

面向对象的业务支撑平台与建模工具*

实现“应您所需,随时而变”的应用*

业务管理应用软件最佳选择*



LiveBOS 产品白皮书

福建顶点软件股份有限公司

www.livebos.com

Copyright © 1996-2022 版权所有 福建顶点软件股份有限公司

目 录

1. 背景	4
1.1. 信息化企业的困境	4
1.2. 开发者的困境	4
1.3. 软件发展的趋势	5
1.4. 业务架构平台改变管理软件模式	5
2. LIVEBOS 介绍	5
2.1. 平台概念	5
2.2. 平台构成	6
2.3. 平台关键能力	7
2.4. 平台适用环境	7
3. 平台技术特性	8
3.1. 平台支持的运行环境	8
3.2. 系统安全特性	8
3.3. 系统性能	9
3.4. 外部数据交换	9
3.5. 单一平台的多方案部署	9
3.6. 平台展现特性	9
3.7. 多数据库支持	9
4. 系统基础架构	10
4.1. 组织/用户/角色模型	10
4.2. 功能树权限管理	10
4.3. 协同消息服务	11
4.4. 界面方案与菜单布局管理	11
4.5. 系统运行管理与维护	12
5. 对象引擎与业务对象建模	12
5.1. 实体对象概念	13
5.2. 对象属性定义	13
5.3. 对象操作定义	15
5.4. 界面事件逻辑	15
5.5. 对象细分设计	15
5.6. 对象展现定义	16
5.7. 对象继承与子对象	17
5.8. 对象关联定义	17
5.9. 对象模板	18
5.10. 对象视图	18

5.11.	虚拟对象.....	18
5.12.	图表对象.....	18
6.	报表引擎与报表工具.....	18
7.	workflow引擎与设计工具.....	19
7.1.	workflow对象.....	19
7.2.	组织体系与用户选择.....	19
7.3.	workflow流程控制模式.....	19
7.4.	workflow节点的业务处理.....	20
7.5.	workflow界面.....	20
7.6.	workflow监控.....	21
7.7.	流程设计工具.....	22
8.	门户引擎与界面个性化.....	23
8.1.	门户设置.....	23
8.2.	门户内容.....	23
8.3.	门户模板与虚拟门户管理.....	25
9.	MANAGER 运维管理工具.....	25
10.	系统高级开发与扩展性.....	25
10.1.	用户存储过程支持.....	25
10.2.	开发专用的 WebService.....	25
10.3.	JavaBean 支持.....	26
10.4.	JSP 支持.....	26
10.5.	外部数据接口.....	26
11.	应用开发方式.....	26
11.1.	敏捷开发模式.....	26
11.2.	组织建模.....	27
11.3.	业务元素对象建模.....	27
11.4.	业务流程建模.....	27
11.5.	菜单角色界面设计.....	27
11.6.	模型导入、导出与部署.....	28
11.7.	使用辅助开发工具.....	28

1. 背景

1.1. 信息化企业的困境

在企业信息化过程中，可能需要面对大量的问题，在大量企业信息化的实际案例中，最主要的问题是：

个性化与通用性的选择：企业实施信息化有两种路径选择，一是选择市场上成熟通用的软件包，二是采用定制开发模式，由开发商根据客户的业务实际量身定制开发。如果是面向一个特定的应用，一般采用前者模式，而对于业务复杂或需要综合的解决方案时，则选择后者。但这两种模式都有明显的缺陷：前者不能照顾企业业务的个性化要求，而这些个性化的业务，可能完全体现了企业的核心竞争力，后者的开发周期长，费用高昂，同时业务的变化可能给软件带来致命的影响。

业务变更与应用软件的同步问题：软件往往是根据特定时期的业务实际而设计的，但对企业而言，特别是对快速发展的中国企业而言，随着市场的变化与公司发展的不同阶段，公司需要及时调整公司的业务范围和管理流程。这种情况下，管理应用软件如何适应变化调整后的业务是个严峻的挑战。原来与业务匹配的应用软件在这种情况下可能成为业务发展的障碍。这种情况下，企业有时不得不以“推倒重来”的模式冲击新的信息系统，造成投入的巨大浪费。

信息孤岛与 EAI：在企业信息化过程中，“按需建设”的模式可能带来大量的一个个独立的管理应用软件。许多信息化时间比较长的企业，可能有十多个，甚至数十个不同的软件独立运行。这些独立的软件运行在不同的软硬件平台上，采用不同的技术架构，它带来的问题是大量的信息孤岛，部门之间协作困难，维护成本居高不下等一系列问题。而国外企业的EAI 方案是否符合中国的实际是一个重要的考验。不同于国外企业相对成熟、定型的业务与 IT 技术基础落后之间的矛盾问题，中国企业面临的是已经建立的技术层次较高与业务不确定、随时可能变化的矛盾，这种基础（业务）不确定的问题，不是简单面向用户界面层次的 EAI 所能解决的。

信息化项目失败的风险：在中国，信息化项目成功率低的现实一直存在。这里既有客户的原因，也有开发商的原因。不同于国外企业在业务管理完全成熟基础上的信息化，中国企业在业务不成熟但业务却在高速发展的情况下进行的信息化，大量的不确定因素使国内企业的信息化失败案例屡见不鲜。这不仅出现在本地化的系统解决方案中，也出现在包括大量引入国外已经成熟的应用项目中。

1.2. 开发商的困境

作为企业信息化的开发商，在面对客户的信息化项目中也面临困惑。

需求困境：在实施企业信息化项目中，开发商面临的第一个也是最大的挑战是如何把握客户的需求。通过大量前期的需求调研，形成文档化的客户需求报告是否代表了客户的真正需求？实际上，这里存在两个层次的问题：1. 客户是否完全清楚其对未来才能上线软件的真正要求；2. 文档化的需求是否已经完全表达了客户所表述的需求。实际的经验中，这两方面都存在巨大的“黑洞”。据此开发的软件，往往与客户的真正需求相距甚远。这时开发商面临这样的选择：1. 让客户为其签字画押的需求负责，这样可能激怒客户，甚至导致项目的最后失败；2. 重新修改需求，再次调整开发，这样面临工期延长、成本上升、人员波动等困境。

“人月神话”：这是一个软件开发的著名论断。每个软件开发商都得面对各种各样无法预知的变化因素：客户需求的变化，技术环境的变化，关键人员的变动等等。这些变化直接影响了软件交付的周期，带来开发成本的增长，同时也影响了合同的执行与客户的满意度。

技术抉择：在快速变化的技术面前，如何选择技术架构，中间件，数据库，编程语言，都是应

用软件开发商要面对的问题，选择如何在其原来的技术基础，客户的需求，技术发展的方向三者之间进行平衡，是开发商在技术抉择上的一个难点，如果选择支持多样的技术平台，可能要面临高成本的多平台支持困境。

人员流动与积累困境：应用软件开发的最大价值在于对客户业务的深刻理解，这种理解最终以软件的形式体现出来。不断的客户（行业）经验的积累，形成了应用软件开发的核心竞争力之一。而核心人员的流动，带来的最大挑战是知识经验的流失，在文档（实际上文档也难以完全体现）与软件没有完全同步时，由于许多细节的业务知识只能体现在大量的程序代码上，新的人员往往得通过研究大量的代码来继承原来的业务知识。

1.3. 软件发展的趋势

从计算机软件的发展历史来看，每个新层级软件的出现都带来软件开发模式，软件功能与适应性的巨大提升。从最早的操作系统，到后来的关系型数据库，再到中间件的出现，都给软件的发展带来巨大的影响。这些系统的出现在不同层面上简化了软件的开发，提升了软件的开发效率。

从管理软件的趋势上看，从国际性管理应用软件企业的大购并可以看出，“整合”软件正在替代单项的应用软件，而更加柔性的应用正在替代僵硬的软件包。Peoplesoft 创始人戴夫-杜菲尔德也认为：“软件组织必须改变以前僵硬型 ERP 系统的做法，并提供更为新型的单点解决方案(point solutions)。目前流行的单点解决方案只提供权宜性解决之道，而缺乏应有的灵活变通性。而各大企业如今已经有了更高需求，他们希望能实现产品定制化，同时还应实现与第三方产品的无缝整合。”

1.4. 业务架构平台改变管理软件模式

业务架构平台或业务基础平台是软件发展的最新层级，相对于原来的技术中间件，它是一种业务层次的中间件，它完全实现了管理技术与业务功能的分离。在业务架构平台下，管理应用软件的开发，直接采用业务建模方式实现，大大减少了实现代码，直至完全实现了零代码的应用开发。

业务架构平台的发展，对企业信息化的客户与开发商而言，从高层次上解决了各个大困境，也代表了管理软件“集成化，协同化，个性化，柔性化”的发展方向。

2. LiveBOS 介绍

2.1. 平台概念

LiveBOS 灵动业务架构平台是福建顶点软件股份有限公司开发的业务架构平台。

LiveBOS 面向管理应用，包括运行支撑软件平台及其快速开发工具两大部分。在 LiveBOS 支持下，管理软件可以实现完全的“业务驱动，用户导向”软件开发模式，并实现“应您所需，随时而变”的应用。

作为“业务架构平台”，LiveBOS 最大的特点是面向业务和技术无关性。不同于一般的应用中间件，构件库，或软件代码辅助代码生成器之类的系统。它是一个业务架构运行操作系统，在它的支撑下“描绘”不同的业务模型，即可实现不同的业务应用。而调整业务模型定义，即可立即改变系统功能。它与具体的技术平台无关，目前 LiveBOS 支持所有的主流操作系统、数据库与 WEB 中间件，所有的用户界面为 Browser。

LiveBOS 集成了业务模型运行平台与业务建模平台。在 LiveBOS 支持下，用户只需要基于业务和管理的层面，而非技术的层面来理解、设计、构架和集成企业的信息系统（基于业务层面是指开

发人员只需描述企业的组织机构、业务流程、业务信息、业务资源、业务逻辑、业务事件等业务内容，而不考虑技术层面的东西)，就可以实现各类基于 **WEB** 的高层次信息化应用。而且，用户可以随时在运行中重新定义或调整模型，从而达到使自己的信息系统完全贴近不断变化的业务，这也是“灵动”的价值体现。

2.2. 平台构成

LiveBOS 平台包括运行支撑平台与建模工具两个部分。

其中建模工具包括：

- 对象建模设计器，支持快速配置的对象设计工具
- 工作流设计器，提供了图形化的流程设计功能，
- 报表设计器，可以设计辅助界面的各类报表。
- 辅助工具，主要包括数据库转化工具，文档生成工具。

运行支撑平台是本系统的核心，它包括 5 大组成部分：

- 公共基础架构与服务组件：包括系统的基本运行架构，多模式菜单支持，用户、组织与角色管理，消息协同服务，系统管理与配置服务等，它整合了 LiveBOS 上应用的基础功能。
- 对象模型引擎：实现对各种类型的对象模型（由对象建模设计器生成）的解释执行，它是 LiveBOS 的核心组件，支持各种实体对象，对象视图，虚拟对象等全面的支持。它具有对象管理，对象逻辑处理，对象展现控制，对象权限控制等功能。
- 报表引擎：通过访问对象模型的数据，按照报表定义模型的解释执行，生成报表。
- 工作流引擎：解释执行“工作流设计器”生成的流程定义，实现不同业务流程环节的控制执行。
- 门户引擎：提供不同用户个性化的功能，LiveBOS 应用软件的用户，可以设置自己的个性化 **WEB** 页面，在同一页面上，部署自己所需要的操作，对象信息，工作流，报表等。

在本白皮书后面有运行平台实现功能的具体说明。

2.3. 平台关键能力

从 LiveBOS 的应用开发过程来看，其特点是：

- ✓ **敏捷开发 Agility**：业务驱动，敏捷开发，持续改进是 LiveBOS 平台的开发特点。“模型即软件，建模即开发”是 LiveBOS 最重要的特征。它可以从根本上解决开发过程中的“需求困境”，LiveBOS 支持敏捷开发模式，它允许在开发过程的任何阶段提出需求改进。通过快速循环改进模式，提升软件功能。
- ✓ **快速高效 Efficacy**：LiveBOS 大大提升了应用的开发效率，极大降低了开发成本。整个 LiveBOS 应用的开发过程就是业务建模的过程，它完全替代了一般开发模式中，系统设计，数据库建模，程序设计，代码测试等工作量十分大并且需要大量循环完善的工作。一般管理软件的开发周期可以减少 70%以上。
- ✓ **简单易用 Simplicity**：LiveBOS 的开发导入周期短，有管理应用设计经验的人员，一般经过 2-3 天的体验培训，就可完全掌握 LiveBOS 的开发。它提供的建模工具简单易用。平台提供了高度提炼的，贴合实践的业务概念来描述模型，这样完全屏蔽了技术实现细节。并且在所有的模型定义过程中，可以实时查看设计的结果。
- ✓ **标准化 Standardization**：LiveBOS 完全采用标准架构，而非专用的技术，同时实现跨平台支持，包括各主流的商业数据库与开源数据库，各主流的服务器与开源服务器。在 WEB 端也采用标准的技术，无专用的界面插件。LiveBOS 提供了 Webservice 的支持，同时也支持各类标准的数据交换接口。
- ✓ **可管理性**：LiveBOS 简化软件开发的流程，降低管理的复杂度，同时大量代码编程的工作压缩，也减少了开发团队的规模，项目的可管理性得到改善，同时项目的风险也大为减小。

从用户角度上看，LiveBOS 上的应用有自己独特的价值：

- ✚ **灵动 Agility**：应您所需，随时而变。基于 LiveBOS 平台，所有的业务管理模块，都可以根据企业的不同组织模式，业务特点，流程需求，进行快速调整，这样企业就可以得到完全个性化的，体现企业特色的应用系统，而不是一个僵化的、需要“削足适履”的应用。同时系统可以随企业业务的发展变化动态调整，实现系统与业务同步发展。
- ✚ **集成 Integrated**：单一平台，多样应用。在统一的平台上集成一个企业的关键管理应用，形成“管理应用支撑平台”，可以简化 IT 架构，简化管理，降低 TCO，同时消灭“信息孤岛”。
- ✚ **个性 Personality**：单一门户，多样视图。所有的用户，包括多样的内部员工，以及外部的客户，代理商，供应商，都能在同一门户下访问信息系统，同时在门户的支持下，不同角色，不同用户都能获得个性化信息视图。
- ✚ **协同 Collaboration**：数据关联，流程同步，及时消息，全员协作。LiveBOS 应用，实现三个体系上全面协同：数据关联，流程同步，消息沟通。LiveBOS 的应用在整个组织的层次实现所有内外部用户的全面工作协同。

2.4. 平台适用环境

LiveBOS 平台更加适合以下几个方面高标准、高要求的管理信息化应用：

1. **灵活调整和随需而变** 当业务发展和变化时，需要随需而变，对信息系统调整的响应速度要求很高的用户。
2. **业务深度集成** 信息系统比较复杂和多样化，容易产生信息孤岛，希望业务系统很好整合的用户。
3. **复杂业务协同处理** 需要实现全国或全球范围分支机构的跨部门多角色间进行复杂业务协同交互处理的用户。

4. **复杂 workflow** 业务 workflow 复杂，或流程管理、监控、运作要求较高、随需变动的用户。
5. **要求信息化统一规划** 信息系统比较复杂，希望业务基础架构统一、资源共享、系统一致性的用户。
6. **跨技术平台** 需要发布在不同技术平台上，或者在未来需要无缝迁移的用户。
7. **用户主导和主控** 具有独特业务模式和发展战略，信息系统难以直接外购或需要不断需求进化二次开发的用户。

3. 平台技术特性

3.1. 平台支持的运行环境

LiveBOS 平台支持几乎所有的硬件平台与软件环境。软件包括各类开源平台与商业系统。具体见表 3-1:

表 3-1 平台运行环境

平台	开源系统	商用系统
操作系统	Linux	Windows 以及统信、银河麒麟等国产化 OS
数据库	MySQL	Oracle、SQL Server 以及达梦、人大金仓等国产化数据库
WEB/应用服务器	apache/Nginx/Tomcat	东方通等国产化中间件

3.2. 系统安全特性

LiveBOS 平台在技术层次上具备较高安全性。

登录安全管理：在身份认证方面，LiveBOS 支持多种方式的安全性高的身份认证机制。它可设置，支持多种层次的登录安全控制。

1. 用户账号+密码的安全控制
2. 用户账号+密码+随机码安全登录控制
3. 用户账号+密码+软件数据证书
4. 用户账号+密码+随机码+数字证书（USB-key 硬件数据证书）
5. 用户指纹认证+密码

数据传输安全：LiveBOS 支持采用 SSL 协议建立安全通道来保护授信应用系统的数据，SSL 协议是业界的实际标准。而且对 SSL 协议的使用均采用安全性最高的双向身份认证的密钥协商，密码算法也采用高安全强度的国际标准算法。

操作审计：LiveBOS 的对象数据，可以设置为安全审计模式，这样任何对数据的操作，将有完整的审计日志，这样可以在事后进行操作审计。

权限管理：LiveBOS 提供了完整的权限控制，可以定义数据的操作权限，授权权限。权限可以精确控制到数据记录分类与字段级。另外，LiveBOS 平台的权限管理模式支持“申请一审核”模式，它使任何用户权限的变动，都需要通过独立的“安全管理员”进行复核后方可生效。

3.3. 系统性能

LiveBOS 平台的性能依赖于基础的技术平台。在应用服务器上，可以通过服务器集群实现系统的高度伸缩性。

LiveBOS 平台通过内部各个引擎，通过大量采用的常用数据缓冲技术，业务模型的首次解释，持续引用的模式，实现了平台自身的高性能。

LiveBOS 大量的优化设计，确保了大数据量与复杂业务应用的系统性能。

3.4. 外部数据交换

LiveBOS 平台支持将系统内的数据，如报表直接转为 Excel 文档、Word 文档、PDF 文档、HTML、XML 等几乎全面的主流数据文档格式。同时 LiveBOS 支持从 Excel，XML 文件中导入数据。

为了实现体系应用系统的实时业务处理，LiveBOS 以WebService 方式提供访问其数据与操作的接口。

3.5. 单一平台的多方案部署

LiveBOS 平台支持在单一平台上部署不同的应用，用户登录时，可以选择自己需要登录的系统。LiveBOS 多方案部署的支持，有利于在同一服务器上更加便捷地实现多个不同的应用，同时简化了多个系统的管理与维护，并提升 IT 资源的使用效率。对ASP（应用软件供应商）而言，多方案的支撑更具商业价值。

3.6. 平台展现特性

LiveBOS 平台支持跨浏览器进行浏览，支持Edge、safari、Google，FireFox，360浏览器等主流浏览器。

LiveBOS 平台界面上所有的操作完全基于 Ajax 方式实现局部刷新的功能，将实体对象字段输入检查校验直接在客户端进行，在用户输入完成后，即可进行自动校验，从而减轻了应用服务器的压力。

3.7. 多数据库部署支持

LiveBOS 平台支持在单一平台上进行多数据库部署，可以将相关的表数据分别部署到不同的数据库中。这样使得系统的数据压力分散到多个数据库中，实现平台的系统数据和业务数据的分离，以减轻数据库服务器的压力。同时使得系统拥有更好的信息集成能力。

4. 系统基础架构

系统基础架构是 LiveBOS 的基础支持系统，本章具体介绍其实现的功能。

4.1. 组织/用户/角色模型

对任何一个 LiveBOS 上的应用，都应该有一个明确的组织体系架构。LiveBOS 应用在组织架构体系上进行用户、角色（岗位）的管理。用户的权限管理是基于组织体系、角色（岗位）上进行的。

组织机构是一个多层次的树状结构。角色是一个单层次的平面结构，可以按职能或任务进行分类，一个角色下可包含一个或多个人员。角色可以挂接在组织结构多个单元上。

人员是指可以使用 LiveBOS 系统的用户，可以是员工，客户或合作伙伴。一个用户可以有一或多个角色，即人员与角色是 n:n 关系。用户必须通过角色挂接于组织中，即为组织单元分配用户时必须选定用户的角色。

组织模型如图 4-1 所示：

组织模型

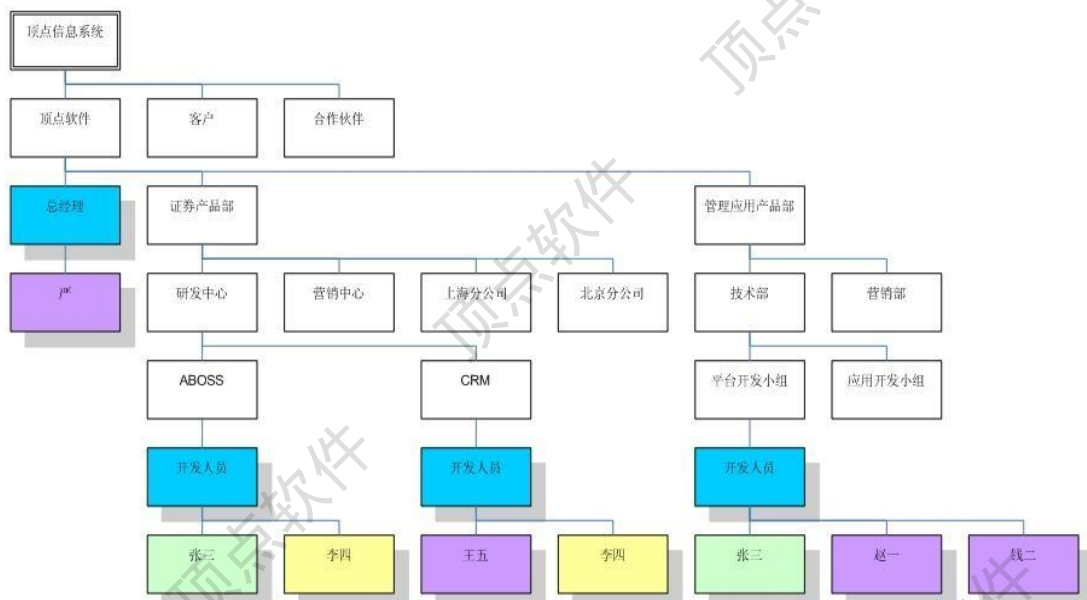


图 4-1 组织模型图

4.2. 功能树权限管理

LiveBOS 引入了一种新的权限管理方式，功能树权限管理。在这种模式下进行权限管理时，管理的最小因子是功能项，功能项中包含了完成一系列功能所需要的操作。比如一个员工基础资料维护的功能项，包含了 员工基本信息的新增/修改/删除，同时也包含了和这个员工有关联的 员工固定资产管理的新增/修改/删除，用餐记录的新增/修改/删除等。在进行授权时，只要授予这个员工基础资料维护这个功能项权限，则拥有了功能项中包含的所有权限。



图 4-2

功能项的维护和定义可以通过应用开发人员在 LiveBOS Studio 中进行定义。

功能树权限管理模式下同样也可以实现权限的分级授权。同时也可以支持临时权限的授权和管理。

支持授权的时候，角色权限的互斥功能。

所有权限的授权，可以设置为必须经过安全管理员的授权，方可正式生效。

4.3. 协同消息服务

基础平台提供了基本的消息协同服务。LiveBOS 平台基础的消息服务包括：

1. 邮件提醒服务
2. 短信提醒服务
3. APP在线提醒服务
4. APP离线提醒服务
5. 企业微信提醒服务
6. 公众号提醒服务
7. 小程序提醒服务
8. 钉钉提醒服务
9. 桌面即时通讯提醒服务

在对象操作、工作流操作中，可以直接调用这些信息服务，实现工作协同。

4.4. 界面方案与菜单布局管理

LiveBOS 提供了多种类型的菜单界面布局方案，用户可以通过修改界面布局和界面主题 2 种方式自由组合配置，修改应用程序的界面布局和菜单方案的展现，从而定制自己喜欢的个性化主题界面风格：

界面布局：

1. Windows 风格下拉菜单
2. 左侧树形风格菜单

3. Outlook 风格菜单
4. 标签风格菜单
5. 标签+树形风格菜单
6. 标签+Outlook 风格菜单

界面主题：

1. 简约风格（默认）
2. 银灰风格
3. 商务风格
4. 红色风格
5. 绿色风格
6. 浅蓝色风格
7. 灰蓝色风格

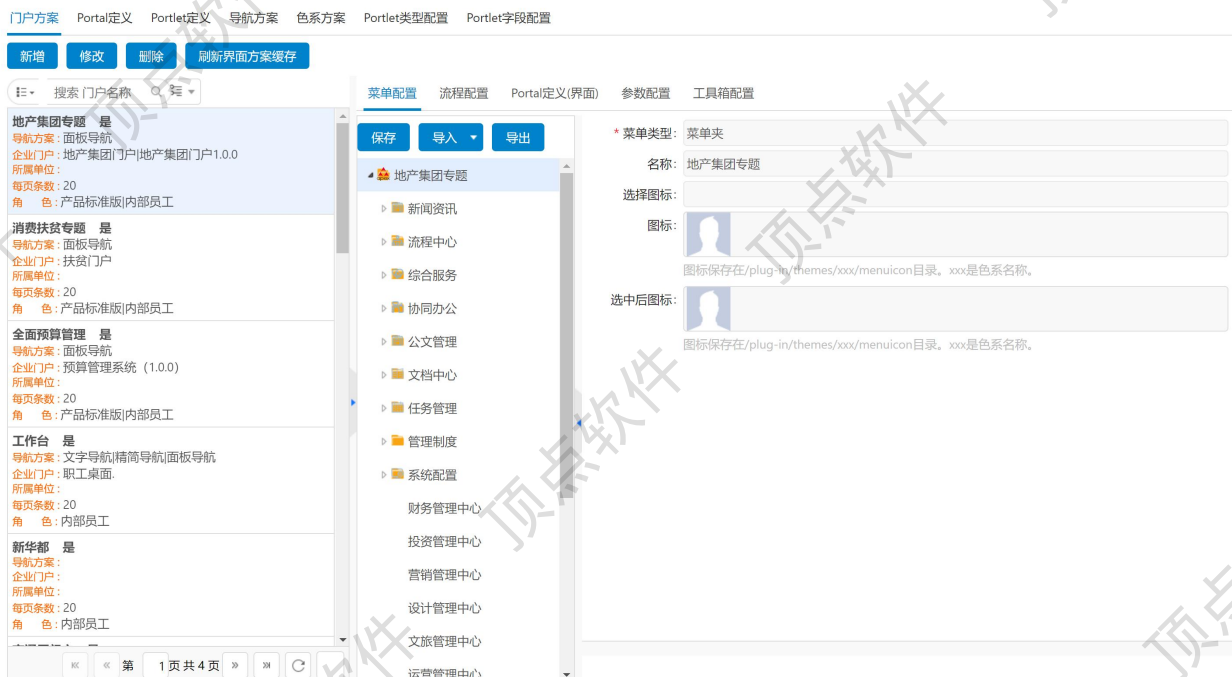


图 4-3

除了界面菜单方案外，LiveBOS 的门户可以由每个用户自行定义，实现集成的、个性化的界面布局。

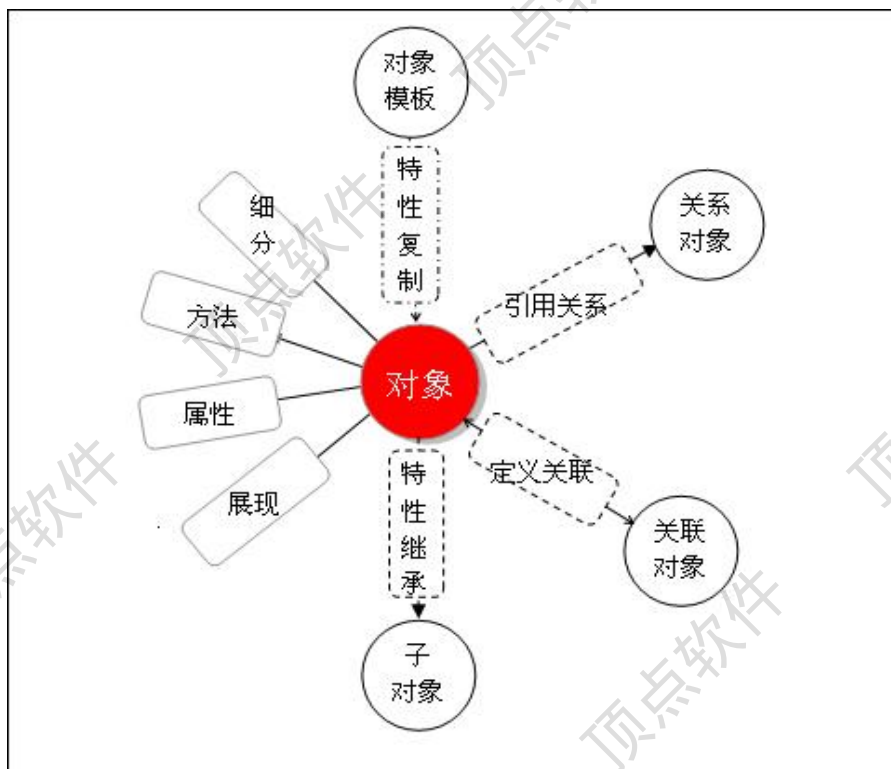
4.5. 系统运行管理与维护

LiveBOS 内建的系统管理与维护功能包括：

- 系统参数管理
- 用户管理
- 用户、角色与组织权限管理
- 功能权限授权查询
- 数据字典管理
- 数据导入/导出工具

5. 对象引擎与业务对象建模

LiveBOS 平台上的应用开发，主要工作就是业务对象建模，或称为业务元素建模。一个完整的对象模型如图 5-1 所示。



5.1. 实体对象概念

实体对象：指具有特定属性，特定操作管理方式的信息记录。它在数据库中体现为一个“表”，也可能由一个以上的表来存储。在 LiveBOS 中不同的场合，对象有两类含义，一个是指对象的所有记录集，一个是指一个对象记录。可能的对象类似：客户，合同，项目等。

使用 LiveBOS 对象设计器，选择“新建”，输入对象名称，选择对象的管理分组，即可创建一个新的实体对象。（注：对象的分组是为了便于对创建对象的分组管理而设计，开发者可以根据业务主题创建不同的对象分组，如“客户关系管理”，“项目管理”等）

5.2. 对象属性定义

类似于一般数据库表设计的字段定义，对象建模的第一步就是定义其属性。数据库表字段对应于数据库中物理字段，但 LiveBOS 中对象属性并不一定与一个数据库的字段对应，如虚拟列，它在数据库中并不存在。另外 LiveBOS 对象属性的类型并不等同于数据库的类型，LiveBOS 的对象属性是一种“高级”业务层次的属性描述，如“内部对象”的对象属性，在 Oracle，MS SQLSever 等数据库中并不存在该类型的字段。

LiveBOS 的对象属性可以充分体现业务的实质，使得对象之间的关联关系，操作模式管理变得十分容易。同时，“泛对象”类型的支持，更加高效地实现信息之间的多样性关联。比如一个文档管理功能，它希望能管理到文档可能相关的产品，相关的客户，相关的员工，相关的项目，相关的会议等等。在一般软件中，软件只能通过一一设置大量的冗余字段来维护这些文档相关信息，而在 LiveBOS 中，只需要定义一个泛对象的属性，如“相关信息”，并选择该属性可能保持的对象类型即可。在用户使用文档管理功能时，即可以在“相关信息”中保持所有可能涉及的产品，客户，项目，会议等信息。

目前 LiveBOS 支持如下对象属性的类型：

表 5-1 对象属性的类型

	类型	说明	长度
简单类型	字符	一般的文字	1-5000
	数据	包括整数与小数	1 位以上
	日期	表达日期与时间	
	密码	特殊加密文字	
特定类型	身份证	符合身份证规则的字符串	
	Email	符合 Email 地址规则的字符串	
	URL	符合 URL 地址规则的字符串	
高级类型	选择项	用于表达可罗列的情况中的一种情况。必须事先在数据字典中定义	系统预设
	多值选项	用于表达可罗列的情况中的一种或多种存在的情况。必须事先在数据字典中定义	系统预设
	内部文档	用于保存文档的字段，可以上传或下载	
	外部文档	用于保存文档的字段，可以上传或下载，但其实际保存在数据库外。	
	多文档	用于保存多个文档的字段，可以同时上传或下载多个附件文档。其实际保存在应用服务器	
	内部对象	表达内部的其他指定类型对象	
复合类型	泛对象	表达一个不事先约定类型的对象	
	多值对象	表达多个不事先约定类型的对象	

对象属性除了有类型差别外，还有**字段类别限制**（包括预计算列，绑定列，虚拟列）和允许空值，禁止重复和可用性（包括是否显示，只读）。

其中预计算列，绑定列和虚拟列的属性，都可以通过定义公式确定其取值。他们的不同之处在于以下几个方面：

A. 支持操作上的不同：预计算列只能在新增操作下才有效，并且在“新增”操作“确定”之前就已经计算出结果。而绑定列和虚拟列则只有事件触发或者操作“确定”之后才进行计算。

B. 列的编辑属性不同：绑定列和虚拟列是不允许用户进行修改的，它最终的值是根据表达式计算得到。而预计算列在计算出结果之后，可以允许修改。

C. 在数据库中的物理存在形式不同：绑定列和预计算列在实体对象对应的数据库表中，表现为物理表中具体的列，而虚拟列在物理表中并不存在，只是在显示，并且引用时由 LiveBOS 平台进行实时的计算取得。

LiveBOS 对象工具提供了计算公式定义功能，如“金额”=“数量*价格”，计算公式具有强大的定义能力，它可以引用本对象的属性、相关对象的属性、系统的全局属性等作为计算元素。每个属性可以定义其默认值。

为了让有许多属性的对象在展现时更加清晰，开发者还可以定义字段的分组，如“客户基本信息”，“客户联系信息”等，这样在对象显示时，对象的属性可以自动显示分组属性。

为了简化部分对象的属性定义，LiveBOS 引入了“元数据”的概念，开发者可以预先定义一系列的对象属性，如“电话号码”，“联系地址”等，这样的元数据可以被任意多个对象属性定义时直接采用，这样的优点是多个对象使用同一属性时具有一致性。

对象属性的定义，LiveBOS 可以自动为每个属性分配内部代码，开发者也可以自己定义属性的代码。

5.3. 对象操作定义

类似 C++ 与 Java 语言中类的方法，LiveBOS 对象也可以定义自己的操作。针对每个新建的对象，系统为其设计了新增,删除,修改,导入,导出,统计,打印等操作。开发者可以定义自己的对象操作。

对象操作是完成业务逻辑处理的重要手段。对象操作的设计包括自定义用户方法，对象流程方法和向导方法。

其中用户方法的设计包括操作的启动条件，逻辑处理，例外处理等。对象操作的逻辑处理部分，既可以使用 LiveBOS 平台提供的各类函数，调用预定义的服务（如协同消息），也可以直接使用 SQL 语言，数据库的储存过程，Java 语言。这样大大强化了对对象操作的处理能力。

对象流程方法，逻辑处理方式与用户方法类似，与用户方法不同的地方在于，它是通过定义一系列的节点实现对象复杂逻辑的处理，节点类型分为控制节点和操作组件节点。控制节点主要用于流程的基本控制，包括判断、循环、迭代等常用的控制类型；操作组件节点为 LiveBOS 提供可供调用的组件相关操作，包括对象相关操作、执行工作流动作、调用存储过程、执行 SQL 语句、调用系统服务、执行脚本、调用消息确认框、Fix 操作以及调用其他对象业务流程等操作。

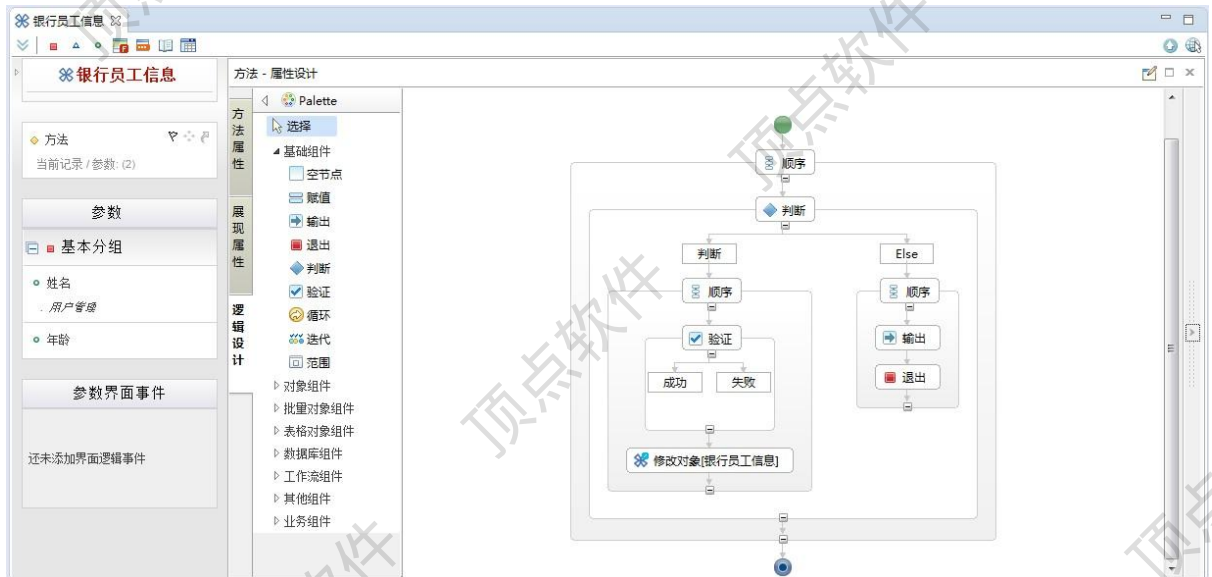


图 5-2

而向导方法，是以向导模式提示用户在当前页面进行参数的输入，提供给用户进行操作，其逻辑的主体设计仍是在对象流程方法中完成。例如，我们在进行邮箱注册的过程中，可以按照指示逐步执行下去，从而告诉用户当前正在进行哪些操作、下一步将要执行哪些操作，页面提供选项供用户进行填写或者选择，从而完成注册操作。

5.4. 界面事件逻辑

界面逻辑事件设计和对象流程方法的设计相似。界面逻辑事件支持 2 种类型。一种是界面初始化的逻辑事件，当界面进行初始化时触发事件；一种是普通字段或参数的逻辑事件，当字段值发生变化时触发相应的逻辑事件。并且支持复杂的逻辑判断。

通过这个功能，可以更好地完成对用户交互界面的支持。

5.5. 对象细分设计

对象细分管理是通过对满足指定条件的对象，进行命名、定义其筛选条件。对象细分的定义，

简化了信息管理应用。如客户对象，可以细分为 VIP 客户，潜在客户，我的部门客户，我的客户等细分。

对象细分可以完全代替对象被使用，如：直接对对象细分进行授权，菜单中引用对象细分，在对象属性定义时，所以“对象”类别的属性可以直接应用其他对象的细分。如合同中的属性“业务主办”，可以直接引用“员工—销售人员”这样的细分，这样输入“业务主办”时，只能选择“销售人员”，而不能选择技术人员。

5.6. 对象展现定义

对象展现包括：1.对象浏览界面；2.对象操作界面；3.对象关联界面；4.分组标签；5.对象的引用显示与输入标识。

一个典型的对象浏览界面如图 5-2。它包括对象数据的表格，对象快速分组栏，关联对象链接栏，对象操作栏，对象快速搜索栏，对象标题与栏目控制栏。

LiveBOS 提供了多样化的浏览模式：表格浏览模式，表格主从模式（图 5-2 模式，主对象表格与关联对象多标签显示），列表主从模式，单记录浏览模式，单记录主从模式，卡片浏览模式，卡片主从模式等。

对象展现定义时，可以设定默认的展现方式，允许切换的展现方式，它为对象信息展现提供了丰富的表现形式。

浏览模式下，也可以通过设置，使对象表格具备类似 Excel 的全屏幕编辑功能。

对于对象操作（如新增，删除，编辑）的表单展现模式，可以选择“嵌入模式”与“弹出窗口”模式。可以设定数据分栏数量，熟悉标签与输入框的组合模式（左右，上下）等。

如果需要精细设计表单，LiveBOS 提供了深入定义表单界面的功能。

在界面上，还可以设计出现的关联对象及其显示名，出现的操作及其显示名，出现的快速分组标签的类别等。

LiveBOS 对象中“输入标识”与“显示标识”是两个重要的概念。它们都可以在对象设计中设定其为某个对象的属性。

概念

输入标识，是指引用该对象的一个记录时，用户是通过哪个属性来输入，并进行定位的。如员工中的“工号”，“身份证号”都可能被选择为输入标识。

显示标识，是指该对象在被其他对象引用，需要显示的属性。如员工姓名、客户名称等，一般会被分别设定为员工对象、客户对象的显示标识。



图 5-3 一个典型的对象浏览界面

5.7. 对象继承与子对象

在业务信息描述中，同一类别的业务元素既有共性，也有个性。如合同，技术服务合同与设备销售合同，具体公共的信息，如：合同名称、甲方、乙方、合同金额等。也有不同的信息，如：技术服务合同中可能涉及的服务期限、服务人员、服务内容等。而设备销售合同中，需要有多条记录的设备的产品名称、规格、单价、数量等。

在一般软件开发中，有两个方案处理合同管理：1.采用并集模式，建一个表来保存各类合同中所有的信息，这样字段冗余多，信息不严谨。2.为不同的合同类型建立不同的表。这样管理复杂，分析统计困难。

LiveBOS 的对象支持对象继承，从一个对象上可以继承出多个子对象，子对象继承其父对象的属性、操作等属性。同时可以有自己的属性、操作、展现、细分等。

子对象还可以重定义父对象计算列的逻辑，如合同金额的计算等。

LiveBOS 在浏览父对象时，只展现父对象的属性，但单记录显示时，可以显示相关子对象的完整信息。同时在父对象上新增对象时，可以选择子对象之一或是父对象本身。

LiveBOS 对象继承功能，超越了一般关系型数据的局限，大大方便了管理软件功能的实现。

5.8. 对象关联定义

对象关联定义是用来表达两个对象之间 $n:m$ 的关系。如会员与协会，一个会员参加多个协会，一个协会有多个会员。通过建立对象关联，可以实现会员与协会之间的关系管理。

对象设计时，可以独立设计“会员”与“协会”对象，然后创建两者之间的“对象关联”。

LiveBOS 平台自动在“协会”对象上增加“添加会员”，“删除会员”等关系维护操作。在“会员”对象上同样也会增加相应的关系维护操作。

在“主从模式”展现对象时，关联对象将自动作为被关联对象“页标签”信息显示内容。

5.9. 对象模板

为了简化特定类型的对象，LiveBOS提供了对象模板的支持，如：层次对象模板，新闻对象模板，文档对象模板，试卷对象模板，投票对象模板等。开发人员在此对象模板上定义的对象，将继承对象模板所具备的全部特定属性、操作、展现模式等。

层次对象模板，用于表达有层次关系的信息，如公司的部门组织体系，文档的目录体系等。层次对象一般以树形方式展现。层次对象关联其他对象时，可以选择是否展现该节点以下所有节点的关联对象。如查询选择部门的员工时，是否除了本部门的直接员工外，还要包括所有下级部门的员工。

新闻对象模板，该模板以主题方式展现，同时可以有相应的评论，新闻对象可以自动记录被访问的次数，评论的数量等，新闻浏览时，还支持置顶，推荐，精华等评级功能。并且新闻模板还提供了新闻分类管理的功能，允许对每一个新闻的分类项进行控制，如分类的排序方式，是否允许评论，是否允许匿名发布，是否允许匿名评论等。

试卷对象模板，该模板建立试卷，配合题目模板使用，从题库中获取数据作为试卷的题目展现。

投票对象模板，可以让用户快速地对相关主题以投票形式进行调查，投票选项可单选也可多选。

5.10. 对象视图

类似关系型数据库的“视图”，对象视图是建立在某一对象基础上的一个虚拟“对象”，它没有自己真实的数据，数据完全在对象上保存。

视图要定义记录的筛选条件，同时还可以选择展现的属性（字段）。

对象视图可以定义自己的操作和展现方式。

5.11. 虚拟对象

虚拟对象类似于实体对象。同实体对象一样，可以定义对象属性和对象操作等。但其数据是通过定义复杂的查询语句或存储过程来获取的，通过字段映射关系将数据投影到对象属性中。

5.12. 图表对象

LiveBOS图表对象是通过数据记录集构造的图表类型的对象，将数据以可视化图表的形式呈现，便于进行数据的统计分析。目前支持的图表类型有饼图、折线图、分类折线图、柱状图、分类柱状图、堆叠柱状图、仪表盘、表格等。

LiveBOS不仅支持二维图表的展现，还支持三维的图表展现，例如统计不同公司不同岗位的人数，并将数据集成到一张图表上，这里就需要使用三维图表（分类柱状图、分类折线图、堆叠柱状图）来展现。

6. 报表引擎与报表工具

LiveBOS提供对象基础数据查询与统计报表功能，开发人员与用户可以方便地设计出各类统计报表(如图6-1).图形化设计工具可以帮助用户方便快速地调整报表格式.LiveBOS报表支持直接打印，支持多种方式输出，如输出成Excel文件，PDF文件，Word文件等。

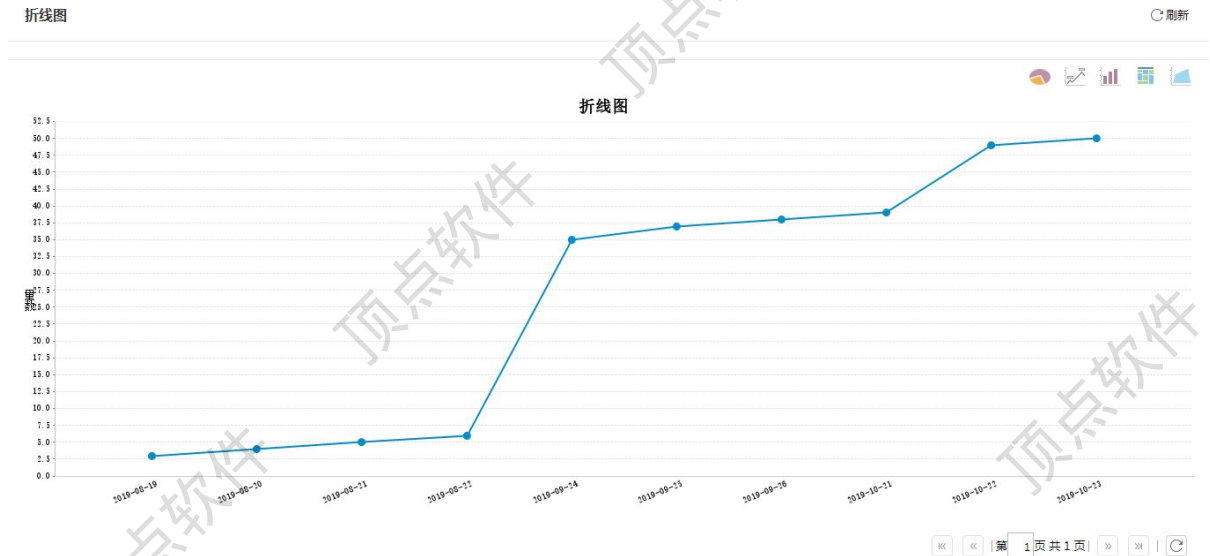


图 6-1 统计报表

7. workflow引擎与设计工具

工作流是 LiveBOS 实现复杂业务流程处理，实现跨组织，多角色人员之间协同工作的关键模块。工作流模块包括工作流设计工具与工作流运行平台（工作流引擎与工作流客户端）。

LiveBOS 的工作流不同于一般工作流，它与 LiveBOS 对象紧密结合，完成信息管理与事务处理的协同。

LiveBOS 工作流引擎是应用控制和运行的中心，负责解释、控制并协调各种复杂工作流程的执行并且同步各个客户端的反应，通过与对象引擎与通信服务模块的配合，实现协同的事务流程执行。

7.1. 工作流对象

LiveBOS 的工作流与 LiveBOS 对象模型紧密结合，每个工作流都对应于一个主工作流表单，以及若干个从属的工作流表单。

工作流参与者的操作界面。完全类似对象的操作界面。

通过直接使用工作流对象，LiveBOS 应用系统可以使用到平台提供的强大信息管理功能，以及所有实体对象具备的强大业务处理功能。

7.2. 组织体系与用户选择

LiveBOS 直接采用 LiveBOS 平台提供的组织体系来管理每个流程环节的参与人角色与用户，LiveBOS 工作流提供了多样化的参与人选择模式，参与人可以是独立的组织节点，角色，用户组，系统用户，也可以是某组织节点下的角色。还支持动态用户组，如发起人部门的某角色等。

7.3. 工作流流程控制模式

LiveBOS 流程支持顺序转型，按条件转向，流程分支，合并，机器执行。在每个流程节点还支

持单人单审，多人单审，多人并审，多人串审等多种审批方式。

7.4. workflow 节点的业务处理

LiveBOS 工作流每个流程节点或动作中加入预处理和后处理功能。可以在这些处理中加入调用系统服务功能，如发送邮件，发送短信等。还可以加入表单对象操作或外部对象操作。

7.5. workflow 界面



图 7-1



图 7-2

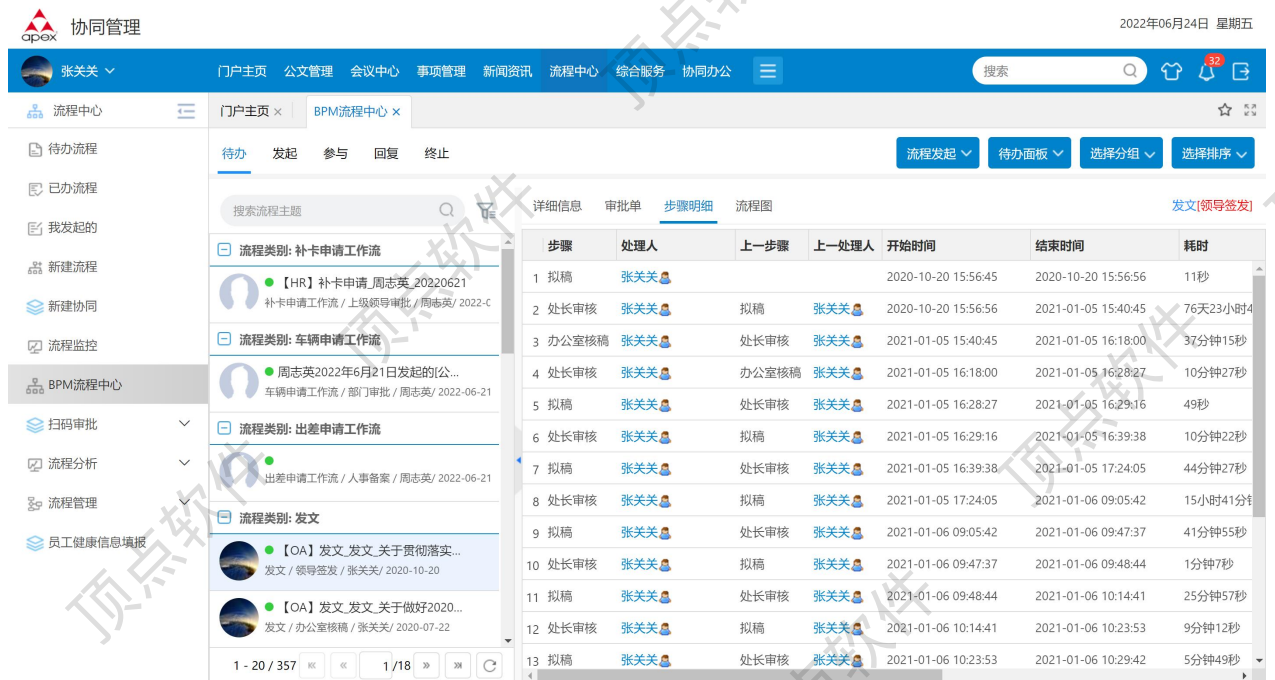


图 7-3



图 7-4

7.6. workflow 监控

LiveBOS 提供流程监控功能，能跟踪各项工作的进度，了解所有工作的执行状况。并为流程发起人和参与人提供监控我发起的任务与我参与的任务等功能。



图 7-5

7.7. 流程设计工具

LiveBOS Manager 提供图形化的流程设计工具，让开发者用最少的时间设计或修改企业业务流程。

- 图形化拖拽方式设计 workflows 模型
- 方便快捷的指定流程节点的对象要素。
- 直观定义流程参与者
- 直观指定流程转向逻辑
- 快速定位流程元素
- 提供流程验证，支持模型的合法性检查

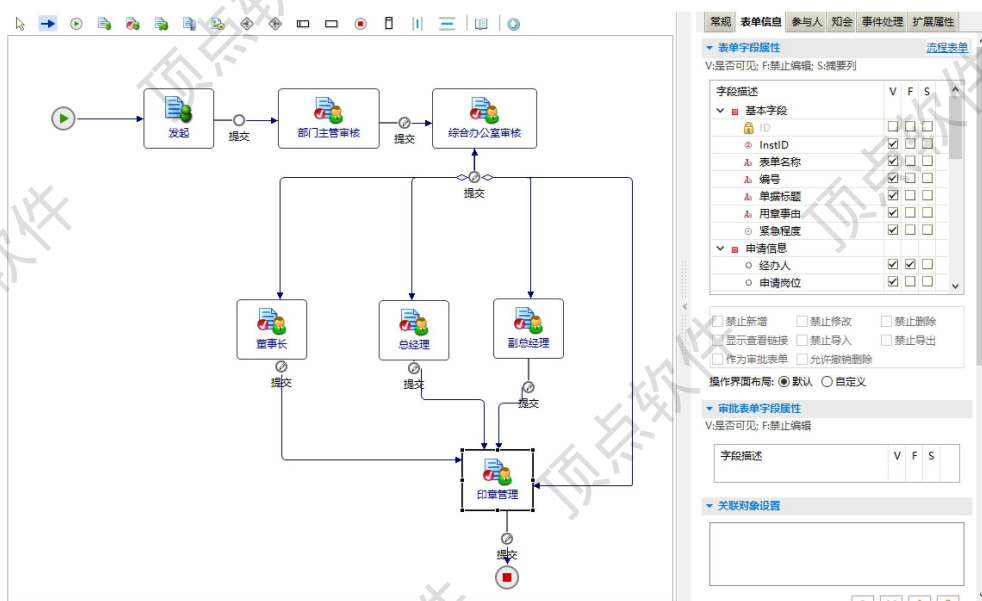


图 7-6

8. 门户引擎与界面个性化

LiveBOS 内置的门户引擎，提供了个性化界面的能力，让用户可以按照自己使用的要求，灵活部署自己的工作桌面；门户还支持开发人员创建虚拟的门户页面，直接挂接在菜单中供所有用户使用。大大改善了 LiveBOS 应用软件的用户体验。

LiveBOS 的门户信息部署具有最大的灵活性，它可以将页面分为任意层次的栏目组合，栏目之间有上下平铺，左右平铺，叠加等组合模式。每个栏目都提供：参数配置，最大化，最小化，复原等操作。

8.1. 门户设置

每个用户都可以设置自己的工作页面，LiveBOS 提供了丰富的栏目控制功能。用户可以将每个页面分为任意数量的分区，而每个分区内可以再任意数量的内容与任意数量的二级分区，通过多层次的分区嵌套，可以实现十分复杂的页面布局。

同一分区内的内容与二级分区，有三种组合模式：上下排列模式，左右排列模式，叠加页标签模式。用户可以在分区属性设置中选择所需要的模式。

[概念] 分区：指页面上的设定区域，它用来容纳“内容”或下级的分区。在未设置前，LiveBOS 的门户就是一个独立的、没有内容、也没有下级分区的根分区。用户的设置，实际就是在根分区中加入“内容”或“分区”。

[概念] 内容：在 LiveBOS 的门户中内容指的是可以在区域上展现的信息，它可能是一个静态的 HTML，一个对象，一个链接列表，一个搜索栏目等等。内容是 LiveBOS 的页，不能再加入其他的内容或分区了。

分区的属性：用户可以设置分区的标题名称，分区的布局模式，分区的显示风格。

8.2. 门户内容

每个用户都可以设置自己的工作页面，LiveBOS 提供了丰富的栏目控制功能。用户可以将每个页面分为任意数。

信息内容

包括 6 种 Portlet：对象记录-列表模式，对象记录-主题模式，图表分析，投票，图片展示，嵌入页面。

如，对象记录-列表模式 Portlet，用来显示某一选定对象或对象视图的内容，它为用户提供了在门户页面上查看其关注信息的手段。



图 8-1

在门户页面中，一般对象信息的显示都只显示最新的对象信息的主题，具体完整的信息查询与操作，可以通过“最大化”按钮进入完全的对象操作界面。用户可以通过“设置”按钮选择“对象”，“对象的显示模式”，“记录数量”，显示的字段等。

快捷事务

包括 5 种 Portlet：单类型信息查询，快速信息登记（新建），多类型主题搜索，快速链接，分组菜单。

如，单类型信息查询 Portlet：查询内容在门户界面中直接输入参数，获得查询结果的功能，对于常用的需要查询的信息，用户可以直接通过“查询内容”而非菜单模式获得信息。



图8-2

协同事务

包括 7 种 Portlet：流程处理中心，我发起的流程，我参与的流程，单类型待办流程，知会信息，日程，我的邮件。

如，流程处理中心，展现 workflow 系统中流程处理中心的待办流程，可以快速查询和处理当前需处理的待办事宜，快速进入流程处理中心。



图 8-3

外部公共资源

包括 5 种 Portlet：日历，静态页面显示，时钟，简易计算器，科学计算器。

如，静态页面显示，支持包含显示站内的静态网页内容，包括htm、html 页面。

除此之外，LiveBOS 还支持自定义Portlet 组件，可以由用户在 Studio 开发工具中预先定义Portlet 对象，门户中直接添加自定义的 Portlet 组件，便于用户快速地添加要展现的对象信息。

8.3. 门户模板与虚拟门户管理

LiveBOS 门户可以将某个用户的门户设置方案，保存为一个“门户模板”，这样其他的用户就可以直接加载该“门户模板”，而无需从头开始设置。管理员可以对“门户模板”进行管理维护。

LiveBOS 同时可以将一个门户页面保存为一个普通的页面，这样的页面可以挂接到菜单中，这样的页面展现与门户页面是一样的，但所有的设置功能将不存在，同时也不能进行最大化，最小化的操作。

9. Manager 运维管理工具

LiveBOS Manager (LiveBOS 管理控制台，以下简称 Manager)，是作为 LiveBOS 系统的独立模块，用于系统相关属性的设置，为系统的稳定运行提供维护和定制功能。在 Manager 中可以做到对系统运行状况的监控、对象和流程维护、项目管理和部署，并提供很多人性化的细节定制功能；Manager 将涉及系统定制的功能集成，系统管理员可以方便管理和维护 LiveBOS 系统。

10. 系统高级开发与扩展性

10.1. 用户存储过程支持

为了满足应用开发人员在具体应用开发过程中，进行复杂的业务逻辑处理，LiveBOS 系统还提供了调用存储过程和 SQL 语句的接口。LiveBOS 在不同层面提供了不同的调用 SQL 语句的方法。

在定制方法中，用户可以直接调用 SQL 语句或存储过程作为方法处理的逻辑操作。也可以在启动条件或例外判断时调用 SQL 进行验证调用方法的合法性。

在工作流程的每个步骤或动作的预处理或后处理及流转条件判断中也支持直接的 SQL 调用。在一些复杂的报表查询中，系统提供结果集事务与原生报表等类型对象，结果集事务与原生报表都支持直接调用 SQL 语句或存储过程。其中原生报表支持调用多个 SQL 语句，从而报表可以支持复杂的交叉表及多级嵌入式报表。

对于一些无法用普通实体表达的对象，系统提供了虚拟对象，虚拟对象通过查询或结果集事务进行字段映射。将查询或结果集事务模拟成一个普通实体对象，可以在其上定义各种对象方法和属性。

在各种表达式脚本中，系统还提供 SQL 型变量和取 SQL 值的函数供用户调用。

10.2. 开发专用的 WebService

LiveBOS 提供了一个统一的系统服务接口，在对象方法和工作流程中都可以方便的调用系统服务。系统已经实现了 Mail 服务、桌面提醒服务、短信服务及 MSN 通知等各种服务，也允许用户自行实现一些特定的服务，用户只需实现系统服务接口并在服务配置 XML 加入这些信息，系统即可调用。

10.3. JavaBean 支持

LiveBOS 系统整个体系有很好的可扩展性。提供了丰富的 Java 接口。如界面模式接口，操作验证接口，操作接口。用户可以自行实现这些 Java 接口并加入到对象模板配置中，系统即可按这些接口进行界面或操作验证处理。

在表达式脚本中，LiveBOS 也提供了直接加载JavaBean 的函数。JavaBean 加载后，用户即可在脚本中设置 javabean 属性及调用 javabean 方法。如：

用户实现了一个如下的 Employee 的 javabean。

```
package com.apex.test;

public class Employee {

    public void setWorkDays(int days);

    public double payoff();

}
```

用户可以在脚本中直接使用 该 javabean。如：

```
.....
emp = ABS_LOADBEAN("com.apex.test.Employee");
emp.setWorkDays($P{P1});
pay = emp.payoff();
.....
```

10.4. JSP 支持

为了满足用户特定的一些界面方式的需要，系统提供了一个“JSP 资源”类型的系统对象方便挂接应用开发人员定制的JSP 或者是 HTML 文件，当加入 JSP 资源，LiveBOS 可以统一对其进行权限控制。可以将这些JSP 挂入系统菜单，也可与其他对象进行整合。

10.5. 外部数据接口

LiveBOS 为了与外部其他系统进行整合交互。可以通过定制系统服务或 JavaBean 调用外部系统的 WebService 接口。应用系统各个功能应用能以WebService和Restful方式对外提供，同时能提供其它形式的调用，后台数据库支持现有的主流数据库系统。当第三方系统需要调用到顶点LiveBOS 平台业务系统接口时，只需将对象、流程基本属性中的“允许服务调用”选项打勾，即可以获取 LiveBOS Webservice、Restful 的接口描述文件，实现接口的调用。

11. 应用开发方式

11.1. 敏捷开发模式

敏捷开发是一种应对快速变化需求的软件开发能力。相对于“非敏捷”更强调沟通,变化,产品效益.也更注重作为软件开发中人的作用. 它是以人为核心、迭代、循序渐进的开发方法。在敏捷开发中，软件项目的构建被切分成多个子项目，各个子项目的成果都经过测试，具备集成和可运行的特征。简言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

LiveBOS 应用软件最适合敏捷开发模式。LiveBOS 具备敏捷开发的几个重要基础：1. 提供了统一的技术架构平台，可以确保所有的开发项目自然实现完全的集成,所有的信息流程,可以实现在同

一组织架构下的协同. 2. 快速的建模方式实现,可以快速验证需求,并实现持续改进。

11.2. 组织建模

在 LiveBOS 平台上, 开发面向一个企业, 或政府组织的管理信息系统, 首先应该建立相应的组织模型, 分析系统参与者的岗位角色。

组织建模时, 如果系统涉及组织外部用户 (如客户, 经销商), 应该建立外部用户体系。

在组织体系中, 除了应该加入一般的层次管理体系, 对于特定的团队, 如项目团队等也应加入整个组织模型中。

组织模型的建立后, 对象建模, 流程建模中涉及管理体系的设计。

11.3. 业务元素对象建模

针对应用的需求定义, 在 LiveBOS 上进行开发, 应该采用面向对象的分析方法。业务管理要素分析就是要在应用需求的基础上, 分析其信息关键主题元素, 如客户, 员工, 合同等要素, 同时结合 LiveBOS 对象的模型, 深入分析这些对象要包括的信息要素, 可能的操作, 可能的管理分类, 不同角色的信息视图等。

在对象分析基础上, 通过 LiveBOS 对象建模平台创建对象。并及时验证对象模型的正确性。业务元素对象建模的关键 (也是难点) 是如何合理地使用对象间的关系模式, 包括父子对象, 从对象, 对象关联等。它除了要考虑对象间的合理约束, 也要结合最终用户对信息的使用习惯。

11.4. 业务流程建模

业务流程主要指需要通过多人协同处理, 或人工处理与系统自动处理相配合的事务。对于简单一次性的处理, 可以直接定义为对象操作。

业务流程设计需要在其相关的对象完成之后进行。业务流程有两种形式:

1. 单表单 (对象) 流程
2. 多表单 (多对象) 流程

单表流程是基于单一的表单对象基础上实现的, 一般适用于简单的审批流程的业务处理。单表流程将所有流程中涉及的信息集中到一个表单上。不同的处理步骤, 参与者输入的是同一表单中的不同信息要素。

如果流程涉及的信息要素多, 并且可能涉及多个主题的信息, 则适合选择多表单的流程。但应该选择一个表单作为主表单。流程的主控制信息是放在多表单上的。

11.5. 菜单角色界面设计

完成系统的对象设计, 流程设计后, 接下来主要的工作是功能信息的用户界面组织。界面组织主要通过菜单与门户设计实现。

LiveBOS 通过用户界面与菜单方案来控制整个应用的布局。包括界面菜单的布局和界面主题的选择。其中菜单的定义包括层次菜单的分类选择, 菜单级次选择, 一般不超过 3 级, 如果选择标签布局的菜单, 则应该选择 2 级菜单。

LiveBOS 门户提供更加便捷易用的导航界面, 它提供在同一界面上按任意的方式部署常用信息, 常见操作或常用功能链接等功能。

原则上用户的门户是用户使用时自行设计, 但为了简化用户的设置, 一般设计人员需要根据用

户的不同角色，设计几套对应的模板，供用户或系统管理员加载使用。

11.6. 模型导入、导出与部署

完成全部的系统设计后，如果需要在生产系统上部署，或部署第二套系统，需要将设计模型导出到模型定义文件中，然后在目标系统上导入设计模型文件。

11.7. 使用辅助开发工具

LiveBOS 提供了辅助的工具，包括：

数据库系统快速转换工具：它可以将已存在的数据库，尽量按照 LiveBOS 的对象模型，转为相应的对象。设计者可以在转化后的对象原型上进行深化设计。

文档生成工具：它可以根据对象模型定义反向生成可以阅读的系统设计文档。