广泛的应用, 一些公司已经积累了大量由这类系统产生的信息。

例如, 公司数据库常常包含了大量关于客户和交易的信息。信息存储的规模可能高达几吉(10^9) 字节, 在大型零售连锁店时甚至达到几太(10^{12}) 字节。零售商的交易信息可能包括客户的姓名或标识(比如信用卡号)、购买的商品、支付的金额以及购买日期。所购买的商品信息可能包括该商品的名称、生产商、型号、颜色和大小。客户信息可能包括信贷历史、年薪、住址、年龄, 甚至教育背景等。

这样的大型数据库可以作为制定商业决策的信息宝库, 譬如要决定进什么商品, 或者应打多少折扣。例如, 一家零售公司发现在北京地区突然盛行购买中式外衣, 意识到了这个购买趋势, 它可能就会开始在该区域的商店大量购进这种外衣。又如, 一家汽车公司在查询数据库时发现, 它的大多数小型运动汽车是由一些年薪超过 \$ 50 000 的年轻妇女购买的。该公司可能调整其市场定位, 从而吸引更多这类妇女购买它的小型运动型汽车。在这两个例子中, 公司识别出了客户行为的模式并利用该模式制定商业决策。

用于决策支持的数据的存储和检索将涉及以下领域:

- **联机分析处理(OLAP)**。尽管许多决策支持查询可以用 SQL 书写, 但另一些则无法用 SQL 表示或无法用 SQL 简便地表示。因此人们提出了一些 SQL 的扩展以利于更方便地进行数据分析。联机分析处理(OLAP) 领域涉及用于数据分析的工具和技术, 在即使数据库可能相当大的条件下, 仍能对汇总数据的查询请求给出几乎是即时的答复。

- **数据仓库**。大型公司在实际运作中拥有形式多样的数据源, 这些数据源中的数据都可用于做出商业决策。这些数据源可能将数据按不同的模式存储。出于性能的考虑(也可能出于管理控制的考虑), 某个部门的数据源通常不允许公司其他部门按照自己的需求进行数据检索。为了能够基于来自多个数据源的数据进行高效的决策支持查询, 一些公司已创建了数据仓库。数据仓库系统使用统一的模式从多个数据源中收集数据, 并对数据单独存放。因此, 数据仓库提供给用户一个统一的用于决策支持的数据接口。

- **数据挖掘**。知识发现技术试图自动发现数据中的统计规则和模式。数据挖掘领域将人工智能研究人员提出的知识发现技术和统计分析结合起来, 同时采用了有效的实现技术使他们能够在超大型数据库中使用。

9.3.1 OLAP 系统与 OLTP 系统的比较

如前所述, 数据库应用可以被广义地划分为事务处理和决策支持。传统的数据库系统作为数据管理手段, 主要用于事务处理。在这些数据库中已经保存了大量的日常业务数据。传统的决策支持系统 DSS(decision support system)一般是直接建立在这种事务处理环境上的。数据库技术一直力图使自己能胜任从事务处理、批处理到分析处理的各种类型的信息处理任务。尽管数据库在事务处理方面的应用获得了巨大的成功, 但它对分析处理的支持一直不能令人满意, 尤其是当以业务处理为主的联机事务处理 OLTP(on_line transaction processing) 应用与以分析处理为主的 DSS 应用共存于同一个数据库系统中时, 这两种类型的处理发生了明显的冲突。人们逐渐认识到, 事务处理和分析处理具有极不相同的性质, 直接使用事务处理环境来支持 DSS 是行不通的。

联机分析处理 OLAP(on_line analytical processing)是专门为支持复杂的分析操作而设计的, 侧重于对决策人员和高层管理人员的决策支持, 可以应分析人员的要求快速、灵活地进行大数据量的复杂查询处理, 并且以一种直观易懂的形式将查询结果提供给决策人员, 以便他们准确掌握企业的经营状况, 了解市场需求, 制定正确方案, 增加效益。

下面概述 OLAP 系统与 OLTP 系统的主要区别。

- **所面向的用户和系统**: OLTP 是面向客户的, 由职员、信息技术专业人员或客户进行事务处理或查询处理。OLAP 是面向市场的, 由经理、主管和分析人员进行数据分析和决策制定。

- **数据内容**: OLTP 系统管理当前数据, 这些数据通常很琐碎, 难以用于决策。OLAP 系统管理大量历史数据, 提供汇总和聚集机制, 并在不同的粒度级别上存储和管理信息, 这些特点使得数据适合于决策分析。

• **数据库设计:** 通常, OLTP 系统采用 E-R 模型和面向应用的数据库设计。而 OLAP 系统通常采用星型模式或雪花模式(9. 3. 3 节中介绍)和面向主题的数据库设计。

• **视图:** OLTP 系统主要关注一个企业或部门内部的当前数据, 而不涉及历史数据或不同组织的数据。与之相反, OLAP 系统常常跨越一个企业的数据库模式的多个版本, OLAP 系统也处理来自不同组织的信息, 由多个数据源集成的信息。

• **访问模式:** OLTP 系统的访问主要由短的原子事务组成, 这种系统需要并发控制和恢复机制。而对 OLAP 系统的访问大部分是只读操作, 其中大部分是复杂查询。

OLTP 和 OLAP 的其他区别还包括数据库大小、操作的频度、性能度量等。这些都概括在表 9. 5 中。

9. 3. 2 多维数据模型

数据仓库和 OLAP 操作都基于多维数据模型。下面介绍多维数据模型的基本概念。

1.度量属性(measure attribute)

度量属性是决策者所关心的具有实际意义的数量。例如: 商品的销售量、仓库中物品的库存量等。这些属性测量了某个值, 可以对它们进行统计, 聚集操作等。

2.维属性(dimension attribute)

维是人们观察度量属性和度量属性的汇总的特定角度。例如, 企业常常关心产品销售数据随着时间推移而产生的变化情况, 这时他是从时间的角度来观察产品的销售, 所以时间就是一个维(时间维)。企业也时常关心自己的产品在不同地区的销售分布情况, 这时他是从地理分布的角度来观察产品的销售, 所以地理分布也是一个维(地理维)。

3.维的概念分层(concept hierarchy of a dimension)

人们观察数据的某个特定角度(即某个维)还可以存在细节程度不同的多个描述方面, 称这多个描述方面为维的概念分层。一个维往往具有多个层次, 例如描述时间维时, 可以从日期、月份、季度、年等不同层次来描述, 那么日期、月份、季度、年等就是时间维的层次; 同样: 城市、地区、国家等构成了一个地理维的多个层次。

4.多维数据(multidimensional data)

能够模式化为维属性和度量属性的数据统称为多维数据。图 9. 14 是多维数据的一个例子。

5.数据立方体(data cube)

数据的多维视图称作数据立方体。图 9. 15 显示美国、加拿大、墨西哥几个国家在一年的各个季度中 TV, PC, VCR 几种家电产品的销售量的数据立方体。

6.方体和数据立方体(cuboid and data cube)

在一些文献中, 将图 9. 15 所示的那样的立方体称作方体。给定若干个维, 可以构造出方体的格, 其中的每一个方体都表示数据在不同的概括层次上的汇总。在这儿, 将方体的格称作数据立方体。图 9. 16 所示是一个数据立方体和对应的各个层次的方体。最底层的方体称作基本方体, 它们代表最低的概括层次。最顶层的方体称作顶点方体, 它代表最高的概括层次, 即求所有数据的总和。

9.3.3 数据仓库

一些大公司可能位于许多地点, 包括许多部门, 每个地点每个部门都可能产生大量的数据。例如, 大型零售连锁店可能拥有成百上千的零售店, 保险公司可能拥有成千的各地分支机构, 在这些连锁店和分支机构中积累了大量的数据。而且, 大型组织有复杂的内部组织结构, 因此不同的数据在不同的操作系统管理之下, 或者具有不同的数据模式。例如, 生产管理数据和顾客意见数据可能存储在不同地点、不同的数据库系统中。企业决策者需要存取来

表 9.5 联机事务处理(OLTP)与联机分析处理(OLAP)的区别

特 性	OLTP	OLAP
特征	操作处理	信息处理
面向	事务	分析
用户	职员、DBA、数据库专业人员	经理、主管、分析人员
功能	日常操作	长期信息需求、决策支持
数据库设计	基于 E-R, 面向应用	星型/雪花, 面向主题
数据	当前的, 确保最新	历史的, 跨时间维护数据
汇总	原始的, 高度详细	汇总的, 统一的
视图	详细的, 平面关系	汇总的, 多维的
工作单位	短的, 简单事务	复杂查询
存取	读/写	大多为读
关注	数据进入	信息输出
操作	主码上的索引/散列	大量扫描
访问记录数	数十个	数百万
用户数	数千个	数百个
数据库规模	100MB 到 GB 级	100GB 到 TB 级
优先	高性能, 高可用性	高灵活性, 最终用户自主权
度量	事务吞吐量	查询吞吐量, 响应时间

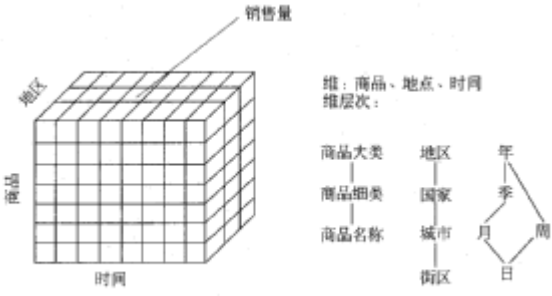


图 9.14 多维数据示例

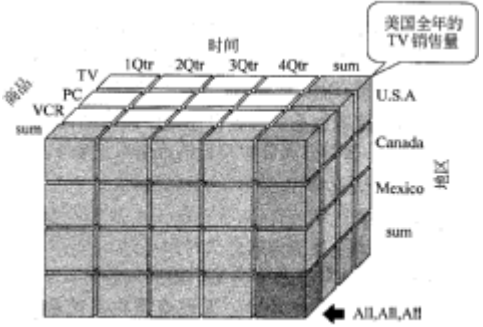


图 9.15 一个数据立方体

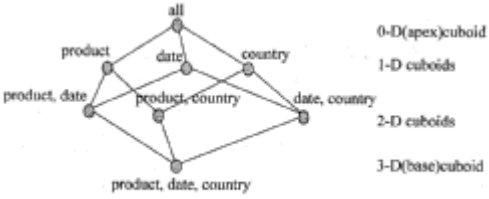


图 9.16 数据立方体和对应的方体

自所有这样的资源的信息。对各个资源分别建立查询既麻烦又低效。而且，数据源一般只存储当前数据，而决策者可能还需要访问历史数据，例如，对于大型零售连锁店的管理者来说，有关在过去的几年里客户购买模式是如何改变的信息可能非常重要。如何有效地利用企业中分散的数据源中积累的大量数据来支持分析决策呢？数据仓库对这样的问题提供了一个解决方案。

9.3.3.1 数据仓库基本概念

数据仓库是一个用以更好地支持企业或组织的决策分析处理的、面向主题的、集成的、相对稳定的、体现历史变化的数据集。下面着重讨论数据仓库数据的 4 个基本特征。

1. 数据仓库是面向主题的

数据仓库中的数据是面向主题进行组织的。什么是主题呢？主题是一个抽象的概念，是在较高层次上将企业信息系统中的数据综合、归类并进行分析利用的抽象。在逻辑意义上，它是对应企业中某一宏观分析领域所涉及的分析对象。面向主题的数据组织方式，就是在较高层次上对分析对象的数据的一个完整、一致的描述，能完整、统一地刻画各个分析对象所涉及的企业的各项数据，以及数据之间的联系。所谓较高层次是相对面向应用的数据组织方式而言的，是指按照主题进行数据组织的方式具有更高的数据抽象级别。例如，在一个采用“会员制”经营的商场信息系统中，用于宏观分析的主题包括供应商、商品、顾客等。

2. 数据仓库的数据是集成的

数据仓库的数据是从原有的分散的数据库数据中抽取来的。操作型数据与分析型数据之间差别甚大。第一，数据仓库的每一个主题所对应的源数据在原有的各分散数据库中有许多重复和不一致的地方，且来源于不同的联机系统的数据都和不同的应用逻辑捆绑在一起；第二，数据仓库中的综合数据不能从原有的数据库系统直接得到。因此在数据进入数据仓库之前，必然要经过统一与综合，这一步是数据仓库建设中最关键、最复杂的一步。下面我们还要进一步讨论。

3. 数据仓库的数据是相对稳定的

数据仓库的数据主要供决策分析之用，所涉及的数据操作主要是数据查询，一般情况下并不进行修改操作。数据仓库的数据反映的是一段相当长的时间内历史数据的内容，是不同时间点的数据库快照的集合，以及基于这些快照进行统计、综合和重组的导出数据，而不是联机处理的数据。数据库中进行联机处理的数据经过集成输入到数据仓库中。因为数据仓库只进行数据查询操作，所以数据仓库管理系统 DWMS 相比 DBMS 而言要简单得多。DBMS 中许多技术难点，如完整性保护、并发控制等等，在数据仓库的管理中几乎可以省去。但是由于数据仓库的查询数据量往往很大，所以就对数据查询提出了更高的要求，它要求采用各种复杂的索引技术；同时由于数据仓库面向的是企业的高层管理者，他们会对数据查询的界面友好性和数据表示提出更高的要求。

4. 数据仓库数据是反映历史变化的

数据仓库中的数据相对稳定是针对应用来说的，也就是说，数据仓库的用户进行分析处理时是不进行数据更新操作的。但并不是说，在从数据集成输入数据仓库开始到最终被删除的整个数据生存周期中，所有的数据仓库数据都是永远不变的。

数据仓库的数据是反映历史变化的，这主要表现在以下 3 方面：

- **数据仓库随时间变化不断增加新的数据内容。**数据仓库系统必须不断捕捉 OLTP 数据库中变化的数据，追加到数据仓库中去，也就是要不断地生成 OLTP 数据库的快照，经统一集成后增加到数据仓库中去；但对于每次的数据库快照确实是不再变化的，捕捉到新的变化数据，只不过又生成一个数据库的快照增加进去，而不会对原来的数据库快照进行修改。

- **数据仓库随时间变化不断删去旧的数据内容。**数据仓库的数据也有存储期限，一旦超过了这一期限，过期数据就要被删除。只是数据仓库内的数据时限要远远长于操作型环境中的数据时限。在操作型环境中一般只保存有 60-90 天的数据，而在数据仓库中则需要保存较长时限的数据（如 5-10 年），以适应 DSS 进行趋势分析的要求。

- **数据仓库中包含大量的综合数据，这些综合数据中很多跟时间有关，**如数据经常按照时间段进行综合，或隔一定的时间片进行抽样等等。这些数据要随着时间的变化不断地进行重新综合。

9.3.3.2 数据仓库的数据模式

典型的数据仓库具有为数据分析而设计的模式，使用 OLAP 工具进行联机分析处理。因此，数据通常是多维数据，包括维属性和度量属性。包含多维数据的表称作事实表，通常很大。例如，一个表 sales 记录了零售商店的销售信息，其中每个元组对应一个商品售出记录，这是事实表的一个典型例子。表 sales 的维包括售出的是何种商品（通常是一个商品标识，比如条形码中使用的标识）、商品售出日期、商品售出地点、哪个顾客购买该商品等等。度量属性可能包括售出商品数量及销售金额。

为了减小存储需求, 维属性通常是一些短的标识, 作为参照其他表(维表)的外码。例如, 某个事实表 sales 可能含有属性 item-key, time-key; branch-key 和 location-key, 以及度量属性 units-sold 和 dollars-sold。属性 item-key 是一个参照维表 item 的外码, 表 item 含有关于商品的一些信息, 比如商品名称、商品的品牌、商品所属类别等。属性 time-key 是一个参照维表 time 的外码, 表 time 给出了每个日期的月、季和年的信息。属性 branch-key 是一个参照维表 branch 的外码, 表 branch 含有关于出售商品的分销商的其他属性, 比如分销商的名称、分销商的类型等。属性 location-key 是一个参照维表 location 的外码, 表 location 含有关于销售地点的街道、城市、省份、国家等属性。

由此得到的模式如图 9.17 所示。这样的具有一个事实表、多个维表以及从事实表到维表的参照外码的模式称为星型模式。

更复杂的数据仓库设计可能含有多级维表, 例如维表 item 可能含有属性 supplier-key, 作为参照给出商品供应商的细节信息的另一个维表 supplier 的外码; 维表 location 可能含有属性 city-key, 作为参照给出城市的细节信息的另一个维表 city 的外码。这种模式称作雪花模式。图 9.18 给出了一个数据仓库的雪花模式示意图。

复杂的数据仓库设计也可能含有不止一个的事实表。例如图 9.19 所示的数据仓库模式中含有 Sales 和 Shipping 两个事实表, 它们共享 item, time, branch, location 等维表。这样的模式称作事实星座模式。

9.3.3 数据仓库体系结构

数据仓库系统通常采用 3 层的体系结构, 如图 9.20 所示。底层为数据仓库服务器, 中间层为 OLAP 服务器, 顶层为前端工具。

底层的数据仓库服务器几乎总是一个关系数据库系统。数据仓库服务器从操作型数据库或外部数据源(例如由外部咨询机构提供的客户背景信息)提取数据, 对数据进行清理、转换、集成等, 并装入到数据仓库中。

中间层 OLAP 服务器的实现可以是关系型 OLAP (ROLAP), 即扩充的关系型 DBMS, 提供对多维数据的支持; 也可以是多维 OLAP (MOLAP), 它是一种特殊的服务器, 直接支持多维数据的存储和操作。

顶层的前端工具包括查询和报表工具、分析工具、数据挖掘工具等。

从结构的角度看, 有三种数据仓库模型: 企业仓库、数据集市和虚拟仓库。

企业仓库(enterprise warehouse)收集跨越整个企业的各个主题的所有信息。它提供全企业范围的数据集成, 数据通常来自多个操作型数据库和外部信息提供者, 并且是跨多个功能范围的。它通常包含详细数据和汇总数据。企业数据仓库可以在传统的大型机上实现, 例如 UNIX 超级服务器或并行结构平台。它需要广泛的业务建模, 可能需要多年的时间来设计和建造。

数据集市(data mart)包含对特定用户有用的、企业范围数据的一个子集。

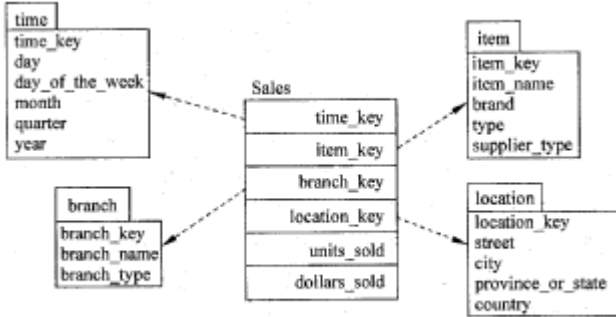


图 9.17 数据仓库的星型模式示意图

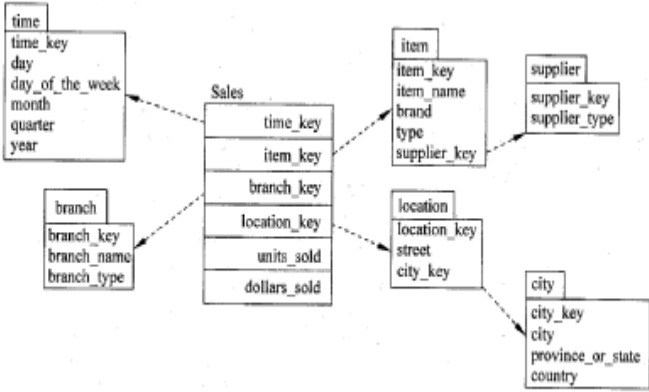


图 9.18 数据仓库的雪花模式示意图

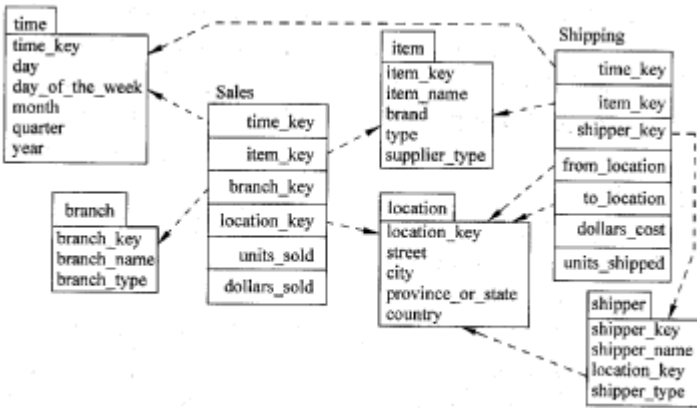


图 9.19 数据仓库的事实星座模式示意图

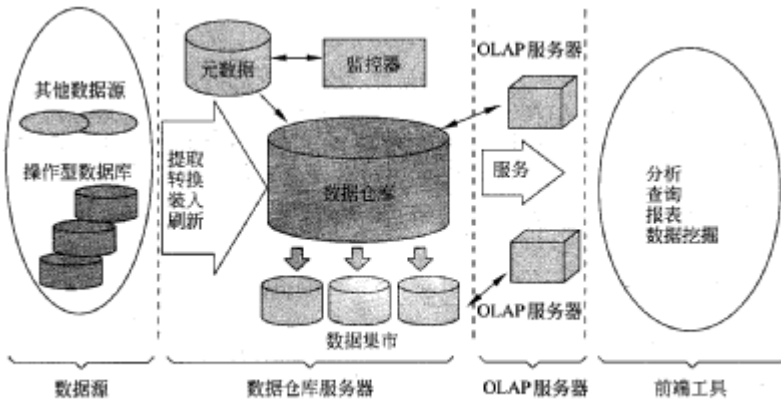


图 9.20 数据仓库体系结构

它的范围限于选定的主题，例如一个商场的数据集市可能限定它的主题为顾客、商品和销售。包括在数据集市中的数据通常是汇总的。

通常数据集市可以在低价格的部门服务器上实现，基于 UNIX 或 Windows/NT, 实现数据集市的周期一般是数周，而不是数月或数年。然而，如果它的规划不是企业范围的，从长远讲，可能会涉及很复杂的集成。根据数据的来源不同，数据集市分为独立的和依赖的两类。在独立的数据集中，数据来自一个或多个操作型数据库或外部信息提供者，或者是一个特定的部门或地区本地产生的数据。在依赖的数据集中，数据直接来自企业数据仓库。

虚拟仓库(virtual warehouse)是操作型数据库上视图的集合。为了有效地处理查询，只有一些可能的汇总视图被物化。虚拟仓库易于建立，但需要操作型数据库服务器具有剩余能力。

9.3.3.4 数据仓库系统的开发

开发企业的数据仓库是一项庞大的工程。一种方法是自顶向下地开发，从全面设计整个企业的数据仓库模型开始。这是一种系统的解决方法，并能最大限度地减少集成问题。然而，它费用高，费时长，并且缺乏灵活性，因为整个企业的共同数据仓库模型要达到一致是很困难的。另一种方法是自底向上地开发，从设计和实现各个独立的数据集市开始。这种方法花费低，灵活性高，并能快速回报投资。然而，将分散的数据集市集成起来，形成一个一致的企业仓库可能很困难。

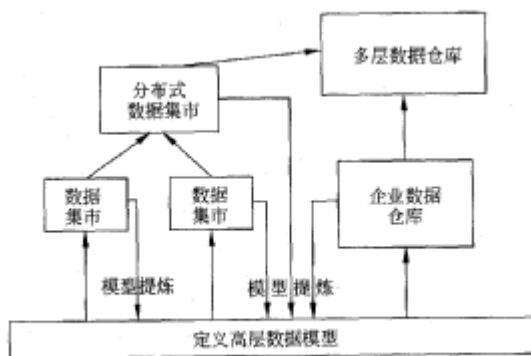


图 9.21 推荐的数据仓库开发方法

对于数据仓库系统的开发，一般推荐采用增量的、演进的方式，如图 9.21 所示。首先，在一个合理的短时间内(例如，一两个月)定义一个高层次的企业数据模型，在不同的主题和可能的应用之间，提供企业范围的、一致的、集成的数据视图。这个高层模型将大大减少以后的集成问题，尽管在企业数据仓库和部门数据集市的开发中，它还需要进一步提炼。然后，基于上述企业数据模型，可以并行地实现各自独立的数据集市和企业数据仓库。然后，再可以构造分布式数据集市，对不同的数据集市进行集成。最后，可以构造一个多层数据仓库。在这个多层数据仓库中，企业数据仓库是所有数据仓库数据的全权管理者，而这些数据分布在各个相关的数据集市中。

下面讨论创建一个数据仓库时的一些关键问题。

- **何时如何收集数据。**数据收集可以是资源驱动的，也可以是目的地驱动的。在资源驱动的数据收集架构中，数据源连续地(交易处理发生时)或周期地(例如，每天晚上)传输新的数据到数据仓库中。在目的地驱动的数据收集架构中，数据仓库周期地给数据源发送需要新的数据的请求。

除非资源更新通过两阶段提交的方式在数据仓库里复制下来，否则数据仓库不可能与资源保持同步更新。两阶段提交十分昂贵，所以数据仓库具有含有少许过期数据的特点，这对于决策支持系统通常不是问题。

- **使用何种模式。**曾经是单独构造的数据资源很有可能具有不同的模式。事实上，它们甚至可以使用不同的数据模型。数据仓库的部分任务就是执行模式整合，将数据转化成整合后的模式后再存储。因此，存储在数据仓库中的数据可看作是一个数据源数据的实体化视图，而不单单是数据源数据的简单拷贝。系统通常采用前面介绍过的星型模式或雪花模式。

- **数据清理。**对数据进行纠正和预处理的过程称作数据清理。数据源传送给数据仓库的数据往往具有一些不一致性的问题，这些不一致性是可以纠正的。例如，姓名经常拼写错误，住址可能有街道/区/城市名称的拼写错误，或者邮政编码输入错误。这可以通过参考每个城市中的街道名和邮政编码数据库将错误纠正到一个合理的程度。从多个数据源中收集到的地址列表可能含有重复项，在预处理中可以将重复项删除；一所住宅中的多个个人记录在预处理中可以被分为一组，这样一所住宅只需投递一封邮件就可以了；等等。

- **如何传播更新。**数据源中的关系更新必须传播到数据仓库。如果数据仓库中的关系与数据源中的完全一样，传播是直接的。否则，更新的传播问题基本上就是视图维护问题。

- **汇总何种数据。**事务处理系统产生的原始数据可能很大，无法在线存储。然而，许多问题只需通过维护由关系上的聚集得到的汇总数据即可得到解答，无需保存整个关系。例如，可以存储按商品名和类别汇总的服装销售额，而不是存储所有服装的销售数据。

9.3.4 联机分析处理的基本分析功能

联机分析处理系统是以数据库或数据仓库为基础的，它是一个交互式的系统，允许分析人员观察多维

数据的不同种类的汇总数据。联机的意思是指分析人员必须能够在线地请求新的汇总数据并在几秒钟时间内得到响应，无需为了看到查询结果被迫等待很长的时间。

联机分析处理系统包括以下的基本分析功能。

1. 上卷(roll-up)

在数据立方体中执行聚集操作，通过在维层次中上升或通过消除某个或某些维来观察更加概括的数据。

例如，图 9.22 所示的数据立方体经过沿着地点维的概念层次上卷，由城市上升到国家，得到图 9.23 所示的立方体。现在销售量数据不是按照城市分组聚集求值，而是按照国家分组聚集求值了。

也可以通过消除一个或多个维来观察更加概括的数据，例如，图 9.24 所示的二维立方体就是通过从图 9.22 所示的三维立方体消除“国家”维后得到的结果。现在所有国家的销售数据都累计在一起了。

2. 下钻(drill-down)

通过在维层次中下降或通过引入某个或某些新的维来观察更加细节的数据。

例如，图 9.22 所示的数据立方体经过沿着时间维的概念层次下钻，由季度下降到月，就得到了图 9.25 所示的立方体。现在销售量数据不是按照季度分组聚集求值，而是按照月分组聚集求值了。

3. 切片(slice)

在给定的数据立方体的一个维上进行选择操作，得到一个子立方体。

例如，在图 9.22 所示数据立方体的基础上，使用条件 time="Q1" 进行选择，相当于在原来的立方体中切出一片，结果如图 9.26 所示。

4. 切块(dice)

在给定的数据立方体的两个或更多个维上进行选择操作，得到一个子立方体。

例如，在图 9.22 所示数据立方体的基础上，使用条件(location="东京" or "京都") and(time="Q1" or "Q2") and(item="家电" or "食品") 进行选择，相当于在原来的立方体中切出一小块，结果如图 9.27 所示。

5. 转轴(pivot or rotate)

改变一个报告或页面显示的维方向，将一个三维立方体转变为一系列的二维平面等。例如，图 9.28 所示是图 9.26 的二维切片的“商品轴”和“地点轴”交换位置的结果。

9.3.5 数据挖掘

随着信息技术的普遍应用，人类获取数据的能力不断增强。据有关统计，全世界在业务管理、政府管理、科学与工程数据管理和其他应用领域方面所使用的数据库数以百万计。而且，随着数据库技术的发展，数据库的数量和规模还在迅速增加。然而，如何从大量的数据中及时有效地提取有用的信息，这几乎是所有经营管理者所面临的一个共同难题。为了解决这一课题，有关人员逐步研究开发了一系列的技术和方法，这就是数据库知识发现和数据挖掘技术，其目标就是要智能化和自动化地把数据转换为有用的信息和知识。一般认为，数据库中的知识发现是识别数据库中以前不知道的、新颖的、潜在有用的和最终可被理解

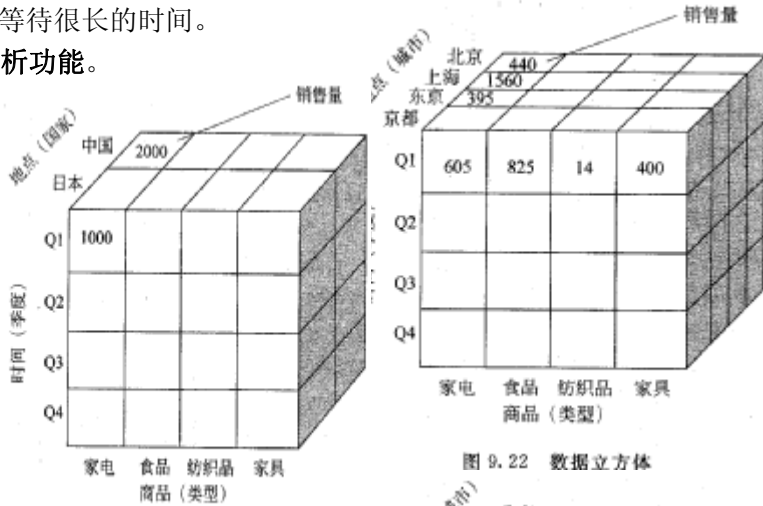


图 9.23 图 9.22 所示的数据立方体上卷的结果

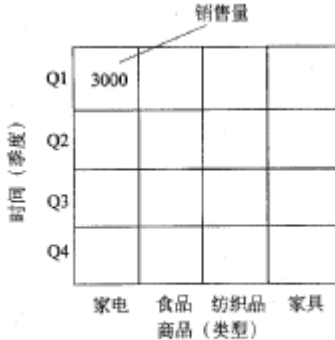


图 9.24 图 9.22 所示的数据立方体消除“国家”维的结果

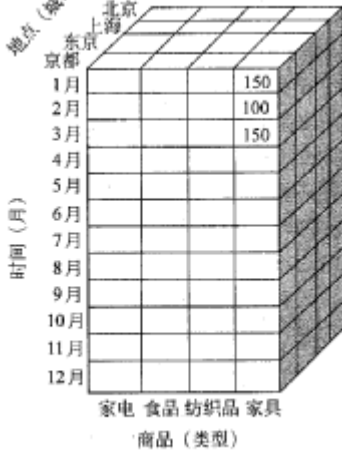


图 9.25 图 9.22 所示的数据立方体下钻的结果

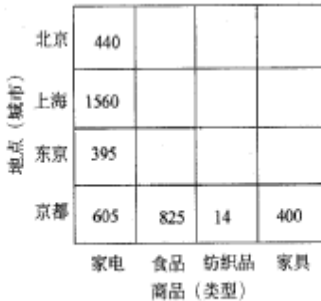


图 9.26 图 9.22 所示的数据立方体切片的结果

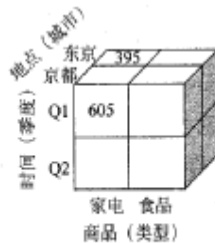


图 9.27 图 9.22 所示的数据立方体切块的结果

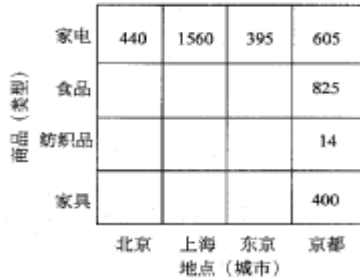


图 9.28 图 9.26 所示的切片转轴的结果

的模式非平凡过程，而数据挖掘是数据库知识发现过程的一个步骤。

有一个很普通却能说明数据挖掘如何产生效益的例子：美国加州某个超级连锁店通过数据挖掘，从记录着每天销售和顾客基本情况的数据库中发现，在下班后前来购买婴儿尿布的顾客多数是男性，他们往往也同时购买啤酒。于是这个连锁店的经理当机立断地重新布置了货架，把啤酒类商品布置在婴儿尿布货架附近，并在二者之间放上土豆片之类的佐酒小食品，同时把男士们需要的日常生活用品也就近布置。这样一来，上述几种商品的销量几乎马上成倍增长。通过上面的例子可以看出，数据挖掘能为决策者提供多么重要而有价值的信息或知识，从而产生不可估量的效益。

在数据库知识发现和数据挖掘过程中，可以从数据库或数据仓库的相关数据集中抽取知识或规律，并从不同的角度进行分析研究，所发现的知识可以运用到信息管理、查询处理、决策支持、过程控制等许多领域。现在，数据库知识发现与数据挖掘已经成为一个非常重要和非常活跃的研究领域，吸引了来自数据库系统、知识库系统、人工智能、机器学习、统计学、空间信息处理、数据可视化等许多领域的研究人员，进行跨学科、跨领域的综合研究。数据挖掘的过程如图 9.29 所示。

对数据挖掘技术的分类有多种角度。例如，可以按所挖掘的数据库的种类来分，分成关系型数据库的数据挖掘、数据仓库的数据挖掘、面向对象数据库的数据挖掘、空间数据库的数据挖掘、正文数据库和多媒体数据库的数据挖掘等。又如，可以按所发现的知识类别来分，数据挖掘可以发现多种类型的知识，包括关联规则、特性描述、分类分析、聚类分析、趋势和偏差分析等。此外，数据挖掘还可以按其发现的知识的抽象层次来分类，如一般化的知识、初级知识、多层次知识等。灵活的数据挖掘系统能够在多个层次上进行知识发现。

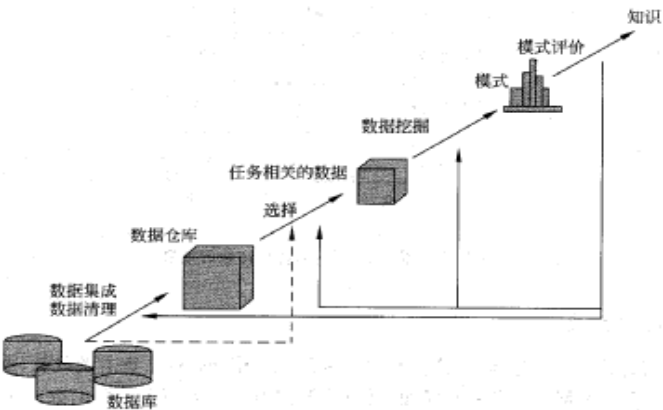


图 9.29 数据挖掘的过程

下面简单介绍比较常用的几种数据挖掘方法。

• **关联规则挖掘**：关联规则挖掘的典型问题是，给定一个销售交易的数据库，要求发现数据项之间的重要关联性，即在一个交易中出现某些数据项蕴含着其他一些数据项也可能会在同一交易中出现。例如前面所举的许多顾客在购买尿布的同时也购买啤酒的结论就是通过关联规则分析所得到的结果。关联规则分析是一个从现象到本质的揣测推理过程，也就是说，通过关联分析所得到的结果，仅仅是一种可能的因果关系，它能够协助业务专家对事物的本质进行分析，深化对事物关系的认识，但需要业务专家加以确认，并予以合理的解释，才能够成为对决策进行指导的规律。

• **特征描述**：数据库中通常存放大量的细节数据，然而，用户常常希望能够得到对于所关心的一类数据的简洁的概貌描述。特征描述是对目标类数据的一般特征或特性进行汇总，并以直观易理解的方式显示给用户。通常，用户首先通过数据库查询来对目标类数据进行查询，例如为研究上一年在某超市消费超过 \$1000 以上的顾客的特征，可以通过执行一个 SQL 查询收集关于这些产品的数据。特征描述通常采用的方法是进行数据概化，将庞大的任务相关的数据集从较低的概念层抽象到较高的概念层。例如，对于上述消费超过 \$1000 以上的顾客，特征描述的结果可能是顾客的一般轮廓，如年龄在 40-50 岁之间、已婚、有工作等。

天气	气温	湿度	刮风	类别
晴	热	高	否	N
晴	热	高	是	N
多云	热	高	否	P
雨	温暖	高	否	P
雨	冷	正常	否	P
雨	冷	正常	是	N
多云	冷	正常	是	P
晴	温暖	高	否	N
晴	冷	正常	否	P
雨	温暖	正常	否	P
晴	温暖	正常	是	P
多云	温暖	高	是	P
多云	热	正常	否	P
雨	温暖	高	是	N

图 9.30 将天气状况分为能否进行锻炼两个类的训练集

• **分类分析**：分类分析是找出数据集中各组对象的共同特征，并建立分类模型，从而能够将数据集中的其他对象分到不同的组中分类也称作制导的学习，为了建立分类模型，需要有一个用做训练集的示例数据库 E，它中的每个元组都有一个给定的类标识。分类过程是首先分析训练集中的数据，根据每个类中数据的特征为每个类生成分类模型，然后用得到的分类模型对未知类别的数据进行分类。表示分类模型的一种常用方法是决策树。例如，基于图 9.30 的训练集，通过分类分析得到图 9.31 所示的决策树。

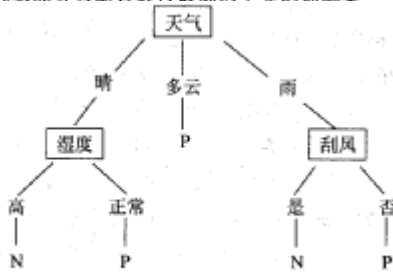


图 9.31 基于图 9.30 的训练集建立的决策树

• **聚类分析**：若干个相似的数据对象组合在一起称作一个聚簇。聚类分析是将数据集分割为若干个有意义的聚簇的过程。聚类分析也称作无制导的学习，因为聚类分析与分类分析不同，它不依赖于没有事先确定的类，也没有已具有类标识的训练集。好的聚类分析算法应该使得所得到的聚簇内的相似性很高，而不同的聚簇间的相似性很低。

数据库知识发现和数据挖掘已经有了许多研究成果，并有一些商品系统和实验系统。它们实现了通常的数据挖掘算法，并提供了一些方便的相关功能。业务人员可以结合自己的业务需求，通过使用这些数据挖掘工具，方便灵活地对自己的业务数据进行深入的分析研究，这对于管理决策人员而言，无疑是一个福音。

第 10 章 计算机网络

10.1 计算机网络的产生和发展

1946 年，第一代计算机的诞生在人类科学发展史上是一个重要的里程碑。在机械化、电气化时代，人们用机器代替了部分的体力劳动，而计算机的诞生，使得人们可以用它部分地代替人的脑力劳动。20 世纪 80 年代微型计算机的出现，改变了主机模式的集中管理和运行方式，把强大的计算和处理能力交到了个人手里，这为各行各业普遍使用计算机奠定了基础。计算机的普及也正是从微机的出现开始的。第三个发展阶段是网络，人们称网络就是计算机，深刻地反映了网络在计算机发展史中极为重要的作用和影响。

Internet 是全球最大、的、开放的、由众多网络互联而成的计算机一网络。Internet 的发展已经历了 3 个阶段，逐渐走向成熟。从 1969 年 Internet 的前身 ARPANET 的诞生到 1983 年，这是研究试验阶段，主要是进行网络技术的研究和试验。从 1983 年到 1994 年是 Internet 的实用阶段，在美国和一部分发达国家的大学和研究部门中得到广泛应用，它是用于教学、科研和通信的学术网络。从 1994 年以后，开始进入 Internet 商业化阶段，除了原有的学术网络应用外，政府部门、商业企业以及个人广泛使用 Internet，而且全世界绝大部分国家都纷纷接入 Internet，这种迅猛发展的进程反映了 Internet 正日益成熟。当前 Internet 技术和应用的高速发展，对信息技术的发展、信息市场的开拓以及信息社会的形成起着十分重要的作用。但同时，Internet 也面临着多种挑战该包括网络的频宽和可扩展性、网络的安全性、网络的服务质量、多种新的网络应用需求以及引发的商业、文化和社会问题。美国为此启动了两个项目，一个是下一代 Internet，即 NGI，另一个是 Internet2，以迎接网络时代所面临的挑战。

计算机网络的发展趋势可概括为：1 个目标，2 个支撑，3 个融合，4 个热点。

1. 1 个目标

21 世纪计算机网络发展的总体目标就是要在各个国家、进而在全球建立完善的信息基础设施。信息基础设施将改变人们的生活、学习、工作、人际交往的方式，减轻人们的工作负担，提高人民的生活水平，推动社会的进步。

2. 2 个支撑

在实施面向 21 世纪计算机网络发展的总体目标时，有两个重要的支撑技术，即微电子技术和光技术。

3. 3 个融合

支持全球建立完善的信息基础设施的最重要的技术是计算机、通信、信息内容这 3 种技术的融合。计算机包括计算机硬件、计算机软件以及相应的服务；通信包括电话、电视电缆、卫星以及无线通信等；信息内容包括教育、娱乐、出版、信息提供者等。信息时代的新经济是计算机、通信和信息内容 3 种关键经济成分的融合。在企业组织结构上，负责这 3 种技术的组织也应融合。3 种技术也应结束各自发展的状况，而成为集成的技术。

4. 4 个热点

热点包括以下领域：

• **多媒体**。随着数字化技术的成熟，数据、文本、声音、图像这些媒体都能数字化，从而产生了多媒体技术。人们在事务处理和日常生活中，本来就是将各种媒体的信息集成在一起的，所以多媒体技术更加接近于人的生活。

• **宽带网**。要建立真正的宽频多媒体网络，达到信息高速公路的目标，需要高速的传输载体，即信息高速公路的物理结构，包括网络、软件、交换设备。

• **移动通信**。便携式智能终端 PCs 可以使用无线技术，在任何地方以各种速率与网络保持联络。用户利用 PCs 进行个一人通信，可以在任何地方接收到发给自己的呼叫。这些 PCS 系统支持语音、数据和报文

等各种业务。PCS 网络和无线技术将改进入们的移动通信水平，成为未来信息高速公路的重要组成部分。

• **信息安全。**当前网络与信息的安全受到严重的威胁，一方面是由于 Internet 的开放性以及安全性不足，另一方面是由于众多的攻击手段，诸如病毒、陷门、隐通道、拒绝服务、侦听、欺骗、口令攻击、路由攻击、中继攻击、会话窃取攻击等难以防护。以破坏系统为目标的系统犯罪，以窃取篡改信息、传播非法信息为目标的信息犯罪，对国家的政治、军事、经济、文化都会造成严重的损害。为了保证信息系统的安全，需要建设完整的安全保障体系，应具备保护功能、检测手段，以及对攻击的快速反应和事故恢复能力

10.2 网络体系结构及协议

网络体系结构是计算机之间相互通信的层次、各层中的协议和层次之间接口的集合。网络协议是计算机网络和分布系统中互相通信的对等实体间交换信息时所必须遵守的规则的组合。

10.2.1 网络体系结构及协议的定义

共享计算机网络的资源，以及在网中交换信息，就需要实现不同系统中实体的通信。实体包括用户应用程序、文件传送包、数据库管理系统、电子邮件设备以及终端等，系统包括计算机、终端和各种设备等。一般说来，实体是能发送和接收信息的任何东西，而系统是物理上明显的物体，它包含一个或多个实体。两个实体要想成功地通信，它们必须具有同样的语言。交流什么，怎样交流及何时交流，都必须遵从有关实体间某种互相都能接受的一些规则，这些规则的集合称为协议，它可以定义为在两实体间控制数据交换的规则的组合。

协议的关键成分是：

- **语法(syntax)。**包括数据格式、编码及信号电平等。
- **语义(semantics)。**包括用于协调和差错处理的控制信息。
- **定时(timing)。**包括速度匹配和排序。

不同系统中的实体间通信的任务十分复杂，相互不可能作为一个整体来处理，否则任何一方面的改变，就要修改整个软件包。一种替代的办法是使用结构式的设计和实现技术，用分层或层次结构的协议集合。较低级别的、更原始的功能在较低级别的实体上实现，而它们又向较高级别的实体提供服务。图 10.1 表示一般的结构或协议集合，并画出了两个站经由多个交换网连接的情况。1 号和 2 号站每个都有一个或多个希望通信的应用程序。在每一对图 10.1 通信协议之间的关系相似的实体中需要一种面向应用的协议，以协调两个应用模块的行动，并保证共同的语法和语义，这一协议不要知道有关中间通信网络设施的情况，但是要利用网络服务实体所提供的服务。网络服务实体与另一个站中的相应实体要有一个进程的协议，这一协议要处理诸如信息流控制和差错控制之类的事务。在 1 号站和 A 网之间以及 2 号站和 B 网之间也必须有协议。

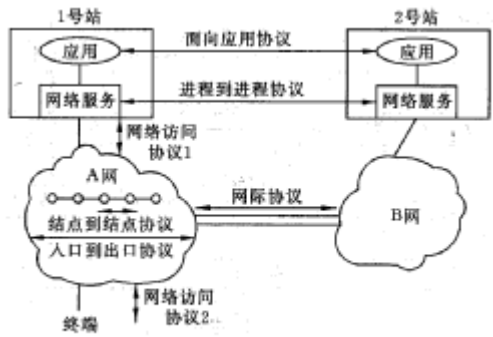


图 10.1 通信协议之间的关系

当采用结构式协议设计时，将计算机之间相互通信的层次以及各层中的协议和层次之间接口的集合称为网络体系结构。

10.2.2 开放系统互连参考模型

国际标准化组织 ISO 在 1979 年建立了一个分委员会来专门研究一种用于开放系统的体系结构，提出了开放系统互连 (Open System Interconnection, OSI) 模型，这是一个定义连接异种计算机的标准主体结构。由于 ISO 组织的权威性，使 OSI 协议成为广大厂商努力遵循的标准。OSI 为连接分布式应用处理的“开放”系统提供了基础，“开放”这个词表示能使任何两个遵守参考模型和有关标准的系统进行连接。

OSI 采用了分层的结构化技术。ISO 分委员会的任务是定义一组层次和每层所完成的服务。层次的划分应该从逻辑上将功能分组。层次应该足几够多，以使每一层小到易于管理，但是也不能太多，否则汇集各层的处理开销太大。OSI 参考模型共有 7 层：物理层，数据链路层，网络层，传输层，会话层，表示层，应用层。图 10.2 为 OSI 参考模型。

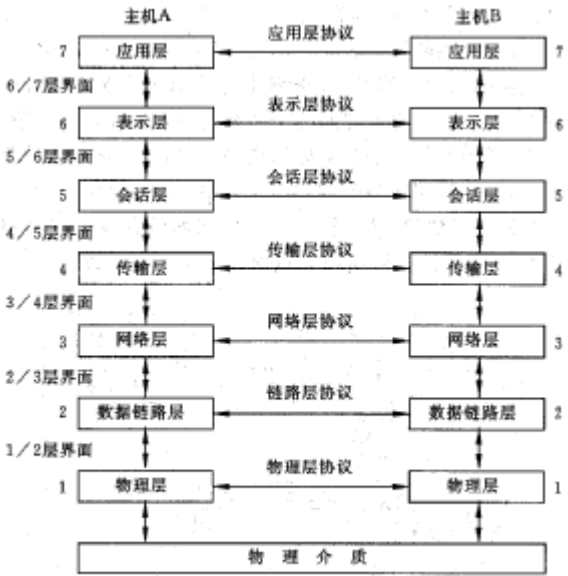


图 10.2 OSI 参考模型

OSI 参考模型的特性为：它是一种将异构系统互连的分层结构；它提供了控制互连系统交互规则的标

准骨架；它定义了一种抽象结构，而并非具体实现的描述；不同系统上的相同层的实体为同等层实体；同等层实体之间通信由该层的协议管理；相邻层间的接口定义了原语操作和低层向上层提供的服务；所提供的公共服务是面向连接的或无连接的数据服务；直接的数据传送仅在最低层实现；每层完成所定义的功能，修改本层的功能并不影响其他层。

开放系统互连参考模型的基本构造技术是分层。每层的目的都是为上边的层提供某种服务，把这些层与提供服务的细节分开就形成结构化模型。

在互连的开放系统中，各子系统的同一层共同构成开放系统中的一层，一般表示为(N)层—某一特定层，(N+1)层—相邻的高层和(N-1)层—相邻的低层。

在 OSI 参考模型中，对等实体的通信必须通过相邻低层以及下面各层通信来完成。从(N+1)实体看，对等(N+1)实体间的通信只能通过相邻对等(N)实体完成。(N)实体向(N+1)实体提供相互通信的能力称(N)服务，即(N+1)实体通过请求(N)服务完成对等实体通信。应注意的是，(N)服务同时也要使用较低层提供的服务功能。

1. 物理层

物理层是 OSI 分层结构中最低层，它建立在物理介质之上，是开放系统与物理介质的接口。它通过与物理介质建立、维护、断开物理连接，为数据链路实体之间提供数据位流的透明传输。

物理连接是开放系统互连的基础，它可分为：永久连接或动态的交换连接，全双工传输或半双工传输，同步传输或异步传输。

物理层应向链路提供下列服务：数据电路标识，物理连接及其端点，物理服务数据单元，排序，故障状态通知和服务质量参数。

作为物理层协议，它具有如下处理功能：激活和拆除物理连接，传输物理服务数据单元，完成物理层一些管理工作。

2.数据链路层

数据链路层的作用是为网络层提供服务，基本目的是从源开放系统网络层向目标开放系统网络层传输数据。

数据链路层向网络层提供的功能有：在物理层提供物理连接的基础上建立、维护和释放数据链路，数据链路服务单元的透明传送，数据传送的流量控制，数据链路服务提供者的差错指示，服务质量管理。

链路层这些功能是对目前常用的标准链路层协议的总结。

链路层协议标准分成两类：第一类是面向字符的传输控制规程；第二类是面向位的链路层规程。

3.网络层

网络层为传输层提供建立、维护和释放网络连接的手段，也提供通过网络连接在传输实体之间交换网络服务数据单元的方法。

网络层通过网络层服务访问点，给传输层提供如下服务：网络地址服务，网络连接及端点标识，网络服务数据单元(NSDU)，服务质量，差错通知，用于保证接收顺序和控制的排序，流量控制，加速网络服务数据单元，复位(网络层差错处理方法，可将顺序计数清零)，释放网络连接，接收确认。

4.传输层

传输层位于网络 7 层模型的正中，具有承上启下的作用。传输层给 OSI 高层提供高可靠、低费用的透明数据传输，它提供端到端的控制和可靠的信息交换，并保证低层的服务质量能满足高层的要求。传输层也是低层中最后的数据传送服务功能的集合。

根据残留差错率和可通告差错率，可把网络服务分为 3 类:A 型网络服务, B 型网络服务,C 型网络服务。针对不同类型的网络服务，ISO 设计了不同类型的传输协议以支持不同的服务类型。

由于网络层以下对传送的协议数据单元大小都有限制，而高层则没有，因而传输层提供了分段/合段功能以满足两方面的要求。

传输实体向会话实体提供的传输服务由 3 个阶段组成：传输连接建立阶段，数据传送阶段，传输连接释放阶段。

根据传输实体所用不同类型的网络服务，可将传输协议分为 5 类，不同的传输协议用于不同的环境，网络性能越差则传输协议就越复杂，分别如表 10.1 所示。

表 10.1 传输协议与网络服务

协议组别	网络类型	名称
0	A	普通级
1	B	基本差错恢复级
2	A	多路复用级
3	B	差错恢复与多路复用级
4	C	差错检测与恢复级

5. 会话层

会话层给会话用户提供一种称为会话(session)的连接,并在其上提供以普通方式传输数据的方法。

- 会话层的主要功能是数据交换,它分为 3 个阶段:会话的建立、使用和拆除。
- 会话层的另一功能是对话管理。
- 同步(synchronization)是会话层另一服务。
- 会话层另一个与同步密切相关的关键特性是活动管理(activity management)。

6. 表示层

有 4 个主要的功能:给用户提供一种执行会话服务的方式,提供一种确定复杂数据结构的方法,管理当前请求数据结构组,在内部和外部形式间实现数据转换。

7. 应用层

是 OSI 的最高层,它借助应用实体(AE)、应用协议和表示服务交换信息,并给应用进程访问 OSI 提供手段。应用层的作用是在实现多个进程相互通信的同时,完成一系列业务处理所需的服务功能,这些服务功能与业务功能(如远地文件操作、远地报文分发等)有很密切的关系。

应用层又可以分为几部分:一是给各种特定的应用服务实体提供公共服务的公共应用服务元素(CASE);二是支持特定应用服务的特定应用服务元素(SASE);三是与用户有关的用户元素(UE),UE 可以作为系统和用户进程交互的手段,在 OSI 中起数据源和数据宿的作用。应用层包括以下几个部分:应用层 CASE(联系控制服务单元 CASE,提交、并发、恢复);文件传送、访问及管理;虚拟终端 VT;作业传送和操作 JIM;电子邮件。

10.2.3 TCP/IP 的分层

1. TCP/IP 分层模型

Internet 采用 TCP/IP 协议,如同 OSI 参考模, TCP/IP 也是一种分层模型。它是由基于硬件层次上的 4 个概念性层次构成,即应用层、传输层、IP 层和网络接口层。图 10.3 给出了这些概念性层次结构以及这些层次之间传送数据的形式。

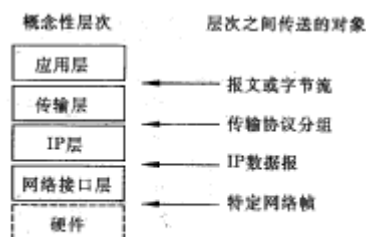


图 10.3 TCP/IP 概念性层次结构

(1)应用层。在最高层,用户调用应用程序来访问 TCP/IP 互连网络提供的多种服务,应用程序负责发送和接收数据。每个应用程序选择所需的传送服务类型,可以是独立的报文序列,或者是连续的字节流。应用程序将数据按要求的格式传送给传输层。

(2)传输层。传输层的基本任务是提供应用层之间的通信,即端到端的通信。传输层管理信息流,提供可靠的传输服务,以确保数据无差错地按序到达。传输层软件将要传送的数据流划分成分组,并连同目的地传送到下一层。

(3)IP 层。IP 层处理机器之间的通信。它接收来自传输层的请求,将带有目的地址的分组发送出去。将分组封装到数据报中,填入数据报头,使用路由算法以决定是直接数据报传送到目的主机还是传给路由器,然后把数据报传送到相应的网络接口来传送。IP 层还要处理接收到的数据报,检验其正确性,并决定是由本地接收还是路由至相应的目的站。

(4)网络接口层。也称数据链路层,这是 TCP/IP 的最底层。该层负责接收 IP 数据报并传送到选定的网络。网络接口包括一个设备驱动器,也可能是一个复杂的具有数据链路协议的子系统。

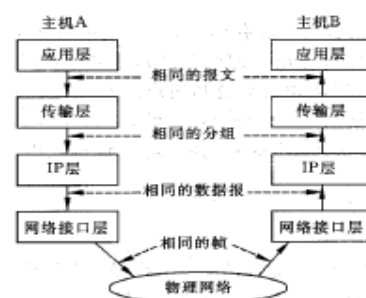


图 10.4 TCP/IP 分层工作原理

2. TCP/IP 分层工作原理

TCP/IP 的分层工作原理如图 10.4 所示,表示了两台主机上的应用程序之间传输报文的路径。主机 B 上的第 N 层接收到的正是主机 A 上的第 N 层发送出来的对象。使用路由器的 TCP/IP 分层工作,如图 10.5 所示。图中报文使用了两种不同的网络帧,一个是从主机 A 到路由器,另一个是从路由器到主机 B。主机 A 发出的帧和路由器 R 接收到的帧相同,但不同于路由器 R 和主机 B 之间传送的帧。应用层和传输层处理端到端的事务,因此发送方的软件能和接收方的对等层软件进行通信。

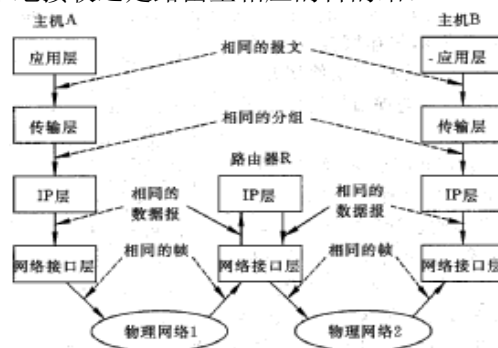


图 10.5 使用路由器的 TCP/IP 分层

3. TCP/IP 模型的分界线

TCP/IP 协议的概念性层次包含两个重要的分界线,一个是协议地址分界线,以区分高层和低层的寻址,另一个是操作系统分界概念性层次分界线,以区分系统与应用程序。TCP/IP 概念层模型的分界如图 10.6 所示。

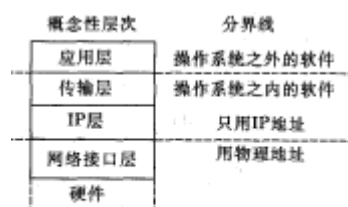


图 10.6 TCP/IP 概念层模型的分界

高层寻址使用 IP 地址，低层寻址使用物理地址。应用程序 IP 层之上的所有协议软件只使用 IP 地址，而网络接口层处理物理地址。

通常将软件分成操作系统软件和非操作系统软件两部分。当协议软件集成到操作系统中后，在协议软件的低层之间进行数据传送的开销比应用程序和传输层之间进行数据传送的开销要小得多。

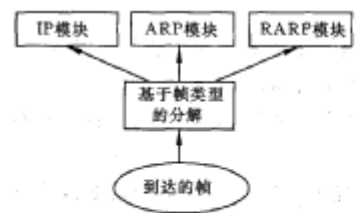


图 10.7 基于帧的报头中的类型字段进行帧的分解

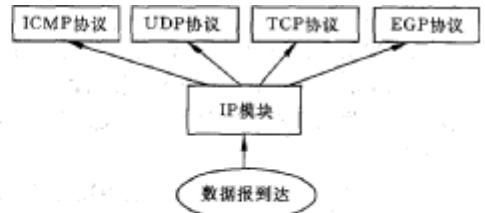


图 10.8 IP 层的分解

4.复用和分解

在整个层次结构中，通信协议使用了复用和分解的技术。当发送报文时，发送方在报文中加入了报文类型、选用的协议等附加信息。所有的报文以帧的形式在网络中复用传送，形成一个分组流。在接收方收到分组时，参考附加信息对接收到的分组进行分解。图 10.7 是根据帧的报头中的类型字段对接收到的帧进行分解。图 10.8 是 IP 层的分解，IP 层的软件模块检查数据报报头，根据其中的协议类型选择相应的协议进行处理。

10.2.4 IP 协议

1. Internet 体系结构

从概念上讲，一个 TCP/IP 互联网提供了 3 组服务。互联网服务的 3 个概念层及其相互的依赖关系如图 10.9 所示。在最底层，无连接传送服务为其他层的服务提供了基础。在第二层，一个可靠的传送服务为应用层提供了一个高层平台。最高层是应用服务层。

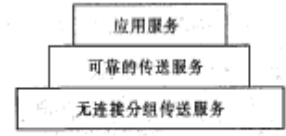


图 10.9 互联网服务的 3 个概念层

Internet 软件是围绕着 3 个层次的概念化网络服务设计的。最基本的互联网服务是由一个分组传送系统组成。该服务被定义为不可靠的、尽最大努力传送的、无连接分组传送系统。所谓不可靠，指的是不能保证正确传送，分组可能丢失、重复、延迟或不按序传送，而且服务不检测这些情况，也不通知发送方和接收方。所谓无连接，指的是每个分组都是独立处理的，可能经过不同的路径，有的可能丢失，有的可能到达。所谓尽最大努力传送，指的是互联网软件尽最大努力来传送每个分组，只有当资源用尽或底层网络出现故障时，才会出现不可靠服务。

这种不可靠的、无连接传送机制称为 Internet 协议，简称 IP 协议。IP 提供了 3 个重要的定义：

- IP 定义了 TCP/IP 互联网上数据传送的基本单元，规定了互联网上传送的数据格式。
- IP 软件完成路由选择功能，选择数据传送的路径。
- IP 包含了一组不可靠分组传送的规则，指明了分组处理、差错信息发生以及分组丢弃等的规则。

2. IP 数据报

互联网的基本传送单元是 IP 数据报，包括数据报报头和数据区的部分。IP 数据报的格式如图 10.10 所示，IP 数据报的格式说明如表 10.2 所示。

表 10.2 IP 数据报格式说明

名 字	位 数	用 途
版本	4	协议的版本
IHL	4	用 32 位字表示的报头长度
服务级别	8	规定优先级、可靠度和延迟参量
数据单元长度	16	以 8 位位组表示的数据报长度
标识	16	标识协议、源和目的
标记	3	包括还有的标记
分段偏移	13	以 64 位为单位表示的分段偏移
生命周期	8	允许跨步的数
用户协议	8	要求 IP 的协议层
报头检查	16	只应用于报头
源地址	64	16 位用于网,48 位用于主机
目的地址	64	16 位用于网,48 位用于主机
任选项	可变	规定附加的服务
填充	可变	保证报头在 32 位的边界处结束

在数据报格式中有一项服务类型规定了本数据报的处理方式。该项为 8 位服务类型字段，分成 5 个小字段。

占 3 位的优先级子字段指示本数据报的优先级，允许发送方指示每个数据报的重要程度。例如，给拥塞控制信息赋予较高优先级，实现不受拥塞影响的拥塞控制算法。

另有 3 位说明数据报希望的传送类型，分别表示低延迟、高吞吐量和高可靠性 3 种类型。传送类型的说明只是对路由选择算法的提示，帮助它根据底层的物理网络技术，选择不同的路由。互联网并不保证满足对传送类型的要求。

数据报要通过底层的物理网传输。为使互联网传送更有效，要保证每个数据报用不同的物理帧传送。因此要把物理网络分组的抽象直接映射到一个实际的分组上。

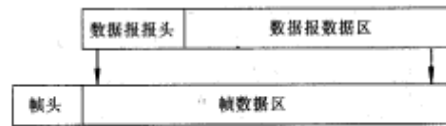


图 10.11 IP 数据报的封装

IP 数据报封装在一个帧中，物理网络将包括数据报报头的整个数据报作为数据。IP 数据报的封装如图 10.11 所示。

在理想情况下，将整个数据报封装在一个物理帧中，可使物理网络上的传送十分有效。为此，可以选

择一个最大数据报长度，使其总能完整地放到一个帧里。但实际上，不同类型的物理网对一个物理帧可传送的数据量规定了不同的上界，称为网络最大传送单元(MTU)。TCP/IP 软件选择了一个方便的初始数据报大小，并且提供了一种方法，对 MTU 较小的网络，把大的数据分成较小的单位，即段。这种划分的过程称为分段，如图 10.12 所示。图中网络 1 和网络 3 的 MTU 为 1500，网络 2 的 MTU 为 620，当 A 和 B 主机通信时，路由器 R₁ 和 R₂ 要分别将数据报分段。

数据报的重新组装有两种方法：一是在通过一个网络后就将分段的数据报重组；二是在到达目的主机后重组。一般来说，后者较好，它允许对每一个数据报段独立地进行路由选择，且不要求路由器对分段存储或重组。

数据报格式中有一个生存时间字段，用来设置该数据报在互联网中允许存在的时间，以秒为单位。当机器向互联网输入一个数据报，就为该数据报设置一个最大生存时间。当数据报通过的主机和路由器对该数据报进行处理时，就递减其生存时间的值，若此值为 0，就把它从网上删除，并向源站点发回一个出错信息。

10.2.5 用户数据报协议

1. UDP 协议功能

TCP/IP 互联网能提供在主机之间传送数据的能力，每个数据报根据其目的主机的 IP 地址进行互联网中的路由选择。为了在给定的主机上能识别多个目的地址，同时允许多个应用程序在同一台主机上工作并能独立地进行数据报的发送和接收，在 TCP/IP 协议簇中设计用户数据报协议(UDP)。它提供了应用程序之间传送数据报的机制。

可将每台机器看作是一些抽象的协议端口的集合，协议端口能区分在一台机器上运行的多个程序。每个 UDP 报文不仅传送用户数据，还传送发送方和接收方的协议端口号，以使接收方的 UDP 软件能将报文送到正确的接收进程，并回送应答报文给对应的发送进程。

UDP 使用底层的互联网协议来传送报文，同 IP 一样，它提供不可靠的无连接数据报传输服务。它不提供报文到达确认、排序以及流量控制等功能，因此报文可能会丢失、重复以及乱序等。而可靠性的问题将由使用 UDP 的该应用程序来解决。

2. UDP 报文格式

每个 UDP 报文称为一个用户数据报，分 UDP 报头和 UDP 数据区两部分。报头由 4 个 16 位长的字段组成，分别说明该报文的源端口、目的端口、报文长度以及校验和。UDP 报文格式如图 10.13 所示。

源端口字段和目的端口字段包含了 16 位的 UDP 协议端口号。长度字段记录该数据报的长度，以 8 位为单位计算，包括报头和用户数据区。校验和字段是可选择的，如该字段值为 0，则表明不进行校验。一般说来，使用校验和字段是必要的。

3. UDP 的协议分层与封装

在 TCP/IP 协议层次结构模型中，UDP 位于 IP 层之上。应用程序访问 UDP 层，然后使用 IP 层传送数据报，如图 10.14 所示。将 UDP 层放到 IP 层之上，表示一个 UDP 报文在互联网中传输时要封装到 IP 数据报中。最后，网络接口层将数据报封装到一个帧中再进行物理传输通道上的传输。封装过程如图 10.15 所示。

由图可知，IP 层的报头指明了源主机和目的主机的地址，而 UDP 层的报头指明了主机上的源端口和目的端口。

4. UDP 的复用、分解与端口

UDP 也提供复用和分解的功能。它接收多个应用程序送来的数据报，把它们送给 IP 层去传输，同时它接收 IP 层送来的 UDP 数据报，把它们送给对应的应用程序。

从概念上讲，所有的 UDP 软件与应用程序之间的复用和分解都要通过端口机制来实现。实际上每个应用程序在发送数据报之前必须与操作系统进行协商以获得协议端口和相应的端口号。凡是利用指定的端口

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1																							
版本				IHL				服务级别				数据单元长度											
标识				标识				分段偏移															
生命期				用户协议				报头检查和															
源地址																							
目的地址																							
任选项+填充																							
数据																							

(a) IP 协议数据单元

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1															
类 型				编 码				检 查 和							
参 数															
信 息															

(b) ICMP 协议数据单元

图 10.10 IP 数据报格式

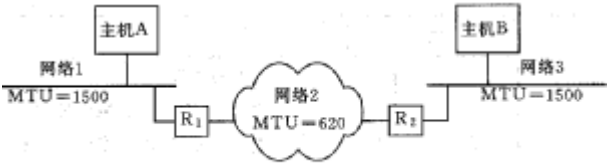


图 10.12 数据报的分段

UDP源端口															
UDP目的端口															
UDP报文长度															
UDP校验和															
数据															
...															

图 10.13 UDP 报文格式

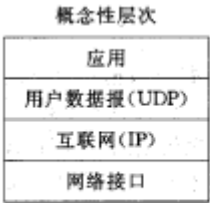


图 10.14 分层模型中的 UDP 层

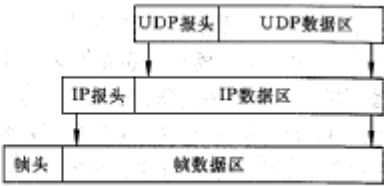


图 10.15 UDP 的封装

发送数据报的应用程序都要把端口号放入 UDP 报文中的源端口字段中。

UDP 的分解操作如图 10.16 所示。UDP 从 IP 层接收了数据报之后，根据 UDP 的目的端口号进行分解操作。

UDP 端口号的指定有两种方式：一种是由某些管理机构指定的称为著名端口，供用户使用。另一种是动态绑定方式，由应用程序指定端口。表 10.3 给出了一些已指定的著名端口号及其用途。表中第 2 列是 Internet 的标准关键字，第 3 列是多数 UNIX 系统中用来表示这些端口的关键字。

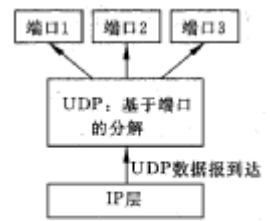


图 10.16 UDP 的分解操作

10.2.6 可靠的数据流传输

UDP 提供的服务是不可靠的数据传送服务，当传送过程中出现差错时，在网络软件发生故障或网络负载太重时，分组可能会丢失，数据可能被破坏。这就需要应用程序员编制程序，负责进行差错检测和恢复工作。对于传输数据量很大的应用来说，采用不可靠的数据传输是不合适的。因此需要有一种可靠的数据流传输方法，TCP 就是 Internet 协议簇中解决这一问题的服务和协议。

TCP/IP 的可靠传输服务有以下 5 个特征：

- **面向数据流。**当两个应用程序传输大量数据时，将这些数据当做一个可划分为字节的位流。传输时，在接收方收到的位流与发送方发出的完全一样。

- **虚电路连接。**在传输开始之前，接收应用程序和发送应用程序与操作系统进行交互，双方的操作系统的协议软件模块通过互联网通信，进行数据传输的准备与建立连接。通常用虚电路连接来描述这个过程，对应用程序来说，这种连接好像是一条专用线路，而实际上是由数据流传输服务提供的虚电路连接。

- **有缓冲的传输。**使用虚电路服务来发送数据流的应用程序不断地向协议软件提交以字节为单位的数据，并放在缓冲器中。当积累到足够多的数据时，将它们组成大小合理的数据报，再送到互联网上传输。这样可提高传输效率，减少网络流量。当应用程序传送特别大的数据块时，协议软件将它们划分为适合于传输的较小的数据块。在接收端，协议软件收到的数据流与其发送的顺序完全相同。

- **无结构的数据流。**TCP/IP 协议并未区分结构化的数据流。使用数据流服务的应用程序必须在传输数据前就了解数据流的内容，并对其格式进行协商。

- **全双工连接。**TCP/IP 提供的连接功能是双向的。对一个应用程序而言，全双工连接包括两个独立的、流向相反的数据流，而且这两个数据流之间不进行显式的交互。全双工连接方式节省了网络的带宽。

10.2.7 传输控制协议

1. TCP 功能

传输控制协议 (TCP) 定义了两台计算机之间进行可靠的传输而交换的数据和确认信息的格式，以及计算机为了确保数据的正确到达而采取的措施。协议规定了 TCP 软件怎样识别给定计算机上的多个目的进程，如何对分组丢失和分组重复这类差错进行恢复。协议还规定了两台计算机如何初始化一个 TCP 数据流传输以及如何结束这一传输。

虽然 TCP 描述了应用程序使用 TCP 软件的一般方式，但它并未指定应用程序和 TCP 软件之间的接口细节。也就是说，协议规定了 TCP 提供的操作，但并未指定应用程序调用这些操作的具体过程。

TCP 在协议层次结构中位于 IP 层之上，如图 10.17 所示。TCP 允许一台计算机上的多个应用程序同时进行通信；也能对接收到的数据进行分解，分别送到多个应用程序。TCP 使用协议端口号来标识一台计算机上的多个目的进程。每个端口被赋予一个小的整数以便识别。

TCP 是建立在连接的抽象概念上的。它所标识的对象不是某个端口，而是一个虚电路连接。TCP 使用连接而不是协议端口作为基本的抽象概念，连接是用一对端点来标识。

TCP 将端点定义为一对整数 (host, port)，其中 host 是主机的 IP 地址，port 是该主机上的 TCP 端口

表 10.3 著名端口及关键字

十进制	关键字	UNIX 关键字	描述
0	—	—	Reserved——保留
7	ECHO	echo	Echo——回送
9	DISCARD	discard	Discard——丢弃
11	USERS	sysstat	Active Users——活动用户
13	DAYTIME	daytime	Daytime——白天
15	—	netstat	Who is up of NETSTAT
17	QUOTE	qotd	Quote of the Day——日期引用
19	CHARGEN	chargen	Character Generator——字符发生器
37	TIME	time	Time——时间
42	NAMESERVER	name	Host Name Server——主机名服务器
43	NICNAME	whois	Who is——是谁
53	DOMAIN	nameserver	Domain Name Server——域名服务器
67	BOOTPS	bootps	Bootstrap Protocol Server——引导协议服务器
68	BOOTPC	bootpc	Bootstrap Protocol Client——引导协议客户
69	TFTP	tftp	Trivial File Transfer——简单文件传送
111	SUNRPC	sunrpc	Sun Microsystems RPC
123	NTP	ntp	Network Time Protocol——网络时间协议
161	—	snmp	SNMP net monitor——SNMP 网络监控器
162	—	snmp-trap	SNMP traps——SNMP 陷阱
512	—	biff	UNIX comsat
513	—	who	UNIX rwho daemon
514	—	syslog	system log——系统登录
525	—	timed	Time daemon

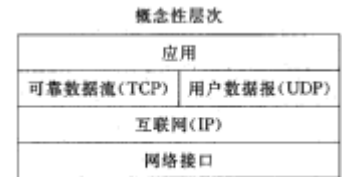


图 10.17 TCP 在协议分层中的层次

号。一个连接是用它的两个端点表示的。由于 TCP 使用端点来识别连接，一台计算机上的某个 TCP 端口号可以被多个连接所共享。因此，程序员能设计提供同时为多个连接服务的程序，而不需要为每个连接设置各自的本地端口号。

TCP 是一个面向连接的协议，它需要两个端点都同意连接才能进行通信。在 TCP 进行互连网络通信之前，连接双方的应用程序必须建立连接。采用客户机服务模式建立这种连接时，客户方应用程序主动打开请求，通知操作系统要建立一个连接，服务方应用程序通知操作系统，希望建立一个输入的连接，即被动打开的功能。连接建立之后，应用程序开始传输数据。

TCP 将数据流看作字节的序列，为了便于传输，又将这个序列划分为若干段。通常每个段被放置到一个 IP 数据报中在互联网上传送。

TCP 使用专门的滑动窗口机制来解决传输效率和流量控制这两个问题。TCP 协议允许随时改变窗口大小。在每个确认中，除了指出已经收到的分组外，还包括了一个窗口通告，用来说明接收方还能再接收多少数据。可以将窗口通告的值当做当前的接收缓冲区大小。通告值增加，发送方扩大发送滑动窗口；通告值减少，发送方缩小发送窗口。采用滑动窗口机制，不仅提供了可靠传输服务，而且还提供了流量控制功能。TCP 采用的滑动窗口机制解决了端到端的流量控制，但并未解决整个网络的拥塞控制。

2. TCP 报文格式

两台计算机上的 TCP 软件之间传输的数据单元称为报文段。通过报文段的交互来建立连接、传输数据、发出确认、通告窗口大小以及关闭连接。TCP 报文格式如图 10.18 所示。

报文分为两部分，即报头和数据。报头携带了所需的标识和控制信息。源端口和目的端口字段包含了标识连接两端的应用程序的 TCP 端口号。顺序号字段指示该报文段在发送方的数据字节流中的位置。确认号字段指示本机希望接收的下一个字节组的序号。顺序号字段的值指的是该报文段流向上的数据流的位置，即发送序号；确认号指的是与该报文段流向相反方向上的数据流。

报头长度字段是一个以 32 位为单位的报头长度值。由于选项字段的长度是根据所选内容而变化的，因此 TCP 报头长度随着它所选择的选项而变化。6 位长的保留字段是为将来的应用而保留的。

TCP 报文段有多种应用，包括传输数据、携带确认信息、携带建立或关闭连接的请求。TCP 软件使用 6 位长的码位来指示报文段的应用目的和内容，给出了对报头中其他字段的解释，如表 10.4 所示。

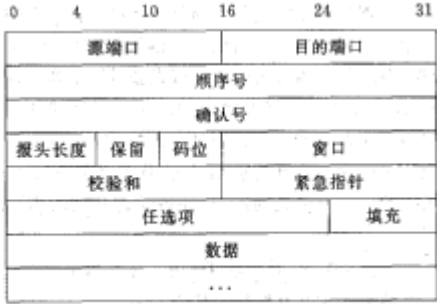


图 10.18 TCP 报文格式

10.3 局域网技术、

局域网是将小区成内的各种通信设备互连在一起的通信网络。决定局域网特性的主要技术有以下 3 个：用以传输数据的传输介质，用以连接各种设备的拓扑结构，用以共享资源的介质访问控制方法。

这 3 种技术在很大程度上决定了传输数据的类型、网络的响应时间、吞吐率和利用率，以及网络应用等各种网络特性。其中最重要的是介质访问控制方法，它对网络特性起着十分重要的影响。基于交换技术的局域网不同于传统局域网采用的共享介质访问控制技术。

10.3.1 局域网定义和特性

局域网(local area network, LAN)是将小区域内的各种通信设备互连在一起的通信网络，从这个定义可引出局域网的 3 个属性：

- 局域网是一个通信网络，从协议层次的观点看，它包含着下 3 层的功能，将连接到局域网的数据通信设备加上高层协议和网络软件组成为计算机网络，我们称它为计算机局域网。
- 这里指的数据通信设备是广义的，包括计算机、终端和各种外围设备等。
- 这里指的小区域可以是一建筑物内、一个校园或者大至几十公里直径的一个区域。局域网的典型特性如下：高数据速率(0.1Mbps—100Mbps)；短距离(0.1km—25km)，低误码率(10^{-11} — 10^{-8})。

在局域网中，不论采用什么样的拓扑结构，都可以把传输介质作为各站点共享的资源。将传输介质的频带有效地分配给网上各站点的用户的方法称为介质访问控制协议，设计一个好的介质访问控制协议有 3 个基本目标：协议要简单，获得有效的通道利用率，对网上各站点用户公平合理。传统的局域网介质访问控制协议有载波监听多路访问/冲突检测 CSMA/CD(carrier sense multiple access/ collision detection)、标记环 Token Ring 和标记总线 Token Bus 等。

表 10.4 TCP 报头的码位字段的含义

位(从左到右)的标识	该位置 1 的含义
URG	紧急指针字段可用
ACK	确认字段可用
PSH	请求急迫操作
RST	连接复位
SYN	同步序号
FIN	发送方字节流结束

10.3.3. 快速以太网

1.快速以太网类型

快速以太网(fast ethernet)是一个新的 IEEE 局域网标准,于 1995 年由原来制定以太网标准的 IEEE 802.3 工作组完成。它为现有广大以太网用户提供了一个平滑升级的方案。快速以太网标准的正式名为 100Base-T。表 10.5 是以太网和快速以太网的比较。

共享介质快速以太网和传统以太网采用同样的介质访问控制协议,即 CSMA/CD 所有的介质访问控制算法不变,只是将有关的时间参量加速 10 倍。

快速以太网和 10Base-T 一样,采用 HUB 的拓扑,这是当前在 10Mbps 以太网中广泛采用的拓扑。它使用同样的缆线配置、同样的软件,并由大量厂商支持,因此它为用户提供了 10Base-T 平滑过渡到 100Mbps 性能的方案。

快速以太网支持结构化布线,包括 3 类、4 类、5 类非屏蔽双绞线 UTP,150Ω 屏蔽双绞线 STP 以及光纤,这些介质都是当前流行的,且各类介质可混合使用,通过中继器或交换器连接起来。

为了支持各种类型的介质,快速以太网提供了 3 种类型的发送接收器,两种用于双绞线(即 100Base-T4 和 100Base-TX),一种用于光纤(即 100Base-FX),表 10.6 列出了这 3 种类型收发器支持的缆线。

3 类线原本是适用于 10Base-T 的,在 100Base-T4 中使用 3 类 UTP,信号速度仅为 25MHz,为了能得到足够的频宽,100Base-T4 需要 4 对线缆并联使用,这样就解决了 3 类线也能运行 100Mbps 的速率的问题。100Base-TX 只需两对线,但必须用 5 类线。100Base-FX 最大距离可达 2km,可适用于大的企业网。其他两种类型收发器都只支持 100m 距离,这是采用 HUB 结构对距离的基本要求。

2.快速以太网产品

快速以太网产品分两类,即适配器和 HUB。适配器结构较简单,一边是总线结构,将数据传送至主机。中继器或 HUB;另一边接到所选的介质,可以是双绞线、光纤,或者是一个介质独立接口 MIT, MII 是用来连接外部收发器用的,其功能类似于以太网的 AUI。

HUB 的产品则名种名样,下要可分成两类,一类是共享机制的中继器,一类是交换机制的交换器,且都采用星型的布线结构。共享结构的 HUB 便宜,但同时只允许一个用户发送数据;交换结构的 HUB 要贵一些,允许同时有多个用户设备相互之间传送数据,提高了网络的性能,这两种结构都能用于 10Mbps 或 100Mbps,由此可组合产生 4 种不同类型的以太网 HUB,如表 10.7 所示。

凡快速以太网可根据用户需求组成不同性能的局域网,一种是混合使用 10Mbps 和 100Mbps 的网,另一种是纯 100Mbps 的网,并可混合使用共享和交换机制。

10.3.4 千兆位以太网

1. 千兆位以太网规程和标准

20 世纪 70 年代末问世的以太网至今仍被广泛采用,它采用 CSMA/CD 的访问控制机制,是一种共享介质的局域网。但是以太网的类型和传输速率有了很大变化。最初的以太网传输速率为每秒 10 兆位,基于同轴电缆的 10Base 5 类 10Mbps 以太网成为大学和企业的骨干网。随后,一种基于细的同轴电缆的 10Base 2 类 10Mbps 以太网问世,由于这类以太网安装简单,价格低廉,很快得到了推广应用。

后来,有人想到用电话系统使用的双绞线来代替专用的同轴电缆。这类以太网比同轴电缆的以太网更便宜、更灵活,也更适合于局域网环境。采用 HUB 结构易于安装,且可靠性高。10Base 5 类和 10Base 2 类以太网采用的同轴电缆或连接发生故障可能造成整个网络瘫痪,而 HUB 结构的以太网发生的故障往往是局部性的。1996 年 IEEE 802 委员会将这类以太网定名为 10Base-T。随后又发展了以光缆作为传输介质的 10Base-F,既提高了可靠性,又扩展了传输距离的极限。

近年来,随着对局域网传输速率和频宽要求的增加,发展了 100Base-T 的以太网,采用了 5 类双绞线作为这类网络的传输介质的推荐标准。也可采用质量较低的 3 类双绞线,应用于短距离的传输。

在传统的 10Mbps 或 100Mbps 以太网基础上,减少其传输距离,就能获得更高的速度。技术的进步,使一种新型的以太网能达到千兆位速率,同时又能达到和传统以太网相同的传输距离,称为千兆位以太网。

千兆位以太网遵守同样的以太网通信规程,即 CSMA/CD 访问控制方法,因此它仍然是一种共享介质的

表 10.5 以太网和快速以太网的比较

特点	以太网	快速以太网 100Base-T
速率	10Mbps	100Mbps
价格	X 元	低于 2X 元
IEEE 标准	802.3	802.3
介质访问控制	CSMA/CD	CSMA/CD
拓扑结构	总线或星型	星型
线缆支持	电缆,UTP,光纤	UTP,STP,光纤
用 UTP 连接距离	100m	100m
介质独立接口	AUI	MII
全双工能力	是	是
厂商广泛支持	是	是

表 10.6 3 种类型收发器支持的缆线

收发器类型	缆线类型	缆线对数
100Base-T4	3 类 UTP	4 对
	4 类 UTP	4 对
	5 类 UTP	4 对
100Base-TX	5 类 UTP	2 对
	150Ω STP	2 对
100Base-FX	62.5μm/125μm 多模光纤	2 对

表 10.7 4 种类型以太网 HUB

产品类型	链路速率 (Mbps)	结构	总频宽	特 点
10Mbps 中继器	10	共享	10Mbps	连到快速以太网 HUB 上任一 10Mbps 端口
100Mbps 中继器	100	共享	100Mbps	也包括连到现存的 10Mbps 以太网端口
10Mbps 交换器	10	交换	N×10Mbps	也包括 100Mbps 端口连到服务器或 100Mbps 网
100Mbps 交换器	100	交换	N×100Mbps	也包括连到现存的 10Mbps 以太网端口

局域网。发送到网上的信号是广播式的，接收站根据地址接收信号。网络接口硬件能监听线路上是否已存在信号，以避免冲突，或在没有冲突时重发数据。可以设计成简单的半双工通信；也可设计成全双工通信，可以同时发送和接收信息。

千兆位以太网也有铜线及光缆的两种标准。铜线标准为 1000Base-CX，最大传输距离为 25 英尺，并需用 150Ω 的屏蔽双绞线 STP，以 1.25Gbps 的串行线速率在一种专用电缆 Twinax 上传输。目前大部分以太网产品都是基于非屏蔽双绞线的，这是为了简单和减少成本，但在千兆位传输速率下 STP Twinax 能抗电磁干扰。如果换成非屏蔽双绞线 UTP，则只能缩短传输距离。也有另一种方案，即采用 4 对非屏蔽双绞线，可支持 100m-200m 的传输距离。

基于光缆传输的千兆位以太网使用与光纤通信相同的物理信号系统来进行通信，对 850 nm 的短波长，标准为 1000Base-SX 能支持 300m 的传输距离。使用 1300nm 的波长，标准为 1000Base-LX，支持的传输距离为 550m。如采用单模光缆，则可支持更长的距离。

2. 交换式 LAN 结构的千兆位以太网

著名的 Moore 定律告诉我们，集成电路芯片的集成度每 18 个月提高一倍，随之而引起的 PC 处理器的能力也以同样的速度增长。有一个反映网络发展速度的 Metcalfe 定律，表明网络性能的增长速度等于连在网上的 PC 数的平方，也就是说网络的频宽每年提高了 3 倍。图 10.20 描述了网络速率的增长情况。

另一方面局域网的流量分布也发生了变化，以往大部分的通信限于内部，而现在有相当一部分的通信是和外部网络交换信息，这主要是因为 Internet 的规模和应用的发展。图 10.21 描述了局域网通信分布的趋势。

千兆位以太网的问世，反映了当前局域网技术的发展趋势。它不仅满足了应用对网络速率和频宽的要求，而且较好地解决了和传统的 10Mbps 以太网、100Mbps 以太网的兼容以及升级，因此它在今后局域网的市场将作为主流技术发展。已经有不少厂商提供了交换式 LAN 结构的千兆位以太网解决方案，且集成了传统的 10Mbps 和 100Mbps 以太网。图 10.22 是交换式 LAN 结构的千兆位以太网配置。

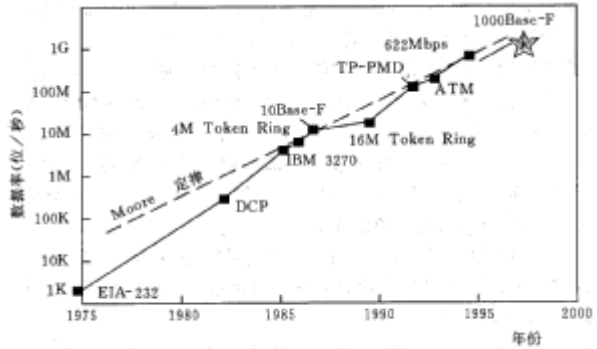


图 10.20 网络速率的增长

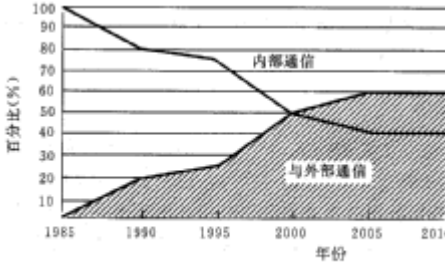


图 10.21 局域网通信分布趋势

10.4 广域网技术

广域网是作用的地理范围从数十公里到数千公里，可以连接若干个城市、地区甚至跨越国界、遍及全球的一种计算机网络。

10.4.1 点到点通信

电话网络是分布最广的通信网络，经过调制解调器，能够传输 9600 波特率或更高速率的数据。它使用起来相对简单，而且对用户来说可以有很大的流动性和灵活性。通过电话网连接远程计算机，特别适合于家庭用户和外出出差人员与总部联络。通过电话网传输的主要问题在于电话话路质量，在话路质量较高的情况下可以有比较高的传输速率和可靠的传输，而话路质量低会严重影响通信质量。

Internet 是由众多的主机和路由器以及将它们连接在一起的通信基础设施构成。在一个大楼里，人们广泛使用各种局域网，但大部分广域网基础设施是借助于点到点的租用线。

通常点到点的通信主要适用于两种情况：第一种情况是成千上万个组织有各种局域网，每个局域网含有众多主机和一些联网设备以及连接至外部的路由器，通过点到点的租线和远地路由器相连；第二种情况是成千上万个用户在家里使用调制解调器和拨号电话线连到 Internet，这是点到点连接的最主要应用。

不论是路由器对路由器的租线连接，还是拨号的主机到路由器的连接，都需要制定点到点的数据链路协议，用以组成帧、差错控制，以及其他数据链路层功能。有两种协议广泛用于 Internet，即串行 IP 协议 (serial line IP, SLIP) 和点对点协议 (point to point protocol, PPP)。

为了解决 SLIP 存在的问题 Internet IETF 成立了一个组来制定点到点的数据链路协议的 Internet 标

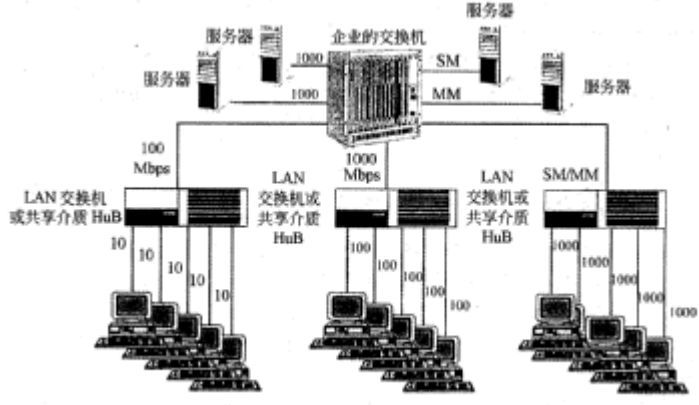


图 10.22 交换式 LAN 结构的千兆位以太网配置

准。该标准命名为 PPP, 即点到点协议。该协议文本描述于 RFC1661, 以及改进后的文本 RFC1662, RFC1663。PPP 能支持差错检测, 支持各种协议, 在连接时 IP 地址可赋值, 具有身份验证功能, 以及很多对 SLIP 改进的功能。虽然目前很多 Internet 服务提供者同时支持 SLIP 和 PPP 这两种协议, 但从今后发展看, 很明显 PPP 是主流, 它不仅适用于拨号用户, 而且适用于租用的路由器对路由器线路。PPP 提供以下 3 个功能:

- 成帧的方法可清楚地区分帧的结束和下一帧的起始, 帧格式还处理差错检测。
- 链路控制协议(Link Control Protocol, LCP)用于启动线路、测试、任选功能的协商以及关闭连接。
- 网络层任选功能的协商方法独立于使用的网络层协议, 因此可适用于不同的网络控制协议(Network Control Protocol, NCP)。

10.4.2 分组交换网

1. 分组交换网原理

在传统的网络概念中, 解决远程计算机联网的主要手段是通过租用电话线路, 通过调制解调器将数据信号转变成模拟信号, 在公共交换电话网(public switched telephone network)上进行传输。

公共分组数据网络(packet switched data network)已经成为广域网中的重要传输系统。分组交换是一种在距离相隔较远的工作站点之间进行大容量数据传输的有效方法, 它结合线路交换和报文交换的优点, 将信息分成较小的分组进行存储、转发, 动态分配线路的带宽。它的优点是出错少、线路利用率高。它有数据报和虚电路两种方式, 虚电路方式能在一条物理链路上建立若干条逻辑上的虚电路, 使用户感觉到仿佛有若干条物理链路一样。数据报方式灵活、快速可靠。分组交换技术是数据网络中最广泛使用的一种交换技术, 现有的公共数据交换网都采用这种技术。

CCITT X. 25 是常见的公共数据网的一种协议, 因此公共数据网一般也称为 X. 25 网。X. 25 网实际上包括相关的一组协议, 如 CCITT X. 3, X. 28, X. 29, X. 7 5 等。X. 25 描述了将一个分组终端连接到一个分组网络上所需要做的工作。通过虚电路方式, 它能负责维护一个通过单一物理连接的多用户会话, 每个用户会话被分配一个逻辑信道, 这是 X. 25 一个很强的功能。X. 25 的另一个特性是允许建立一个能提供两种类型的分组交换网络, 这两种类型是高优先级类型和正常优先级类型。

X. 25 网络与计算机之间的接口一般是通过专用设备或网关、路由器来解决的。3 个 CCITT 协议定义了配备异步通信设备的终端或个人计算机与 X. 25 网络的常规连接, 通常是使用带有 X. 25 的分组组装、拆装设备(PAD)的异步通信设备完成与分组交换网络的连接。CCITT X. 3 描述了一个 X. 25 PAD 的功能和控制参数; CCITT X. 28 定义了一台终端与 X. 25 PAD 之间的交互作用, 为每个用户提供了一个常规的 X. 25 网络的连接; X. 29 定义了一台主机和其相连的 PAD 之间的交互作用。现在微机局域网与 X. 25 网互连可以有几种方案, 一种可以采用现在的路由器和网关同时连接 X. 25 网和本地局域网, 这种方案适合规模较大、多种协议共存的网络, 具有安装配置简单、维护方便的特点, 可以通过 X. 25 与远程其他协议的网络进行互联; 另一种方案是采用一台微机作为路由器, 安装相应的 X. 25 网卡和路由软件, 使用于中小规模且协议比较少的网络; 再有一种方案就是使用 PAD, 这种方案适合 X. 25 协议的环境, 与远程其他协议的网络互联受限制。

X. 25 网络是一种中速数据网络, 一般速率在 64kbps 以内。但是随着计算机应用需求的不断扩大, 它已经无法满足先进的网络应用需求了, 例如, 交互式的会话通信对实时性要求很高, 延迟要很小; 数字化的多媒体数据的传输和巨型机并行计算产生的高速数据交换要求高速宽带的通信网。

2. X.25 分层协议

X. 25 包含有 3 层协议, 即物理层, 链路层和分组层, 如图 10.23 所示。这 3 层对应于 OSI 模型最底下的 3 层。物理层涉及站点(计算机, 终端)与把这个站连到分组交换网的链路之间的接口。X. 25 标准把用户机器称为数据终端设备(DTE), 把 DTE

所连的分组交换结点称为数据电路终端设备(DCE)。X. 25 所用的物理层标准是 X. 21, 但是在许多场合, 也用其他标准, 例如 EIA-232。链路层提供可靠的数据传输。链路层所用的标准是 LAP B(Link Access Protocol-Balanced)。LAP B 是 HDLC 的一个子集。分组层提供外部虚电路服务。

图 10.24 是 X. 25 3 层之间的关系。用户数据被送到 X. 25 的第 3 层, 在第 3 层加上含有控制信息的报头, 从而组成了一个分组。控制信息用于协议的操作。整个 X. 25 分组然后送到 LAP B 实体, LAP B 在此分组的前后各加上控制信息组成一个 LAP B 帧, 在帧中加入控制信息也是为了协议的操作。

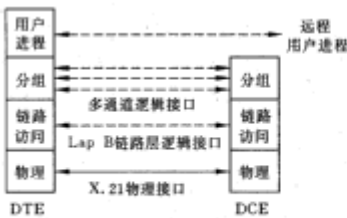


图 10.23 X.25 接口



图 10.24 用户数据和 X.25 协议控制信息

10.4.3 帧中继网

1. 帧中继网产生背景

帧中继是一种只是简单地提供面向连接的、将数据从甲地传递至乙地的、廉价的、中速的公共网。这种网的产生是由于近 20 年技术的变化和发展。20 年前,采用电话线的通信是低速的、模拟的、不可靠的,而计算机也是处理速度低、造价高,这样就需要复杂的协议来屏蔽差错,而用户计算机要具有这种功能价格又太高。现今租用的电话线可以是高速的、数字的、可靠的,而计算机处理速度既快又便宜,这样只需要简单的协议,且大部分处理可由用户计算机自行处理,不需要网络来处理。

帧中继技术是由 X.25 分组交换技术演进而来的,由于光纤通信的误码率低,为了提高网络速率,省去了很多在 X.25 分组交换中的纠错功能,使帧中继的性能优于 X.25 分组交换的性能。帧中继的主要特点是中速到高速的数据接口;标准速率为 DSI,即 T1 速率;可用于专用和公共网;仅传输数据;使用可变长度分组。

帧中继可看成是一条虚拟的租用线,用户租用一条两点之间的永久虚电路,可传送最多 1.6KB 字节的帧,也可租用连接多个场地的永久虚电路,用户租用一条线路,可整天以最大速率传送数据。对于虚拟租用线,用户可在某一时刻以最大速率传送数据,但长时间的平均速率必须低于预定的水平,所以租用费也便宜得多。

帧中继只提供最低的服务,即决定每个帧的起始和结束,以及检测传输差错。如果接收到一个坏的帧,帧中继只是简单地把它丢弃,由高层协议来处理这个差错,不像 X.25,帧中继不提供响应和正常的流控。

2. 帧中继网与 X.25 网比较

传统的分组交换网均采用 X.25 网,它不仅决定了用户—网络接口,而且也影响网络的内部设计。

X.25 网的主要特性有以下 3 点:

- 由于建立和拆除虚电路的呼叫控制分组和数据分组在同一通道和同一虚电路上传输,其结果是占用了通道频带。

- 虚电路的复用发生在第 3 层。

- 在第 2 层和第 3 层都需要流控和差错控制的机制。

由此得出结论 X.25 的开销相当可观。图 10.25(a)表示在 X.25 分组交换网中,从源端系统传输一个数据分组到目的地所需的数据链路帧的流。在通过网络的每一步,数据链路控制协议要完成数据帧和响应帧的交换。而在每一个中间结点要为每个虚电路保持一些状态表,以处理 X.25 协议的呼叫管理、流控以及差错控制等。

当网络链路的差错概率很大时,这些开销是值得的,但是现代数据通信设施具有高质高可靠的传输链路,很多场合采用光纤,提供了高可靠的传输技术,上述开销就显得不合适了。采用光纤和数字传输技术可得到高的数据速率,这样,采用 X.25 不仅其开销是不必要的,而且会大大影响高速数据链路的利用率。

帧中继的设计,消除了很多 X.25 对最终用户系统和分组交换网所需的开销。

帧中继与 X.25 分组交换的最主要区别有以下 3 点:

- 载送呼叫控制信令的逻辑连接和用户数据是分开的。因此中间结点毋需为每个连接的呼叫控制保持状态表。

- 逻辑连接的复用和交换发生在第 2 层,而不是第 3 层,从而减少了处理的层次。

- 结点到结点之间毋需流控和差错控制,由高层负责端到端的流控和差错控制。

图 10.25(b)表示帧中继网的操作,只有单一的用户数据帧从源站送到目的站,而由高层回送响应。

与 X.25 相比,帧中继没有链路到链路的流控和差错控制能力。在 X.25 中,在单个物理链路上载送多个虚电路,使用 LAP-B 协议为源站点到分组交换网以及从分组交换网到目的站提供链路层的可靠传输,而且在网络的结点到结点的每一步,都提供可靠的控制协议。帧中继不提供上述 X.25 的控制功能,然而,随着传输和交换设施可靠性的提高,帧中继这个缺点就显得不重要了。

帧中继的优点是精简了通信处理。协议对用户—网络接口以及网络内部处理的功能降低了,从而得到了低延迟和高吞吐率的性能。使用帧中继与 X.25 相比,其吞吐率可提高一个数量级,甚至更高。ITU-T 推荐的 I.233 指出帧中继可用于 2Mbps 的访问速率。

在高速信道上提供帧中继服务,对下面一些应用很有用:

- 大信息量的交互数据应用,如高清晰度图形,这种应用的特性是低延迟和高吞吐率。

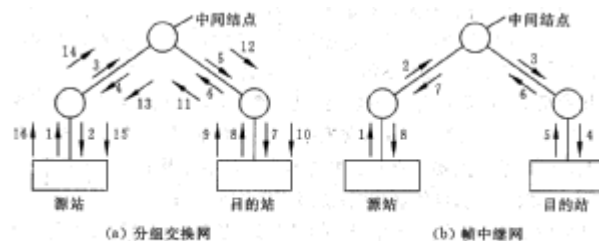


图 10.25 分组交换和帧中继比较

- 大的文件传送。这类应用延迟特性不是主要的，但要求高吞吐率。
- 低数据率的多路复用。对于大量用户、但每个用户的应用数据率不高的情况，可提供经济的访问设施。
- 字符交互通信，如文本编辑。这类应用的特性是短帧、低延迟、低吞吐率。

10.4.4 ATM 网

10.4.4.1 ATM 协议参考模型

在 1.321 中定义了 ATM 协议参考模型，如图 10.26 所示。这是一个类似 OSI 模型的逻辑层次结构，但是 ATM 与 OSI 模型之间的整个关系尚待决定，图中的含义如下：

用户面提供用户信息的传输；控制面负责呼叫控制和连接控制功能。管理面负责网络维护和完成运行功能；面管理执行与整个系统有关的管理功能；层管理处理各层的运行和维护功能；物理层主要是传输信息；ATM 层主要完成交换、路由及多路复用；ATM 适配层(AAL)主要负责与较高层信息的匹配。

1.物理层

它由两个子层组成，即物理介质子层和传输汇聚子层。物理介质子层支持纯粹与介质有关的位功能，如光电转换、位定时及线路编码。传输汇聚子层把 ATM 信元流转换成在物理介质上传输的位，如把帧匹配成在传输系统中所用的格式(SDH, PDH, 基于信元的格式)、信元定界等功能。

2.ATM 层

ATM 层的基本功能是负责生成信元，它不管载体的内容，且与服务无关。它只为载体生成信元头并附给载体，以形成信元标准格式，所以跨越 ATM 层到物理层的信息单元只能是 53 个字节的信元。ATM 层的 4 个主要功能为：多路复用、多路复用分解，信元 VPI, VCI 的转换，信元头的产生和去除，流控。

3.ATM 适配层(AAL)

AAL 由两个子层组成，即分段重组子层和汇聚子层。前者把高一层送来的信息单位分段成 ATM 信元，或者把 ATM 信元重组为高一层的信息单位；后者与服务有关，可以完成的功能有信报标识和时钟恢复等。

4.信元类型

在 ATM 中有以下几种信元类型：

- **空信元(物理层)**：为了使信元流的速率(ATM/物理边界)与传输系统可用的有效负载容量相匹配而在物理层插入或除去的信元。
- **有效信元**：没有头差错的信元或已由头差错控制进程修正过的信元。
- **无效信元(物理层)**：有头差错且尚未由头差错控制进程修正的信元。
- **指定的信元(ATM 层)**：使用 ATM 层服务为应用提供服务的信元。
- **非指定的信元(ATM 层)**：尚未指定的信元。

10.4.4.2 ATM 层

1. 信元结构

图 10.27 表示 ATM 的信元结构。字节是按递增顺序发送，从第 1 个字节开始；字节中的位是按递减顺序发送，1 从第 8 位开始。

2.信元头的结构

图 10.28 表示信元头的结构。

图 10.28 中符号说明如下：GFC 是总流控。VPI/VCI 是虚拟通路标识符/虚拟通道标识符(路由域)。PT 表示有效载荷类型。CLP 表示信元丢失优先权。HEC 表示信元头差错控制。

ATM 逻辑通道标识可分为两个层次的实体：虚拟通路 VP 和虚拟通道 VC，它们在信元头中标出(VPI: 8/12 位；VCI: 16 位)。在接口处，通信通道用整个域(VPI+ VCI)来标识。VPI 可使网络支持端点之间的半永久性连接。为了交换或多路复用/多路复用分解，多至 2^{16} 个 VC 可以作为一组来加以处理。ATM VP 交换或交叉连接只交换 VP。ATM 交换用(VPI + VCI)标识交换通信通道。VC 交换由呼叫建立过程控制。交叉连接由网络管理实体来控制。预先指定的 VCI 值，对元信令 VCI 为 0000000000000001，广播 VCI 为 0000000000000010。

信元优先权(CLP)设置 CLP 位以标识较低优先权的信元，在网络发生冲突时，首先被删去的就是这些较低优先权的信元。

有效负载类型 PT 指出在信元信息域中包含的是用户信息还是网络信息，这两种类型的信元可以混合于相同的 VC。

总流控 GFC 用于多用户组成情况，需保证每个用户都有一定的访问容量，这个机制尚在研究中。

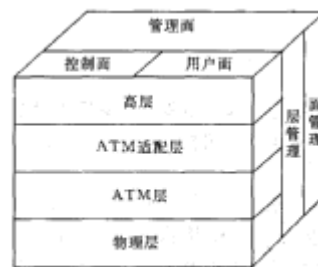


图 10.26 ATM 协议参考模型

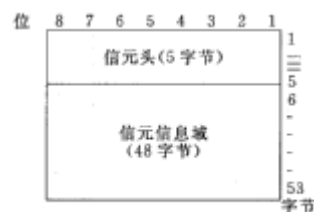


图 10.27 ATM 信元结构

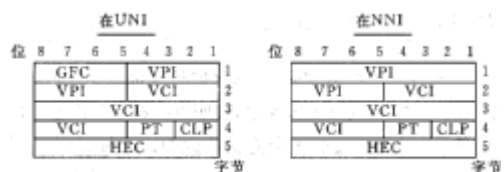


图 10.28 ATM 信元头结构

信元头差错控制 HEC 机制属于物理层。

3. ATM 层原语

在每个服务访问点 SAP, 通过 ATM 层及 AAL 层边界的服务数据单元 SDU 是信元信息域。定义的原语是: ATM-DATA-REQUEST, AAL 请求把与此原语相关的 ATM-SDU 传送给它的对等实体; ATM-DATA-INDICATION, 指示 AAL 与原语相关的 ATM-SDU 可用。这里没有列出所有 ATM 层原语。

10.4.4.3 ATM 适配层

1. ATM 适配层目的

ATM 层具有的服务性能独立于应用速率, 也独立于应用数据单元的结构。

在 ATM 信元流中会发生下列情况:

- 没有关于服务时钟的精确信息。
- 由于访问和网络排队使信元延时变化。
- 没有关于应用数据单元定界的信息。
- 由于网络拥挤而信元丢失或信元头差错且未校正。
- 由于信元头差错且未纠正使信元误插入。
- 传输错误可能影响信元有效负载。

AAL 必须从 ATM 层服务中恢复数据流, 使之满足上面的应用层的需求。AAL 协议与应用有关, 也就是每种应用可以有它自己的 AAL 协议。由于在应用层之间有一些共同点, 因此有限的 AAL 协议组是足够的。

2. AAL 服务分类

AAL 服务分类如图 10. 29 所示, 图中的符号说明如下: A 类表示线路仿真, B 类表示 VBR 视频, C 类表示文件传送, D 类表示无连接信报。

3. AAL 的子层

AAL 的子层包括汇聚子层(CS), 分段和重组子层(SAR)两层。CS 层负责为来自用户面的信息单元作分段准备, 以使这些分组再重组成原始状态。CS 层的主要功能是在 AAL-SAP 提供 AAL 服务。CS 完全与服务有关, CS 要求有控制信息, 控制信息附在用户信息上, 控制信息包括头和后缀, 或只是后缀, 控制信息的利用是由 AAL 服务的类型所决定的。CS 功能的例子有源时钟恢复、信元延时补偿和丢失信元的恢复等。

SAR 子层将来自汇聚子层的信元(称谓汇聚子层协议数据单元 CS-PDU)分段成 48 字节的载体, 或把来自 ATM 层的信元信息域内容组装成高层信息单位。在 SAR-PDU 的头和后缀包含有以下信息: 信元的顺序号; AAL-SDU 的开始与结束; 信元信息域的填充级别; 任选项, 用于信息域差错保护的 FCS。

4. AAL 类型

AAL 有 5 种类型, 在服务级别与 AAL 类型之间没有必然关系, 但有些 AAL 类型被建议成与某种服务级别有关。不同的 AAL 类型有不同的 SAR-PDU 结构。

提供的 A 级服务(CBR 服务)用 AAL 类型 1。对 AAL 类型 2 还没有一致的定义。提供的 D 级服务(无连接服务)用 AAL 类型 3/4。AAL 类型 5 没有规定明确的服务级别。

10.4.5 移动通信

1. 移动通信网

移动通信网是指支持通信的一方或双方可以在网内随意移动进行通信的网络。其中, 至少有通信一方处于运动中或暂时停留在某一非预定的位置上。移动通信网按用途、频段、制式、入网方式等不同, 可以有不同分类方法。如按使用对象分, 可分军用、民用要按用途和区域分, 可分陆上、海上、空间; 按经营方式分, 可分为公众网、专用网等等。

2. 全球移动通信系统

全球移动通信系统(GSM)是一个完整的数字移动通信标准体系。GSM 原意是指“移动通信特别小组”(Group Special Mobile), 它是 1982 年欧洲邮电管理委员会(CEPT)为开发第二代数字蜂窝移动系统而成立的机构。1982 年至 1983 年着重讨论基本原理; 1984 年成立了 3 个工作组分别研究业务、制定无线传输规范和定义网络结构及开放接口的信令规程; 1986 年成立 GSM 的永久研究机构; 1987 年确定 GSM 的主要无线传输技术; 1988 年成立欧洲电信标准协会(ETSI)GSM 转入这个新机构下; 1991 年 GSM 阶段 1 的技术规范全部完成; 同年底, 世界上第一个 GSM 网络开始运营; 1995 年 GSM 阶段 2 的技术规范全部完成。

3. 无线软件应用协议

无线软件应用协议(WAP)是以国际互联网上所采用的 HTTP/HTML 协议为基础, 针对无线移动通信的特性建立的通信协议, 是对小型显示界面、低功率、小内存、CPU 运算能力低的通信工具, 以及低带宽、延迟大和较不可靠的无线移动通讯网络进行修改设计而成的协议。这一标准的诞生是 WAP 论坛成员努力的结果, WAP 论坛是在 1997 年 6 月由诺基亚、爱立信、摩托罗拉和无线星球共同组成的。WAP 的目标是通过该技术

将 Internet 的大量信息及各种各样的业务引入到移动电话, PALM 等无线终端之中。

4. 个人通信业务/个人通信网

个人通信业务(PCs)是指利用对通信网络透明的个人电信号码, 经过传输网和智能网的接入、传输、交换等处理, 可实现任何用户(whoever)在任何时候(whenever)任何地方(wherever)能够同其他任何用户(whomever)实现用户自己预定的任何电信业务组合服务(whatever)的通信。个人通信的能力仅受通信终端网络能力及网络经营者所施加限定的约束。提供 PCs 的任何网络称为个人通信网(PCN)。

10.5 网络管理与网络安全、

网络管理是对计算机网络的配置、运行状态和计费等进行的管理。它提供了监控、协调和测试各种网络资源以及网络运行状况的手段, 还可提供安全管理和计费等功能。

信息安全是在分布式计算环境中对信息的传输、存储、访问提供安全保护, 以防止信息被窃取、篡改和非法操作。信息安全的 3 个基本要素是保密性、完整性和可用性服务, 在分布网络环境下还应提供鉴别、访问控制和抗否认等安全服务。完整的信息安全保障体系应包括保护、检测、响应、恢复等 4 个方面。

10.5.1 网络管理功能

在实际网络管理过程中, 网络管理应具有的功能非常广泛, 包括了很多方面。在 OSI 网络管理标准中定义了网络管理的 5 大功能: 配置管理、性能管理、故障管理、安全管理和计费管理, 这 5 大功能是网络管理最基本的功能。事实上, 网络管理还应该包括其他一些功能, 比如网络规划、网络操作人员的管理等。不过除了基本的网络管理 5 大功能, 其他的网络管理功能实现都与具体的网络实际条件有关, 因此只需要关注 OSI 网络管理标准中的 5 大功能。

下面针对每个功能进行具体的描述。

1.配置管理

自动发现网络拓扑结构, 构造和维护网络系统的配置。监测网络被管对象的状态, 完成网络关键设备配置的语法检查, 配置自动生成和自动配置备份系统, 对于配置的一致性进行严格的检验。

- **配置信息的自动获取:** 在一个大型网络中, 需要管理的设备是比较多的, 如果每个设备的配置信息都完全依靠管理人员的手工输入, 工作量是相当大的, 而且还存在出错的可能性。对于不熟悉网络结构的人员来说, 这项工作甚至无法完成。因此, 一个先进的网络管理系统应该具有配置信息自动获取功能。即使在管理人员不是很熟悉网络结构和配置状况的情况下, 也能通过有关的技术手段来完成对网络的配置和管理。在网络设备的配置信息中, 根据获取手段大致可以分为 3 类: 第一类是网络管理协议标准的 MIB 中定义的配置信息(包括 SNMP 和 CMIP 协议); 第二类是不在网络管理协议标准中有定义, 但是对设备运行比较重要的配置信息; 第三类就是用于管理的一些辅助信息。

- **自动配置、自动备份及相关技术:** 配置信息自动获取功能相当于从网络设备中“读”信息, 相应的, 在网络管理应用中还有大量“写”信息的需求。同样根据设置手段对网络配置信息进行分类: 第一类是可以通过网络管理协议标准中定义的方法(如 SNMP 中的 set 服务)进行设置的配置信息; 第二类是可以通过自动登录到设备进行配置的信息; 第三类就是需要修改的管理性配置信息。

- **配置一致性检查:** 在一个大型网络中, 由于网络设备众多, 而且由于管理的原因, 这些设备很可能不是由同一个管理人员进行配置的。实际上, 即使是同一个管理员对设备进行的配置, 也会由于各种原因导致配置一致性问题。因此, 对整个网络的配置情况进行一致性检查是必需的。在网络的配置中, 对网络正常运行影响最大的主要是路由器端口配置和路由信息配置, 因此, 要进行一致性检查的也主要是这两类信息。

- **用户操作记录功能:** 配置系统的安全性是整个网络管理系统安全的核心, 因此, 必须对用户进行的每一配置操作进行记录。在配置管理中, 需要对用户操作进行记录, 并保存下来。管理人员可以随时查看特定用户在特定时间内进行的特定配置操作。

2.性能管理

过滤、归并网络事件, 有效地发现、定位网络故障, 给出排错建议与排错工具, 形成整套的故障发现、告警与处理机制。

- **性能监控:** 由用户定义被管对象及其属性。被管对象类型包括线路和路由器; 被管对象属性包括流量、延迟、丢包率 CPU 利用率、温度、内存余量。对于每个被管对象, 定时采集性能数据, 自动生成性能报告。

- **阈值控制:** 可对每一个被管对象的每一条属性设置阈值, 对于特定被管对象的特定属性, 可以针对不同的时间段和性能指标进行阈值设置。可通过设置阈值检查开关控制阈值检查和告警, 提供相应的阈值管理和溢出告警机制。

- **性能分析**: 对历史数据进行分析、统计和整理, 计算性能指标, 对性能状况作出判断, 为网络规划提供参考。阈值可视化的性能报告: 对数据进行扫描和处理, 生成性能趋势曲线, 以直观的图形反映性能分析的结果。

- **实时性能监控**: 提供了一系列实时数据采集、分析和可视化工具, 用以对流量、负载、丢包、温度、内存、延迟等网络设备和线路的性能指标进行实时检测, 可任意设置数据采集间隔。

- **网络对象性能查询**: 可通过列表或按关键字检索被管网络对象及其属性的性能记录。

3.故障管理

采集、分析网络对象的性能数据, 监测网络对象的性能, 对网络线路质量进行分析。同时, 统计网络运行状态信息, 对网络的使用发展作出评测、估计, 为网络进一步规划与调整供依据。

- **故障监测**: 主动探测或被动接收网络上的各种事件信息, 并识别出其中与网络和系统故障相关的内容, 对其中的关键部分保持跟踪, 生成网络故障事件记录。

- **故障报警**: 接收故障监测模块传来的报警信息, 根据报警策略驱动不同的报警程序, 以报警窗口、振铃(通知一线网络管理人员)或电子邮件(通知决策管理人员)的形式发出网络严重故障警报。

- **故障信息管理**: 依靠对事件记录的分析, 定义网络故障并生成故障卡片, 记录排除故障的步骤和与故障相关的值班员日志, 构造排错行动记录, 将事件一故障一日志构成逻辑上相互关联的整体, 以反映故障产生、变化、消除的整个过程的各个方面。

- **排错支持工具**: 向管理人员提供一系列的实时检测工具, 对被管设备的状况进行测试并记录下测试结果以供技术人员分析和排错; 根据已有的排错经验和管理员对故障状态的描述给出对排错行动的提示。

- **检索/分析故障信息**: 浏览并且以关键字检索查询故障管理系统中所有的数据库记录, 定期收集故障记录数据, 在此基础上给出被管网络系统、被管线路设备的可靠性参数。

4. 安全管理

结合使用用户认证、访问控制、数据传输、存储的保密与完整性机制, 以保障网络管理系统本身的安全。维护系统日志, 使系统的使用和网络对象的修改有据可查。控制对网络资源的访问。

安全管理的功能分为两部分, 首先是网络管理本身的安全, 其次是被管网络对象的安全。

网络管理过程中, 存储和传输的管理和控制信息对网络的运行和管理至关重要, 一旦泄密、被篡改和伪造, 将给网络造成灾难性的破坏。

网络管理本身的安全由以下机制来保证:

- 管理员身份认证采用基于公开密钥的证书认证机制; 为提高系统效率, 对于信任域内(如局域网)的用户, 可以使用简单口令认证。

- 管理信息存储和传输的加密与完整性 Web 浏览器和网络管理服务器之间采用安全套接字层(SSL)传输协议, 对管理信息加密传输并保证其完整性; 内部存储的机密信息, 如登录口令等, 也是经过加密的。

- 网络管理用户分组与访问控制, 网络管理系统的用户(即管理员)按任务的不同分成若干用户组与不同的用户组中有不同的权限范围, 对用户的操作由访问控制检查, 保证用户不能越权使用网络管理系统。

- 系统日志分析记录用户所有的操作, 使系统的操作和对网络对象的修改有据可查, 同时也有助于故障的跟踪与恢复。

网络对象的安全管理有以下功能:

- **网络资源的访问控制**: 通过管理路由器的访问控制链表, 完成防火墙的管理功能, 即从网络层(UP)和传输层(TCP)控制对网络资源的访问, 保护网络内部的设备和应用服务, 防止外来的攻击。

- **告警事件分析**: 接收网络对象所发出的告警事件, 分析与安全相关的信息(如路由器登录信息 SNMP 认证失败信息), 实时地向管理员告警, 并提供历史安全事件的检索与分析机制, 及时地发现正在进行的攻击或可疑的攻击迹象。

- **主机系统的安全漏洞检测**: 实时地监测主机系统的重要服务(如 WWW, DNS 等)的状态, 提供安全监测工具, 以搜索系统可能存在的安全漏洞或安全隐患, 并给出弥补的措施。

总之, 网络管理通过网关(即边界路由器)控制外来用户对网络资源的访问以防止外来的攻击; 通过告警事件的分析处理, 发现正在进行的可能的攻击; 通过安全漏洞检测来发现存在的安全隐患, 以防患于未然。

5.计费管理

对网际互连设备按 IP 地址的双向流量统计, 产生多种信息统计报告及流量对比, 并提供网络计费工具, 以使用户根据自定义的要求实施网络计费。

- **计费数据采集**: 计费数据采集是整个计费系统的基础, 但计费数据采集往往受到采集设备硬件与软

件的制约，而且也与进行计费的网络资源有关。

- **数据管理与数据维护：**计费管理人工交互性很强，虽然有很多数据维护系统自动完成，但仍然需要人为管理，包括交纳费用的输入、联网单位信息维护，以及账单样式决定等。

- **计费政策制定：**由于计费政策经常灵活变化，因此实现用户自由制定输入计费政策尤其重要。这样一个制定计费政策的友好人机界面和完善的实现计费政策的数据模型。

- **政策比较与决策支持：**计费管理应该提供多套计费政策的数据比较，为政策制定提供决策依据。

- **数据分析与费用计算：**利用采集的网络资源使用数据，联网用户的详细信息以及计费政策计算网络用户资源的使用情况，并计算出应交纳的费用。

- **数据查询：**提供给每个网络用户关于自身使用网络资源情况的详细信息，网络用户根据这些信息可以计算、核对自己的收费情况。一

10.5.2 网络管理协议

随着网络的不断发展，规模增大，复杂性增加，简单的网络管理技术已不能适应网络迅速发展的要求。以往的网络管理系统往往是厂商在自己的网络系统中开发的专用系统，很难对其他厂商的网络系统、通信设备软件等进行管理，这种状况很不适应网络异构互连的发展趋势。20 世纪 80 年代初期 Internet 的出现和发展使人们进一步意识到了这一点。研究开发者们迅速展开了对网络管理的研究，并提出了多种网络管理方案，包括 HEMS、SGMP、CMIS/CMIP 等。

IAB 最初制定的关于 Internet 管理的发展策略，其初衷是采用 SGMP 作为暂时的 Internet 管理解决方案，并在适当的时候转向 CMIS/CMIP。SGMP 是在 NYSERNET 和 SURANET 上开发应用的网络管理工具，而 CMIS/CMIP 是 20 世纪 80 年代中期国际标准化组织和 CCITT 联合制定的网络管理标准。同时，IAB 还分别成立了相应的工作组，对这些方案进行适当的修改，使它们更适于 Internet 的管理。这些工作组随后相应推出了 SNMP (Simple NetWork Management Protocol 1988) 和 CMOT (CMIP/CMIS Over TCP/IP 1989) 等网络管理协议，下面进行简单介绍。

10.5.2.1 SNMP

简单网络管理协议 (SNMP) 的前身是 1987 年发布的简单网关监控协议 (SGMP)。SGMP 给出了监控网关 (OSI 第 3 层路由器) 的直接手段 SNMP 则是在其基础上发展而来。最初，SNMP 是作为一种可提供最小网络管理功能的临时方法开发的，它具有以下两个优点：

- 与 SNMP 相关的管理信息结构 (SMI) 以及管理信息库 (MIB) 非常简单，从而能够迅速、简便地实现。
- SNMP 是建立在 SGMP 基础上的，而对于 SGMP，人们积累了大量的操作经验。

SNMP 经历了两次版本升级，现在的最新版本是 SNMPv3 0 在前两个版本中 SNMP 功能都得到了极大的增强，而在最新的版本中 SNMP 在安全性方面有了很大的改善，SNMP 缺乏安全性的弱点正逐渐得到克服。

10.5.2.2 CMIS/CMIP

公共管理信息服务/公共管理信息协议 (CMIS/CMIP) 是 OSI 提供的网络管理协议簇。CMIS 定义了每个网络组成部分提供的网络管理服务，这些服务在本质上是普通的 CMIP 则是实现 CMIS 服务的协议。

OSI 网络协议旨在为所有设备在 ISO 参考模型的每一层提供一个公共网络结构，而 CMIS/CMIP 正是这样一个用于所有网络设备的完整网络管理协议簇。

出于通用性的考虑 CMIS/CMIP 的功能与结构跟 SNMP 很不相同 SNMP 是按照简单和易于实现的原则设计的，而 CMIS/CMIP 则能够提供支持一个完整网络管理方案所需的功能。

CMIS/CMIP 的整体结构是建立在使用 ISO 网络参考模型的基础上的，网络管理应用进程使用 ISO 参考模型中的应用层。也在这层上，公共管理信息服务单元 (CMISE) 提供了应用程序使用 CMIP 协议的接口。同时该层还包括了两个 ISO 应用协议：联系控制服务元素 (ACSE) 和远程操作服务元素 (ROSE)，其中 ACSE 在应用程序之间建立和关闭联系，而 ROSE 则处理应用之间的请求/响应交互。另外值得注意的是，OSI 没有在应用层之下特别为网络管理定义协议。

10.5.2.3 CMOT

公共管理信息服务与协议 (CMOT) 是在 TCP/IP 协议簇上实现 CMIS 服务，这是一种过渡性的解决方案，直到 OSI 网络管理协议被广泛采用。

CMIS 使用的应用协议并没有根据 CMOT 而修改，CMOT 仍然依赖于 CMISE、ACSE 和 ROSE 协议，这和 CMIS/CMIP 是一样的。但是 CMOT 并没有直接使用参考模型中表示层实现，而是要求在表示层中使用另外一个协议——轻量表示协议 (LPP)，该协议提供了目前最普通的两种传输层协议 TCP 和 UDP 的接口。

CMOT 的一个致命弱点在于它是一个过渡性的方案，而没有人会把注意力集中在一个短期方案上。相反，许多重要厂商都加入了 SNMP 潮流并在其中投入了大量资源。事实上，虽然存在 CMOT 的定义，但该协议已

经很长时间没有得到任何发展了。

10.5. 2. 4 LMMP

局域网个人管理协议 (LMMP) 试图为 LAN 环境提供一个网络管理方案。LMMP 以前被称为 IEEE 802 逻辑链路控制上的公共管理信息服务与协议 (CMOL)。由于该协议直接位于 IEEE 802 逻辑链路层 (LLC) 上, 它可以不依赖于任何特定的网络层协议进行网络传输。

由于不要求任何网络层协议 LMMP 比 CMIS/CMIP 或 CMOT 都易于实现, 然而没有网络层提供路由信息 LMMP 信息不能跨越路由器, 从而限制了它只能在局域网中发展。但是, 跨越局域网传输局限的 LMMP 信息转换代理可能会克服这一问题。

10. 5.2. 5 简单网络管理协议

简单网络管理协议 (SNMP) 是最早提出的网络管理协议之一, 它一推出就得到了广泛的应用和支持, 特别是很快得到了数百家厂商的支持, 其中包括 IBM, HP, SUN 等大公司 and 厂商。目前 SNMP 已成为网络管理领域中事实上的工业标准, 并被广泛支持和应用, 大多数网络管理系统和平台都是基于 SNMP 的。

SNMP 的前身是简单网关监控协议 (SGMP), 用来对通信线路进行管理。随后, 人们对 SGMP 进行了很大的修改, 特别是加入了符合 Internet 定义的 SMI 和 MIB 体系结构, 改进后的协议就是著名的 SNMP。SNMP 的目标是管理互联网上众多厂家生产的软硬件平台, 因此 SNMP 受 Internet 标准网络管理框架的影响也很大。现在 SNMP 已经出到第 3 个版本的协议, 其功能较以前已经大大地加强和改进了。

SNMP 的体系结构是围绕着以下 4 个概念和目标进行设计的: 保持管理代理 (agent) 的软件成本尽可能低; 最大限度地保持远程管理的功能, 以便充分利用 Internet 的网络资源; 体系结构必须有扩充的余地; 保持 SNMP 的独立性, 不依赖于具体的计算机、网关和网络传输协议。在最近的改进中, 又加入了保证 SNMP 体系本身安全性的目标。

另外 SNMP 中提供了 4 类管理操作: get 操作用来提取特定的网络管理信息; get-next 操作通过遍历活动来提供强大的管理信息提取能力; set 操作用来对管理信息进行控制 (修改、设置); trap 操作用来报告重要的事件。

1. SNMP 管理控制框架

SNMP 定义了管理进程 (manager) 和管理代理 (agent) 之间的关系, 这个关系称为共同体 (community)。描述共同体的语义是非常复杂的, 但其句法却很简单。位于网络管理工作站 (运行管理进程) 上和各网络元素上利用 SNMP 相互通信对网络进行管理的软件统称为 SNMP 应用实体。若干个应用实体和 SNMP 组合起来形成一个共同体, 不同的共同体之间用名字来区分, 共同体的名字则必须符合 Internet 的层次结构命名规则, 由无保留意义的字符串组成。此外, 一个 SNMP 应用实体可以加入多个共同体。

SNMP 的应用实体对 Internet 管理信息库中的管理对象进行操作。一个 SNMP 应用实体可操作的管理对象子集称为 SNMP MIB 授权范围。SNMP 应用实体对授权范围内管理对象的访问仍然还有进一步的访问控制限制, 比如只读、可读写等。SNMP 体系结构中要求对每个共同体都规定其授权范围及其对每个对象的访问方式。记录这些定义的文件称为“共同体定义文件”。

SNMP 的报文总是源自每个应用实体, 报文中包括该应用实体所在的共同体的名字。这种报文在 SNMP 中称为“有身份标志的报文”, 共同体名字是在管理进程和管理代理之间交换管理信息报文时使用的。

管理信息报文中包括以下两部分内容:

- 共同体名, 加上发送方的一些标识信息 (附加信息), 用以验证发送方确实是共同体中的成员, 共同体实际上就是用来实现管理应用实体之间身份鉴别的。
- 数据, 这是两个管理应用实体之间真正需要交换的信息。

在第 3 版本前的 SNMP 中只是实现了简单的身份鉴别, 接收方仅凭共同体名来判定收发双方是否在一个共同体中, 而尚未应用前面提到的附加信息。接收方在验明发送报文的代理或管理进程的身份后要对其访问权限进行检查。

访问权限检查涉及到以下因素:

- 一个共同体内各成员可以对哪些对象进行读写等管理操作, 这些可读写对象称为该共同体的“授权对象” (在授权范围内)。
- 共同体成员对授权范围内每个对象定义了访问模式: 只读或可读写。
- 规定授权范围内每个管理对象 (类) 可进行的操作 (包括 get, get-next, set 和 trap)。
- 管理信息库 (MIB) 限制对每个对象的访问方式 (如 MIB 中可以规定哪些对象只能读而不能写等)。

管理代理通过上述预先定义的访问模式和权限来决定共同体中其他成员要求的管理对象访问 (操作) 是否允许。共同体概念同样适用于转换代理 (proxy agent), 只不过转换代理中包含的对象主要是其他设

备的内容。

2. SNMP 协议

SNMP 是一个异步的请求/响应协议，即 SNMP 的请求和响应之间没有必定的时间顺序关系，换句话说，SNMP 是一个非面向连接的协议。这样，SNMP 实体不需要在发出请求后立即等待响应的到来，因此 SNMP 响应也可能丢失或出现错误。

SNMP 中设计了 4 种基本协议交互过程。

- **管理进程从管理代理处提取管理信息：**管理进程通过 SNMP 和传输网络发送 get-request 给管理代理，请求中包括管理对象的标识符等参数；管理代理收到请求后返回相应内容的 get-response，响应中包括待提取的管理信息。

- **管理进程在管理代理的可见范围内遍历一部分管理对象实例：**管理进程通过 SNMP 和传输网络发送 get-next-request 给管理代理，管理代理收到后完成遍历的一次操作，用 get-response 将遍历结果返回给管理进程。

- **管理进程在管理代理中存储信息，即对管理代理的管理信息库 MIB 进行写操作（包括设置工作参数）：**管理进程发送一个 set-request 给管理代理，由管理代理完成 set 操作，然后用 set-response 返回操作结果。

- **管理代理主动向管理进程报告事件：**管理代理通过 SNMP 和传输网络将 trap 发送给管理进程，这个操作没有响应。

注意，上面的各个请求都是管理进程发给管理代理的，响应则都是由管理代理发给管理进程的。只有 trap 是无响应的、由管理代理单向发给管理进程。另外，请求、响应和 trap 的传输处理都要受“共同体”定义的限制，包括访问权限。

SNMP 协议是一个对称协议，没有主从关系。SNMP 上的管理进程和管理代理都可以得到 SNMP 完全相同的服务。

10.5.3 信息安全术语

1. 密码学

密码学是以研究数据保密为目的，对存储或者传输的信息

采取秘密的交换以防止第三者对信息的窃取的技术。被变换的信息称为明文(plaintext)，它可以是一段有意义的文字或者数据；变换过后的形式称为密文(ciphertext)，密文应该是一串杂乱排列的数据，从字面上没有任何含义。从明文到密文的变换过程称为加密(encryption)，变换本身是一个以加密密钥 k 为参数的函数，记作 $E_k(P)$ 。密文经过通信信道的传输到达目的地后需要还原成有意义的明文才能被通信接收方理解，将密文 C 还原为明文 P 的变换过程称为解密或者脱密(decryption)，该变换是以解密密钥 k' 为参数的函数，记作 $D_{k'}(C)$ 。密码学加密解密模型如图 10.30 所示。

在传统密码体制中加密和解密采用的是同一密钥，即 $k = k'$ ，并且 $D_{k'}(E_k(P)) = P$ ，称为对称密钥密码系统(symmetric key cryptography)。现代密码体制中加密和解密采用不同的密钥，称为非对称密钥密码系统(asymmetric key cryptography)，每个通信方均需要有 k, k' 两个密钥，在进行保密通信时通常将加密密钥 k 公开(称为公钥 public key)，而保留解密密钥 k' (称为私钥 private key)，所以也称为公共密钥密码系统(public key cryptography)。传统密码系统中最常见的算法有 DES, IDEA 等, DES(data encryption standard, 数据加密标准)算法是 IBM 开发的，并于 1977 年被美国政府采纳为非机密信息的加密标准，它的原始形式已经在 1995 年被攻破，但是修改后的形式仍然是有效的；IDEA(international data encryption algorithm, 国际数据加密算法)是 Lai 和 Massey 提出的，目前还没有发现有效的攻击方法。

密码学研究包含两部分内容：一是加密算法的设计和研究，另外一部分是密码分析，也就是密码破译技术。在密码学模型中，假设进行密码分析的攻击者能够对密码通信进行攻击，他能够被动地监听通信信道上的所有信息，我们称之为被动攻击；他还能够对通信信道上传输的消息进行截取、修改甚至主动发送信息，我们称之为主动攻击。攻击者与报文接收方的区别在于他不知道解密密钥，因此无法轻易将密文脱密还原为明文。

公共密钥方案较对称密钥方案处理速度慢，因此，通常把公共密钥与对称密钥技术结合起来实现最佳性能，即用公共密钥技术在通信双方之间传送对称密钥，而用对称密钥来对实际传输的数据加密、解密。另外，公钥加密也用来对对称密钥进行加密。

在现代密码学研究中，对于加密和解密算法一般都是公开的，对于攻击者来说，只要知道解密密钥就能够破译密文，因此，密钥设计成为核心，密钥保护也成为防止攻击的重点。对于密钥分析来说，对密钥

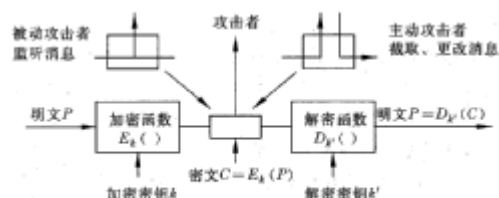


图 10.30 密码学模型

进行穷举猜测攻击是任何密码系统都无法避免的，但是，当密钥长度足够大并且足够随机时，就会使得穷举猜测在实际上变得不可能。例如，密钥长度为 256 位的加密算法，密钥空间为 2^{256} ，对应为 1077 量级，如果一台计算机每秒可以对密钥空间进行一亿次搜索，那么，全部搜索一遍的事件所需的时间以年为单位将大于 1062。如果密钥空间小或者分布具有一定可预见性，那么，攻击者就可能利用相关知识大大缩小搜索空间，从而破译密文。

2. 鉴别

鉴别是指可靠地验证某个通信参与方的身份是否与他所声称的身份一致的过程，一般通过某种复杂的身份认证协议来实现。身份认证协议是一种特殊的通信协议，它定义了参与认证服务的所有通信方在身份认证过程中需要交换的所有消息的格式和这些消息发生的次序以及消息的语义，通常采用密码学机制，例如用加密算法来保证消息的完整性、保密性。身份认证是建立安全通信的前提条件，只有通信双方相互确认对方身份后才能通过加密等手段建立安全信道，同时它也是授权访问(基于身份的访问控制)和审计记录等服务的基础，因此，身份认证在网络安全中占据十分重要的位置。尤其在分布式开放环境中，鉴别协议起着很重要的作用，其中系统的组成部分以及连接它们的网络可以跨越地理和组织的边界。

口令技术是常用的一种身份认证手段，使用口令存在的最大问题是口令的泄露。口令泄露可以有多种途径，例如登录时被他人看见，攻击者可能从计算机中存放口令的文件中读到，口令可能被在线攻击猜测出，也可能被离线攻击搜索到。所谓在线攻击，是指攻击者在联机在线状态下对用户口令进行的猜测攻击；所谓离线攻击，是指攻击者通过某些手段进行任意多数量的口令猜测，采用攻击字典和攻击程序，最终获得口令，这种离线攻击方法是 Internet 上常用的攻击手段。

另一种身份认证技术是采用身份认证标记，是用户携带用来进行身份认证的物理设备。常用的身份认证标记是磁卡 and 智能卡。磁卡存储着关于用户身份的一些数据，用户通过读卡设备向联网的认证服务器提供口令才能证明自己的身份。最简单的智能卡称作 PIN(personal identification number)保护记忆卡，PIN 是由数字组成的口令，只有读卡机将 PIN 输入智能卡后才能读出卡中保存的数据，这种卡比磁卡安全，可以存放一些秘密信息。另一种智能卡是加密挑战/响应卡，卡中有一个加密密钥，可使用该密钥进行加密和解密，但该密钥是无法被读出的，这种智能卡通常采用公开密钥算法，存储的是用户的私钥，可在离线状态下进行认证。在与计算机进行交互时首先递交代表自己身份的公钥证书，计算机验证证书的签发者后就获得了用户的公钥。

基于密码学原理的密码身份认证协议比基于口令或者地址的认证更加安全，而且能够提供更多的安全服务。各种密码学算法，如私钥算法、公钥算法和哈希算法都可以用来构造身份认证协议，它们各有特点。他们可以分为共享密钥认证、公钥认证和零知识认证等几类。共享密钥认证是基于通信双方共同拥有的但是不为别人知道的秘密，利用计算机强大的计算能力，以该秘密作为加密和解密的密钥。计算机可以存放高质量的密钥，进行复杂的加密、解密运算。计算机可以代表用户进行加密、解密操作，但是需要用户提供口令，将用户口令经过变换可以获得加密使用的密钥，或者是用口令来解密一个存放在某处的高质量密钥，例如用 PIN 获得存放 PIN 保护记忆卡中的高质量密钥。

3. Kerberos 鉴别

Kerberos 鉴别是一种使用对称密钥加密算法来实现通过可信第三方密钥分发中心(KDC)的身份认证系统。它是由美国麻省理工学院(MIT)为了保护 Athena 项目中的网络服务和资源开发的，Kerberos 版本 5 的协议已被 Internet 工程部 IETF 正式接受为 RFC 1510。

Kerberos 在学术界和工业界都获得了广泛的支持，被众多系统选作身份认证的基础平台。例如，开放软件基金会 OSF 开发的分布式计算环境 DCE 就是以 Kerberos 为身份认证平台的，而在国外应用最广泛的分布式文件系统 AFS(Andrew File System)也采用了 Kerberos 作为身份认证平台。目前各主要操作系统都支持 Kerberos 认证系统，例如，SUN 公司在其高端服务器产品 Windows NT 5.0 中也支持 Kerberos 系统。Kerberos 实际上已经成为工业界的事实标准。

Kerberos 使用对称密钥加密算法来实现通过可信第三方 KDC 的认证服务，它提供了网络通信方之间相互的身份认证手段，而且并不依赖于主机操作系统和地址。Kerberos 设计的目标是在开放网络上运行，不要求网络上所有主机的物理安全，同时还假设通过网络传输的包可以被任意截获、修改和插入。Kerberos 系统非常适合在一个物理网络并不安全的环境下使用，它的安全性经过了实践的考验。

4. 公钥基础设施

公钥基础设施(PKI)是在分布式计算系统中提供的使用公钥密码系统和 X.509 证书安全服务的基础设施。PKI 产品和服务允许使用者在网络上建立一个安全领域，在该领域中可以签发密钥和证书。PKI 支持

使用者在建立的安全领域中进行加密密钥和证书的使用和管理,提供密钥管理(包括密钥更新、恢复和托管)、证书管理(包括产生和撤消)以及安全政策管理等。PKI 还提供通过证书层次结构(certificate hierarchy)或者通过直接交叉证书(cross certificate)的方法在本地安全领域与其他安全领域之间建立相互信任的关系。

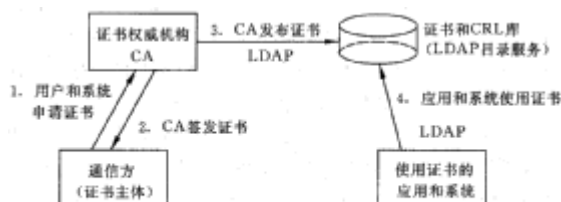


图 10.31 PKI 体系结构

图 10.31 表示了 PKI 的体系结构,除了证书以外,PKI 还包括其他几个组成成分。PKI 最基本的组成是证书的主体,它通常是用户,也可以是任何拥有公钥的一个公司、组织、系统或者应用。例如 Web 站点就可以成为证书主体,它通过 SSL 或者其他协议与浏览器建立安全通信信道。用户和应用软件系统可以成为证书的客体,它们和其他实体也将是证书的使用者。

CA 创建并签发证书。通常一个 CA 将为一个有限的用户团体签发证书,这样的用户团体常被称为安全领域(security domain)。CA 还维护并且发布证书撤消列表(certificate revoke list, CRL),当证书以及证书中的公钥失效时,通常采用 CRL 这种集中方式通知用户和应用。CA 通常将 CRL 发布在目录服务的某个位置甚至某个特定的 URL 上。

5. 数字签名

数字签名是通信双方在网上传换信息用公钥密码防止伪造和欺骗的一种身份签证。在传统密码中,通信双方用的密钥是一样的,既然如此,收信方可以伪造、修改密文,发信方也可以抵赖他发过该密文;若产生纠纷,谁也无法裁决谁是谁非。

对于公钥密码每个用户都有两个密钥,实际上有两个算法,如用户 A,一个是加密算法 E_A ,一个是解密算法 D_A 。若 A 要向 B 送去信息 m ,A 可用 A 的保密的解密算法 D_A 对 m 进行加密得 $D_A(m)$,再用 B 的公开算法 E_B 对 $D_A(m)$ 进行加密得: $C=E_B(D_A(m))$ 。B 收到密文 C 后先用他自己掌握的解密算法 D_B 对 C 进行解密得: $D_B(C)=D_B(E_B(D_A(m)))=D_A(m)$,再用 A 的公开算法 E_A 对 $D_A(m)$ 进行解密得: $E_A(D_A(m))=m$,从而得到了明文 m 。

由于 C 只有 A 才能产生,B 无法伪造或修改 C,所以 A 也不能抵赖,这样达到签名的目的。不是所有公钥系统都具有数字签名的能力,RSA 第一个提出这样的功能。

6. 访问控制

访问控制是指确定可给予哪些主体访问的权力、确定以及实施访问权限的过程。被访问的数据(如文件、数据报文、分组数据包、数据帧等)统称客体。能访问或使用客体的活动实体称作主体,如用户以及作为用户代理的进程、作业或任务等。访问控制一般都是基于安全政策和安全模型的。Lampson 提出的访问矩阵(access matrix)是表示安全政策的最常用的访问控制安全模型。该矩阵中列表示访问者(即主体),行表示被访问对象(即客体)。访问者对访问对象的权限就存放在矩阵中对应的交叉点上。

通常实际系统并不直接对矩阵进行操作,为节省存储空间采用访问控制表或者权力表。前一种方法是按照行来存储矩阵,在对象服务器上存储着每个对象的授权访问者及其权限的一张表,即访问控制表(access control list, ACL)。负责保护访问对象的程序称为引用监控器(reference monitor),它根据 ACL 的内容来判断是否授权某个访问者某些访问权限。后一种方法则是按照列来处理矩阵,每个访问者存储有访问权力表,该表包含了他能够访问的特定对象和操作权限。引用监视器根据验证访问表提供的权力表和访问者的身份来决定是否授予访问者相应的操作权限。

根据能够控制的访问对象粒度,我们可以将访问控制分为粗粒度(coarse grained)访问控制、中粒度(medium grained)访问控制和细粒度(fine grained)访问控制,这里并没有严格定义的区分标准,但是人们通常认为能够控制到文件甚至记录对象的访问控制可以称为细粒度访问控制;而只能控制到主机对象的访问控制被认为是粗粒度的。

目前很多计算机系统的安全都是采用 ACL 访问控制模型,分布式系统和网络系统也不例外。ACL 模型提供安全保密和完整性安全策略的基础。

10. 5.4 网络安全技术

10.5.4.1 网络安全层次模型

国际标准化组织在开放系统互连标准中定义了 7 个层次的网络参考模型,它们分别是物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。不同的网络层次之间的功能虽然有一定的交叉,但是基本上是不同的。例如,链路层负责建立点到点通信,网络层负责寻径,传送层负责建立端到端的通信信道。从安全角度来看各层能提供一定的安全手段,针对不同层次的安全措施是不同的。

需要对网络安全服务所属的协议层次进行分析,没有哪个单独的层次能够提供全部的网络安全服务,

每个层次都能做出自己的贡献。在物理层，可以在通信线路上采用某些技术，使得搭线偷听变得不可能或者容易被检测出。在数据链路层，点对点的链路可以采用通信保密机进行加密和解密，当信息离开一台机器时进行加密，而进入另外一台机器时进行解密。所有的细节可以全部由底层硬件实现，高层根本无法察觉。但是这种方案无法适应需要经过多个路由器的通信信道，因为在每个路由器上都需要进行加密和解密，在这些路由器上会出现潜在的安全隐患，在开放网络环境并不能确定每个路由器都是安全的。当然，链路加密无论在什么时候都是很容易而且有效的，也被经常使用，但是在 Internet 环境中并不完全适用。

在网络层，防火墙技术被用来处理信息在内外网络边界的流动，它可以确定来自哪些地址的信息可以或者禁止访问哪些目的地址的主机。在传送层，这个连接可以被端到端地加密，也就是进程到进程间的加密。虽然这些解决方案都有一定的作用，并且有很多人正在试图提高这些技术，但是它们都不能提出一种充分通用的办法来解决身份认证和不可否认问题。要解决这些问题必须在应用层。

应用层的安全主要是指针对用户身份进行认证并且建立起安全的通信信道。有很多针对具体应用的安全方案，它们能够有效地解决诸如电子邮件、HTTP 等特定应用的安全问题，能够提供包括身份认证、不可否认、数据保密、数据完整性检查乃至访问控制等功能。但是在应用层并没有一个统一的安全方案，通用安全服务 GSS-API 的出现是试图将安全服务进行抽象，为上层应用提供通用接口。在 GSS-API 接口下可以采用各种不同的安全机制来实现这些服务。

总结前面讨论，可以用图 10.32 来表示网络安全层次。

10.5.4.2 防火墙技术

防火墙是指建立在内外网络边界上的过滤封锁机制。内部网络被认为是安全和可信赖的，而外部网络(通常是 Internet)被认为是不安全和不可信赖的。防火墙的作用是：防止不希望的、未经授权的通信进出被保护的内部网络，通过边界控制强化内部网络的安全政策。

由于防火墙是一种被动技术，它假设了网络边界和服务，因此，对内部的非法访问难以有效地控制。防火墙适合于相对独立的网络，例如 Intranet 等种类相对集中的网络。

防火墙的主要技术类型包括网络级数据包过滤(network-level packet filter)，应用代理服务器(application-level proxy server)，状态检测防火墙。

1.包过滤防火墙

数据包过滤技术是在网络层对数据包进行分析、选择，选择的依据是系统内设置的过滤逻辑，称为访问控制表(access control table)。通过检查数据流中每一个数据包的源地址、目的地址、所用端口号、协议状态等因素，或它们的组合来确定是否允许该数据包通过。

数据包过滤防火墙的优点是速度快、逻辑简单、成本低、易于安装和使用、网络性能和透明度好。它通常安装在路由器上，内部网络与 Internet 连接，必须通过路由器，因此在原有网络上增加这类防火墙，几乎不需要任何额外的费用。

这类防火墙的缺点是不能对数据内容进行控制：很难准确地设置包过滤器，缺乏用户级的授权；数据包的源地址、目的地址以及 IP 端口号都在数据包的头部，很有可能被冒充或窃取，而非法访问一旦突破防火墙，即可对主机上的系统和配置进行攻击。

2. 应用层网关

应用层网关(application level gateways)技术是在网络的应用层上实现协议过滤和转发功能。它针对特定的网络应用服务协议使用指定的数据过滤逻辑，并在过滤的同时，对数据包进行必要的分析、记录和统计，形成报告。实际的应用网关通常安装在专用工作站系统上。

应用层网关防火墙和数据包过滤有一个共同的特点，就是它们仅仅依靠特定的逻辑来判断是否允许数据包通过。一旦符合条件，防火墙内外的计算机系统便可以建立直接联系，外部的用户便有可能直接了解到防火墙内部的网络结构和运行状态，这大大增加了非法访问和攻击的机会。

针对以上缺点，出现了应用代理服务(application-level proxy server)技术。

3.应用代理服务器

应用代理服务技术能够将所有跨越防火墙的网络通信链路分为两段。防火墙内外计算机系统间应用层的连接，由两个代理服务器之间的连接来实现，外部计算机的网络链路只能到达代理服务器，从而起到隔离防火墙内外计算机系统的作用。另外代理服务器也对过往的数据包进行分析、记录、形成报告，当发现攻击迹象时会向网络管理员发出警告，并保留攻击痕迹。

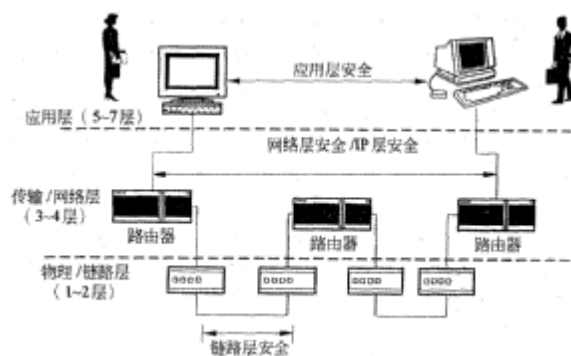


图 10.32 网络安全层次图

应用代理服务器对客户端的请求行使“代理”职责。客户端连接到防火墙并发出请求，然后防火墙连接到服务器，并代表这个客户端重复这个请求。返回时数据发送到代理服务器，然后再传送给用户，从而确保内部 IP 地址和口令不在 Internet 上出现。

本技术的优点是比包过滤防火墙安全，管理更丰富，功能提升容易，易于记录并控制所有的进/出通信，并对 Internet 的访问做到内容级的过滤。缺点是执行速度慢，操作系统容易遭到攻击。

4.状态检测防火墙

状态检测又称动态包过滤，是在传统包过滤上的功能扩展，最早由 CheckPoint 提出。传统的包过滤在遇到利用动态端口的协议时会发生困难，如 ftp，你事先无法知道哪些端口需要打开。而如果采用原始的静态包过滤，又希望用到此服务的话，就需要实现将所有可能用到的端口打开，这往往是个非常大的范围，会给安全带来不必要的隐患。状态检测通过检查应用程序信息(如 FTP 的 port 和 pass 命令)，来判断此端口是否需要临时打开，而当传输结束时，端口又马上恢复为关闭状态。

状态检测防火墙在网络层由一个检查引擎截获数据包并抽取出来与应用层状态有关的信息，并以此作为依据决定对该连接是接受还是拒绝。检查引擎维护一个动态的状态信息表并对后续的数据包进行检查。一旦发现任何连接的参数有意外的变化，该连接就被中止。这种技术提供了高度安全的解决方案，同时也具有较好的性能、适应性和可扩展性。状态检测防火墙一般也包括一些代理级的服务，它们提供附加的对特定应用程序数据内容的支持(如从 HTTP 连接中抽取出来 Java Applets 或 ActiveX 控件等)。

状态检测技术最适合提供对 UDP 协议的有限支持。它将所有通过防火墙的 UDP 分组均视为一个虚拟连接，当反向应答分组送达时，就认为一个虚拟连接已经建立。每个虚拟连接都具有一定的生存期，较长时间没有数据传送的连接将被中止。

状态检测防火墙克服了包过滤防火墙和应用代理服务器的局限性，不仅仅检测“to”或“from”的地址，而且也不要求每个被访问的应用都有代理。状态检测防火墙根据协议、端口及源、目的地址的具体情况决定数据包是否可以通过。对于每个安全策略允许的请求，状态检测防火墙启动相应的进程，可以快速确认符合授权流通标准的数据包，这使得本身的运行非常快速。

10.5.4.3 IP 层安全性

在 IP 加密传输信道技术方面 IETF 已经指定了一个 IP 安全性工作小组 IPSEC 来制定 IP 安全协议(IP security protocol, IPSP)和对应的 Internet 密钥管理协议(Internet key management protocol, IKMP)的标准。IPSP 的主要目的是使需要安全服务的用户能够使用相应的加密安全体制。该体制应该是与算法独立，可以自己选择和替换加密算法而不会对应用和上层协议产生任何影响。此外，该体制必须能支持多种安全政策，并且对其他不使用该体制的用户完全透明。按照这些要求，IPSEC 工作组使用认证头部(AH)和安全内容封装(ESP)两种机制，前者提供认证和数据完整性，后者实现通信保密。1995 年 8 月，Internet 工程领导小组(IESG)批准了有关 IPSP 的 RFC 作为 Internet 标准系列的推荐标准。除 RFC 1828 和 RFC 1829 外，还有两个实验性的 RFC 文件，规定了在 AH 和 ESP 体制中，用安全散列算法(SHA)来代替 MD5(RFC 1852)和用三元 DES 代替 DES(RFC 1851)。现在一些防火墙产品已经实现了 IP 层的加密，使用了 AH 或 ESP，支持 IPSEC，一些主要路由器厂商也称支持 IPSEC。IPSEC 技术能够在两个网络结点间建立透明的安全加密信道。

IP 层是非常适合提供基于主机的安全服务的。相应的安全协议可以用来在 Internet 上建立安全的 IP 通道和虚拟专网。例如，利用它对 IP 包的加密和解密功能，可以简捷地强化防火墙系统的防卫能力。

10.5.4.4 传输层安全性

由于 TCP/IP 协议本身非常简单，没有加密、身份认证等安全特性，因此要向上层应用提供安全通信的机制，就必须在 TCP 之上建立一个安全通信层次。传输层网关在两个通信结点之间代为传递 TCP 连接并进行控制，这个层次一般称作传输层安全。最常见的传输层安全技术有 SSL、SOCKS 和安全 RPC 等。

在 Internet 应用编程中，通常使用广义的进程间通信(IPC)机制来同不同层次的安全协议打交道。比较流行的两个 IPC 编程界面是 BSD Sockets 和传输层界面(TLI)，在 UNIX 系统 V 里可以找到。

在 Internet 中提供安全服务的首先一个想法便是在它的 IPC 界面加入安全支持，如 BSD Sockets 接口等，具体做法包括双向实体的认证、数据加密密钥的交换等。Netscape 通信公司遵循了这个思路，制定了建立在可靠的传输服务(如 TCP/IP 所提供)基础上的安全套接层协议(SSL)。SSL 版本 3(SSLv3)于 1995 年 12 月制定。SSL 分为两层，上面是 SSL 协商层，双方通过协商层约定有关加密的算法、进行身份认证等；下面是 SSL 记录层，它把上层的数据经分段、压缩后加密，由传输层传送出去。SSL 采用公钥方式进行身份认证，但是大量数据传输仍使用对称密钥方式。通过双方协商 SSI，可以支持多种身份认证、加密和检验算法。

同网络层安全机制相比,传输层安全机制的主要优点是它提供基于进程对进程的(而不是主机对主机的)安全服务和加密传输信道,利用公钥体系进行身份认证,安全强度高,支持用户选择的加密算法。这一成就如果再加上应用级的安全服务,就可以提供更加安全可靠的安全性能了。

10.5.4.5 应用层安全性

IP 层的安全协议能够为网络连接建立安全的通信信道,网络层(传输层)的安全协议允许为主机(进程)之间的数据通道增加安全属性。但它们都无法根据所传送内容不同的安全要求作出区别对待。如果确实想要区分一个个具体文件的安全性的要求,就必须在应用层采用安全机制。本质上,这意味着真正的(或许再加上机密的)数据通道还是建立在主机(或进程)之间,但却不可能区分在同一通道上传输的一个个具体文件的安全性要求。比如说,如果一个主机与另一个主机之间建立起一条安全的 IP 通道,那么两个进程间传输的所有报文都要自动地被加密。提供应用层的安全服务,实际上是最灵活的处理单个文件安全性的手段。例如,一个电子邮件系统可能需要对要发出的信件的不同段落实施数据签名。较低层的协议提供的安全功能一般不会知道任何要发出的信件的段落结构,从而不可能知道该对哪一部分进行签名。只有应用层是惟一能够提供这种安全服务的层次。

一般说来,在应用层提供安全服务有几种可能的做法,首先容易想到的就是对每个应用(应用协议)分别进行修改和扩展,加入新的安全功能。一些重要的 TCP/IP 应用已经这样做了。例如在 RFC 1421 至 1424 中 IETF 规定了私用强化邮件(PEM)来为基于 SMTP 的电子邮件系统提供安全服务。由于种种原因,Internet 业界采纳 PEM 的步子还是太慢,一个主要的原因是 PEM 依赖于一个既存的、完全可操作的 PKI 公钥基础)。PEM PKI 是按层次组织的,由下述 3 个层次构成:顶层为 Internet 安全政策登记机构(IPRA),次层为安全政策证书颁发机构(PCA),底层为证书颁发机构(CA)。

S-HTTP 是 Web 上使用的超文本传输协议(HTTP)的安全增强版本,由企业集成技术公司设计。S-HTTP 提供了文件级的安全机制,因此每个文件都可以被设成保密/签字状态。用作加密及签名的算法可以由参与通信的收发双方协商。S-HTTP 提供了对多种单向散列(Hash)函数的支持,如 MD2, MD5 及 SHA;对多种私钥体制的支持,如 DES,三元 DES, RC2, RC4 以及 CDMF;对数字签名体制的支持,如 RSA 和 DSS。

除了电子邮件系统外,另一个重要的应用是电子商务,尤其是信用卡交易。为使 Internet 上的信用卡交易安全起见,MasterCard 公司(与 IBM, Netscape, GTE 和 Cybereash 等公司一起)制定了安全电子付费协议(SEPP),Visa 国际公司与微软(和其他一些公司一道)制定了安全交易技术(STT)协议。同时,MasterCard, Visa 国际和微软已经同意联手推出 Internet 上的安全信用卡交易服务。他们发布了相应的安全电子交易(SET)协议,其中规定了信用卡持卡人用其信用卡通过 Internet 进行付费的方法。这套机制的后台有一个证书颁发的基础设施,提供对 X.509 证书的支持。SET 标准在 1997 年 5 月发布了第一版,它提供数据保密、数据完整性、对于持卡人和商户的身份认证以及与其他安全系统的互操作性。

上面提到的所有这些加入安全功能的应用都会面临一个主要的问题,就是每个这样的应用都要单独进行相应的修改,甚至对于同一应用使用不同类型的加密算法都需要进行修改。直接修改应用程序或其协议可能带来应用协议和系统的不兼容性,对用户来说使用起来并不方便。因此,需要有一个通用的安全服务接口,将底层安全服务进行抽象和屏蔽。这就是提供应用层安全的另一条思路,即通过中间件(middleware)层次实现所有安全服务的功能,通过定义统一的安全服务接口向应用层提供身份认证、访问控制、数据加密等安全服务。可以将中间件层次定位为在传输层与应用层之间的独立层次,与传输层无关。SSL 也可以看成是一个独立的安全层次,但是它与 TCP/IP 紧密捆绑在一起,因此不把它看作中间件层次。

由于应用程序要使用身份认证和密钥分发系统的 APT,这就要求有统一的 API,使得应用程序能不作修改就使用不同的身份认证和密钥分发系统提供的服务。能做到这一点,开发人员就不必再为增加很少的安全功能而对整个应用程序动大手术了。因此,认证系统设计领域内最主要的进展之一就是制定了标准化的安全 API,即通用安全服务 API(GSS-API)。GSS-API 可以支持各种不同的加密算法、认证协议以及其他安全服务,对于用户完全透明。目前各种安全服务都提供了 GSS-API 的接口,例如 Kerberos V5 就定义了自己的 GSS-API 接口。GSS-API(v1 和 v2)对于一个非安全专家的编程人员来说可能仍显得过于技术化了些,但德州 Austin 大学的研究者们开发的安全网络编程(SNP),把界面做到了比 GSS-API 更高的层次,使同网络安全性有关的编程更加方便了。

10.5.4.6 WWW 应用安全技术

随着 WWW 应用领域的扩大,安全和管理等问题日益受到重视。

解决 WWW 应用安全的方案需要结合通用的 Internet 安全技术和专门针对 WWW 的技术。前者主要指防火墙技术,后者包括根据 WWW 技术的特点改进 HTTP 协议或者利用代理服务器、插入件(plugin)、中间件等技术来实现的安全技术。

前面已经提到中间件技术可以提供标准的安全服务接口 GSS-API, 但是仍然要求对应用程序作一定的修改, 不能算是一个很完满的解决方案。如何使现有应用不加修改地使用中间件提供的安全机制并且对用户基本透明是一个问题。利用 WWW 技术中的代理技术, 我们可以巧妙地解决这个问题。这一技术需要修改过的 Web 服务器(也可能使用服务器的 plugin 功能)和一个额外的后台程序(本地安全代理)运行在客户方用来进行所需的安全交互。这种方案有很多诱人的地方: 服务器可以很可靠地验证用户身份而并不需要由各管理员自己来维护一个本地的用户口令文件; 它对于 Web 用户来说几乎是透明的, 利用了现有的网络安全措施, 有利于集中实现网络安全政策。它还允许加入安全审计、记录、报警等功能。这种方式并不依赖于 Web, 任何其他采用中间件技术的应用都可以从中获益。

10.6 Internet 与 Intranet

Internet 是全球最大的、开放的、采用 TCP/IP 协议、由众多网络互联而成的计算机互联网络。Intranet 是基于 Internet TCP/IP 协议, 使用 WWW 工具, 采用防止外界侵入的安全措施, 为企业内部服务, 并有连接 Internet 功能的企业内部网络。

10.6.1 Internet 路由结构

Internet 的路由结构是由少量而集中的路由器来保存全部关于目的站的信息, 而大量的外部路由器仅包含部分信息。因此, 对某个给定的路由器中的路由选择表只包含关于可能的目的站的部分信息。使用部分信息进行路由使各个路由结点能自动地改变本地路由, 但也可能引起不一致性。路由协议应具有足够的健壮性, 能很快地检测和纠正不一致的差错, 尤其重要的是, 路由协议应能限制这些差错的影响。

早期的 Internet 路由器大致可分成两类: 一类是少量的核心路由器, 它们由 Internet 网络控制中心 NOC 来控制; 另一类是大量的非核心路由器, 分别由各个群体控制。核心系统对全部可能的目的站提供可靠的、一致的、授权的路由, 实现了 Internet 的全球互联。被赋予 Internet 网络地址的网站必须将它的地址通知核心系统。核心系统内部互相通信, 确保了共享信息的一致性。由于有一个中央管理机构来监控这些路由器, 确保了系统的可靠性。核心路由器的体系结构如图 10.33 所示。

随着 Internet 的迅速发展, 核心路由器体系的结构在实现时出现了不少问题:

- Internet 已经发展成一个集中管理的远程主干网, 其拓扑结构极为复杂, 要确保所有核心路由器的一致性变得很困难。

- 并非每个网站都有一个核心路由器连至主干网, 因此需要其他的路由结构和协议。
- 由于所有的核心路由器要交互信息以保证路由信息的一致, 因此, 核心体系结构范围不能太广。

原始的 Internet 核心体系结构是在 Internet 仅有一个主干网的那个时期开发的。但是这种体系结构存在以下一些问题:

- 这种体系结构不能适应互联网扩展到任意数量的网点。
- 许多网点由多个局域网组成, 且用多个路由器互连。由于一个核心路由器在每个网点上与一个网络相连, 核心路由器就只知道那个网点中的一个网络的情况。
- 一个大型的互联网是独立的组织管理的网络的互连集合, 路由选择体系结构必须为每个组织提供独立的控制路由选择和访问网络的方法, 因此必须用一个单一的协议机制来构造一个由许多网点构成的互联网, 同时, 各个网点又是一个自治系统。

从路由选择的作用看, 由一个管理机构控制的网络和路由器的集合称为一个自治系统。在一个自治系统内的路由器可以自由地选择寻找路由、传播路由、确认路由, 以及检测路由一致性。按照这个定义, 核心路由器也组成一个自治系统。自治系统与主干网相连的互联网体系结构如图 10.34 所示。

为了能通过 Internet 到达隐藏在自治系统中的网络, 每个自治系统必须把网络的可达信息传播给其他自治系统。虽然在核心体系结构中可以把路由通知送给任一个自治系统, 但每个自治系统有必要将自己的信息传播给某个核心路由器。通常由自治系统中的一个路由器负责路由广播, 并直接和一个核心路由器交互信息。

总的来说, 一个大型的 TCP/IP 互联网有一个附加的结构来适应管理的界限。一个自治系统可自由地选择其内部的路由选择结构, 但必须收集其内部所有的网络的信息, 并责成若干个路由器将这些可达信息传送给其他自治系统。由于 Internet 使用核心体系结构, 每个与之相连的自治系统都要将可达信息送到 Internet 核心路由器。

10.6.2 Internet 地址

Internet 地址又称 IP 地址, 它能惟一确定 Internet 上每台计算机、每个用户的位置。Internet 上的每台计算机、每个用户都有一个惟一的地址以确定是谁和在何处, 区别在 Internet 上几百万台计算机、



图 10.33 核心路由器的体系结构



图 10.34 具有自治系统的互联网体系结构

成千上万的组织和上亿用户。

在 TCP/IP 协议中,规定分配给每台主机一个 32 位数作为该主机 IP 地址。在 Internet 上发送的每个数据包都包含了一个 32 位的发送方地址和一个 32 位的接收方地址。从概念上说,每个 IP 地址由两部分组成,即网络标识 netid 和主机标识 hostid。网络标识确定了该台主机所在的物理网络,主机标识确定了在某一物理网络上的一台主机。

IP 地址的层次结构具有两个重要特性:第一,每台主机分配了一个惟一的地址;第二,网络标识号的分配必须全球统一,但主机标识号可由本地分配,不需全球一致。

将 32 位 IP 地址分成两部分,需要确定如何进行分配。网络标识部分需要足够的位数,从而保证能给 Internet 上的每一个物理网络分配惟一的网路号。主机标识部分也需要足够的位数,以保证给物理网络上的每台主机分配惟一的主机号。由于 Internet 上的网络规模有很大区别,因此 IP 的编址方案将 IP 地址空间划分为 A、B、C 3 种基本类,每类有不同长度的网络标识和主机标识,如图 10.35 所示。

A 类地址分配给少数规模很大的网络,每个 A 类地址的网络有众多的主机,具体规定如下:32 位地址域中第一个 8 位为网络标识,其中第 1 位为 0,表示 A 类地址,其余 24 位均为主机标识,由该网的管理者自行分配。

B 类地址分配给中等规模的网络,每个 B 类地址的网络有较多的主机,具体规定如下:32 位地址域中前两个 8 位为网络标识,其中头两位为 10,表示 B 类地址,其余 16 位均为主机标识,由该网的管理者自行分配。

C 类地址分配给小规模的网络,每个 C 类地址的网络只有少量主机,具体规定如下:32 位地址域中前 3 个 8 位为网络标识,其中前 3 位为 110,表示 C 类地址,其余 8 位为主机标识,由该网的管理者自行分配。

图 10.36 为 3 类用户 IP 地址空间分布,每个 A 类网络有 1700 万台主机,共有 126 个 A 类地址网络,每个 B 类网络有 65000 台主机,共有 16000 个 B 类地址网络,每个 C 类网络有 254 台主机,共有 200 万个 C 类地址网络。

近年来,随着 Internet 用户数的急剧增加,可分配的 IP 地址空间也随之减少,尤其是 B 类地址空间大部分已分配给用户,C 类地址还有足够空间可分配。预计到 2000 年,原有的 32 位 IP 地址空间将大部分被分配完,现正在制定新的 128 位 IP 地址空间的方案。

IP 地址是 32 位数,用户很难读数和输入,因此用一种点分十进制表示法来表示。将 32 位数中每 8 位为一组,用十进制表示,利用点号分割各部分。最小值为 0,即一组内的所有位都为 0,最大值为 255,即组内所有位数都为 1,因此 32 位数用点分十进制表示的地址范围为 0.0.0.0 到 255.255.255.255。

根据上述规则,IP 地址的头 8 位,A 类为 0-127,B 类为 128-191,C 类为 192-223。还有两类不属于基本类的地址 D 类和 E 类。D 类用于广播传送至多个目的地址用,头 4 位标识为 1110,因此 IP 地址的头 8 位范围为 224-239。E 类用于保留地址,头 4 位标识为 1111,因此 IP 地址的头 8 位范围为 240-255。

10.6.3 Internet 域名系统

1. 域名系统原理

Internet 对每台计算机的命名方案称域名系统(DNS)。语法上,每台计算机的域名由一系列字母和数字构成的段组成。例如,清华大学计算机科学与技术系的域名为 cs.tsinghua.edu.cn,其中, cn 代表中国,edu 表示教育部门 tsinghua 表示清华大学,CS 代表计算机科学与技术系。由此看出域名是有层次的,域名中最末一部分通常都表示国家,最左边的部分代表该台计算机的名字。

域名和 IP 地址是一一对应的,域名易于记忆,用得更普遍。当用户要和 Internet 上某台计算机交换信息时,只需使用域名,网络会自动转换成 IP 地址,找到该台计算机。

域名系统规定了最高域的值,称 DNS 的顶层。当一个组织希望参加域名系统时,必须申请一个顶层域下的一个域名。一旦一个组织拥有一个已申请的域,就可以决定是否设置进一步的层次结构。一般来说,对于一个规模小的组织可以不再设置下一层的域,而对于规模大的组织就有必要设置多层结构。

由于域名是一个逻辑概念,所以它不必与物理地点相一致。而且对一个组织来说有不同的部门,可以根据部门的实际情况选择不同层次的域名构造。

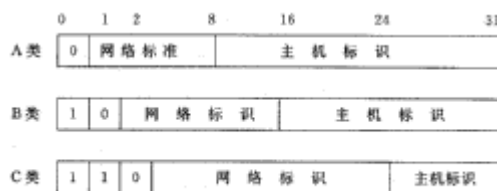


图 10.35 IP 地址编码

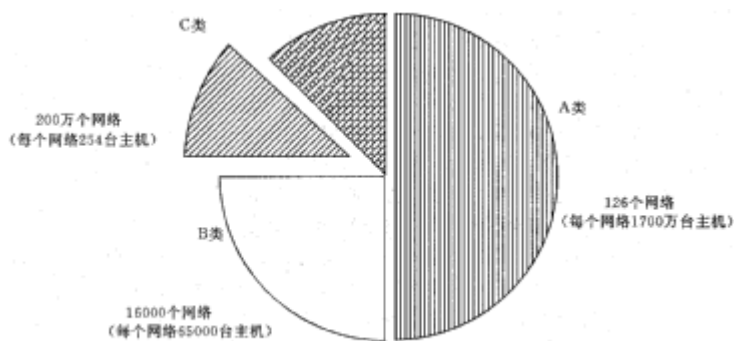


图 10.36 3 类用户 IP 地址空间分布

域名系统的一个特点是自治，即允许每个组织为计算机设置域名或改变这些域名。大多数具有 Internet 连接的组织运行一个域名服务器，称 DNS 服务器。每当应用需要将域名翻译为 IP 地址时，应用成为域名系统的一个客户。这个客户将待翻译的域名放在一个 DNS 请求信息中，并将这个请求发给 DNS 服务器。服务器从请求中取出域名，将其翻译为对应的 IP 地址，然后，在一个回答信息中将结果地址返回给应用。

2.域名的分级

早期的 Internet 使用非等级的名字空间来描述域名。整个 Internet 上的计算机使用名字的集合组成一个非等级的名字空间，每个名字由字符序列组成。中心网点，即网络信息中心(network information centre, NIC)，管理名字空间。非等级名字空间的主要优点是名字简单和短小。主要缺点是非等级名字空间由于技术和管理上的原因，不能推广至具有大量计算机的网络。由于名字取自单一的标识字符集，随着网点数的增加，潜在冲突亦随之增多；名字管理机构的管理工作负担也会随着网络规模的扩大而加重。

Internet 不采用集中命名的机制，而采用分布的名字空间管理机构，使名字和地址的映射任务分布执行。

名字空间的划分机制既要高效地支持名字映射，又要保证名字分配能自治控制。如果只考虑映射的效率，只需采用非等级名字空间，通过在多个映射机器之间划分名字来减少通信流量。如果只考虑管理的方便，只需让管理机构的授权容易，但会增加名字映射的开销和复杂性。

Internet 采用分级名字空间的管理，如同一个大的单位的管理。在最高层划分名字空间，并指定代理负责下一级的名字管理。例如，一个名字空间的表示为 local.site，site 是由中心管理机构授权的网点名，local 是受 site 网点控制的名字的一部分，点是分隔它们的分界符。原则上，可继续细分到足够小以便管理。

TCP/IP 互联网的分组命名方案不一定要根据物理位置来划分，可以按照组织结构或部门类型来划分，也就是说 TCP/IP 命名方案允许任意选取分级名字空间的管理结构，而与物理连接无关。

3. Internet 域名

DNS 有两个概念上独立的要点：一个抽象的，即指明名字语法和名字的授权管理规则；另一个是具体的，即指明一个分布计算系统的实现，它能高效地将名字映射到地址。

域名系统将域名的每一部分称为标号。例如，域名 cs.tsinghua.edu 含有三个标号。在域名中一个标号的任一后缀也称为域。上例中最低层的域是 cs.tsinghua.edu，表示清华大学计算机科学与技术系的域名；第二级域是 tsinghua.edu，表示清华大学的域名；最高层域是 edu，表示教育部门的域。

理论上，可以用任意标号规则定义一个抽象分级的名字空间。然而，大多数用户还是沿用 Internet 域名系统正式使用的分级标号规则。因为 Internet 域名系统命名方案能适应于多种组织结构，允许它们选择地理的或组织的命名分级。而且不需改动命名，就可将该组织的 TCP/IP 网方便地连到整个 Internet 上。

在概念上，顶层的名字可采用两种完全不同的命名分级，即地理的和组织的。前者是按照国家或地区来划分，后者是按组织的类型划分。

一个给定的名字可以映射到域名系统的多个项目。当客户机能解析名字时，指定所需对象的类型，并由服务器返回那种类型的对象。名字的语法不能决定它命名对象的类型，因此，如果只使用域名语法，就不能从对象的名字或对象的类型中区分出子域的名字。

域名方案应包括一个高效、可靠、通用的分布系统，实现名字对地址的映射。系统是分布的，由分布在多个网点的一组服务器协同操作解决映射问题。系统是高效的，大多数名字映射在本地操作，只有少数名字映射需要在互联网上通信。系统是通用的，因为它不限于仅使用机器名。系统是可靠的，单台计算机故障不会影响系统的正确运行。

10.6. 4 Internet 地址空间的扩展

Internet 协议第 4 版(IPv4)为 TCP/IP 协议簇和整个 Internet 提供了基本的通信机制。自 1970 年发布该协议，已沿用至今，说明该协议设计的灵活和功能强大。在这个期间，处理器的性能提高了两个数量级，典型的存储器容量提高了 32 倍，Internet 主干网带宽提高了 800 倍，Internet 上主机的数量已接近千万。总的来说，IP 也不断适应了这些变化。

虽然 IPv4 的设计是比较完善的，但随着技术和应用的发展，尤其是 Internet 用户的飞速增加，IP 地址空间很快会耗尽，迫切需要对 IPv4 进行更新。在最初设计 IP 时，一个 32 位地址空间是很充裕的。但现在 32 位 IP 地址空间已不能满足 Internet 的飞速增长。

除了地址空间需要扩展以外，还有一些其他因素也要求改变现有的 IP 设计，主要是日益增加的各种

新的应用需求。例如，实时话音和图像通信要求低的延迟，新版的 IP 应当提供一种机制，能为特定应用预留资源。又如一些新的应用需要安全通信，新版的 IP 应具有鉴别发送者的安全机制。

新版的 IP 已正式命名为 IPv6，它保持了 IPv4 许多成功的特点。IPv6 仍支持无连接传送；允许发送方选择数据报大小；要求发送方指明数据报在到达目的站前的最大跳数。IPv6 保留了 IPv4 中的大多数选项，包括分段和源站路由选择。

但是,IPv6 对协议细节作了许多修改。IPv6 的修改可分成以下 5 大类：

- **更大的地址空间。**这是 IPv6 最显著的变化。IPv6 将原来的 32 位地址空间增大到 128 位地址空间。IPv6 的地址空间是足够大的，在可预见的将来是不会耗尽的。
- **灵活的报头格式。**IPv6 使用一种全新的、不兼容的数据报格式。在 IPv4 中，使用固定格式的数据报报头，在报头中，除选项以外，所有的字段都在一个固定的偏移位置上占用固定数量的 8 位组数，而 IPv6 使用了一组可选的报头。
- **增强的选项。**IPv6 允许数据报包含可选的控制信息，包含了 IPv4 不具备的选项，提供新的功能。
- **支持资源分配。**IPv6 提供一种新的机制，允许对网络资源预分配，取代了 IPv4 的服务类型说明。这些新的机制支持实时话音和视像等应用，保证一定的带宽和延迟。
- **支持协议扩展。**IPv6 的一个很重要的改变是该协议允许新增特性，协议不需描述所有细节。这种扩展能力使协议能适应底层网络硬件的改变和各种新的应用需求。

10.6.5 Intranet 的定义和应用

Internet 是未来 NII 和 GII 的雏形，它对信息技术的发展、信息市场的开拓以及信息社会的形成起着十分重要的作用。近年来遍布在 Internet 上的万维网的建立和发展，大大充实了 Internet 的信息资源。基于图形的客户浏览器的开发，更加推动了万维网技术的发展。随着 Internet 用户数的迅速增长，TCP/IP 这个事实上的协议工业标准为各个计算机、网络制造厂商和广大用户普遍接受。

另一方面，在 20 世纪 90 年代，企业网络已经成为连接企、事业内部各部门并与外界交流信息的重要基础设施。基于局域网和广域网技术发展起来的企业网络技术也得到了迅速的发展，尤其是企业网络开放系统集成技术受到人们普遍重视。在市场经济和信息社会中，企业网络对企业的综合竞争能力的增强有着十分重要的作用。

但是，Internet 在如何满足企业的特定需求以及网络的安全性方面还不能完全满足企业网的需求。另一方面，由于历史发展的原因，企业网的开放性和外部世界的连接以及建立和使用企业网的方便性方面也存在许多不足之处，针对如何将 Internet 技术和现存的企业网络相结合，能更好地满足企业经营和发展需求的一种新型的企业网络—Intranet (内部网)的研究，近年来得到了很快的发展。

10.6.5.1 Intranet 的定义

Intranet 是基于 Internet TCP/IP 协议、使用环球网 WWW 工具、采用防止外界侵入的安全措施、为企业内部服务，并有连接 Internet 功能的企业内部网络。

从这个定义出发，可概括 Intranet 的若干要点如下：

- Intranet 是根据企业内部的需求而设置的，它的规模和功能是根据企业经营和发展的需求确定的。
- Intranet 不是一个孤岛，它能方便地和外界连接，尤其是和 Internet 的连接。
- Intranet 采用 TCP/IP 协议及相应的技术和工具，是一个开放的系统。
- Intranet 根据企业的安全要求，设置相应的防火墙、安全代理等，以保护企业内部的信息，防止外界侵入。
- Intranet 广泛使用环球网 WWW 的工具，使企业员工和用户能方便地浏览和采掘企业内部的信息以及 Internet 的丰富的信息资源。这些工具包括超文本标记语言，(hypertext markup language; HTML), 公共网关接口(common gateway interface, CGI)以及新的编程语言 Java 等。

10.6.5.2 Intranet 的应用

可以从两方面来讲述 Intranet 的应用：从技术上看 Intranet 的工具如何工作，它的功能以及这些工具如何适应特定的使用；从企业经营管理看，则是 Intranet 的工具如何使经营管理增值，以及对经营管理产生影响。本节着重讲述后者，其内容包括以下几方面。

1. 企业内部主页

企业内部主页包括以下几方面：工具和资源，包括搜索工具、索引和内容表、场地图、反馈意见、Internet 使用规则、Internet 资源、起始点、支持、指导和帮助、最新信息以及其他工具；目录、电话本以及组织结构图、历史和企业的宗旨；服务；组织的主页。

2. 通信处理

包括组织机构的通信和个人之间的通信两类通信处理，前者包括公务合作和部门之间通信，后者指用

于个人通信或工作小组内的通信工具。

组织机构的通信有以下 3 类：企业的快报、公告栏、新闻等，经营单位或部门通信，企业的信息库。

用于个人之间或小组内通信的最常用工具是电子邮件，它具有简单、明了、价廉等优点。还有一些 Intranet 的通信工具也日益普遍，主要是新闻组 (Newsgroups)，谈话 (Chat) 和视频会议系统 (Videoconferencing)。

3. 支持处理

支持处理用于企业内部，企业的客户只是间接的受益，它包括人事处理、财会处理、信息系统和技术支持、法律事务以及基础设施的开发和建设等。

财会处理也是 Intranet 的重要应用，很多财会制度和手续，包括收付款、财务账目、税收、审计等，还有财务报告、资产管理等。

信息系统和技术支持处理都可使用 Intranet。该系统具有以下功能：软件和应用的开发、分发，包括电子软件分发、软件模块库、应用开发准则和方法；用户手册和电子性能支持系统；技术支持和服务台；网络管理；信息和知识库；Internet 资源缓存。

法律事务处理可借助于 Intranet 查阅法律资源库以及合同草本，以加速书写合同文本的过程。

4. 产品开发处理

产品开发处理是企业经营的核心部分，它和企业的经营目标有关，同时也是企业专有的，为了竞争需要，一般都属于内部使用，不被外界共享。从内容分大致可分为研究开发和工程两部分。

根据不同的企业经营方式，可有多种不同方法使用 Intranet 进行研究开发。企业研究图书馆被经常使用，从那里可得到一些非公开的数据和文件、专利和商标、政府的经济和入口统计数据、工业数据、国际贸易信息，这些信息对市场研究和产品开发十分有用。各个企业愿意付一些费用，将有关信息放在企业内部 Web 上供内部使用。

产品开发处理的工程部分也有各种应用通过 Intranet 实现，如发表设计指南、工程参考材料等工程参考信息，出版和产品开发处理有关的专门文献，发表客户需求和反馈，发表一些竞争者的信息，生成项目的新闻组，出版研究报告和会议文章，发布工程项目的信息，发表产品规范和设计，组织对专门问题进行讨论，共享设计图纸和计算机辅助设计模型，今后还可使用基于 Intranet 的计算机辅助设计模型。

5. 运行处理

这也是企业经营的核心部分，包括采购、电子数据交换 (EDI)、库存、制造，以及专门的服务开发等。

企业将 Internet 上的产品目录移至 Intranet，供企业采购用。

传统的经营要花费大量纸张、时间和人力资源，在企业之间传递采购订单、发票、文件，采用电子数据交换可解决这类问题，且已在很多企业中使用。目前的趋势是使用 Intranet 在企业之间直接传送 EDI 的订单和文件。

从 Intranet 可检索库存信息，可传送生产计划、生产过程、质量统计报表。跨地区、跨国的企业可共享信息，以加速产品生产和提高产品质量。

6. 市场和销售处理

这也是企业经营的核心处理部分。由于竞争的原因，一般这些信息也不共享。

销售人员可随身带手提计算机，并随时和企业 Intranet 相连，获取有关销售需要的信息。

7. 客户支持

企业利用 Internet 做客户支持，如通过企业的 Web 主页给客户的信息，提供客户提问、反馈意见的通道，提供客户和产品开发者联系的通道，以改善产品质量，将企业内部数据库通过 Internet 供给客户使用。

下面是使用 Intranet 提供客户支持的一些方法：将从 Internet 得到的客户信息放在 Intranet 上，将 Internet 上的一些客户支持应用放到 Intranet 上，生成一些智能管理系统解决客户关心的问题，针对客户提出的问题发表有关解答，通过内部 Web 对客户进行培训，提供客户订购状态的信息，提供 FTP 服务器的服务，保持详细的维修记录作为分析和改进产品用。总之，Intranet 可以十分有效地帮助企业做好客户支持工作。

10.7 信息服务与网络应用

以 Internet 为代表的计算机网络是近十年来发展最迅速、应用最广泛的技术，它对当今社会政治、经济和文化产生了深远的影响，它正在改变人们的生活方式、工作方式和思维方式。

万维网 (WWW) 是基于客户/服务器方式的信息发现技术和超文本技术的结合。WWW 服务器把信息组织成分布式的超文本，这些信息结点可以是文本、子目录或信息指针。WWW 浏览程序为用户提供基于超文本传

输协议(HTTP)的用户界面。WWW服务器的数据文件由超文本标记语言(HTML)描述。HTML利用通用资源访问地址(URL)表示超媒体链接,并在文本内指向其他网络资源。

10.7.1 万维网

从各种网络信息服务在Internet所占的信息流量比例来看,前两位为WWW和FTP,Web的应用前景尤为光明,而其他的应用(如gopher)日渐衰落。事实上Web已成为很多人在网上查找、浏览信息的主要手段,而初露端倪的Intranet的信息服务则更多地基于Web网络应用技术。FTP虽比不上WWW的增长速度,但它一直是远程传送文件的主要方法。

万维网(WWW)是一种交互式图形界面的Internet服务,具有强大的信息连接功能,目前是Internet增长最快的网络信息服务。万维网使得成千上万的用户通过简单的图形界面就可以访问各个大学、组织、公司等最新信息和各种服务。商业界很快看到了其价值,许多公司建立了主页(homepage),利用Web在网上发布消息,并把它作为各种服务的界面,如客户服务、特定产品和服务的详细说明、宣传广告以及日渐增长的产品销售和服务。商业用途促进了万维网络的迅速发展。

使用WWW的浏览程序,如Lynx, Mosaic, Netscape, Hotjava等,Homepage的超文本链接将引导用户找到所需要的信息资源。超文本文档包含着一些借用标题、章节本身等构造文本的命令,从而允许浏览程序格式化为一种文本类型,以获得最佳的屏幕显示效果。有的浏览程序还可以自动调用其他应用程序,以显示特殊类型的文档。

10.7.1.1 浏览器

在Web的Client/Server工作环境中,Web浏览器起着控制的作用。Web浏览器的任务是使用一个起始URL来获取一个Web服务器上的Web文档,解释这个HTML,并将文档内容以用户环境所许可的效果最大限度地显示出来。当用户选择一个超文本连接时,这个过程重新开始:Web浏览器通过超文本连接相连的URL来请求获取文档,等待服务器发送文档,处理这个文档并显示出来。

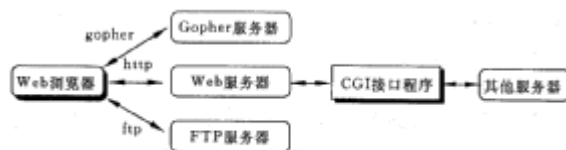


图 10.37 Web 浏览器的工作环境

图 10.37 表示了 Web 浏览器与 Internet 信息服务器之间的关系。与 Gopher 服务器信息交互时,Web 浏览器就像一个 Gopher 客户程序,采用 Gopher 协议。与 FTP 服务器交互时,采用 FTP 协议,更似 FTP 客户程序。与 Web 服务器交互时,采用 HTTP 协议。一些 Web 浏览器还可以采用 Z39.50 协议直接访问 Wais 服务器。Web 可以启动一个 Telnet 会话进程远程登录,可以作为一个新闻阅读器访问电子新闻。多数浏览器允许用户自由使用当前文本,如磁盘保存、邮件发送、打印、检索、查看文档的 HTML 编写等。

目前有适合不同平台、操作系统以及图形用户界面的 Web 浏览器,大致分为两类:线模式的和图形界面的。编写 HTML 时,最好手边有多种 Web 浏览器,以便测试 HTML 文档在不同环境下不同的显示效果。这里重点介绍一些 UNIX 的 Web 浏览器。要了解目前所有的浏览器列表请参考 <http://www.w3.org/hypertext/WWW/Client8.html>。

1. lynx

lynx 是 Kansas 大学为建立校园信息系统开发的全屏幕字符界面的浏览器。lynx 是功能完善的线模式浏览器,使用箭头键来浏览内在的 HTML 连接,支持书签和表格功能。lynx 的特点是:在交互状态,可以将文章发布到新闻组;在非交互状态,可以将 HTML 过滤为纯文本。

lynx 获取地址: <ftp://ftp2.cc.ukans.edu/pub/lynx/>, 其中包含已经编译好的适合不同平台的执行程序。lynx 运行中如出现问题或系统错误,可以与 lynx-help@ukanaix.cc.ukans.edu 及 lynx.bug@ukanaix.cc.ukans.edu 联系。

2. midasWWW

midasWWW 是由 slac 开发的基于 X-Windows 系统的浏览程序,其功能类似于 mosaic,但支持更多的嵌入图形,例如,它无需附加外部显示程序,可以直接显示 postscript, gif, tiff, jpeg 等格式的文档,它还能保存所有访问过的文档地址。

详细介绍请参考 <http://www.Midas.slac.stanford.edu/midasv22/introduction.html>。

作者地址: Tony Johnson, (tong-johnson@slac.stanford.edu)。问题求助地址: midasWWW@slac.stanford.edu。

3. Mosaic

Illinois 大学国家超级计算机应用中心开发的 Mosaic 程序,可以支持嵌入的 gif 和 xbrn 图形,其他的如 mpeg 视频影像、声音文件, Jpeg 图形 postscript 文档等,自动送到外部程序显示播放出来。Mosaic 也支持表格输入功能。

Mosaic 获取地址: <ftp://ftp.ncsa.uiuc.edu/Mosaic/>, 其中包含已经编译好的适合不同平台的执行程序。问题求助地址: mosaic-x@ncsa.uiuc.edu。

4. Netscape

Netscape 支持 HTML 3.0 版本, 具有比 Mosaic 更强、更全面的功能。支持 gif, jpeg 和 xbm 等格式的嵌入图形、图形背景设置, javascript, java applet 等。页面显示采取边传送文档边显示的方式, 增强了交互效果。

Netscape 获取地址: <http://home.netscape.com/>, 访问其中的下载页面, 包含已经编译好的适合不同平台的执行程序。问题求助地址: <http://home.netscape.com>。

5. Microsoft Internet Explorer

Microsoft 公司开发的基于 PC-Windows 的 Web 浏览器。支持 HTML 3.0 版本。

IE 获取地址: <ftp://ftp.microsoft.com>, 其中包含已经编译好的适合不同平台的执行程序。

问题求助地址: <http://www.microsoft.com>。

目前较为流行的 WWW 浏览器是: Netscape, IE, lynx。

10.7.1.2 Web 服务器

目前有 3 种主要的基于 UNIX 的 Web 服务器公用软件, 各有千秋。用户可以跟踪最新版本, 了解新增强功能和特点。另外, 也有基于其他操作系统的服务器软件, 不过, 基于 UNIX 的服务器软件最为灵活、有效, 其中用户广泛使用的是 NCSA Web 服务器。

1. NCSA Web 服务器

NCSA Web 服务器(httpd)是用 C 语言编写的, 程序小, 速度快, 与 HTTP/0.9, HTTP/1.0 协议兼容, 可以单独作为服务进程运行, 也可以设置在 inetd 中运行。NCSAhttpd 主要特点如下:

- 安全性。可以设置用户访问连接活动, 限制某些计算机访问, 支持用户认证。
- 灵活性。允许编写自己的服务接口来产生交互性的实时文档。当文件数据转移到别的目录或不同的服务器时, 无需重写 HTML 文件, 也不必辗转通知用户。另外, 支持在本机用户目录的 HTML 文档。

获取 NCSA Web 服务器地址: <ftp://ftp.ncsa.uiuc.edu/web> 或 <http://hoohoo.ncsa.uiuc.edu/docs/setup/precompiled.html>, 可以找到基于所有主要 UNIX 操作系统平台的编译好的 httpd 执行程序和配置文件。问题求助地址: httpd@ncsa.uiuc.edu。

2. CERN httpd

CERN 开发了最早期的 C 语言编写的 Web 服务器。支持表格输入、图形点击连接坐标映射、可执行的服务端接口程序实时更新文档、用于索引查找的 CGI 接口、用户认证等功能。CERN httpd 主要特点是提供 proxy 代理和缓存功能。proxy 服务器常常运行在防火墙的主机上, 为防火墙内的用户提供访问外部世界的接口。并且由于具有缓存功能, 将会提高用户访问的响应速度。

获取 CERN httpd 地址: <ftp://ftp.w3.org/pub/www/bin>, 其中包含编译好的各种平台的 httpd 执行程序和源程序。

3. Plexus httpd

Plexus Web 服务器软件是用 perl 语言编写的, 可扩展性好, 易于使用和更新, 但运行时开销较大。详细内容参考 <http://www.bsdi.com/server/doc/plexu8.html>。

10.7.2 动态 Web 文档与 CGI 技术

1. Web 文档的 3 种基本形式

根据文档内容的确定时间 Web 文档可划分成以下 3 种基本形式:

- **静态文档**。它是一个存储于 Web 服务器的文件, 静态文档由作者在写作时决定文档内容, 它的内容不会变化, 因此, 对静态文档的每次访问都得到相同结果。
- **动态文档**。它在浏览器访问 Web 服务器时创建, 没有预先定义的格式。当浏览器向服务器发出请求后, Web 服务器运行一个应用程序, 创建动态文档, 并返回给浏览器, 作为应答。动态文档的内容是变化的, 每次访问都要创建新的文档。
- **活动文档**。它不完全由服务器产生, 一个活动文档包括一个计算和显示的程序。当浏览器访问活动文档时, 服务器返回一个浏览器可以局部执行的程序副本, 活动文档可以和用户交互执行, 并不断改变显示。只要用户程序保持运行, 该文档可以不断地变化。

静态文档是一种排版语言, 易于编程和创建。在创建之后可一直使用, 且浏览器可以快速存取静态文档。静态文档的主要优点是简单、可靠、性能好。静态文档的主要缺点是灵活性差, 当信息变化时, 必须重新设计文档。因此, 对于变化频繁的文档, 不宜采用静态文档。

动态文档可用来显示天气预报、股市行情等时效性很强的信息。动态文档将任务放在服务器端, 浏览器采用与静态文档相同的方法访问文档, 两类文档都采用 HTML 语言编写。因此浏览区无法区分文档来自服务器的磁盘文件还是来自计算机程序。动态文档的主要缺点是创建的费用较高、访问的时间较长, 且浏

览器取得一个复制的文档后也不会再改变。

活动文档能够直接访问信息源和连续更改显示。与动态文档比较，它具有持续更改信息的能力。活动文档的主要缺点是创建和运行这类文档的费用高，安全性差。

2. 动态文档的实现

创建动态文档的任务在服务器端，支持动态文档的任务只需在服务器上实现。动态文档是静态文档的扩展，因此，管理动态文档的服务器也能处理静态文档的代码。

处理动态文档的 Web 服务器需要有以下 3 个特性：

- 服务器必须扩展，对来自浏览器的每次请求，能执行一个创建文档的应用程序，并将产生的活动文档返回给浏览器。

- 必须为每个动态文档写一个应用程序。

- 服务器使用设置信息来区分动态文档和静态文档。服务器使用设置信息和来自浏览器请求的 URL 来决定取出静态文档还是生成动态文档的应用程序。



图 10.38 CGI 与服务器交互的流程

3. 通用网关接口

构建动态 Web 文档广泛使用的技术是通用网关接口 (common gateway interface, CGI)。CGI 标准说明了服务器如何和应用程序交互作用，以实现一个动态文档，这种应用程序称 CGI 程序。

CGI 是服务器和 HTML 文件之间的接口程序，负责处理 HTML 文件与运行在服务器中的非 HTML 程序之间的数据交换。当用户输入查询信息或传送数据信息后，便激活一个 CGI 程序。该 CGI 程序又可调用操作系统下的其他程序，完成读者的查询任务，并将查询结果传给 CGI，通过 CGI 传给 Web 服务器。没有 CGI 程序用户不可能进行交互查询。

CGI 可以是一个编译的程序，或者是一个批处理文件，或者任何可执行的二进制文件。CGI 程序存放在 Web 服务器的 cgi-bin 的子目录下，必须要求系统管理员开放对 cgi-bin 目录的访问权。

Web 服务器调用 CGI 程序，CGI 又调用系统下的其他程序，而入口信息来自浏览器用户的输入，它们之间的关系如图 10.38 所示。首先在浏览器方发出请求，即用户输入查询条件，服务器接收到请求后，根据请求中提供的文件名到 cgi-bin 子目录中去执行 CGI 程序。这个 CGI 程序也许是访问数据库，也许是计算一个值，或者是调用系统下的某个程序。该程序返回执行结果给 CGI 程序，CGI 程序又将结果转换成 Web 服务器能识别的 HTML 格式 Web 服务器再将 HTML 格式表达的数据返回给提出请求的 Web 浏览器。经浏览器对 HTML 格式返回的数据进行处理后，就是呈现在用户面前的 CGI 执行结果。

CGI 实现交互查询的方法有两种，一种是基于文件的查询，另一种是使用 FORM。FORM 是 HTML 提供了一种标识，可以使用 FORM 为用户建立窗口，让用户在此窗口上输入信息。

CGI 提供通用程序，并且允许程序员选用大多数细节。例如，CGI 没有指定特别的编程语言。CGI 标准允许程序员选择一种语言，并且对不同的动态文档采用不同的语言。

10. 7. 3 活动 Web 文档和 Java 技术

1. 活动文档技术

随着 HTTP 和 Web 应用的发展，动态文档已明显不能满足发展的需要。由于动态文档一旦建立，其包含的信息内容也就固定下来，因而无法及时刷新屏幕显示，无法提供动画之类的功能。

有两种技术可用于屏幕显示的连续更新。一种技术称 Server push，是基于服务器的，在服务器上连续运行一个动态文档程序，以保证文档的不断更新。这种技术的缺点是造成过多的服务器开销和增加延迟，尤其是当很多客户同时要求 Server push，会使服务器过载。另一方面由于可用 CPU 和网络带宽资源的限制，会产生大的延迟。

另一种提供屏幕连续更新的技术称活动文档，是基于浏览器的，活动文档技术将主要工作交给浏览器在本地运行，服务器只提供活动文档。这种技术不需要使用大量的服务器 CPU 时间，它对网络带宽的需求也比较少。

与采用动态技术相比，采用活动文档技术的服务器开销少得多。因为浏览器和服务器将活动文档看成是静态文档今活动文档本身不包含运行所需的软件，大部分支持软件在浏览器上；活动文档还可以处理成压缩文档，以节约空间。

2. Java 技术

由美国 Sun Microsystem 公司开发的 Java 是一项用于创建和运行活动文档的技术。在 Java 中，使用术语 Applet 来描述活动文档程序，它区别于传统的计算机程序。Java 技术由 3 个关键成分组成，它们是程序设计语言、运行环境和类库。Java 包含一个新的程序设计语言，它可以用于编写传统的计算机程序，也可用于编写 Java Applet。Java 系统定义了一个运行 Java 程序所必需的运行环境。为了编写 Applet 更

容易, Java 提供了强大的类库支持, 可提高编程效率。

程序员在创建活动文档时使用源程序语言。Java 并不依赖于某种已有的语言, 它专门定义了一种新的程序设计语言, 称 Java 程序设计语言。它是一种高级语言, 通用性强, 类似于 C 月十, 是一种面向对象的语言, 对象实例在运行时动态创建。每个数据项都有一个确定的类型, 每个操作符的操作对象的类型都是固定的。在 Java 中, 类型检查不是在运行时做的, 而是在程序编译时完成的。Java 允许程序具有可并发执行的多个线程。

Java 技术定义了 Java 程序的运行环境。Java 程序是解释执行的, 允许多线程执行。运行系统包括一个套接字库, 用于程序对 Internet 的访问。Java 程序可以完成各种图形操作。当程序放弃了对某个对象的所有引用, 垃圾收集器自动将分配给该对象的内存予以收回。Java 语言和其运行系统设计成独立于计算机硬件, 因此, Java 程序可移植于不同计算机体系结构。

Java 类库的内容很广泛, 包括数十种类定义, 以及约 2000 个类成员函数。类库中的类涵盖了各种基本需求, 按功能可分为如下几大类: 图形操作, 低级的网络 I/O, 与 Web 服务器的交互, 支持 Applet 对 Java 运行系统本身的访问, 可以使用类库文件 I/O 的功能来读写本地计算机上的文件, 传统的数据结构, 事件捕获以及异常处理等。

10.7.4 网络化经济的新模式

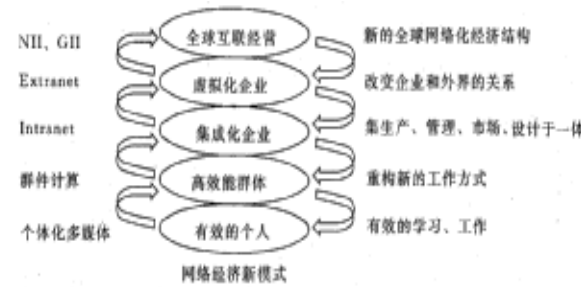


图 10.39

新的联网的组织结构不只是简单的面向处理的组织, 也不只是简单的基于工作组的结构, 而是根本上重新构思组织的功能和性质, 以及组织之间的关系, 这种新的组织被称为互联网络的企业(internetworked enterprise), 这是包括各个层次和经营功能的巨大的关系网。它使企业能有效地获得资源, 使企业变成由更小的分子簇组成, 且能很方便、很灵活地协同工作。互联网络的企业将扩展成虚拟的企业, 经常重构经营的关系, 如同 Internet 一样, 企业的每个成员可方便地介入, 且为企业做更多的贡献。

整个经济也将以同样的方式运行, 按照 Internet 的模式, 拆除各个企业之间的“墙”, 不断涌现出网络互联的企业、网络互联的政府、网络互联的教学、网络互联的医疗等等。这就需要建立 NII -21 世纪的信息高速公路。而每个基层组织需建立企业的信息基础设施 Intranet, 并且接入 NII, 这种新的基础设施将极大地增加新的经济活力。

这种联网的经济模式形成了人们活动的新的范式, 包括 5 个不同的层次, 如图 10.39 所示, 即有效的个人、高效能的工作组、集成的组织、扩展的企业以及全球互联的经营。每个层次都由相应的技术驱动, 并相应地改变现今的工作性质。

有效的个人是由个人化的多媒体技术驱动的。20 世纪 80 年代的个人计算机将计算能力交到了个人手里, 但早期的个人计算机在处理能力、用户界面等方面存在一定的局限性, 尤其是它是基于文本的用户接口。多媒体的出现大大改变了每个人的工作和学习方式, 提高了效率, 从根本上改变了在工业时代知识工作的性质。

高效能的工作组是由强大的协同工作计算技术驱动的, 联网的强大功能开辟了人类通信和合作的新途径, 不仅在办公室内, 而且可以超越时空限制。借助于网络, 知识工作正在变成协同工作, 其结果是改变了传统的经营处理方式, 重新构造新的工作方式, 如交互式的多媒体技术, 使人们采用能生成三维空间的计算机辅助设计系统; 采用并行工程方法设计和制造产品, 改变了传统的串行的工作流程。由于采用了这些新技术, 使波音 777 成为第一种没有物理模型和蓝图的飞机。

企业的信息基础设施是新的集成化企业的骨干, 它使企业组织从垂直的层次结构转变成水平结构, 使企业成为协同工作的组织, 企业不再是一些自动化的孤岛, 而是集生产、市场、管理、设计于一体的集成企业; 不仅是单纯的技术集成, 更重要的是人和经营的集成, 这种新的信息一技术结构使企业组织发生了根本性的变化。

通过企业间的计算(interenterprise computing)技术形成了扩展的企业, 使企业延伸到客户、供应商、合作伙伴, 甚至包括竞争对手, 从一个有形的企业变成一个虚拟的企业。这种变化的结果, 不仅可使企业降低成本、加速通信、提供及时的信息, 更重要的是改变了现存的人际交往和组织间通信, 建立起新的人际关系和组织间的关系, 改变着企业和外界的关系。

最后是建立全球网络互联的企业、商业、贸易, 这需要将这些经营活动移到信息高速公路上。微处理器技术的迅速发展, 基于 Internet 模式的公共网络的建立, 正在形成的数字化经济和网络化经济, 建立起了新的经济结构和关系。传统的经济活动模式将被取代, 新的模式正在建立, 这就是: 工作有效的个人,

基于高效能工作组的结构，集成的组织网络，有效的联接外部世界，最后进入公共的网络。它将改变产品和服务的生成、市场和分配的方式，最终形成一个支撑社会发展、改善生活品质的全新的系统。

10.7.5 电子商务

电子商务(electronic commerce, EC)是一种现代商业经营方法，可满足企业、商贸、消费者的需求，以达到降低成本、改进产品和服务质量、提高服务传递速度的目的。电子商务通过计算机网络实现信息、产品、服务的交换。电子商务涵盖的业务包括：信息交换、售前售后服务(提供产品和服务的细节、产品使用技术指南、回答顾客意见)、销售、电子支付(使用电子资金转账、信用卡、电子支票、电子现金)、运输(包括商品的发送管理和运输跟踪，以及可以电子化传送的产品的实际发送)、组建虚拟企业(组建一个物理上不存在的企业，集中一批独立的中小公司的权限，提供比任何单独公司多得多的产品和服务)、公司和贸易伙伴可以共同拥有和运营共享的商业方法等。从技术上，电子商务是多种技术的集合体，在网络环境下，实现交换数据(如电子数据交换、电子邮件)、获得数据(共享数据库、电子公告牌)、自动捕获数据、安全保障等。

因此，从技术和商务的角度来考虑，可以认为：电子商务就是要解决在信息时代一个企业如何做生意的问题。是网络经济中企业商务运作的一种重要模式。企业的商务活动通过计算机网络的整合，特别是 Internet, Extranet, Intranet 的整合，综合运用信息技术以协调和协作方式进行，处理贸易伙伴间的商务活动，使企业达到提升商业经营效率、降低成本、提高服务质量、一改进对市场变化快速反应能力、拓宽业务范围、提高竞争能力等一系列指标达到利润目标。同时，将极大地影响人们(消费者)的消费方式和生活方式。

可以简明地给出电子商务的几个要素或特征，即 2P + 3C。

- 以计算机网络，特别是 Internet, Extranet, Intranet 为环境或平台(舞台)(Platform)。
- 贸易伙伴以协调和协作方式(Collabraption 和 Cooperation)。
- 围绕贸易或商务这个主题(Commerce)。
- 对商务内容和信息计算机化处理(Content 和 Message)。
- 利润—达到企业生存和发展—最终目标(Profit) o

1. 电子商务通用框架

图 10. 40 是电子商务的通用框架。社会法规政策与隐私和电子文本、多媒体和网络协议的技术标准是两个十分重要的支柱。信息超高速公路基础设施、多媒体内容和网络发行基础设施、报文与信息发布基础设施以及公共商务服务基础设施是 4 个不同层次的平台，相当于传统商务活动中的路、车、货以及交易平台。

2. 电子商务的分类

可以从商务贸易活动的伙伴、规模与功能、商务协同过程、网络环境等电子商务的主要要素等方面对电子商务加以分类。

根据贸易商务活动的伙伴或贸易对象分类如下：

- 企业或商户与个人消费者之间的电子商务，即 B to C(Business to Consumer, B2C)。
- 企业与企业之间的电子商务，即 B to B(Business to Business, B2B)。
- 消费者与消费者之间的电子商务，即 C to C 或 P to P(Consumer to Consumer, C2C 或 Person to Person, P2P)，这往往需要有中介机构或网站提供服务，使消费者之间开展商务活动。
- 企业与政府之间的电子商务，即 B to G(Business to Government, B2G)。

按电子商务场所规模与业务功能分类如下：

- 电子商店或虚拟商店(Virtual Store)。
- 电子商场(Virtual Elctronic Merchant)。
- 电子商厦或电子街道(Virtual Electronic Mall)。
- 电子商城(Virtual Electronic Commerce City)。

按执行的电子商务的业务性质分类如下：

- 网络电子商情业务。
- 网上交易业务。
- 网络电子银行业务。

按照网络环境分类如下：

- 采用 EDI 专用网络。

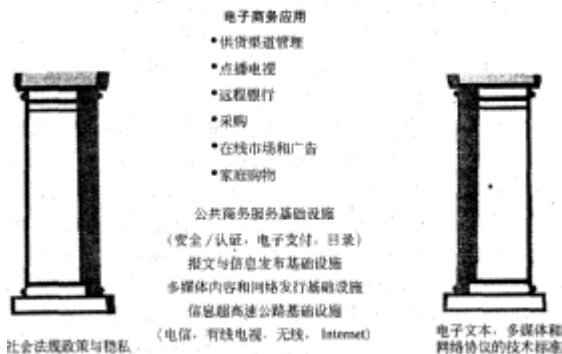


图 10.40 电子商务通用框架

- 在 Internet 上开展。
- 在 Extranet/Intranet 环境中进行。

10.8 网络工程

网络工程是根据用户单位的需求及具体情况,结合现时网络技术的发展水平及产品化程度,经过充分的需求分析和市场调研,从而确定网络建设方案,依据方案有步骤、有计划实施的网络建设活动。网络工程建设是一项复杂的系统工程,一般可分为网络规划和设计阶段、工程组织和实施阶段以及系统运行维护阶段。

10.8.1 网络规划

网络规划是在用户需求分析和系统可行性论证基础上确定网络总体方案和网络体系结构的过程。网络规划直接影响到网络的性能和分布情况,一项网络工程能不能既经济实用,又兼顾长远发展,网络规划是重要的一环。

1. 需求分析

需求分析可以采用自顶向下的分析方法,了解用户单位所从事的行业、该单位在行业内的地位及和其他单位的关系等。不同行业的用户,同行业内的不同单位,对信息网络的需求和它本身在信息网络中所承担的角色各不相同,不同角色的单位在进行网络规划建设时所采取的策略也不同。了解项目背景,有助于更好地了解用户单位建网的目的和目标。

对用户单位的建网目的和目标进行分析之后,应进行纵向的、更加细致的需求分析和调研,从而明确以下几个主要方面的情况。

- **地理布局。**了解用户单位的建筑物布局、入网站点的分布情况,记录下述信息:网络中心(或计算中心)及各级设备间的位置;用户数量及其位置;任何两个用户之间的最大距离;用户群组织,即在同一楼里或同一楼层里的用户,尤其注意那些地理上分散,却属于同一部门的用户;特殊的需求或限制,例如网络覆盖的地理范围内是否有道路、山丘,建筑物之间是否有阻挡物,电缆等介质布线是否有禁区,是否已存在可利用的介质系统等等。

- **用户设备类型。**包括终端(指没有本地处理能力的用户设备),个人计算机(指具有本地处理能力的单用户或多任务个人计算机),主机及服务器(具有本地处理能力的多用户设备),模拟设备(电话,传感器,视频设备)。

- **网络服务。**包括数据库和程序的共享;文件的传送,存取;用户设备之间的逻辑连接;电子邮件;网络互连;虚拟终端。

- **通信类型和通信量。**通信类型有以下几种类型:数据,视频信号,声音信号。

不同类型的通信量用不同的度量,一般数据的通信量用平均的以及高峰时每秒传送的位数表示。视频信号的通信量用电视通道数表示,每个通道占 6MHz 带宽,声音信号则用欧拉数表示。估计通信量比较好的做法是分析用户的网络应用,估计每个应用产生的通信量,再把各种通信量累计得出结果。最后应把通信量都表示为每秒传输的位数。通信量的估计还可以通过对已有网络系统的调研得到。

- **容量和性能。**网络容量是指在任何时间间隔,网络能承担的通信量。网络性能一般用经过网络的响应时间或端一端延时表示。通常,当网络的通信量接近其最大容量时,响应时间就变长,网络性能恶化。网络规划者只有掌握了网络上将负担的通信量以及用户响应时间的要求后,才能选择网络的类型及其配置,以便更好地满足需求。

- **网络现状。**如果要在已有的网络上规划建设新系统,那么了解用户单位现有网络的情况,尽可能在设计新系统的时候考虑旧系统的利用,既可保护用户投资,又能够使用户在系统的使用上有一个平滑过渡,节省培训时间和费用。

2. 系统可行性分析

可行性分析是结合用户单位的具体情况,论证建网目标的科学性和正确性。通过可行性分析可以提出一个解决用户问题的网络体系结构,它包括以下 4 方面内容。

- **传输:**传输方式用基带还是宽带传输,通信类型及通道数,通信容量,数据传输速度。
- **用户接口:**支持的协议,工作站类型,主机类型。
- **服务器:**类型,容量,协议。
- **网络管理能力:**网络管理,网络控制,网络安全。

对于网络体系结构的描述,在可行性论证阶段应尽可能用与厂家无关的功能术语。主要是要说明所提出的网络结构是怎样满足用户需求的。网络结构中可包含多个网络或网络段,例如包含多个局域网,或者既有局域网又有广域网。

• **系统可行性的另一个重要影响因素是造价**，而这一部分是要进行方案设计之后才能确定的。网络系统的方案往往不止一个，而且实施的效果和可靠性保证也不尽相同，用户的决策者可以从中选择出最佳方案。

10.8.2 网络设计

网络设计是根据网络规划及总体方案，对网络体系结构、子网划分、逻辑网络组成及网络技术和设备选型进行工程化设计的过程。

10.8.2.1 网络设计原则

网络设计过程中为了使方案可行且能保护用户投资，要注意以下一些原则：

- **成熟性**。采用成熟的技术，选用成熟的产品。如果片面追求新技术、新产品，会冒风险。
- **开放原则**。要保证与其他系统良好的互操作性，必须遵循开放原则。遵循国际、国内及相关行业的标准，采用开放技术、开放的体系结构、开放的系统组件和开放的用户接口。
- **安全可靠原则**。稳定可靠，具有高 MTBF 平均无故障时间)和低 MTBR 平均无故障率)，提供容错设计，支持故障检测和恢复，可管理性强。安全措施有效可信，能够在多个层次上实现安全控制。
- **先进原则**。应尽可能地利用先进而又成熟的技术，采用先进的设计思想、先进的软硬件设备及先进的开发工具。但要注意实用性，以获得较高的性能价格比。
- **完整性原则**。实现优化的网络设计、安全的数据管理、高效的信息处理、友好的用户界面。
- **可扩展性**。既能满足用户单位在人网机器数量上的增长需求，又能满足用户因增加新的业务、新的应用而引起的对带宽增加的需求，能够在规模和性能两个方向上进行扩展。

10.8.2.2 网络体系结构

网络系统的体系结构包括功能的分层及各层功能通信所遵守的协议。网络系统的体系结构也称为“层次与协议的集合”。网络系统设计的第一步就是选择网络体系结构，核心内容是决策应当采用的协议集合。

现时可供选择的协议体系有很多，一个现代化的网络要适应多厂商、多协议、互操作性的要求，在设计时可以采取过渡和共存的策略，使之既能适应发展的潮流，又能最大限度地保护已有的网络投资。

TCP/IP 已成为网络通信协议的事实上的国际工业标准，并已得到普遍的推广。与此同时，ISO/OSI 是公认的网络通信体系结构的国际标准，但至今尚未普遍流行，并缺少计算机厂商众多产品的支持。

10.8.2.3 子网规划

在很多种情况下需要对一个网络进行子网划分：同一网段上的联网结点太多，引起效率下降，同时给管理带来不便；不同部门间的信息需要隔离，尤其对于像财务部这样的特殊部门更是如此；某些特殊用户对于网络带宽有特殊要求，如进行视频会议的用户；同一单位需要运行多种网络操作系统(这也是采取多协议共存策略带来的必然结果)等等。

划分子网的方式有多种，经常使用的有：

- **通过物理连接来实现**。通过配线系统将相关的结点连接在同一子网上，这种方法需要更改配线系统，操作管理都不方便，除非特殊情况，一般不使用。
- **虚拟局域网(virtual LAN, VLAN)**。它是标准局域网的仿真，它允许数据的传送不受网络传输的物理限制。网络管理员使用管理软件将一组用户组成 VLAN，他们可以互相通信，就好像连在同一条电缆上一样，而事实上他们位于不同的物理 LAN 网段上。因为 VLAN 是基于逻辑连接而不是物理连接的，因此它们非常灵活。

划分子网的策略也有多种。其中之一是按部门划分，即不同的业务部门所属的子网不同。另一种方法是按任务来划分，如将同一项目组的结点成员划在一个子网上，尽管他们可能来自不同的业务部门。需要根据具体情况才能确定哪种方式最佳。

10.8.2.4 逻辑网络设计

1.设计网络拓扑结构

拓扑结构是指互联网络的结构映像图，它用来指示组成网络的网段、互联点及用户分布。网络拓扑结构说明的是网络的几何形状，而不是它的地理位置或技术实现。在拓扑结构设计阶段，要确定网段和互联点，明确网络的大小和范围，以及所需要的网络互联设备类型。

逻辑拓扑结构设计经常采用层次型网络设计方法，该方法采用分层的、模块化的模型来设计园区网和企业网。

现在常用的层次网络结构是 3 层结构：

- **核心层**：由高端路由器、交换机组成的网络中心。
- **分布层**：由路由器和交换机构成。

• **用户访问层**：由网络集线器、交换机和其他设备组成，用来连接接入网用户。

3 层层次结构如图 10.41 所示。3 层层次模型几乎适用于各种规模的网络设计，小到企业网、校园网，大到因特网。尽管该模型是在描述路由器层次时开发的，但该模型除了用于路由网络外，也可在交换式网络或桥接网络中使用。

2.网络地址分配和命名策略

在进行网络层地址分配时，下面这些规则可以参考：

- 在分配地址之前设计结构化寻址模型。
- 为寻址模型的扩充预留空间。
- 以分层方式分配地址块，以改进可伸缩性和可用性。
- 为了避免组或个人移动所带来的问题，应根据物理网络而不是组成成员分配地址块。
- 分配网络地址时使用有意义的编号。
- 为了最大限度满足灵活性，而又使配置最少，可以在用户端使用动态寻址。

• 为了使安全性和适应性得到最大满足，在 IP 环境中使用网络地址翻译(network address translation, NAT)技术，在单位内部使用私有地址。

在设计过程中，可以对多种资源进行命名，包括路由器、服务器、主机、打印机等。简短而有意义的名字可以简化网络管理，增强网络的性能和可用性。一个好的命名模型可以让用户通过名字而不是地址透明地访问服务。

3.选择桥接、交换和路由选择协议

桥接和交换方法的决策较为简单，因为选择范围很有限。在选择以太网网桥或交换机时，最好使用带生成树协议的透明网桥。对于标记环网，可以选择源路由选择网桥(source route bridging, SRB)、源路由透明(source route transparent, SRT)网桥和源路由交换(source route switching, SBS)网桥。为了连接标记网和以太网 LAN(或其他不同的 LAN)，可以使用翻译或封装网桥。

不同的路由选择协议其工作方式也不一样，有些路由选择向其他路由器发送完整的路由表；有些路由选择协议则传送关于直接连接链路状态的特定信息；还有一些路由选择协议不定期地向其他路由器发送握手信号，以维护与对等路由器的状态。

4.设计网络安全和管理策略

安全性和网络管理的设计应当在网络物理设计阶段开始前完成，以免影响物理设计。

安全性设计一般包括：

- 安全性需求分析。
- 确定确保网络安全的策略。
- 开发实现安全策略。
- 测试安全性，发现问题及时修正。
- 制定周期性的独立审计，阅读审计日志，响应突发事件，阅读最新的文献和代理警告，不断测试和培训，以及更新安全性计划和策略维护安全性。

网络管理设计包括：

- 确定网络管理的目标，即用户对性能管理、故障管理、配置管理、安全管理、记账管理等方面的需求及实现的可能性。
- 确定网络管理结构，包括：网络管理设备，即收集和存储管理信息的网络结点，可以是路由器、服务器、交换机、网桥、集线器等；网管代理，即驻留在管理设备上的网络管理软件；网络管理系统是一个具有高级图形、内存、外存和处理能力的功能强大的工作站，它运行网管软件，显示管理数据，监控和控制管理设备，并与网络代理通信。
- 确定网络管理工具和协议。

10. 8. 2. 5 网络技术和设备选型

主要是选择网络布线系统、物理层和数据链路层协议，以及网络互连设备(例如集线器，交换机和路由器)。

1. LAN 布线设计

布线系统是网络互联的物理基础，由各种传输介质、接插件组成。选择什么样的传输介质、采用什么样的布线方式，不仅关系到网络系统的可靠性、网络的性能，还直接影响网络的可用性和可管理性。网络设计中的其他部件在技术改变前一般只有几年的生命周期，而布线结构经常必须保持许多年。现在，网络

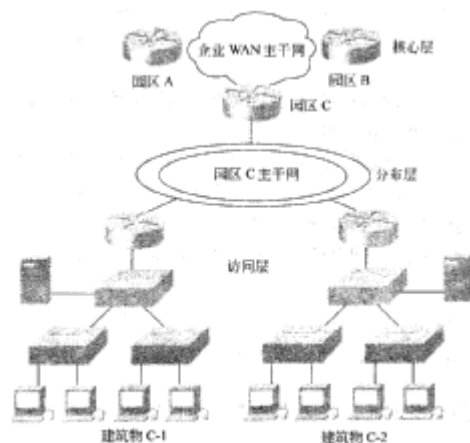


图 10.41 层次型拓扑结构

布线系统经常和大楼的电源电缆等系统一样作为建筑物的基础设施来建设，因此对于它的选型一定要仔细考虑，使之不仅能充分满足用户现阶段的使用，还要考虑将来足够长时间内的应用需求。

2. LAN 选型

(1) LAN 技术选型。

在 LAN 中常用的数据链路层技术主要有以太网、标记环 FDDI 和 ATM。ATM 在 WAN 和 LAN 中都可以使用。ATM 因其良好的伸缩性和对 QoS 的质量保证，而一度被认为将取代其他网络技术。但是 ATM 网络价格很贵，安装管理复杂。另一方面，以太网的发展也显示出它的良好扩展性，而且支持在 IP LAN/WAN 网络上 QoS，所以在园区级的网络设计中，除非因有视频会议、医学成像、语音、远程教育或其他混合了数据、视频和语音的需求从而需要高带宽、低延迟和小的甚至没有抖动的应用以外，一般不推荐在 LAN 中使用 ATM 技术。

(2) LAN 网络互连设备选型。

选择网络互连设备的条件一般包括下列内容：

- 端口数量。
- 处理速度。
- 延迟。
- 所支持的 LAN 技术 (10Mbps, 100Mbps, 1000Mbps 以太网, 标记环, FDDI, ATM)。
- 自动检测速度 (如 10Mbps 或 100Mbps)。
- 支持的介质 (如电缆)。
- 配置的简易性。
- 可管理性 (如支持 SNMP 和 RMON)。
- 费用。
- 平均无故障时间 (MTBF) 和平均修复时间 (MTTR)。
- 对热插拔部件的支持。
- 对冗余电源供应的支持。
- 技术支持的可用性和质量。
- 文档的可用性和质量。
- 培训的可用性和质量 (对复杂的交换机和路由器)。
- 供应商的信誉和生存能力。
- 确保设备运行的独立测试结果的可用性。

对于网桥，可以增加下列条件：

- 所支持的网桥技术 (透明网桥, 生成树算法, 源路由网桥, 远程网桥等)。
- 所支持的 WAN 技术。
- 学习网桥可学习的 MAC 地址的数目。
- 是否支持过滤。

对于交换机，可以增加下列条件：

- 每秒分组吞吐量 (对 ATM 来说为每秒信元吞吐量)。
- 是否支持直通式交换。
- 是否支持自适应直通式交换。
- 半双工/全双工操作的自动检测。
- 所支持的 VLAN 技术 (例如虚拟主干协议或交换机链路协议)。
- 对多媒体应用的支持 (如参与 Internet 组管理协议来控制广播分组的分散的能力)。
- 用于交换表、路由表 (如果该交换机有一个路由模块) 的存储器容量和协议例行程序所使用的存储器容量。

- 路由模块的可用性。

对路由器 (和有路由模块的交换机) 可增加下列条件：

- 所支持的网络层协议。
- 所支持的路由协议。
- 对多媒体应用的支持 (如支持 RSVP, IP 组播、受控负载和保证的服务)。
- 作为 ATM BUS, LECS 或 LES 的能力, 以及这些功能的性能。
- 对高级排队、交换及其他优化特征的支持。

- 对压缩的支持(如果支持, 它的压缩效率)。
- 对加密的支持(如果支持, 它的加密效率)。
- 对数据分组过滤和其他高级防火墙特性的支持。

3.远程访问设计

(1)远程访问技术选型。

点对点协议(PPP)是经常使用的远程访问技术, 它可以用在 ISDN ;模拟线路、数字租用线路等场合中。PPP 协议提供以下服务: 网络层协议多路复用, 链路配置, 链路质量测试, 链路选择协商, 认证, 报头压缩, 差错检测。PPP 支持两类验证:。密码验证协议(password authentication protocol, PAP)和请求握手验证协议(challenge handshake authentication protocol, CHAP)。CHAP 比 PAP 更加安全, 在大多数情况下, 建议使用 CHAP 方式。

模拟调制解调线路(公用电话网)是常用的远程访问接入技术, 当远程工作人员或移动用户每天上网时间不超过 2 个小时, 或者与总部之间的通信量很小时, 可以使用模拟调制解调线路。模拟调制解调器要花费长时间进行连接, 延迟时间长, 速度慢(当前模拟调制解调器的最高速度是 56Kbps)。

综合业务数字网(ISDN)为远程办公人员和远程办公室提供了一个性能价格比很高的远程访问解决方案。ISDN 部件包括终端、终端适配器(TA)、网络端结设备(NT)、线路端结设备和交换端结设备。

数字用户线路(DSL)与 ISDN 相似, 但 DSL 利用复杂的调制策略提供比 ISDN 更高的速率。DSL 对下行传输支持 32Mbps 的带宽, 对上行支持可达 16Kbps 到 1. 5Mbps。DSL 技术类型有多种, 包括非对称 DSL(ADSL); 高比特速率 DSL(HDSL); 极高比特速率 DSL(VDSL); 单线 DSL(SDSL), 也称为对称 DSL; 速率自适应 DSL(RADSL); ISDN DSL(IDSL); 用户 DSL(CDSL)。

电缆调制解调器是另一种可供选择的远程访问手段, 它使用有线电视(CATV)用的同轴电缆, 可以支持比电话更高的速率, 典型的电缆网络系统为下行提供 30~50Mbps 带宽, 为上行提供约 3Mbps 的带宽(有线电视信号可以使用额外的带宽, 这些数字指的是数据传送)。

(2)远程访问设备选型。

远程访问设备分为远程接入设备和中心站点设备。对于所要求速度比模拟调制解调器所提供的速度更高的用户, 远程访问可由电缆调制解调器, DSL 或含有 ISDN 或其他广域网的小路由器来完成。

远程访问服务的中心站点一般指的是可以接受多个远程站点的连接请求的远程访问服务器, 它允许多个用户同时与中心站点连接。远程访问服务器提供 5 种类型的服务: 远程结点服务, 终端服务, 协议转换服务, 异步服务, 拨号服务。

表 10.8 北美数字线路容量标准

信号	容量	DS-0 数量	通用名
DS-0	64kbps	1	信道
DS-1	1.544Mbps	24	T1
DS-1C	3.152Mbps	48	T1C
DS-2	6.312Mbps	96	T2
DS-3	44.736Mbps	672	T3
DS-4	274.176Mbps	4032	T4

4.广域网设计

(1)广域网带宽系统。

不论选择哪种广域网技术, 首先要确定广域网必须要提供的容量, 有关国际组织为此定义几个标准, 下面介绍最常用的几种:。

- **DS 系列:** 北美地区常用的标准, DS-1 和 DS-3 是最通用的容量, 如表 10.8 所示。
- **E 系列:** 由欧洲邮电委员会(CEPT)定义, 如表 10.9 所示。
- **同步数字线路(SPH):** 是一种在光纤上传送数据的国际标准, SDH 定义了 51.84Mbps 的传输标准速率, 称为同步传输信号 1(UP STS-1), 高速率的传输是基本 STS-1 的倍数。STS 速率与 SONET 光载波级别一样, 如表 10. 10 所示。

表 10.9 E 系列数字线路容量标准

信号	容量	E1 数量
E0	64kbps	N/A
E1	2.048Mbps	1
E2	8.488Mbps	4
E3	34.367Mbps	16
E4	139.264Mbps	64

表 10.10 SDH 标准

STS 速率	OC 标准	速 度
STS-1	OC-1	51.84Mbps
STS-3	OC-3	155.52Mbps
STS-12	OC-12	622.08Mbps
STS-24	OC-24	1.244Gbps
STS-48	OC-48	2.488Gbps
STS-96	OC-96	4.976Gbps
STS-192	OC-192	9.952Gbps

(2)广域网接入技术。

• **专线:** 即用户从电信公司租用的专门线路, 用于该用户企业网中两点之间的点对点连接。专线速度范围为 64kbps~45Mbps; 数据传输一般采用标准协议, 如 PPP 或 HDLC 等。

• **同步光纤网络(SONET):** 同步光纤网络是在光纤上高速同步传输分组或信元的物理层规范。SONET 使用 SDH 系统的 STS-1 作为它的基本构件。ATM 和分组交换网都可以基于 SONET。对于分组传输, SONET 网络常在数据链路层使用 PPP 协议, 在网络层使用 IP 协议。

• **帧中继**：帧中继运行在 OSI 参考模型的物理层和数据链路层，是一个高性能的广域网协议。帧中继出现在 20 世纪 90 年代初期，是对复杂的分组交换技术(如 X.25)的加强。与 X.25 适合于高出错率的物理电路相反，帧中继假定设备不易出错，从而使得帧中继比 X.25 更有效、更易于实现。帧中继比专线提供了更多的带宽分配选择，它包含动态带宽分配和拥挤控制，以支持突发的通信流量。由于帧中继的高效率、灵活的带宽支持和低延迟，帧中继有逐步取代 X.25 和专线的趋势。

• **ATM 广域网**：尽管 ATM 技术较为复杂，但 ATM 支持非常高的带宽需求(使用铜缆，ATM 可以以 T3 或大于 T3 的速度传输；使用光缆，ATM 支持的传输速度可以达到 OC-192 甚至更高)。ATM 还能满足不同的 QoS 需求，进行动态带宽分配和拥挤控制，因而对有更高带宽要求和高级 QoS 需求的用户来说，ATM 仍不失为广域网主干网的一种较好的选择。

(3)广域网设备及服务提供商的选择。

①选择广域网路由器。

企业网路由器应具有高吞吐量、高可用性和能优化链路利用率的高级特性。为企业广域网设计选择路由器和为局域网络设计选择路由器的情形类似，根据对网络流量的分析，选择能提供所需广域网接口的路由器，以支持所需的带宽。同时所选路由器拥有充足的内存和处理能力来转发数据并处理路由协议，而且有优化功能，如高级交换和排队技术、通信整形、随机早期测试(RED)和快速转发能力。

②选择广域网交换机。

从 20 世纪 90 年代中期开始，局域网交换技术逐渐应用到园区网络和广域网的设计中。广域网交换机可处理 ATM、帧中继和远程访问技术等多重服务，支持多种类型的网络传输，包括 TCP/IP 和其他局域网协议 X.25，SNA、视频、语音、电路仿真通信等。与旧的电信设备相比，它们提供的许多特性可相对节约开销，这些特性包括统计多路复用、动态带宽分配、语音活动检测(VAD)、语音压缩、重复模式抑制(RPS)等。

广域网交换机应能支持多种数据类型、接口和服务，并且可优化带宽利用。为支持不同类型的应用，WAN 交换机采用智能的排队处理算法。当遇到超负荷的情况时，广域网交换机用智能排队算法处理不同的传输类型，把应用的信元丢失降至最低。除了排队机制，缓冲区对于广域网交换机也是必需的。为了给不同类型的传输提供性能保障，当混合传输时，交换机必须有重新分配缓冲区的功能。

广域网交换机应具有在传输失败时快速自动重建路由的功能。除此以外，它们还应提供自动端到端连接管理。使用自动端到端连接管理，可以根据网络拓扑结构、负载、距离等因素来选择路由。

③选择广域网服务提供商。

广域网设计除了选择技术和设备，还必须选择服务提供商和(或)载波公司。

选择服务提供商的一个明显的标准就是服务费用。然而，费用不应是惟一的标准。

以下标准往往比费用更重要：

- 服务商提供的服务和技术范围。
- 服务商覆盖的地理范围。
- 服务商网络的性能和可靠性。
- 服务商提供的安全性。
- 服务商的技术支持水平。

在选择服务商时，还应尽可能了解服务商所提供网络的以下特性：

- 网络链接的物理路由。
- 网络内的冗余。
- 服务商依赖于其他服务商以获得冗余的程度。
- 网络承受超载的级别。
- 用来保证应用程序 QoS 需求的带宽分配机制。
- 使用的交换机类型及带宽分配和交换机优化特性。
- 网络老化的频率和典型原因。
- 保护网络不受侵犯的安全方法。
- 顾客专用数据保密的安全方法。
- 在地震、火灾、暴风雨、小行星与卫星碰撞或其他自然或人为灾难出现时的系统恢复计划。

许多服务商向客户提供服务级别协议(SLA)，该协议规定了服务项目和服务的评价及保证，以及预期的技术支持水平。

10.8.3 网络实施

网络实施是在网络设计的基础上进行设备的购买、安装、调试、培训和系统切换工作。

表 10.11 一个典型的工程实施计划

完成日期	主要阶段性成果
×月×日	设计完成,将设计文档的 beta 版分发给主管领导、部门经理、网络管理员和最终用户
×月×日	讨论设计文档
×月×日	最终分发设计文档
×月×日	广域网服务供应商在所有建筑物之间完成专用线的安装
×月×日	培训新系统的网络管理员
×月×日	培训新系统的最终用户
×月×日	完成建筑物 1 中的试验系统
×月×日	从网络管理员和最终用户那里搜集试验系统的反馈信息
×月×日	完成建筑物 2、3、4 和 5 的网络实施
×月×日	从网络管理员和最终用户那里搜集建筑物 2、3、4 和 5 网络系统的反馈信息
×月×日	完成其余建筑物内的网络实施
×月×日	监控新系统,验证其是否满足目标

网络实施包括以下步骤。

1.工程实施计划

在网络安装前,需准备一个工程实施计划,列出需安装的项目、安装费用、安装负责人等以便控制投资,按进度要求完成安装任务。工程计划必须包括在网络实施阶段的设备验收、人员培训、系统测试以及网络运行维护等具体事务的处理,必须控制处理所有可预知发生的时间并调动有关人员的积极性。表 10.11 为一个典型的工程实施计划。

2.网络设备到货验收

系统中要用到的网络设备到货后,在安装调试之前,必须先进行很好的功能和性能测试,以保证购买的产品能很好地满足用户需要。

3.设备安装

网络系统的工程安装和调试要由专门的技术人员负责。安装项目一般可分为:布线系统、网络设备、主机服务器、系统软件、应用软件等几个部分,不同部分应由专门的工程师进行安装调试。

4.系统测试

系统安装完毕,要进行系统测试。系统测试是保证网络安全可靠运行的基础。

5.系统试运行

系统调试完毕,进行试运行阶段。这一阶段,是验证系统在功能上、性能上是否达到预期目标的重要阶段,也是对系统进行不断调整直至到达用户要求的重要时期。

6.系统切换

系统经过一段时间的试运行,达到稳定可靠的水平,就可以进行系统切换了。系统切换指从原有人工或计算机系统上迁移到新的工作平台上工作,可以有 3 种切换方法:双运行方式(两种方式同时运行,以侧试新系统正确性)、逐步替代法(新系统逐步替代原有的网络系统)和直接切换法(停止旧系统,启动新系统),显然这 3 种方法其可靠性和成本各不相同,应视具体情况而定。

7.人员培训

对有关人员的培训是网络建设的重要一环,也是保证系统正常运转的一个重要因素。一个规模大、结构复杂的网络系统往往需要网络管理员来维护网络,协调网络资源的使用。

10.8.4 网络测试

网络测试是对网络设备、网络系统以及网络对应用的支持进行检测,以展示和证明网络系统能否满足用户在性能、安全性、易用性、可管理性等方面需求的测试。网络测试的实施一般包括以下环节:根据测试目的,确定测试目标;在对相关网络技术和实现细节透彻掌握的基础上,设计测试方案;建立网络负载模型;配置测试环境,包括测试工具的选择及必要的测试工具的研发;采集和整理数据;分析和解释数据;准确、直观、形象地表示测试结果。

网络测试包括网络设备测试、网络系统测试和网络应用测试 3 个层次。

1.网络设备测试

主要包括功能测试、可靠性和稳定性测试、一致性测试、互操作性测试和性能测试等方面。

功能测试验证产品是否具有设计的每一项功能。

可靠性和稳定性测试往往通过加重负载的办法来分析和评估系统的可靠性和稳定性。

网络产品不同于其他产品的最大特点是必须符合标准,不同的网络产品之间要能互操作。一致性测试验证产品的各项功能是否符合标准。如交换机对 IEEE 802.3, IEEE 802.3z, IEEE 802.1p, IEEE 802.1q, IEEE 802.3x 等的支持。

互操作性测试考察一个网络产品是否能在一个不同厂家的多种网络产品互连的网络环境中很好地工作。

性能测试的主要目标是分析产品在各种不同的配置和负载条件下的容量和对负载的处理能力,如交换机的吞吐量、转发延迟等。

典型的网络设备测试方法有两种:第一种方法是将设备放在一个仿真的网络环境中进行测试;第二种方法是使用专用的网络测试设备对产品进行测试。

2.网络系统和应用测试

网络系统测试除了普通意义上的物理连通性、基本功能和一致性的测试以外,主要包括网络系统的规划验证测试、网络系统的性能测试、网络系统的可靠性与可用性的测试与评估、网络流量的测量和模型化

等。

网络系统的规划验证测试主要采用的两个基本手段是模拟和仿真。模拟是通过软件的办法，建立网络系统的模型，模拟实际网络的运行。通过设定各种配置和参数模拟系统的行为，对系统的容量、性能以及对应用的支撑程度给出定量的评价。这对于大型网络的规划设计是不可缺少的环节。国外有支持不同网络技术或多种网络技术结合的模拟系统，售价十分昂贵。仿真是指通过建立典型的试验环境，仿真实际的网络系统。规划验证测试的目的在于分析所采用网络技术的可行性和合理性，网络设计方案的合理性，所选网络设备的功能、性能等是否能够合理地有效地支持网络系统的设计目标。

网络系统的性能测试是指通过对网络系统的被动监测和主动测量，确定系统中站点的可达性、网络系统的吞吐量、传输速率、带宽利用率、丢包率、服务器和网络设备的响应时间、哪些应用和用户产生最大的网络流量，以及服务质量等。此项工作同时可以发现系统的物理连接和系统配置中的问题，确定网络瓶颈，发现网络问题。测试设备记录一段时间内的网络流量，实时和非实时地分析数据。被动测量不干涉网络的正常工作，不影响网络的性能。主动测量向网上发送特定类型的数据包或网络应用，以分析系统的行为。

网络应用层次上的测试则主要体现在测试网络对应用的支持水平，如网络应用的性能和服务质量的测试等。例如部署基于 IP 的语言传输 VOIP 时，最直接的问题是网络中的交换机和路由器设备能否有效地支持语音传输，如网络能支持多大的语音流量、多少个语音通道；如果网络支持 VOIP，对网络的其他业务，特别是关键业务，会产生什么样的影响；网络是否支持服务质量 QoS。这些问题都需要通过网络测试来回答。

网络系统测试的核心工具是协议分析仪。这是一种专用的网络测试设备，它能够连接到网络上，产生并向网上发送数据、捕捉网上数据、分析数据。协议分析仪一般具有网络监测、故障查找、协议解码和流量产生等功能。

网络流量的测量和模型化对于分析网络性能和带宽的利用率，指导网络流量管理，开发高效的网络应用十分重要。这方面的工作主要有：

- 产生已知特征的流量，使该流量沿网络传播，最后回到测试仪。记录和分析流量特性的任何改变(如延迟漂移)。
- 对链路总体流量的测量和传输时间、吞吐量、带宽利用率等的分析。
- 分析特定流量的特征和提供的 QoS；收集一个时间段的测量数据进行分析，分析流量沿网络传播过程中流量特征的变化和网络流量的统计行为，建立流量模型。

第 11 章 计算机系统与配置

计算机系统由硬件和软件构成。硬件是计算机系统中的实际装置，是系统的基础和核心，一般由中央处理器(CPU)、存储器、输入/输出设备等组成，它以机器语言提供给程序员使用，机器语言即是指令系统。软件指的是操作系统、文本编辑程序、调试程序、汇编程序、各种高级语言的编译程序、数据库管理系统、文字处理系统、网络软件以及各种应用程序等，近年来不断有新的编程工具出现，并配以用户友好的屏幕显示，提供了更为方便的编程环境。

计算机系统的性能和价格是由软件和硬件共同决定的，硬件的结构和性能对程序处理的能力和速度影响很大。从原则上讲，解题必须具备的硬件可以很简单，某些操作可以通过软件来实现，但是这将花费更多的时间，有时是不允许的。如果扩充硬件，由硬件直接来完成这些操作，可以大大地提高处理速度。这说明了某些操作可以用软件也可以用硬件来完成，因此软硬件之间没有一成不变的分界面，而是受实际应用的需要以及系统性能价格比所支配。具有相同功能的计算机系统，其软、硬件之间的功能分配可以有很大差别。随着科学的进步，技术的发展，尤其是集成电路基本上符合摩尔定律，每隔 18 个月集成度提高 1 倍，价格只有原来的 1/20 促进了硬件技术的飞速发展，无论在体系结构和运算速度方面都有了很大提高。

系统分析员要担负起设计和实施企业的信息系统的任务，而信息系统的基础就是该企业所建立起来的计算机系统。为此系统分析员必须对现代计算机系统的概况、系统配置和系统性能有较为清楚的了解。这就是设置本章的目的。

11.1 计算机体系结构

11.1.1 计算机指令系统的发展

无论是使用何种编程语言和编程工具编写的软件，都要通过相应的编译程序将其转换成机器语言，即以二进制形式表示的指令，才能在计算机硬件上运行。一台机器上所能执行的指令的集合称之为指令系统。

1. 复杂指令系统计算机(CISC)

计算机的性能与它所设置的指令系统有很大的关系，而指令系统的设置又与机器的硬件结构密切相关。通常性能较好的计算机都设置有功能齐全、通用性强、指令丰富的指令系统，但这需要复杂的硬件结构来支持。

在 20 世纪 50 年代和 20 世纪 60 年代早期，由于计算机采用分立元件(电子管或晶体管)，其体积庞大，价格昂贵，因此，大多数计算机的硬件结构比较简单，所支持的指令系统一般只有定点加减、逻辑运算、数据传送和转移等十几至几十条最基本的指令，而且寻址方式简单。到 20 世纪 60 年代中、后期，随着集成电路的出现，计算机的价格不断下降，硬件功能不断增强，指令系统也越来越丰富，除了具有以上最基本的指令以外，还设置了乘除法运算指令、浮点运算指令、十进制运算指令以及字符串处理指令等，指令数多达一、二百条，寻址方式也趋于多样化。

随着集成电路的发展和计算机应用领域的不断扩大，计算机的软件价格相应不断提高，为了继承已有的软件，减少软件的开发费用，人们迫切希望各机器上的软件能够兼容，以便在旧机器上编制的各种软件也能在新的、性能更好的机器上正确运行，因此，在 20 世纪 60 年代出现了系列(Series)计算机。

所谓系列计算机是指基本指令系统相同、基本体系结构相同的一系列计算机，如 IBM 370 系列, VAX-11 系列, IBM PC(XT/AT/286... Pentium)微机系列等。一个系列往往有多种型号，各型号的基本结构相同，但由于推出的时间不同，所采用的器件也不同，因此在结构和性能上有所差异。通常是新推出的机种在性能和价格方面要比早推出的机种优越。系列机能解决软件兼容问题的必要条件是该系列的各机种有共同的指令集，而且新推出的机种的指令系统一定包含旧机种的所有指令，因此在旧机种上运行的各种软件可以不加任何修改地在新机种上运行。

计算机发展至今，其硬件结构随着超大规模集成电路(VLSI)技术的飞速发展而越来越复杂化，所支持的指令系统也趋于多用途、强功能化。指令系统的改进是围绕着缩小与高级语言的语义差异以及有利于操作系统的优化而进行的。例如高级语言中的实数计算是通过浮点运算实现的，因此对于科学计算的计算机来讲，如能设置浮点运算指令，则能显著提高运算速度；另外在高级语言程序中经常用到 IF 语句, DO 语句等，为此设置功能较强的条件转移指令是有好处的；为了便于程序嵌套，设置了调用指令(Call)和返回指令(Return)等。上述这些措施都是为了便于高级语言程序编译以及提高机器运行速度而采取的，这对简化汇编语言程序设计也是很有利的。为了便于操作系统的实现和优化，还设置有控制系统状态的特权指令、管理多道程序和多处理机系统的专用指令等。然而，指令结构太复杂也会带来一些不利的因素，如设计周期长，正确性难以保证且不易维护等；此外，实验证明，在如此庞大的指令系统中，只有诸如算术、逻辑运算，数据传送，转移和子程序调用等几十条最基本的指令才是经常使用的，而需要大量硬件支持的大多数较复杂的指令却利用率很低，造成硬件资源的极大浪费。为了解决这个问题，在 70 年代末人们提出了便于 VLSI 实现的精简指令系统计算机，简称 RISC。

2. 精简指令系统计算机(RISC)

1975 年 IBM 公司开始研究指令系统的合理性问题, IBM 的 John Cocke 提出精简指令系统的想法。后来美国加州伯克莱大学的 RISC 工和 RISC II 机、斯坦福大学的 MIPS 机的研究成功，为精简指令系统计算机(简称 RISC)的诞生与发展起了很大作用。

对 CISC 机进行测试表明，机器执行程序时各种指令的出现频率相差悬殊，最常使用的是一些比较简单的指令，在程序中仅占指令系统中指令总数的 20%，但出现的频率却占 80%。而且，在微程序控制计算机的指令系统中，占指令总数 20%的最复杂的指令，却占用了控制存储器容量的 80%。上述测试数据为 RISC 的进一步发展提供了充分的依据。

90 年代初 MIPS 公司, Sun 微系统公司 IBM 公司, HP 公司, DEC 公司相继推出新一代 RISC 微处理器。

大部分 RISC 机具有以下特点：

- 指令系统设计时选择一些使用频率较高的简单指令，和选择一些很有用但不复杂的指令。
- 指令长度固定，指令格式种类少，寻址方式种类少。
- 只有取数/存数(Load/Store)指令访问存储器，其余指令的操作都在寄存器之间进行。

• 在 20 世纪 80 年代中、后期，大部分指令在一个机器周期内完成，采用流水线技术。到 20 世纪 80 年代末 90 年代初，第三代 RISC 机采用超级标量及超级流水线技术，增加了指令执行的并行度，使得一条

指令的平均执行时间小于一个机器周期。

- CPU 中通用寄存器数量相当多,可以减少访存(存/取数据)次数。
- 以硬布线控制逻辑为主,不用或少用微码控制。
- 采用优化的编译程序,力求有效地支持高级语言程序。

同 CISC 机比较 RISC 机有以下优点:

• 可以充分利用 VLSI 芯片面积。用微程序控制的 CISC 整数运算部件,其微码电路在芯片上所占的面积要占整个芯片面积的 50%以上,因而可将空出的面积供其他功能部件用,例如可增加大量的通用寄存器,或将其他逻辑部件集成到 CPU 芯片中。

• 可以提高计算机运算速度。指令数、寻址方式和指令格式的种类都较少,且指令的编码很有规律,使指令译码加快;在简化指令的情况下,硬布线连接比微程序控制的延迟小,可缩短 CPU 的周期,CPU 的通用寄存器多,减少了访存次数,加快了速度;大部分指令能在一个周期内完成,特别适合于流水线工作,假如有的指令执行周期数过多,使流水线不畅通,将使大部分部件处于空闲状态。

- 设计容易,可降低成本,并提高可靠性。
- 能有效支持高级语言程序。RISC 靠编译程序的优化来支持高级语言程序。

指令少,寻址方式少,反而使编辑程序容易选择更有效的指令和寻址方式;通用寄存器多,可尽量安排快速的寄存器操作,使编译程序的代码优化效率较高;在编译时尽量做好程序优化工作,从而减少程序执行时间。

11.1.2 提高计算机系统运算速度的方法

提高计算机运算速度的方法很多,可归纳为两个方面:改进器件工艺,提高集成度与时钟频率;改进体系结构。

对于单机系统(系统内含一个 CPU)可采用下述方法:

- 采用多个通用寄存器来暂存运算的中间结果,以减少访问存储器(存/取数据)次数。
- 采用流水线工作方式。
- 采用多体交错存储器和/或 cache,以协调 CPU 和存储器之间的速度匹配。
- CPU 和输入/输出设备并行工作,以减少 CPU 等待和空闲时间(见 11. 2. 6)。

对于多机系统(系统内含有多个处理器或 CPU) I 除了采用上述方法外,还可采用并行处理技术进一步提高系统的处理能力。

11.1.3 流水线技术

计算机进行数据处理是通过按一定次序执行由若干条指令组成的程序而实现的。由 CPU 来控制指令的执行及其执行次序,即由 CPU 发出取指令信号和完成该指令操作所需的控制信号。在早期的或结构最简单的计算机中,指令之间是不重叠执行的,即执行完一条指令后再取下一条指令并执行之。通常需要若干个时钟周期才能完成一条指令的操作,例如在执行加法指令时可能需要 4 个时钟周期分别完成取指令、取操作数、执行加法运算和写结果 4 个操作(不同指令所需的周期数可能不同)。分析程序中各条指令的执行过程可以发现,机器的各部分在某些周期内进行操作,而在某些周期内是空闲的。仍以加法指令为例,如果操作数存放在通用寄存器,且运算结果也送回(写)到通用寄存器,那么仅在取指令周期访问存储器,在其他周期存储器都是空闲的。如果 CPU 中的控制部件调度恰当,让各个部件都紧张工作,就可提高计算机的运行速度,于是便研究出流水线结构。

1.指令的重叠执行

采用流水线技术的计算机将若干条指令在时间上重叠起来进行,如图 11. 1 所示。

图中假设计算机顺序执行第 n 条指令、第 $n+1$ 条指令……。当第 n 条指令从主存储器取出后,交给 CPU 去处理,并立即到存储器中取第 $n+1$ 条指令。这样存储器始终处于忙碌状态。而且尽管执行一条指令仍需 4 个周期(4 级流水线),但当第 n 条指令完成后,每个周期都能得到一条指令的运算结果,机器的运算速度提高到 4 倍。然而上述理想情况常常被打破,假如加法运算(第 n 条指令)的一个操作数存放在存储器中,那么取 n 条指令的操作数将与取第 $n+1$ 条指令发生访存冲突,因而要将取第 $n+1$ 条指令的动作推迟一个周期进行,这种现象称之为流水线阻塞或产生了“气泡”。

2.流水线中的相关问题

流水线不能连续工作的原因,除了编译形成的程序不能发挥流水线的作用或存储器供应不上为连续流动所需的指令和数据以外,还因为出现了“相关”情况或遇到了程序转移指令。例如在图 11.1 的 4 级流

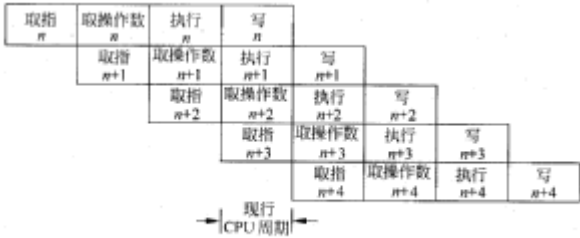


图 11.1 4 级流水线重叠工作情况

流水线中, 假如第 $n+1$ 条指令的操作数地址即为第 n 条指令的运算结果, 那么取操作数 $n+1$ 的动作需要等待 2 个时钟周期才能进行, 否则取得的数据是错误的, 这种情况称为数据相关, 该数据可以是存放在存储器中或通用寄存器中, 分别称为存储器数据相关或寄存器数据相关。此时流水线中指令流动情况将如图 11.2(a) 所示。为了改善流水线工作情况, 一般设置相关专用通路, 即当发生数据相关时, 第 $n+1$ 条指令的操作数直接从数据处理部件得到, 而不是存入后再读取, 这样指令能按图 11.2(b) 流动。由于数据不相关时, 仍需到存储器或寄存器中取数, 因此增加了控制的复杂性。另外由于计算机内有较多指令存在, 其繁简程度不一, 执行时间及流水线级数不同, 相关的情况各异, 有时避免不了产生不能连续工作的情况, 这种现象均称为流水线阻塞。

一般来说, 流水线级数越多, 情况越复杂, 而两级流水线则不存在数据相关现象。

3. 程序转移对流水线的影响

在大多数流水线机器中, 当遇到条件转移指令时, 确定转移与否的条件码往往由条件转移指令本身或由它前一条指令形成, 只有当它流出流水线时, 才能建立转移条件并决定下条指令地址。因此当条件转移指令进入流水线后直到确定下一指令地址之前, 流水线不能继续处理后面的指令而处于等待状态, 因而影响流水线效率。在某些计算机中采用了

“猜测法”技术, 机器先选定转移分支中的一个, 按它继续取指并处理, 例如条件码生成后, 说明猜测是正确的, 那么流水线可继续进行下去, 时间得到充分利用; 假如猜错了, 那么要返回分支点, 并要保证在分支点后已进行的工作不能破坏原有现场, 否则将产生错误。编译程序可根据硬件上采取的措施, 使猜测正确的概率尽量高些。有的机器还采用其他方法(见 11.1.4 节)。

在计算机运行时, 当 I/O 设备有中断请求或机器有故障时, 要求中止当前程序的执行而转入中断处理。在流水线机器中, 在流水线中存在几条指令, 因此就有一个如何“断流”的问题。当 I/O 系统提出中断时, 可以考虑把流水线中的指令全部完成, 而新指令则按中断程序要求取出; 但当出现诸如地址错、存储器错、运算错而中断时, 假如这些错误是由第 i 条指令发生的, 那么在其后的虽已进入流水线的第 $i+1$ 条指令、第 $i+2$ 条指令……, 也是不应该再执行的。流水线机器处理中断的方法有两种: 不精确断点法和精确断点法。有些机器为简化中断处理, 在不影响结果正确性的条件下, 采用了“不精确断点法”, 对那时还未进入流水线的后续指令不允许其再进入, 但已在流水线中的所有指令则仍执行完毕, 然后转入中断处理程序。由于集成电路的发展允许增加硬件的复杂性, 因此当前大部分流水线计算机采用“精确断点法”, 即不待已进入流水线的指令执行完毕, 尽早转入中断处理。

4. 其他

在图 11.1 中的现行 CPU 周期, 有 4 条指令的不同周期在同时工作。在设计时希望能做到取指、取操作数、执行和写 4 种操作所需的时间相等, 如果不等, 那么 CPU 周期应设计成与最长的操作时间相等的值。

每条指令所完成的操作都是有差异的, CISC 中有不少复杂指令要完成很多操作, 因此完成这条指令所需的时间也较长, 这对流水线工作是不利的。于是在某些机器中将一条复杂指令分解为若干条简单指令。RISC 的指令流水线较易实现。

以上讨论的是指令执行流水线, 经常采用的还有运算操作流水线, 例如执行浮点加法运算, 可以分成“对阶”、“尾数加”及“结果规格化”3 段, 每一段设置有专门的逻辑电路完成指定操作, 并将其输出保存在锁存器中, 作为下一段的输入, 如图 11.3 所示。当浮点加法对阶运算完成后, 将结果送入锁存器, 然后就可进行下一条浮点指令的阶码运算, 实现流水线操作。又如执行浮点乘法运算, 若与浮点加运算相似, 分成阶码运算、尾数乘和规格化 3 级流水线, 这是不合理的, 因为尾数乘所需的时间比阶码运算与规格化操作大得多, 同时尾数乘可以与阶码运算同时进行。为了提高运算速度, 尾数乘本身可用流水线方式组织起来。

由于流水线相邻两段在执行不同的指令(或操作), 因此无论是指令流水线或运算操作的流水线, 在相邻两段之间必须设置锁存器或寄存器, 以保证在一个周期内流水线的输入信号不变, 当流水线各段工作饱满时, 能发挥最大作用。在上例中, 假如浮点运算部件没有足够的数据来源, 那么流水线中的某些段、甚至全部将处于空闲状态, 这样就没有充分发挥流水线的作用。因此, 是否采用流水线组织, 在计算机的哪一部分采用流水线组织要根据实际情况确定。

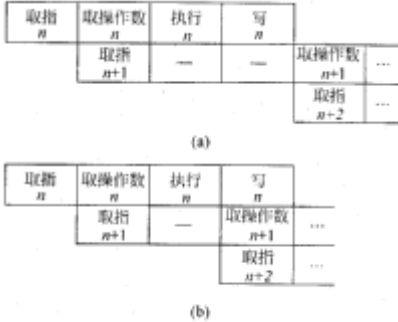


图 11.2 数据相关时的流水线

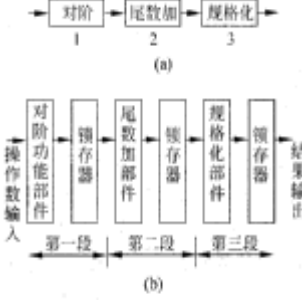


图 11.3 运算操作流水线

11. 1. 4 指令预取和无序执行

由于存储器接收到地址后取出指令(或数据)需要的时间与 CPU 操作相比要长得多,这对流水线的安排是很不利的,因此可以考虑提前从存储器取出指令,暂存在 CPU 的硬件(指令预取部件)中,由于程序的指令一般是顺序存放在存储器(转移例外)中的,指令预取是有用的。指令取出后可以预先进行分析,如果该指令执行时所需的操作数在存储器中,则亦可以提前取出放在 CPU 的数据寄存器,这样当指令进入流水线后的取指和取数操作都可在 CPU 内部完成,提高了速度。

指令预分析的另一好处可以适当调整一些指令的执行顺序,以利于程序的优化执行,其前提是不影响运算结果的正确性,称之为无序执行或动态执行。

这样一来在 CPU 中存在多条指令,有的还处于流水线的不同周期中,增加了 CPU 的控制复杂程度。

在对指令进行预分析时,如遇到无条件转移指令,则可以按转移后的地址取后续指令;遇到条件转移指令时,有时的机器采用“猜测法”,有的机器甚至同时选取几个转移分支的指令,按分支排成几个队列,当转移条件确定后选择其中一个队列继续执行程序,并舍弃其余队列。

11. 1.5 存储系统的发展

在现代计算机中,存储器处于全机中心地位,其原因是:

- 当前计算机正在执行的程序和数据(除了暂存于 CPU 寄存器以外的所有原始数据、中间结果和最后结果)均存放在存储器中。CPU 直接从存储器取指令或存取数据。

- 计算机系统中输入输出设备数量增多,数据传送速度加快,因此采用了直接存储器存取(DMA)技术和输入输出通道技术,在存储器与输入输出系统之间直接传送数据。

- 共享存储器的多处理机的出现,实现了利用存储器存放共享数据,并实现处理机之间的通信,更加强了存储器作为全机中心的作用。

上面提到的存储器通常又称为主存储器(简称主存)或内存储器(简称内存)。它是计算机系统中必不可少的部件。

现在大部分计算机中还设置有辅助存储器(简称辅存)或外存储器(简称外存),通常用来存放主存的副本和当前不运行的程序和数据。在程序执行过程中,取每条指令所需的数据及取下一条指令的操作都不能直接访问辅助存储器。

由于中央处理器是高速器件,不少指令的执行速度受限于主存储器的速度。所以,计算机解题能力的提高、应用范围的日益广泛和系统软件的日益丰富,无一不与主存储器的技术发展密切相关。

11.1. 5.1 多层次存储系统

衡量存储器有 3 个指标:容量、速度和价格/位。一般来讲,速度高的存储器,每位价格也高,因此容量不能太大。早期计算机主存容量很小(如几千字节),程序与数据从辅存调入主存是由程序员自己安排的,程序员必须花费很大精力和时间把大程序预先分成块,确定好这些程序块在辅存中的位置和装入主存的地址,而且还要预先安排好程序运行的各块如何和何时调入调出。现代计算机主存储器容量已达到几百兆字节,但是程序对存储容量的要求也提高了,因此仍存在存储空间的分配问题。

操作系统的形成和发展使得程序员尽可能摆脱主、辅存之间的地址定位,同时形成了支持这些功能的“辅助硬件”,通过软件、硬件结合,把主存和辅存统一成了一个整体,其速度接近于主存的速度,其容量则接近于辅存的容量,而每位平均价格也接近于廉价的慢速的辅存平均价格。这种系统不断发展和完善,就逐步形成了现在广泛使用的虚拟存储系统。在系统中,应用程序员可用机器指令地址码对整个程序统一编址,如同程序员具有对应这个地址码宽度的全部虚存空间一样。该空间可以比主存实际空间大得多,以致可以存得下整个程序。这种指令地址码称为虚地址(虚存地址、虚拟地址)或逻辑地址,其对应的存储容量称为虚存容量或虚存空间;而把实际主存的地址称为物理地址、实(存)地址,其对应的存储容量称为主存容量、实存容量或实(主)存空间。

当用虚地址访问主存时,机器自动地把它经辅助软件、硬件变换成主存实地址。查看这个地址所对应的单元内容是否已经装入主存,如果在主存就进行访问,如果不在主存内就经辅助软件、硬件把它所在的那块程序和数据由辅存调入主存,而后进行访问。这些操作都不必程序员来安排,也就是说,对应用程序员是透明的。

主-辅存层次解决了存储器的大容量要求和低成本之间的矛盾。

在速度方面,计算机的主存和 CPU 一直保持了大约一个数量级的差距。显然这个差距限制了 CPU 速度潜力的发挥。为了弥合这个差距,仅采用一种工艺的单一存储器是行不通的,必须进一步从计算机系统结构和组织上去研究。设置高速缓冲存储器(cache)是解决存取速度的重要方法。在 CPU 和主存中间设置高速缓冲存储器,构成高速缓存(cache)-主存层次,要求 cache 在速度上能跟得上 CPU 的要求。Cache-主存

间的地址映像和调度吸取了比它较早出现的主-辅层次的技术,不同的是因其速度要求高,不是由软、硬件结合而完全由硬件来实现。

从 CPU 的角度看, cache-主存层次的速度接近于 cache, 容量与每位价格则接近于主存。因此, 解决了速度与成本之间的矛盾。

以上叙述了主存-辅存和 cache-主存这两种存储层次。现代大多数计算机同时采用这两种存储层次, 构成 cache-主存-辅存 3 级存储层次, 如图 11.4 所示。其中 cache 容量最小, 辅存容量最大, 各层次中存放的内容都可在下一层次中找到。



图 11.4 3 层次存储系统

11.1.5.2 cache 存储器

1. cache 工作原理

对大量的典型程序的运行情况的分析结果表明, 在一个较短的时间间隔内, 地址往往集中在存储器逻辑地址空间的很小范围内。程序地址的分布本来就是连续的, 再加上循环程序段和子程序段要重复执行多次, 因此, 对程序地址的访问就自然地具有相对集中的倾向。数据分布的这种集中倾向不如指令明显, 但对数组的存储和访问以及工作单元的选择都可以使存储器地址相对集中。这种对局部范围的存储器地址频繁访问, 而对此范围以外的地址访问甚少的现象就称为程序访问的局部性。

根据局部性原理, 可以在主存和 CPU 之间设置一个高速的容量相对较小的存储器, 如果当前正在执行的程序和数据存放在这个存储器中, 那么程序运行时, 不必从主存储器取指令和取数据, 访问这个高速存储器即可, 所以提高了程序运行速度, 这个存储器称作高速缓冲存储器 cache。

cache 存储器介于 CPU 和主存之间, 它的工作速度数倍于主存, 全部功能由硬件实现, 并且对程序员是透明的。

cache 由存储体和标记两部分组成。

“存储体”内存放的是从存储器中取来的指令和数据, “标记”指示出这些指令和数据在主存中的位置(与地址有关)及其有效性。当 CPU 向存储器发出读指令/数据请求时, 查看标记以证实该指令/数据是否在 cache 中, 若是, 直接从 cache 中读出即可, 这种情况称为“命中”。否则需要从主存中读出, 并将读出的内容同时送 CPU 和 cache(并作好标记), 这种情况称之为“不命中”, 当下次再访问该存储单元时就可直接从 cache 中读出了。

在 CPU 和 cache 之间, 通常一次传送一个字块, 块长一般取一个主存周期所能调出的信息的长度。

cache 的容量和块的大小是影响 cache 效率的重要因素。通常用“命中率”来测量 cache 的效率。命中率指 CPU 所要访问的信息在 cache 中的比率, 而将所要访问的信息不在 cache 中的比率称为失效率。一般来说, cache 的存储容量比主存的容量小得多, 但不能太小, 太小会使命中率太低; 也没有必要过大, 过大不仅会增加成本, 而且当容量超过一定值后, 命中率随容量的增加将不会有明显的增长。但随着芯片价格的下降, cache 的容量不断增大, 已由几千字节发展到几兆字节。

cache 的命中率一般可达到 90%-95%以上。

在主存读出新的字块调入 cache 存储器时, 如果遇到 cache 存储器中相应的位置已被其他字块占有, 那么就必须要去掉一个旧的字块, 让位于一个新的字块。这种替换应该遵循一定的规则, 最好能使被替换的字块是下一段时间内估计最少使用的。这些规则称为替换策略或替换算法, 由替换部件加以实现。

cache 存储器中保存的字块是主存中相应字块的一个副本。如果程序执行过程中要对该字块的某个单元进行写操作, 就会遇到如何保持 cache 与主存的一致性问题。通常有两种写入方式: 一种方式是暂时只向 cache 存储器写入, 并用标志加以说明, 直到经过修改的字块被从 cache 中替换出来时才一次写入主存, 称之为“写回”方式; 第二种方式是每次写入 cache 存储器时也同时写入主存, 使 cache 和主存保持一致, 称之为“直写”方式。另有一种写操作方法是, 当被修改的单元根本就不在 cache 存储器时, 写操作直接对主存进行, 而不写入 cache 存储器。

2. 指令 cache 和数据 cache

计算机开始实现 cache 时, 是将指令和数据存放在同一 cache 中的, 后来随着计算机技术的发展和处理速度的加快, 存取数据的操作经常会与取指令操作发生冲突(如在指令流水线), 从而延迟了指令的读取, 发展的趋势是将指令 cache 和数据 cache 分开而成为两个相互独立的 cache。

3. 多层次 cache

当芯片的集成度提高后, 可以将更多的电路集成在一个微处理器芯片中, 于是近年来新设计的高速微处理器芯片都将 cache 集成在片内, 片内 cache 的读取速度要比片外 cache 快得多。

片内 cache 的容量受芯片集成度的限制, 其命中率比大容量 cache 低, 于是推出了二级 cache 方案, 其中第一级 cache(L1)设置在处理器芯片内部, 第二级 cache(L2)设置在片外, 也有的微处理器将其移至

片内，片外 cache 的容量可从几百千字节到几兆字节。有的系统还推出了三级 cache 方案。

11.1.5.3 虚拟存储器

主存—辅存层次的基本信息传送单位可采用几种不同方案：段、页或段页。

段是利用程序的模块化性质，按照程序的逻辑结构划分成的多个相对独立部分，例如过程、子程序、数据表、阵列等。段作为独立的逻辑单位可以被其他程序段调用，这样就形成段间连接，产生规模较大的程序。

把主存按段分配的存储管理方式称为段式管理。段式管理系统的优点是段的分界与程序的自然分界相对应；段的逻辑独立性使它易于编译、管理、修改和保护，也便于多道程序共享；某些类型的段(堆栈、队列)具有动态可变长度，允许自由调度以便有效利用主存空间。但是，正因为段的长度各不相同，段的起点和终点不定，给主存空间分配带来麻烦，而且容易在段间留下许多空余的零碎存储空间，不好利用且造成浪费。

页式管理系统的基本信息传送单位是定长的页，主存的物理空间也被划分为等长的固定区域，称为页面。页面的起点和终点地址是固定的，给造页表带来了方便。新页调入主存也很容易掌握，只要有空白页面就可容纳。惟一可能造成浪费的是程序最后一页的零头不能利用的页内空间，它比段式管理系统的段外空间浪费要小得多。页式管理系统的缺点正好和段式管理系统相反，由于页不是逻辑上独立的实体，所以处理、保护和共享都不及段式来得方便。

段式和页式存储管理各有其缺点，可以采用分段和分页结合的段页式存储管理系统。程序按模块分段，段内再分页，进入主存仍以页为基本信息传送单位。

下面介绍页式虚拟存储器

在页式虚拟存储系统中，把虚拟(辅存)空间分成页，主存空间也分成同样大小的页，称为实页或物理页，而把前者称为虚页或逻辑页。假设虚页号为 $0, 1, 2, \dots, m$ ，实页号为 $0, 1, \dots, 1$ ，显然有 $m > 1$ 。由于页的大小都取 2 的整数幂个字，所以，页的起点都落在低位字段为零的地址上。可把虚拟地址分为两个字段，高位字段为虚页号，低位字段为页内字地址。

虚拟地址到主存实地址的变换是由页表来实现的。在页表中，对应每一个虚存页号有一个表目，表目内容至少要包含该虚页所在的主存页面地址(页面号)，用它作为实(主)存地址的高字段，与虚拟地址的字地址字段相拼接，就产生完整的实主存地址，据此访问主存。页式管理的地址变换如图 11.5 所示。

通常，在页表的表项中还包括装入位(有效位)、修改位、替换控制位及其他保护位等组成的控制字段。如装入位为“1”，表示该虚页已从辅存调入主存；如装入位为“0”，则表示对应的虚页尚未调入主存，这时如访问该页就会产生页面失效中断，从而启动输入输出子系统，从磁盘等辅存中读出新的页到主存中来。修改位指出主存页面中的内容是否被修改过，替换时是否要写回辅存。替换控制位指出需替换的页等。

假设页表被保存或已调入主存储器中，那么，在访问存储器时，首先要查页表，即使页面命中，也得先访问一次主存去查页表，再访问主存才能取得数据，这就相当于主存速度降低了一倍。如果页面失效，要进行页面替换，页面修改，访问主存次数就更多了。因此，把最近访问过的页表部分存放在快速存储器中组成快表(见图 11.6)，采用按内容查找的相联存储器(在本节后面介绍)组成快表以实现并行查找。该快表通常称之为转换检测缓冲器 TLB(translation lookaside buffer)，一般在 16 行~64 行之间，是慢表(主存中的页表)的小小的副本。查表时，由虚页号同时去查快表和慢表，当在快表中有此虚页号时，就能很快得到对应的实页号，送入实主存地址寄存器，并使慢表的查找作废，从而就能做到虽采用虚拟存储器，但访主存速度几乎没有下降。如果在快表中找不到时，就要花费一个访主存时间查慢表，从中查到实页号，送入实主存地址寄存器，并将此虚页号和对应的实页号送入快表。如快表已写满，则替换快表中某一行内容。根据程序访问的局部性原理，在快表中查找到的几率是很大的。

11.1.5.4 访问存储器(取指或存取数据)的工作过程

1. 虚地址转换成实地址

对虚拟存储器来说，程序员按虚存储空间编制程序，在直接寻址方式下由机器指令的地址码给出地址。

这个地址码就是虚地址，可由虚页号及页内地址组成，如下所示：

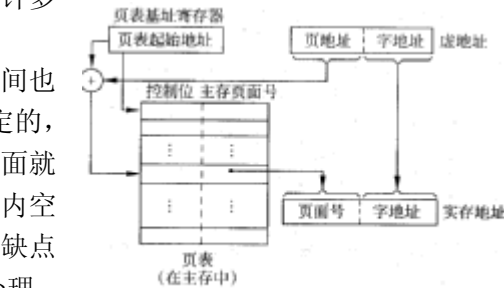


图 11.5 页式虚拟存储器结构

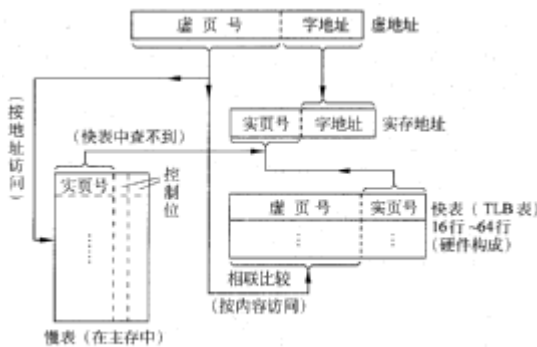
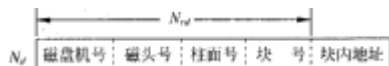


图 11.6 经快表和慢表实现的虚实地址变换

这个虚地址实际上不是辅存的实地址，而是辅存的逻辑地址。以磁盘为例，按字编址的实地址 N_d 如下：



因此，在虚拟存储器中还应虚拟地址到辅存实地址的变换。辅存一般按信息块编址，而不是按字编址，若使一个块的大小等于一个虚页面的大小，就只需把虚页号变换到 N_{vd} 即可完成虚地址到辅存实地址的变换。为此，可采用页表的方式。把由 N_v 变换成 N_{vd} 的表称为外页表，而把 N_v 变换到主存页号的表称为内页表。

虚拟存储器的工作全过程如图 11.7 所示。

在虚拟存储器每次访问主存时，都需要将多用户虚地址变换成主存实地址①②，因此，需要有虚页号变换成主存实页号的内部地址变换，可由查内页表来实现。当对应的有效位为+1时，就按主存实地址 n_p 访主存③；如果对应虚页的装入位为“0”，表示该页不在主存中，就产生页面失效中断④，由中断处理程序到辅存中调页。先通过外部地址变换⑤，例如查外页表，将多用户虚地址变换成辅存中的实地址 N_{vd} ；到辅存中去选页⑥，将该页内容经过 I/O 处理机或通道送入主存⑦。此时还需要确定调入页应该进入主存中哪一个页面位置，这就需要查实存页表⑨。当主存（允许装入的位置）未装满时，只需找到空页面⑩；而当主存已装满时，就需要通过替换算法寻找替换页⑪⑫。把确定了的实页号送入通道⑬。在进行页面替换时，如果被替换的页调入主存后一直未经修改，则不需送回辅存，如果已修改，则需先将它送回辅存原来的位置⑭，而后再把调入页装入主存⑦，是否修改过是可以由主存页表指明的。如果所需的页未装入辅存，还需要进入中断，进行出错处理或其他处理⑧。

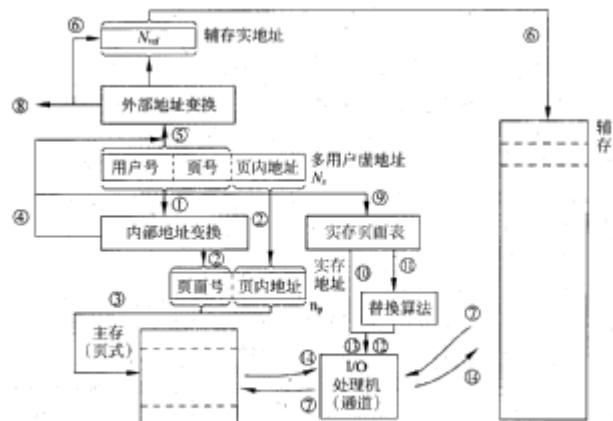


图 11.7 多用户虚拟存储器工作过程

2. 根据实地址访问主存

在有 cache 的计算机中，如果指令或数据已在 cache 中，则直接从 cache 中取出，否则要从主存储器取出。

某些计算机采用虚拟地址 cache，直接用虚地址访问，此时 cache 的标记要作相应的改变，假如命中，取指/取数的速度可以很快；假如不命中，那么虚地址转换成实地址的一系列工作仍需进行。

11.1.5.5 主存储器

半导体存储器按存储元件能否长期保存信息而分为静态存储器 (SRAM) 和动态存储器 (DRAM)。前者只要不断电，信息是不会丢失的；后者利用 MOS 电容存储电荷来保存信息，使用时，需不断给电容充电才能保持信息。SRAM 的集成度低，功耗较大；DRAM 的集成度高、功耗小，但存取速度比 SRAM 低。计算机的主存储器一般都用 DRAM，cache 的存储体用 SRAM。

1. DRAM 的研究与发展

近年来，开展了基于 DRAM 的研究和发展工作，并应用于计算机中。

- **增强型 DRAM (EDRAM)**。EDRAM 改进了 CMOS 制造工艺，使晶体管开关加速，其结果使 EDRAM 的存取时间和周期比普通 DRAM 减少一半，而且在 EDRAM 中还集成了小容量 SRAM cache，其内容与存储器的结构有关。

- **cache DRAM (CDRAM)**。其原理与 EDRAM 相似，主要差别是它的 SRAM cache 的容量比 EDRAM 的大，其工作方式与前面介绍的 cache 相同。在存储器直接连接处理器的系统中，cache DRAM 可用作第二级 cache 与主存储器（第一级 cache 在处理器芯片中）。

- **EDO DRAM**。即扩充数据输出 (extended data out, EDO) DRAM，它在完成当前存储器周期前即可开始下一存储器周期操作，因此能提高数据带宽或传输率。—

- **同步 DRAM (SDRAM)**。SDRAM 的读/写周期比 EDO DRAM 短，现已取代 EDO DRAM。典型的 DRAM 是异步工作的，处理器送地址和控制信号到存储器后等待存储器读出信号。SDRAM 采用成组传送方式（即一次顺序传送一组数据），除了传送第一个数据需要存储器地址建立时间和内部行线充电时间以外，在以后顺序读出数据时均可省去上述时间，它的读写周期为 10ns–15ns，与处理器之间的数据传送是同步的。

- **Rambus DRAM (RDRAM)**。RDRAM 着重解决存储器频带宽度问题，它采用垂直封装，所有引出线都从一边引出，使得存储器的装配特别紧凑，它与 CPU 之间传送数据是通过专用的 RDRAM 总线进行的，该芯片采取异步成组数据传输协议，在开始传送时需要较大存取时间，以后可达到很高的传输率，这是因为精

确地规定了总线的阻抗、时钟和信号。但目前生产成本高。

• **DDR SDRAM**。一般称为 DDR，它可以利用时钟的上升沿与下降沿传送数据，因此比 SDRAM 的频宽高一倍，目前使用呈上升趋势。

2. 交错存储器

计算机中大容量的主存实际上都是由多个存储体组成的，每个都具有自己的读写线路、地址寄存器和数据寄存器，称为“存储模块”。这种多模块存储器可以实现重叠与交错存取。如果在 M 个模块上交错编址，则称为模 M 交错编址。

设存储器包括 M 个模块，每个模块的容量为 L，各存储模块进行交错编址，连续的地址分布在相邻的模块中。第 i 个模块 M_i，的地址编号应按下式给出：M_j + i，其中，j=0, 1, 2, ..., L-1 i=0, 1, 2, ..., M-1

表 11. 1 列出了模 4 交错各模块的编址序列。这种编址方式使用地址码的低位字段经过译码选择不同的存储模块，而高位字段指向相应的模块内部的存储字。这样，连续地址分布在相邻的不同模块内，而同一模块内的地址都是不连续的。在理想情况下，如果程序段和数据块都连续地在主存中存放和读取，那么，这种编址方式将大大地提高主存的有效访问速度。但当遇到程序转换或随机访问少量数据时，访问地址就不一定均匀分布在多个存储模块之间，这样就会产生存储冲突而降低了使用率，所以 M 个交错模块的使用率是变化的，大约在 \sqrt{M} 与 M 之间。例如在大型计算机中 M 取 16-32，则平均有效存取时间至少可以缩短到单存储体的 1/4 至 1/6。高档微机 M 值可取 2 或 4。

一般模块数 M 取 2 的 m 次幂，但有的机器采用质数个模块，如我国银河机的 M 为 31，其硬件实现比较复杂，要有专门逻辑电路，用来从主存的物理地址计算出存储体的模块号和块内地址，但这种办法可以减少存储器冲突，只有当连续访存的地址间隔是 M 或 M 的倍数时才会产生冲突。这种情况的出现机会是很少的。

11. 1. 5. 6 相联存储器

相联存储器不按地址访问存储器，而按所存数据字的全部内容或部分内容进行查找(或检索)。例如在虚拟存储器中，将虚地址的虚页号与相联存储器中所有行的虚页号进行比较，若有内容相等的行，则将其相应的实页号取出，这是按数据字的部分内容进行检索的例子。

相联存储器的基本组成如图 11. 8 所示，设存储器有 W 个字，字长 n 位。CR 为比较数寄存器，字长也为 n 位，存放要比较的数(或要检索的内容)。MR 为屏蔽寄存器，与 CR 配合使用，字长也为 n 位。当按比较数的部分内容进行检索时，相应地把 MR 中要比较的位设置成“1”，不要比较的位置设成“0”。图 11. 8 中表示需要按 2-6 位的内容进行比较，所以 MR 的第 2-6 位置“1”，其余各位均置“0”。置成“1”的字段称为关键字段。SRR 为查找结果寄存器，字长为 W 位，假如比较结果第 i 个字满足要求，则 SRR 中的第 i 位为“1”其余各位均为“0”，若同时有 m 个字满足要求，则相应地就有 m 位为“1”。有的相联存储器还设置有字选择寄存器 WSR，用来确定哪些字参与检索，若字选择寄存器某位为“1”，则表示其对应的存储字参与检索；若某位为“0”，则表示其对应的存储字不参与检索。

为了进行检索，还要求相联存储器能进行各种比较操作(相等、不等、小于、大于、求最大值和最小值等)。比较操作是并行进行的，即 CR 中的关键字段与存储器的所有 W 个字的相应字段同时进行比较。

一般用门电路与触发器来进行比较与保存信号，所用电路较多，因此尽管在 50 年代中期已提出相联存储器概念，后来也有一些基于相联存储器原理的相联处理机出现，但没有得到很快发展。

在相联处理机中，来自控制器的一条命令能对许多数据同时执行算术或逻辑运算，因此各个存储单元除了有存储信息的功能外，还应有处理信息的能力，也就是说每个存储单元必须有一个处理单元。

20 世纪 80 年代后，由于集成电路的迅速发展，才使得半导体相联存储器有条件作为商品上市。

相联存储器除了应用于虚拟存储器与 cache 中以外，还经常用于数据库与知识库中按关键字进行检索。从按地址访问的存储器中检索出某一单元，平均约进行 w/2 次操作(w 为存储单元数)，而在相联存储器中仅需要进行一次检索操作，因此大大提高了处理速度。近年来相联存储器用于一些新型的并行处理和人工智能系统结构中，例如在语音识别、图像处理、数据流计算机，Prolog 机中都有采用相联存储器的例子。

11.1.6 系统总线 and 外设接口

计算机系统大多采用模块结构，一个模块就是具有专门功能的插件板，或叫做部件、插件、插卡，例

表 11. 1 地址的模 4 交错编址

模块	地址编序列	对应二进制地址最低二位	
M ₀	0, 4, 8, 12, ..., 4j+0, ...	0	0
M ₁	1, 5, 9, 13, ..., 4j+1, ...	0	1
M ₂	2, 6, 10, 14, ..., 4j+2, ...	1	0
M ₃	3, 7, 11, 15, ..., 4j+3, ...	1	1

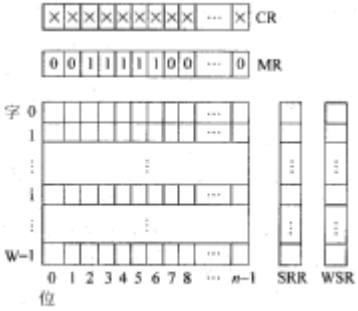


图 11. 8 相联存储器框图

如主板、存储器卡, I/O 接口板等。随着集成电路集成度的提高, 一块板上可安装多个模块。各模块之间传送信息的通路称为总线。为便于不同厂家生产的模块能灵活构成系统, 形成了总线标准。一般情况下有两类标准, 即正式公布的标准和实际存在的工业标准。

在标准中对插件引线的几何尺寸、引线数、各引线的定义、时序及电气参数等都作出明确规定, 这对子系统的设计和功能的扩充都带来了方便。

总线是从两个或两个以上源部件传送信息到一个或多个部件的一组传输线, 如一根传输线仅用于连接一个源部件(输出)和一个或多个目的部件(输入), 则不称为总线。

由于多个模块(或部件)连接到一条共用总线上, 必须对每个发送的信息规定其信息类型和接收信息的部件, 协调信息的传送; 必须经过选择判优, 避免多个部件同时发送信息的矛盾; 还需要防止信息的丢失。这就需要设置总线控制线路。总线控制线路包括总线判优或仲裁逻辑、驱动器和中断逻辑等。

常用的微机总线有 ISA 总线, EISA 总线 VME 总线和 PCI 总线等。

计算机的外部设备, 如磁盘驱动器、鼠标器、键盘和显示器等, 都是独立的设备, 这些设备与主机相连时, 必须按照规定的物理互连特性、电气特性等进行连接, 这些特性的技术规范称为外设接口。从物理结构来看, 例如硬盘驱动器, 通过电缆与适配器(磁盘控制器)相连, 适配器插在主机板的槽中, 这个适配器就是磁盘机的接口卡。它一方面通过槽背面的引线与 CPU 相连, 符合主机的系统总线规范; 另一方面与磁盘驱动器相连, 要符合外设接口规范。

常用的外设接口有 IDE 接口、SCSI 接口、USB 接口、RS232 接口和 PCMCIA 接口等。

现将 PCI 总线和 PCMCIA 接口简介如下。

外围部件互连 PCI(peripheral component interconnect)是由 Intel 公司推出的, 目前已得到广泛应用, 能实现即插即用(P&P)。

图 11. 9(a)所示为 PCI 在单处理器系统中的典型应用。“主存控制器/桥”模块加到 PCI 总线上, 其中桥的作用类似数据缓冲器, 因此 PCI 总线的速度可以不同于处理器。在多处理器系统中(图 11. 9(b))可以有 1 个或多个桥连接到系统总线上, 而系统总线仅连接处理器/cache, 主存控制器和桥。PCI 总线还可通过扩展总线桥, 连接低速外部设备。

PCMCIA(personal computer memory card international association)接口是广泛应用于笔记本电脑中的一种标准接口, 是一个小型的用于扩展功能的插槽, 通常用来插上存储器(flash memory)卡或 FAX/Modem 卡。一般在机箱的旁侧留下了 PCMCIA 插槽位置, 不用打开机箱就可接插 PCMCIA 卡。

大多数笔记本电脑可提供 I、II 和 III 型插槽, 插槽的区别仅在它们的厚度、长和宽是相同的, 均为 86.6mm×54mm, 卡的引出端均为 68 针, I 型卡插槽厚度为 3.3mm, II 型卡插槽厚度为 5.0mm, III 型卡插槽厚度为 10.2mm。

在 PCMCIA 卡上允许制作任一种 PCI 设备, 具有即插即用功能。

11. 1. 7 超级标量处理机、超级流水线处理机和超长指令字处理机

长期以来, 计算机设计人员在提高单处理机并行操作方面做了大量工作, 20 世纪 70 年代的向量处理机、20 世纪 80 年代的 RISC 机反映了这方面的成就。但是还不能突破一个时钟周期完成一条指令的框框。而本节要介绍的超级标量计算机和超长指令字计算机在一个周期内可流出多条指令。超级流水线以增加流水线级数的方法来缩短机器周期, 图 11.10 为 4 种处理机的指令流水线, 其中(a)为 RISC 指令流水线, 其余 3 种流水线分别介绍如下(假设采用取指、取操作数、执行、写回 4 级流水线)。

1. 超级标量(Superscalar)处理机

在超级标量处理机中, 配置了多个功能部件和指令译码电路, 采取了多条流水线, 还有多个寄存器端口和总线, 因此可以同时执行多个操作, 以并行处理来提高机器速度。它可以同时从存储器中取出几条指令同时送入不同的功能部件。例

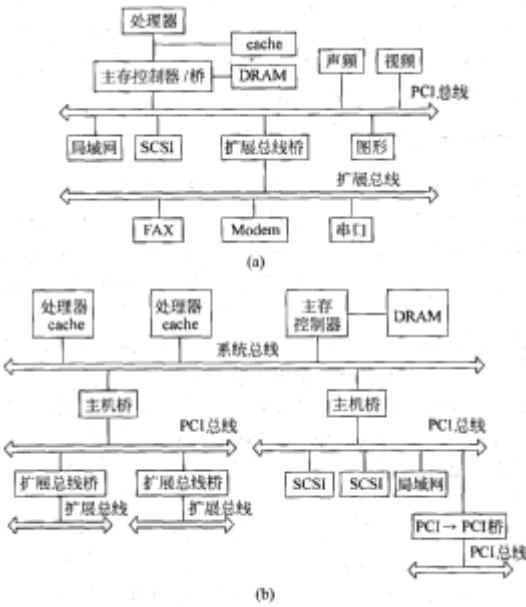
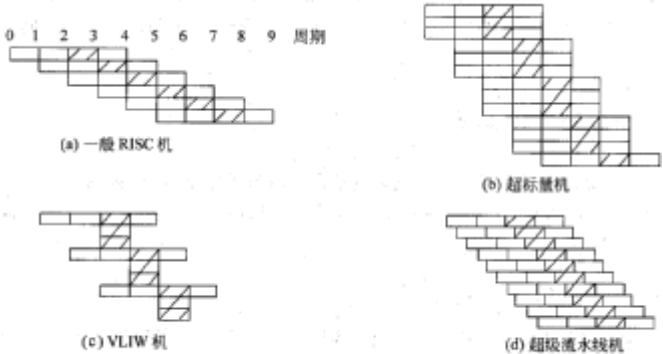


图 11.9 PCI 总线应用举例



注：图中阴影为“执行”段

图 11.10 4 种处理机的指令流水线

如 Intel 80960A 配置 3 条流水线, 分别执行整数运算、转移处理和访存操作, 能同时对 4 条指令进行译码, 但最多将 3 条能并行执行的指令分别送入 3 条流水线。超级标量机的硬件是不能重新安排指令的前后次序的, 但可以在编译程序时采取优化的办法对指令的执行次序进行精心安排, 把能并行执行的指令搭配起来。

1989 年, 在 Tandem 公司发表的 Cyclone 高可靠计算机系统中、开始采用超级标量技术。差不多同时, Intel 公司宣布了 i80960 处理器, IBM 公司推出了 POWER 86000 处理器。1992 年 SunTI (德州仪器公司) 推出 Super SPARC 芯片 (Viking), 以上这些芯片都是采用超级标量技术的。

2. 超级流水线(super pipeline)处理机

超级流水线处理机的周期比其他结构的处理机短, 在图 11. 10(d) 中, 周期缩短到 1/3, 执行一个操作需要 3 个周期。与超级标量计算机一样, 硬件不能调整指令的执行次序, 而由编译程序解决优先问题。

3. 超长指令字(VLIW)处理机

VLIW 是一种单指令流多操作码多数据的系统结构, 编译程序在编译时把多个能并行执行的操作组合在一起, 成为一条有多个操作段的超长指令, 由这条超长指令控制 VLIW 机中多个互相独立的功能部件, 每个操作段控制一个功能部件, 相当于同时执行多条指令。

11.2 并行处理计算机

在计算机系统中广泛采用并行处理技术来提高系统性能, 前面讲到的流水线技术以及在 CPU 中设置多个运算部件都属于并行处理范畴, 本节主要介绍向量计算机和多处理器/多处理机系统。

11.2.1 向量处理机

在科学研究和工程设计中的很多应用领域, 需要对巨大的数组进行高精度计算, 为此发展了向量处理机。向量数组是一个含有 N 个元素(数据)的有序数组, N 称为向量的长度, 向量中的每一个元素是标量, 它可以是浮点数、定点数、逻辑值或字符。因此向量处理机是一种具有向量数据表示、并设置有相应的指令和硬件、能对向量的各个元素进行并行处理的计算机, 当进行向量运算时, 它的性能要比大型机好得多。

向量处理机有巨型计算机和向量协处理机(或称为数组处理机)两种类型。巨型计算机能对大量的数据进行浮点运算, 同时它还是可以进行标量计算和一般数据处理的通用计算机。向量处理机一般采用流水线工作, 当它处理一条数组指令时, 对数组中的每个元素执行相同的操作, 而且各元素间是互相无关的, 因此流水线不会阻塞, 能以每个时钟周期送出一个结果的速度运行。

为了存储系统能及时提供数据, 向量处理器配有一个大容量的、分成多个模块交错工作的主存储器。

为了提高运算速度, 在向量处理机的运算部件中可采用多个功能部件, 例如向量部件、浮点部件、整数运算部件和计算地址用的地址部件。

向量协处理机是专门处理浮点和向量运算的数组处理机, 它连接到主机总线上。

11.2.2 多处理机系统

多处理机属于多指令流多数据流(MIMD)计算机。多处理机系统中的各个处理机执行不同的指令, 因此处理机之间不可能同步, 如果各处理机执行的程序段相互之间要传送数据或控制信息, 就要采用专门的同步措施, 以保证程序的正确执行。

1. 多处理机系统结构

多处理机系统从结构上可分为两类: 一类是具有共享存储器的系统, 另一类是具有分布存储器的系统。

图 11. 11 是具有共享存储器的多处理机框图。处理器($P_0 \sim P_N$)通过互连网络共享主存储器, 为了保证足够高的传输率, 主存由多个并行存储器($M_0 \sim M_M$)组成, 一般 $M > N$ 。I/O 设备与外存储器(例如磁盘存储器)通过互连网络和处理器 P 共享主存储器, 处理器之间也可以通过互连网络交换信息, 例如发送中断信号等。这种系统的优点是: 由于共享主存, 系统资源管理和使用比较方便, 缺点是受互连网络传输率的影响, 系统中处理器数目较少。

在分布存储器系统中, 每个处理器都有自己的局部存储器, 从而减少了相互访问存储器的频率, 有利于系统的扩充。缺点是编程困难, 因为要使每个处理器尽量访问局部存储器, 而且现有软件的可继承性差。

图 11. 12 是具有局部存储器和 cache 的多处理机系统。目前的处理器一般都带有 cache。但是有了 cache 以后, 又会产生系统中多个 cache 以及 cache 和存储器中的数据一致性问题。解决 cache 一致性问题

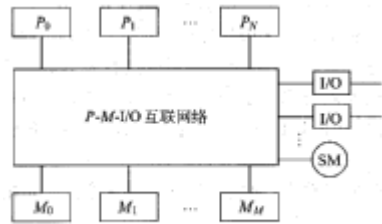


图 11.11 共享存储器的多处理机系统

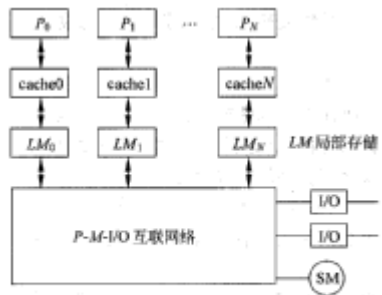


图 11.12 具有局部存储器和 cache 的多处理机系统图

处理机系统技术中的一个重要问题。

图 11.13 表示在有/无局部存储器条件下处理机数和系统性能的关系,在无局部存储器的系统中,当处理器数增加时,由于访存冲突的增加而影响了系统性能。

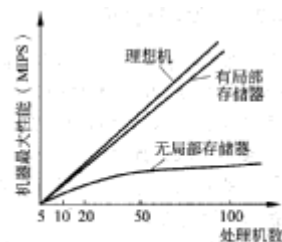


图 11.13 处理器数和机器速度的关系

2.大规模并行处理机 MPP 和对称处理机 SMP

MPP 是指由大量处理机组成的并行处理系统。处理机可采用当前市场上出售的微处理器,再加上 cache 和局部存储器,可获得很高的性能价格比。现今的 MPP 可理解为可扩展的并行机,一般分成两类:分布式共享存储结构 DSM(distributed shared memory)和群集系统(cluster clusted system)。

群集系统是将多台工作站(或微机)用互联网连接起来,充分利用各计算机资源,统一调度,协调处理,以实现高效并行计算的分布式计算机系统。它的互联网络通常采用局部网。

SMP 有一个统一的共享存储器。

3.互联网络 ICN(inter connection network)

ICN 用来连接一个并行计算机系统中各个处理单元(或处理机)、存储模块以及各种外部设备,在系统软件控制下,使各处理单元或各个功能部件相互通信的硬件网络结构。

并行计算机系统内部的互联网与连接多个计算机系统的局域网(或广域网)在地理分布、传输速度与控制方式等方面有差别,因此互联网这一术语一般只在讨论并行计算机内部通信时使用,但群集系统 cluster 例外。

常见的互联网结构如下:

- **总线结构。**并行计算机最简单的互连方式是把所有功能部件(处理器、存储器模块和外部外备)连接到总线上,总线由一组连线及控制部件(接口)组成,用于传输数据。总线基于分时方式工作,每一时刻只允许与它相连的一对设备(发送设备只能一个,接收设备可允许多个)进行通信,当有多个通信要求同时出现时,总线仲裁部件要按优先级进行仲裁。

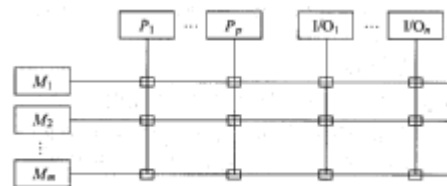


图 11.14 交叉开关

- **交叉开关(Crossbar switch)。**在处理器和 I/O 设备都与存储器进行通信的并行系统中,每个存储器模块都有一套总线同 P 个处理器和 n 个输入/输出通道相连,如图 11.14 所示。如果存储器模块数 $m > p+n$,则在同一时刻, p 个处理器和 n 个输入/输出通道都能分到一套总线,与 m 个存储器模块中的一个相连,因此能大大提高传输带宽和系统效率。

交叉开关结构中的每一个交叉接点都是一套开关,不仅要有传送数据和地址码的开关,还要有在多个处理机和 I/O 通道向同一个存储器模块发出访问请求时的排队能力,所以每一交叉接点上都有相当大的硬件设备量,当 $m=p+n$ 时设备量和 m^2 成正比。

- **多级互联网。**多级互联网由多级开关模块和级间连接组成。

4. cache 一致性

在采用层次结构的计算机中,保证 cache 和主存储器中的数据一致。

在单处理机系统中,cache 和主存数据的不一致主要由输入输出(I/O)操作引起的。通常, I/O 数据总是直接从主存进出而不经 cache;因此在使用“写回”型 cache 的系统中,当输出数据时, I/O 设备看到的只是主存中的数据,而在 cache 中的相应数据可能已修改过;应由硬件检查 cache 中的相应标志,发现与输出数据地址相同时,先将 cache 中的数据写入主存,再输出;当从 I/O 输入数据时,主存接收新数据,并将 cache 中有关地址的内容(如有的话)置以“无效”的标志。

多处理机系统的 cache 一致性问题更为突出,在这种系统中,每个处理器带有 cache;cache 中存放的可以是其本身的局部存储器中数据的副本,也可能是共享存储器或其他处理器局部存储器中的数据副本。多处理器对 cache 一致性的要求可表达为:任意一次取数操作得到的结果都必须是各处理器(或 I/O 设备)最近一次对该数据进行写入的值。解决此问题的方法是:保证当某一处理器更新 cache 中的共享数据时,主存和持有该数据副本的其他 cache 能及时知道。

5.非均匀存储存取 NUMA(non uniform memory access)

NUMA 并行模型和 DSM 并行结构所包含的并行机制基本一致,因为 DSM 的主存储器分布在不同结点上,一个处理机存取远程结点的数据,比存取同一结点的局部数据“路途”远一些,速度慢一些,所以称之为非均匀存储存取。由多个处理机共享同一存储数据而引起的存储一致性问题,在 NUMA 模型中一般采用 cache 一致性的办法来解决,这就是 CC-NUMA。

下面以 SGI 公司的 Origin 服务器为例来介绍多处理机系统的结构。

SGI Origin 采用 CC-NUMA 结构, 系统易扩展, 可从 1 个处理机扩展到 128 个处理机而维持系统的性能价格比不变。

(1) SGI Origin 的基本结构。

图 11.15 为 Origin 系统基本结构, 该系统基本上由结点, I/O 子系统、路由器 Hub 和互连网络构成。图中所示的结点实际上是一张卡, 卡上可安装 1 个或 2 个 MIPS R10000 微处理器芯片, 内含一级 cache (L1 cache)、二级 cache (L2 cache)、主存储器、目录存储器和 Hub 等。

Origin 存储系统采用分布式共享存储器 DSM 结构, 存储器被分配到各个结点, 但统一编址, 可被系统中所有处理器共享, 每一卡上主存容量为 4GB, 目录存储器是为解决 cache 一致性而设置的。

结点卡上的 Hub 是公司自行设计的专用集成电路 ASIC 芯片, 内含 4 个接口和交叉开关。

Origin 系统的 I/O 设备也采用分布式结构, 通过 ASIC 电路与各结点的 Hub 相连接, 系统内任一台 I/O 设备可被任一处理器访问。

Origin 的路由器和互连网络都是 ASIC 芯片, 通过芯片内部的交叉开关选择数据传送路径, 并提供无差错传送。

(2) Origin 的拓扑结构。

Origin 系统可由 1~128 个处理器构成, 图 11.16 中的 (a) 为 2 结点 4 个处理器, (b) 为 16 个处理器, (c) 为 32 个处理器, (d) 为 64 个处理器的互联结构。为了减少数据在路由器之间的传送延迟, 并充分利用路由器的端口, 将 (b) 和 (c) 中处于对角线位置的路由器进行连接。64 个处理器由两个立方体构成。

从上述的系统结构可以看到: 在结点卡内部实现的是 SMP 结构, 由于只有两个处理器, 所以不存在 SMP 的总线瓶颈问题。在结点卡之间实现的是 MPP 结构, 又通过 cache 一致性的实现解决了共享存储问题。

11.3 计算机系统的可靠性、可用性、可维护性技术和容错技术

计算机的可靠性、可用性和可维护性 (computer reliability, availability and serviceability, RAS) 技术和容错技术是研究、设计、生产、评价计算机系统的重要内容, 尤其是由于超大规模集成电路的发展和计算机应用的普及, 人们对 RAS 的要求越来越高, 军事、航天和金融等部门对计算机的 RAS 要求更高。

对于实用的计算机系统, 由于受系统规模、应用范围、环境条件和成本价格等因素的影响, 其 RAS 的投入强度和实现方法是不同的。

11.3.1 计算机系统的可靠性

11.3.1.1 计算机系统的可靠性指标

计算机系统的可靠性是指在规定的条件下和规定的时间内计算机系统能正确运行的概率。“规定的条件”包括环境、使用、维修等条件和操作技术。“规定的时间”通常用平均故障间隔时间 MTBF 来表示。

提高系统可靠性一般有两类技术方法, 即避错法和容错法。

硬件避错技术的作用是减少系统失效的可能性, 主要包括:

- 系统可靠性预计。计算机系统由各类元、部件组成, 系统可靠性预计即根据各类元、部件的可靠性以及各元、部件之间的连接关系构成的可靠性模型作系统可靠性的预先分析计算, 以预测系统是否可达到可靠性指标, 为改进可靠性设计提供依据。

- 可靠性分配。根据系统总的可靠性指标, 将系统分解, 并对各分系统直至器件、工艺提出相应的可靠性指标。

- 元件的优选及老化筛选。

- 使用可靠的连接组装工艺, 严格生产过程中的质量控制。

- 在设计时对元器件的额定参数留有足够余量 (例如电压、功耗等), 考虑元器件参数的离散性以及负载、温度变化而引起的参数变化 (例如延迟时间)。

- 降低系统内部的电磁干扰, 屏蔽外界电磁干扰。

- 合理布局热源, 制定冷却方案, 控制元器件工作环境的温度和湿度。

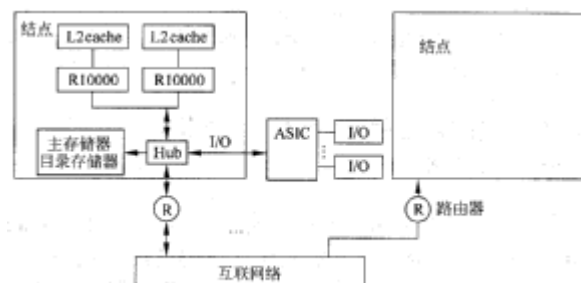


图 11.15 Origin 系统的基本结构

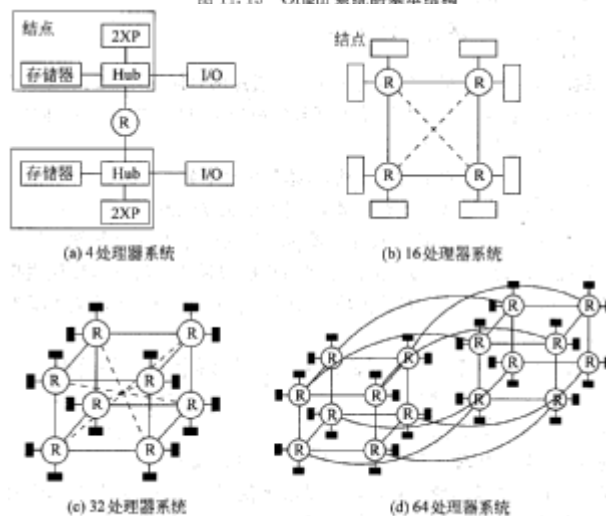


图 11.16 Origin 系统的拓扑结构

。采取防震、防冲击、防盐雾等机械结构措施。

容错技术主要采用硬件冗余、软件冗余、信息冗余和时间冗余等方法，将在 11.3.4 节介绍。

对系统可靠性的估算方法是先求出分系统的失效率，再求出系统的失效率，最后计算出 MTBF。

$$\text{分系统的失效率 } \lambda_i = m_1 \lambda_{i1} + m_2 \lambda_{i2} + \cdots + m_k \lambda_{ik}$$

式中 k 为组成第 i 个分系统的成分种类，如集成电路、元器件、接插件、印制板金属化孔和焊点等，

今统称为元件。 λ_{ik} 为第 i 分系统第 k 种成分单个元件的失效率。m 为各种成分的数量， $m_k \lambda_{ik}$ 为第 k 种成分的失效率。

$$\text{系统的失效率 } \lambda = n_1 \lambda_1 + n_2 \lambda_2 + \cdots + n_L \lambda_L$$

式中 L 为分系统(部件、设备)的种类, n 为分系统的数量 n 挤: 为第 L 种分系统的失效率，则系统的平

均故障间隔时间 $MTBF = \frac{1}{\lambda}$ ，其单位为 h(小时)。

单个元件的失效率是由厂家提供或测试得到的，这是一个统计数字，不可能很正确，其值等于单位时间内(小时)失效的元件数与参加运行的元件数的比值。

在实际使用时，MTBF 还受系统所处的环境、设计的优劣、系统的冗余程度和容错能力等因素的影响。

11.3. 1.2 采用附加的数据校验码来提高计算机系统的可靠性

数据在计算机系统内形成、存取和传送的过程中可能产生错误。为减少和避免这类错误，一方面是精心设计各种电路，提高计算机硬件本身的可靠性，另一方面是在数据编码上找出路，即采用带有某种特征能力的编码法，通过少量的附加电路，使之能发现某些错误，甚至能确定错误的性质和准确的出错位置，进而实现自动改错的能力。

数据校验码就是一种常用的带有发现某些错误或自动改错能力的编码方法。它的实现原理是，在合法的数据编码之间，加进一些不允许出现的(非法的)编码，使合法数据编码出现某些错误时，就成为非法编码。这样，就可以通过检测编码的合法性来达到发现错误的目的。合理地安排非法编码数量和编码规则，就可以提高发现错误的能力，甚至达到自动改正错误的目的。这里用到一个码距的概念。码距是指任意两个合法码之间至少有几个二进制位不相同，仅有一位不同，称其码距为 1，例如用 4 位二进制表示 16 种状态，则 16 种编码都用到了，此时码距为 1，就是说，任何一个状态的 4 位码中的一位或几位出错，就变成另一个合法码，此时无查错能力。若用 4 个二进制位表示 8 个状态，就可以只用其中的 8 种编码，而把另 8 种编码作为非法编码，此时可使合法码的码距为 2。一般来说，合理地增大合法码的码距，就能提高发现错误的能力，但表示一定数量的合法码所使用的二进制位数变多，增加了数据存储的容量或数据传送的数量。在设计数据校验码的时候，通常要考虑在不过多增加硬件开销的情况下，尽可能发现更多的错误，甚至能自动改正某些最常出现的错误。

假设电路有 p 个输入端和 q 个输出端，则可能出现的输入组合为 2^p 个，可能出现的输出组合为 2^q 个，如果在正常运行时 2^q 个输出组合均可能出现，则就无法通过观察和验证输出值来确定电路是否存在故障。如果在 2^q 个输出组合中仅有 $m < 2^q$ 个输出组合是有效的，那么在出现另一些属于 $2^q - m$ 中的组合时就可断定电路中必定存在故障。通常称属于 m 中的组合为有效的输出组合，称为合法码字。否则为错误的输出组合，称为非法码字。

能自动证实电路或系统中是否存在故障的检验技术与纠错编码技术密切相关。常用的发现错误或纠错的数据校验码有奇偶校验码、海·明校验码和循环冗余校验码(CRC)。

1. 奇偶校验码

奇偶校验码是一种开销最小、能发现数据代码中一位出错情况的编码，常用于存储器读写检查，或数据传送过程中的检查。它的实现原理是，使原来合法编码码距由 1 增加到 2。若合法编码中有一个二进制位的值出错了，由 1 变成 0，或由 0 变成 1，这个码都将成为非法编码。实现的具体方法通常是为一字节补充一个二进制位，称为校验位，用设置校验位的值为 0 或 1，使字节的 8 位和该校验位含有 1 值的个数为奇数或偶数。在使用奇数个 1 的方案进行校验时，称为奇校验，反之称为偶校验。**奇偶校验不能用于纠错。**

2. 纠错码 ECC

ECC 是具有对信息进行检测并纠正其错误的码。汉明码(Hamming Code)是能纠正可能出现的各种单个错误的纠错码。扩展汉明码能纠正一个错误并能检测两个错误。在计算机的存储器中，经常使用汉明校验

码。它的实现原理是，在数据中加入几个校验位，将数据代码的码距均匀地拉大，并把数据的每一个二进制位分配在几个奇偶校验组中。当某一位出错后，就会引起有关的几个校验的值发生变化，这不但可以发现出错，还能指出是哪一位出错。为进一步自动纠错提供了依据。

假设校验位的个数为 r ，则它能表示 2^r 个信息，用其中的一个信息指出“没有错误”，其余的 $2^r - 1$ 个信息指出错误发生在哪一位。然而错误也可能发生在校验位，因此只有 $k = 2^r - 1 - r$ 个信息能用于纠正被传送数据的位数，也就是说要满足关系： $2^r = k + r + 1$

如果能检测与自动校正一位错，并发现两位错，此时校验位的位数 r 和数据位的位数 k 应满足下述关系： $2^{r-1} \geq k + r$

按上述不等式，可计算出数据位 k 与校验位 r 的对应关系。

3. 循环冗余检验码 CRC

CRC 是循环码的子类。若码 C 中任一码字任意循环移位后仍为码 C 中的码字，即码 C 对循环移位具有封闭性，则称 C 为循环码。一个码字长为 n ，其中信息码长 k ；校验码长 r ，即 $n = k + r$ 。

在计算机的磁盘机和磁带机中常用 CRC 来纠错，其纠错能力很强，且是高效码，例如在磁带机中选用的 CRC-16 码型和软磁盘机中选用的 CRC-CCI TT 码型，其校验码位长 r 为 16，生成的码长 $n = 2^{15} - 1 = 32767$ ，编码效率 $R = (n - r) / n = 99.95\%$ ，是高效码。

上述两种码型可直接选用市售 CRC 芯片来实现，不必设计编译电路。

11.3.2 计算机系统的可用性

计算机系统可用性是指该系统在某一时刻提供有效使用的程度，以可用度 A 表示，可用度是在任意指定时刻系统能正确运行的概率。一般情况下，系统发生故障后是可以修复的，可用下述公式计算：

$$A = \frac{MTBF}{MTBF + MTTR}$$

式中 MTBF 为平均无故障时间 MTTR 为平均修复时间。平均修复时间是指多次故障中，从开始失效到系统修复所用的平均时间，可用下式计算：

系统修复时间 = 申请维修时间 + 等待时间 + 维修时间 + 恢复时间

其中，申请维修时间—开始失效到报告维修人员的时间；

等待时间—请求现场服务，等待维修人员到达和备品配件到达的时间；

维修时间—进行故障检测、诊断和修复所需的时间。

恢复时间—系统重新启动并开始运行所需的时间。

提高计算机可用性的途径是：提高计算机可靠性；提高计算机的可维护性，完善故障诊断与测试技术以及系统恢复和部件更换技术；提高维护人员的素质。

11.3.3 计算机系统的可维护性

计算机系统可维护性是指该系统失效后在规定时间内可修复到规定功能的能力。反映系统可维护性高低的参数是修复率和平均修复时间 MTTR。修复率表示在单位时间内完成修复的概率。

MTTR 可用实验方法求得：对 1 个系统注入不同的故障，进行修复并记下修复时间，最后可求出平均修复时间。为了使所求得的 MTTR 比较准确，应注入不同类型的故障，并让不同技术水平的维修人员进行维修。

在实际工作中，一般将维修分成 3 级：

- 第一级维修为现场维修，通常将故障定位到电路板，更换有故障的电路板，使系统恢复正常运行，现场维修通常要借助于系统提供的内部测试功能。
- 第二级维修为中间级，在现场不能实施的维修可到邻近现场的维修点进行，称为中间级维修。
- 第三级为工厂级维修，现场和维修点都不能解决的问题只能由生产厂家维修，这时失效的系统或部件送回工厂进行维修。

为了提高系统的可维护性，在设计时就要考虑使系统易于测试、诊断和修理，包括采用系统的自诊断技术、可测试性设计技术和系统结构的模块化设计等。

11.3.4 容错技术

容错是指计算机系统在运行过程中发生一定的硬件故障或软件错误时仍能保持正常工作而不影响正确结果的一种性能或措施。具有容错能力的计算机称为容错计算机。该系统在出现某些硬件故障或软件差错时能自行检测并采取适当的措施来保证完成预期的任务，整个过程不需要人工干预。

容错计算机的主要设计目标是为了提高计算机系统的可靠性和可用性。提高计算机可靠性的技术可以分为避错技术和容错技术。后者主要运用冗余技术来抵消由于故障而引起的影响。所谓冗余技术是指在系统正常运行所必须的软硬件基础上加上一定数量的信息、时间或后备软硬件的方法。冗余技术是计算机容错技术的基础。

冗余一般可分为下列几种类型。

- **硬件冗余：**以检测或屏蔽故障为目的而增加一定硬件设备的方法。
- **软件冗余：**为了检测或屏蔽软件中的差错而增加一些在正常运行时所不需要的软件的方法。
- **信息冗余：**在实现正常功能所需的信息外，再添加一些信息，以保证运行结果正确性的方法，纠错码就是信息冗余的例子。
- **时间冗余：**使用附加一定时间的方法来完成系统的功能。这些附加的时间主要用在故障检测、复执或故障屏蔽上。

故障可归结为永久性故障、间隙性故障和瞬时性故障 3 类。永久性故障表现出稳定性及持续性，例如元器件的损坏，电路的断线或短路，程序编写的错误等，它的特点是故障可以重复出现。间隙性故障表现出不稳定性和对系统状态的依赖性，此时可表现出机器时好时坏的现象，或在运行某些程序时机器能正常运行，而在其他情况下又出错误。上述两种故障不会自行消失，需要进行维修纠正。瞬时性故障是由偶然原因引起的短暂故障，例如电网电压的波动引起的故障，一般不用维修就能正常恢复，通常采用复执方法消除故障的影响，但若频繁出现，也要影响工作。因此无论何种故障，均需及时发现、及时纠正，避免故障造成的损失，故障可以是硬件造成的，也可能是软件造成的。

软件故障来源于程序错误，具有永久性，必须纠正，为了减少软件设计错误，必须对软件进行调试、测试以证明其正确性。

测试并确定计算机硬件有无故障的过程叫做故障检测，判定故障发生在某个子系统、功能模块直到元器件的过程称为故障定位，故障诊断包括故障检测和故障定位两部分。

最常用的硬件冗余是硬件的重复设置。硬件冗余一般可分为 3 种类型：静态冗余、动态冗余和混合冗余。

静态冗余是将已发生的故障屏蔽起来，使不影响运行结果，常用表决机制来屏蔽发生的故障，使用最广的是 3 模冗余，其基本概念是使用 3 套完全相同的硬件系统执行相同的任务，然后由一个多数表决器对 3 套系统的输出进行表决以确定系统的输出。多数表决器的原则是 3 中取 2。也就是说 3 模冗余系统可以最多允许有一个模块发生故障而不影响结果的输出。3 模冗余涉及多数表决器本身的可靠性问题。提高多数表决器可靠性的方法有多种，其中最常用的方法是多数表决器本身也使用 3 模冗余，即利用 3 个独立的多数表决器，每个多数表决器分别接受来自 3 个模块的输出作为它的输入，然后再分别输出。

动态冗余技术是通过故障检测、故障定位及故障排除和系统恢复等步骤达到容错的目的，最典型的方法是构造带有比较器的双工系统。在这种方法中，使用两套完全相同的硬件，且同时完成完全相同的任务，然后对它们的结果作比较，这种方法只能检测到有无故障而不能确定故障发生在哪一套硬件，必须增加一些措施才能做出故障定位。

混合冗余技术是将静态冗余技术和动态冗余技术结合起来，且取二者之长处，它先用静态冗余中的故障屏蔽技术，以消除可被屏蔽故障的影响，而对那些无法屏蔽的故障则采用动态冗余中的故障检测、故障定位、故障排除和系统恢复等技术，因此混合冗余的效果好，但附加的硬件开销大，成本高。

信息冗余是将冗余信息添加到数据中从而达到故障检测、故障屏蔽和容错的目的。信息冗余的最好例子是检错码和纠错码。这是将冗余信息加到一个数据字上使每一个数据字变成一个带有冗余信息的字。前面介绍的奇偶校验码、纠错码 ECC 和循环冗余校验码 CRC 都是信息冗余的例子。信息冗余也要付出一定的硬件代价。

时间冗余是以时间为代价，以减少硬件冗余和信息冗余的开销来达到提高可靠性的目的。在某些实际应用中，当不希望出现过高的硬件冗余和信息冗余的开销，而时间又不是最重要的指标时，可以使用时间冗余。其基本概念是重复执行多次相同的计算，称为重复执行或复执，然后将每次的计算结果存放起来再进行比较；若每次检测的结果都相同，说明没有故障，若不相同，则说明检测到了故障。显然这种方法只能检测瞬时性故障而不能检测永久性故障，系统有永久性故障时，每次计算结果都是相同的，但是错误

的。因此在采用时间冗余的系统中往往还附加少量的冗余硬件。

软件冗余是利用冗余的软件来检测硬件和软件。常用的有一致性检查、能力检测和多版本程序设计等。一致性检查是对某一运行结果先作一定的预测，然后将程序在运行中和运行后的结果与预测值作比较。若实际结果在期望值的范围内，则认为正常，否则认为有故障。能力检查是用检查程序(又称诊断程序)去检查系统中各个部件应有的能力，例如用程序来读写存储器的每个单元，以检查其读写和存储能力，又如用一组数据去检查运算逻辑部件的运算功能。多版本程序设计是对一项相同的任务用不同的方法进行程序设计，然后对不同版本的程序运行结果进行比较，不同版本的运行结果相同，则认为无故障，否则为有故障。多版本程序设计不仅能检查硬件故障，也能检查软件自身的故障。

上述的冗余技术，即硬件冗余、信息冗余、时间冗余和软件冗余，是使系统提高可靠性的基本措施和方法。厂在实际应用中，这4种冗余技术经常是结合起来使用的。

一个采用冗余技术的计算机系统在处理运行中产生故障时，通常采用以下 10 个步骤(或其中的一部分)。

(1) 故障限制：限制故障的影响范围，防止故障影响到系统的其他部件。

(2) 故障检测：一般分为联机检测和脱机检测。联机检测要求系统提供实时检测的能力，这种检测工作与系统的正常工作同时进行。脱机检测时，系统必须停止正常工作。

(3) 故障屏蔽：将出现的故障屏蔽起来，使系统的正常运行不受故障的影响。

(4) 复执：这是一种检测瞬时性故障的有效方法，提高系统抗瞬时性干扰的能力。

(5) 故障诊断：在故障检测基础上，对故障进行定位，这对以后的修复和重配置有用。

(6) 系统重配置：故障定位后，将发生故障的元件或部件替换下来；或者隔离故障元件、部件，系统仍能继续运行，只是系统运行速度下降，功能减弱，称为系统降级使用。双工系统和多机系统是典型的应用系统。

(7) 系统恢复：当检测出故障，必要时进行系统重配置后，即可消除故障引起的差错，这时系统应能返回到出现故障时的断点继续运行，称为系统恢复。

(8) 系统重新启动：如果故障造成几的错误影响了信息的恢复，就不能采用系统恢复的办法而必须重新启动运行。

(9) 修复：对已确定有故障的部件进行修复，修复分脱机修复和联机修复两种，如果有故障的部件取下后对系统影响不大或修复故障部件必然会停机，则采用脱机修复的办法。联机修复通常是指系统能启用备用部件代替故障部件，并继续运行，然后再修复切换下来的故障部件。

(10) 系统重组：故障部件修复后，重新投入系统运行，这时系统必须重新组合，以便完全恢复正常工作。

11.4 计算机性能评测

11.4.1 计算机性能评测概述

计算机性能评测是为了一定目的，按照一定步骤，选用一定的度量项目，通过建模、计算和实验，对计算机性能进行测试并对测试结果作出评价的技术，在计算机系统的研制、改进、选型或选购过程中对计算机系统作出评价是很重要的。

不同的用户对系统性能的要求是不一样的。例如：科学计算的用户关心 MIPS, MFLOPS 和加速比。MIPS 表示计算机每秒钟能执行多少个百万条定点指令，MFLOPS 表示计算机每秒钟能执行多少个百万条浮点指令，加速比是度量多结点并行处理系统比单

结点处理的加速倍数，用来描述并行处理的效果。军事用户最关心的是可靠性和环境适应性；过程控制人员最关心的是实时性和可靠性；维护人员最关心的是可维护性和可用性。因此测试的目的、项目、测试方法和测试工具都不相同。

11.4.1.1 计算机性能评测的度量项目

包括以下项目。

• **性能指标**。计算机系统的硬件和软件有许多具体指标，如加法时间、字长、存储器容量、存取时间、磁盘容量、编译速度等，但要较完整地反映系统的性能还需要综合起来考虑，有 3 种类型的综合性指标：第一，工作量类，吞吐率、指令执行速率和数据处理速率等属于工作量类，例如计算机系统在单位时间内能处理的事务数；第二，响应性类，是各种响应时间，例如从计算机系统获得输入到给出相应结果之间的时间；第三，利用率，系统中各种资源的利用率，也就是在给定时间内各种资源的使用时间与整个时间之比，例如 CPU 的利用率、I/O 通道的利用率、操作系统的利用率和数据库的利用率等。

上述性能指标在一定程度上反映了计算机系统的性能和特征，但有时不够正确。必须注意，上述性能

指标与系统的工作负荷以及系统特性有关，评价者必须清楚是在什么样的系统和负载情况下测得的性能指标，否则可能会得出错误的结论。

- **可靠性、可用性和可维护性。**参见 11.3 节。
- **环境适应性。**如过程控制计算机、抗恶劣环境计算机等工作环境要求。
- **兼容性。**一个系统的软件或硬件与另一系统或多个系统的软件或硬件的兼容能力。
- **开放性。**参见 11.3.2 节。
- **可扩充性。**指系统的软、硬件扩充能力，以提高系统性能。
- **安全性。**程序和信息安全程度，例如数据库中的数据不被破坏等。
- **保密性。**在系统内设置保密措施，确保系统内的保密信息不被非法人员窃取。
- **性能价格比。**性能价格比中的价格，除了直接购买计算机系统的价格外，还要考虑安装、培训费用，若干年的运行费、维修费。如果租用软件，还要考虑软件租用费。

11.4.1.2 评测方法

评测方法大致分成两类，即测量法和模型法。

• **测量法：**通过一定的测量工具或测量程序，测得实际运行的计算机系统的各种性能指标或与之有关的量，通过运算求出相应的性能指标。当需要比较各种计算机系统性能时，经常使用一些已编制好的典型程序来进行评测，这些典型程序称之为基准程序(benchmark)。

• **模型法：**对要评价的计算机系统建立一个模型，然后求出模型的性能指标。此法可用于已有的系统或尚未存在的系统，可方便地应用于系统的设计或改造。

11.4.2 开放系统

遵循公开标准的计算机系统称为开放系统。负责开发可移植操作系统接口(POSIX)标准的美国电气和电子工程师协会(IEEE) P1003 工作组把开放系统定义为：按照开放的接口、服务和支持的规范而实现的系统。

开放系统应能做到：

- 应用软件基本上不作修改就可在不同系统间进行移植。
- 本地或远程系统的各应用间实现互操作。
- 各应用间可方便地实现交互作用。

开放系统环境的程序界面、人机界面、系统管理工具、互操作服务、通信服务和安全性等方面都是按公开标准(国际标准、工业标准或事实标准)实现的，从而使得在这种环境中的各计算机平台间能确保应用软件的可移植性、可裁剪性和互操作性。另外硬件的系统总线和外设接口也应遵循公开的标准，以方便用户选用合适的产品。这样一来，就能方便、有效地将多台不同类型的计算机连接在一起，今后系统扩展时，能保证已有的软、硬件资源可继续使用。因此，标准是开放系统的依据，可移植性、可裁剪性和互操作性是开放系统的目的，用户有可能从多个供应商处以最佳的性能价格比选择计算机的软、硬件来构成系统。

对标准的形成产生影响的机构有：政府组织的标准化机构，特定应用领域用户组织的标准化机构和工业界生产厂家组织的标准化机构。ISO(国际标准化组织)，ANSI(美国国家标准学会)和我国的国家技术监督局为政府组织的标准化机构。一些生产厂家，为了使自己的产品能与其他产品互连和兼容，也联合起来制定标准。

11.4.3 系统兼容性

系统兼容性用来衡量一种计算机系统的软件或硬件适用于另一种或其他多种计算机系统的能力。

系统兼容性是系列计算机的基本特征，是在硬件技术迅速发展、生产厂家不断推出新产品情况下，让用户在老产品上开发的软件能继续使用的一种重要设计思想和技术措施，它节省了生产厂家的开发投资，加快了计算机的研制过程，保护了用户已有的资源。

由于计算机硬件技术的发展，计算机的处理能力和应用领域不断扩大，对软件的需求不断提高，但软件开发效率低、周期长、费用高，因此在厂家开发新计算机或用户更新计算机时希望原有、的软件能在新机器上运行，对计算机的硬件产品也希望能兼容。

计算机的兼容性表现在软件和硬件的各个方面。

• **硬件设备或部件兼容。**这是指一种设备或部件可不加改动地用于多种计算机，要求设备或部件要遵循标准或规范，包括功能和接口等。

• **机器语言程序兼容。**要求计算机的指令系统兼容，对计算机的体系结构要求极为严格。如果要想实现兼容的两台计算机的指令系统、体系结构略有不同，可采用硬件仿真或软件模拟的方法实现兼容，这将使程序的运行速度明显降低，尤其是采用软件模拟的方法效率更为降低。

• **汇编语言程序兼容。**实现汇编语言兼容遇到的问题基本与机器语言程序兼容相同。但用汇编语言编写的程序需要经过编译(汇编)后才能在计算机上运行,因此可以通过汇编解决一些问题。例如用户程序和操作系统接口有差异,可通过汇编程序转换解决,缺少某条指令可由汇编程序用广义指令来实现其功能,如果实现兼容的计算机体系结构差异较大,则难以实现汇编语言程序兼容。

• **高级语言程序兼容。**高级语言程序文本与计算机体系结构无关,因此容易实现兼容。但要采用标准化的语言文本,由于不同计算机对语言文本有不同的解释,可能会影响兼容性的实现。

• **系统软件兼容。**系统软件是指控制计算机运行和为用户程序服务的一类软件,主要包括操作系统、编译程序和数据库管理系统等。编译程序将用高级语言编写的源程序编译成机器语言程序,它与计算机的体系结构密切相关,因此不同体系结构的计算机之间难以实现编译程序兼容。操作系统是与计算机体系结构相关的部分和与用户程序相关的部分,前者构成操作系统内核,不同计算机之间不能兼容,后者构成操作系统的外层,可以实现兼容。数据库管理系统也与计算机体系结构无关,可在操作系统支持下实现兼容。

• **软件系统兼容。**这是指用户的应用程序的兼容问题,要求应用程序在兼容的环境下开发,例如采用标准的高级语言或兼容的数据库等。

系列机和兼容机是实现系统兼容性的典型例子。系列机是计算机体系结构相同、具有标准的外设接口、软件向上兼容而性能和价格不同的计算机系列。兼容机是一些厂家为了利用别人的软件成果而开发的,软件兼容,有的还实现了插件兼容。

随着计算机的发展,实现兼容性的技术和方法越来越受到人们的重视,但实现兼容性的要求往往给新产品的开发带来约束,例如指令系统的向上兼容性使得指令系统越来越庞大。

系统的开放性和兼容性很难量化,但是很重要,尤其是在选购时或需要考虑系统的可扩展性时。

下面介绍性能评测的具体方法。在这里分为“评估”和“测试”两类方法。评估基本上是基于一些原始数据进行推算,典型的有 MIPS、MFLOPS、数据处理速率 PDR、综合理论性能 CTP 等。测试是用基准测试程序来度量计算机的性能。

11. 4.4 性能评估

1. MIPS 和 MFLOPS

MIPS 表示每秒百万次指令,用来描述计算机的定点运算速度。常用的有峰值 MIPS、基准程序 MIPS 和以特定系统为基准的 MIPS。峰值 MIPS 通常是以指令集中最基本指令的执行速度计算得到的,如果指令集中最基本指令的执行需要 a 个机器周期,每一机器周期为 t 微秒,则其峰值 MIPS 值为 $1/(at)$ 。一台处理机的平均 MIPS 是将指令集中各条指令加权处理后的平均值。基准 MIPS 值是用基准程序测得的 MIPS 值,对于某一台计算机,所用的基准程序不同,测得的 MIPS 值也不相同,这主要是由于不同基准程序中各种指令的使用频度不同而引起的。以特定系统为基准的 MIPS 通常以 VAX11/780 系统的性能定义为 1MIPS,以它为基准测得相对 MIPS 值。

MIPS 指标用于评价同一厂商生产的同一计算机系列的定点运算速度比较正确,因为这些计算机有相似的系统结构、操作系统、语言和编译器,尤其是有相似的指令集;如果两台机器的结构和指令集不同,那么用 MIPS 值来比较它们的运算速度可能会得出错误的结论。

MFLOPS 表示每秒百万次浮点运算速度,衡量计算机的科学计算速度,常用的有峰值 MFLOPS 和以基准程序测得的 MFLOPS,其峰值 MFLOPS 通常以最快的浮点指令执行时间计算得到,如果执行一条最快的浮点指令需要 b 个机器周期,每个机器周期为 t 微秒,则峰值 MFLOPS 为 $1/(bt)$ 。以基准程序测得的 MFLOPS 的计算方法是:设基准程序 A 在计算机上执行的时间为 t 微秒,程序 A 中含 F 个浮点操作,则其 MFLOPS 等于 F/t MFLOPS。由于程序 A 中还包含一些非浮点操作,会影响其结果,所以有时根据问题的复杂性分析,从数学模型中计算出来。

MFLOPS 可用于比较和评价在同一系统上求解同一问题的不同算法的性能,还可用于在同一源程序、同一编译器以及相同的优化措施、同样运行环境下对不同系统测试浮点运算速度。由于实际程序中各种操作所占比例不同,因此测得的 MFLOPS 也不相同。MFLOPS 值没有考虑运算部件与存储器 I/O 系统等速度之间相互协调等因素,所以只能说明在特定条件下的浮点运算速度。

2. 吉普森混合法

是由吉普森(Gibson)提出的一种评价计算机性能的方法,也称为混合比例算法。

早期用加法指令的运算速度来表征计算机的运算速度,但这不够正确,因此出现了一些改进的办法,其中之一为等效指令速度法。

等效指令速度法是通过各类指令在程序中所占的比例(W_i)进行计算得到的。若各类指令的执行时间

$$T = \sum_{i=1}^n w_i t_i$$

为 t_i ，则等效指令的执行时间 $T = \sum_{i=1}^n w_i t_i$ ，式中 n 为指令类型数。Gibson 对 IBM7090 机上运行的程序进行了统计分析，于 1970 年提出了一个各类指令在程序中所占的比例值，曾得到广泛应用。基于这个比例的等效指令速度法就称为 Gibson 方法。

采用这种固定比例的方法对某些程序来说可能严重偏离实际，尤其是对复杂的指令集，其中某些指令的执行时间是不固定的，数据的长度，cache 的命中率、流水线的效率等都会影响计算机的运算速度，因此后来又发展了其他评价方法。

3. 数据处理速率 PDR(processing data rate)

PDR 的估算可用公式表示如下： $PDR=L/R$

式中： L 为指令操作数平均长度，即每条指令传送数据的平均位数，单位为 b ； R 为指令的平均执行时间，单位为 μs 。

$$L = 0.85G + 0.15H + 0.4J + 0.15K$$

$$R = 0.85M + 0.09N + 0.06P$$

式中： G 是定点指令字长 ($>24b$)，单位为 b ；

H 是浮点指令字长 ($>30b$)，单位为 b ；

J 是定点数长度，单位为 b ；

K 是浮点数长度，单位为 b ；

M 是定点加法时间，单位为 μs ；

N 是浮点加法时间，单位为 μs ；

P 是浮点乘法时间，单位为 μs 。

参加运算的两个数，其中一个数在累加器中或在用作累加器的主存单元中，另一个在主存储器中，运算结果保存在累加器或用作累加器的主存单元中。

PDR 主要是对 CPU 和主存数据处理速度进行计算而得出的，它允许并行处理和指令预取的功能，这时，所取的是指令执行的平均时间。带有 cache 的计算机，因为存取速度加快，其 PDR 值也就相应提高。

PDR 不能全面反映计算机的性能，但它曾是美国及巴黎统筹委员会用来限制计算机出口的系统性能指标估算方法。1991 年 9 月停止使用 PDR，取而代之的是 CTP(综合理论性能)。

4. 综合理论性能 CTP(composite theoretical performance)

是美国政府为限制较高性能计算机出口所设置的运算部件综合性能估算方法。CTP 以每秒百万次理论运算 MTOPS 表示，从 1991 年 9 月 1 日启用。

CTP 的估算方法为首先算出处理部件每一计算单元(如定点运算单元，定点乘法单元，浮点加单元、浮点乘法单元)的有效计算率 R ，再按不同字长加以调整，得出该计算单元的理论性能 TP ，所有组成该处理部件的计算单元 TP 的总和即为综合理论性能 CTP。

$$\text{定点加法单元的 } R = \frac{1}{3 \times t_{\text{定点}+}}$$

式中 $t_{\text{定点}+}$ 为定点加法执行时间 (μs)；

$$\text{定点乘法单元的 } R = \frac{1}{t_{\text{定点}\times}}$$

式中 $t_{\text{定点}\times}$ 为定点乘法执行时间 (μs)；

$$\text{浮点加单元的 } R = \frac{1}{t_{\text{浮}+}}$$

$$\text{浮点乘单元的 } R = \frac{1}{t_{\text{浮}\times}}$$

式中 $t_{\text{浮}+}$ 和 $t_{\text{浮}\times}$ 分别是浮点加和浮点乘的执行时间 (μs)。

按操作数字长对 TP 加以调整。

$$TP = RL,$$

$$L = (1/3 + WL/96)$$

式中 WL 为字长。

对单个计算单元的处理部件：

$$CTP = TP$$

对由 n 个计算单元组成的处理部件：

$$CTP = TP_1 + C_2 TP_2 + \cdots + C_n TP_n$$

TP_1 为 n 个 TP 中的最高值。对共享存储的多计算单元的处理部件，其 $C_2 = C_3 = \cdots = C_n = 0.75$ 。

11.4.5 基准测试程序

1. Whetstone 基准程序

Whetstone 的一组综合程序，主要由浮点运算、整数算术运算、超越函数、功能调用、数组变址、条件转移等程序组成。主要用于评测浮点运算功能。用 Fortran, PASCAL 语言编写，程序规模不大，cache 利用率高。当今已很少使用。

2. Dhrystone 基准程序

Dhrystone 基准程序用于测试编译系统和定点运算、逻辑运算的性能，程序规模不大，用 C 语言编写的版本有 100 条语句。当今已很少使用。

3. UNPACK 基准程序

LINPACK 基准程序用于测试系统在解算密集线性代数方程组的性能，该程序由美国 Jack J. Dongarra 发表于 1976 年。由于在科学和工程计算中线性代数方程应用非常广泛，而且 UNPACK 的适应性很强，在标量机、向量机和并行机上都能较方便地移植，并能得到较满意的效率，因此得到计算机产商的欢迎。目前 UNPACK 基准程序已成为工业上广泛用来评测不同计算机系统性能的行业标准。

UNPACK 由 Fortran 语言写成，由于密集线性代数方程组中存在大量浮点运算，所以 UNPACK 可用于评价计算机的浮点运算性能。

用优化的 UNPACK 基准程序和扩大的矩阵(阶数 $n=1000$)可以测得系统接近于理论峰值性能指标。所谓理论峰值性能是根据处理器数量和时钟周期计算所得的理论结果，是计算机的最高极限速度。当矩阵阶较小时，性能明显下降。由于 UNPACK 测试运行的是实际使用程序，比理论峰值更能反映系统的性能。

4. SPEC(system performance evaluation cooperative)基准程序

随着计算机技术的飞速发展，厂商和用户都希望有一个标准、客观、公正的评测工具，在此背景下，一个非赢利组织 SPEC 于 1988 年成立。SPEC 最初是由几个工作站厂商共同发起的，发展十分迅速，现已有 40 多个成员，每个季度都要公布一次成百上千的性能测试结果。

SPEC 下设 3 个一级分会，一级分会还设有若干二级分会。OSG(open system group)是 SPEC 最早最大的一级分会；1994 年新成立了 HPG(high-performance group)分会，1996 年，一些图形性能测试组加入 SPEC，成为第三个一级分会 GPC(graphics performance characterization group)。

SPEC 于 1989 年发表第一套标准化测试基准程序 SPEC89，以后多次发表新的基准测试程序。SPEC 的基准测试程序全部选自实际的应用程序。1992 年 SPEC 推出 SPEC92 替代了 SPEC89，1995 年推出了 SPEC95 替代了 SPEC92，2000 年又推出了 SPEC CPU2000 取代了 SPEC95。

SPEC89 包括整数和浮点数运算在内的 1 组共 10 个基准测试程序，其中用于测量机器整数运算性能的程序共 4 个，称为 SPEC_{int} 89，用于测试浮点数运算性能的共 6 个，称 SPEC_{fp} 89。

SPEC92 是在 SPEC89 的 10 个基准程序基础上再增加了 4 个测试整数运算性能的程序和 8 个测试浮点数性能的程序，分别称之为 SPEC_{int} 92 共 8 个程序)和 SPEC_{fp} 92 共 14 个程序)。

每个基准测试程序是基于不同的应用写成的程序 SPEC92 主要测试 32 位中央处理器、主存储器、编译器和操作系统的性能。

SPEC_{int}92 和 SPEC_{fp} 92 两组是分别进行测试的，测试方法是先将基准程序先在 VAX11/780 机上运行，记下运行时间，求出几何平均值；然后把这组程序在被测机器上运行，也记下时间，并求出几何平均值；再用 VAX11/780 机上测得的几何平均值去除被测机器上测得的几何平均值，测得的结果即为被测机器的 SPEC_{int} 值和 SPEC_{fp}。SPEC 值越高说明机器性能越好。

SPEC95 由两组基准测试程序组成：

- SPEC CINT95 是用 C 语言写成的整数高强度计算基准程序，由 8 个基准程序组成。
- SPEC CFP95 是用 Fortran 语言写成的浮点高强度计算基准程序，由 10 个基准程序组成。

表 11.2 SPEC CPU2000 的整数和浮点基准程序

项目 组成	基准程序	语言	常驻存储(MB)	虚存(MB)	应 用
SPEC _{int} 2000	164. gzip	C	181	200	压缩
	175. vpr	C	50	55.2	FPGA 电路布局和选定路线
	176. gcc	C	155	158	C 程序语言编译
	181. mcf	C	190	192	组合优化
	185. crafty	C	2.1	4.2	游戏：下棋
	197. parser	C	37	62.5	字处理
	252. eon	C++	0.7	3.3	计算机视觉
	253. perlhm	C	146	159	Perl 程序语言
	254. gap	C	193	196	组合论：解释器
	255. vortex	C	72	81	面向对象数据库
SPEC _{fp} 2000	256. bzip2	C	185	200	压缩
	300. twolf	C	1.9	4.1	区域和路由模拟器
	168. wupwise	F77	176	177	物理：量子力学
	171. swim	F77	191	192	浅水模拟
	172. mgrid	F77	56	56.7	多栅格解算器：3D 势场
	173. applu	F77	181	191	偏微分方程
	177. mesa	C	9.5	24.7	3D 图形库
	178. galgel	F90	63	155	流体动力学
	179. art	C	3.7	5.9	图像识别/神经网络
	183. eguske	C	49	51.1	地震波传播模拟
	187. facerec	F90	16	18.5	图像处理：面貌识别
	188. ammp	C	26	30	计算化学
	189. lucas	F90	142	143	数论
	191. fma3d	F90	103	105	有限分子碰撞模拟
	200. sixtrack	F77	26	59.8	核物理加速器设计
	301. apsi	F77	191	192	气象学：污染分布

SPEC95 基准测试程序都是从实际的应用中优选出来的，重点测试计算机的处理器、存储结构和编译器的性能，对 I/O、网络和图形部件等的测试未加考虑，经光驱安装的 SPEC 大约需要 150MB 硬盘空间和 64MB 主存以保证运算的需要。

SPEC95 的主流版本是用于 UNIX 操作系统的版本。

SPEC95 以 SUN SPARC Station 10/40 工作站作为参考机。CINT 和 CFP 两组基准程序在参考机上运行的时间约为 48 小时，每个基准程序经参考机的运行和测试得到相应的参考时间，将用于被测试机器的 SPEC95 性能指标的计算。

为了保证测试结果的公正性和可比性，所有的测试必须在 SPEC 提供的工具环境中运行，包括配置文件的生成、程序的编译、运行环境的建立和实施、报告结果的产生等环节在内，整个测试过程在严格的规则下自动完成，任何手工干预是不必要的且是绝对禁止的。

SPEC95 对计算机性能的测试有两种不同的方法：一种是测试一台计算机完成单个任务有多快，称为速度测试；另一种

是测试一台计算机在一定时间内能完成多少个任务，称为吞吐量或速率测试。SPEC95 的两种性能指标又分为基本的和非基本的两类。基本的是指在编译基准程序的过程中严格限制所用的优化选项，非基本的可以使用不同的编译器和编译选项以得到最好的性能，这样做的结果使得测试结果的可比性降低。测试结果提交给 SPEC 组织审定，得到 SPEC 的确认和批准后，才能在 SPEC 的专利(SPEC 通信)上发表，并在 Internet 的 SPEC 网址上公布。在呈送给 SPEC 的报告中，基本指标是强制提供的，非基本指标是可选的，非基本指标更多地加进了编译器的性能因素。

SPEC 规定在 SPEC95 参考机上测试每个基准程序的得分为 1，把在参考机上完成每个基准程序的运行时间称为它的 SPEC 参考时间。在 SPEC95 的测试中，CINT95 和 CFP95 中的每个基准程序各自计算得分，然后再用这些得分计算各项合成指标。如果被测计算机的得分为 10，则表示该系统的相应能力是参考机的 10 倍。

在进行速度测试时，每个基准程序的得分称为它的 SPEC 比率(SPECratio)。一个基准程序的 SPEC 比率用它在被测系统上运行时间除它的 SPEC 参考时间。设某基准程序为 A，则

A 的 SPEC 比率=A 的参考时间/A 的运行时间。

合成指标计算如下。

- **SPECint 95:** CINT95 的 8 个整数基准程序使用非基本优化编译时，取这 8 个 SPEC 比率的几何平均值。

- **SPECint-base 95:** CINT 的 8 个整数基准程序使用基本优化编译时，取这 8 个 SPEC 比率的几何平均值。

- **SPECfp 95:** CFP 的 10 个浮点基准程序使用非基本优化编译时，取这 10 个 SPEC 比率的几何平均值。

- **SPECfp-base 95:** CFP 的 10 个浮点基准程序使用基本优化编译时，取这 10 个 SPEC 比率的几何平均值。

SPEC95 和 SPEC92 的基准程序、运行规则和使用的 SPEC 工具都不相同，因此无法提供这两类测试结果指标之间的相应转换关系。

SPEC CPU2000 由 12 个整数基准程序和 14 个浮点基准程序组成，表 11.2 列出各个基准程序的名称、用何种语言编写、所需的主存储器空间和虚存空间以及程序的应用类型。表中 F77 为 Fortran77，F90 为 Fortran90。

图 11. 17 示出 4 个不同的系统的整数和浮点性能, 其中 3 个使用 Alpha21164 芯片, 1 个使用 Alpha 21264 芯片, 3 个 Alpha21164 系统是 500MHz AlphaStation 500/500, 500MHz Personal Work Station 500au 和 533MHz AlphaServer 4100 5/533, 1 个 Alpha 21264 系统是 500MHz Alpha Server DS20 6/500。参考机是 300MHz Sun ultra 5_10, 其得分定为 100。

Alpha 21164(主频 500MHz/530MHz)芯片内含 2 级 cache, L1 容量为 8KB(指令)+8KB(数据), LZ 的容量为 96KB, Alpha 21264 芯片内含 1 级 cache, 容量为 64KB 指令)+64KB(数据)。

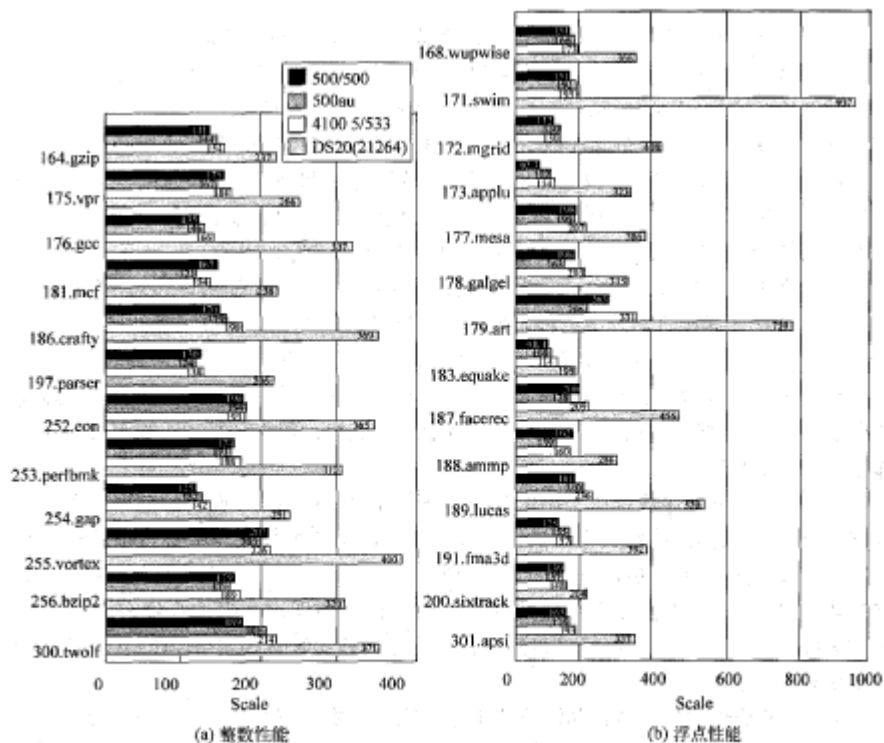


图 11.17 性能: 3 个 Alpha 21164 系统 (500MHz Alphastation 500/500, 500MHz Personal workstation 500au 和 533MHz Alphaser 4100 5/533) 和 1 个 Alpha 21264 系统 (Alpha Server DS20 6/500)

比较 21164 500MHz 系统与 Alpha Sever DS20 6/500 系统(采用 21264 处理器), 其主频都是 500MHz, 但是存储器系统(cache, 主存的容量和存取时间)以及 CPU 的内部结构不同, 后者的基准程序性能却提高很多, 21264 可以同时执行 8 个取数操作和 8 个存数操作, 以及提供无序执行(参见 11.1.4), 而 21164 只能同时执行 2 个取数操作和 2 个存数操作, 不支持无序执行。

本节中所有的结果使用同一组编译器, 不同的编译器将导致不同的结果。

SPEC CPU2000 基准程序测试了 CPU, 存储器系统和编译器的性能。

5. TPC(transaction process performance council)基准程序

事务处理性能测试委员会 TPC 是一个专门负责制定计算机事务处理能力测试标准并监督其执行的组织。TPC 由硬件供应商、软件供应商和用户组成, 旨在建立一套完整的基准程序以评测包括事务处理、数据库处理、企业计算和决策支持等广泛的商业计算。TPC 成立于 1988 年, 当时只有 8 个公司参加, 目前已有 40 多个成员。

TPC 在 1989 年 11 月发布了其成立后的第一个标准 TPC-A, 1990 年 8 月推出 TPC-B, 1992 年推出 TPC-C, 1994 年公布 TPC-D, 后来又推出了 TPC-H 和 TPC-R, 1998 年公布了 TPC-W。目前 TPC-A, TPC-B, TPC-D 已不用。

TPC 按商业领域的公共理解来定义事务(transaction)这个术语, 认为事务就是商业活动中货物(库存管理)、服务(订票)、金钱(银行)的交换。典型的事务一般都会涉及数据库系统中数据的更新。

TPC-A 基准程序规范用于评价在联机事务处理(OLTP)环境下的数据库和硬件的性能, 不同系统之间用性能价格比进行比较。该基准程序模拟了一个银行出纳员进行存款和出票时的网络, 其性能用每秒完成的事务数 TPS 来表示。性能价格比(实际上是价格性能比)用系统的价格除以系统的 TPS 值来表示, 即以每 TPS-A 需要多少千美元(K\$/TPS-A)来表示, 系统价格包括硬件的购置费和安装费、5 年期内的软件租金和硬件维修等全部费用。

TPC-B 测试的是不包括网络的纯事务处理量。系统仅由 1 个中央处理器、海量存储器和 1 个数据库组成, 测试该系统 1 秒钟内能完成的银行类型的事务处理数量, 而不考虑网络、终端等其他系统组成部分。

TPC-C 测试的是联机订货系统, 例如制造业、流通业以及商店一般订货系统, 它模拟了联机事务处理(OLTP)环境中分销商的订单一输入(order-Entry)应用环境。TPC-C 测试比 TPC-A 和 TPC-B 复杂得多。它包括了混合的 5 种只读和更新密集的事务处理类型。模拟了实际的应用。TPC-C 定义的 5 种处理类型为:

- New-Order。输入一张新的订单, 平均包含 10 种商品, 每一仓库在其产品目录上库存有 100000 种商品。
- Payment。更新客户账号的金额以反映新的交易付账。
- Delivery.Order。交付提货单, 处理一批(10 张)提货单。
- Order-status。检索以前存放的订单状态。

- stock-level。检查本地仓库的库存情况以确定系统是否缺货。

测试的结果(事务吞吐量)用每分钟完成的事务处理数(TPM)来表示。

事务吞吐量和性能价格比是 TPC-C 的两个重要测试指标。

TPC-C 的性能取决于计算机、操作系统和数据库等很多因素,许多计算机厂商及数据库厂商都将它们的产品的测试结果公布于 Internet 网上(<http://www.tpc.org>)。就操作系统而言 windows NT 的性能价格比 UNIX 好,但 UNIX 在性能上具有很好的可扩展性。普遍认为 TPC-C 是接近实际 OLTP 运行环境的较好的基准测试程序,经常用它来测试系统的性能。

TPC-D, TPC-H 和 TPC-R 测试的都是决策支持系统。它们对决策支持系统形成特殊的查询。TPC-H 包括一组面向特殊查询和并发数据更新的商务。它检验了大量数据,执行高复杂度的查询,并对关键性的商业问题作出回答。TPC-R 与 TPC-H 相似,但允许基于查询的先进知识进行优化。它们的性能用每小时完成的 TPC-H 或 TPC-R 组合查询数来表示。

TPC-W 是基于 Web 商业(commerce)的测试标准,用来表示一些通过 Internet 进行市场服务和销售的商业行为,如零售店、机票预定等,TPC-W 测试系统处理用户浏览商业 Web 站点和通过商业 Web 站点操作订单的性能。由于在整个 Web OLTP 环境中,网络接入和人机交互情况是不同的,但 TPC-W 没有考虑这些开销,所以 TPC-W 可以看作是一个服务器的测试标准。

TPC 基准测试程序在商业界建立了用于衡量系统性能和性能价格比的标准,生产厂家根据 TPC 基准程序进行测试,并向 TPC 提交 1 份详细的完全公开的报告,且经审查通过后才能正式公布测试结果。

比较不同系统的性能优劣需要公共的衡量标准,用户需要公共标准来帮助他做出购买决定,厂家要用它为产品进行市场宣传,TPC 为用户和厂家提供了帮助。TPC 标准还可以进行裁剪,以便恰当地表示具体的系统。但是由于用户的应用系统环境的多样性,而 TPC 的测试标准只能是相对简单的模型,所以 TPC 的测试标准不能当作精确度量具体系统性能的工具。用户应深入了解 TPC 测试标准,并对其各种模型进行分析,以确定哪一类的用户交互处理、数据库设计、数据库大小、事务复杂性、存储备份测试等模型更适合于用户的系统环境,然后用来对系统进行测试和比较。

6.对计算机性能评测的评估

基准测试应该公正、准确。公正是基本原则,准确是技术上的要求,要尽量不发生或少发生偏差。由于计算机性能评测的复杂性,所以对其公正性和准确性的争论颇多。

- 公正性:所有的基准测试组织都是中立和不赢利的,但是任何一个测试组织都可能与厂商和用户团体有联系,因此用户与厂商之间,不同厂商之间的矛盾会反映到测试组织中来。客观地讲,主要的基准测试还是相对公正的,但有时会发生一些不妥当的现象,所以对基准测试不可不信,但也不可迷信。

- 准确性:计算机系统性能是一个整体的综合性能,包括所有软硬件的有机结合,但是所有的基准测试都有局限性,例如 SPEC CPU 可以测试 CPU、存储器和编译能力,但对 I/O 和操作系统还是无能为力,因此不能全面反映系统的性能。另外基准测试未对如何检测系统的瓶颈给以足够重视,所以有些厂商扩大宣传其长处,而掩盖其不足之处。为了反映系统性能,TPC 在进行测试时建立一个模拟的用户环境,结果导致测试费用巨大,而广大用户还是认为这种模拟环境与实际使用情况相差太大。准确性的另一个问题是计算机不少重要性能不可量化,如可靠性、可用性和可维护性。所以基准测试所得结果基本可信,但不足以准确反映实际使用效果。

第 12 章信息安全技术

信息安全是一项具有多种功能需求的系统工程,信息系统应当保障信息的机密性、完整性、不可否认性、可用性及可控性等功能特性。其中,机密性保证信息不泄露给未授权的用户、实体或过程;完整性保证信息的完整和准确,防止信息被非法修改;可用性保证信息及信息系统确实能为授权者可用;而可控性保证对信息的传播及内容具有控制的能力,防止为非法者所用。信息安全不仅是一种技术问题,它更是有关业务和管理的问题,因为只靠技术是不能为人们提供所有问题的答案的。关键在于要采取适当的防卫措施应付组织所面对的具体危险,并将这些措施渗透到日常的商务运作中,而不是仅仅作为可有可无的或是需要时才使用的额外的措施。

12.1 访问控制机制和方法学

访问控制涉及到各种机制,包括物理的、逻辑的和管理方面的,访问控制系统保证只有特定的经过授

权的人或进程被允许访问一个系统。

对需要加以保护的关键资源进行访问控制是信息安全的一个基本要求。为了保证关键资源的安全性并防止对资源滥用、泄漏及破坏,大多数组织都实施访问控制来确保其关键业务决策信息的完整性和安全性。在系统的安全配置方案中,访问控制技术需要在不同的层次进行实施,包括主机操作系统、数据库和应用层等。在某些情况下,尤其是数据库和应用层级别,可能需要第三方访问控制产品的参与。访问控制管理模型的构造可以是集中型、分散型或混合型。

一些访问控制技术利用了使用者的某些特征(如此人是谁、此人知道什么、此人拥有什么等),利用的方式也不尽相同。简单的仅利用标识符和固定的口令进行访问控制,复杂的使用了硬件口令字发生器以至于高级生物识别技术(如视网膜扫描和指纹识别等技术)。为了减轻系统访问控制管理的负担,很多组织倾向于使用简化的实施方案或单点登(Single Sign-On, SSO)方案。生物测定学可以用于识别和鉴定用户的身份。在信息的访问控制中,生物测定已经迅速成为广泛使用的方法,因为其能够根据人类本身所具有的属性有效地对个体进行识别。

12.1.1 单点登录技术

大约 50 年以前,系统设计者认识到对计算机系统操作进行跟踪的必要性,因此便出现了一种身份标识,即登录 ID 号,几乎与此同时出现了口令字,两者可以用来证明用户的身份。随着信息技术和网络技术的发展,各种应用服务的不断普及,用户每天需要登录到许多不同的信息系统,如网络、邮件、数据库、各种应用服务器等。每个系统都要求用户遵循一定的安全策略,比如要求输入用户 ID 和口令。随着用户需要登录系统的增多,出错的可能性就会增加,受到非法截获和破坏的可能性也会增大,安全性就会相应降低。对用户来说,他们必须记忆大量的口令字,这就为用户带来了极大的不便。在这种情况下,出现了一种单点登录技术(SSO)来解决这种问题。

单点登录简单地说就是通过用户的一次性认证登录,即可获得需访问系统和应用程序的授权,在此条件下,管理员无需修改或干涉用户登录就能方便地实施希望得到的安全控制。

单点登录的应用机制随着时间的变化也在不断地发生变化。一种方式是利用 Kerberos 机制,Kerberos v5 是业界的标准网络身份认证协议,该协议是在麻省理工学院起草的,旨在给计算机网络提供身份认证。Kerberos 协议的基础是基于信任第三方,它提供了在开放型网络中进行身份认证的方法,认证实体可以是用户或用户服务。这种认证不依赖宿主机的操作系统或主机的 IP 地址,不需要保证网络上所有主机的物理安全性,并且假定数据包在传输中可被随机窃取篡改。另一种方式是使用外壳脚本机制,通过原始认证进入系统外壳,然后外壳就会激发各种专用平台的脚本来激活目标平台的账号以及资源的访问。这种方式简化了用户的登录,但其没有提供同步的口令字以及其他管理方法。另外,单点登录的实施可以采用一些方案,比如通用安全服务应用程序接口(generic security service application program interface, GSS-API),分布式计算环境(distributed computing environment, DCE)等。

目前,实施一个 SSO 平台具有多种配置方法和各种可用的选项,各种机制本身所具有的缺陷也较好地得到了解决。SSO 产品的功能也不仅仅是进行单一用户认证和口令管理,而且能够解决一些复杂的问题,比如终端系统的集中管理和终端用户的管理等。

一个理想的 SSO 产品具备以下的特征和功能。

- **常规特征:** 支持多种系统、设备和接口。
- **终端用户管理灵活性:** 包括通常的账号创建、口令管理,以及用户识别。口令管理包括口令维护、历史记录以及文法规则等。支持各种类型的令牌设备、生物学设备。
- **应用管理灵活性:** 若多个会话同时与一个公共主体相关,设备场景管理能保证若其中一个会话发生改变,其他相关会话自动更新;应用监控能监控特定信息的使用;应用融合可将各种应用绑定在一起来保证应用的一致性。
- **移动用户管理:** 保证用户在不同的地点对信息资源进行访问。
- **加密和认证:** 加密保证信息在终端用户和安全服务器之间传输时的安全性;认证保证用户的真实性。
- **访问控制:** 保证只有用户被授权访问的应用可以提供给用户。
- **可靠性和性能:** 包括 SSO 和其他访问控制程序之间的接口的可靠性和性能以及接口的复杂度等。

12.1.2 集中式认证服务

对计算机网络信息资源安全、可靠、有效地存取是信息系统安全的关键,身份认证技术是主要的实现手段。在生物测定技术还不能大规模使用的今天,系统密码仍然是身份认证的主要武器,使用密码的身份认证技术是保障网络安全最基本的手段之一。

在通常的计算机网络密码的管理和使用上,基本上是一种分布式的密码管理体系。但随着计算机技术

的不断发展和普及，分布式的密码管理已经远远不能适应网络社会的发展，于是人们对集中式的身份认证和管理方法提出了需求。

RADIUS(remote authentication dial-in user service)、TACACS(terminal access controller access control system)以及 DIAMETER 集中式认证服务正是在这种情况下产生的。它们具有很强的灵活性而且易于实施，不仅大大增强了远程访问的安全性，而且降低了远程访问服务器的客户端的管理所需的时间和复杂度。RADIUS, TACACS 以及 DIAMETER 能够实现所谓的 AAA(authentication, authorization, accounting)服务。IETF 在 1998 年成立了 AAA 工作组来开发网络访问中所需的认证、授权和记账服务，其目的在于开发一个基本协议来支持一系列不同的网络访问模型，包括传统的拨号网络服务器以及移动 IP、漫游操作等等。

RADIUS 最初是由 Livingston 公司为他们们的网络访问服务器(NAS)所开发的，用于辅助分时操作、巩固互联网服务提供者的账单信息和连接配置。针对远程用户的 RADIUS 协议，采用分布式的客户机/服务器结构完成密码的集中管理和其他身份认证功能；网络用户通过 NAS 访问网络 NAS 同时作为 RADIUS 结构的客户端，AAA 功能通过 NAS 和安全服务器或 RADIUS 服务器之间的 RADIUS 协议过程完成，而用户的控制功能在 NAS 实现。这种结构具有开放、可伸缩性强等优点，因此很适合与其他的第三方产品协同工作。

目前通常所指的 TACACS 实际上代表此协议的两个发展阶段。最初的 TACACS 具有有限的功能并使用 UDP 协议。1990 年，协议得到扩展，包含了一些附加的功能，传输也采用 TCP 协议。为了向上兼容，原始的功能作为扩展功能的一个子集，新的协议称为 XTACACS(Extended TACACS)，几乎目前所有的 TACACS 产品都是基于这种扩展的协议，其在 RFC1492 中有详细的描述。Cisco 系统在它的产品中采用 TACACS 协议来对 AAA 服务进行支持，增加了对所有 NAS 服务传输的加密功能，并且对其进行扩展从而允许在认证交换中使用任意长度以及内容的参数。这种新的协议称为 TACACS+。事实上，TACACS+与原始的 TACACS 并不相像，包格式也不兼容，一些服务器为了保持兼容性通常对两种格式都进行支持。

1998 年，Pat Calhoun 和 Allan Rubens 向 IETF 提交 DIAMETER AAA 框架并成为 IETF 的一个草案标准。DIAMETER 不是单词的缩写，而是为了同 RADIUS 相对应。DIAMETER 的设计主要用来支持漫游应用并克服 RADIUS 和 TACACS 协议的扩展限制。它提供了基本协议来支持大多数的 AAA 扩展，包括 NAS;移动 IP 和基于 Web 的需求。

1. AAA 服务

AAA 服务的主要特征包括：分布式的(客户机用及务器)安全模型，认证式交易，灵活的认证机制和协议的可扩展性。

分布式安全模型将认证过程和通信过程分开，因此可以将用户的认证信息统一集中在单一的集中式数据库中。，网络访问设备(比如 NAS)作为客户机将用户信息传送给 AAA 服务器并对返回的响应进行处理。服务器接收到用户的连接请求，对用户进行认证后，将传送服务所需的配置信息发送给客户机 NAS。返回的信息包括传输和协议参数、附加认证需求(如 SecureID)、认证指示(如允许的服务)以及记账请求等，其过程如图 12.1 所示。

客户机和服务器之间通过数据验证来确保数据的完整性，而敏感信息(如口令)则使用共享密钥进行加密以保证其机密性，从而防止口令以及其他认证信息在传输的过程中被监听或获取。当数据在公网中传输时这一点尤其重要。

AAA 服务器能够支持多种认证机制，这种灵活性是 AAA 的一个主要特征。用户的访问可以通过口令认证协议(password authentication protocol, PAP), 请求握手认证协议(challenge handshake authentication protocol, CHAP)以及标准的 UNIX 登录过程来进行验证，或者服务器可以作为一个代理将认证工作交给其他的认证服务器来处理。

因为各种技术不断在发展变化,AAA 服务器使用的协议可以进行扩展。RADIUS, TACACS 以及 DIAMETER 均使用变长的属性值，从而可以使用任何新的参数而不会干扰已经存在并执行的协议。DIAMETER 的框架方法通过将传送机制进行标准化从而能够提供附加的可扩展性来支持多种定制的 AAA 模块。

从一个管理者的角度来看，AAA 服务器具有下列优点：

- 缩短了用户建立和维护的时间，因为所有用户多在同一主机进行维护。
- 需要较少的安全管理培训，因为只需要学习一种系统规则。
- 易于审计，因为所有的登录和认证都通过同一系统进行请求。
- 方便了用户的使用，因为所有访问方法中的用户接口都是一致的。

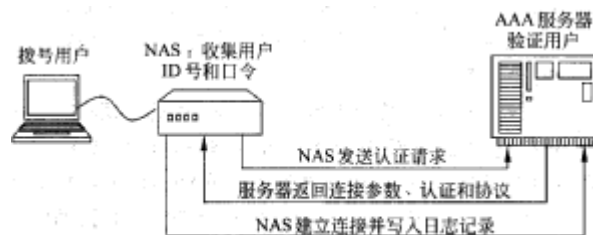


图 12.1 集中式 AAA 服务的主要特征

- 访问信息的扩散更加快捷，因为信息只需要复制给有限的 AAA 服务器。
- 通过对使用附加认证体制(如 SecureID)的支持，从而增强了安全性。
- 可扩展性设计，易于添加新的设备而不会影响已有的配置。

2 . RADIUS(remote authentication dial-in user service)

RADIUS 是目前最常用的 AAA 服务，其普遍性应归功于 RADIUS 的源代码的公开性。用户可以很快地在各种平台上使用并可以增加自定义的特征。一个基本的 RADIUS 服务器的实施主要与两个配置文件相关：客户机配置文件和用户文件。客户机配置文件包含客户机的地址和用于认证事务的共享秘密，用户文件包含用户的识别和认证信息(比如用户 ID 和口令)以及连接和授权参数。客户机和服务器之间传递的各种参

数利用一个简单的具有 5 个字段的格式封装在一个 UDP 包中，格式的简洁性和 UDP 协议的高效性使得服务器可以高效率地处理大量的请求。

RADIUS 有 8 种标准的事务类型：访问请求、访问接收、访问拒绝、记账请求、记账响应、访问询问、状态服务器和状态客户机。认证的过程为：服务器将 NAS 访问请求包解密，然后认证 NAS 源，将用户文件中的访问请求参数变为有效，最后服务器将返回 3 种认证响应中的一种：访问接收、访问拒绝或访问询问。访问询问是一种对附加认证信息(比如令牌的一次性口令)的要求。

在 RADIUS 协议中，授权不是一个独立的功能而是认证响应中的一部分。若 RADIUS 服务器批准了访问请求，其将用户文件中所有具体的连接属性返回给 NAS 客户机。此过程通常包括数据连接(比如 PPP 和 SLIP)和网络的规范(比如 TCP/IP 和 IPX)，还包括特定的授权参数信息。在 RADIUS 协议中，记账是一个独立的功能，并不是所有的客户机都能够执行此功能。如果 NAS 客户机配置成具有记账功能，在用户得到认证后，它将产生一个记账开始的包，用户断开连接时产生一个记账结束的包。记账开始包描述了 NAS 所传送服务的类型、使用的端口以及所服务的用户；记账结束包复制了开始包的信息并添加了会话信息，比如使用的时间、输入输出字节数以及断开连接的原因等。

RADIUS 是为远程访问认证所设计，而不适用于主机以及应用认证。RADIUS 仅仅提供了基本的记账功能和监视系统事件的功能。RADIUS 的连接参数是基于用户的而不是基于设备的，这也是 RADIUS 的另一主要局限性。当一个 RADIUS 服务器管理多种类型的 NAS 设备时，用户的管理复杂度大大增加。对于检查用户是否是组成员、通过日期和时间来限制访问以及在指定的日期终止用户的账户等功能，标准的 RADIUS 认证并不具备。为了提供这些功能，RADIUS 服务器必须同其他的认证服务相结合。

3. TACACS

TACACS 认证有 3 种类型的包：开始、继续和回复。客户机在认证的开始发出一个开始包，其描述了所进行的认证的类型。对于简单的认证类型比如 PAP，包中可能还包含用户 ID 和口令字。服务器通过一个回复包进行响应。如果需要，一些附加的信息可以通过客户机的继续包和服务器的回复包进行传递。执行的事务包括登录以及使用多种认证协议(比如 CHAP，PAP，PPP 等)改变口令字。同 RADIUS 一样，一次成功的 TACACS 认证返回连接配置的属性值对(Attribute-Value, AV)，其中可以包含授权参数信息也可以单独进行传送。

TACACS 的授权功能包含请求和响应 AV 对，其主要用于以下目的：

- 对某些操作、地址、服务和协议的允许和拒绝。
- 设置用户特权级别。
- 调用输入输出包过滤功能。
- 设置访问控制列表(Access Control List, ACL)。
- 分配特定的网络，地址。

这些功能可以作为认证事务的一部分返回，也可以作为特定的授权请求返回。

TACACS 的记账功能使用类似于授权功能的格式。记账功能包括开始、停止、继续以及监视。除了 RADIUS 所支持的标准记账功能 TACACS 还具有事件日志的功能，能够记录系统级别在访问权利及特权的改变，而且不仅仅将与其相关的所有事件记入日志，也将引起事件的原因记入日志。

虽然 TACACS 是一个多用途而且稳健的协议，很少的服务器使用它，在 NAS 中的使用更加少。另外，TACACS 的规模和性能也是一个问题，与基于单个 UDP 包的 RADIUS 的设计不同，TACACS 在 TCP 上利用多次询问来建立连接，这样明显会对性能造成影响。TACACS 服务器没有代理请求的能力，因此它们不能在同等级进行配置来支持多个域之间的认证。

4. DIAMETER

DIAMETER 是一个高度扩展的 AAA 框架，能够支持多种认证、授权或记账方案以及连接类型。

DIAMETER 协议可以分为两个截然不同的部分：基本协议和协议扩展。DIAMETER 基本协议定义了消息格式、传输错误报告以及所有的协议扩展所用的安全服务。DIAMETER 的协议扩展则是为执行特殊类型的认证、授权或记账事务(如 NAS, 移动 IP 等)所设计的模块。

DIAMETER 建立在 RADIUS 协议之上,但是对其进行了补充,从而克服了 RADIUS 本身所具有的局限性。尽管这两种协议并不共享一个公共数据单元,但两者有很多相似点以至于可以很容易地从 RADIUS 过渡到 DIAMETER。DIAMETER 同 RADIUS 一样使用 UDP 协议来传输,但是前者是在对等的配置下进行,而后者是客户机/服务器模式的配置。这样服务器就可以在本地进行初始化请求操作并处理传输错误。DIAMETER 使用可靠的传输扩展来减少重传次数,并增加了失败结点的检测,减少了结点拥塞。这些措施减少了一些潜在的不安定因素并大大增强了服务器的性能,尤其是在高密度的 NAS 以及分级的代理配置的情况下。另外的改进还包括对漫游的完全支持,跨域的认证,对扩展认证协议(extensible authentication protocol, EAP)的完全支持,自定义属性值对(AVP)以及命令,加强了重传攻击保护安全功能以及个人 AVPs 的机密性。

DIAMETER 的认证是由扩展协议来管理的。客户机(比如一个 NAS)向服务器发起一个认证请求,其中除了用户名、口令以及状态值还包括请求指令、会话 ID、客户机地址以及主机名。会话 ID 唯一地标识了此次连接并克服了 RADIUS 中的类似问题,后者在高密度的配置中会产生重复的连接标识符。这样保证每次连接同服务器都有自己惟一的会话。在整个连接期间会话都得以维护,因此所有的与本次连接相关的事务都使用相同的会话 ID。AVP 的状态用于跟踪多事务认证方案的状态,比如 CHAP 和 SecureID。服务器在验证用户的有效性以后返回一个应答包,其中可能包含失败的 AVP,或者是所提供服务的授权的 AVP(比如 PPP 参数, IP 参数以及路由参数等)。

DIAMETER 的授权可以同认证请求绑定在一起,也可以独立进行。授权也是由所使用的扩展协议所管理,使用同认证相同的命令。授权请求必须在已经存在的会话中执行,它们不能用于发起一次会话。

DIAMETER 增加了事件检测、定期报告、实时记录传输的功能,因此记账功能比 RADIUS 和 TACACS 都有了明显的增强。DIAMETER 的记账功能是由授权服务器来指导的,关于客户机如何产生记账记录的指示作为授权过程的一部分进行传送。而且,DIAMETER 的记账服务器均能够强迫一个客户机传送记账数据,这对于解决连接困难以及当一个记账服务器遭到破坏时获取记账数据来说都是极为有用的。

对高强度安全性的支持是 DIAMETER 基本协议的一个标准部分。对于一些应用,比如移动 IP 来说,要求敏感的连接信息在多个域之间进行传输。而端到端的安全对于这些应用来说是不够的,因为数据可能会在每个段的转结点获取。而 DIAMETER 的增强代理扩展协议能够解决这个问题,它将敏感数据在 S/MIME 对象中进行加密并将其封装在标准的 AVP 中。

RADIUS, TACACS 以及 DIAMETER 所提供的 AAA 服务解决方案是较好的集中式认证解决方案。只需要进行仔细的规划和并不复杂的配置,就能增强其安全性、减少管理时间,而且远程的访问也能够在这个简单的、集中式的、灵活的、可扩展的方案中得到统一。

12. 2 通信和网络安全

通信和网络的安全主要用于保证电信媒体及网络本身的可用性,并保证通过电信媒体及网络所传输信息的完整性和机密性。

在一个多层的通信体系中实施安全服务是非常复杂的。然而,大多数企业及组织不得通过与他们的贸易伙伴、提供商、消费者以及公网相连接以扩展他们业务范围。每个连接必须保证在某种程度上是安全的。随着电子商务的发展,大多数企业和组织开始在网上进行产品交易。安全提供商通过各种安全技术如防火墙、虚拟专用网等对网络的安全做出保障。然而,所有这些技术对信息安全管理者也提出了挑战,他们必须能够分析采用哪种产品能够适合长期的网络安全策略要求,而且必须清楚何种策略能够保证网络具有足够的健壮性、互操作性并且能够容易地对其进行扩展和升级。

计算机和通信技术正在迅速地发展,同时各种设备也变得越来越小而且功能更加强大。这就使得用户能够更加方便灵活地使用各种技术设施,这种情况在无线领域表现得尤为突出。但是无线技术由于其本身的特性也具有相应的脆弱性,本文将对互联网的无线环境的安全性进行讨论。

目前很多研究成果与技术的成熟,意味着无线互联网的用户将会超过有线互联网的用户,这种假设基于目前全球范围已经有数亿的移动电话用户而且每天都以数千的人数在增长。如果每个移动用户都选择通过移动电话访问互联网,则无线互联网的使用人数将大大超过有线互联网的使用人数。正是因为这个潜在的巨大的市场,使得很多组织不惜巨资进行投资,以期能够在这个不断增长的市场中占领一定的份额。

但是无线互联网目前仍处于起步阶段,绝大多数移动用户还没有使用移动电话接入互联网。而由于无线设备本身所具有的局限性,目前无线设备还不能替代台式机和笔记本电脑。

根据无线互联网用户的反映，无线互联网主要具有以下局限性：

- 接入互联网时速度太慢。
- 当用户在移动中时，连接很可能在会话期间中断。
- 仅仅使用数字键打字对通信来说是一种障碍。
- 使用无线互联网价格过于昂贵。
- 无线设备具有很小的图形图像显示功能。
- 无线设备的屏幕过小，造成了用户的使用不方便。
- 当浏览网站时经常会出现错误。

众所周知，模拟移动电话的通信很容易进行拦截，几乎从模拟移动电话的出现开始就存在这个安全问题，通过使用特殊的无线电扫描设备可以轻易地对信号进行拦截。由于这个原因以及其他的原因，很多移动电话的提供商转人提供数字服务。对于数字移动电话传输来说，对信息进行拦截比较困难，而新的无线互联网服务就基于这种数字传输。

目前用于数字移动电话传输的方法有多种，主要的方法有 TDMA(time division multiple access)、GSM(global system for mobile communication)、CDMA(code division multiple access)这些方法所使用的无线电频谱不同，因此用户所分配使用的频率也有所不同。移动电话用户如果想使用无线互联网，通常不会考虑选择某种特殊的方式，而往往只是选择他们所喜欢的服务提供商。因此对用户来说提供商使用的传输方法是透明的，而对于服务提供商来说就完全不同，因为采用的方法在很大意义上依赖于他们的基础设施，比如，他们所使用的无线电设备、基站的位置和数量、处理业务的数量以及用户所使用的数字移动电话的类型等等。

由于各种原因，互联网的安全性很难在无线电话和 PDAs(Personal Digital Assistants)上实施，主要原因是这些设备的 CPU、内存、带宽以及存储能力是有限的，因此具有有限的计算能力。这些设备所具有的能力只是通常的计算机设备中的一部分，因此从互联网安全的角度看，它们几乎不具备任何安全特征。然而，这些设备正在一些公司的内部网中使用，尽管无线设备从各个角度看都比较小，但其安全性却同样重要。对于任何 IT 公司和信息安全部门，在构建内部网的时候，若忽略了无线设备的安全性则是一种重大失误。对于无线设备，必须同网络中的其他结点一样，都要对其安全性进行细致的考虑。

下面主要对无线设备的相关安全特征进行讨论。

1. 认证性

对无线电话用户的认证是数字移动电话最重要的安全特征之一。由于服务提供商所关心的是哪些用户在使用他们的服务，因此对移动用户的认证是极为重要的。

对于 GSM 电话，通常使用 SIM 卡或芯片来存储用户的认证信息。SIM 卡通常可存放认证和加密密钥、认证算法、标识信息、用户电话号码等信息。他们可以使用户向所使用的电话网络认证自己。在北美 TDMA 和 CDMA 使用与 GSM 的认证机制相类似的方法，使用了密钥、认证中心和请求响应技术。然而，由于 TDMA 和 CDMA 电话并没有使用可替换的 SIM 卡或芯片，而是依赖于嵌入在电话中的认证信息，因此用户的身份是同单一的电话相联系在一起的。在认证性方面，TDMA 和 CDMA 电话与 GSM 电话相比所具有的一个明显的缺点是缺乏一定的灵活性。如果要更换身份认证信息，GSM 电话所需要做的仅仅是更换 SIM 卡或芯片，而对于 TDMA 和 CDMA 电话来说，使用新的认证信息往往需要用户买一个新的电话。

这种形式的认证仅仅是服务提供者网络对用户进行了认证，这只是互联网业务中传输的一部分。为了保证互联网端到端业务的安全性，用户还需要对他们所要连接的互联网 Web 服务器进行认证，同时，互联网 Web 服务器需要认证用户的合法性。目前一些方法在应用层提供了端到端的认证，大多数的应用采用了 ID 号和口令的方式，这种方式有其局限性，因为其仅仅提供了单方的认证。有的组织正试验在 GSM 的 SIM 卡中添加额外的安全部件，比如密钥对、数字证书、以及其他的一些 PKI 组件等等。但是由于移动电话以及手持设备本身的局限性，使用这些安全组件可能是一种沉重的负担。为了适应无线设备中的处理器，可以根据无线设备中的可用资源，将数字证书和相关的密钥的规模减小。而有的组织正在试验使用椭圆曲线密码(ECC)进行认证,ECC 对于移动设备来说是一种理想的工具，因为其能够提供高强度的安全能力，而使用的资源却远远少于其他公开密钥算法所需要的资源。

2. 机密性

在无线设备上实现机密性主要会带来几个挑战。最典型的，当我们使用浏览器访问 Web 站点并输入口令字来获得权限的时候，通常口令字使用“*”或其他字符进行掩盖以防止其他人偷看。对于移动电话或手持设备来说，对口令字的掩盖会对键入字母带来一些麻烦。移动电话中的字母通常是通过数字键来代表的，而数字键可能代表 3-4 个字母，键入一个字母可能需要对同一个数字键按键 1-4 次。如果结果被掩盖，

用户可能会不知道实际所提交的字母。为了解决这个问题，一些无线互联网应用不对口令字进行掩盖而直接显示，而有的应用在用用户输入字母的过程中直接显示几秒钟，然后再用其他字符掩盖，这两种方法中后者具有比较好的灵活性。

第二个挑战是必须确保机密信息(比如口令字或者信用卡号)在用用户使用完毕后从手持设备的内存中清除。这些敏感信息很可能被无线互联网应用作为变量存放在设备的内存中，这样就可能被他人所利用。应用程序设计者必须认识到，在应用执行完毕后，对存储敏感信息的设备内存进行清空的重要性。

第三个挑战是必须确保敏感信息在无线设备与互联网的终点之间传输的过程中是保密的。对于有线互联网来说，很多站点使用 SSL(secure sockets layer)以及后来的 TLS(transport layer security)来加密从客户端到服务器的端到端的整条路径。然而，大多数无线设备，尤其是移动电话，缺乏足够的运算能力和带宽来有效地运行 SSL。作为替代，无线互联网应用开发使用了无线应用协议(wireless application protocol, WAP)。安全 WAP 应用使用 SSL 和 WTLS(wireless transport layer security)来保护安全传输的不同部分。通常，SSL 用来保护应用中的有线连接部分，而 WTLS 主要用来保护无线连接部分。WTLS 在操作上类似于 SSL，然而 WTLS 对 RSA 和 ECC 都提供支持。不同于 SSL 的是 WTLS 可以在速度慢、资源少的环境下提供安全服务，而 SSL 则只能加重环境的负担。这是因为 SSL 加密需要一个可靠的传输协议，比如 TCP(transmission control protocol)。TCP 提供错误检测、通信确认、重传等功能以确保可靠的网络连接，由于这些特征，TCP 需要高的带宽和资源，远远超过了无线连接和设备所能提供的范围。

针对这些局限性，WAP 讨论组设计了一个协议栈来提供对无线环境的支持。由于无线环境有连接速度低、可靠性低、带宽小的特点，协议栈使用压缩的二进制数据会话。WAP 协议栈位于 OSI 参考模型的第 4, 5, 6 和 7 层，对基于 IP 的网络，应用 UDP(user datagram protocol)协议，而对于非 IP 网络，应用 WDP(wireless datagram protocol)协议。WTLS 是 WAP 协议栈中的安全协议，可用在无线环境中保护 UDP 和 WDP 业务。

3. 恶意代码以及病毒

与针对于工作站和服务器的攻击相比较，针对于无线设备的安全攻击比较少。因为大多数移动设备，尤其是移动电话，缺乏足够的处理能力以及存储空间来供恶意代码以及病毒使用。然而，移动设备仍然容易受到攻击，而且随着移动设备的计算能力、内存以及存储单元的增长，这种脆弱性更加明显。目前，移动电话生产商声称，下一代的移动电话将提供对一些语言比如 Java 的支持，这样用户就可以下载一些软件程序到支持 Web 的电话中，同时也就为用户在无意中下载恶意程序代码提供了更多的机会。

尽管移动设备比较小，一旦它们访问到了组织中的敏感信息，带来的危害可以同任何一台计算机相比。如果忽略了无线设备以及它的能力，组织中的信息安全部门就相当于给攻击者留了一个忽视的没有保护的区域，而这正是攻击者所需要的。所有组织都应该在其信息安全策略中包含有对无线设备的保护策略，因为无线设备也是其基础设施中的一部分。

12.3 安全管理实施

安全管理实施包含几方面的内容：安全策略、指导方针可用来保证安全级别的合理性和一致性；对信息进行分类可用来保证敏感信息能得以有效的保护；而风险管理是对资源进行最有效利用的基本工具。

12.3.1 安全策略以及标准

安全策略可以认为是一系列政策的集合，用来规范对组织资源的管理、保护以及分配，以达到最终安全的目的。这种安全目的必须同组织的目标以及形势相一致，而且需要确定组织如何实施这种安全目的。组织目标同安全目的相结合构成了管理控制的基础，几乎在所有的业务实施中都需要使用管理控制来减小一些人为的欺骗以及失误所带来的风险。

1. 安全模型

安全策略是由管理部门做出的一系列决定，在某些情况下，安全策略的制定需要基于一些安全模型。安全模型定义了执行策略以及技术的方法，通常这些模型是经过时间证明为有效的数学模型。根据这些数学模型，可以制定一些安全政策。假设定义了一个模型，若该模型未经数学证明，则称为非正式安全模型，若模型经过了数学证明，就变为正式安全模型。下面主要介绍 3 种正式安全模型：Bell-LaPadula, Biba 以及 Clark-Wilson 模型。

Bell-LaPadula (BLP) 模型是基于机密性的访问模型，模型对安全状态进行了定义，并具有一个特殊的转换函数，能够将系统从一个安全状态切换到另一个安全状态。模型还定义了关于读写的基本访问模式以及主体如何对客体进行访问。安全状态是指根据一定的安全策略，仅有经过允许的访问模型是可用的。在这种状态下，定义了安全保留的概念，即如果系统在安全的状态下，新规则的应用将会把系统从一个安全状态转移到另一个安全状态。BLP 模型基于对主体和客体的分类级别来判断对客体的访问权限，通常有三种访问权限：只读、只写以及读写。

模型主要基于两种属性。一种是简单安全属性(simple security property)，它指出高保密性的客体(文件)不能被低保密性的主体(进程)读取，低保密性的客体可以被高保密性的主体来读取，这称为“不能从上读”，这样就保证了高保密级别的内容不被窃取。另一种属性称为星属性(star property)，它指出主体只能向相同级别以及更高级别的客体中写信息，这称为“不能向下写”。以这种方式，就可以防止主体从一个级别向一个更低的级别中复制信息，从而保证了高保密性的内容不会泄漏。

Biba 模型是基于完整性的访问模型的最初尝试。完整性模型通常与机密性模型相互冲突，因为要兼顾两者是很困难的。Biba 模型很少应用的一个主要原因是它与现实世界的安全策略没有直接的相关性。Biba 模型主要是建立在具有不同级别的完整性程度的单元之上，每个单元的元素是主动的主体(进程)的集合或者是被动的客体(文件)的集合。Biba 模型的主要目的就是为了解决完整性问题：防止未授权用户对信息的修改。

同 BLP 模型类似，Biba 模型也使用了 simple security property 和 star property，并增加了一种属性。simple security property 规定，低完整性的主体可以读取(访问)高完整性的客体，高完整性主体不可以读取(访问)低完整性的客体。star property 规定，低完整性的主体不能写(修改)高完整性的客体，高完整性主体可以写(修改)低完整性的客体。另外，第三种属性是请求属性(invocation property)，其规定低完整性的主体不能向高完整性的客体发送消息(比如对服务的逻辑请求)。

Clark-Wilson 模型也是基于完整性的访问模型，但同 Biba 模型不同的是 Clark-Wilson 模型主要有 3 个完整性目标：

- 阻止未授权的用户修改信息。
- 维护内部和外部的一致性。内部的一致性是指程序每次都确切地按照所期望的状态运行，而外部的一致性是指程序数据同真实世界的的数据是一致的。
- 阻止授权的用户对信息进行不适当的修改。

Clark-Wilson 模型依赖于良好的处理，这些处理都进行充分的构造和约束，从而能够保持内部和外部一致性的需求。

2.安全策略的必要性

安全策略以及安全技术的设计主要是为了防止个人发生错误行为并确保管理控制能够保持一种良好的状态。

策略对于一个组织中的成员了解他们所能执行的行为是至关重要的，而且一些法律法规上的需求也要求制定相应的安全政策。除此之外，从业务的角度出发也要求建立合理的策略。一个公认的事实是，保护组织的关键信息，如同保护组织的金融财产一样都是至关重要的。这就意味着需要对员工、提供商、顾客以及其他的授权网络用户进行有效的控制。随着网络技术和普及，人们可以从任何位置对信息进行访问，因此在整个组织范围内建立信息安全策略、程序以及标准是非常必要的。

随着计算环境从基于网络的系统逐渐变为基于客户机/服务器的系统，需要保护环境的复杂程度也急剧上升。如果不能正确地实施好的策略和程序，就有可能导致经济损失并可能导致市场信誉的损坏。而且在制定策略时，必须对公司的任务、价值以及业务的运作有一个实质的理解。

随着组织业务的发展，有必要对策略进行重新审查以确保其仍能够符合组织的需要。而且，对一个组织来说，可以与其他组织之间建立联系并相互交换有用信息。持续发展对于任何组织来说都是主要目标，因此对一些具有良好的实施情况的企业进行审查和借鉴对于企业的发展来说是非常重要的。一个组织可能以某种方式来实施特定的策略，而另一组织可能以另一种完全不同的方式实施，通过共享信息，使安全组织可以改善他们的实施方法并与组织的利益保持一致性。

3.安全策略的制定过程

安全策略的制定可以参照一个通用的步骤来完成，在制定安全策略的过程中，可能需要对所涉及的人员和时间进行合理的调度。通常安全策略的制定过程可以分为以下 6 个阶段。

(1)初始与评估阶段。最初某一部门向管理部门提交请求，指出制定一些特定策略的目的以及必要性；管理部门将会对这些请求进行评估，判断这些策略是否符合组织的利益并能够在期望的花费之内为组织带来效益。如果答案是肯定的，则将成立一个专门的小组对所需制定的安全策略进行研究并着手制定，否则不会采取任何行动。

(2)制定阶段。在此阶段的开始，首先要筹集一定的资金并组织一部分人员组成一个小组，这个小组必须与管理部门密切配合并且要确定谁对最终的安全策略负责。此小组是整个策略制定过程中的核心，相关的技术需求必须向小组中的专家提交，由他们进行讨论并考虑各方面的因素，从而形成文字。经过一系列的讨论和修改之后，基本形成了安全策略的初稿，然后需要组织中的一般人员对其进行检查和评议。检查

和评议的时间需要足够长，从而保证小组以外的人员能够参与。在检查的过程中，安全策略需要在模拟的试验中进行测试。当各种评论经过了小组的深思熟虑并认为整个文件在技术上是完整的之后，则可进入下一阶段。

(3) 核准阶段。当制定小组完成了设计及制定的任务后，文件将提交给组织中相关的实体，后者将对制定的安全策略进行详细的讨论。审核实体将对文件所制定的各个细节进行投票，最终的文件必须一致通过，若有任何异议，必须继续进行讨论并修改。

(4) 发布阶段。经过了核准以后，文件就可以在组织内发布。从现在开始，文件就可以在适宜的时候进行实施，在某些情况下，文件在发布后可能会立即实施。

(5) 执行阶段。在策略的执行阶段，与其相关的部门和人员开始实施新的策略。对于不同的部门和人员来说，策略的实施所产生的反应是不同的。比如说，用户的期望可能同管理部门的期望是不同的。因此在文件执行的过程中，鼓励各位相关人员提出任何问题和评论，这些评论在审查和维护阶段将会是非常重要的。

(6) 维护阶段。根据策略的规定，在文档的审查日期需要对文件进行审查。在审查的过程中，必须对文档的可行性进行审查并做出决定。如果文件不再需要，应立即将其进行收回或撤销；如果文件有继续的可行性但需要进行修改，则整个小组又开始转入第二阶段，然后周而复始。

上面主要论述了策略对于信息安全的重要性以及在策略的制定和实施过程中的相关问题。信息安全策略主要是根据管理者的需要来对本组织的智力财富以及信息财产进行保护。安全策略对于任何组织来说都是至关重要的一部分，因为它明确了组织中的成员如何对自己的行为进行引导。

12. 3. 2 风险管理与分析

在目前不断扩展的工作环境中，风险管理与分析很容易被认为仅仅是一种时髦的趋势，而且其占用了大量的时间。然而，对一个公司来说，如果以风险管理与分析作为基础来制定对策以及规划策略，它将会为公司提供很大的利益并能够节省大量费用。很多公司已经认识到了风险管理的重要性而且设立了首席风险长官的职务，并认识到风险管理应该是公司中很多部门的一个重要功能。通过将各个部门的工作以及成果相结合，就会对整个公司的情况有一个比较清楚的了解。有的部门将风险管理作为他们的职责之一，这样的部门包括安全、审计以及紧急策略规划部门。所有这些部门都需要进行风险分析，因此各个部门之间的联合就尤为重要，这种联合包括信息的共享以及对突发事件的反应。

风险管理是一种以公司利益为目的，以某种可能的最好的方式来对特定风险进行处理的能力。

而风险常常可以描述为一个数学公式：风险=威胁×脆弱性×资产价值

在公式中，

- **风险**是指对公司的业务产生影响并且阻碍公司目标实施的任何事件。
- **威胁**是指公司业务遭受到某一事件的影响的可能性。有的专家不仅从消极意义上对其进行描述，而且从积极意义上对其进行描述，也就是说，并不是所有的威胁都会带来消极的影响，但是人们往往都是如此认为。
- **脆弱性**是指威胁可以利用的弱点。它是指一个系统的软肋，不管系统的其他部分如何坚固，攻击者都可以利用其进行攻击并通过其打开整个系统。
- **资产**是指可以被风险所影响的系统组件。这是对风险分析的典型的定量解释，定量风险分析试图从纯数学的观点来描述风险，给每个风险一个估计值并以此作为进一步进行风险管理的指导方针。

1. 风险分析

风险分析主要包含三个重要因素：知识、观察力以及敏锐性。有效的风险分析依赖于对公司所运作的环境的彻底而真正地理解。风险管理者必须理解公司所面对的可能的威胁和其脆弱性，这些管理者必须对新的威胁、倾向、系统的组件、工具和体系结构有清楚的了解，以期能够认识到公司的脆弱性并找到解决方案。这些知识必须通过对安全公告、贸易流水账以及审计日志等的不断审查来获得。因此，首席风险长官必须在高层管理队伍中，这样他就具备了关于公司的战略方向的知识。首席风险长官还必须对所有正在发生的有可能对公司产生影响的事件做到清晰的把握。

观察能力是第2个关键因素。我们生活在一个数据和通信急剧膨胀的年代，观察是一种透过外部影响而理解内在本质的一种能力和技巧。观察意味着要对各种工具和报告进行定期检查以便注意到是否有非正常的事件发生。值得注意的是，很多有用的审计日志和高效工具所输出的报告都被束之高阁，因为对大多数人来说，从其中挑选出有用的细节信息是困难的而且很费时间。安装了入侵检测系统之后并不意味着观察的开始，这仅仅说明已经具备进行观察的能力。因此观察能力和有效工具的使用是对运作环境的特征和风险进行充分理解的关键因素。

对风险进行分析是为了获取到分析的结果，因此第三个重要因素便是业务敏锐性。业务敏锐性是指在业务领域进行有效运作的的能力，即对于所使用的方法和技术的充分理解以便能够获取到期望的结果。业务敏锐性将一般的管理者和有效的管理者区分开来，具有业务敏锐性的管理者知道如何处理事情、如何写出有力而可信的陈述，知道何时应该进攻以及何时应该撤退。因为风险分析的整体基础就是建立在对业务任务的理解和定位之上的，风险管理者在对风险和对策进行评估时，必须有能力抛弃传统的偏见。一个理想的风险管理解决方案需要用户、业务领域管理员以及有效的管理的支持，这意味着解决方案对于用户来说负担不能过于沉重，也不要对相关的支撑业务进程和系统在性能和生产力上产生很大的影响。

2.风险管理

风险管理是在业务进程和系统中加入一种平衡控制来防止、检测和矫正潜在的事件，前提是风险管理方案没有对业务的正常流通产生任何阻碍或限制。一旦风险评估结束，得到的结果将是对组织的所有可能威胁的精确的概括。在结果报告中包含所有被识别的威胁、威胁潜在的危害、威胁所能造成的损失和破坏以及在每个业务部门的主要角色的列表。根据评估，风险管理者必须判断针对于被识别的风险能否使用某种对策。通常，这些对策可以划分为三大类：降低、转移和接受。

通常是使用一些新的控制手段降低风险，这些控制手段可以是管理方面的(如 ID 控制、物理访问规则等)，也可以是技术方面的(如入侵检测系统、防火墙、新的工具等)。通过对风险的真实程度以及业务需要的分析评估，风险管理员将会提出一些针对于风险的可能方案。然后就对这些方案从费用、有效性、用户接受程度等各方面进行评估。

风险管理者必须制定一些纲要，对新的控制方案的重要性进行阐述，并促进用户对这些风险方案的理解，使用户认识到在他们的部门和公司中这些风险方案所起到的关键作用。

风险转移通常是指将风险推迟或者将其转移给其他的公司，通常可以通过一些保险或服务层次的协议来进行。保险业者往往需要对所担保的公司面临的风险进行彻底的检查，以确保所有的风险都得到认可而且对其实施了好的策略。这样的保险业务通常有拒绝服务保险、电子商务中断保险以及网站毁坏保险等。

当认为风险是无关紧要或者通过各种措施将风险降低到可以接受的水平时，对残余风险进行接受是必要的。为了接受一定水平的风险，风险管理员必须得到对风险进行评估的风险分析进程的通知，一旦管理员得到这种通知，他们必须决定接受风险。接受风险的前提是风险被认为是达到可接受的水平，可能是因为造成的影响不重要，或者是由于所采取的对策的花费超出了风险本身将会造成的损失，也可能是目前没有可行的方法来有效地阻止风险。

风险分析和管理是一个正在不断发展而且令人激动的领域。对于一个公司来说，具有风险评估并能有效阻止意外事件发生的能力对于确保其稳定持续的发展是非常有价值的。而对于风险管理员来说，如果他们具有将他们的努力成果与业务的需求相结合并及时应用新的产品和服务的能力，就能成为出色的风险管理。

12. 4 应用和系统开发安全

应用和系统开发安全涉及到在系统和应用软件中采用的各种有效控制手段，以及在系统和应用软件开发中所使用的手段和步骤。

随着客户机/服务器模式的广泛应用以及基于互联网应用的不断扩展，用户识别和认证以及数据的访问控制分布在系统体系结构的多个层次。这种分散的安全模型同集中控制管理的模型有着巨大的不同，分散的系统安全体系结构需要一种全面的保护机制。下面将对应用和系统设计开发中 Web 应用以及 XML 的安全性进行讨论。

12.4.1 Web 应用安全

Web 应用并不是可区分的、限定的程序，而是包括很多的组件和服务器。通常 Web 应用包括 Web 服务器、应用服务器以及数据库服务器。Web 服务器为终端用户提供图形接口，应用服务器提供业务逻辑处理，而数据库服务器用于存储应用中关键的数据信息。Web 服务器能够提供几种不同的方式向应用服务器提交请求并返回用户一个修改的或新的 Web 页面，包括通用网关接口(common gateway interface, CGI), 微软的 ASP(active server page)以及 JSP(java server page)。在某些情况下，应用服务器还支持请求代理接口，比如通用对象请求代理体系结构(Common Object Request Broker Architecture, CORBA)以及 HOP 协议(Internet InterORB Protocol)。

缺乏 Web 应用的安全性可以使入侵者迅速而容易地进入企业的网络。虽然所有的 Web 应用都在相同的 Web 服务器上运行，并使用相同的应用服务器和数据库服务器，但 Web 应用之间是不同的，因为至少运行的 Web 应用中有一部分代码是自己编写的。而企业往往没有足够的时间和资源来合理地加强他们的服务器，并没有对他们的应用程序代码在互联网上使用前作彻底的检查。而且，很多程序员不知道如何开发一个安

全的程序，可能仅仅开发一些独立的程序，因而即使出现一些小的差错也不会酿成大祸。大多数情况下，人们往往为了尽快地拿出产品而没有时间考虑应用的安全性问题。

因此，很多 Web 应用的脆弱性往往体现在服务器、应用以及自己开发的代码上。而很多攻击往往能够穿过外围的防火墙的原因是：为了使 Web 应用能够正常工作，80 端口服务必须开放。Web 应用攻击包括对 Web 应用的拒绝服务攻击、修改 Web 网页内容、窃取公司或用户的敏感信息，比如信用卡号等。Web 应用攻击不同于典型的攻击方式，它们难于检测到，而且攻击可能来自任何在线的用户，甚至可能是得到认证的用户。目前，这种安全性往往被忽略的原因是，各个企业仍然通过防火墙以及入侵检测系统来保障其网络的安全，但它们并不能检测到 Web 的攻击。

Web 应用的攻击弱点主要在于以下几个方面：

- **已知的脆弱性和错误配置。**已知的脆弱性包括 Web 应用中使用的操作系统以及第三方应用中的所有的 BUG，例如微软的 IIS(Internet Information Server)就存在一些安全漏洞，微软也针对此问题发布了一些补丁 6 而应用中的不安全默认配置以及管理员的不安全配置也造成了 Web 应用的弱点。一个典型的例子是，将 Web 服务器配置为允许所有的用户都可以进入系统目录，这样就可能导致一些存放在 Web 服务器上的敏感信息比如口令、源代码以及用户信息的泄漏。另外还可能允许用户在 Web 服务器上执行程序，这样就很容易导致 Web 服务器受到攻击。

- **隐藏区域。**隐藏区域指的是 HTML 表格的区域，对于很多应用来说，这些区域通常用来存放系统口令以及商品价格等信息。然而，这些区域往往不能很好地隐藏，可以通过 Web 页上的查看源码看到，而很多 Web 应用允许恶意用户对 HTML 源中的这些区域进行修改，从而为其作案提供了机会。这种攻击往往会成功的原因是，很多应用对返回的 Web 页面并不进行确认检查，它们总认为返回的数据是安全的。

- **后门以及调试选项。**开发者为了方便进行应用故障检查往往会留下一些后门或者调试选项。在开发过程中这样做有其方便之处，但这些问题往往仍然会遗留在最终版本中，甚至会允许用户不需密码即可登录或者一个特殊的 URL 被允许直接进入应用到应用配置中。这种弱点的存在主要是由于在程序开发过程中没有一个正式策略或者没有按规定秩序执行。因此在 Web 应用中的一个关键的步骤就是将所有的后门删除并关闭调试选项，这样会大大减少应用的脆弱性，然而这一步骤往往被遗漏。

- **参数篡改。**通过 URL 中所包含的 SQL 调用可以访问到 Web 应用后端的数据库，恶意用户能够利用 SQL 调用来获取所有用户列表、口令、信用卡号以及数据库中存储的其他数据。

- **Cookie 攻击。**Cookie 攻击指的是修改存放在 Cookie 中的数据。Web 站点往往会在用户的系统中存放 Cookie，其中包括用户的 ID、口令以及账号等等。通过修改这些数据值，恶意用户可能获取其他人的账号访问权限。攻击者也可能直接盗窃其他人的 Cookie 从而获取其权限。大多数的商业 Web 应用，比如基于 Web 的电子邮件以及网上银行，使用 Cookie 来进行认证。如果攻击者获取到 Cookie 并将其导入到自己的浏览器中，就能够访问该用户的账号而不需要提供 ID 和口令以及任何其他认证形式。虽然，此账号仅仅在会话期内有效，但所造成的破坏已经发生了。

- **缓冲溢出。**缓冲溢出是一种典型的攻击技术，恶意攻击者通过向服务器发送大量数据从而使系统崩溃。通常系统中有一定量的缓冲区用于存放这样的数据，若接收的数据大于缓冲区的容量上限就会发生溢出到堆栈。如果数据是一些代码，则系统就会执行任何溢出到堆栈的代码。

- **直接访问浏览。**直接访问浏览指的是在经过认证后直接对 Web 页进行访问。任何 Web 应用的不恰当配置都会使恶意用户直接访问到一些包含有敏感信息的 URL，这样就会给公司造成损失。

Web 应用攻击能够对公司财产、资源和信誉造成重大的破坏。尽管 Web 应用增加了公司遭受攻击的风险，

但现有的一些方案能够尽量减小这种风险。

1. 预防

预防 Web 应用攻击的最好的方法是通过培训提高警惕性。开发者应该在安全编码方面进行培训，管理者应该进行风险培训，以确保系统在正式运行之前经过彻底的测试对于开发者来说，永远不要相信得到的数据而只能相信自己所控制的数据。因为他们不能控制终端用户，因此应该将所有输入的数据都看作具有潜在的威胁。不要假想任何发给用户的信息会没有改变地返回而 Web 表格中的数据都会按照要求填写。通过使用过滤和输入检查将会大大减少 Web 应用受到攻击的风险。开发者在应用开发设计的过程中也要考虑到安全性措施，虽然在开发过程中使用匿名 Web 服务器账号可以节约时间，但这可能会引起问题。在验证程序编码中很可能存在 BUG，而这种 BUG 往往在应用投入使用若干天以后才会发现。

如果可能的话，尽量不要使用超级用户账号来运行程序，虽然在处理访问权限时会节约时间，但同样会带来安全问题。如果在超级用户账号下运行所有的东西，则 Web 应用用户对所有数据库的表都有写的权

限。通过使用 SQL 语句修改 URL, 恶意的用户能够轻易地访问所有的数据库。安全性的一个原则是给予用户完成任务所需要的尽量低的权限。

使用 HTTP 的 GET 请求从客户机向服务器发送敏感信息会带来很多安全漏洞。GET 请求被 Web 服务器的日志以明文的方式所记录, 任何人都可以看到。因此通过 GET 请求向服务器发送的信用卡号将会在 Web 服务器的日志中以明文方式存放, 这就带来了安全隐患。而 SSL 也不能防止这个问题, 因为 SSL 仅仅加密传输中的数据, 而 GET 请求仍然会以明文方式存放。在客户机和 Web 服务器之间发送数据最好使用 HTTP 的 POST 命令, POST 命令使用 HTTP 实体来传送数据, 而没有被 Web 服务器所记录。虽然信息仍然以明文方式传送, 但可以使用 SSL 协议进行加密传输。

开发者应该时刻意识到 HTML 命令以及错误消息可能泄漏信息。虽然这不会直接导致一次攻击, 但是攻击者可能会得到关于应用体系结构的足够信息, 从而可能发起成功的攻击。一些错误消息可能会提供 Web 服务器的物理路径, 也可能提供一些所使用的应用服务器或者特定数据库的信息, 这些信息可以被攻击者利用来增加对体系结构的了解从而发起有效进攻。

2. 一些可用的技术工具

编码的安全可能会增加 Web 应用的安全性, 但这样是不够的, 而使用一些工具则可以增强 Web 应用的审计功能以及安全性。如果 Web 应用使用 CGI 脚本, 则可以使用 RFP 的 `whisker.pl` 脚本, 这个 perl 脚本可以监测一个站点的一已知的 CGI 的弱点。对源代码的检查也很关键, 一些工具能够对源代码进行检查, 比如 NuMega(www.numega.com), L0pht(www.l0pht.com/slnt.htm) 以及 Lclint(lclint.cs.Virginia.edu)。有些产品专门致力于 Web 应用的安全, 而且数量也在不断地增加。比如 Sanctum 公司的 AppShield 产品(www.sanctuminc.com)能够保护 Web 站点免受上面所讨论的各种攻击。AppShield 对于 Web 应用来说就像是防火墙, 仅允许经过批准的数据和请求通过。

利用 Web 应用的漏洞已经迅速成为攻击者获取敏感信息以及对服务器进行访问的手段。而且目前有很多工具能够帮助攻击者, 甚至在某些情况下允许攻击者获取对系统的完全控制。而大多数系统漏洞的存在主要是因为应用开发者对程序的安全性重视不够, 或者是没有足够的安全意识。人们往往为了尽快将产品投入市场, 从而缺乏足够的时间对产品进行充分的安全测试。为了保护应用软件免受攻击, 对软件开发者的培训是关键所在, 同时, 可以利用一些商业工具来发现产品的脆弱点, 以防止它们被攻击者所利用。总之, 对 Web 应用的攻击仍然在不断增加, 而保护 Web 应用中数据和资源的关键就是提高安全意识和警惕性。

12.4. 2 XML 的安全性

XML 是互联网联合组织创建的一组规范, 以便于软件开发人员在网页上组织信息, 其目的不仅在于满足不断增长的网络应用需求, 同时还希望借此能够确保在通过网络进行交互合作时, 具有良好的可靠性和交互操作性。从本质上来说, XML 是一种简单的基于文本的语言, 能够描述复杂的数据结构。因为 XML 的简单性, 几乎所有的计算机都有使用 XML 的能力, 而所有类型的网络都能够对其进行传输, 而且计算机系统使用 XML 时并不需要对已存在的基础设施做很大的改动。

为了对 XML 及其安全问题有一个更好的理解, 首先应该了解 XML 的背景。超文本标记语言(HyperText Markup Language, HTML)是万维网的一个基础, HTML 因其简单而且易于使用成为最受欢迎的语言之一, 即使非编程人员也能够初步掌握 HTML。

但是随着网络以及 Web 应用的快速发展, HTML 不再能够满足人们的需要, 其局限性主要体现在:

- HTML 不具有可扩展性, 不能够为一些特殊的需要定义标记。
- HTML 仅仅描述了文档的表面现象, 而没有描述内容, 因此在 Web 上查找某些特定的内容是比较困难的。

鉴于 HTML 的局限性, 在 1996 年互联网联合组织(World Wide Web Consortium, W3C)找到了一种解决方法, 即标准通用标记语言(Standard Generalized Markup Language, SGML) 而 HTML 只是 SGML 的一个简单应用。SGML 成为一种由广大的软件提供商所支持的通用标准, 其描述了数据本身, 而不是仅仅描述了表述的方法。SGML 还支持一些结构化的环境, 任何 SGML 文档可以是其他文档的容器, 可以有复杂的嵌套, 并允许通过简单手的文档来构造复杂的文档。SGML 的主要问题是过于通用, 以至于对于 Web 浏览器来说处理起来比较复杂, 仅仅描述它的规范就多达 500 页。在这种情况下就产生了一 XML, XML 是 SGML 的一个子集, 它是一种新型的元语言(meta language), 能够允许用户构建自己的标记语言。XML 的规范限制在 50 页以内, 远远小于 SGML 的 500 页, 然而 XML 包含有足够的规则, 任何人都可以从零开始创建一种标记语言。

1. XML 的优点

由于 XML 具有其突出的优越性, 越来越多的企业都开始转人使用 XML, 而 XML 所提供的好处大多是

HTML 所不具有的。其中包括:

- 简单化。XML 对于人和计算机来说都具有易读性和易懂性, 利于计算机的处理, 而且能够用来描述复杂的数据结构。与其他一些分布式的软件技术(比如 CORBA, DCOM)相比, XML 更加容易学习, 从而节约了开发时间。

- 开放式标准。XML 是开放的, 是 W3C 标准, 世界上几乎所有的软件开发都对 XML 表示认可, 尽管一些大公司(比如微软、Oracle 以及 IBM)在某些事情上很难达成协议, 但他们在其软件产品中全都承认 XML 的开放式标准。

- 对数据的描述。XML 对信息提供元数据或描述性数据更加容易, 这样就有利于进行数据挖掘以及构建有效的搜索引擎。

- 编排的便利。XML 的一个最大的优点在于其将内容和设计相分离。XML 提供了一种方案, 使得在修改文档的外观时对文件的内容没有影响, 同样, 在修改文档内容时对文件的外观设计没有影响。

XML 所包含的规则使用户可以方便地从零开始创建标记语言。因此, 在创建 XML 文件时, 可以为自己的元素指定自己所喜欢的任意的名称, 以这种方式 XML 可以用来描述所有类型的文件。

一个基本的 XML 文件主要包括:

- 文件元素。每个文件必须包含惟一的最高等级元素, 即文件元素或称为根元素。
- 元素嵌套。如果一个元素在另一个元素内部开始, 其必须在此元素内部结束。
- 开始和结束标签。每个元素必须都有一个开始标签和结束标签, 而且开始和结束标签中的元素名称必须完全吻合。

为了增强 XML 文件结构化要求, 必须利用 XML 的辅助技术—文件类型定义(Document Type Definition, DTD)。DTD 是定义 XML 文档格式最古老最简单的方法, 也是由 W3C 定义的。DTD 能实现指定可行标签、元素、属性、每一元素的应用次数以及元素在指定 XML 文件中的排序等功能。DTD 可以提供数据类型强化的限定类型, 从另一种意义上来说, 也就是可以自行确定一个元素是否含有其他元素、其他数据或者为空。

XML 主要有三个要素: 模式(Schema 可扩展样式语言(eXtensible Stylesheet Language, XSL)和 XLL(eXtensible Link Language, 可扩展链接语言)。Schema 规定了 XML 文件的逻辑结构, 定义了 XML 文件中的元素、元素的属性以及元素和元素的属性之间的关系, 它可以帮助 XML 的分析程序校验 XML 文件标记的合法性; XSL 是用于规定 XML 文件样式的语言, 它能在客户端使 Web 浏览器改变文件的表示法, 从而不需要再与服务器进行交互通信; XML 将进一步扩展目前 Web 上已有的简单链接。良好的数据存储格式、可扩展性、高度结构化、便于网络传输是 XML 主要的特点, 决定了其卓越的性能表现。由于 XML 能针对特定的应用定义自己的标记语言, 使得 XML 可以在电子商务、政府文档、报表、司法、出版、联合、CAD/CAM, 保险机构、厂商提供各具特色的独立解决方案。

2. XML 的安全问题

信息安全领域的工作者都倾向于成熟的产品, 因为成熟产品所带来的风险较小, 从而使工作环境能够得到稳定控制。虽然 XML 还不是足够成熟而且是一种新的标准, 但它不仅对互联网产生了影响, 而且在其他很多业务以及数据库应用等方面都产生了影响。

XML 很快就会成为互联网上各种业务应用的一种世界性语言, 在其成熟之后, XML 能够提供简单而且无缝的购买业务、银行业务以及其他功能。但是互联网是不安全的, 而 XML 也没有在安全性上对其有所增强, 实质上, XML 的主要目的是使在互联网上传输的数据更加容易理解和阅读。几乎所有的提供商, 包括 W3C 在内, 都看到了互联网的不安全性对 XML 的使用所带来的问题。这些问题从本质上归结为两个众所周知的安全问题: 机密性和认证性。这两个问题通过 XML 加密以及 XML 签名来解决, XML 加密可以提供重要数据的机密性, 而 XML 签名可以提供可靠性、完整性和不可否认性。目前, 一些团体正积极投身于关于这些问题的标准开发的活动中。

可以将 XML 文件整篇加密, 然后安全地发送给一个或多个接收方。但是人们更感兴趣的是如何对同一文件的不同部分进行不同的处理。XML 的一个有价值的好处是可以将一整篇 XML 作为一个操作发送, 然后在本地保存, 从而减少了网络通信量。但是, 这就带来了一个问题: 如何控制对不同元素组的授权查看。在 2001 年 3 月, W3C 发布了 XML 加密的需求规范, 根据这个规范, W3C 工作组的任务是开发一个用于数字内容(包括 XML 文件及其部分元素)加解密的进程以及一种 XML 文法, 用于描述加密的内容以及使接收者能够进行解密的信息。

而 XML 的签名需求是由 XML 密钥管理规范(XML Key Management Specification, XKMS)同时提出的。XKMS 是由一些包括微软 VeriSign, IBM 等大的软件提供商在 2001 年提交的。目前, 对 XML 文件整体进行数字化签名不是问题。然而, 当需要对文档的不同部分(可能由不同的人)进行签名, 以及需要与选择的方

法一起这样做时，就会出现困难砂

随着一些新技术的引入，它们与 XML 的集成将可能会导致安全漏洞的出现，而最大的安全威胁则来自于对一 XML 的模式(Schema)，DTD 以及 XSL 的样式(Stylesheet)进行有意或无意的修改。一个很小的修改就可能导致 DTD 中的致命错误，从而可能大范围地终止 XML 的运行已而对这种情况的攻击并不复杂，攻击者只需要将一个可选属性改为必选属性即可。如果一个 XML 文件头中包含一个 URL，同网络中的另一个 DTD 建立路径与则客户机必须能够访问 DTD 来对 XML 对象进行评估。若 DTD 主机服务器在防火墙的后面，那么一旦客户机和服务器之间建立了通信连接，防火墙就可能被攻破。当上述的问题得到解决以后，新的问题很可能出现，而且 XML 的开放特性使得这些不是很复杂的攻击会继续存在，尤其对于一些没有采取足够措施来保护其数据的公司更是如此。

安全专家、程序员、行政人员还必须认识到一个以前从未考虑过的趋势，即 XML 正在成为安全方案中的一部分。XML 将不仅仅用于为信息安全提供一个公共的文件框架，而且可用于将各种安全任务与应用和计算机系统集成在一起。随着这种趋势的发展，对于安全专家来说，增加对 XML 的基本原理以及在各种安全方案中 XML 的使用情况的理解尤为重要，因为 XML 很可能成为各种安全组件之间的关键纽带。

12.5 密码术与安全观念的发展

密码术是为保证信息的机密性、完整性等所采用的各种技术。密码术是一种秘密写作的艺术，为了保证信息的秘密性而将信息进行变形的这种能力，其历史大约与写作本身的历史一样悠久。在凯撒、的时代，秘密写作主要通过使用字母替换来实现。随着时间的推移，各种技术不断在发展，一些加密方法也变得脆弱而容易被攻破，因此也不断涌现更复杂的方法来保证信息的安全性。

当今世界各国信息安全都是以密码术为基础的，机密性通过序列密码、分组密码等加密技术来实现，完整性利用 HASH 算法、公钥密码等形成的密码协议来保证，可控性的实现是由各种密码体制的综合利用来完成。

密码学有它自身的发展历程，密码编码术是技术发展的产物，从技术看，它经历了手工密码、机械密码、机电密码、电子密码、计算机密码各个阶段，事实上现在流行的应该称为芯片密码。这些密码都是数学和技术结合的典范，数学是密码编码和密码分析的基本工具。

数学通常用假设作为建立某个体系的前提，在这个前提下数学论证的结果是正确的，前提变了或假设不能被完全满足时所建立的体系就会垮掉。

值得指出的是，当今密码体制是建立在三个基本假设的基础上的。

- **随机性假设：**在单一地域内产生均匀分布的随机比特序列是可能的。密码学保护信息的基础是随机性，一个好的密码设备、安全协议必须构成两类不可或缺的乱源。一类是物理乱源，一类是数学乱源。数学乱源是基于以下计算假设。

- **计算假设：**合理的计算时间有一个量的界限，在一个合理的计算时间内，单向函数是存在的，它正向计算容易求逆难。密码算法是在计算假设下设计出来的伪随机函数。它具备密钥参量，知道密钥容易计算，不知密钥求逆困难。密码算法的研究，主要是序列密码、分组密码和公钥密码。序列密码比较成熟，分组密码和公钥密码都密切依赖于器件和数学难题，理论上都是可破的，在计算假设下，它们具备实际安全性。

- **物理假设：**对单一地域的信息实行物理保护是可能的，物理地保护长距离传送中的信息是非常困难的。信息系统的安全模型中总是设计一块物理保护区，对系统的关键数据实行严格的物理保护，这是信息系统安全性保证的物理前提。例如，安全路由器、安全服务器等信息安全设备都设计了采取物理保护措施的部件。

显然，软件产品是不利于实行物理保护的。“系统建在硅片上”(system on a chip, SOC)是安全信息系统构造的一个重要技术措施，在物理假设的意义下，这也是一个理论措施，非如此就不可能安全。soc 是电子设计一个大潮流，对一般电子产品而言，这是个提高产品质量的问题，而对信息安全产品来说就是能不能真正安全的问题。应该说，现在是芯片密码年代。

密码学的发展密切依赖于技术，但它的基础是数学，当然，人们期待着物理学的应用。当今非数学加密手段的研究也受到广泛的重视。量子密码的研究希望突破随机性假设的单一地域前提，异地产生物理噪声，实现一次一密的理想密钥传递(QKD)，另外，量子传真研究为秘密传输开发崭新的途径。尽管如此，信息安全的核心技术是密码术这个论断预计在今后若干年内还不会有实质性的改变。

从文明的纵向结构看，密码与信息安全技术的发展有一个清晰的由数据和通信保护(DCS)、信息安全(IS)到知识安全(KS)的进程。在工具时代，原始人类的知识多数是共享的，拥有较多知识的人获得族人的尊敬，少有秘密；在农业时代，知识传播受到官府的控制，知识垄断成为统治者的治国秘方，教育的目的

是为统治者培养人才，不是培养知识劳动者，这个时代安全性的主要追求是数据和通信保护，而密码术主要用于实现机密性需求；在工业时代，科学研究独立，信息垄断成为治国工具，教育的目的是培养高素质劳动者，安全性的追求是信息安全，在这一阶段，密码术被用作实现可认证性的技术基础；在知识时代，知识网络化、人格化，对话式工作方式、伙伴关系和灵活性，知识与经济互动，专有信息的普及成为治国工具，财富体现为精神享受，知识安全成为安全性的主要考虑，从知识安全的观点出发，可控性被放在首要地位，密码术是实现可控性的不可或缺的一种手段；知识的概念比信息的概念要宽展。信息是事实和数据的某种集合，不像知识那样反映事实和数据之间的内在联系和规律；信息可以通过告知而获得，知识则需要学习和思考，甚至钻研方能掌握。和自然资源、社会资源一样，知识也是一种资源。信息和知识是相交的两个集合，信息化的知识和知识化的信息都是知识。在计算机意义上，知识和信息__可以相互转化。真实世界里的知识映射到数字世界是困难的，显然，不论在哪个文明时期，安全方面人类关注的对象归根到底是知识。技术的发展使知识映射到数字世界成为现实，当今的信息系统、庞大到(丰富到)人们可以在其中讨论(处理)知识问题了。

数字世界的知识安全问题研究。重大而迫切，知识安全研究内容至少包括以下 5 个方面。

- **知识的表达：**我们主张用 $0^{\sim}9$ 的数字序列表述知识，落实到汉字，如何更好地用纯形码来处理汉字的编码、键盘输入和存储的统一，进而解决基于汉字的知识处理问题都是亟待研究的课题。

- **知识的抽取：**如何把人的直观的判断和经验抽取出来；如何从浩大的信息海洋中提取、归纳出知识。

- **知识的传递：**包括信源编码、信道编码(纠错和减小通信量)、加密编码(利用冗余和随机性)，考虑如何三码合一。

- **知识的保存：**也就是存储技术的研究。知识可以发展到人们无法想象的程度，但是如果没有相应的技术来保存，那么随着时间的推移，什么知识也没有了。因此，知识的保存是一个非常重要的问题。

- **知识的保护：**采取新理论、新技术对映射到数字世界的知识进行分级管理、控制与保护。

12.6 安全体系结构和模型

计算机体系结构涉及到计算机组织和构造的各个方面，其可用于解决计算机的安全问题，计算机安全模型主要涉及到一些用于维护系统和信息安全的概念。书中在系统体系结构和设计部分，主要对 UNIX 系统的安全执行问题进行了讨论，并对数据库的完整性问题进行了讨论。

12.6.1 UNIX 系统的安全性

在任何的安全方案中，操作系统的安全性都是一个关键的因素。操作系统是运行在机器上的任何软件的基础。下面主要讨论了操作系统的安全服务及其相关问题。

操作系统通常可以提供以下的安全服务。

- **身份识别及认证。**一个安全的操作系统必须能够区分不同的用户(即具有身份识别的功能)，而且必须能够通过某些手段对用户进行认证。身份识别和认证对操作系统的其他安全服务来说也是非常重要的。典型的认证方式主要有 3 种：用户是谁，用户所知道的(比如口令字)以及用户所拥有的(比如智能卡)。口令字是最常用的认证手段，但是这种方法非常脆弱，容易被攻破。

- **访问控制。**操作系统通过使用一些主体、目标、访问权限以及访问确认等方法提供逻辑访问控制。主体包括每个用户的 ID、口令、是否组成员以及特权等，目标安全信息包括所有者、组、访问限制等。基本的访问权限包括读、写和执行。而操作系统通过相应的访问确认规则来对一次访问请求(包括主体、目标以及访问请求)进行评估。

- **可用性和完整性。**可用性和完整性服务包括：系统是否以安全模式启动，系统在遭受攻击时是否按照期望的方式进行响应，系统内部的数据是否是一致的，系统中的数据是否与其所代表的真实世界中的实体相对应等。

- **审计功能。**审计日志包含有按时间顺序排列的事件记录。审计日志是非常有用的防御措施，而在事件调查中所起的作用则更为重要。在计算机犯罪案件中，审计日志可以作为合法的证据。操作系统必须记录所有与安全相关的事件，并保护审计日志的机密性和完整性，以确保数据随时是可用的。

- **用户可用的安全设施。**无特权用户需要一些手段来保证对他们的文件以及改变口令的权利，而特权用户需要一些额外的设施，包括封锁账号、访问其他用户文件、改变用户的组成员属性等能力。

下面对于在 UNIX 系列操作系统中上述服务的实施情况进行详细的讨论。

1. 身份识别及认证

UNIX 通过用户名进行识别并通过口令字进行认证。作为安全策略，UNIX 并不以明文方式存储回令字，而是油过一种变形的 DES 算法对口令进行加密以密文方式存储。加密的口令字；以及其他些账号信息存放在/etc/passwd 文件中，所遵循的格式为：

Username: encrypted password: UserID: GroupID: user's full name:

home directory: login shell

不幸的是/etc/passwd 文件是所有人可读的,这样就存在一些危险,任何具有系统访问权限的人都可以对口令进行强力攻击。如果有足够的计算机资源和稳定可用的工具,攻击者最终可叙猜测系统中所有的口令字。鉴于系统的这种脆弱性,目前各种版本的 UNIX 系统均提供了对一种称为“影子口令”的支持。基本思想是在另外一个单独的文件(/etc/shadow)中存放加密的口令字,其仅仅可以被“root”账号所阅读。尽管口令有效期(password aging)并不是标准 UNIX 的一部分,但是很多 UNIX 的版本提供了对它的支持。

UserIDs (UIDs)通常是注 6 位整数,操作系统使用 UIDs 而不是用户名称来跟踪用户。因此完全有可能在一 U 南 X 系统中有两个或更多的用户有相同的 UIDs,通常两个用户共享同“个 In 是共种糟糕的做法,而且系统保留了一些特殊的 ID(如任何 ID 为 0 的用户认为是系统的 root 用户)。一些程序比如 /bin/passwd(用于修改口令)等需要在 root 下执行、用户不应该任意获得系统的 root 权限,因此 UNIX 通过使用 Set UserID(SUID)使得特定的程序能够在其他 UIDs 的权限中使用。当然,这种程序存在一定的危险性:如果攻击者能够攻破 SUID 程序,就可能获得 root 权限。

GroupIDs (GIDs)通常也是 16 位整数,在/etc/passwd 中所列的是用户的主 GID,在 UNIX 的一些版本中,一个用户可以属于几个组。文件/etc/passwd 中包含所有组的完整列表,包括姓名又以众以及成员。

一旦用户成功登录,系统使用/etc/passwd 中所指定的用户 shell,同用户主目录下的.profile 一同执行全局文件/etc/profile。如果这些文件的权限没有进行适当的限制,攻击者就可能修改这些文件,从而每次用户登录后都会导致执行未授权的指令。UNIX 的/usr/adm/lastlag 文件中保存了每个账户最后一次登录的日期和时间,而且系统不断对此文件进行更新扩这些信息均可以通过 figger 命令获得,这样就会给系统带来新的威胁:若系统允许 figger 命令,就会给攻击计划提供一些有用的信息。

2. 访问控制

标准的 UNIX 系统通过慎重的访问控制来阻止对系统资源(文件、内存以及设备)的未授权使用。基本权限分为 3 类:所有者、组成员和其他,而特权账户可以忽略这些访问控制。UNIX 系统一致对待所有的系统资源而不具体区分文件、内存以及设备,所有的资源为了访问控制的目的都作为文件进行处理。UNIX 系统的每一个文件(目录)都具有一个所有者、组以及权限集。当用户或者进程产生一个新文件时,文件被赋予默认的权限。对于进程产生的文件(例如由文本编辑器产生的文件),进程将指定默认的权限;对于用户产生的文件,根据用户 shell 程序的开始文件中所指定的权限进行赋值。文件所有者可以使用 chmod 命令对文件的权限进行修改。

3.可用性和完整性

可用性的一个表现是系统在失败后是否能够安全地重启。传统的 UNIX 系统能以单用户模式启动,通常以 root 身份登录。而不幸的是,单用户模式允许任何坐在系统控制台前的人执行特权命令,因此传统 UNIX 系统的单用户模式实际是系统的一个安全漏洞。根据 UNIX 系统的不同版本,安全管理员通常可有几种方法修补此漏洞。首先,在操作系统的支持下,安全管理员可以对系统重新配置,当以单用户模式启动时需要提供口令。其次,严密的物理控制可以用来阻止对系统控制台的物理访问。

系统重新启动也与系统的完整性有关。若系统非正常关闭,UNIX 系统的 fcsk 命令就会检查文件系统的完整性并进行修复(自动进行或通过管理员执行)。有很多方法可以用来维护 UNIX 系统的文件系统的完整性,有一种方法逐渐被广泛使用,即由 Gene Kim 和 Gene Spafford 开发的 TripWireo TripWire 可以通过对每一个监测的文件产生一个签名或者消息摘要来对文件系统提供完整性保护。TripWire 允许管理员指定哪些文件和目录需要检测、需要监测的对象属性以及使用哪种消息摘要算法(MD5 或 SHAT 等)。当执行 TripWire 时,将对修改、增加以及删除的文件进行报告,因此 TripWire 不仅能够检测到特洛伊木马,而且能够检测到违反组织政策的任何改动。

4. 审计日志

UNIX 的不同版本使用不同的目录来存放日志文件(比如/usr/adm, /var/adm 或/var/log),但不管目录在何位置,

通常 UNIX 系统在下列日志文件中记录并存放与安全相关的事件。

- lastlog:记录用户最后登录的时间。
- utmp :记录 who 命令所使用的账号信息。
- wtmp :记录用户每次登录退出的时间。
- acct: 记录所有的可执行命令,此信息可以使用 lastcomm 命令获取(缺点是没有方法可以对指定的

事件或用户进行记录，因此若使用此日志将会耗费大量的磁盘空间)。

- `slog:`, 记录所有的: `u` 命令, 并记录其是否成功执行。
- `messages;`: `%E` 所有发送到控制台的消息以及其他的 `syslog` 消息。

绝大多数的 UNIX 版本提供一种称为 `syslog` 的功能, 起初是为 `sendmail` 程序所设计的。`syslog` 来自所有合法程序的消息, 消息中包含程序名称、设备、优先级以及日志消息本身, 同时系统为「每条消息添加系统日期、时间以及名称。`syslog` 功能可以进行配置, 即管理员可以在 `/etc/syslog.conf` 中指定对什么进行记录和如何进行记录。`syslog` 可以识别多种安全状态以及级别, 包括 `emerg`(紧急情况)、`alert`(需要立即响应)、`crit`(关键条件)、`err`(一般错误)、`warning`(警告)、`notice` 注意)以及 `debug`(调试)。而且 `syslog` 允许消息存放在多个地方, 包括文件、设备以及其他的机器。这样就会使入侵者难于隐藏其入侵痕迹。

5.用户可用的安全设施 ‘

通常 UNIX 系统支持一个特权管理账号(即 `root` 账号)。使用 `root` 账号可以新建、修改、追加以及删除用户账号, 配置日志选项, 管理组成员, 添加和删除文件, 执行系统中的任何程序和关闭系统等等。简而言之, `root` 账号拥有所有可能的特权。无特权的用户可以使用 `passwd` 命令改变自己的口令, 而且能够使用 `chmod` 程序改变自己的文件以及目录的权限。

通常 UNIX 系统对其操作系统中的安全组件有不同程度的实施, 一些附加的安全工具能够对 UNIX 系统核心服务进行有效的补充。如果进行合理的配置, UNIX 系统完全能够抵抗任何入侵者以及攻击者。

12.6 . 2 数据库的完整性

数据库可以定义为是一些相关的或相互依赖的数据元素的整体集合, 或者是信息整体集合, 这些信息由一些编码的数据元素所表述, 在数据元素之间具有一些特定的关系。数据库通常可以在用户之间以及应用中进行共享。数据库中的部分信息可以直接通过编码的数据元素泊勾形式表现出来, 然而有些信息是通过数据元素之间的关联的形式表现的。

关系是数据之间的关联的一种特殊形式, 而这种信息同数据元素本身所代表的信息是同样重要的。‘关系的表现形式可以是数据本身(表示关系的数据)、数据库中数据的编排、元数据(metadata)以及关于数据的数据。它们可以直接表述或者将关系进行编码(比如索引或者面向对象方式)。

数据库的完整性是指其维护数据以及关系中的信息的能力, 数据库的完整性是关于记录的完整性, 而不同于数据的完整性。关系的完整性是数据库完整性的一种表现, 用来维护数据元素之间的特殊关系。

下面对于数据库管理以及维护数据库完整性的方法进行简单的讨论。

- **整体化。**通过完善定义, 数据库可以成为一个整体, 也就是说, 其所有的数据元素以及关系对于整体来说都是重要的。如果任何元素或者关系丢失或者被破坏, 那么完整性就会破坏。当然, 这可能与实际的数据库管理有所不同, 因为可能包含两个或多个相互独立的数据库, 除此之外基本一致。将数据库的所有元素统一管理有利于完整性的维护, 因此, 很多数据库管理系统倾向于将数据库统一管理。

- **惟一的所有者进程。**因为数据库是一个整体, 因此必须有一个进程可以对所有的数据进行浏览、管理, 并对完整性负责, 这个进程通常是数据库管理进程, 这也就意味着数据库管理进程通常是一个单进程。

- **冗余。**为了增加数据库在媒体和设备上存储时的可靠性, 多数数据库管理系统通常使用一些冗余数据, 因此记录的数据往往比其最小的数据量要大很多。

- **动态错误检测与纠正。**通常, 冗余是以错误检测与纠正码的形式存在。数据以编码的形式记录, 从而能够检测出任一位的改动并可能定时自动纠正, 一个典型的例子是奇偶校验位。有些编码非常强大, 能够动态检测并纠正多位的错误。这些编码可以用在存储设备上, 也可以用在数据库管理中。

- **复制。**对一个数据库或者其元素进行复制也是一种冗余, 因为对数据库中关系最为熟悉的莫过于数据库管理系统, 因此复制工作通常由数据库管理系统完成。

- **镜像。**镜像是复制的一种形式, 利用其可同时维护数据的两份拷贝。镜像是一种内部机制, 从外部是不可见的。镜像可以在同一设备进行, 也可在不同的设备上。当在同一设备上时, 镜像能够保护数据库不受部分设备毁坏的影响(比如坏磁道)。当在不同的设备上时, 则能够保护数据库不受整个设备崩溃的影响。

- **备份。**数据库的备份通常可以独立于数据库管理系统。除了其他的原因, 备份主要用于若管理系统发生意外时仍能够保证数据库不受损坏。备份工作可以通过数据库管理系统或其他程序进程自动执行。当然, 尽管备份主要为了防止数据库管理系统的意外事故, 备份系统的独立性本身可能是对数据库完整性的一种破坏。

- **重构**。当数据库崩溃后，备份数据库可以用来重构数据库，从而可以尽量减少对系统的影响。从某种意义上来说，至少在某些情况下，数据库的完整性将依赖于备份数据库的完整性。

- **分割**。分割就是将一个事物划分为若干分离的部分。这样做的主要目的是，当在某一部分中发生意外事故时，将影响限制在这一部分中而不会影响其他部分。

- **隔离与独立**。数据库管理系统常常对子进程实施隔离与独立的原则来保持完整性。例如，一个进行刷新的进程，可能与检查其是否正确执行的进程以及矫正进程相隔离。这样做的目的是为了减小同样的错误会影响三个进程的机会。

- **封装**。数据库管理系统可以看作是一个包或者是容器，其目的是为了防止数据库免受来自外界的影响。封装可以是物理的或者是逻辑的。对于一个数据库管理系统，物理封装可以通过一台分离的计算机来提供，而逻辑封装可以由一个共享计算机及其操作系统中的一个孤立且受保护的进程来提供。大多数的数据库管理系统都对其所包含的数据库进行封装，面向对象的数据库管理系统更是如此。逐渐地，数据库管理系统本身也可以被封装在它们所在的硬件设施中。

- **隐藏**。对数据库信息进行隐藏使它们对于外部是不可见的。虽然这样做不能够使数据库免受破坏，但能够保护数据库免受未经授权的数据泄漏以及恶意的改动。隐藏可以通过几种方式进行，最常用的方式是进程之间相隔离以及使用密码等。

- **原子刷新(Atomic Update)**。原子刷新意味着任何对数据库的改动要么完全执行完毕，要么不执行，也就是不存在部分的刷新。大多数数据管理系统通过回卷所有不能完成的部分刷新来执行这一功能。

- **锁定**。对数据库的一个潜在的威胁是多个进程同时对数据库的使用。例如，假设两个用户同时修改数据库，很有可能第二个人的修改会覆盖第一个人的结果。因此数据库管理系统需要提供一些机制来防止此类事件的发生。通常数据库管理系统使用锁定机制来确保部分更新的数据以及关系不能被使用，包括将元素标记为“正在使用”以及对一次更新中所涉及的所有元素的“请求锁定”。这种机制将不允许对正在使用元素的再次使用，而且在所涉及的所有元素的锁未全部获得之前不会进行更新。这里的锁是一种逻辑机制而不是物理机制，通常只是一个比特或者一个标志来表明锁定和解锁。

- **访问控制**。访问控制是数据库管理系统提供的一种机制，使数据库的拥有者和管理者能够控制哪些用户使用哪些进程可以更改数据库的元素和关系。这些访问控制主要用于管理数据库被多个用户使用时的情形。它是一种完整性机制，将能够更改数据库的人数减小到预定的人数。

- **特权控制**。大多数数据库管理系统，尤其是提供了访问控制的系统，往往会提供一些特权控制。这些特权控制主要由系统的管理员所使用，它们用于执行最终的控制，尤其是对一些特殊情况进行处理。一种特殊情况是用于使访问控制无效，这对于避免死锁状态非常必要，另一种特殊情况是对数据库本身进行修复。

- **复原**。这是完整性机制所最后能够依赖的手段。当数据库崩溃后而任何其他的机制都不能对其进行修复时，只能尽力对数据库进行复原。这通常需要从外部执行并依赖于外部资源，比如备份的数据库。虽然必须将数据库恢复到具有完整性的状态，但很可能会花费很高，也可能遗失数据。

数据库的完整性是至关重要的，任何人都不能不依赖于数据。相比完整性的重建，完整性的维护更容易一些，但是，没有任何单一工具或机制能够满足此要求。因此数据库管理系统将采用多种工具，通过使用外部工具来弥补数据库管理系统内在的不足。

至少下列因素对于维护数据库的完整性是必要的：

- 必须维护数据库中的数据元素以及它们之间的关系。
- 必须理解并能够利用数据库管理系统所提供的机制。
- 必须做到不能泄漏任何一种数据库管理系统所提供的机制。
- 必须理解数据库管理系统的局限性并对其进行弥补。

12.7 计算机操作安全

操作安全包括与操作员和系统管理员特权相关的数据中心和分布式处理的安全性，对计算机资源的安全保护，以及对于重要资源的潜在威胁的洞察等。在操作环境中所遇到的最困难的问题是采取何种措施来控制入侵者的攻击问题。因此下面会对如何对入侵者进行有效处理以及入侵检测系统进行讨论。

12.7.1 安全威胁

随着计算机以及网络技术的发展，互联网的规模也在迅速扩大，而整个系统的复杂性也在不断增长。每年都有更多的系统接入互联网，而系统所具有的存储单元容量也不断扩大。巨大的存储单元允许程序员能够开发更为庞大而且复杂的程序，同时也为程序员犯更多的错误埋下了伏笔，而且庞大的程序也为入侵者提供了更多的空间来隐藏恶意代码。

对一个好的网络安全管理员来说，必须对最新的攻击手段以及相应的对策了如指掌。通常的做法是对已知存在的问题进行及时的处理，这就意味着必须密切关注相关组织的报告，若发现了任何新的弱点，安全管理员必须立即采取合适的措施。但不幸的是，各种问题在以不可思议的速度增长，将安全管理员置于一种困难的境地。不论采取何种措施以及使用何种设备，都很难评估改善的程度，因为互联网是一个高度动态的环境而且没有提供任何好的控制方法来进行监测。通常的观点是：无论情况多么糟糕，如果不采取任何措施的话情况会更糟。有时管理员认为情况得到了有效控制，而实际上却没有，从而造成了管理员的麻痹。而对于入侵者所将要造成的影响程度的评估也是困难的，因为危险的入侵者往往会尽可能对他们的身份以及采用的方法进行保密。

1.对威胁的评估

有很多方法可以用于对威胁进行评估，大多数的网络安全管理者通常采取几种方式，其中包括一些主观方式。主要方式通常有以下几种。

- **查阅。**可以查阅一些以网络安全为主题的信息资源，包括书籍、技术论文、报刊文章、新闻组以及邮件列表等。各种资源都有自己的长处，也有不可避免的弱点。报刊文章尽管很少带有偏见，但是在技术细节上缺乏一定的深度；技术论文大多数情况下过于技术化，有时所描述的威胁并不是公众所面临的威胁；而书籍中所述的信息往往很快就会过时；新闻组以及邮件列表虽然能够提供及时的信息，当通过网络进行传输时同样会遭受攻击。因此需要对各种资源进行权衡利弊，充分利用其优点。

- **实验。**获知入侵者进入系统的困难性的一种方法是进行自我攻击。自我攻击日志对于发现自身的弱点以及增强保护意识来说是一种非常有用的方法。类似的，模拟信息战争也为入侵复杂程度提供一些有用信息。但是这些方法只是人为设计的，而并不能完全代表实际的攻击。

- **调查。**安全调查所获得的统计数据可以提供给管理者一些有用信息以便做出决断。然而，有很多计算机入侵事件没有被发觉或者没有报告，这样就使得安全调查的价值大大减少。

- **测量。**对潜在的威胁进行测量是安全管理者的又一选择，而这一点是非常关键的，因为没有对目标进行好的衡量就不会有好的管理。这就涉及到测量手段的问题，通常使用一些陷阱可以有效地对威胁做出真实的评估而没有将个人和组织暴露的危险。下面会对一个好的陷阱所要具备的特性做进一步的讨论。

2.陷阱的好处以及陷阱的特性

使用陷阱主要有3方面的好处。首先，陷阱提供了真实世界的信息。如果通过适当的设计，入侵者会完全意识不到陷阱的存在，发起入侵时会丝毫没有做作，因此所检测到的入侵是真实的。其次，精心设计的陷阱能够安全地提供一些测量手段。最后，陷阱能够用于延缓将来的攻击。陷阱对于触发事件的响应是陷阱设计的一部分，其超出了入侵检测系统所提供的功能，但入侵检测系统也可以作为陷阱的一部分。一个陷阱主要有3个组成部分：诱饵、触发机关以及圈套。

显然，一个好的陷阱应该是能够捕获到猎物的陷阱，而且一个好的陷阱还应该具备以下的特性。

- **良好的隐蔽性。**显而易见，网络陷阱必须对于入侵者是不可见的，当然，陷阱的诱饵部分是不需要隐藏的，而只需确保诱饵的特性不会暴露陷阱的存在。有很多方法可以使陷阱难以发觉。有些设备比如 SCSI 分析器以及网络协议分析器能够对活动进行监测而不影响被监测系统的性能。同样，日志信息可以通过单向连接传输到系统而进行实时入侵检测。

- **有吸引力的诱饵。**一个陷阱若要有效地吸引猎物，必须具有吸引力好的诱饵。诱饵的选择必须与环境相适应，在某些情况下，诱饵可能是冠有敏感信息的文件或文件夹。当选择诱饵时，网络安全管理员必须考虑入侵者的可能目标，目标可能与该单位的业务有关也可能无关。如果已经检测到以前的入侵行为，则管理员就可能知道入侵者所感兴趣的类型。而且必须注意防止诱饵暴露陷阱的存在，如果诱饵看起来过于诱人以至于不像是真的，则入侵者就会转向其他处，从而避免了被检测的可能。

- **准确的触发机关，**一个好的陷阱应该捕获入侵者而不应该捕获无辜者，因此触发机关应该进行精心设计以期将失误率降低到最小。设计时必须考虑由于失误所引起的失去信用的问题，这是非常重要的。因为失误事件可能会对该单位造成比入侵本身更重大的损害。陷阱所在的位置也可以有效地提高触发机关的精确性。比如，陷阱可以置于合法操作所不能到达的地方，而只有非法用户才可能触发机关，这就大大降低了失误率。

- **强有力的圈套。**一个有效的陷阱必须有足够的能力来抵抗入侵者，这一点是设计一个有效的陷阱最为困难的事情。首先应该能够识别入侵者的身份，而目前一些好的陷阱通常具有保留证据的能力。在比较复杂的事件中，入侵者可能通过网络穿梭技术对系统实施入侵，中间可能穿过了一个或几个系统。在这种情况下，实现对入侵者的路径回溯是比较困难的，有可能需要第三方组织的协助。

为了使陷阱的有效性最大化，通常可以同时使用多个陷阱。而且一个好的陷阱应该是惟一的，因为入

侵者知道了某种特殊类型的陷阱，他就不会被再次欺骗，尤其对陷阱的可见组件诱饵来说更是如此。

网络入侵者可能会非常聪明，发起的攻击可能是未曾遇到过，因此需要有相应的技术来检测并延缓这种攻击。虽然使用陷阱并不能使网络安全管理员一劳永逸，但能够使管理员将注意力放在最重要的安全区域，从而能够有目的地对入侵行为做出有效的措施。

12.7.2 入侵检测

假设一个陌生人站在你的房前（他环顾四周后走到你的门前并开始旋转把手，门是锁着的。然后他又走到最近的窗前并试图打开它，窗也是锁着的。这样看来你的房子是安全的，那么为何还需要安装警报器呢？同样，对于一个系统已经安装了防火墙、操作系统补丁以及验证口令，为什么还需要入侵监测系统呢？答案很简单：因为即使如此，仍然会发生入侵行为。如同人们有时会忘记锁上门窗一样，系统管理员有时会忘记及时更新防火墙的规则集。即使采用最先进的保护技术，计算机系统仍然不可能是百分之百的安全。事实上，绝大多数的计算机安全专家也承认，永远也不可能拥有一个完全安全的系统。因此，必须采用入侵检测技术来对付计算机系统的非法入侵者。

1. 入侵检测的历史简单回顾

入侵检测的最初形式是系统管理员坐在控制台监视器前观察用户的行为。他们也可能发现入侵行为，例如某人非法登录或某台机器非正常运转等等。尽管在当时这种方式已经足够有效，但是这种入侵检测方式非常特殊而且不宜于大规模使用。

后来入侵检测使用了审计日志，系统管理员可以通过其来检查非正常或恶意的行为。在 20 世纪 70 年代后期 80 年代初期，管理员通常将日志打印在纸上进行审查，这样既浪费时间又耗费资源。通过这种过量的信息和人工的分析，管理员仅仅只能将审计日志作为入侵行为的一种证据而对于入侵行为的攻击却束手无策。随着存储器价格的降低，审计日志可以在线执行，数据可以使用程序来进行分析。然而，分析的速度较慢而且计算较为复杂，因此入侵检测程序常常在夜间系统的负载量较小的时候运行，所以大多数的入侵行为仍然是在事件发生后才检测到。在 20 世纪 90 年代初期，人们研制出实时入侵检测系统并能够即时检查审计数据。这样系统可以在攻击或试图攻击时及时做出反应，从而在攻击与反攻击中占据主动地位。最新的入侵检测技术致力于开发一些人们可以在大规模网络上使用的产品。

2. 入侵检测概述

入侵检测的目的看起来比较简单，即检测入侵行为，然而，具体的任务却大不相同。事实上，入侵检测系统根本不检测任何入侵行为，它们只是识别入侵行为过程中或结束后所遗留的痕迹。这种痕迹有时可以作为入侵行为的“表现形式”，如果没有这种入侵的表现形式或表现形式中缺乏足够的信息来确认其是入侵行为，那么系统就不能检测出入侵行为。一个简单的例子，假设房屋的监视系统显示出有人在开门，那么监视器的数据信息可以作为正在发生的入侵行为的证据。但是如果摄像头、未准确聚焦从而导致图像模糊而无法分辨此人是入侵者还是主人，那么此数据信息就不能成为入侵的证据。

为了准确判定入侵行为，必须掌握目标系统行为足够多而且可靠的数据信息。可靠数据的收集本身是一个非常复杂的问题。大多数操作系统具有某种形式的审计功能从而对不同的用户可以提供操作日志。这些日志可能仅局限于与安全相关的事件（比如登录失败）或者能够提供每个用户对系统所执行的每个操作的详细报告。同样，路由器和防火墙能够提供网络活动的事件日志。这些日志可能只记录到一些简单的信息，比如网络连接的建立与关闭，也可能完整地记录了流经此线路的每一个包的情况。

检测系统所收集的系統活动信息的数量应该是数据的有效性和系统开销之间的一个折中。系统若详细记录每一个事件就会潜在地降低系统性能并需要更多的存储空间。例如，对于一个 100MB 以太网结点，对所有包数据的日志收集每天将需要几百吉字节的存储空间。信息的收集是昂贵的，而收集正确的信息是至关重要的。确定收集哪些信息并从哪里收集，这也是一个比较重要的问题。

3. 检测技术

入侵检测系统如何对所收集的信息分析是系统的一个重要特征。目前主要有两类入侵检测技术：异常检测(anomaly detection)和误用检测(misuse detection)。

异常检测的一个基本假设是入侵者行为异常于正常主体的行为。比如，我们能够比较准确地将一个人的日常活动进行模式化，假设某人通常在上午 10 点登录，阅读邮件，处理数据库，然后在中午休息一会，并且其对文件进行访问时极少出错。如果某天系统检测到同一个人在下午 3 点登录，使用了编译和调试工具，访问文件时出现了很多错误操作，那么这种行为就有可能值得怀疑。异常检测的主要优点是能够检测到以前所未知的攻击。通过定义正常的行为规则，可以检测出任何的异常行为，不管其是否是威胁模型的一部分。然而，在实际系统中，这种检测未知攻击的特点也往往伴有较高的失误率。异常检测系统在一个高度动态的环境中会受到很大的影响。

误用检测系统本质上定义了入侵行为模式，这一检测假设入侵者活动可以用一种模式来表示，系统的目标是检测主体活动是否符合这些模式。它可以将已有的入侵方法检查出来，但对新的入侵方法无能为力。其难点在于如何设计模式既能够表达“入侵”现象又不会将正常的活动包含进来。误用检测系统的主要优点在于它注重于分析审计数据，从而产生较小的失误率。其缺点在于只能检测已知的入侵方法，因为它具有此入侵方法的特征，当新的入侵方法出现时，管理员必须将其模式化并加入到特征数据库中。

当系统检测到入侵时，它会及时做出响应。响应措施往往表现为几种形式，最通常的方式是产生警告提醒入侵行为已经发生。系统还可以采取一些野蛮响应方式，比如进行反攻击。反攻击的方式可以是对路由器进行重新配置从而阻塞攻击者的地址。显然，野蛮响应具有一定的危险性，因为它可能伤及到无辜。比如，攻击者使用假冒地址对一网络进行攻击，表面上看来攻击来自此地址，实质上却另有它处。但是当入侵检测系统检测到攻击时，重新配置了网络路由来阻塞来自此地址的数据，就很容易对一个无辜的站点造成拒绝服务攻击。

4.存在的问题

尽管入侵检测在近几年内有了飞速的发展，仍然存在一些尚需解决的问题。首先，入侵系统必须具有更高的效率，以最小的失误率检测到最大范围的攻击。其次，随着现代网络不断向大范围、高速度和动态发展，入侵检测必须紧跟网络发展的步伐。我们需要进一步的分析技术来支持对各种类型攻击的有效识别。

5.系统的有效性

对不断发展的系统有效性所提出的一个挑战是，如何开发出一个系统，使其能够检测出接近百分之百的攻击行为而具有最小的失误。目前我们距离这一目标还很远。目前的入侵检测系统主要依赖于误用检测技术。比如 Snort 以及 RealSecure 的产品主要还是使用了特征来分析网络业务。因为他们仅仅能够对已知的攻击进行模式化，因此管理者必须对他们的特征集经常进行更新。还需要使用异常检测的能力去检测新的攻击行为，但必须提高异常检测的准确度。目前，多数研究人员提倡使用误用和异常检测相混合的技术，但这种方法还需要更进一步的调查和研究。

6.性能

仅仅能够检测到各种攻击是不够的，入侵检测系统必须能够承受高速度网络和高性能网络结点所产生的事件流的压力。目前所用的网络 G 比特以大网是比较通常的，高速光纤连接也越来越普遍。网络结点的速度也变得更快，能处理更多的数据从而产生更多的审计日志。这就使我们重新面临着以前系统管理员所面临的庞大的数据量的问题。有两种途径可以用来实时分析这样庞大的信息量：分割事件流和使用外围网络传感器。

在第一种方法中，可以使用一个分割器将事件流切分为更小的可以进行管理的事件流，从而入侵检测传感器就可以对它们进行实时分析。这种方法的问题在于分割器必须以某种方式来切分事件流以保证能够检测到相关攻击行为的所有现象。如果对事件流进行随机分割的话，则传感器可能得不到足够的数据来检测入侵行为，因为攻击现象的不同部分可能是由不同的分片所表现出来的。

第二种方法是在网络外围并靠近系统必须保护的主机附近使用多个传感器。这种方法的问题在于使用和管理一系列分布的传感器是困难的。首先，正确选择传感器的位置很困难。依赖于网络拓扑结构的攻击，比如基于路由选择和地址欺骗的攻击，需要检测传感器放置在网络中的特定地点方可起作用。其次，这种方法还有一个控制和协调的问题。网络是随着时间不断发展的动态实体，而攻击也是不断发展的，每天都可能产生新的攻击方式，因此检测基础设施也必须相应地发展。

7.基于整个网络范围的分析

若将传感器放置于网络的关键位置，管理员可以将网络作为一个整体进行攻击行为检测。这样，可以为整个网络的安全状态提供一个整体的描述。一个攻击行为在单机环境中可能表现得无关紧要，但在整个网络的环境下进行考虑可能是极为危险的。比如，一次攻击行为涉及到多个步骤，假设每个步骤在不向的主机实施，但是因为此系统有一个共享的文件系统，从而使整个系统受到攻击。当分析一个单独的传感器信息时，系统可能不会意识到攻击的某一个步骤是恶意的，但通过对整个网络活动的综合分析可以揭示攻击的不良意图。这种整体协同警告系统，即基于不同传感器的警告来检测入侵的系统是目前入侵检测中最富有挑战性的问题。

尽管网络会变得更加安全，入侵检测仍然是任何重要的安全解决方案不可分割的一部分。随着传感器进一步使用和推广，一个系统中可能包含有成百成千个入侵检测传感器，它们通过基础设施相连并支持相互通信、控制和重新配置。逐渐地，对低水平传感器进行分析会转变为对高水平分析器进行分析，这样就会使系统管理员对整个网络中的重要的安全事件有一个更好更准确的掌握。

在不久的将来，传感器技术将集成到我们日常的计算环境中。现在已经有一些类似于防火墙的技术集成到操作系统中，比如，UNIX 和 Windows 提供了一些基于主机的防火墙的技术。目前，将入侵检测传感器集成到操作系统和网络软件中的时机已经成熟，毫无疑问，入侵检测将成为一种默认配置，而不是一种秘密选项。只有当不同类型的传感器能够集成运行在不同的平台环境和系统中时，才有可能使用这种传感器网络。而且，需要制定一些标准来支持它们之间的互操作性。目前，IETF 的入侵检测工作组提出了入侵检测消息交换格式(intrusion detection message exchange format, IDMEF)标准，IDMEF 定义了警报的格式和警报交换协议，进一步需要做的事情是提供一个公共的实体，使得所有的传感器能够对出现的现象达成共识，否则，当检测到相同的入侵时，不同的传感器将会做出不同的反应。随着技术的不断发展，基于软件的入侵检测可能发展为基于硬件的入侵检测技术，而且随着新型的传感器的产生，入侵检测将会开辟一个新的领域。

12. 8 业务持续和灾难恢复规划

业务持续性规划和灾难恢复规划涉及到一些特定的或相关的规划，当正常的信息处理业务突然中断时，用来减轻甚至避免其所带来的影响。它们用来保证维持组织运作的关键系统的可用性。

信息系统和业务的持续性很容易受到自然的和人为的攻击。各个组织必须经常对潜在的业务破坏做出规划并经常对自动系统的恢复规划进行检测。而且，针对于各种新技术发展所提出的挑战，各个组织必须对持续规划进行相应的重新管理配置。

在 IT 环境中，确保业务的持续性和灾难恢复是一种挑战。而且，大家公认当前的计算环境同以前相比更加复杂而难于管理，随着系统向分布化发展，对系统的控制和管理比一个集中系统更加困难。在 Web 应用领域中，很多控制都来自于本组织的外部、因此管理愈加困难。

12.8.1 业务持续性规划

一些组织往往不能准确地估量持续性规划(Continuity Planning, CP)对整体的成功所做的贡献，这就导致了整体业务持续性规划的螺旋式下降。这种循环的螺旋式下降过程主要表现为规划、测试、维持、下降，然后再规划、测试、维持、下降这样循环往复。过去，持续性规划及管理调查一再证实：持续性规划对于行政管理来说是极为重要的。但是在实际中一直存在着持续性规划的实施与如何评价其价值的方法之间相脱节的问题。由于缺乏有效的评价手段，常常使得持续性规划不能取得预期的成果。在这种情况下，对持续性规划的实施方式进行改造已经成为目前所急需解决的问题。

最近，企业管理效率方面的专家已经开始引进流程改进原则，这些原则正在逐渐被很多企业所采用用来增强生产与管理业务流程。一个企业的各项流程是企业主要组成单位，如果各项流程能够得以高效的实施，则会明显减少企业业务中的失误并增加企业的生产力。一个企业的流程主要是一系列的活动，当它们集合在一起时，便构成了该企业任务的基础。这些流程与企业的基础设施(比如个人业务单元、部门等)交织在一起，并且同企业的支撑结构(如数据处理、通信网络、物理设施以及人员等)联系胜一起。

人们的管理意识不断在提高，但是持续性规划的实施效率仍然很低，而且缺乏一种一致而有效的手段来评测持续性规划，这也对改进现有的恢复规划实施手段提出了要求。

对于恢复规划人员来说，需要进行下列工作：

- 建立一个恢复规划的实施所需要的工程组以及相应的支撑基础设施。
- 实施对攻击行为以及风险的管理评估，从而识别其是否是恢复规划所需解决的问题。
- 实施业务影响分析，用来判定业务的时间急迫性以及确定最大可忍受停工期。
- 恢复规划的保存和实施。
- 建立并采取一种可实施的测试和维护策略。

以前当主体框架比较简单时，这些方法是行之有效的。随着分布式以及客户机/服务器系统加入到整体的恢复规划基础设施中，这些方法也是有效的。但是当企业开始与业务单元恢复规划息息相关时，这些传统的灾难恢复方法在设计和实施业务单元恢复规划上的效果并不理想。当在整个企业范围内试图实施恢复规划时，主要的问题是各个业务单元之间的相互依赖性问题。

能够对持续性规划的各部分之间的相互依赖性做到清楚明了是非常困难的。大多数的现代化企业正在经历着高速度的变化，其中包括企业的重构、人员的变动、竞争环境的改变等，从而导致了持续性规划各部分之间的相互依赖性的变化。每次企业的结构发生了变化，持续性规划必须进行相应的改变，这种相互依赖性也需要进行重新评估，而且变化越快，持续性规划的情形就越糟糕。因为我们并不能够对其中很多相互依赖性完全追踪，因此就会破坏持续性规划的完整性，这种困境很难得到解决。

而这种相互依赖性是什么呢？为什么这种相互依赖性如此重要呢？从很大程度上来说，这些相互依赖性就是企业的业务流程，它们之所以重要是因为它们的正确运算是完成各项任务的保证。利用业务流程的

观点来进行恢复规划从很大程度上来说会减少这种对相互依赖性失去把握的程度，而且能够保证恢复规划的重点放在企业最关键的组件上。对企业关键业务流程结构的理解，有助于规划人员将流程与业务单元或部门建立对应关系，有助于规划对各种技术系统、网络、设施、关键记录、人员等等的支持，而且能够帮助规划人员在企业重构或发生变化时对流程实施跟踪。

1.持续性规划的方法

传统的面向主体框架的灾难恢复规划方法重点在于满足恢复企业技术和通信平台的需要。目前，很多企业的重点已经从技术恢复转向优先业务流程的持续性规划以及特定业务流程的恢复规划。很多大型企业正在使用业务流程改进(business process improvement, BPI)和业务流程重组(business process reengineering, BPR)来增强整体的企业生产力。持续性规划本身也应该看作是一个流程，而企业范围内的持续性规划流程主体框架主要由 4 部分组成。

- **灾难恢复规划(DRP)**:灾难恢复规划由许多关系到业务持续性的特殊操作项目组成，还概述和详细说明了发生人为破坏或自然灾害时对各种潜在危害企业的事件所采取的特殊步骤。灾难恢复规划过程中的某些典型阶段包括认识与发现，风险评估，缓解，准备，测试，响应和恢复。

- **业务恢复规划(business resumption planning, BRP)**:业务恢复规划涵盖了业务持续性规划的运作面)它对于数据可用性来说是至关重要的。主要包括紧急事件处理，资源需求，规划开发，规划实施，质量保障和变化管理。

- **危机管理规划(crisis management planning, CMP)**:危机管理规划注重于帮助企业发展一种有效而且高效的紧急事件以及灾难响应能力。这种响应能力包括组建适当的管理团队以及对成员进行培养使之能够在紧急的情况下(比如飓风、地震、水灾、火灾、病毒入侵等)迅速做出反应。

- **持续可用性(continuous availability, CA)**: 对一个企业来说，如果其业务能力哪怕中断很短一段时间，都会对企业造成重大的金融(比如税收减少、额外支出等)或运作(比如顾客服务、威信降低等)方面的伤害。而 CA 服务主要致力于将企业的支撑基础设施的正常工作时间维持在 99%甚至更高。

2.持续性规划流程的衡量手段

对于持续性规划来说，需要有一个有效的衡量手段作为其流程的完善和补充。同时为企业提供了一种度量标准来衡量其整体持续性规划流程的效果。

这些手段通常包括：

- 在企业中的热点领域投入了多少资金？
- 有多少人员致力于持续性规划活动？
- 对热点领域的测试是否取得了成功？

而现在焦点应该放在测量持续性规划流程对企业的整体目标所作的贡献上，这样做有以下好处：

- 识别持续性规划发展中的重大事件。
- 为任务的实施建立一个基础标准。
- 增强持续性规划的实施。
- 为管理者成功地管理预期事件建立一个有力基础。

为了对持续性规划流程进行有效的衡量，提出了一种持续性规划平衡记分卡的概念，其中包括以下的定义：价值综述，价值计划，持续性规划风险度量标准，执行协议，有效方法。

图 12.2 描述了平衡记分卡的概念。

在平衡记分卡方法中，企业需要确定持续性规划流程的远景目标。远景目标的确定需要同企业的高级管理以及持续性规划流程基础设施的发展相协调。一旦确定了远景目标后，持续性规划流程发展人员就可以勾画出持续性规划流程改进中的成功关键因素，其中的领域包括增长与改革、顾客满意度、员工情况、流程质量以及财政情况。在持续性规划流程改进中可以进行评测的持续性规划流程组件包括流程方法论，DRP 文献，BRP 文献，风险管理计划文献，紧急情况响应计划文献，网络恢复计划文献，企业普查的持续性，员工意识培养情况，恢复变更的费用，持续有效性基础设施，正在进行的测试计划。

上述这些组件可以从人员、过程、技术、任务以及利润等各个角度进行评测。但是这些评测标准必须基于特定的企业文化以及环境进行实施。

12. 8. 2 灾难恢复规划

据估计，多于一半的企业都没有一个可靠的、完整的、可用的灾难恢复规划。因此，很多企业正在致力于创建自己的灾难恢复规划。灾难恢复规划的主要目的在于，当企业的正常运作受到了突发事件的影响

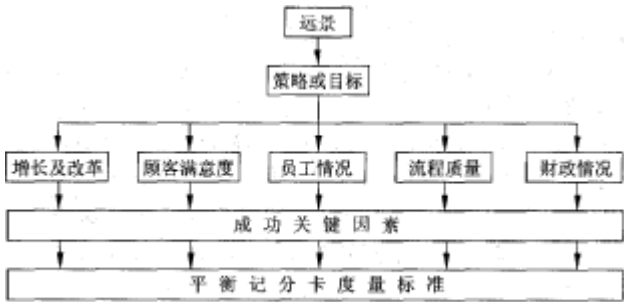


图 12.2 平衡记分卡的概念

而中断时，提供一种变通的手段。灾难恢复规划必须对可能对业务运作造成中断的所有类型的事件进行处理，这些事件可能是计算机故障，但它们往往是一些其他的内部或外部的突发事件，可能是环境的因素(比如火灾或水灾)或者是其他外部因素(比如电力中断)。

灾难恢复规划的一个主要用途是尽量减少灾难发生的可能性。当灾难恢复规划组在检查各个领域并制定规划时，他们会意识到系统或者是企业所面临的风险，以及导致运作失败的弱点所在。这些弱点可能会存在于系统、进程、硬件、软件或人员问题中，或者是来自于其他形式的环境或外部威胁。因此规划的目的在于建立一个框架，使得业务流程在中断后能够迅速以一种变通的方式恢复到原来的运作状态。灾难恢复规划的执行速度主要依赖于系统的重要性，一个关键的系统必须具有一个能够在几秒钟或几分钟内实施的规划，而非关键系统的规划实施的速度则可能是几天甚至几周。

一旦灾难发生，业务组的首要任务是尽快恢复关键系统并尽可能小地减少对关键系统的影响，同时灾难恢复规划开始实施。灾难恢复规划的第一个目标是阻止进一步的破坏，即首先是保证人员的安全。然后灾难恢复规划就可分为 3 部分：清扫被破坏的区域(抢救或修复)、实施变更的业务运作、返回到正常的流程。其最终目标是业务运作能够恢复到正常的未破坏之前的状态。为了提高有效性，灾难恢复规划必须进行存档记录。所有的责任和任务、软件和硬件、通信连接以及安全需求都必须进行记录以便需要时立即可以使用。

任何有规模的机构必然需要有完整和清晰的危机处理和灾难恢复规划，从最平常的停电、不同程度的系统损毁以至于火灾，甚至和整个地区有关的严重天灾等等。虽然不可能说可以兼顾所有情况，但根据损毁程度来进行策划及处理是不可忽略的。更进一步来说，甚至应该根据不同系统的重要性，再评定出各种处理方法。

信息系统的危机处理及灾难恢复主要可以分成下列几种。

- 与日常生产及运作息息相关的关键性系统。例如股票交易系统甚至航空控制等，未经预先计划的停顿可能引致灾难性结果。一般来说不但系统本身应该拥有高度自动恢复能力，使系统出现故障时可以迅速继续运作，并且通常会在另一个区域内有全面的后备系统，而数据会不停更新常规和后备系统，确保出现问题时能迅速地转由后备系统继续保持不间断的运作。

- 部分机构的重心系统，也会采用类似的架构，但限于同时保持两套系统同步运作不但技术难度高，系统高昂，而通讯和保安也是大难题，折中方法是容许常规和后备系统有时限上的差异(最常见的情况是 24 小时)，数据不一定能完全同步，后备系统需要若干时间才能上线运作，但一般情况下是足够应用的。有时候配合应用系统增加额外的操作记录，也能加快备用系统上线。

- 在另一地一区设立规模较小但架构相同的系统，使用离线的方法，例如用数码磁带复制常规系统后再注入后备系统中。这种方法执行上类似于离线备份，但好处是不需要在紧急时再为寻找后备服务器和设定系统费心，特别是较复杂的专用服务器的系统设置需要耗费很长时间，能够进行预先准备可以减少很多麻烦。

- 最基本的灾难恢复当然是利用备份工具，包括数码磁带、高容量的磁带、磁盘或读写光盘等等，根据所需备份的数据量来进行策划。许多系统，特别是与数据库有关的系统，都要有预先准备的备份方法。如果没有正确执行备份方法，就会导致备份工具不能正常发挥作用，那么问题发生后就会发现系统无法恢复，从而导致严重的失误。因此备份系统的定期测试也是必须的，理想的方法是定期，例如三个月或半年等作仿真恢复试验。而更重要的是保存两个或以上的备份，并把其中一个储存在其他地方。

12.9 物理安全

物理安全涉及到为进行信息处理所提供的安全环境的问题，特别是要阻止未经授权的人对计算机装置进行物理的或技术上的访问问题。

物理层安全是整个体系安全的假设前提，而且有理由认为物理层是围绕其他层周围的第一层防御。物理安全不是很简单的规则，它可能会被足智多谋的外部的信息犯罪所破坏，如果犯罪者是内部以官方身份进入特定设备并进行破坏的话，物理安全对这个人是不起作用的，但无论如何，物理安全是一个必要的最初的保护形式。

物理安全是一个连续的过程，不能通过预先考虑的手段来完成。物理安全所采用的方法必须与组织的目标相一致，而且它的实施必须符合信息安全政策所规定的标准。因为物理安全的世界往往变化不大(至少比信息安全中的其他技术的发展速度较慢)，因此经常被认为是枯燥而且不重要。这种误解往往导致物理安全被忽视。从本质上看，任何信息安全控制的弱点不是控制本身，而是控制的不合理应用。对待物理安全，必须同对待其他信息安全控制一样要保持足够的警惕性。事实上，安全控制必须实施而且要保持一致性和前瞻性，从而可以获取有效的信息安全。

锁、警卫、监控摄像机以及各种识别标记仅仅是物理安全的一些工具和设备。为了计划和设计物理安全，还必须考虑以下几个问题：

- 保护的是什么？
- 保护的信息有多重要(经济上、政治上还是公共安全)？
- 为谁进行保护？对他们来说什么更重要，机密性、完整性还是可用性？
- 实施保护的目的是为了防止什么？

当然，并不是所有的地方都需要武装到牙齿的物理安全，但是物理安全必须同所保护对象的重要性和敏感性相适应。

1.分层的防御体制

一个分层的防御体制通过提供冗余以及扩展的保护等手段在访问控制方面提高了机密性的级别。对一个安全管理者来说，必须能够对一个分层的防御做出正确的评价，并能清楚地认识到其所能提供的保护。设计一个分层的防御体制通常需要遵循 3 条基本原则：广度、深度及阻碍度。

广度可以理解为在一堵墙上打了几个洞，每个洞代表一个不同的方法或者是不同的系统脆弱点。广度的引进是因为单独一种类型的控制往往很难解决所有的脆弱点。比如在 IT 领域，假设某人决定通过使用登录口令来控制对数据的访问，但是当数据在互联网中传输时，单独的口令并不能保证数据的安全性，而需要另外的加密手段。物理安全也是同样的道理，比如一个房子，它有前门、后门和窗户。门上的锁控制了从门进入房子的一个途径，但是不能阻止人们从打碎的窗户中进入，因此还需要其他的障碍物。

对于分层的防御体制来说，深度是最最重要的一个因素，然而往往会被忽略。在任何实际的安全中没有一种完美的保护措施。因此在深度上，必须增加其他的访问控制层次作为增援措施。从本质上来说，可以理解为将一堵墙变为几堵墙。例如，口令字不会永远保持其机密性，因此在访问控制中仅仅使用口令是不安全的。作为深度上的增强保护，可以使用智能卡，即使口令被攻破，仍然不能进入系统。对于物理安全来说，在距离被保护对象较远的区域到被保护对象的核心区域，深度都起到一定的作用。

第三个原则是阻碍度，也就是实施保护时，所控制的花费应低于所保护的对象的價值。这里主要的问题是，针对于一定的阻碍度，如何获得最大的广度和深度。

2.多方面防御机制

物理安全的一个基本任务是让不受欢迎的人离开并保持内部人员的诚实性。对于 IT 安全来说，物理安全的基本任务也是类似的，不仅仅要保护人员、纸张以及财产，更重要的是要保护数据信息。物理安全涉及到多个方面，各个方面之间是相辅相成的，而其中每一部分的使用必须同其他部分的使用保持一致。各个部分之间的基本关系就是共同满足防止安全事件、检测安全事件和评估安全事件的需要。

物理安全的实施通常包括以下几个方面。

- **确认。**通常美国政府将此作为分类指导方针。确定什么需要保护，从而根据如何对其进行识别(是通过主观事件还是关键字)来制定指导方针。指导方针必须使得即使一个新手也能分辨出文档是否是敏感的(基于内容)。比如制定一个电子分类向导，使得用户可以容易地判定材料的敏感度以及需要的策略。

- **标注。**使用橡皮印章或其他手段来对敏感文件进行标识。文件夹应该是明显区分的而且进行了标注。标注应该指明特殊的处理需求、数据的敏感度以及授权访问的人员等。

- **安全。**基于存在的风险构造物理防御层。下列是每个物理防御层可能要考虑的因素，但并不是所有人都需要这些因素。

周边设施：周边访问控制，包括物理屏障，比如栅栏、墙、倒钩等等，可以在周边设施安装警报器以及摄像机。

建筑入口：建筑的入口可能是门或窗户，可以使用锁、警报器、摄像机或者卡访问系统等。

建筑楼层：进入建筑后，可以通过楼层限制来进行控制，比如使用特殊的电梯钥匙或在楼梯上安装警报器以及摄像机。

办公室装置：办公室装置的访问控制包括锁、卡访问系统、警卫以及保险门等。在办公室内部，可以使用保险柜、保险库、防盗窃装置以及警报系统等。

使用防火墙、代理服务器、安全路由器、网络地址转换、网络监视等等，使用口令、用户认证、文件访问权限、数据备份、数据加密等技术，以及采用不同断电源、备用设施等防御手段。

- **跟踪：**使用访问列表、列表检查、目录控制、审计日志等方式对访问进行跟踪控制。

- **技能：**保证人员知道如何进行保护以及保护原因，并制定策略来实施这些保护措施，保护措施应该明确所需要的访问控制以及处理手续。不同的工作具有不同的责任，因此表现形式也有所不同。处理手续应该包括如何进行复制、如何发放邮件、资料的保密期以及资料销毁等等。

3.物理安全存在的缺陷

当实施物理安全时，必须认识到一些普遍存在的局限性以及缺点。

- **社会工程学：**从社会工程学的角度，绕过一个物理安全控制是可能的。比如，当入侵者提供了一个似乎合理而且真实的理由时，警卫就有妥协的可能。而当入侵者骗取信任共享密码锁的密码时，就有可能打并门。

- **密码的泄漏：**同口令一样，为了方便密码也经常会写在某处或者进行邮寄，因此就有些泄漏的危险。

- **尾随：**尾随是进入一所设施所常用的一种方法。当授权的人进入设施时，在门未关上之前跟随进入设施，而在一群人中混杂进入会更加容易。

- **环境因素：**污浊的空气、强烈的阳光、反射以及雾气等等都能够影响摄像机的性能或者使传感器产生错误的警报。强冷和强热可能使设备无法正常工作，树木以及鸟类也可能会影响到周边设施的警报器。

- **装置可靠性：**过冷或者过热可能会影响到装置的可靠性。而一些不稳定因素也会影响到传感器的性能。

- **信任度：**当无关紧要的警报以及由不法分子蓄意产生的虚假警报频繁发生的时候，警报系统的信任度就会降低，从而会忽略警报。

- **用户接受度：**当用户感觉安全措施过于困难或者并不安全，甚至对其造成干扰时，可能会妨碍安全措施的实施，而不管其干涉正确与否。

物理安全是信息安全中的一个重要部分。在一些情况下，一个组织可能缺少信息安全的一些保障措施，但必须有一个好的、坚固的物理安全设施和方案。对于信息安全管理来说，必须清楚物理安全的范畴，并且掌握如何使用物理安全来保护有价值资源。一个完善的物理安全方案将会保护所有的有价值资源，在这个基础上，可以实施其他的保护方案，因此毫无疑问，物理安全是信息安全的基础。