应用集成

名词解释

1. EAI

全称：Enterprise Application Integration（企业应用集成）

定义：EAI 是将基于各种不同平台、用不同方案建立的异构应用集成的一种方法和技术。

2. SOA

全称：Service-oriented architecture（面向服务的体系结构）

定义：SOA 是一种应用程序的组织架构，是设计原则，指导如何设计应用程序。它的基本原理是通过组件或 web service 提供的分布式通信能力，把系统中的功能抽象成一个个服务。在 SOA 中服务通过基于消息机制的定义明确的接口和调用协议相互作用，构成应用系统。

3. SOAP

全称：Simple Object Access Protocol（简单对象访问协议）。

定义：SOAP 是在松散的、分布的环境中使用 XML 交换结构化的和类型化的信息的一种简

单协议,本身并不定义任何应用语义,只定义了一种简单的以模块化的方式包装数据的机制, 可以使用任何底层传输协议，如 HTTP、FTP、SMTP 等，其中最常用的是 HTTP 协议。

4. WSDL

全称：Web Services Description Language 网络服务描述性语言

定义：WSDL 基于 XML 格式，定义了 Web Service 接口，描述了服务的操纵信息、服务提供了什么功能、服务位于何处、服务如何调用。/是一个用来描述 Web 服务和说明如何与 Web 服务通信的 XML 语言。

5. MOM

全称：Message Oriented Middleware 基于消息的中间件

定义：指利用高效可靠的消息传递机制进行平台无关的数据交流，并基于数据通信来进行分布式系统的集成。它通过消息传递来完成分布式计算环境下数据和控制的处理,采用多种机制来保证消息可靠、高效、安全。

6. JRMP

java remote method protocol Java 远程方法协议

是一种与 JAVA 技术相关的协议，用于查找和引用远程对象。它是一个线级的协议，运行在技术相关的协议，用于查找和引用远程对象。它是一个线级的协议，运行在 RMI 和 TCP/IP 之间

7. RDF

Resource Description Framework 资源描述框架

一个用于描述 Web 上的资源的框架；针对数据的模型以及语法，供不同的用户来交换和使用；使用 XML 编写。（一个 RDF 文件包含多个资源描述，而一个资源描述是由多个语句构成，一个语句是由资源、属性类型、属性值构成的三元组，表示资源具有的一个属性。）

8. XSLT

eXtensible Stylesheet Language Transformations 扩展样式表转换语言

扩展样式表转换语言,用于将一种 XML 文档转换为另外一种 XML 文档,或者其他类型的文档,比如 HTML 和 XHTML。

9. CORBA

Common + Object Request Broker + Architecture 公共对象请求代理体系结构

一种异构平台下的语言无关的对象互操作模型

由 OMG 组织制订的一种标准的面向对象应用程序体系规范

10. RMI

(Remote Method Invocation) 远程方法调用

一种用于实现远程过程调用的应用程序编程接口。它使客户机上运行的程序可以调用远程服务器上的对象。（远程方法调用特性使 Java 编程人员能够在网络环境中分布操作。RMI 全部的宗旨就是尽可能简化远程接口对象的使用。）

11. CRM

Customer Relationship Management 客户关系管理

企业利用相应的信息技术以及互联网技术来协调企业与顾客间在销售、营销和服务上的交互，从而提升其管理方式，向客户提供创新式的个性化的客户交互和服务的过程。其最终目标是吸引新客户、保留老客户以及将已有客户转为忠实客户。

12. ERP

全称：Enterprise Resource Planning 企业资源计划

定义：是指建立在信息技术基础上，以系统化的管理思想，为企业决策层及员工提供决策运行手段的管理平台。

13. DCOM

Distributed Component Object Model分布式组件对象模型

是一系列微软的概念和程序接口，利用这个接口，客户端程序对象能够请求来自网络中另一台计算机上的服务器程序对象。

（DCOM 基于 COM，COM 提供了一套允许同一台计算机上的客户端和服务器之间进行通信的接口，DCOM 支持不同的两台机器上的组件间的通信，而且不论它们是运行在局域网、广域网、还是 Internet 上。）

14. IDL

全称：Interface Definition language（接口定义语言）

定义：描述性语言，是 CORBA 规范的一部分，是跨平台开发的基础，提供一套通用的数据类型，用于描述接口，不定义实现，类似 C 中的头文件。

问答

1. 应用集成大致分为几种类型，分别解决什么样的问题？

（平台集成；数据集成；组件集成；应用集成；流程集成；B2B 集成）

1) 表示集成：软件用户界面。为原来基于终端的应用软件提供 PC 界面。提供一个由多组件合成的应用软件

（案例：1 为大型机提供 windows 界面；2 为 SAP R/3 与大型机程序提供统一的 HTML 界面；3 为多个大型机应用程序提供统一的基于 Java 的界面）

2) 数据集成：直接访问软件创建、维护并存储的信息。多个信息源综合数据进行分析和决策。向多个应用软件提供公共信息源的只读权限。以一个信息源的信息来更新另一个数据源。

（案例：1 综合 sybase、DB2 和 SAP P/3 数据库中的数据；2 使用大型机和 Oracle 的可执行信息系统；3 允许其他应用程序在 peoplesoft 和定制的 Oracle 数据库中获取数据）

3) 功能集成：代码级别的软件集成。能够解决前两种方法可解决的问题。要求新软件具有其他程序的功能。在集成中暗含工作流。确保应用间的事务完整性

（案例：1 获取用户信息，对 java 程序、大型机程序、Oracle 数据库作更新；2 把供应商的系统集成到采购系统中）

2. 什么是格式良好的 XML 文档？

格式良好的文档遵守 XML 语法，但没有 DTD 或模式。

格式良好的 XML 文档要满足六原则：

1) XML 文件的第一行必须是声明该文件是 XML 文件以及它所使用的 XML 规范版本。（在文件的前面不能够有其它元素或者注释。）
2) 在 XML 文件中有且只能够有一个根元素。
3) 在 XML 文件中的标记必须正确地关闭。（也就是说，在 XML 文件中，控制标记必须有与之对应的结束标记。）
4) 标记之间不得交叉。
5) 属性值必须要用引号括起来。
6) 控制标记、指令和属性名称等英文要区分大小写。

3. 设计模式 ppt5

设计模式帮助实现软件复用.可以作为应用集成的解决方案有：工厂方法模式、代理模式、适配器模式

设计模式：

一套被反复使用、多数人知晓的、经过分类编码的、代码设计经验的总结；可重用代码、让代码更容易被他人理解、保证代码可靠性。

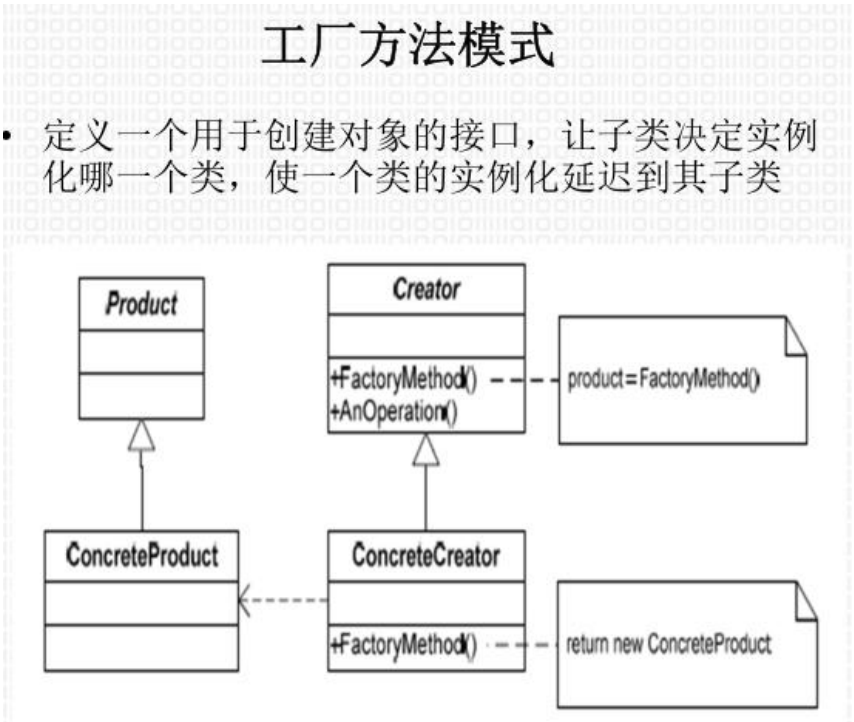
基本要素：模式名称、模式问题、解决方案、效果

好处：权衡已经被证实的解决方案；为交流提供一个共同的词汇表；约束解决方案的范围

分类：

		目的		
		创建型	结构型	行为型
范围	类	工厂方法（Factory Method）	适配器（Adapter）	解释器（Interpreter） 模板方法（Template Method）
	对象	抽象工厂（Abstract Factory） 生成器（Builder） 原型（Prototype） 单件（Singleton）	适配器（Adapter） 桥接（Bridge） 组成（Composite） 装饰（Decorator） 外观（Facade） 享元（Flyweight） 代理（Proxy）	职责链（Chain of Responsibility） 命 令（Command） 迭代器（Iterator） 中介者（Mediator） 备忘录（Memento） 观察者（Observer） 状 态（State） 策 略（Strategy） 访问者（Visitor）

- 工厂方法模式：Web Service 的异构系统之间的方法调用。



特点

良好的封装性，代码结构清晰

优秀的扩展性

屏蔽产品类

- 调用者只需要关心产品的接口，只要接口保持不变，系统中的上层模块就不用发生变化

典型的解耦框架

- 迪米特法则，不需要的就不要去交流
- 依赖倒置原则，只依赖产品类的抽象
- 里氏替换原则，可以使用产品子类替换产品父类

使用场景

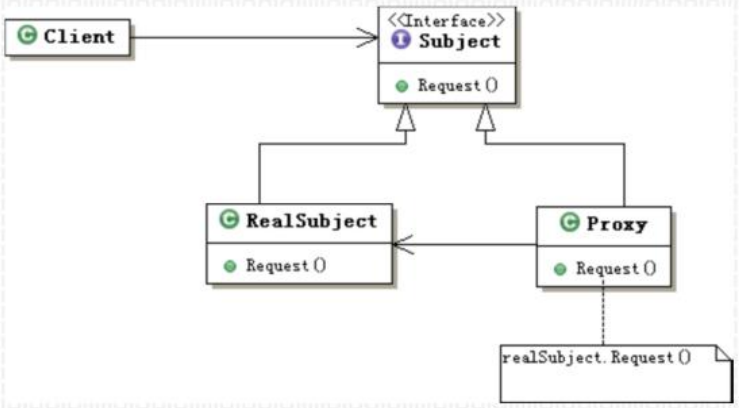
- 在所有需要生成对象的地方都可以使用，但是需要慎重地考虑是否要增加一个工厂类进行管理，增加代码的复杂度
- 需要灵活的，可扩展的框架时，可以考虑采用工厂方法模式
- 工厂方法模式可以用在异构项目中，如从WSDL中产生的对象都认为是一个产品，然后由一个具体工厂类进行管理，减少与外围系统的耦合
- 可以使用在测试驱动开发的框架下

➢ 代理模式：Spring 等动态代理。

代理模式

为其他对象提供一种代理以控制对这个对象的访问

许多其他的模式，如状态模式，策略模式，访问者模式本质上是在更特殊的场合采用了代理模式



优点

职责清晰

- 真实的角色就是实现实际的业务逻辑，不用关心其他非本职责的事务，通过后期的代理完成一件事务，附带的结果就是编程简洁清晰

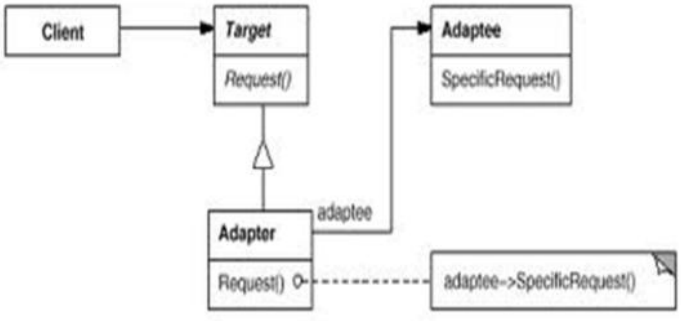
高扩展性

- 具体主题角色是随时都会发生变化的，只要它实现了接口，不管它如何变化，代理类都可以在不做任何修改的情况下使用

➢ 适配器模式：接口转换。第三方 API 集成；新旧系统集成。

适配器模式

- 将一个类的接口变换成客户端所期待的另一种接口，从而使原本因接口不匹配而无法在一起工作的两个类能够在一起工作



优点

- 适配器可以让两个没有任何关系的类在一起运行，只要适配器这个角色能够连接他们就行
- 增加了类的透明性。我们访问Target目标角色，但是具体的实现都委托给了源角色，而这些对高层次模块是透明的，它也不需要关心
- 提高了类的复用度。源角色在原有的系统中还是可以正常使用，而在目标角色中也可以充当新的角色
- 灵活性非常好。如果不想用这个适配器，直接删掉就可以了，其他代码都不用修改。想用就用，不想就卸载

使用场景

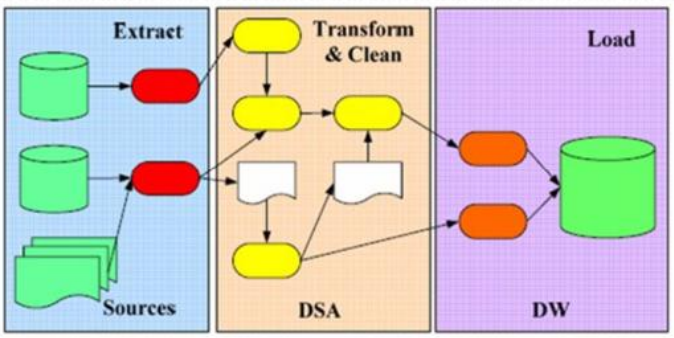
- 要调用的接口类型无法满足新系统的使用需求，将旧系统的接口通过适配器进行转配，达到支持新接口调用的目的
  - 在使用第三方的类库，或者说第三方的API的时候，如果接口不匹配，可以通过适配器转换来满足现有系统的使用需求
  - 当已经投入使用的旧系统与新系统进行集成的时候，如果发现旧系统的接口无法满足新系统的需求，可以通过适配器转换来满足新系统的使用需求，从而实现集成

4. ETL （定义、步骤）ppt3

ETL：Extraction-Transformation-Loading 数据提取、转换和加载

➢ 将分布的、异构数据源中的数据如关系数据、平面数据文件等，抽取到临时中间层后进行清洗、转换、集成，最层后进行清洗、转换、集成，最后加载到数据仓库或数据集市中，成为联机分析处理、数据挖掘的基础。

ETL的过程



- 抽取、转换和加工、装载
- 节点代表操作：过滤，转变，传输，压缩，加密等
- 增量、转换、调度和监控等处理

➢ 数据的抽取

• 全量抽取和增量抽取——触发器；时间戳；快照

➢ 数据的清洗

• 数据格式不一致、数据输入错误、数据不完整
• 源数据和目标数据需要进行数据模式或语义映射的转换
• 在数据库中进行数据加工

➢ 数据转换

• 不一致数据转换：
• 数据粒度的转换：
• 商务规则的计算：

➢ 数据装载

• 最佳方法取决于所执行操作的类型以及需要装入多少数据
- 直接 SQL 语句操作，进行了日志记录并且可恢复
- 采用批量装载方法，易于使用，并且在装入大量数据时效率较高

➢ ETL 的三种实现方法

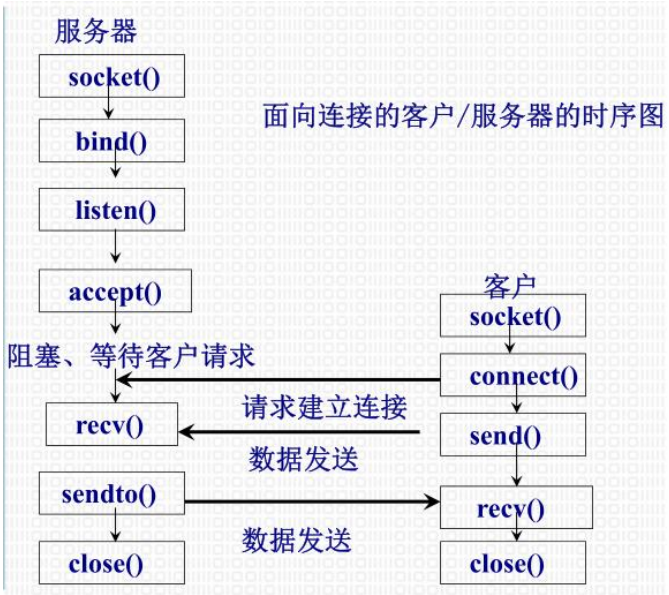
借助 ETL 工具—— SQL 方式实现——两种结合。

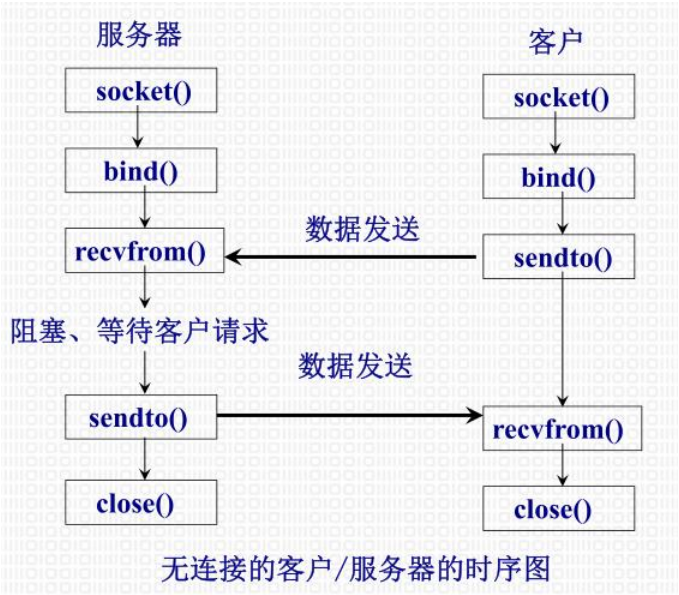
➢ 关键技术

• 增量复制（• 触发器• 时间戳• 快照方式• 日志法）

• 数据的清洗转换

5. socket 进行通讯的过程





网上：

socket 建立连接套接字；

bind 在套接字上绑定地址

connect 在套接字上建立连接

listen 监听套接字

accept 接受连接请求

6. 描述 ODBC 的基本结构和工作流（JDBC）

➢ 基本结构

(1)应用程序

应用程序嵌有的 SQL 语句在运行时被转换为若干个动态连接库中的 ODBC 函数。

(2)驱动程序管理器

负责管理和调度驱动程序。

(3)驱动程序

是相应于某个数据源的 ODBC 函数执行码，存放于动态连接库，提供给应用程序调用。一个 SQL/CLI 接口一般可连接若干个 DBMS，故有若干个驱动程序。

(4)数据源

提供的数据可以是 RDBMS，也可以是 OODBMS 或各类文件形式。

PPT3: ODBC (Open DataBase Connectivity)

• 应用程序

– 执行处理并调用 ODBC API 函数，以及提交 SQL 语句并检索结果

• 驱动程序管理器

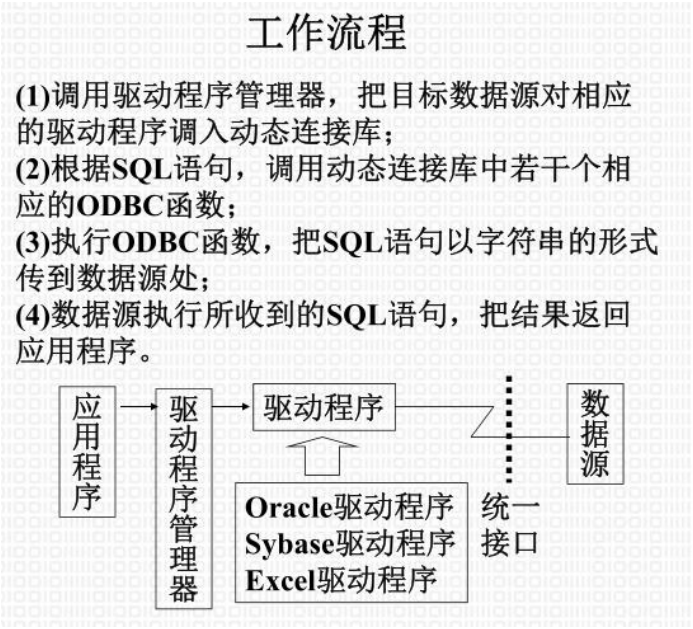
– 根据应用程序需要加载/卸载驱动程序，处理 ODBC 函数调用，或把他们传送到驱动程序

• 驱动程序

– 处理 ODBC 函数调用，提交 SQL 请求到一个指定的数据源，并把结果返回到应用程序

• 数据源

– 包含了数据库位置和数据库类型等信息，实际上是一种数据连接的抽象



7. XML Schema 与 XML DTD 的区别

XML Schema 是 XML 文档，遵循 XML 语法规则；而 DTD 不遵守 XML 语法，它有其特有的语法；

XML Schema 支持数据类型；而 DTD 数据类型有限（与数据库数据类型不一致）

XML Schema 是一个开放的模型

XML Schema 是可扩展的；而 DTD 不可扩展

XML Schema 支持命名空间；而 DTD 不支持，因为命名冲突

DTD 比 Schema 更加精炼，文档篇幅相同的情况下能表示更多内容

（DTD 是已经淘汰的技术，因为 DTD 不具有规范性并且没有数据类型。）

8. 元数据 ppt3

元数据：是描述、解释、定位一个信息资源的结构化的信息。元数经常被称作数据的数据。元数据是信息共享和交换的基础和前提，用于描述源数据集各种特征。

元数据主要有下列几个方面的作用：

1) 描述和发现资源； 2) 管理资源集合； 3) 保存数字化资源；4) 提供数据互操作和数据转换方面的信息。

元数据标准 ： OMG、CORBA 、UML、XML

PPT3:

➢ 概念：

• data about data，即关于数据的数据
• 通过一组属性或元素来描述特定的资源
• 元数据模型提供了描述一类资源的具体对象时所有规则的集合
  - 描述资源属性的术语：元数据元素
  - 关系、结构约束、语法表示等
• 提供规范、普遍的描述方法和检索工具，为分布的、由多种资源组成的信息体系提供整合的工具与纽带

➢ 作用：

• 需要一个中央集线器确保 Oracle 电子商务套件与 3 个老的 ERP 系统之间一致的数据映射
• 大量的企业数据要么深锁在数据库中，要么就被封闭在应用中
• 随着企业跨应用组合不同的功能，这些数据模型也被混合在一起
• 对数据而言，始终存在一种上下文关系。甚至当一个字段为空白时，不同应用会对它的含意做出不同的假设

➢ 分类：

• 从元数据的作用角度: - 描述型 - 结构型 - 管理型
• 从信息系统的角度: - 业务 - 技术 - 操作

9. 消息中间件的特点，两种模式

特点：

- 通信程序可在不同的时间运行
  - 程序不在网络上直接相互通话，而是间接地将消息放入消息队列，因为程序间没有直接的联系，所以它们不必同时运行。

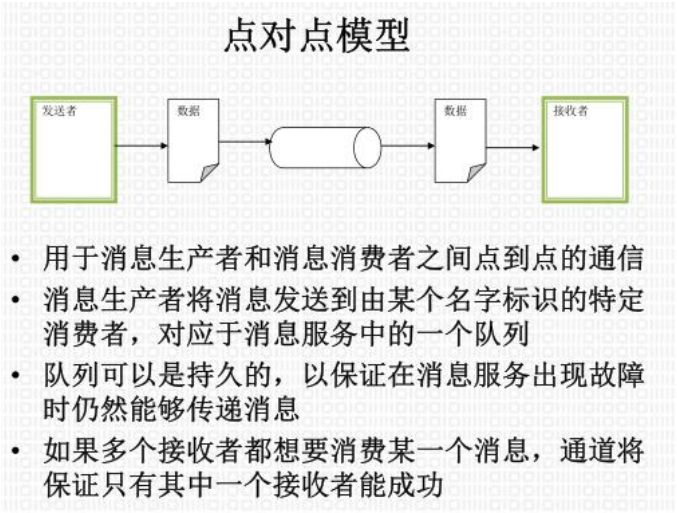
程序不在网络上直接相互通话，而是间接地将消息放入消息队列，因为程序间没有直接的联系，所以它们不必同时运行。

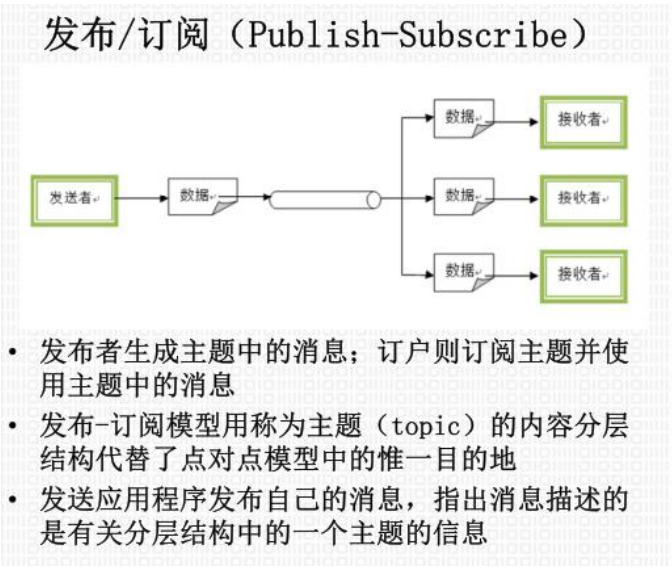
- 对应用程序的结构没有约束
  - 多种通信方式不会增加应用程序的复杂性
- 程序与网络复杂性相隔离
  - 程序将消息放入消息队列或从消息队列中取出消息来进行通信，与此关联的全部活动是 MOM 的任务，程序不直接与其他程序通信，

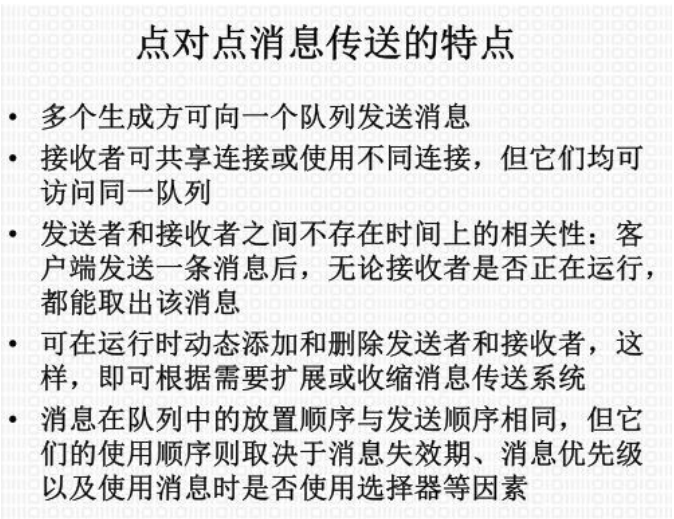
也不涉及网络通信的复杂性

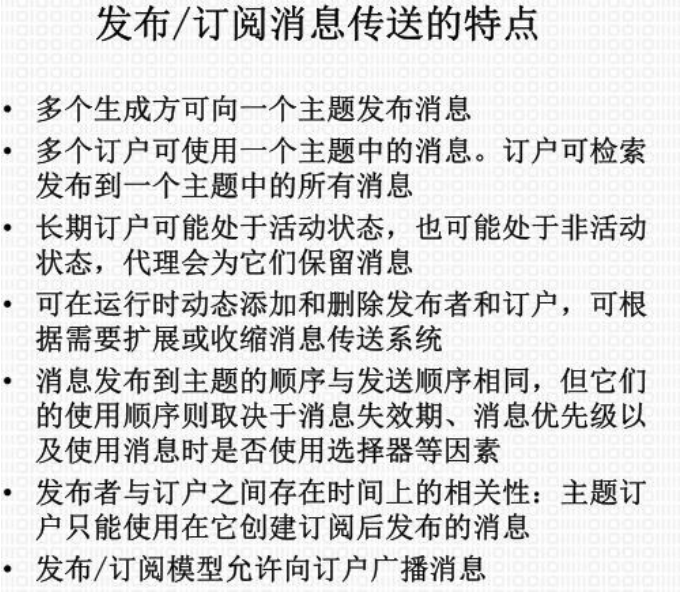
消息传递系统（6）:• 消息构造• 消息通道• 管道和过滤器• 消息路由• 消息转换器• 消息端点

消息模式（2）:









10. 解释 DOM 和 SAX 解析 XML 文件的不同之处？

1 DOM 是基于树形结构的 W3C 推荐的 API 标准； SAX 是事件驱动的有广泛支持的 API 标准

2 DOM 适合于结构化编辑 XML 文档,如排序,记录移动等；SAX 适合内存不足和文档结构无关的行为,如计算 XML 文档结点数,提取特定节点内容等

3 DOM 整体装入和处理 XML 文档,系统资源占用大,效率低,速度慢； SAX 是一种事件驱动接口,不需要把整个 XML 文档加载到内存,只需要处理关心的数据,对内存的占用不会随着文档的大小而有所变化,效率高,速度快

4 SAX 是一种快速而简单的接口,绝大多数的解析器可以由开发者自己编程实现.

另一个：

- SAX 是事件驱动的流式解析技术，在解析 XML 文件时，不需要把整个 XML 文档加载到内存，不会建立对象模型、不支持随机访问、不支持数据更新、接口易用性较差、解析效率中等；
- DOM 是基于树形结构的对象 XML 解析技术，解析时整体装入和处理 XML 文档,系统资源占用大,效率低,速度慢，支持建立对象模型、支持随机访问、支持数据更新、接口易用性中等、解析效率较差。