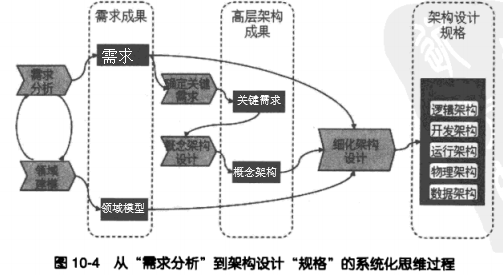
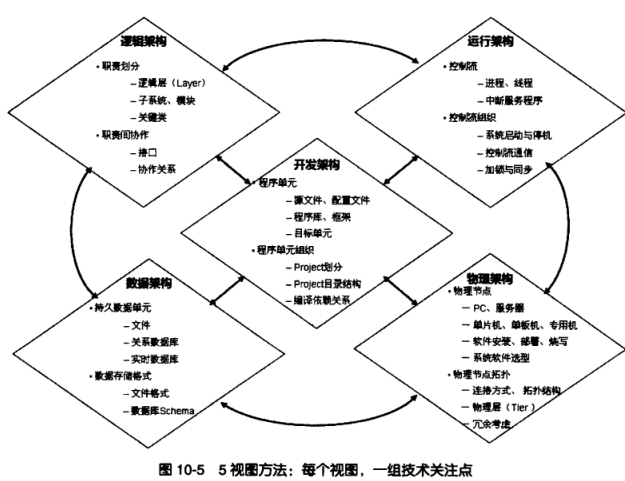
细化架构设计需要概念架构、需求、领域模型作为输入架构



**5个视图，15个设计任务**

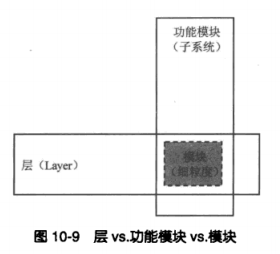
对于不同的涉众，我们需求不同的视图，通常划分5个视图，不同的视图有不同的关注点



**逻辑架构=模块划分+接口定义+领域模型**

1. 设计任务：模块划分

首先进行分层，然后划分子系统，最后依据职责划分模块



1. 设计任务：接口定义

协作决定接口，在定义接口前先考虑模型是如何进行交互的

1. 设计任务：领域模型细化

逻辑架构一般设计到模块一级，但以下4种类必须在架构中明确指出，通过参考领域模型细化逻辑架构

1. 接口定义类
2. Facade实现类
3. 核心控制类
4. 另外，就是对系统可扩展性有根本影响的构成领域模型的那些类

**开发架构=技术选型+工程划分+编译关系**

1. 设计任务：技术选型

包括开发语言（如c++），开发工具（如vc++），程序库，框架（如ORM框架EF）

1. 设计任务：工程（即VS中的”解决方案“）划分

系统划分为几个工程

1. 设计任务：编译关系

指明工程的高层目录以及工程的项目（类库，程序等）的编译关系

**物理架构=硬件分布+软件部署+方案优化**

1. 设计任务：硬件分布

物理机器的分布

1. 设计任务：软件部署

软件是如何部署在硬件上面的（一个系统可不止一个软件）

1. 设计任务：方案优化

对物理架构的方案进行优化

**运行架构=技术选型+控制流划分+同步关系**

运行架构描述系统运行时进程，线程的交互

1. 设计任务：并发技术选型

进程，线程，或中断服务程序

1. 设计任务：控制流划分
2. 设计任务：控制流间的同步关系

描述控制流之间是如何同步的

**数据架构=技术选型+存储格式+数据分布**

1. 设计任务：技术选型

持久化技术选型，如关系数据库，nosql数据库等

1. 设计任务：数据存储格式
2. 设计任务：数据分布策略