

摘 要

我国的 ERP 产业在经历了一段时间的快速扩张之后，发展速度明显放缓，这主要是由于 ERP 软件自身的功能局限性和 ERP 市场同质化现象严重造成的。为此，应该发展差异化的 ERP 产品同时提高产品的服务水平来帮助 ERP 厂商们摆脱这种困境。本文将 SOA 和智能客户端技术在 ERP 软件领域结合进行研究，试图通过技术创新来探索一种 ERP 软件的新模式。本文先介绍了智能客户端和 SOA 的基本原理、发展过程、应用现状、成功实例等内容，说明智能客户端和 SOA 技术理论上的先进性。之后，说明为什么 SOA 和智能客户端技术有助于提高 ERP 软件的质量。然后以 SOA 松耦合、可重用的指导思想为基础，将 ERP 的业务流程分解成服务，又将这些服务封装成 Web 服务，发布到网络上供客户端调用以完成一定的业务功能。在客户端的设计上，采用智能客户端技术，设计出了一个易于安装和部署，可自动更新，支持在线离线两种模式使用并且界面丰富的客户端应用程序。最后，对本文完成过程中遇到的问题进行探讨，并对本设计提出了几点可能的改进建议。

关键词： SOA， 智能客户端， Web 服务， ERP

Abstract

After rapidly expanded for more than 10 years, the speed of development on ERP Industry is getting slower. The main reasons are the limitation on ERP software's function and the similar on existing ERP products. So we should develop diversified ERP products and improve ERP service to help ERP manufacturers get rid of this situation. This research tries to seek a new ERP software mode by using SOA and Smart Client technologies together. This thesis introduces SOA and Smart Client technologies first, including basic principle, development process, successful examples and so on, in order to show their advances in theory and tell why SOA and Smart Client can improve ERP software's quality. Then, base on the SOA theory which emphasizes loose couple and reuse, a Smart Client of SOA ERP system is been analyzed, designed and partly implemented. The business process is decomposed into services, and these services are designed into Web services which can be issued to the Internet and wait clients to call. The client in this system is designed in Smart Client technology. It's easy to install and deployment, can update automatically, can use both online and offline and has a rich user interface. Finally, the author tells the problems encountered in the process of finishing this thesis, and gives some suggestions.

Keywords: SOA, Smart Client, Web Service, ERP

学位论文原创性声明

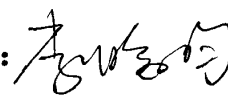
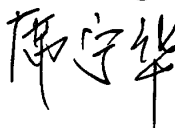
本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文所涉及的研究工作做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

特此声明

学位论文作者签名：李晗韵 2019年5月16日

学位论文版权使用授权书

本人完全了解对外经济贸易大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或部分的阅览服务；学校有权按照有关规定向国家有关部门或者机构送交论文；在以不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。保密的学位论文在解密后遵守此规定。

学位论文作者签名： 2009 年 5 月 26 日
导师签名： 2009 年 5 月 26 日

第 1 章 引言

本研究以利用技术创新提高 ERP (Enterprise Resource Planning, 企业资源规划) 产品质量为目的, 将 SOA (Service-Oriented Architecture, 面向服务架构) 和智能客户端技术在 ERP 软件领域结合应用, 完成一个具有智能客户端功能而且基于 SOA 的 ERP 中采购子系统的原型设计, 该设计克服了现有 ERP 软件灵活性差和不能与企业实际业务匹配的问题。本章简要介绍本文的写作目的、题目意义、主要成果等。

1.1 研究目的

本文研究将智能客户端技术和 SOA 引入 ERP, 从而提高 ERP 对客户多变需求的适应性, 降低开发与实施成本。在此基础上, 主要采用面向服务的思想对实现了智能客户端功能的 ERP 原型系统进行分析和设计, 并对该原型系统采购子系统的智能客户端部分功能进行了实现。

1.2 研究的意义

ERP (Enterprise Resource Planning, 企业资源规划) 在引入我国之初, 由于其先进的管理思想以及给企业带来的巨大预期效益, 使得在近十几年中 ERP 产业在我国得到了迅速的发展壮大。但是, 近年来, 这种发展有了明显减速的趋势, 一方面这是由于 ERP 软件自身性能上存在一定局限性, 比如不同行业企业的业务特点、软件基础设施等区别很大, 同一家企业的经营也在不断变化, 而多数 ERP 厂商的软件不能适应复杂多变的客户需求。如果不改革 ERP 软件模式, 使之能更好的适应企业灵活多变的业务, 势必将制约 ERP 产业的发展。另一方面, 现今 ERP 市场上产品同质化现象严重也是制约其发展的重要原因, 如果不能通过技术创新来发展差异化的产品, 那将严重影响各 ERP 厂商在市场上的竞争力。

近些年来, 随着 IT 技术的发展, 不同种类的操作系统、应用软件和应用基础结构间的相互交织越来越频繁, 这使得 IT 企业不得不通过改变自己的业务模式来适应复杂多变的外部环境。但是要从头建立一个新的基础环境是不可能的, 企业应该能对业务的变化做出快速的反应, 利用对现有的应用程序和应用基础结构的改造来解决新的业务需求, 为客户、商业伙伴以及供应商提供新的互动渠道。SOA (面向服务架构) 便是在这种背景下产生的, 凭借其松耦合的特点, SOA 架构使得企业可以按照模块化的方式来添加新的服务, 更新现有的服务, 并可以把企业现有的应用也作为服务加以利用, 从而呈现出一个既具灵活性又支持有机扩展的应用基础构架, 同时也保护了企业现有的 IT 基础建设投资。

目前较流行的软件客户端技术有胖客户端和瘦客户端两种，但是二者都有不可避免的局限性，而智能客户端集二者的优点于一身，恰好能解决它们的问题，提高用户满意度。因此，有专家指出，智能客户端是管理软件业发展的新方向，在未来的两至三年，智能客户端将会逐渐得到软件服务提供商与用户的认可，产业界也会掀起一股向智能客户端迁移的热潮。

SOA 与智能客户端，一个是 IT 界公认的企业级应用开发领域的新趋势，一个是 ERP 行业新的技术发展方向；一个以 IT 灵活性为核心思想，一个以提高客户服务质量为目的，二者的先进性和经济意义通过大量成功实施的案例就可证明。若将二者在 ERP 领域结合应用，不仅会促进 ERP 在企业的应用，也会有助于有关 ERP 的理论研究，因而具有较高的理论和实践价值，给 ERP 产业带来技术创新的希望。部分 ERP 管理软件厂商们也一致认为，智能客户端或将成为目前 ERP 产业技术创新的最佳切入点。因此，基于 SOA 的 ERP 系统的智能客户端研究，既是对 ERP 软件发展方向的新探索，又是对 SOA 方法应用可行性的论证与扩展。

1.3 研究内容

本文先介绍了 ERP 理论形成的基本历程，分析 ERP 在我国发展到现阶段遇到的困境，指出将 SOA 思想与智能客户端技术在 ERP 软件领域结合是帮助提高 ERP 产品灵活性，提升服务质量的出路。之后，又介绍了智能客户端和 SOA 的基本原理、发展过程、应用现状、成功实例等相关内容，说明智能客户端和 SOA 技术理论上的先进性。然后，以 SOA 的指导思想为基础，对实现了智能客户端的 ERP 原型系统的一个采购管理子系统进行面向服务的分析和设计，完成该子系统的需求分析、业务流程图、服务模型、SOA 体系结构、网络拓扑图等，最后在设计的指导下，选取几个典型的智能客户端功能模块进行编程实现。

第 2 章 基于 SOA 的 ERP 智能客户端产生背景

本章通过回顾 ERP 的历史指出基于 SOA 的 ERP 是 ERP 软件的发展趋势之一，通过对 SOA 和智能客户端相关概念的研究，分析了将 SOA 与智能客户端技术结合应用在 ERP 领域的原因和将会产生的意义。

2.1 ERP 发展历程回顾

为什么有必要研究 SOA 与智能客户端技术在 ERP 中的应用？其原因在于 ERP 的发展在目前遇到的困难。在这一节里，通过对 ERP 发展历程的回顾，论证了 SOA 与智能客户端技术作为 ERP 发展趋势的合理性。

2.1.1 ERP 的形成历程

从 20 世纪三、四十年代提出订货点法至今，ERP 理论的形成与发展大致经历了 5 个阶段：订货点法——时段式 MRP（Material Requirements Planning, 物料需求计划）——闭环式 MRP——MRP-II（Manufacturing Resources Planning, 制造资源计划）——ERP（Enterprise Resource Planning, 企业资源规划）

1. 第一阶段：订货点法。

订货点法（Order Point Method），是一种使库存量不低于安全库存的库存补充方法。基本思想是：对于某种物料或产品，当库存量降低到某一预先设定的点时，即开始发出订货单(采购单或加工单)来补充库存，到库存量降低到安全库存时，发出的订单刚好到达仓库，补充前一时期的消耗，此订货的时点即为订货点。在订货点，剩余库存减去安全库存后可供消耗的时间刚好等于订货需要的时间。

订货点法本身具有一定的局限性：一、订货点法能够为企业正常生产提供足够的原料，但有时会造成库存积压现象。比如，某物料虽然降低到了订货点，但可能在近一段时间企业没有收到新的订单，对这种物料没有新的消耗，但由于在订货点补了货，所以会造成库存积压和资金占用。二、订货点法将各种物料的需求看成是相对独立，物料需求是连续发生的并假设提前期是已知的和固定的等。三、无法很好的解决何时定货的问题。

另外，订货点法只是针对库存管理的，只涉及到企业单个部门的业务目标。

2. 第二阶段：时段式物料需求计划。

在订货点法的基础上，1957 年美国生产与库存控制协会推出了第一套物料需求计划（MRP, Material Requirements Planning）软件，较好的解决了何时订货的问题，这是现代企业管理软件的开端。

这个阶段的 MRP 被称为时段式 MRP 或基本 MRP，它能根据有关数据（主生产计划、物料清单数据和库存数据）计算出相关物料需求的准确时间与数量，并按时间段下达补充材料订单，是一种按零件提前期组织生产或者采购的基本的计划技术，也是一种保证订单按期交货的有效方法。

时段式 MRP 的设计缺陷是没有考虑到生产企业现有的生产能力和采购的有关条件的约束。因此，计算出来的物料需求的日期有可能因设备和工时的不足而没有能力生产，或者因原料的不足而无法生产。同时，它也缺乏根据计划实施情况的反馈信息对计划进行调整的功能。

需要强调的是，和订货点法比较，时段式 MRP 在功能和管理涵盖范围上都有了扩充，在功能上，把库存管理和企业的主生产计划和物料需求联系在了一起，在系统覆盖的业务部门方面，也从仅仅局限在库存扩展到包括了生产部门的需求和采购部门的计划。可见，时段式 MRP 虽然在功能和管理范围方面有所扩展，与生产与采购有了联系，但是并没有深入到生产与采购的内部，其功能和管理范围的扩展还有限。

3. 第三阶段：闭环式物料需求计划。

为了解决时段式 MRP 的问题，20 世纪 70 年代末，出现了闭环式 MRP 系统。闭环式 MRP 系统除了物料需求计划外，还将生产能力需求计划（CRP）、车间作业计划和采购作业计划也全部纳入 MRP 范畴一并考虑，形成了一个完整的生产计划与控制系统。

闭环式 MRP 的主要特点是：一、以整体生产计划为基础，在计划的各个阶段均包含了能力需求计划，这样使物料需求计划的可行性提高。二、有反馈机制，能力的执行情况最终都反馈到计划制定层。

闭环式 MRP 系统在制定物料需求计划时，其视野不是只局限于库存以及库存和生产、采购之间的外部联系，还深入到生产和采购部门的内部，将生产和采购部门的内部计划都包括进来。具体地说，就是它还包括了生产部门的生产能力需求计划（CRP）、车间作业计划，以及采购部门的采购作业计划。这样，库存、生产和采购都被纳入 MRP 范畴一并考虑，形成了一个完整的生产计划与控制系统。但闭环式 MRP 还是存在一定的局限性，因为它只考虑了生产管理过程中的物流方面，而忽略了其他方面。

从订货点法、时段式物料需求计划到闭环式物料需求计划，在这三个阶段的发展过程中，所包括的业务功能从单纯的库存管理发展到既包括库存又包括生产和采购，所涵盖的管理范围从只局限在一个部门到跨越三个部门，可见，ERP 软件的发展趋势是处理的业务越来越复杂，涉及的部门越来越多。

4. 第四阶段：制造资源计划。

闭环 MRP 运行过程主要是物流过程，但实际生产过程中，从原料加工到成品产出过程都不可避免伴随着企业资金的流通，而且资金的流通情况会影响到生产的运作，对这一点闭环 MRP 就没有反应出来。为解决这个问题，在 80 年代制造资源计划（Manufacturing Resource Planning）这个概念被提出，由于缩写都是 MRP，为了与物料资源计划区分，人们通常将它称为 MRPII。MRPII 的基本思想：是把企业作为一个有机整体，从整体最优的角度出发，通过运用科学方法对企业各种制造资源和产、供、销、财各个环节进行有效地计划、组织和控制，使他们得以协调发展，并充分地发挥作用。

MRPII 在业务功能上包括了生产、财务、销售、工程、采购等各个部分，所涵盖的管理范围也覆盖了企业的大多数部门，最重要的是它从单纯的物流管理扩展到了资金流管理范畴。

虽然 MRPII 系统综合了企业多数部门的业务目标，并通过计划的及时滚动来控制整个生产过程，但它的实时性较差，一般只能实现事中控制，MRP II 主要是面向企业内部资源的，而且各个子系统间缺乏联系。

5. 第五阶段：企业资源计划。

随着市场竞争日趋激烈、企业管理模式的不断创新以及科学技术的不断进步，对企业的信息化有了更高层次的要求，不仅是要对企业的制造资源进行整合，而是要对企业的所有资源进行管理。而且，企业信息系统所包括的信息也不能仅局限在企业内部，必须扩展到包括企业外部的客户、供应商、分销商等在内的整个供应链上。在这种背景下，ERP 诞生了，它除包括 MRP II 的基础模块外，还大大扩展了管理范围。

ERP 系统集中信息技术与先进的管理思想于一身，合理调配企业资源，降低生产成本，能给企业带来巨大的效益。真正实现了企业对企业的资源——物流资源、资金流资源和信息流资源进行全面的集成管理。

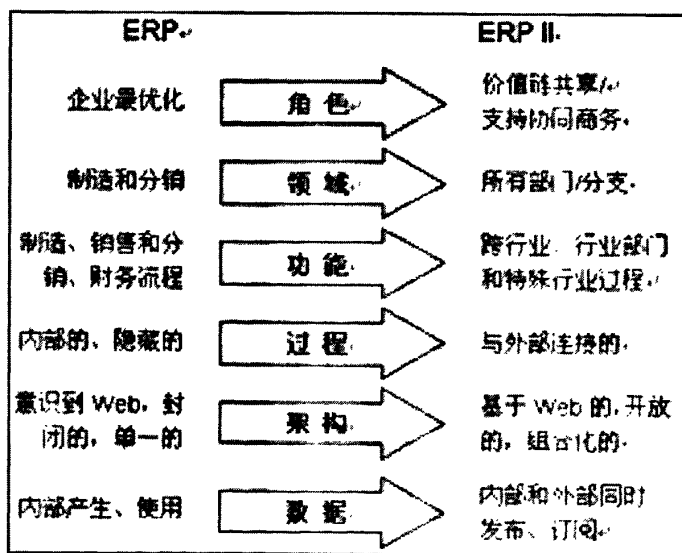
虽然 ERP 在现代企业管理中已经得到了广泛的应用，但是从当前的理论研究和应用实践来看，对这个概念至今没有一个统一的定义，各大厂商和行业协会对 ERP 的定义大致可以分为两种，一是把 ERP 视为一种先进的管理思想，如美国生产与库存管理协会的定义：ERP 是一种能有效地对制造、分销和服务中的客户订单所需要的资源，包括获取、加工、传输及会计处理等，进行计划和管理的方法。一是从软件角度定义 ERP，如美国 Gartner Group 公司的定义：ERP 这个概念描述了下一代制造商业系统和制造资源计划（MRP II）软件。它将包含 C/S 架构、图形用户接口的使用，以及开放系统的制作。除了已有的标准功能之外，它还包括品质、过程运作管理，以及调整报告等。另外，ERP 所采用的基础技术，将使用户的软件、硬件具有独立性，从而更易于升级。ERP

的关键在于有用户能够定制其应用，因而具有天然的易用性¹。

自从 1990 年 Gartner 公司提出了上述 ERP 的概念之后，ERP 系统基本上都是依照这个概念进行构建的。然而，纵观 ERP 的发展史不难看出，ERP 的每一个新的发展都使其功能涵盖的范围进一步扩大，参与的实体也逐渐增多。时至今日，随着 ERP 应用领域的扩大，各行各业对 ERP 的需求都有所增加，由于行业、类型和规模的不同，企业间业务差别加大，不同企业的 IT 应用环境也有很大差异。同时，随着全球经济一体化进程的不断深入，企业的生存环境发生了深刻的变化，产业上下游企业之间的关系由竞争转向合作，因而对加强协作的要求更高。可以说，如今的 ERP 面临着客户的需求多样和多变的新特点。

传统的 ERP 软件在面对需求和应用环境的改变的时候，暴露出了一些不足，可以说，无论是从 ERP 系统的管理思想和管理内容来看，还是从 ERP 系统的实施过程和实际应用角度观察，ERP 系统都还需要进一步地发展和完善以适应企业经济环境的变化。当前，关于 ERP 系统的未来有着各种各样的观点，其中 Gartner 公司的 ERP II 系统和 Charles Moller 的 ERP II 概念框架是两种最主要也是最典型的观点。

前面已经提过，ERP 的概念就是由 Gartner 公司在 1990 年率先提出的，10 年之后，Gartner 公司又率先提出了 ERP II 这个新的概念，试图重新定义企业资源计划的应用领域。Gartner 公司的 ERP II 是指通过支持、整合和优化针对具体行业的公司内部和公司之间的协作、运作和财务过程以创造客户和股东价值的一种商务发展战略和一套软件应用系统。ERP II 涉及到业务、应用和技术战略方面的 6 项要素：角色、领域、功能、过程、构架、数据，在这六个方面 ERP II 都有别于传统 ERP 系统。



¹ 程控，, 革扬: MRPII/ERP 原理与应用, 清华大学出版社, 2002。

图 2.1 ERP 与 ERP II

资料来源：《ERP 系统原理与实施》²

从图 1.1 的比较中可以看到，ERP II 系统从优化企业资源向价值链共享和协同服务发展；它不仅仅是支持制造和分销行业，而是向所有的行业扩展；它从单个企业应用向行业的信息共享扩展；它处理的业务流程已经从内部的扩展到了外部的，它是完全 Web 化的开放的系统。

与 Gartner 公司提出的 ERP 系统思想不完全相同，ERP 系统专家 Charles Moller 提出了一种 ERP II 概念框架，这种框架明确了 ERP II 的内容和与其他信息系统之间的关系，并且给出了 ERP II 的 4 个层次，即基础层、流程层、分析层和协同层。基础层是 ERP II 的核心组件和基础结构，包括集成的数据库和应用框架。数据库不一定是唯一的。流程层是 ERP II 的中心，是基于事务的系统。ERP II 是基于 Web 的、开放的和组件化的。传统 ERP 是整个 ERP II 概念框架的中心，除了包括传统的财务、销售、物流、制造、人力资源等功能模块之外，质量管理、项目管理、维修管理等也是 ERP II 的功能模块。需要强调的是，ERP 是基于最佳业务流程参考模型的，而 ERP II 是基于业务流程管理的，这使得 ERP II 更加灵活。分析层是对 ERP 功能的增强和扩展，包括 SCM（供应链管理）、CRM（客户关系管理）、CPM（企业绩效管理）等。

观察上面两种 ERP II 的观点，我们不难发现，它们都不约而同地提到了未来的 ERP 软件应该具有的特点：基于 Web，开放的，组件（组合）化的，基于业务流程，功能全面集成，具有灵活性等。

2.1.2 我国 ERP 发展概况

20 世纪 80 年代，我国沈阳机床厂与德国合作开发 ERP 系统，开创了我国企业引进并实施 ERP 的先河。在经历了十年左右的起步阶段之后，到 1998 年，国内开始大力宣传 ERP，新型的咨询公司数量也快速上升，有能力的软件厂家开始研发自己的 ERP 软件，企业也纷纷计划启动 ERP 项目，政府也给予了正面的宣传和指导。但是，在这个阶段，由于企业规模和预算投入所限，国内上 ERP 系统的多是财大气粗大型国有企业和中外合资企业。

在接下来的三、五年时间里，ERP 进入了发展的“黄金时间”，ERP 市场迅速壮大起来，一方面致力于提供 ERP 产品与服务的供应商数量暴增，一方面客户的需求也明显增多，再加上技术的发展带来的成本降低，使得 ERP 已不再是大企业的专属品，越来越多的中小企业、民营企业也上线了自己的 ERP 系统，ERP 普及率在工业尤其是制造业企业中有了明显的提高。但就发展阶段来看，

² 图片来源：闪四清，ERP 系统原理与实施，清华大学出版社

ERP 在我国的发展尚处于初级阶段，这个阶段虽然市场上的供求量都很庞大，但是市场并不成熟，供应商多而不精。目前国内、国外的 ERP 软件商多达几百家，其中不乏鱼目混珠者，单咨询公司就多达几百家。面对如此混乱的市场，客户根本就无从选择。

果不其然，在经历了一段时间的风光之后，如今的 ERP 产业的发展速度明显放缓。出现这种情况，从用户的角度来说，是由于现今的 ERP 产品功能不足、种类单一导致用户对实施 ERP 的效果产生了质疑造成的。据调查，ERP 用户对现有管理软件最不满意的三个因素分别是：不适应变化、灵活性差和不满足企业实际情况³。从 ERP 厂商的角度来说，是由于 ERP 产业的同质化现象越来越严重造成的，在形成了这样的同质化竞争局面之后，由于产品功能相似，所以只能从功能以外寻求差异，结果导致了恶性竞争。其直接后果是，ERP 厂商们不得不通过降价来吸引用户，厂商们逐渐陷入了营业额在上涨，但利润率却在下滑的尴尬局面当中。这种现象在市场走向成熟的过程中是不可避免的，当 ERP 市场成熟起来之后，厂商经过前一阶段的竞争，适者生存，自然的分出了伯仲，更重要的是各厂商的定位会更加准确，产品的特征被市场认可，不仅自然形成多层次的局面，而且还真正形成“差异化”的格局，客户能够根据各种产品的特征和自己的需求很快圈定选择范围，不像现在这样无所适从。

可以说，这种现象是在 ERP 市场走向成熟的过程中不可避免的，ERP 发展的趋势就是包含的业务功能越来越多，覆盖的管理范围越来越宽，支持的业务面广了，业务之间的差别必然会越来越大；应用范围大了，用户多了，不同用户的计算机硬件设备、软件平台等的差异也就越来越明显。因此，ERP 厂商产品的单一与用户需求的多样性和多变性的矛盾也越来越突出，各 ERP 厂商亟需通过技术创新来从根本上改变 ERP 产品的同质化局面。

2.1.3 ERP 发展的新方向

对于现阶段 ERP 所遇的困境，各大 ERP 厂商都已有了清醒的认识，因此大家都在积极地尝试走技术创新之路。总结现今各大公司对 ERP 软件新技术方向的探索，比较热门的大概有这么两个：

1. 变 C/S 结构为 B/S 结构的 SaaS ERP。SaaS (Software-as-a-service) 即软件作为服务，是一种基于 B/S 结构的软件服务提供模式，它让用户可以通过互联网使用实时运行的软件，而省去了安装客户端软件的麻烦，并且它没有大笔的软件授权费，通常是按月收使用费。客户几乎不需要除了个人 PC 和互联网连接之外的其它任何 IT 投资。通过互联网，多客户共享一套服务器软硬件设备。

³李莹等：基于 .Net 的多模 ERP 技术架构，计算机工程与应用，2005 年 8 月

它具有很多优势，如成本低、部署迅速、定价灵活等。

这是一种针对系统结构的改变，变原 C/S 结构为 B/S 结构。国外的一些厂商如微软、甲骨文和 SAP 等，在出售传统软件授权外，也实验性地为客户提供 SaaS ERP 服务作为一种附加的软件交付方式。对于一些急于上马 ERP 的新公司、或现金流有限但需要 ERP 的公司，这种模式在初期不失为一种不错的选择。国内的厂商也有过类似的尝试，如速达软件公司推出的“每天租金仅 1 元”的速达在线 ERP 就是个典型的例子。但有业内人士指出，目前的“在线 ERP”只能算是基于原有软件产品基础上的配套增值服务而已，还不具备单独形成市场的条件。可见，传统 C/S 结构的 ERP 确实有它自身的局限性，因此才会有向 B/S 结构转变的探索，但 B/S 结构似乎又不足以完全满足用户对 ERP 软件的要求。在软件体系结构这个方面，要是能采用集 C/S、B/S 结构优点于一身的智能客户端技术，也许能为 ERP 产品的创新找到一个新的方向。

2. 变面向对象为面向服务 (SOA)。传统的 ERP 各业务模块之间往往是复杂交错的关系，某块业务功能升级将引发整个 ERP 产品的升级，对于业务和信息部门都带来很大的压力。某一业务模块的调整将对整个公司或集团的信息处理工作带来影响，比如因为升级需要频繁的停机安装和备份，甚至带来相关模块操作界面和流程的改变。在 SOA 出现之后，各 ERP 厂商都在 SOA 的技术规范下建立了自己的 SOA 开发环境，如 SAP 的 NetWeaver 平台，Oracle 的 Fusion 架构，金蝶的 BOS 方案。通过 SOA 开发平台的迁移，ERP 厂商们逐渐对原来紧密结合的业务模块进行重新定义，作为单独的产品来发布和升级。

这是针对系统开发方法的变革，变 OOA (面向对象架构) 为 SOA，由于 SOA 自身的诸多优点，使得 SOA ERP 在功能上更全面，结构上更灵活，由于是基于开放的标准，所以其在扩展性、重用性以及与其他系统的集成方面也有了较大提高。SOA ERP 对于用户来说，可以根据不同用户的不同需求进行重组，最大限度地满足客户多样的需求，同时其组件化的内部结构对客户业务的灵活多变也能给予很好的支持；对于 ERP 软件厂商来说，SOA ERP 使它们可以基于自身的优势和定位拓展现有 ERP 的功能，发展差异化的产品，从而提高竞争力。回想一下 2.2.1 节提到的未来 ERP 软件应具有的特点，其中开放的，组件化的，灵活的，功能集成这些都与 SOA 的核心思想不谋而合，因此，SOA ERP 必将成为 ERP 软件发展的一个重要方向。

2.2 智能客户端简介

目前较为常见的胖客户端和瘦客户端程序有哪些不足？为什么说智能客户端集二者的优点于一身？本节通过介绍智能客户端的概念及与以往技术的比

较，分析了智能客户端的技术优势，并通过一个智能客户端的应用实例证明了智能客户端的经济性。

2.2.1 胖客户端概念

胖客户端是基于 C/S (Client/Server, 客户端/服务器) 结构的软件模式。C/S 结构通常由两部分构成：运行服务器程序的机器称为服务器，运行应用程序的机器称为客户端，客户端和服务器通过网络通信。这种结构的业务逻辑集中在客户端，客户端需要通过安装程序、保存大量的代码来处理业务逻辑，因此，称为“胖”客户端。当客户端接受用户指令之后，向服务器端发出请求，由服务器端完成数据的处理并响应客户端的请求。将数据提交客户端，客户端再将得到的数据进行一定的处理之后显示给用户。

由于胖客户端的程序和数据都存储在本地，业务处理也是合理地分配于客户端和服务器之上，而不是全部集中于服务器端，因此服务器端运行负荷较轻，系统通讯量也较小。而且，胖客户端程序在本地有独立的应用程序界面，用户甚至还可以根据自己的喜好调整软件的布局，进行丰富的界面元素的设置。另外，用户还能获得较快的反应速度，程序可以充分利用本地机器的资源，在不使用网络访问远程资源时，本地资源的访问在正常情况下都能得到很快的处理。

但是，在胖客户端进行部署时，由于客户端的类型各异，所以要针对每台机器进行必要的设置，包括安装单独的程序和配置数据库等，工作量大，部署起来比较困难。升级也是如此：使用传统的 DLL (动态链接库) 方式共用系统文件的 Windows 系统及应用软件，在对软件的版本进行升级时，由于需要改写共享的 DLL，所以可能使该机器上的其他软件因此不能运行。而且，对于不同的操作系统可能需要开发不同版本的软件，开发成本高，也不利于升级维护。另外，由于需要在客户端实现数据库支持，因此牵扯到本地数据的安全性问题，存在客户端非法访问数据库的风险。

2.2.2 瘦胖客户端概念

瘦客户端是基于 B/S (Browser/Server, 浏览器/服务器) 结构的软件模式，是 Internet 技术兴起后对 C/S 结构的一种改进。在这种结构下，用户界面是通过 www 浏览器来实现的，极少部分事务逻辑在客户的机器上实现，主要事务逻辑都在服务器端实现，这样就大大简化了客户端电脑载荷，因此，称为“瘦”客户端。在客户端不用安装专门的软件，只要一个浏览器即可。它能让不同的人员，从不同的地点，以不同的接入方式访问和操作共同的数据库。

瘦客户端由于程序和数据都集中在服务器端，业务逻辑的处理也几乎全部是由服务器端完成，客户端应用界面也是由服务端的文件生成，因此在客户端上占用资源少、配置简单，只需一个浏览器软件和可用的网络浏览器便能开始工作。对于系统的升级修改，也是只需在服务器端更新文件，部署和升级实现起来都比较简单。而且，即使是大量客户连接并使用系统，也不会因客户增加而加大管理难度，因为几乎不用在客户端安装单独的软件或备份数据。另外，客户端浏览器的页面都是基于开放的技术基础的（http、html 等），这样便提供了异种机、异种网、异种应用服务的联机、联网、统一服务的便利。

但是，瘦客户端完全依赖网络，当网络不可用或性能不稳定时，就会导致客户端失效，一切的工作将要在网络恢复后才能得以继续。

2.2.3 智能客户端概念

智能客户端（Smart Client）的定义：智能客户端应用程序是瘦客户端应用程序的强大替代产品。它可以为用户提供内容丰富且响应迅速的用户界面，提供脱机工作的能力，并且提供利用本地硬件和软件资源的方法。此外，可以将它设计为在各种各样的客户端设备上运行，包括桌面 PC、平板 PC 以及手持移动设备。智能客户端为用户提供了在强大且直观的客户端环境中访问信息和远程服务的能力，并且是一种用来开发灵活的、面向用户的应用程序以及提高用户工作效率和满意度的有效的解决方案。它结合了胖客户端和瘦客户端二者的长处，是下一代的客户端软件技术⁴。

智能客户端应用程序可以将胖客户端应用程序的响应快和体验丰富与瘦客户端应用程序的易部署和可管理性优点结合起来，也就是说在这两种方法之间取到了一个平衡，但这个平衡点的位置要根据具体的情况而定。胖客户端的程序和数据多集中在客户端，与之相反，瘦客户端的程序和数据则主要集中在服务器端，正是由于这种对立，使得胖、瘦客户端的优缺点成了互补状态，智能客户端的目标就是要将程序和数据在客户和服务器这两个端点之间合理分配：对于通用的、较固定的程序模块，保存在客户端，可以单独运行；对于新开发、有变化的程序模块，通过服务器端发布在 Internet 上供客户端调用。对于数据，也不是只存储在服务器端，客户端可以缓存备份，这样程序的运行就不用完全依赖于网络，在离线状态下也可以持续工作了。因此，智能客户端也就具备了胖客户端和瘦客户端二者的优点。

由于业务需求、开发环境、系统目标等的不同，对智能客户端的设计往往也有着不同的要求，因此在设计和实现方面会有极大的差异。有些智能客户端，

⁴ 微软 MSND Library, <http://msdn.microsoft.com/zh-cn/library/ms998468.asp>, 2009 年 2 月 20 日下载

在功能和使用方面可能更偏向于胖客户端，大部分程序和数据都配置在客户端，客户端大部分时间都是在本地运行，只是偶尔需要联机获取更新及扩充模块，这样的智能客户端在设计上就会更像胖客户端一些。而另一些智能客户端，由于本身所处理的业务的灵活性非常强，业务流程经常调整，业务范围变动也比较大，这样就需要频繁的更新或扩充系统的各功能模块，因此，在设计上可能就会更偏重于将数据和业务逻辑配置在服务器端，以便更新升级起来更容易，但同时也要保证在偶尔的离线状态下程序的可用性。无论怎样，所有智能客户端都具有下列部分或全部特征：

1. 更合理地利用本地资源。

设计良好的智能客户端应用程序最大限度地利用了代码和数据部署在客户端上并且在本地执行和访问这一事实。它为应用程序提供了内容丰富且响应迅速的用户界面，以及强大的客户端处理能力。例如，它可能使用户能够执行复杂的数据操作、可视化、搜索或排序操作。

智能客户端可以利用客户端硬件资源（如电话或条码读取器）以及其他软件和应用程序。这使它们非常适合于解决瘦客户端应用程序（如销售点终端应用程序）无法很好解决的问题。智能客户端还可以利用本地软件（如 Microsoft Office 应用程序）或客户计算机上安装的任何应用程序。通过创建能够与多个应用程序集成并对这些应用程序进行协调的解决方案，使用户可以更为有效地工作，进行更好的决策，并减少数据输入错误。此类解决方案还可以使用户的应用程序更加紧密地与用户的工作环境集成（例如，采用自定义的或熟悉的用户界面），从而降低培训成本。

2. 更合理地利用网络资源。

智能客户端可以通过网络消耗和使用不同的服务和数据。智能客户端应用程序应该尽可能地连接，并且应该利用可以通过网络使用的资源和服务。它们不应该是独立的应用程序，而应该是能构成更大的分布式解决方案的一部分。智能客户端应用程序起码应该使用有助于维护该应用程序以及提供部署和更新服务的集中式服务。

智能客户端应用程序的连接性质使其可以提供有价值的整合、分析和转换服务。它们使用户可以实时地或者在一段时间内协作完成任务。在许多情况下，智能客户端应用程序可以向用户提供类似于门户的功能，从而将完全不同的数据和服务加以协调并集成到总体解决方案中。

3. 更经济地在线和离线工作。

可以将智能客户端设计为向偶尔连接到网络的用户提供功能，从而使用户可以在明确脱机、使用低带宽或高延迟网络，或者连接时断时续的情况下继续

高效地工作。对于移动应用程序，智能客户端还可以优化网络带宽。例如，通过将请求批量发送到服务器，以便更好地使用代价昂贵的连接。

即使当客户端大多数时间都连接到网络时，智能客户端应用程序也可通过以智能方式缓存数据和管理连接来改善性能和可用性。例如，在低带宽或高延迟环境中，智能客户端应用程序可以用特殊的方式管理连接，不损害应用程序的可用性和响应性，使用户可以继续高效地工作。

通过使用户能够在断开连接或只偶尔连接的情况下工作，提高了用户的工作效率和满意度。

4. 提供智能安装和更新。

传统胖客户端的最大的问题多发生在部署或更新应用程序的时候。许多胖客户端应用程序具有大量复杂的安装要求，而且由于是通过注册组件或者在公共位置安装 DLL（动态链接库）来共享代码，所以可能导致应用程序脆弱和更新困难。

通过对智能客户端应用程序的设计，可以使其按照远比传统胖客户端应用程序更智能和灵活的方式来部署和更新，因此避免了上述常见问题，从而有助于减少应用程序的管理成本。

5. 提供客户端设备灵活性。

智能客户端还可以提供灵活且可自定义的客户端环境，从而使用户可以将应用程序配置为支持用户喜欢的工作方式。智能客户端应用程序没有被限制到桌面计算机或膝上型计算机。随着小规模设备的连接性和能力的增加，愈发需要能够提供对多个设备上重要数据和服务的访问的有用客户端应用程序。

智能客户端应用程序能够感知目标设备的功能和使用环境（无论它是桌面、膝上型、平板还是移动设备），以及能够定制应用程序以提供最适当的功能。

值得注意的是，一个智能客户端应用程序不一定要具有上述所有特征，当设计智能客户端时，需要仔细考虑所支持的业务目标，以便决定智能客户端应用程序到底要具备上述哪些特征，到底要在胖客户端和瘦客户端之间取怎样的一个平衡点。

2.2.4 智能客户端的应用实例

在2007年的一项关于煤电厂生产效益决策分析系统研究的项目中，一套基于智能客户端的煤电生产企业ERP系统在河南平顶山市的煤电企业得到了应用，通过对煤电企业生产运行的仿真，在实现系统标准化建模的基础上，运用智能客户端技术，结合面向服务的策略，实现了煤电生产企业ERP系统的设计。这套ERP系统在服务端和客户端分别用一个XML格式文件申明服务器上可用的版本及当前

客户端使用的版本。当程序更新时，在服务器端部署新版本并修改xml文件中声明的版本信息；在客户端，应用程序在启动时能够自动开启一个线程，在后台核对客户端和服务端两个文件声明的版本号。如果客户端版本不低于服务器端，那么线程就睡眠；否则就把服务器端的新程序下载到本地，完全实现了自动更新。同时，该系统在应用程序与网络数据和服务的交互方式上采用了异步通讯机制。为了确保系统在离线时能够提供用户继续工作所需要的所有数据和功能，系统在本地可以缓存必要的数据库。在联机时，这些缓存数据与服务器同步刷新，保证了数据的正确性和有效性。在用户离线时，其已更改的任何数据都没有被确认为服务器上的更改，只有在这些数据和服务器上的数据进行整合并且没有任何冲突后，才会将这些数据视为确认数据。对数据同步过程中可能出现的数据和业务规则冲突问题，该系统采取返回给客户详细的冲突信息，并提供解决方案供用户选择，实现了强大的离线操作功能。在安全性的设计上，该系统既考虑了服务器端的安全性，又考虑了客户端的安全。在用户与应用程序交互的过程中，首先采用身份验证方法来确认合法用户。在对用户进行身份验证后，通过授权来确定他们有权访问系统内的那些资源。在本地缓存数据时，采用加密算法来保护敏感数据。客户端的安全通过代码访问安全模型来实现，即满足了客户端“智能”的要求又注重了安全性。这套智能客户端ERP系统的成功设计，为企业创造经济效益80万元，为企业ERP提供了切实可行的运行模式⁵。

2.3 SOA 架构简介

SOA 是近些年来十分热门的一个概念，在本节，对 SOA 的基本概念做了介绍，透过这些概念，SOA 的优越性得到了明显的体现。基于 SOA 的系统分析和设计方法是一种综合的软件设计思想，它与以往方法论的关系在本节中也有详细的说明。

2.3.1 SOA 的基本概念

在当今商场上，企业面临着比以往任何时候都激烈的竞争，再加上客户需求的不确定性，使得业务“灵活性”成为了企业赖以生存的一条根本法则。因此，企业对信息系统的要求也有了改变，企业希望能从 IT 系统中获得对快速反应和按需应变业务的支持。把企业的业务通过内、外部的流程、人员、信息串联起来，灵活迅速地响应变化，是企业信息化的关键目标，而面向服务架构（Service-Oriented Architecture, SOA）正是帮助企业实现这一目标的 IT 架构和方法。

⁵ 刘建粉，吕海莲《智能客户端在煤电生产企业 ERP 中的应用》，微计算机信息，2007 年第 7-3 期。

面向服务的架构的概念最早是由 Gartner 公司在 1996 年提出的，在 2002 年 12 月，Gartner 公司又提出面向服务的架构是“现代应用开发领域最重要的课题”。

不过，对于面向服务架构的概念至今没有统一的定义。下面列举几条有代表性的定义。

Gartner 公司将面向服务架构描述为：“客户端服务器的软件设计方法，一项应用由软件服务和软件服务使用者组成。面向服务架构与大多数通用的客户端服务器模型的不同之处在于，它着重强调软件组件的松耦合，并使用独立的标准接口”。⁶

来自 IBM Redbooks 文章中的定义：“面向服务架构是服务的集合，这些服务通过网络中彼此通信实现业务过程。这种通信可能是简单的数据传送，也可能是两个或更多的服务协调进行某些活动。服务间是松耦合的，具有独立于平台的接口，完全支持重用”。⁷

Looselycoupled.com 将面向服务架构定义为：“按需连接资源的系统。在面向服务的架构中，资源被认为是可通过标准方式访问的独立服务，并提供给网络中的其他成员。与传统的系统结构相比，面向服务架构规定了资源间更为灵活的松散耦合关系。”⁸

SOA 是一种业务驱动的 IT 架构方法，支持对业务进行整合，使其成为一种相互联系、可重用的业务或者服务。通过在不同的应用和信息源之间建立联系，SOA 能够帮助客户提高业务的灵活性，增强企业底层的 IT 基础架构，并重用现有的 IT 投资⁹。

SOA 广义的来讲，不仅是一种架构风格，而且是包含了运行环境、编程模型、架构风格和相关方法论等在内的一套新的分布式软件系统构造方法和环境，涵盖服务的整个生命周期：建模-开发-整合-部署-运行-管理。

SOA 的核心是服务，业务被划分为一系列粗粒度的业务流程和业务服务，一个服务即一个由提供商开发的完成客户某种需求的工作单元，这些服务相对独立、自包含，即是松耦合的；具有独立于开发平台的定义良好的接口，并且支持重用。通过服务间的协作可以完成商务过程，协作既可以是一次简单的数据传送，也可以是激发另一系列服务实施的事件。

SOA 从提供服务、使用服务的角度去看待企业已有的信息资源。在这种方式下，同样的一种资源既可能是服务提供者，也同样可以是服务使用者；在这

⁶ 余浩，朱成等，SOA 实践——构建基于 Java Web 服务和 BPEL 的企业级应用，电子工业出版社，2009.1。

⁷ IBM, SOA with IBM Data Servers, www.ibm.com/redbooks, 2009 年 2 月 10 日下载。

⁸ 余浩，朱成等，SOA 实践——构建基于 Java Web 服务和 BPEL 的企业级应用，电子工业出版社，2009.1。

⁹ Sandy Carter, SOA & Web 2.0——新商业语言，清华大学出版社，2007 年 7 月。

种方式下，一个应用模块可能只提供一种服务，因此被封装成一个服务，也可能由于提供了多种服务，而需要进一步划分。只需开发目前还没有的业务功能服务，这样可以充分利用现有的资源，降低成本。

SOA 不是简单的应用系统集成，SOA 把企业中各个需要集成的应用程序进一步分解，抽象为服务，通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种各样的系统中的服务能够通过统一和通用的方式进行交互。SOA 的集成方式，其本质是对业务功能服务化后根据业务流程进行重新编排，是真正意义上的基于功能服务的集成，在面向服务的集成中包含了 EAI（企业应用集成）中基于消息集成的功能。在 SOA 架构下，各个服务是以完全独立的方式通过服务目录暴露在 SOA 集成平台上的，当新集成进来的应用系统需要使用现有的某个服务时，可以直接使用，无需再次开发，即服务是可重用的。

SOA 是一种进化而不是一种变革，因为它并不是一项全新的技术，而是建立在许多一直在使用的技术之上的，例如 Web 服务、事务技术、业务驱动、松耦合、组件、面向对象设计与开发方法以及 .NET 等，将这些技术与业务联结起来最终的目标就是支持业务的灵活性。

2.3.2 SOA 的应用实例

2006 年下半年，长安铃木与 IBM 在重庆的一个合作伙伴联手共同朝着 SOA 的方向转型。长安铃木信息系统存在的问题很多，经过一番分析整理后发现，真正需要解决的是三个方面的问题，即：信息的整合，把分散的系统整合起来，让用户更方便地获取信息；流程的整合，需要把底层各个应用系统之间的数据统一起来，把各个部门之间需要的协作流程统一起来；人员的整合，为使用者提供方便。在这三个层面的整合上，IBM 的 SOA 都有相应的解决方案。

借助 IBM 提供的 SOA 解决方案，长安铃木的 IT 系统实现了易用性、重用性、敏捷性。这三个价值是传统的解决方案所不具备的，传统的解决方案只是满足了当时的需求，但是并没有节约开支。长安铃木信息系统总监董斌说，他们单是完成对一些主要设备的升级有时都可能花费上百万，更不要说全套更换设备。而现在实施 SOA，通过利用已有的信息系统，为他们节约了大量开支，而且部署的成本相对来说也较低。通过实施 SOA，长安铃木获取信息的速度加强了，用户有新需求的时候 IT 满足用户需求的时间缩短了，响应加快了，用户

的满意度自然就高了。最重要的是，成本（实施成本、培训成本）下降了，并且会为企业带来更多的后续价值¹⁰。

2.3.3 SOA 方法学：面向服务分析与设计（SOAD）

与 SOA 的概念类似，SOA 方法学并不是全新的方法学，它是现有方法学的继承和发展，是一种混合的方法学。一方面，原有的方法学像面向对象的分析与设计（Object-Oriented Analysis and Design, OOAD）、企业体系结构（Enterprise Architecture, EA）框架和业务流程建模（Business Process Modeling, BPM）虽为我们提供了部分实践的指导，如帮助标识和定义体系结构内的适当抽象，但是经验表明，这些实践在各自单独应用时都达不到最佳效果，因此产生了一种对混合方法学的需求；另一方面，新产生的方法学 SOAD（Service-Oriented Analysis and Design, 面向服务的分析与设计）把现有方法学中的原理与引进服务概念后产生的新原理组合起来，使成功地进行 SOA 开发更容易，而且需要处理现有方法学产物和服务的关系，如业务流程和服务，系统架构和服务，服务和对象等。SOAD 最主要的职责在于发现、定义和实现服务，并指导如何和其他方法学集合完成这些职责。

下图展示了现有的 EA、BPM 和 OOAD 建模方法的主要应用领域。图中的水平轴表示软件项目生命周期阶段：分析、设计和开发；垂直轴表示不同抽象层或领域：业务，架构和应用，而建模活动通常就是分别在不同的抽象层上进行的。流程建模（BPM）用于业务领域的分析和设计，如业务流程的定义等；企业架构（EA）和方案架构（SA）侧重在架构领域的分析和设计，如根据业务需求确定 IT 系统目标，根据 IT 系统需求目标设计系统架构元素及他们之间的关系。面向对象分析和设计（OOAD）则贯穿分析、设计和开发三个阶段，它以用例为出发点，设计和实现用例的类和对象。

¹⁰ 长安铃木的 SOA 之路，IBM 中国，<http://www-01.ibm.com/software/cn/info/takebackcontrol/soa>，2009 年 3 月 1 日下载。

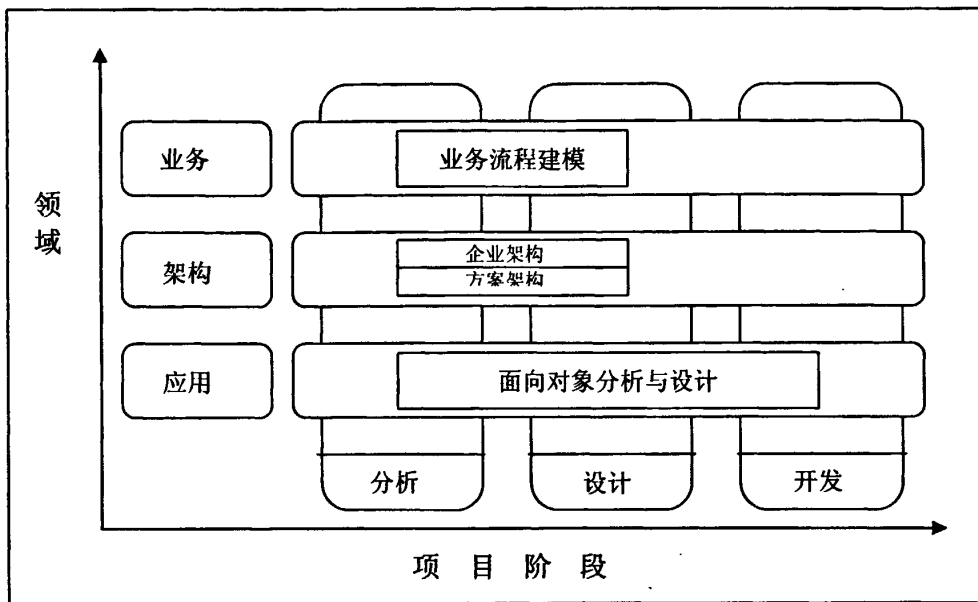


图 2.2 传统方法学层次分布

资料来源：《SOA 原理、方法、实践》¹¹

OOAD、EA\SA 和 BPM 在各自独立地应用时不能最佳的描述和实现系统需求，这正是 SOAD 产生的原因。关于原有方法学各自的局限和它们与 SOAD 的关系可以概括如下：

1. BPM 与 SOA

业务流程建模可以帮助企业对业务进行合理的划分，从而求得业务层面的灵活性，但它多侧重于流程建模本身，虽然在功能单元之上提供了端到端的视图，但是它们通常都没有涉及体系结构和实现领域。从面向服务角度来看，业务流程建模的结果是 SOAD 的重要输入，如业务流程、业务目标等是发现服务的重要依据，业务规则提供了服务约束的条件等。

2. EA 与 SOA

EA 是定义跨越业务单元边界的系统架构，企业内系统的主要构成元素，这些元素间的关系，以及将这些元素有机结合在一起的参考架构。但各种 EA 技术都缺乏业务领域的蓝图指导架构的设计，而且 EA 并没有解决如何找到易于重用且具有持久性的高质量企业抽象的问题。从面向服务角度来看，“服务”的概念贯穿于业务—架构—应用各个层面，在业务层 SOAD 与 BPM 结合将业务分解为各个服务，这可作为企业业务的蓝图指导架构的设计；而服务架构设计的结果，又是服务实现的重要依据。

3. OOAD 与 SOA

¹¹ 毛新生：SOA 原理、方法、实践，电子工业出版社，2007 年 07 月。

OOAD 方法在定义 SOA 方面提供了非常好的起点，同时不可否认在体系结构层次中应用 OOAD 技术和统一建模语言 (Unified Modeling Language, UML) 表示法是一个有效的实践，但是 OOAD 还是与类和单独的对象实例这样的微观层次的抽象有关。由于常常在每个问题域都创建单独的用例模型，因此，在纵观整个应用程序开发的大方向时许多目标变得模糊。此外，由于种种原因，用例模型并不总是与其对等的 BPM 保持同步，流程建模与开发活动彼此分离的情况时有发生。

由于这些原因的存在，所以需要混合的 SOAD 方法。这种方法以最佳的方式组合了 OOAD、EA/SA 和 BPM 中的原理，并且融入了一些具有创新性的思想。图 2.3 展示了这种新的方法的所融合的资源 (现有的原理和技术)。

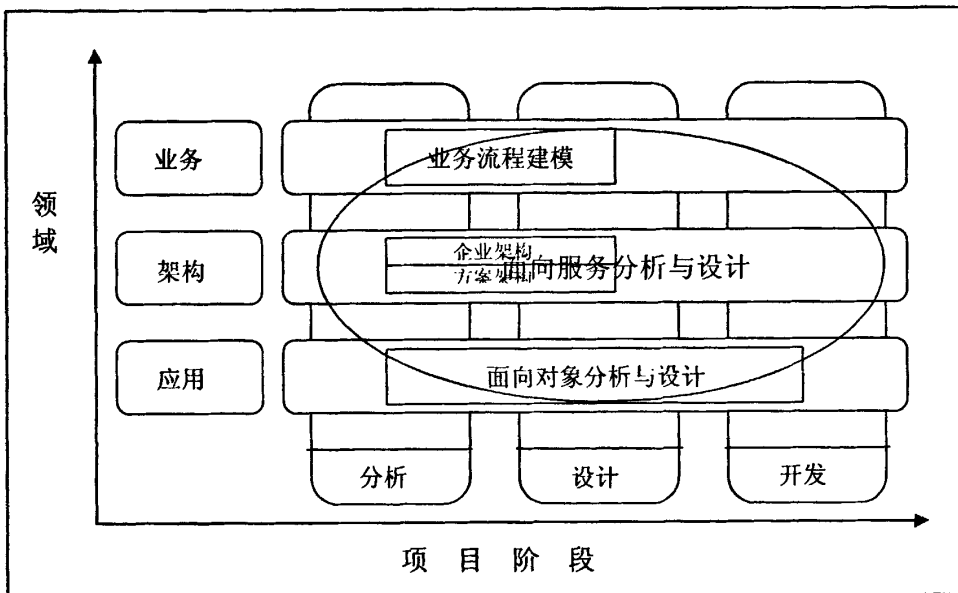


图 2.3 SOAD 与传统方法学的关系
资料来源：《SOA 原理、方法、实践》¹²

如图，面向服务的分析和设计贯穿项目周期的三个阶段和 IT 系统的三个领域。这暗示着，在操作层面上，面向服务的分析和设计与其他方法学关联紧密。

2.4 基于 SOA 的智能客户端

SOA 是从系统的流程和整体架构入手，来创造新的价值，这种在系统逻辑层面上的改变所带来的价值需要通过一个媒介传递给最终用户，用户也需要通过这个媒介与基于 SOA 的系统进行交互，智能客户端正好可以充当这样一个媒介。基于 SOA 的智能客户端，在设计上以 SOA 的核心思想——面向服务和系统灵活性为指导，通过“服务”将业务逻辑映射到 IT 系统中，并通过 Web 服

¹² 毛新生：SOA 原理、方法、实践，电子工业出版社，2007 年 07 月。

务向用户提供系统的功能，用户通过智能客户端有选择性的调用、下载系统的这些服务，从而最大限度地提升了系统的灵活性。

本文将基于 SOA 的智能客户端应用于 ERP 软件，主要是基于以下这些考虑。首先，由于 ERP 软件自身的特点，它与企业许多部门的业务活动（生产、销售、库存、人事）都有联系，在传统的 ERP 系统中，部门某一业务模块的调整将对整个企业的信息处理工作带来影响。比如，因为升级需要频繁的停机安装和备份，甚至带来相关模块操作界面和流程的改变。其次，传统的 ERP 系统各模块之间往往是复杂交错的紧耦合关系，某模块功能的升级将引发整个 ERP 产品的升级，对于业务和信息部门都会带来很大的压力。第三，采用 ERP 的制造业企业，由于其业务上的特点：客户数量大、层次不一，订单量大、处理时间短等，这些对 ERP 系统的灵活性和响应速度都提出了更高的要求。最后，因为 ERP 系统往往不是公司仅有的 IT 系统，在实施 ERP 系统是会涉及到和公司其他系统的集成问题，对于与异构的系统集成将会遇到很多困难。

对于上述问题，基于 SOA 的智能客户端技术都能较好地解决。首先，SOA 系统中由于各模块之间相对独立，是松耦合的，所以某一模块的调整或升级不会影响其他模块的正常工作，这提升了整个系统的稳健性和效率。其次，由于采用智能客户端技术，整个软件易于升级和维护，即使是频繁的升级也不用经常的停机安装和备份。而且，由于有离线运行功能，即使服务器端进行停机维护，客户端一样可以正常工作，增加了系统的可用性。第三，基于 SOA 的灵活性，新的业务需求可以设计成单独的服务或者调整现有的服务，由于有基于标准的接口，新的服务或调整之后的服务很容易嵌入现有系统，从而丰富系统功能，增加系统灵活性。最后，基于 SOA 的系统可以独立于开发技术和运行平台之上，这有利于将降低系统集成时的技术难度。

第 3 章 基于 SOA 的 ERP 智能客户端系统分析

即便是面向服务的分析方法，还是要从企业的需求和业务开始研究，本章便是从需求入手对本系统的功能进行了详细分解，对采购功能的业务流程做了研究说明，完成了系统的需求和业务模型的设计，为后面进行服务模型的分析做了铺垫。

3.1 需求分析

系统的需求往往涵盖很多方面，本节从业务、技术、开发环境等方面描述了本系统的需求。

3.1.1 需求综述

无论国内外，推行企业信息化进程都已经有很长时间了，投入了巨大的人力财力。现今，企业的大部分 IT 系统都是以部门为导向的，这是因为过去几十年，企业的业务运作模式是部门导向的，每个部门面对一个相对独立的信息系统，这些系统甚至有可能是基于不同开发语言和运行平台的。但是随着业务转型，业务流程开始发生变化，企业在内部需要能够跨部门高效协作，在外部也需要更好地同合作伙伴、客户互动。企业 IT 需与企业的业务对齐，因此，业务的转型催生了企业 IT 的转型。原有的紧耦合的，基于特定协议、技术，接口声明不明确的所有信息系统需要以一种新的架构重新组合——基于开放标准，接口和协议采用中立方式定义，可以实现跨硬件平台、操作系统和编程语言进行交互……这些正是 SOA 架构所追求的目标。

为了达到这个整体目标，各个单独的信息系统在开发或者改造的过程中也应该同样遵循 SOA 的思想：以服务划分业务，以组件实现服务功能，服务支持重用，服务间以松耦合的方式组合在一起，服务间的接口定义和通讯协议基于开放的标准（如 XML、URL、HTTP），消息通过统一的中间件传递等。

作为贯穿于企业整个供应链过程的 ERP 系统，更需要以 SOA 的思想为指导来构造，以适应企业资源管理的需要。可见，基于 SOA 的 ERP 系统在结构上更好地支持了业务的灵活性，与此同时，又为系统用户提供了一个界面丰富、响应迅速、安装和升级简便、使用模式多样的客户端，使得整个系统更具价值和竞争力。

3.1.2 系统的业务性需求

ERP 系统在整体上要实现合理调配企业资源、降低生产成本、促进销售、保证合同进度等目标，以此带给企业巨大的经济效益。这些目标的实现是通过 ERP 系统中不同子系统之间的信息传送和共享的手段来实现的。经过这么多年的发展，ERP 系统的主要功能在业界基本上已经达成了共识，一个 ERP 系统一般都能提供如下功能：生产计划编制；物料需求计划编制；车间作业计划编制；销售管理；采购管理；库存管理；质量管理；人力资源管理；财务会计管理等。

1. 生产计划编制：生产计划的目标是根据预测和客户需求来安排组件物料与产品的生产、采购等过程。帮助制定生产什么产品，什么时候生产，生产多少，分多少批次生产等决策。应该可以支持各种制造类型的企业，如按定单生产、按定单装配生产，批量及流程型生产企业等。

2. 物料需求计划：物料需求计划的主要功能是保证物料的可用性，计算在什么时间需要采购或生产多少数量，以保证满足需求。其目标是一方面保证不出现短缺，一方面尽量使成本和资金占用最小化。

3. 车间作业计划：车间生产计划的目标是制定出具体的生产进度，以尽可能保证按时完成生产任务，同时尽可能减少生产成本。车间作业计划以物料生产计划所产生的自制零部件生产计划为基础，按照交货期的前后和生产优先级原则，利用车间的生产资源将部件生产计划分解下达给不同的车间。

4. 销售管理：销售管理的任务是协助企业的销售活动，通过及时处理订单、准确跟踪订单等方法辅助提高企业的销售量。此模块支持销售和分销活动，具有完备的定价、订单处理、按时交货，所产生的数据可以作为生产计划编制的依据。

5. 采购管理：见后 3.2.1 采购管理子系统业务流程分析部分。

6. 库存管理：对支持生产的原材料、在制品、半成品、生产耗材、维护件等物料的保管、调配，也包括对仓库的管理。目标是保证企业库存的供应，降低库存成本和提高供货率。

7. 质量管理：通过制定质量方针、目标，明确质量控制职责等，保证企业经营活动中各环节、各项工作，以及各种产品的质量。这个目标是通过监控、输入和管理整个供应链与质量保证相关的各类处理、协调检查处理、启动校正措施、以及与实验室信息系统集成来实现的。

8. 人力资源管理：维护所有人员的基本信息、工资绩效信息等，包括招聘管理、员工薪资核算、培训计划、差旅管理等，为公司提供人力资源规划和管理解决方案。

9. 财务管理：集中公司有关会计的所有资料，提供完整的凭证、票据和全面的信息。确保企业内部的所有业务的资金周转，同时对外提供企业的各种财务报表。

还有一些可扩展的功能，如投资管理、生产预测、智能数据仓库等。

3.1.3 系统的技术性需求

SOA 的思想是在一个开放的标准上实现企业各种应用、服务的互联互通，因此基于 SOA 的系统需要能通过一种方法或技术，使得所开发的系统在不考虑系统所处环境时可执行并有良好的互操作性。基于这种考虑，本文采用 Web 服务技术来支持 SOA 架构的松耦合和互操作的特点。

另外，由于本系统智能客户端的特点，在客户端需要具备智能安装、自动更新、调用服务器发布的 Web 服务、可脱机操作的功能。智能安装、自动更新表现在当用户第一次使用系统或登录网站时，会提示下载与用户类型对应的客户端程序及相关必须的组件，在下载成功并安装后则会进入系统。下次再次使用系统时用户便可以直接从本地计算机登录，登录前系统会自动进行版本检测，如果服务器上有了新的软件版本，则会在系统运行前自动进行更新，无需人为干涉。当服务器端发布了一个新的 Web 服务，用户可以通过 Internet 调用此 Web 服务，即使用这个 Web 服务的功能完成某业务流程或是对数据库的操作。为了实现脱机运行，必须把应用程序运行时所必需的数据缓存到本地，这些数据包括用于提供程序验证、用户界面显示以及属于该用户下的业务信息。数据缓存是指将数据写入本地物理存储器中，可以以文本形式或者数据库形式存储下来，在本系统中为了实现客户端的部署，采取以 XML（可扩展标记语言）文件形式将数据存储在本本地。数据缓存包括两部分，一部分是用户在联机状态时定期的将服务器数据更新到本地，即将服务器数据库中的最新数据缓存到本地。还有一部分是在脱机状态下时，如果用户进行操作，需要将用户的操作信息写入本地，待系统处于联机状态时，将数据更新到服务器数据库中。

3.1.4 系统开发环境需求

本系统采用 SQL Server 2005 + MS Visual Studio 2005 的开发环境。此选择是基于这样的考虑：首先，智能客户端的概念本身就是作为 Microsoft.net 平台的一个特性提出的，.NET 平台从系统和语言级别对智能客户端技术都提供了全面支持，因此选用支持 .NET 的 MS Visual Studio 2005 开发系统的智能客户端是最好的选择。其次，Web 服务是 .NET 平台的应用之一，.NET 平台全面支持 Web 服务从创建到发布的过程，而且 .NET 平台支持多种开发语言，如 C#、VB

等。另外，.NET 平台的 ADO.NET 类提供了强大的数据库操作功能，可以与多种类型的数据库连接。在开发时可以灵活选择。本例基于 VB.net 语言开发。

3.2 采购管理业务流程分析

即使是面向服务的系统开发思想，也还是要从最初的业务流程入手进行分析，这是将业务逐步映射成服务的第一个阶段。由于考虑到工作量，本文只选取一个业务流程——采购管理进行业务流程分析。

企业从供应商处购买物料，经过生产加工，最后销售给顾客，这整个过程从资金流的角度看可以归结为两个循环：支出循环和收入循环，每个循环都伴随着相应的物流和信息流的运转，控制好这两个循环，对企业的经营管理至关重要。作为支出循环核心业务的销售管理，即对从采购订单产生到货物收到的全过程进行组织、实施与控制，其目标包括提供生产所需外购的物料、及时补充库存物料，选择最合适的供应商合作，降低采购成本、提高采购效率等。在系统设计上，对于采购管理的这些业务目标在系统中都可以得到支持，比如，可以根据生产计划自动得到采购计划，从而及时根据计划安排采购；对供应商进行评级，以便为每笔采购订单选择最合适的供应商，获得最有利的价格，从而降低采购成本；设定付款期限，及时通知财务部门与供应商结算，以免因付款延误产生滞纳金而增加成本。图 3.1 展示了采购管理的业务流程。

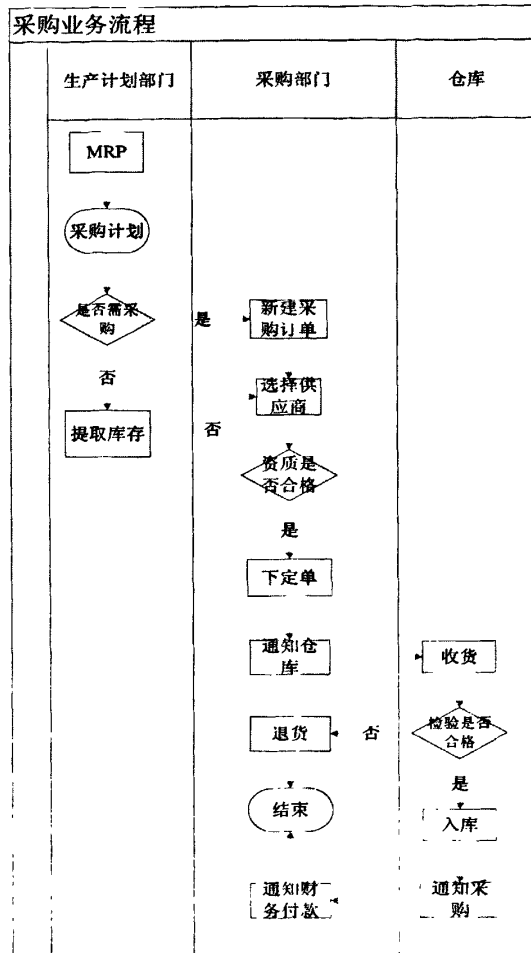


图 3.1 采购管理业务流程图

资料来源：本研究整理

从图中可以看出，采购管理是采购部门的职能，采购部门不仅负责管理采购订单还要负责管理与公司有业务往来的供应商信息，作为日后判定供应商资质的依据；以及与仓库合作完成收货、退货，将采购信息通知财务部门以便按合同付款给供应商。基本流程是：

- 1 根据物料需求计划产出的生产计划确定采购计划，将需求告知采购部门；
- 2 采购部门根据采购计划及供应商信息，判断供应商资质，确定供应商；
- 3 选择好合适的供应商之后，签订采购合同（下订单）；
- 4 跟踪订单完成情况，通知仓库准备收货；
- 5 供应商送货，仓库根据订单验货；
- 6 若货物合格，则将货物入库并告知采购部门；否则，通知采购部门办理退货；
- 7 货物入库后，通知财务部门办理付款；
- 8 财务部门付款给供应商，办理结算；

9 将此次采购完成情况记录到供应商信息中。

经过上面的分析，可以看出采购管理流程实际上可以分解为很多个子流程，这些子流程有些是相对独立的，比如供应商评级，即使没有采购物料，也可以对供应商的信息进行更改、给供应商打分；另外一些子流程很难确定是属于哪个部门的业务，有可能在不同的企业组织结构下归入不同的部门，比如订单货物的检验，在有些企业里可能是仓库负责验货，而有些企业里可能就是采购部门的职责范围。对于上述情况，用 SOA 的方法设计就比较有优势了，SOA 将流程、子流程封装成一个个服务，这些服务都是自包含、松耦合的，使用起来非常灵活，企业可以根据自己的需要组合不同的服务，以满足企业独特的功能。而且企业只需知道这些服务能完成什么功能，并用一种基于标准的技术在需要的时候调用这些服务就可以了，而完全不必考虑这些服务背后的实现技术等局限。

对于采购过程，越快下订单供应商就会越快发货，如果因为服务器停机维护而延误下单时间，有可能延误供货时间，造成企业的损失，所以需要尽量保证采购系统的可用性，不论与服务器连接成功与否，最好都不要影响订单的下达，所以在离线的状态下系统依然可用就显得十分重要，这点智能客户端技术可以提供很好的支持。还有，像货物检验这样的工作，可能是户外作业，不能保证实时联网，如果采用智能客户端技术，质检人员在不联网的情况下输入计算机的检验信息在回到室内之后，只要联上网，程序再次运行时数据就可以自动同步到服务器端，保证了数据的一致性和有效性。随着企业业务的扩展，采购系统需要维护的历史订单信息量越来越多，需要采购的物料的种类也可能会有所扩充，有些时候甚至连订单的格式、条目都需要有所更新。如果是传统的胖客户端程序，不但要逐个更新客户端本地数据库中的信息，还要更新原有程序的部分代码以支持新的业务需求，如果采用基于 SOA 的智能客户端技术，可以很好支持业务的多变，只需发布一个新的订单管理的 Web 服务或者更新原有的订单管理服务到一个新的版本，用户可以通过调用新的 Web 服务满足新的业务需求，如果只是更新了原有服务的版本，客户在运行本地程序时，客户端会自动检验新版本并更新，不会影响系统其他部分的使用。

第 4 章 基于 SOA 的 ERP 智能客户端系统设计

基于上一章对系统需求和业务流程的分析，在本章，将对如何从业务中发现服务、定义服务和实现服务做一个介绍。通过对系统的架构、网络模型、服务模型和数据库全面的设计，完成了系统的架构图、网络拓扑图、服务建模、数据流图、数据库表单等，进一步展示了基于 SOA 的 ERP 智能客户端系统的特点。

4.1 业务到服务的映射关系设计

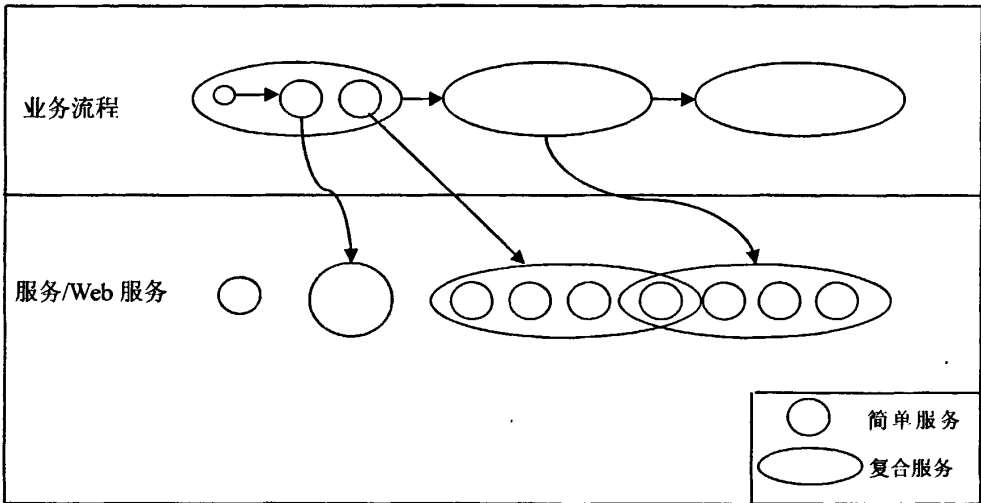


图 4.1 业务与服务映射关系图

资料来源：本研究整理

前面已经分析过 ERP 系统中包括的业务流程，如生产计划编制；物料需求计划编制；车间作业计划编制；销售管理；采购管理；库存管理；质量管理；人力资源管理；财务会计管理等。每个流程都按照一定的逻辑分步进行，当设计系统的服务模型时，依据企业的特点以及业务的重要性和优先级等条件，可能将一个业务流程或者一个子业务流程设计成一个简单服务，也可能设计成由几个简单服务组合而成的复合服务。一个简单服务可以单独实现，也可以与其他一个或多个简单服务组成更粗粒度的复合服务。比如，可以将采购（从采购管理里）和发票（从物料管理里）与总账和应付账（从财务会计里）结合起来实施，完成从向供应商发出采购订单到付款给供应商这一完整的业务流程。

4.2 系统架构设计

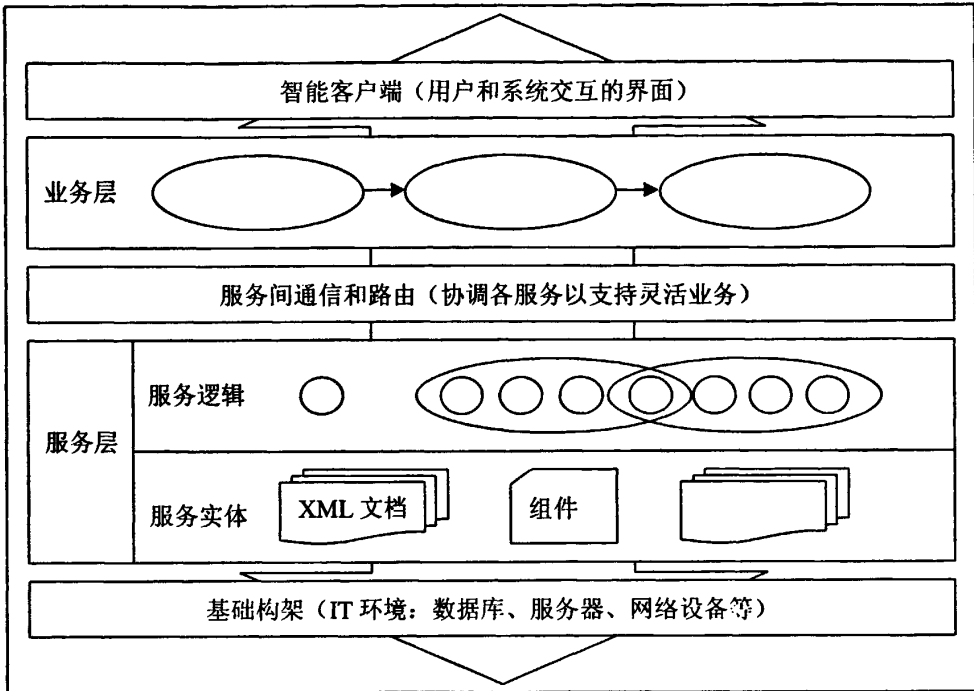


图 4.2 系统体系结构图

资料来源：本研究整理¹³

整个系统自底向上分为五个层次，自下向上每一层都支持上一层的功能，自上向下每一层都是下一层设计的目标。

1. 自下向上：基础构架层包括企业的 IT 环境，如 IT 设备、网络服务器资源、数据库、软件系统等。在基础构架上开发服务层的服务，服务实体是系统的代码、组件、包；这些服务实体用于实现服务逻辑，即完成服务。各服务间通过统一标准定义的接口相互通信，协作组合以完成某一业务流程。一定的业务流程可以用于满足用户的某种需求。

2. 自上向下：用户通过系统操作界面激活某一业务流程，根据业务流程的需要按照一定的次序请求调用各种服务，服务被调用，代码即运行实现，相应的服务器工作，数据随之刷新。

¹³参考“余于平：基于 SOA 的 SmartClient 直销系统设计与研究，中国优秀硕士学位论文全文数据库”改进而成。

4.3 系统网络拓扑设计

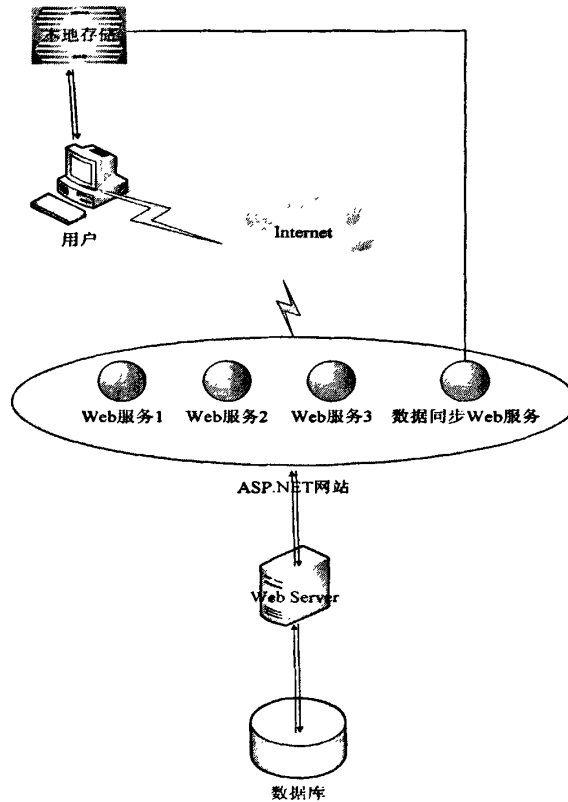


图 4.3 系统网络拓扑图

资料来源：本研究整理

系统通过服务器端向用户提供服务，将这些服务封装成一个个可实现的 Web 服务，用户端通过调用这些 Web 服务，完成一定的业务逻辑。客户端在必要的时候与服务器端进行数据同步，这种同步功能也是通过一个 Web 服务完成的。

4.4 采购管理服务模型设计

服务从何而来？如何定义？有没有统一的标准？这些是这一节将解释的问题，本节对采购管理进行服务建模，并对每个步骤都给出了详细的解释说明。

4.4.1 发现服务

SOA 是以服务为核心的开发思想，服务源于业务需求，因此我们将业务流程进行分解从而获取其相应的服务。根据前面的流程图，本人将采购管理流程

自顶向下分解，直到每一个子流程都可以清楚描述为止，得出如下结果，其中分析出的每一个子流程都是系统的一个候选服务。

表 4.1 初次识别的候选服务列表

| | | | | |
|--------|-----------|---------------|---------------|--|
| 0 采购管理 | 1.1 采购 | 1.1.1 查看采购计划 | | |
| | | 1.1.2 查看供应商信息 | | |
| | | 1.1.3 提交订单 | | |
| | 1.2 收货管理 | 1.2.1 收货 | 1.2.1.1 生成收货单 | |
| | | | 1.2.1.2 收货确认 | |
| | | 1.2.2 退货 | 1.2.2.1 接收退货单 | |
| | | | 1.2.2.2 退货 | |
| | 1.3 结算 | 1.3.1 结算申请 | | |
| | | 1.3.2 接收发票 | | |
| | 1.4 供应商管理 | 1.4.1 添加供应商信息 | | |
| | | 1.4.2 删除供应商信息 | | |
| | | 1.4.3 修改供应商信息 | | |

根据业务流程我们识别出这些候选服务，但是并不能包括所有对系统功能的需求，我们还要从业务流以及信息流过程入手再进行分析。下面利用控制矩阵对系统目标进行进一步的服务识别。

表 4.2 系统目标控制矩阵

| | 系统控制目标 | | | | | | |
|-----------------|--------|----------|-----------|--------|----------|----------|----------------|
| | 业务流程目标 | | | | | 信息流程目标 | |
| | 1 及时订货 | 2 降低采购成本 | 3 防止供应商欺诈 | 4 确保质量 | 5 保证及时到货 | 1 离线数据可用 | 2 客户端响应速度、互操作性 |
| 用于实现目标的系统服务 | | | | | | | |
| 1.1 采购 | 1.1 | | | | | | 1.1 |
| 1.2 收货管理 | | | | 1.2 | | | 1.2 |
| 1.3 结算 | | 1.3 | | | | | 1.3 |
| 1.4 供应商管理 | | 1.4 | | | | | 1.4 |
| M-1 供应商信誉信息 | | | M-1 | | | | |
| M-2 订单完成情况的监控 | | | | | M-2 | | |
| M-3 本地数据保存和更新问题 | | | | | | M-3 | |

借助控制矩阵分析，我们发现有三个系统目标还没有对应的服务来实现，分别是业务目标：3 防止供应商欺诈和 5 保证及时到货，以及信息流目标：1 数

服务 1.3 结算管理是与财务部门配合完成与供应商的结算。1.3.1 结算申请是调用财务部门的服务。1.3.2 接收发票：结算成功之后，供应商会寄出发票，采购部在收到发票后，将发票送财务并将相应订单状态更新为发票已收，这同样是通过订单追踪服务完成的。

服务 1.4 供应商管理是采购管理的又一个重要模块，1.4.1，1.4.2，1.4.3 对供应商信息的增删改操作都是对数据库中供应商信息的操作，可以合并成为 1.4.1 供应商信息维护服务。1.4.2 供应商评级服务是从信誉度、到货及时率、货物合格率、价格竞争力四个维度综合评定供应商的等级和分数。1.4.3 查询供应商信息是对供应商基本信息和评级信息的综合显示。

服务 1.5 订单追踪是根据订单的各个状态（是否收货、是否退货、发票是否收到）来跟踪采购完成情况。

服务 1.6 数据同步，把数据库表中的记录和本地 XML 文件中的信息进行同步，以免出现信息不一致的情况。

经过上面的分析，最后确定的服务和它们从属的业务如下：

表 4.3 最终确定的服务列表

| | | | | | |
|-------------|-----------|---------------|---------------|--------------|-------------|
| 0 采购管理 | 1.1 采购 | 1.1.1 查看采购计划 | | (MRP 系统) | |
| | | 1.1.2 查看供应商信息 | | (采购管理-供应商管理) | |
| | | 1.1.3 提交订单 | | (采购管理-采购) | |
| | 1.2 收货管理 | 1.2.1 收货 | 1.2.1.1 生成收货单 | | (采购管理-采购) |
| | | | 1.2.1.2 收货确认 | | (采购管理-订单追踪) |
| | | 1.2.2 退货 | 1.2.2.1 接收退货单 | | (库存管理系统) |
| | | | 1.2.2.2 退货 | | (采购管理-订单追踪) |
| | 1.3 结算 | 1.3.1 结算申请 | | (财务管理系统) | |
| | | 1.3.2 接收发票 | | (采购管理-订单追踪) | |
| | 1.4 供应商管理 | 1.4.1 供应商信息维护 | | (采购管理-供应商管理) | |
| 1.4.2 供应商评级 | | | (采购管理-供应 | | |

| | | | |
|--|----------|---------------|--------------|
| | | | 商管理) |
| | | 1.4.3 查询供应商信息 | (采购管理-供应商管理) |
| | 1.5 订单追踪 | 1.5.1 订单状态更新 | (采购管理-订单追踪) |
| | | 1.5.2 订单状态查询 | (采购管理-订单追踪) |
| | 1.6 数据同步 | | (采购管理) |

4.4.2 服务规约

通过发现服务已经确定了采购管理系统所有要实现的服务，下一步就是对这些服务进行进一步的约束——定义每个服务的接口以及说明。服务说明在上一小节已经有了基本介绍，这里不再赘述，下面选几个主要的服务进行接口定义：

表 4.4 采购服务接口

| | | |
|--------|-------|--------------------------|
| 1.1 采购 | | 服务从属部门：采购部 |
| 输入参数 | 采购信息 | 订货日期、订单项目、供应商、订单数量、单价 …… |
| | 供应商信息 | 评级、分数…… |
| 输出参数 | 采购订单 | |
| | 收货单 | |

表 4.5 结算申请服务接口

| | | |
|------------|-------|------------------------------|
| 1.3.1 结算申请 | | 服务从属部门：财务部 |
| 输入参数 | 采购订单 | 订单号、订货日期、订单项目、供应商、订单数量、单价 …… |
| 输出参数 | 结算申请表 | |

表 4.6 供应商信息维护服务接口

| | | |
|---------------|---------|------------------|
| 1.4.1 供应商信息维护 | | 服务从属部门：采购部 |
| 输入参数 | 供应商基本信息 | 供应商名称，地址，经营项目 …… |
| 输出参数 | 修改是否成功 | |

表 4.7 供应商评级服务接口

| | | |
|-------------|-----|------------|
| 1.4.2 供应商评级 | | 服务从属部门：采购部 |
| 输入参数 | 信誉度 | |

| | | |
|------|-------|--|
| | 到货及时率 | |
| | 货物合格率 | |
| | 价格竞争力 | |
| 输出参数 | 评级 | |
| | 评分 | |

表 4.8 查询供应商信息服务接口

| | | |
|---------------|---------|------------|
| 1.4.3 查询供应商信息 | | 服务从属部门：采购部 |
| 输入参数 | 供应商名称 | |
| 输出参数 | 供应商基本信息 | |
| | 供应商评级信息 | |

表 4.9 订单状态更新服务接口

| | | |
|--------------|--------|------------|
| 1.5.1 订单状态更新 | | 服务从属部门：采购部 |
| 输入参数 | 订单号 | |
| | 状态信息 | |
| 输出参数 | 更新是否成功 | |

表 4.10 订单状态查询服务接口

| | | |
|--------------|--------|------------|
| 1.5.2 订单状态查询 | | 服务从属部门：采购部 |
| 输入参数 | 订单号 | |
| 输出参数 | 订单状态信息 | |

表 4.11 数据同步服务接口

| | | |
|----------|----------|------------|
| 1.6 数据同步 | | 服务从属部门：采购部 |
| 输入参数 | 数据库表名 | |
| | XML 文件路径 | |
| 输出参数 | 同步是否成功 | |

4.4.3 服务实现

前面已经提到过，本设计采用 Web 服务来实践 SOA 思想，这是因为 SOA 概念定义了服务如何相互理解以及如何交互，应用是服务使用者通过接口访问服务而形成，使用者能够发现由服务提供者发布的该类接口并通过网络来进行调用，这种设计思想同 Web 服务的具体实现技术较为吻合。

Web 服务是用标准的、规范的 XML（可扩展标记语言——跨平台的、依赖于内容的技术，定义了一种互联网上交换信息的标准，是处理结构化文档信息的有利工具）描述的一些操作的接口，这一描述囊括了与服务交互需要的全部

细节，包括消息格式、传输协议和服务端点位置。该接口隐藏了实现服务的细节，允许基于不同硬件、软件平台和编程语言来开发和使用服务。这使得基于 Web 服务的应用程序成为了松耦合、面向服务、分布式和跨平台的。Web 服务可以单独地履行一项特定的服务，或同其他 Web 服务一起用于实现复杂的商业服务或业务流程。

Web 服务是通过三个角色：Web 服务提供者、Web 服务请求者、Web 服务中介者之间发布、发现、绑定三个动作构建起来的。

1. Web 服务提供者就是 Web 服务的拥有者。在本系统中，Web 服务的拥有者是 ERP 软件提供商，ERP 软件提供商通过基于 SOA 的分析，发现并实现一个个服务，包装成 Web 服务发布到 Internet 上等待为其他服务和用户提供这些已经实现的服务；

2. Web 服务请求者就是 Web 服务的功能的使用者，它通过基于特定协议的消息向服务请求者发送请求以获得服务。在本系统中 Web 服务请求者就是购买 ERP 软件的企业，企业需要借助 ERP 软件提供商的服务来实现企业 ERP 管理的功能。

3. Web 服务中介者的作用是把一个 Web 服务者请求与合适的 Web 服务提供者联系起来。本系统中，Web 服务请求者和 Web 服务提供者都是唯一的，所以 Web 服务中介者的作用发挥不出来，本人认为可以把本系统中 Web 服务中介者的功能简单化，设计成连接一个 Web 服务请求与合适的 Web 服务中介。在系统中，本人设计了一个服务中介数据表，表中记录了每个 Web 服务与其所在的 URL 的映射，这样当用户请求一个 Web 服务时，可以方便的找到程序对应的 URL 地址，从而启动服务的功能。

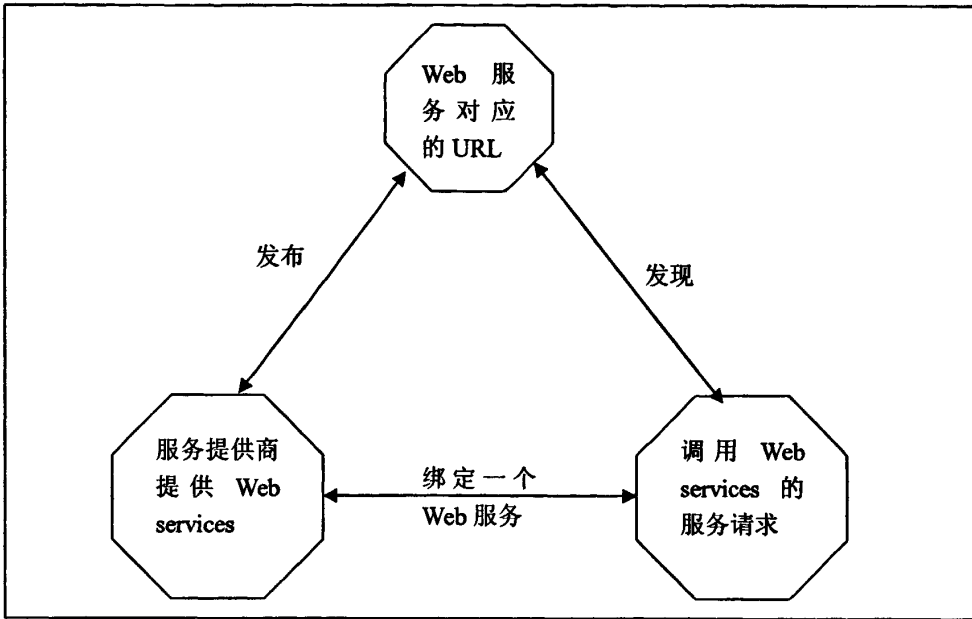


图 4.4 Web 服务结构图
资料来源：本研究整理

4.5 数据库设计

本节根据之前的业务、服务模型分析结果，设计了系统的数据库表以及表之间的关系。

4.5.1 数据表项设计

根据 3.2 节业务流程图分析和 4.4 节服务模型中对各服务输入、输出参数的分析，可以得到系统的数据流程图如下：

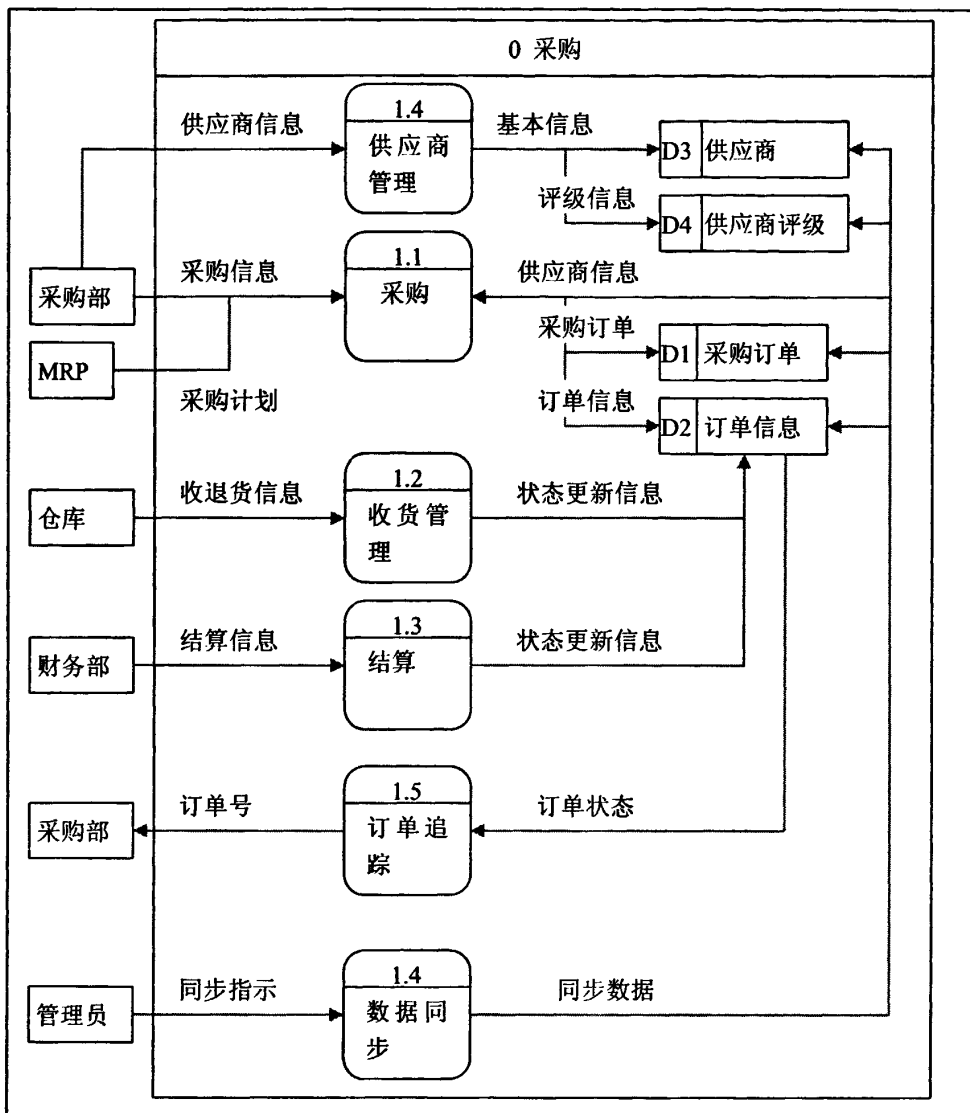


图 4.5 顶层数据流程图

资料来源：本研究整理

数据流程图中涉及到四个存储，在数据库设计时这四个存储分别设计成四张数据表：订单（PO）、订单信息（POInfo）、供应商（Supplier）、供应商评级（Evaluation）。各个表的字段如下：

表 4.12 订单数据表

| 字段 | 数据类型 | 说明 |
|------------|-----------|---------|
| POID | Int | 主键，自动编号 |
| PODate | SmallDate | |
| POItemID | Int | 外键，订单项目 |
| SupplierID | Int | 外键，供应商 |

| | | |
|-----------------|---------------|--------------|
| POQty | Char (30) | 订货数量 |
| Price/Item | Money | 单价 |
| ConsignmentDate | SmallDate | 最迟交货期 |
| Address | Varchar (200) | 送货地址 |
| DepartmentID | Int | 外键, 订货部门 |
| TotalPrice | Money | 订单总价 |
| IsComplete | Bit | 0-没完成, 1-已完成 |

表 4.13 订单信息数据表

| 字段 | 数据类型 | 说明 |
|----------|------|-----------------|
| POID | Int | 主键, 订单编号 |
| Accepted | Bit | 是否已收货, 0-否; 1-是 |
| Returned | Bit | 是否已退货, 0-否; 1-是 |
| Bill | Bit | 是否有发票, 0-否; 1-是 |
| Payed | Bit | 是否已付款, 0-否; 1-是 |

表 4.14 供应商信息数据表

| 字段 | 数据类型 | 说明 |
|------------------|---------------|-----------|
| SupplierID | Int | 主键, 供应商编号 |
| SupplierName | Varchar (30) | 供应商名称 |
| Address | Varchar (200) | 地址 |
| Contact | Varchar (30) | 联系人 |
| Website | Varchar (30) | 网站 |
| TEL | Varchar (30) | 联系电话 |
| ItemIntroduction | Varchar (200) | 经营项目介绍 |

表 4.15 供应商评级数据表

| 字段 | 数据类型 | 说明 |
|---------------|---------|-----------|
| SupplierID | Int | 主键, 供应商编号 |
| Reputation | Float | 信誉等级 |
| IntimeRate | Float | 到货及时率 |
| PassRate | Float | 货物合格率 |
| PriceCapacity | Float | 价格竞争力 |
| Grade | Varchar | 等级 |
| Scores | Float | 总成绩 |

对于订单表中的订单项目和订货部门的信息, 由于考虑到范式要求, 所以将这两个信息单独设计成表: 订单项目和部门, 字段如下:

表 4.16 订单项目数据表

| 字段 | 数据类型 | 说明 |
|---------------|---------------|-----------|
| ItemID | Int | 主键, 项目编号 |
| SupplierID | Int | 主键, 供应商编号 |
| Name | Int | 项目名称 |
| Specification | Varchar (100) | 项目介绍 |
| PacketType | Varchar (30) | 包装类型 |
| Unit | Varchar (10) | 包装单位 |
| Price | Money | 单价 |

表 4.17 部门数据表

| 字段 | 数据类型 | 说明 |
|-------------|---------------|----------|
| Dep_ID | Int | 主键, 部门编号 |
| Dep_Name | Varchar (30) | 部门名称 |
| Description | Varchar (100) | 部门介绍 |
| TEL | Varchar (30) | 电话 |
| Director | Varchar (30) | 负责人 |

4.5.2 数据表关系设计

数据库表之间的关系如下图所示:

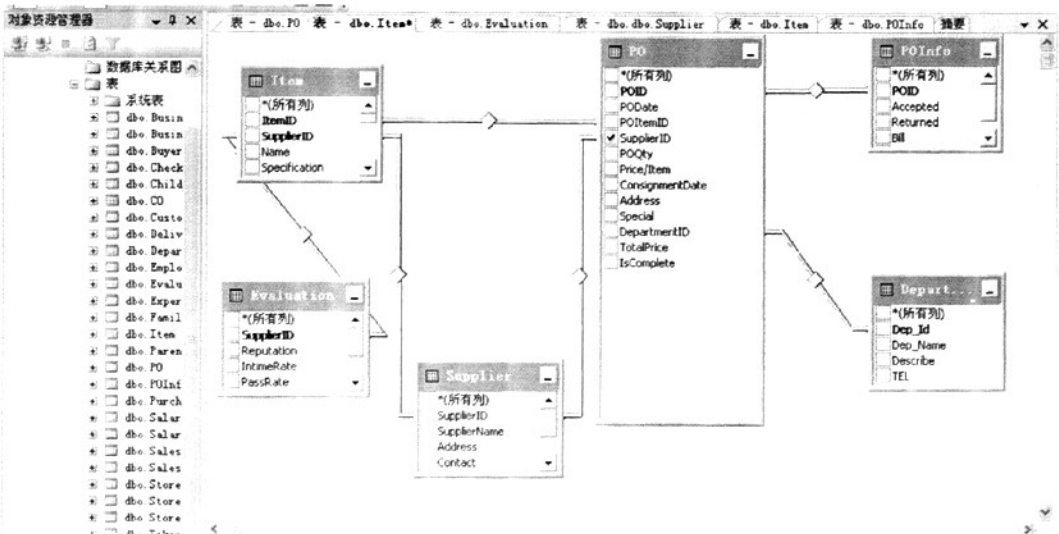


图 4.5 数据库表间关系图

资料来源: 数据库截图

4.6 界面设计

本系统的界面分为客户端界面和 Web 服务界面，这一节，对客户端和用于发布 Web 服务的网站的界面分别进行了设计。

4.6.1 客户端界面设计

1. 客户端启动界面和程序主界面。启动界面：显示系统版本和版权信息，版本信息可以自动更新。主界面：提供进入各个服务的按钮。



图 4.6 系统启动界面和主界面

资料来源：系统界面截图

2. 订单管理中输入订单界面。在此界面上输入各项订单信息后，点击“提交”，系统会首先判断联机状态，如果与服务器联通成功，则直接在服务器数据库中插入一条订单记录；如果联机不成功，则将输入的信息暂时保存到本地 XML 文档，等下次联机时上传到服务器。具体实现的函数和代码在下一章介绍。

图 4.7 订单界面

资料来源：系统界面截图

4.6.2 网站页面

网站上提供所有已发布的 Web 服务的链接, 点击链接可以看到此 Web 服务的说明。用户可以在说明页面点击“调用”按钮在浏览器上调用该服务, 也可以通过在智能客户端点击相应按钮远程调用服务。



图 4.8 Web 服务发布网站界面

资料来源：网站界面截图

第 5 章 基于 SOA 的 ERP 智能客户端部分模块的实现

基于上一章的设计，在 VS.NET + SQL Server 2005 的开发环境中，通过创建 Web 服务和客户端应用程序，本系统最后实现了采购子系统智能客户端的这些功能：丰富的用户界面，支持在线和离线使用，易于安装和更新，通过调用 Web 服务完成业务功能。在本章中，对创建和调用 Web 服务的步骤进行了说明，并给出了实现数据绑定和数据更新功能的 Web 服务的代码，以及客户端调用这两个服务显示和更新数据，在本地生成 XML 文件存储数据的代码，并做了相应解释。

5.1 Web 服务创建

VS.NET 提供了强大的 Web 服务创建功能。此处创建一个 Web 服务，实现客户端与服务器端数据绑定、更新服务器端数据的功能。具体步骤如下：

1. 先安装 IIS (Internet 信息服务)。

2. 运行 VS.NET，新建一个 ASP.NET Web 服务，将其命名为 WebService。创建完成后将会在本地服务器默认地址 C:\inetpub\wwwroot 文件夹下创建一个新的站点 WebSite2，以及该站点的一个新网页 Service.aspx，该页面表示一个 Web 服务。

3. 打开 Service.aspx 的代码窗口，编写代码（代码参见附录）。

- a. 需要引入的命名空间：

System.Web.Services 包含与创建和使用 Web 服务的对象相对应的类。添加到 Public 方法中的 System.Web.Services.WebMethodAttribute 类能够让远程客户端调用方法。

System.Data.SqlClient 包用来访问连接 SQL 类型的数据源。

- b. 需要用到的对象

SqlConnection 对象在 Web 页面和数据库间建立连接，能够在已经创建的连接上通过调用 Open 的方法被明确的打开连接。

SqlCommand 对象可以用来定义一些该如何操作数据库信息的指令。可以用典型的 SQL 语句来表达，比如 Select 语句等。

DataSet 对象中可以包含任意数量的 DataTable，且每个 DataTable 对应一个数据库中的数据表和视图。

SqlDataAdapter 用于结合 SqlConnection 对象和 SqlCommand 对象来向 DataSet 中填充数据,然后将信息解析回 Microsoft SQL Server 数据库。

4. 完成后运行, 将看到如下界面, 这个 Web 服务中有两个方法 Binding 绑定数据源和 Update 更新数据。

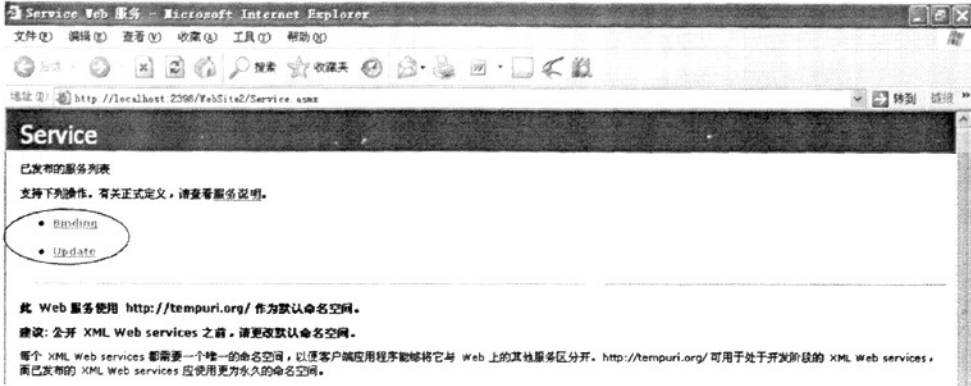


图 5.1 Web 服务发布网站界面
资料来源: 网站界面截图

5. 引用这个 Web 服务。在客户端应用程序的“解决方案资源管理器”窗口中, 选择“添加 Web 引用”, 打开下图的对话框, 在 URL 地址栏输入 Web 服务所在地址, 本例为刚才完成的 Web 服务页面文件的地址: `http://localhost:2389/WebSite2/Service.aspx`, 可以搜索到 Service 这个服务, 并提供 Binding 和 Update 两个操作。单击“添加引用”。

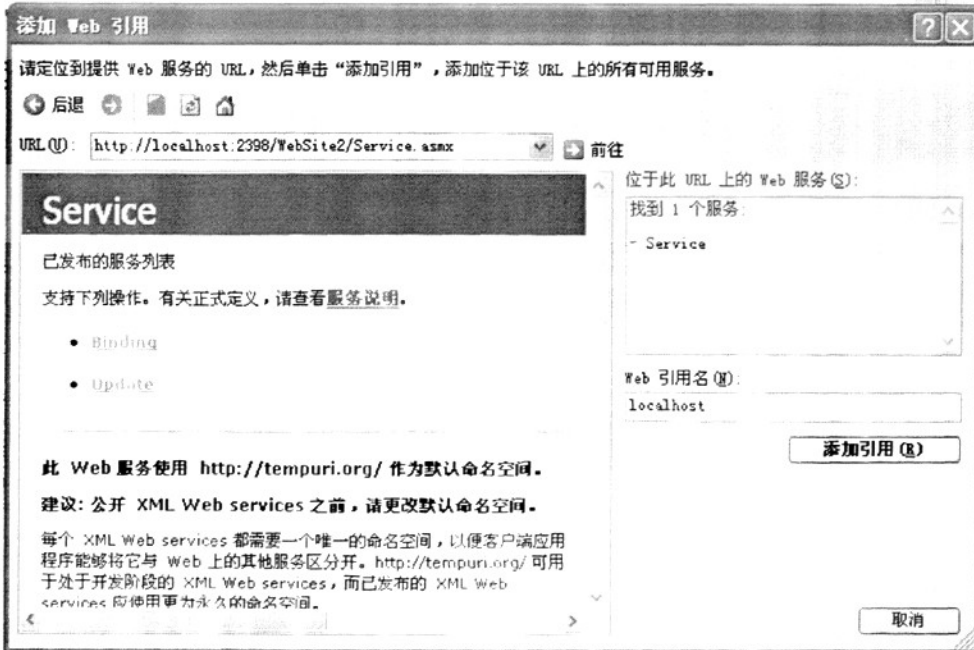


图 5.2 Web 服务引用

资料来源：软件界面截图

6. Web 引用添加完毕后，在应用程序中添加一个新窗体 Form1，此窗体中有一个 DataGrid 控件用于显示数据库中的数据，三个 Button 控件，分别用于绑定数据源，更新数据和生成本地 XML 文件。用于调用 Web 服务的代码：

```
Dim MyService As New Localhost.Service()  
DataGrid1.DataSource = MyService.Binding()
```

7. 至此，编写和调用 Web 服务的工作全部完成，当程序运行时，单击程序中的【绑定】按钮就会对程序中的 DataGrid1 组件实现数据绑定，单击程序中的【修改】按钮，则程序会根据 DataGrid1 中的内容来更新数据库，运行界面如下图：

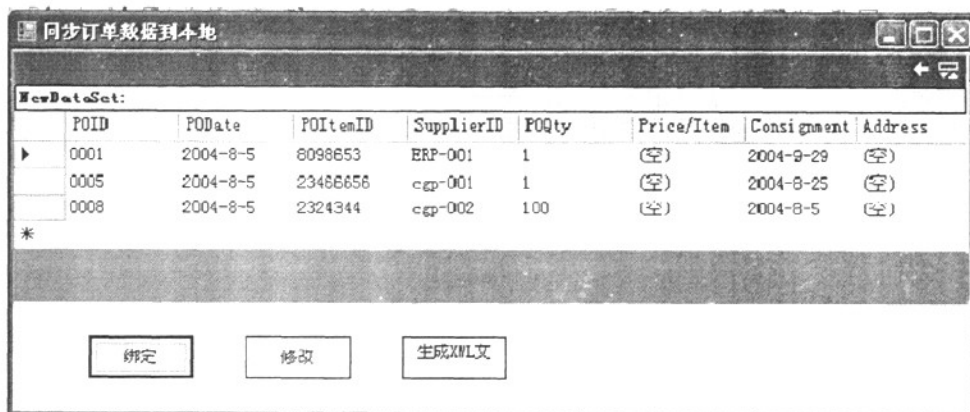


图 5.3 调用 Web 服务实现界面

资料来源：软件界面截图

5.2 在线同步服务器数据到本地

在实现了上一步数据绑定的基础上，我们要将服务器端的数据存储到本地的 XML 文档中，以便满足智能客户端离线使用的特征。在这里，我们需要将从数据库中提取的信息转化成 XML 格式，并保存在本地。

需要引入的命名空间：

System.XML.Serialization 提供了将一个对象串行化为 XML 格式的基本方法。

需要用到的方法：

数据集的 WriteXML 方法，将数据集中的数据以 XML 的格式写入到某一文件中，如 sqlDataSet.WriteXml("E:\ERP\Order\Orderxml.xml")，就是将 Sql 数据库中的某数据集的内容以 XML 格式写入 E:\ERP\Order\Orderxml.xml 文件中。

当在用户端窗体单击【生成 XML 文件】按钮时，将从数据库中同步来的数据以 XML 文件格式保存在本地的。生成的 XML 文档如下：

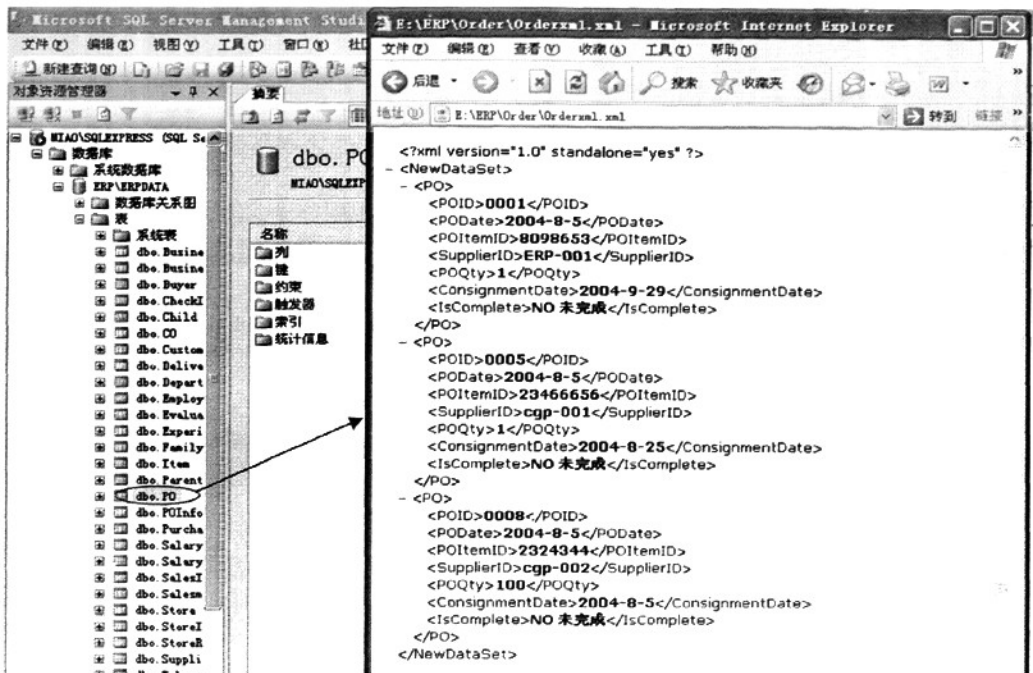


图 5.4 由服务器端数据生成的本地 XML 文件
资料来源：XML 文件截图

5.3 离线生成本地 XML 文档

在脱机时提交的订单信息，需要先缓存在本地 XML 文档，以便下次联机时上传致服务器，步骤如下：

1. 需要引用的命名空间：

System.Xml.Serialization 提供了把对象串行化为 XML 格式的方法。

System.IO 用于对系统中文件的读写操作，提供打开和关闭文件的 Open/Close 方法。

2. 定义一个 Order 类用于存储从窗体获得输入的订单信息，并用自定义的 OrderFromForm 方法实例化这个类。

```
Public Sub OrderFromForm(ByVal Order As Order)
```

```
Order.POID = Form3.tbPOID.Text
Order.ProID = Form3.tbProID.Text
Order.ProName = Form3.tbName.Text
Order.Supplier = Form3.cbSupplier.Text
Order.Count = Form3.tbCount.Text
Order.POTime = Form3.tbTime.Text
Order.MonType = Form3.cbMONTType.Text
Order.Price = Form3.tbPrice.Text
```

Order.Address = Form3.tbAddress.Text 定义了一个方法，用从订单输入窗体获得的数据
Order.Special = Form3.tbSpecial.Text 将Order类实例化，窗体如上图

End Sub

3. 自定义的 SaveXmlFile 方法用于创建 XML 文档

```
Public Function SaveXmlFile(ByVal filename As String)
    Dim tempFilename = filename & ".tmp"
    Dim tempFileinfo As New FileInfo(tempFilename)      创建临时文件
    If tempFileinfo.Exists = True Then tempFileinfo.Delete()      若文件已存在则先删除
    Dim stream As New FileStream(tempFilename, FileMode.Create)      打开文件
    SaveXmlFile(Stream)      保存XML格式的对象
    Stream.Close()      关闭文件
    tempFileinfo.CopyTo(filename, True)      重命名文件
    tempFileinfo.Delete()      删除临时文件
End Function
```

4. 当在订单窗体单击【提交】按钮时，生成的本地 XML 文件如下：

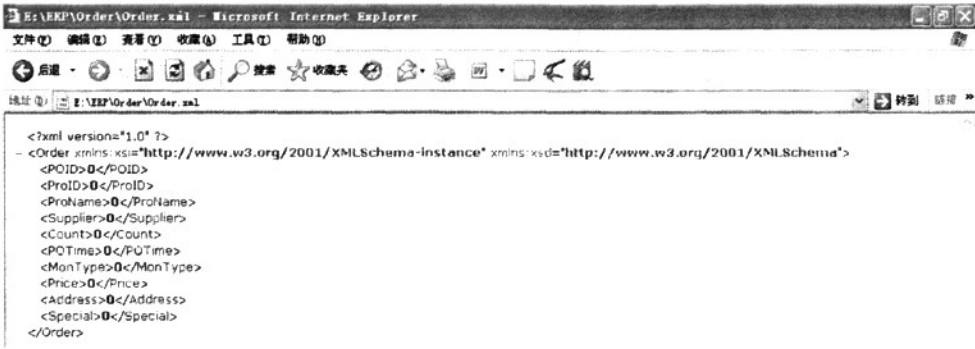


图 5.5 离线生成 XML 文件以缓存数据
资料来源：XML 文件截图

第 6 章 不足与改进

在完成了分析、设计、实现整个过程之后，通过对研究过程的思考，作者提出了实施 SOA 的难点问题——服务粒度的把握这个观点，并说明了原因。最后，对系统设计上的不足作者提出了改进建议。

6.1 研究中遇到的问题

本节是本人在完成此论文的过程中对 SOA 方法的一些见解，提出了本人认为实现 SOA 的核心问题是定义服务粒度的大小。

6.1.1 遇到的问题

本文研究的是基于 SOA 的 ERP 智能客户端系统，在系统完成的过程中，采用了面向服务的分析和设计方法，按照服务发现、服务规约、服务实现的步骤，将企业的业务映射为一个个可实现的服务。在这个过程中，本人认为最为困难的是服务发现阶段确定每个服务的粒度问题，服务粒度是指一个服务所包含功能的数量。一个细粒度服务支持的是少数功能或较简单的业务流程，而一个粗粒度服务支持的则是较多的功能以及较复杂的业务流程。

服务粒度既不能太大也不能太小，粒度太大，会提高服务的耦合度，不利于重用，从而也就丧失了 SOA 的松耦合、可重用的核心思想；粒度太小，服务的性能会降低，而且业务与支持他的服务之间的映射会变得非常复杂，开发工作量也会无谓的增大。什么样的服务粒度才是最佳的，这是关系到 SOA 能否发挥优势的关键。但是，由于企业业务间的差异性，即便是相似的业务流程，在不同的企业所需的业务粒度可能就有所不同，所以想通过单一的规则来指导服务粒度的设计是不现实的。而且，服务粒度的确定是后面定义服务规约和实现服务的基础，所以服务粒度的设计必须是成功的，不然将会严重降低系统的价值。

在本研究中，基于采购管理业务流程的分解，本人虽然力求合理确定每个服务的粒度，但是由于并没有统一的方法指导，因此，此服务模型只是基于本人对业务的分析和对面向服务概念的理解而来，这个模型是否真的合理目前还不能确定，需要在系统运行过程中不断进行检验。

6.1.2 解决方法

虽然没有具体的操作规则可以用来指导服务粒度的设计，但是在业界还是有一些普遍的共识的，一般来说，粒度常常是业务流程分解的功能单元，业务流程需要越详细的定义，服务粒度就应该越细。因此，在设计服务模型之前，应该力求全面详细地去理解业务流程，并且在整个项目分析、设计、实施的过程中要不断地对服务模型进行调整，以便达到最佳。

6.2 研究的不足和改进

本文只是完成一个系统的原型设计，在设计过程中难免会有考虑不周的情况，基于对研究过程的反思，作者对本文理论及设计上的不足做了说明，并提出了有建设性的改进意见。

6.2.1 理论研究上的不足

由于 SOA 和智能客户端都是比较新的概念，所搜集到的与这两个概念相关的资料很有限，本人对这两个概念的理解也比较有限。日后，本人会持续对这两方面理论的研究，争取对理论有更深入和全面的理解

6.2.2 系统设计上的不足

1. 本地缓存数据的安全性问题。由于智能客户端支持脱机操作的特点，当系统需要进行脱机数据操作，客户端会缓存一部分数据到本地 XML 文件，但是客户端对于数据的安全控制关注程度不如服务器端高，因此这给应用程序的安全性带来了更大的挑战。缓存在客户端的数据对于应用程序来说是严格保密的，为了保证这些数据的安全，应该采取一定的安全措施。如，权限管理等对应用程序运行具有重要意义的键数据，不应当缓存在本地，这些应该只允许在联机下进行操作，直接将数据写入服务器数据库中。还有可以用诸如 AES 这类的算法对 XML 文件加密，在使用前进行解密以确保本地数据文件安全。

2. 数据同步的效率问题。还是由于智能客户端支持脱机、联机两种方式操作的特点，对于同时保存在服务器端和本地客户端的数据需要进行同步才能避免数据不一致的问题。在只涉及到少部分数据的同步的时候，因为同步数据所占用的资源可能不会给整个系统造成太大的负担。但是当运行整个 ERP 系统时，需要同步的数据量可能会非常的大，因此可能会造成整个系统运行效率的

降低。对于这个问题，可以考虑用多线程技术优化程序，用一个单独的线程负责监控服务器和本地数据的更改，当监控到有数据不一致时，该线程自动控制数据的同步操作，避免了对系统其他功能的影响。

6.3 总结

基于 SOA 的 ERP 智能客户端系统主要是为了研究如何增强支持业务的 IT 系统的灵活性问题，以及探索 ERP 软件新的技术方向，具有一定的理论和实践意义。

本文首先对 SOA 和智能客户端技术进行了介绍，分析了他们各自的特点及将二者结合应用在 ERP 系统中的技术优势，在借鉴相关研究成果的基础上，采用 Web 服务来实现系统的开发。文中对用面向服务的分析和设计过程中涉及的相关系统结构、模型以及系统的实现技术进行了详细的研究与阐述。

论文的主要研究成果及创新点包括：

1. 系统采用 Web 服务技术，开发出的服务是松耦合、可重用的，由于在应用程序开发方面是基于统一标准的，所以开发出的 Web 服务可供不同语言的应用程序调用。

2. 采用智能客户端技术，所开发的系统能在离线和在线两种网络状态下无缝转换。系统的服务部署方面也较为简单，且升级是自动化的。

由于时间和工作量原因，本文未对所有 ERP 功能模块进行分析和设计，只是对采购管理子系统进行了面向服务的分析设计，也并为对所用服务进行实现。且理论和应用的部分问题还有待进一步研究与探索。

参考文献

- [1]Raouf Boutaba ,Wojciech Golab ,Youssef Iraqi: A web - services - based managemet system[J] . IEEE Communications Magazine ,2004 ,42 (7), P101~107.
- [2]David Hill etc: Smart Client Architecture and Design Guide, Microsoft Corporation, June 2004.
- [3]Raya Fidel: An Ecological Approach to the Design of Information Systems, Bulletin of the American Society for Information Science and Technology. Silver Spring, Oct 2006. Vol. 33, Iss. 1; P6~9.
- [4]Appleton, E.L., 1997:How to survive ERP. Datamation 43 (3),P50-53, Available from website [http:// www.datamation.com/entap/03erp.html](http://www.datamation.com/entap/03erp.html).
- [5]Shang S., Seddon, P., 2000: A comprehensive framework for classifying the benefits of ERP systems. In: Proceedings of AMCIS 2000, vol. II, P1005~1014.
- [6]CCIDNET, 2002: 2001 China's ERP Market Share. CCIDNET, Beijing.
- [7] Legare, T.L., 2002: The role of organizational factors in realizing ERP benefits. Information Systems Management 19 (4), P21~42.
- [8]程控,革扬: MRPII/ERP 原理与应用, 清华大学出版社, 2002年, P1~17.
- [9]姜激扬: ERP 的新方向——智能客户端, 微电脑世界, 2006年3月, P27.
- [10]李莹, 王甲民, 杨子翔等: 基于.Net 的多模 ERP 技术架构, 计算机工程与应用, 2005年8月, P185~187.
- [11]毛新生: SOA 原理、方法、实践, 电子工业出版社, 2007年07月.
- [12]Clive Gee 等: SOA 项目交叉学科建模方法, 电子工业出版社, 2006年06月.
- [13](美)伊尔著, 王满红, 陈荣华译: SOA 概念、技术与设计, 机械工业出版社, 2007年01月.
- [14](美)诺伊斯著, 汪泳译: Windows Forms 2.0 数据绑定: .NET 智能客户端数据应用程序设计, 电子工业出版社, 2007年10月.
- [15]陈娴, 刘旭辉, 陈颖: Visual Basic.NET 企业级开发案例精解, 人民邮电出版社, 2005年10月.
- [16]Sandy Carter: SOA & Web2.0——新商业语言, 清华大学出版社, 2007年7月第一版.
- [17]李鑫等: Visual Basic.NET 课程设计案例精编, 中国水利水电出版社, 2006年1月第一版.
- [18]郭瑞军, 王松: Visual Basic.NET 数据库开发实例精粹, 电子工业出版社, 2006年5月第一版.
- [19]Kenn Scribner, Mark C.Stiver: 应用 SOAP 实现.NET XML Web 服务, 人民邮电出版社, 2002年5月第一版.
- [20]余于平: 基于 SOA 的 SmartClient 直销系统设计与研究, 中国优秀硕士学位论文全文数据库, 2008年3月11日收稿.
- [21]张永胜, 王强: 基于 SOA 和 Smart Client 的企业应用集成研究, 山东师范

大学学报, 2007 年第 24 卷, 第 04 期, P21~23。

[22] 乔坦, 周建中, 陈超, 郭佳, 陈珩: 基于智能客户端架构的 MIS 系统设计与实现, 微计算机信息, 2007 年 30 期, P6~22。

[23] 刘建粉, 张睿哲, 吕海莲: 基于智能客户端的 ERP 系统设计, 河南科学, 2008 年第 26 卷, 第 05 期, P601~603。

[24] 刘建粉, 吕海莲: 智能客户端在煤电生产企业 ERP 中的应用, 微计算机信息, 2007 年第 23 卷, 第 7-3 期, P27~29。

[25] 李莹等: 基于 .Net 的多模 ERP 技术架构, 计算机工程与应用, 2005 年 8 月, P185~187。

附录

1. 用于实现数据绑定和数据更新的 Web 服务的代码:

```
Imports System.Web
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.Data.SqlClient 加载用到的包
<WebService(Description:="已发布的服务列表",
Namespace:="http://tempuri.org/")> _
Public Class Service
    Inherits System.Web.Services.WebService
    <WebMethod()> Public Function Binding() As Data.DataSet
        定义绑定数据源方法
        Dim con As New SqlConnection("Data
Source=MIAO\SQLEXPRESS;Initial Catalog=ERP\ERPDATA;Integrated
Security=True ") 将连接指向主机中的数据库
        Dim daCust As New SqlDataAdapter("Select * From PO", con)
        Dim ds As New Data.DataSet()
        daCust.Fill(ds, "PO") 数据集赋值
        Return ds
    End Function
    <WebMethod()> Public Function Update(ByVal ds As Data.DataSet) As
Data.DataSet 定义更新方法
        Dim con As New SqlConnection("Data
Source=MIAO\SQLEXPRESS;Initial Catalog=ERP\ERPDATA;Integrated
Security=True ")
        Dim daCust As New SqlDataAdapter("Select * From PO", con)
        Dim cbCust As New SqlCommandBuilder(daCust)
        daCust.Update(ds, "PO")
        Return ds
    End Function
#Region " Web 服务设计器生成的代码"
    Public Sub New() 该调用是Web 服务设计器所必需的
        MyBase.New()
        InitializeComponent()
        Private components As System.ComponentModel.IContainer 以下过程是
Web 服务设计器所必需的
        <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
            components = New System.ComponentModel.Container()
        End Sub
        Protected Overloads Overrides Sub Dispose(ByVal disposing As
```



```

Boolean)
    If disposing Then
        If Not (components Is Nothing) Then
            components.Dispose()
        End If
    End If
    MyBase.Dispose(disposing)
End Sub
#End Region
End Class

```

2. 用于显示、绑定和更新数据库中订单数据表内容的窗体的代码:

```

Public Class Form1
    Inherits System.Windows.Forms.Form


---


#Region " Windows 窗体设计器生成的代码"
    Public Sub New()
        MyBase.New()           该调用是Windows 窗体设计器所必需的
        InitializeComponent()   在InitializeComponent () 调用之后添加任何初始化
    End Sub


---


    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        If disposing Then           窗体重写处置以清理组件列表
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub


---


    Private components As System.ComponentModel.IContainer 以下过程是
Windows窗体设计器所必需的
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents Button2 As System.Windows.Forms.Button
    Friend WithEvents Button3 As System.Windows.Forms.Button
    Friend WithEvents DataGridView1 As System.Windows.Forms.DataGridView


---


    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()           初始化窗体控件
        Me.Button1 = New System.Windows.Forms.Button()
        Me.Button2 = New System.Windows.Forms.Button()
        Me.Button3 = New System.Windows.Forms.Button()
        Me.DataGridView1 = New System.Windows.Forms.DataGridView()
        CType(Me.DataGridView1,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        Me.Button1.FlatStyle = System.Windows.Forms.FlatStyle.Flat

```

```

Me.Button1.Location = New System.Drawing.Point(56, 216)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(75, 32)
Me.Button1.TabIndex = 0
Me.Button1.Text = "绑定"
Me.Button2.FlatStyle = System.Windows.Forms.FlatStyle.Flat
Me.Button2.Location = New System.Drawing.Point(168, 216)
Me.Button2.Name = "Button2"
Me.Button2.Size = New System.Drawing.Size(75, 32)
Me.Button2.TabIndex = 1
Me.Button2.Text = "修改"
Me.Button3.FlatStyle = System.Windows.Forms.FlatStyle.Flat
Me.Button3.Location = New System.Drawing.Point(290, 216)
Me.Button3.Name = "Button3"
Me.Button3.Size = New System.Drawing.Size(75, 32)
Me.Button3.TabIndex = 2
Me.Button3.Text = "生成XML文件"
Me.DataGrid1.DataMember = ""
Me.DataGrid1.Dock = System.Windows.Forms.DockStyle.Top
Me.DataGrid1.HeaderForeColor =
System.Drawing.SystemColors.ControlText
Me.DataGrid1.Name = "DataGrid1"
Me.DataGrid1.Size = New System.Drawing.Size(292, 192)
Me.DataGrid1.TabIndex = 2
Me.AutoScaleBaseSize = New System.Drawing.Size(6, 14)
Me.ClientSize = New System.Drawing.Size(292, 273)
Me.Controls.AddRange(New System.Windows.Forms.Control()
(Me.DataGrid1, Me.Button3, Me.Button2, Me.Button1))
Me.Name = "Form1"
Me.Text = "同步订单数据到本地"
CType(Me.DataGrid1,
System.ComponentModel.ISupportInitialize).EndInit()
Me.ResumeLayout(False)
End Sub
#End Region

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
当点击Button1时调用Service服务的Binding操作
Dim MyService As New Localhost.Service()
DataGrid1.DataSource = MyService.Binding()
DataGrid1.DataMember = "PO"
End Sub

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

```

当点击Button2时调用Service服务的Update操作

```
Dim MyService As New localhost.Service()  
Dim ds As DataSet = DataGrid1.DataSource  
Dim dsChanges As DataSet = ds.GetChanges()  
If Not (dsChanges Is Nothing) Then  
    ds.Merge(MyService.Update(dsChanges), True)  
End If  
End Sub
```

End Class

3. 从服务器端同步数据到本地 XML 文件的代码:

```
Public sqlDataSet As DataSet 定义一个数据视图对象
```

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles Me.Load  
    sqlDataSet = New DataSet 定义一个DataSet对象  
    sqlDataSet.Clear()  
    sqlDataSet= MyService.Binding() 填充数据集的数据  
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click  
    sqlDataSet.WriteXml("E:\ERP\Order\Orderxml.xml") 目标文件路径  
End Sub
```

4. 离线时从客户端窗体获得输入的订单信息并存储在 XML 文件中的代码:

Imports System.Xml.Serialization 引入命名空间,Serialization空间中的类XmlSerializer

Imports System.IO 提供了把对象串行化为XML文档的方法

```
Public Class Order  
    Inherits XmlSerializer  
    Public POID As String  
    Public ProID As String  
    Public ProName As String  
    Public Supplier As String  
    Public Count As String  
    Public POTime As String  
    Public MonType As String  
    Public Price As String  
    Public Address As String 定义了一个XmlSerializer类, 该类声明的每个变量对应数  
    Public Special As String 数据库中订单表的一个字段
```

```
Public Sub OrderFromForm(ByVal Order As Order)  
    Order.POID = Form3.tbPOID.Text  
    Order.ProID = Form3.tbProID.Text  
    Order.ProName = Form3.tbName.Text  
    Order.Supplier = Form3.chSupplier.Text
```

```

    Order.Count = Form3.tbCount.Text
    Order.POTime = Form3.tbTime.Text
    Order.MonType = Form3.cbMONTType.Text
    Order.Price = Form3.tbPrice.Text
    Order.Address = Form3.tbAddress.Text      定义了一个方法，用从
订单输入窗体获得的数据
    Order.Special = Form3.tbSpecial.Text      将Order类实例化
End Sub

```

```

Public ReadOnly Property Datafilename() As String
    Get
        Dim folder As String
        Return "E:\ERP\Order\Order.xml"
    End Get
End Property                                指定XML文件在本地的保存路径
Public Function SaveXmlFile(ByVal filename As String)
    Dim tempFilename = filename & ".tmp"
    Dim tempFileinfo As New FileInfo(tempFilename)  创建临时文件
    If tempFileinfo.Exists = True Then tempFileinfo.Delete()  若
文件已存在则先删除
    Dim stream As New FileStream(tempFilename, FileMode.Create) 打
开文件
    SaveXmlFile(Stream)                        保存XML格式的对象
    Stream.Close()                            关闭文件
    tempFileinfo.CopyTo(filename, True)       重命名文件
    tempFileinfo.Delete()                     删除临时文件
End Function
Public Function SaveXmlfile(ByVal stream As Stream)  重载函数
    Dim serializer As New XmlSerializer(Me.GetType)  返回要保存对
象的类型
    serializer.Serialize(stream, Me)           保存XML文件
End Function

```

End Class

2. 单击订单输入界面的“生成”按钮，将信息提交的代码：

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim order As New Order()                  定义一个订单对象
    order.OrderFromForm(order)              将输入的信息拷贝到对象中
    Dim filename As String = order.Datafilename()
    order.SaveXmlFile(filename)             保存订单
    MsgBox("生成订单成功" & filename)      告知用户
End Sub

```

致谢

此论文得以顺利完成，首先要感谢我尊敬的导师席宁华教授，本论文从选题到完成，每一步都得到了席老师悉心的指导。他严谨的治学态度、渊博的知识和敬业的精神都给我留下了深刻的印象，这些会对我今后的工作生活产生积极的影响。

同时，也要感谢学校、学院、老师们这些年来对我的培养。在论文的写作过程中，我也得到了许多同学的宝贵建议，对于这些曾经关心、支持、帮助过我的老师同学们，在这里也要一并致谢。

另外，我要特别感谢我的家人，他们对我的无私关怀和默默支持是我不断前进的动力源泉。

最后，谨对在百忙中抽出时间评审本人论文的各位专家、老师表示衷心的感谢。

2009 年 4 月

个人简历

个人简历:

李晗昀，女，1985年3月1日生。

2003年9月进入对外经济贸易大学信息学院，信息管理与信息系统专业。于2007年6月毕业，获管理学学士学位。

2007年9月进入对外经济贸易大学继续攻读产业经济学专业硕士研究生，研究方向为企业资源规划与电子商务。