迎接Web 3.0

格物致知网

构想与实现

李博杰 高翔 贺羽

2011年1月

**内部资料 请勿传播**

目录

[第一章 Web 3.0设计理念 4](#_Toc283893777)

[1.1 以史为鉴，面向未来 4](#_Toc283893778)

[1.2 畅想Web 3.0 6](#_Toc283893779)

[1.3 创新每个人 7](#_Toc283893780)

[1.4 利益分配 8](#_Toc283893781)

[1.5 过渡阶段的Web 2.5 8](#_Toc283893782)

[第二章 项目与团队 10](#_Toc283893783)

[2.1 外科手术队伍 10](#_Toc283893784)

[2.2 概念完整性 11](#_Toc283893785)

[2.3 文档化 13](#_Toc283893786)

[2.4 用户体验 15](#_Toc283893787)

[2.5 调试与评测 16](#_Toc283893788)

[第三章 系统结构 18](#_Toc283893789)

[3.1 需求和设计目标 18](#_Toc283893790)

[3.2 系统模型 19](#_Toc283893791)

[3.3 关键系统模块 19](#_Toc283893792)

[3.4 数据库结构 19](#_Toc283893793)

[3.5 版本控制系统 20](#_Toc283893794)

[3.6 安全访问系统 20](#_Toc283893795)

[3.7 调试器 21](#_Toc283893796)

[3.8 注册表 22](#_Toc283893797)

[第四章 应用程序接口 22](#_Toc283893798)

[第五章 用户机制 23](#_Toc283893799)

[5.1 设计原则 23](#_Toc283893800)

[5.2 注册与登录 24](#_Toc283893801)

[5.3 昵称管理 24](#_Toc283893802)

[5.4 好友，标签与权限控制标记 25](#_Toc283893803)

[5.5 权限控制 26](#_Toc283893804)

[5.6 积分规则 27](#_Toc283893805)

[5.7 用户组 27](#_Toc283893806)

[5.8 勋章 27](#_Toc283893807)

[5.9 道具 27](#_Toc283893808)

[5.10 处分条例 27](#_Toc283893809)

[第六章 网络服务结构 27](#_Toc283893810)

[6.1 总体设计 27](#_Toc283893811)

[6.2 界面设计 28](#_Toc283893812)

[6.3 多密码权限 29](#_Toc283893813)

[6.4 配置设置 29](#_Toc283893814)

[6.5 聊天大厦 29](#_Toc283893815)

[6.6 小字报 30](#_Toc283893816)

[6.7 今日热点 31](#_Toc283893817)

[6.8 漂流瓶 31](#_Toc283893818)

[6.9 自定义模块 31](#_Toc283893819)

[第七章 消息系统 32](#_Toc283893820)

[7.1 总体设计 32](#_Toc283893821)

[7.2 用户消息 33](#_Toc283893822)

[7.3 系统消息 33](#_Toc283893823)

[第八章 个人主页 34](#_Toc283893824)

[8.1 总体介绍 34](#_Toc283893825)

[8.2 界面结构 34](#_Toc283893826)

[8.3 访问地址 35](#_Toc283893827)

[第九章 百科 36](#_Toc283893828)

[9.1 总体设计 36](#_Toc283893829)

[9.2 分级控制体系 36](#_Toc283893830)

[9.3 词条与评论 36](#_Toc283893831)

[9.4 分类目录 38](#_Toc283893832)

[9.5 保护与审核 38](#_Toc283893833)

[9.6 编辑冲突 39](#_Toc283893834)

[9.7 同义词 39](#_Toc283893835)

[9.8 历史版本 39](#_Toc283893836)

[9.9 编辑词条 40](#_Toc283893837)

[9.10 我的百科 40](#_Toc283893838)

[9.11 后台管理 40](#_Toc283893839)

[9.12 百科活动 41](#_Toc283893840)

[第十章 组织主页 41](#_Toc283893841)

[10.1 总体设计 41](#_Toc283893842)

[10.2 分级管理体系 41](#_Toc283893843)

[10.3 可视化编辑 42](#_Toc283893844)

[10.4 访问权限 42](#_Toc283893845)

[第十一章 论坛模块 42](#_Toc283893846)

[11.1 总体设计 42](#_Toc283893847)

[11.2 讨论版与版内分类 43](#_Toc283893848)

[11.3 边栏 43](#_Toc283893849)

[11.4 帖子类型 43](#_Toc283893850)

[11.5 帖子内容页 44](#_Toc283893851)

[11.6 版主与帖子操作 45](#_Toc283893852)

[第十二章 C语言在线评测系统 45](#_Toc283893853)

[12.1 总体设计 45](#_Toc283893854)

[12.2 关卡模式 45](#_Toc283893855)

[12.3 题库 46](#_Toc283893856)

[12.4 题目 46](#_Toc283893857)

[12.5 评测 47](#_Toc283893858)

[12.6 其他题型 47](#_Toc283893859)

[12.7 模拟题 47](#_Toc283893860)

[12.8 比赛 48](#_Toc283893861)

[12.9 今后研发方向 48](#_Toc283893862)

[第十三章 站内搜索引擎 48](#_Toc283893863)

[13.1 总体设计 48](#_Toc283893864)

[13.2 中文分词 49](#_Toc283893865)

[13.3 排名算法 49](#_Toc283893866)

[13.4 实现 50](#_Toc283893867)

# 第一章 Web 3.0设计理念

## 1.1 以史为鉴，面向未来

自从人类文明出现以来，人们一直致力于解决时空问题。人们发明轮子、汽车、火车、飞机、宇宙飞船，都是在克服空间障碍，节约人类有限的时间。互联网正是解决这样问题的新型工具，它最大限度地克服了空间障碍，使得天各一方的人们在理论上可以光速实现自由交流。

1969年12月，美国的分组交换网ARPANET投入运行，从此计算机网络进入了一个新的发展阶段，标志着现代通信时代的开始。1978年，国际标准化组织ISO提出了著名的开放系统互联参考模型OSI，结束了各大网络体系“各自为政”的局面，开辟了计算机网络的新纪元。我国的公用分组交换网于1989年11月建成，可以认为是中国互联网的起跑线。

我国上网用户数在经历1998年的酝酿后，从1999年到2000年以每年两倍的速度狂飙，互联网泡沫沸腾，现在人们耳熟能详的几大网络公司多是那时起步的。2001年到2002年是中国互联网最痛苦的时代，大量网络公司因泡沫破灭而倒闭或转行。但随着新浪、搜狐、网易等公司宣布盈利，互联网熬过了第一次冬天，浴火重生的胜利者开始了第二次圈地运动。随着互联网春天的到来，更多的网站开通了，2005年7月时网站数量已达67.8万个，网民数量也突破1亿。

就在这时，高歌猛进的互联网界潜移默化地经历着一场革命。这场革命是如此平静，以至于连它们的发动者都没有意识到这将是怎样一个开天辟地的大事变。

2005年对互联网用户的调查，用户最反感的内容中，弹出广告以22.6%名列第二，与排名第一的病毒仅差3个百分点。也许当时的设计者们还没有意识到，弹出广告正是违背了“用户处于控制地位”的设计原则。

被今天的Web设计者称作Web 1.0的“网站处于控制地位”的信息发布网站，正是像传统媒体一样，不断改变花样让用户接受自己的信息。不过网站上允许发表评论了，至少些许实现了信息来源与信息接收端的互动，也算是一种进步吧。但从根本上，用户作为信息被动接受者的角色没有改变。

Google、百度等搜索引擎所做的工作可以相当于Web 1.5，它们使得用户可以选择查看某特定方面的信息，搜索一个关键词，机器就会把最相关的内容呈现出来，而不需要用户自己去浩如烟海的信息世界中搜寻。

为了方便大家共同讨论某个感兴趣的话题，百度出品了贴吧，猫扑、天涯等很多或综合或专业的论坛如雨后春笋般，冲击了几大网络巨头的垄断地位。Web建设者的个人英雄主义时代开始，2005年前后又有一批今天耳熟能详的网络公司从库房里、地下室里脱颖而出。以论坛为主导的新一场圈地运动对传统门户网站的理念是个重大的革新，它把用户从单纯的阅读者和评论者中解放出来，容许大家自由发表观点，每个人都是编辑、作者。用户参与网络活动的积极性提高了，用户在网站讨论中收获了自我满足的精神食粮，网站变得更有粘性。

希望在网上占有一席之地的技术爱好者纷纷架起自己的独立博客，新浪等大公司也紧抓商机，开通了免费blog服务。博客是完全自主的展现自我的平台，进一步深化了每个人都是作者和主人的观念。数以万计的个人站长跨越较低的技术和资金门槛，悄然掀开Web 2.0时代的序幕。

事实上，这一阶段国外以Wikipedia（维基百科）为代表的wiki（百科）站点的兴起也是Web 2.0的重要组成部分，宗旨是由用户来完成自由开放的百科全书。国内也有百度百科、互动百科等影响力较大的wiki社区，但明显专业度与凝聚力不足。

“墙外开花墙内香”，大洋彼岸的Facebook，Youtube，Twitter等网站的兴起奏响了Web 2.0时代的最强音。这些统称为SNS（Social Network System，社交网络）的新型网络传播方式大大拉近了虚拟世界与现实世界的距离，使人们足不出户便可以与好友分享新鲜事、发表心得感言，把现实世界的朋友关系延伸到网络上。新浪、搜狐、网易等网站也曾推出过同学录服务，如Chinaren社区等。目前国内这方面最成熟的代表是人人（原2005年成立的校内网）、QQ（不完全算是）。同时，各大社交网络也纷纷推出了桌面客户端、移动客户端，以满足日益增长的使用移动设备上网聊天的需要。

Web 2.0时代，由于每个人都是内容的作者，因此长尾效应愈加凸显，大部分的流量不是来自百度、新浪等少数几个大网站，而是更多分流到规模小，但专业性强的垂直分类网站上了。事实上，由这些对专业知识更了解的人来解决本行业内的问题恰恰做到了现实生活中难以实现的资源优化调配，使得网络上的信息更具专业性、针对性、合理性，减少了权威性、垄断性。

另一方面，由于现实世界的群聚效应，名人效应也是愈加凸显的。近两年网络上爆出的“门”往往在几天内走红网络，并波及到传统媒体，其传播速度和影响力都是史无前例的。新浪微博并不是第一个在中国引入微博服务的，但凭借新浪博客上众多名人的推广效应，新浪微博在短短半年内风靡全国，并成为各大网站的“标配”。

在某种意义上，网络也是社会舆论监督的重要形式。网络言论的开放性使一大批社会的阴暗面露出冰山一角直至浮出水面，网络舆论的强大压力也左右了很多社会焦点问题的动向。

但不容忽视的是，如今的Web 2.0在带来经济、社会、文化生活巨大变革的背后，隐藏着深层的矛盾与危机。

虚拟性和分布性是网络自出现以来的基本特征。很多人也许津津乐道于虚拟世界的巨大自由，但谎言、虚假与诚信缺失却在动摇着社交网络的根基。在把现实世界的货币引入网络世界时，不得不考虑未曾谋面的双方在利益分配上可能出现的争执和互不信任的困局。因此，网络世界需要真实化，但不是全部真实化。在建立在互相信任基础上的、涉及现实生活的网络层面上实现全面的真实化，而在不需要信任关系维系的公共论坛等处，可继续享受匿名带来的自由。

再谈分布性。分布性的本意是将复杂的任务分解到每个人、每台计算机去完成，充分发挥每个人的能力和每台计算机的运算资源，但在实际运行中，由于网络是自由开放的，不可避免地将分布性异化成了“各自为政”，大量网站如分封的郡县，不但没有合理分工合作，反而为了自身的利益相互开战，损害绝大多数用户的利益，近期的QQ与360之战就是一例。

正如西方经济的发展以罗斯福新政为标志，经历了从完全自由到国家统筹规划的痛苦的蜕变，网络在经历Web 1.0的垄断、Web 2.0的完全自由后，到了民主集中的时候。很多人，尤其是生活在官僚压迫下的人，对“集中”往往误解成“独裁”。事实上，在现实世界中，不论国家，不论政体，都要有一套规范的自上而下的管理决策机制和完善的道德法律体系。没有规矩，不成方圆，从“网络水军”到网络诈骗，网络世界的种种乱象使得真正有技术、有理念的网络公司难以立足，而采取侵犯版权、虚假欺骗等不道德甚至违法措施的人却大发横财。靠行业自我约束是无法洗净自身的，必须引入一套完备的民主集中制的网络政治体系和利益分配机制来规范人们在虚拟世界的行为。或许随着技术的进步，虚拟世界这个名词在可以预见的将来将不复存在。

另外，合理的利益分配也是迄今为止网络发展未能解决的难题。不论是新浪博客还是人人空间，用户撰写再多的内容，也只是纯凭兴趣，可以获得精神上的满足，但不能取得物质上的收益。唯一有效的收益方式便是放置自己的广告，但这显然只是权宜之计。更严重的是，一个实体组织若要创建自己的网络平台，必须自己从头摸索，或者花大价钱请网络公司帮忙构建，而之后的收益方式又需要组织自己出人力物力来摸着石头过河。这样多次无意义的重复摸索降低了生产力，浪费了宝贵的技术力量。

因此有必要为希望创建网络平台的实体组织提供一个开放的开发平台和一套完善、合理的利益分配体系。开发平台足够易用使得组织无需凭借外力即可制作出实际的网络应用；足够可用使得能够充分发挥每个人的创造力，根据自己的意愿实现设计效果。利益分配体系建立在真实信任关系的基础上，引入现实社会中的仲裁机制和民主法制，公平合理地化解虚拟世界的利益纠纷，保证了开发者和使用者的共同利益。

## 1.2 畅想Web 3.0

Web出现之前的主要矛盾是信息传递困难，Web 1.0解决了它，新的主要矛盾变为信息基本静态与用户参与的愿望之间的矛盾；Web 2.0解决了旧矛盾，又产生了信任体系和利益分配机制不完善和虚拟世界与现实世界快速融合的新矛盾。Web 3.0虽然有种种解释，将产生语义网、人工智能等划时代的技术突破，但解决Web 2.0的遗留矛盾始终是不可淡忘的主题。

在Web 1.0中，用户是网站的访问者，可悲的是很多现有的网站仍然抱有类似的陈旧观念，仅仅通过页面点击率衡量得失，通过广告取得收益。在Web 2.0中，用户是网站的参与者，他们发表信息，交流知识，成为现实世界社交关系的便捷渠道和重要补充。在Web 3.0中，用户是网站的建设者，大家充分发挥各自的优势，将创新成果应用到自己的“地盘”，共同开发整个网站的程序和内容，共同享受网站发展的精神愉悦和物质利益。

总而言之，Web 3.0在两个核心理念上将是对传统网络模型的突破：

通过实名制，结合真实的社会组织，将网络的虚拟性转化为真实性；

通过网络政治和利益分配机制，将网络的分布式转化为民主集中式。

下图是Web 3.0所构想的“网民的虚拟人生”：

## 1.3 创新每个人

多米诺骨牌效应最通常的形式，并不是逐级放大的神奇推演，而是点点滴滴的蔓延。尽管每一块的力量都同比例的微小，但是最终的伟大质变却是隐藏在不断积累的过程中。

每天15分钟是什么概念？如果是阅读，一分钟可达300字，15分钟就是4500字，一周可读3.15万字，一个月就是12.6万字。那么一年就是151.2万字。如果一本书以7.5万字算，每天读15分钟，一年可读20本书。有一个人这样坚持了半个世纪，共读了8235万字、1098本书。他就是加拿大籍的世界著名医生奥斯勒。

我们从网站维护谈起。在其他条件不变的前提下，花费在网站上的时间越多，取得的效果越好。

管理员一个人维护网站，每天3小时，每周21小时；

15个人的团队维护网站，每周3小时，共45小时；

科大7500本科生每周登陆一次网站，一次10分钟，每周共1250小时。

按照平均每15秒点击一次网页的估计，每天的网站点击量为42857次。

受到中央电视台等多家媒体报道的大型科普网站科学松鼠会，每天点击量也只是在10万次的量级。

而这一切只考虑了科大一所高校！全国共有四千多所高校，人人网就是这样炼成的。

也许有人会问，这些推算都是不切实际的，7500学生凭什么每个人、每个星期都来这个网站？

这就是用户黏性的威力。例如科大的网上选课系统，每个学生必须登录，这是行政压力所驱使的。而我们前进的方向正是与真实的社会组织、学生社团等结合，一个人所加入的组织机构发布的信息只有在我们这里能够看到，他不就必须经常来我们网站查询新信息了吗？

那么为什么组织机构要选择在我们这里发布信息呢？这就是1.1节所述易用性、可用性和利益分配机制的厉害之处。组织机构不仅不需要为创办网站付出高额成本，还可以充分发挥创造力，满足自我实现的需要，同时还能从利益分配机制中取得今后的收益，何乐而不为呢？

一个团队的创新是有限的，人类的创新相对来说是无限的。人是创新的主体，是最宝贵的资源。要发挥每个人、每个组织的主观能动性，提供一个开放的平台，提高归属感和认同感，让他们不仅成为网站的访问者、参与者，更是网站的建设者。

一个有活力的技术团队是网站成功的必要条件。没有充足的技术资源，上文提到的易用性、可用性就无从谈起，没有足够的优势吸引到组织和个人的关注、认可和参与，也就没有后面的网络政治、利益分配机制了。

一个忠实的用户群体是网站成功的充分条件。有了领先的理念，又有一批数量可观的用户，推行新方法、推销新功能、推广新理念就如探囊取物了。例如，百度推出一个新功能，绝对比自己的小网站推出一个同样的功能能吸引更多人的关注和试用，也就会取得更大的成功。再如，“青年民意领袖”韩寒在博客上发表一篇文章，即使只有一个字，也会有数十万人关注他的“杰作”；相反，自己呕心沥血写的一篇万言长文，就算再有道理，只要没有炒作，也很难达到韩寒杂文的影响力。

QQ和360之战，如果没有工信部的调停，最后肯定两败俱伤，但相对而言腾讯的胜算显然很大。为什么？腾讯QQ拥有4亿多活跃用户，拥有中国最大的用户群体，且在即时聊天领域基本处于垄断地位，诸如MSN等聊天软件应用范围很窄，这种垄断酷似谷歌退出后百度在中国搜索引擎领域的垄断。离开了QQ，很多用户根本无法与人网上聊天，这简直成了网民生活的必需品。而360安全软件则不尽然。首先360的用户份额比QQ要小，另外最重要的是360在安全软件领域不占有垄断地位，不能做到让用户别无选择。没有360，还可以选择瑞星、金山、卡巴斯基等其他杀毒软件和浏览器安全软件。

以上归结起来，就是用户对QQ的忠诚度，或依赖度，大于对360的相应指标。一旦开战，忠诚度高的一方更容易获得认同。在平时网站间的竞争中，也存在着类似的“大鱼吃小鱼”的现象。曾几何时，各大网站的校友录功能群雄并起，但现在还有几家能跟人人网一决雌雄？

因此，格物致知网要在技术和理念上不断创新，把握好现有的忠实用户，稳健发展可能的忠实用户，团结一切可以团结的力量，至少在一个二级领域实现压倒性优势，凭借不断改善的用户体验，不断提升用户的忠诚度和信任度，甚至成为一些人上网的“必需品”，则发挥每个人的创新能力，领跑Web 3.0的理念突破，Nothing is impossible。

## 1.4 利益分配

如上文所述，目前不管内容创作者还是实体组织，除了广告以外，都没有合适的方式从创作的内容中获取利益。广告之所以成为Web 2.0时代最广泛使用的获利机制，是因为广告每次点击可以自动结算，而且并不是由点击广告的人去支付，而是羊毛出在羊身上，由发布广告的商家支付广告费，这样点广告的人就不会每点一次广告都心疼一次钱包。但如果阅读一篇文章或者使用一个网站功能就要付费，就算每次付费的金额非常小，用户也很难愿意付费。

虚拟货币可以解决这一问题。相比传统货币，虚拟货币每次支付更简单、成本更低，不需要打开网上银行做麻烦的付款操作，所有支付动作都是由系统在后台自动结算的。同时，由于内容的阅读操作、网站功能的使用操作都是在同一系统中进行，系统可以追踪用户使用这些内容获取了多少后续收益，从而利益的自动分配也成为可能。整个系统只需要定期向用户收取累积的使用费或者发放累积的收益，就像网络广告公司和发布广告的商家之间定期结算一样。

此外，虚拟货币还可以解决按照组织结构来做利益分配的问题。例如在一家公司工作，取得的收益需要在公司的老板、经理和员工等不同层级间分配。在Web 2.0中，公司如何分配利益对普通员工往往是不透明的，由于利益分配带来的纠纷也非常常见。但如果使用一个中心化的系统，用虚拟货币来做不同层级间的利益分配，只要按照下文所述的层次结构模型设置好不同层级间的分成比例，利益分配机制就是透明的、自动化的，不用担心利益纠纷的问题。

## 1.5 过渡阶段的Web 2.5

从Web 1.0进入Web 2.0经历了以搜索引擎为主导的Web 1.5阶段，且真正抵达Web 2.0的核心——社交网络，也经历了不少新事物的诞生、发展与融合。同样，从Web 2.0也没有直达Web 3.0的高速列车，“跑步进入Web 3.0”只能成为“Web大跃进”的历史笑柄。

为此，我们提出了Web 2.5作为承上启下的中坚力量，它是在可预见的将来能够有步骤、有计划地付诸实施的发展规划。以下是表格形式的“过渡阶段总路线”，以每个学期为一个阶段，以2010-2011学年第一学期为起点，供团队策划实施具体发展战略时参考。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学期 | 1 | 2 | 3 | 4 | 5 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 第二章 项目与团队

## 2.1 外科手术队伍

注：本章前三节参考了*The Mythical Man-Month*（《人月神话》）。

软件项目犹如盖大楼，如果10个人的施工队盖楼需要2年，显然20人不会将工期缩短到1年，7300人在一天内盖完楼更是神话。产生如此“非反比关系”的根本原因在于这些项目需要一定的顺序步骤才能完成，即使有再多的人，也不可能一拥而上。**不论有多少个母亲，都需要十个月才能生出孩子**。

另外，添加人手意味着**更细的分工、更多的交流**，大量宝贵的工作时间被浪费在开发者间的沟通与协商上了。一般来说，n个人“一拥而上”的团队两两需要交流，花费n(n-1)/2的交流时间，是大于人数的线性增长的。对于普通的软件项目，添加人手在开始时由于人数的增加确实可以提前工期，但人数增加到一定程度后由于互相交流等内耗，工期不会继续缩短，甚至还可能因内耗过大反而延长工期。另外，每个程序员对程序的架构与实现都有自己的想法，如果不能得到统一，很可能导致内部的不和谐，从而影响开发的积极性。

更严重的是，如果一个软件项目事先没有充分估计复杂度，而没有完成阶段性目标，从而中途添加一些人手，这样往往会导致**迟到的项目越来越迟**，因为新来的人需要原有的开发者腾出一部分时间对他们进行培训，使之熟悉整个软件的开发流程、设计文档和当前任务，在这段时间中原有的人被迫停下手中的工作，从而耽误了更多的时间。

优秀程序员与较差程序员之间的差异是广泛存在的，最优秀的程序员与平庸程序员的产出率之比可达5:1。因此，并**不是所有会编程序的人都要参与代码的真正编写工作**：一个优秀的程序员应该把全部精力投入到最擅长和最有效率的程序设计、代码编写工作中，而把测试、工具、文档、管理等其余琐事安排给其他人来做；普通的程序员则可以从事与程序开发相关，但并不要求最高水平的辅助性事务。

基于以上事实，我们必须避免“**一拥而上**”的开发方式，一个大型团队应由若干精干的小型队伍组成，**每个队伍负责一个相对独立的模块**。我们把这样的小团队称为“**外科手术队伍**”：

**首席程序员**（外科医生）：**亲自**定义功能和性能技术说明书，设计程序，编制源代码，测试以及书写技术文档。首席程序员需要极高的天分、充足的经验和应用数学、业务数据处理或其他方面的大量系统和应用知识。

**参谋**（副手）：设计的思考者、讨论者和评估人员。首席程序员与参谋沟通设计方案，但不受到参谋建议的限制。参谋在与其他团队的功能和接口讨论中代表自己的小组。他需要详细了解所有的代码，研究设计策略的备选方案，充当首席程序员的保险机制。

**管理员**：首席程序员是老板，必须在重大事件上具有决定权，但决不能在这些事务上浪费任何时间。因而，他需要一个控制财务、人员、工作地点安排和机器的专业管理人员，该管理员充当与组织中其他管理机构的接口。

**测试员**：首席程序员需要大量合适的测试用例，用来对他所编写的代码进行测试。因此，测试人员既是为他的各个功能设计系统测试用例的“挑刺者”，同时也是为他的日常调试设计测试数据的助手。他还负责计划测试的步骤和为测试搭建测试平台。

**辅助人员**：负责文档编辑、工具维护等。根据首席程序员的草稿，对设计文档进行分析和重新组织，形成可读性强的正式文件；向首席程序员提供各种参考信息和资料；保证所有基本服务的可靠性，构建、维护和升级团队必需的各种开发环境、开发工具与实用程序。

在实际项目中，一个管理员可以管理和协调3~5个外科手术队伍，除去管理员外每个外科手术队伍由首席程序员、参谋、测试员、辅助人员组成，由首席程序员牵头与拍板，避免了策略与接口的不一致性，维护了概念完整性，减少了多人反复交流争论的内耗。

此外，每个大型团队中总有一两个乐于掌握复杂编程语言的**语言专家**。这些天才不同于首席程序员，首席程序员主要是系统设计者以及考虑系统的整体表现。而语言专家则寻找一种简洁、有效的使用语言的方法来解决复杂、晦涩或者棘手的问题。他们可以为各开发小组提供程序语言支持。

## 2.2 概念完整性

软件的目的是使计算机更加容易使用。为了做到这一点，计算机装备了语言和各种工具，这些工具实际上也是被调用的程序，受到编程语言的控制。使用这些工具是有代价的：软件外部描述的规模大小是计算机系统本身说明的十倍。用户会发现寻找一个特定功能是很容易的，但相应却有太多的选择，要记住太多的选项和格式。只有**当这些功能说明节约下来的时间，比用在学习、记忆和搜索手册上的时间要少时，易用性才会得到提高**。

对于给定级别的功能，能用最简洁和直接的方式来指明事情的系统是最好的。仅有简洁是不够的。有些程序语言或软件要表达一件待完成的事情，常常需要对基本元素进行意料不到的复杂组合，而且除了了解基本要素和组合规则，还需要学习晦涩的用法，以及在实际工作中如何进行组合。这违反了直白的原则。

**简洁**和**直白**来自概念的完整性。每个部分必须反映相同的原理、原则和一致的折衷机制。在语法上，每个部分应使用相同的技巧；在语义上，应具有同样的相似性。因此，易用性实际上需要设计的一致性和概念上的完整性。

概念的完整性要求设计必须由一个人，或者非常少数互有默契的人员来实现，而大型项目的进度压力却要求很多人员来开发系统。获得概念完整性有两个重要途径，一是体系结构与实现分离，二是外科手术队伍的开明专制。

对于非常大型的项目，将设计方法、体系结构方面的工作与具体实现相分离是获得概念完整性的强有力方法。

系统的体系结构指的是完整和详细的用户接口说明，是用户要完成自己全部工作所需参考的手册的集合。因此，系统的结构师，如同建筑的结构师一样，是用户的代理人，他们需要运用专业技术知识来维护**用户的真正利益**。

体系结构同实现必须仔细地区分开来。**体系结构陈述的是发生了什么，而实现描述的是如何实现**。一个简单的例子是时钟，它的结构包括表面、指针和上发条的旋钮。当一个小孩知道了时钟的外表结构，他很容易从手表或者教堂上的时钟辨认时间。而时钟的实现，描述了表壳中的事物——很多种动力提供装置中的一种，以及众多控制精度方案的一种。

当讨论专制与民主这个带有高度感情色彩的话题时，我们当然不认为首席程序员才有最好的设计与创意，最好的创意往往来自实现者与用户。但**概念的完整性决定了系统的易用程度**，一个设计良好的系统不应该是各种“优秀理念”的大杂烩，而应以一条主线串联所有模块与功能。不能与系统基本概念进行整合的良好想法和特色，最好放到一边，不予考虑。如果出现了很多非常重要但不兼容的构想，就应该抛弃原来的设计，对不同基本概念进行合并，在合并后的系统上重新开始。

“**没有规矩，不成方圆**”，外部的体系结构规定实际上是增强，而不是限制实现小组的创造性。一旦他们将注意力集中在没有人解决过的问题上，创意就开始奔涌而出。在毫无限制的实现小组中，在进行结构上的决策时，会出现大量的想法和争议，对具体实现的关注反而会比较少。

当建议由体系结构的团队来编写计算机和编程系统的所有外部技术说明时，编程人员可能提出三个反对意见：

1. 该说明中的功能过于繁多，而对实际情况中的成本考虑比较少；

2. 结构师获得了所有创造发明的快乐，剥夺了实现人员的创造力；

3. 当体系结构的队伍缓慢工作时，很多实现人员只能空闲地坐着等待。

第一个问题确实是一项危险。在开发第一个系统时，结构师倾向于精炼和简洁。他知道自己对正在进行的任务不够了解，所以他会谨慎仔细地工作。在设计第一个项目时，他会面对不断产生的装饰和润色功能。这些功能都被搁置在一边，作为“下一个”项目的内容。第一个项目迟早会结束，而此时的结构师，对这类系统充满了十足的信心，熟练掌握了相应的知识，并且时刻准备开发第二个系统。

第二个系统是设计师们所设计的最危险的系统，一种普遍倾向是过分地设计，向系统添加很多修饰功能和想法，形成一个庞大而不实用的恐龙。而当他着手第三个或第四个系统时，先前的经验会相互验证，得到此类系统通用特性的判断，而且系统之间的差异会帮助他识别出经验中不够通用的部分。

为避免第二系统效应，设计师可以有意识关注那些系统的特殊危险，运用特别的**自我约束准则**，来避免那些功能上的修饰；根据系统基本理念及目的变更，舍弃一些功能。

第二个问题纯粹是一项误解。整个创造性活动包括了三个独立的阶段：**体系结构**（architecture）、**设计实现**（implementation）、**物理实现**（realization）。在实际情况中，它们往往可以同时开始和并发地进行。

实现同样是一项高级别的创造性活动。在外部说明完成之前，设计实现人员有很多的事情可以做。只要有一些最终将并入外部说明的系统功能雏形，他就可以开始了。首先，必须设定良好定义的时间和空间目标，了解产品运行的平台配置。接着，他可以开始设计模块的边界、表结构、算法以及所有的工具。另外，还需要花费一些时间和体系结构师沟通。

具体实现中创造和发明的机会，并不会因为指定了外部技术说明而大为减少，相反**创造性活动会因为规范化而得到增强**，整个系统也一样。

关于第三个时间顺序和阶段性上的问题，回答很简单，在说明完成的时候，才开始使用编程实现人员，他们之前可以为其他项目服务。这也正是在搭建一座建筑时所采用的方法。

概念的完整性要求系统只反映**唯一的设计理念**，用户所见的技术说明来自**少数人的思想**。实际工作被划分成体系结构、设计实现和物理实现，但这并不意味着该开发模式下的系统需要更长的时间来创建。经验显示恰恰相反，整个系统将会开发得更快，所需要的测试时间将更少。同工作的水平分割相比，**垂直划分**从根本上大大减少了劳动量，结果是使交流彻底地简化，概念完整性得到大幅提高。

## 2.3 文档化

据《创世纪》记载，巴比伦塔是人类继诺亚方舟之后的第二大工程壮举，但巴比伦塔同时也是第一个彻底失败的工程。

这个故事在很多方面和不同层次都是非常深刻和富有教育意义的。让我们将它仅仅作为纯粹的工程项目，来看看有什么值得学习的教训。这个项目到底有多好的先决条件？他们是否有：

1. 清晰的目标？是的，尽管幼稚得近乎不可能。而且，项目早在遇到这个基本的限制之前，就已经失败了。

2. 人力？非常充足。

3. 材料？在美索不达米亚有着丰富的泥土和柏油沥青。

4. 足够的时间？没有任何时间限制的迹象。

5. 足够的技术？是的，金字塔、锥形的结构本身就是稳定的，可以很好分散压力负载。对砖石建筑技术，人们有过深刻的研究。同样，项目远在达到技术限制之间，就已经失败了。

那么，既然他们具备了所有的这些条件，为什么项目还会失败呢？他们还缺乏些什么？两个方面——**交流**，以及交流的结果——**组织**。他们无法相互交谈，从而无法合作。当合作无法进行时，工作陷入了停顿。通过史书的字里行间，我们推测交流的缺乏导致了争辩、沮丧和群体猜忌。很快，部落开始分裂——大家选择了孤立，而不是互相争吵。

在当今的大型编程项目中，其实也是这样的情况。因为左手不知道右手在做什么，所以进度灾难、功能的不合理和系统缺陷纷纷出现。随着工作的进行，许多小组慢慢地修改自己程序的功能、规模和速度，他们明确或者隐含地更改了一些有效输入和输出结果用法上的约定。

那么，团队如何进行相互之间的交流沟通呢？通过所有可能的途径。

1. **非正式途径**：清晰定义小组内部的相互关系和充分利用电话、短信、聊天工具等，从而达到对所书写文档的共同理解。

2. **会议**：常规项目会议。会议中，团队一个接一个地进行简要的技术陈述。这种方式非常有用，能澄清成百上千的细小误解。

3. **工作手册**。在项目的开始阶段，应该准备正式的项目工作手册。

**是什么**

项目工作手册不是独立的一篇文档，它是对项目必须产出的一系列文档进行组织的一种**结构**。

项目所有的文档都必须是该结构的一部分。这包括目的、外部规格说明、接口说明、技术标准、内部说明和管理备忘录。

**为什么**

技术文档几乎是必不可少的。如果某个开发者就系统的某部分去查看一系列相关的技术文档，他发现的不仅仅是**思路**，而且还有能追溯到最早的备忘录对产品的**建议与解释设计**。

基于上述理由，再加上“未来产品”的技术文档将诞生于“今天产品” 的备忘录，所以正确的文档结构非常重要。事先将项目工作手册设计好，能保证文档的结构本身是规范的，而不是杂乱无章的。另外，有了文档结构，后来书写的文字就可以放置在合适的章节中。

使用项目手册的第二个原因是控制信息发布。控制信息发布并不是为了限制信息，而是确保信息能到达所有需要它的人的手中。

**怎样做**

我们认为，每一个编程人员都**可以读到**所有的设计文档，但**没有必要了解**所有的设计文档。在大型项目中，让每一位编程人员了解整个系统的结构不仅是不现实的，也是极为浪费时间的。编程人员只需要了解系统的整体架构与涉及自己团队的模块的设计目标、设计原则、与其他模块的接口规范等。但另一方面，开放全部设计文档给编程人员可以增强团队的互信度与忠诚度，通过查询其他模块的具体实现减少接口设计方面不必要的疑惑与不一致，同时还可以从其他团队的实现方法中获得启发。我们的整个团队应该是互利的战友，而不是各行其道的竞争者。

文档化的根本目的是获得**概念完整性**与**减少交流**。减少交流的方法是**人力划分**（division of labor）和**限定职责范围**（specialization of function）。当使用人力划分和职责限定时，树状管理结构所映出对详细交流的需要会相应减少。

事实上，树状组织架构是作为权力和责任的结构出现。其基本原理——管理角色的非重复性——导致了管理结构是树状的。但是交流的结构并未限制得如此严格，树状结构几乎不能用来描述交流沟通，因为**交流是通过网状结构进行的**。在很多工程活动领域，树状模拟结构不能很精确地用于描述一般团队、特别工作组、委员会，甚至是矩阵结构组织。

在树状编程队伍的每棵子树中，产品负责人和技术主管之间的关系值得重视。

产品负责人的角色是什么？他组建团队，划分工作及制订进度表。他要求，并一直要求必要的资源。这意味着他主要的工作是与团队外部，向上和水平地沟通。他建立团队内部的沟通和报告方式。最后，他确保进度目标的实现，根据环境的变化调整资源和团队的构架。

那么技术主管的角色是什么？他对设计进行构思，识别系统的子部分，指明从外部看上去的样子，勾画它的内部结构。他提供整个设计的一致性和概念完整性；他控制系统的复杂程度。当某个技术问题出现时，他提供问题的解决方案，或者根据需要调整系统设计。他的沟通交流在团队内部是首要的，工作几乎完全是技术性的。

在很小型的队伍中，产品负责人和技术主管可以是同一个人。在大型的项目中则不容易得到应用。原因有两个：第一，同时具有管理技能和技术技能的人很难找到。思考者很少，实干家更少，思考者－实干家太少了。

第二，大型项目中，每个角色都必须全职工作，甚至还要加班。对负责人来说，很难在承担全部管理责任的同时，还能抽出时间进行技术工作。对技术主管来说，很难在保证设计的概念完整性，没有任何妥协的前提下，担任管理工作。

在真正的大型项目中，**技术主管作为总指挥，产品负责人充当其左右手**是一个不错的选择。而对于并不擅长管理的技术天才，可以让**产品负责人作为总指挥，同时树立技术主管在技术决策上的权威**。这要求产品负责人与技术主管在基本的设计理念上趋于一致，且产品负责人必须对技术主管的才能加以尊重，可以通过一些微妙的暗示体现技术主管的威信。

## 2.4 用户体验

不论是百度、新浪，还是最普通的个人博客，都需要考虑用户访问网站后的感受，如页面上的信息是否吸引人，操作是否明显而简单，希望的浏览与互动愿望是否能被实现，这就是用户体验。

一个相同的页面对不同的人群而言意味着或许大相径庭的用户体验。一个粉色充满爱意的页面也许对花季少女充满了诱惑，而正在游戏世界中寻找冲杀快感的男生也许会很快选择关闭页面。一个充斥着分类信息的门户型主页对时事爱好者是一种很好的导航，而正忙着在网上查询学术资料的研究者只会感到杂乱无章的混乱。不同的人群，不同的目的，不同的性格，都严重影响着一个网页在访问者心中的PageRank。

很多人把浏览器的默认首页设置为网址导航或搜索引擎，就是为了避免某个具体网站的限制，根据不同时间的不同上网需求选择合适的网站。除了已经规模化的百度、腾讯等大型网站，试图为所有用户提供服务是不现实的，也是不明智的，最终很有可能功能过于大众化而失去特色。因此，一个全新的理念就是网站提供不同用户群的不同上网目的所需要的功能，**由用户来选择需要的功能**，方便地自定义自己在网站上的“个人主页”，让网站成为量身打造的一站式网络服务平台。

具体来说，对一个特定的用户，除了一些网站主打的默认功能，还会以适当的方式展示与此用户所属群体相关的功能，每次访问网站时以相同的方式显示。对于高级用户，允许最大限度自定义功能模块，打造属于自己的网络空间。从这个角度来说，一个成功的网站必须是拥有完全自主知识产权的，设计师可以从容地**根据用户需求定制程序功能**，而不受制于现有程序或模板；相反，一些小网站为了省事直接使用了网上现成的程序，结果是根据程序功能“定制”了用户需求，这样构架出来的网站千篇一律，一看就知道是用谁家的程序架出来的，设计人员的创造力没有得到充分发挥。

但是，普通功能与高级功能总是网站开发中不可忽略的一对矛盾。一般来说，**80%的用户是很“懒惰”的**，他们只愿意看网站上现成的信息，能少点击一次就少动一次键盘鼠标，更没有心思去钻研所谓的“新功能”；20%的用户则是“高级用户”，他们乐意发掘网站的内容与功能，以独具匠心的方式使用网站，并参与网站上诸如写博客、论坛发帖的互动。80%的普通用户产生了50%以上的点击量和知名度，而**20%的高级用户产生了80%的内容**。

如果网站只注重“**易用性**”而不具备“**可用性**”，即操作很简单但没有什么高级功能，网站的访问者只会是那些匆匆过客，少有人愿意在网站上长期停留，也缺少用户有价值的分享与贡献；如果网站具备“可用性”而没有“易用性”，即功能繁多但主次不清、操作复杂，就会丧失绝大多数不耐心的普通访客。

事实上，易用性与可用性，分别代表普通用户与高级用户的需求，是对立统一的。一个设计良好的网站会兼顾两类人群的需求，将**最常用的操作**以简单的方式展现在页面的明显位置，方便普通用户的使用；网站的各类高级功能以恰当的组织方式放置在相对不明显的位置，但附有详尽的帮助系统，使得高级用户可以“随心所欲”地操纵网站。

以论坛的回帖系统为例，主帖下方会显示只有加粗、高亮等简单功能的快速回复框，但附有“高级回复”的链接，普通用户完全可以简单写上几个字，快速提交回复；但希望插入图片或附件、写大段排版整洁的文字的，完全可以使用功能强大、完备的高级编辑器。即使在高级编辑器内部，也可以对高级用户和“超高级用户”加以区别，例如插入数学公式，使用HTML代码等就是有特殊需要的发帖者才需要的功能，完全可以折叠起来，需要时再展开使用。

在任何一个页面，功能应该足够**少**使得网页不至于眼花缭乱，功能应该足够**清晰**使得用户不需要在两个相似的按钮之间“做出艰难的决定”，功能应该足够**完备**使得用户不会在想做某件事时找不到对应的按钮。这是应用程序开发中重要但容易被忽略的原则。

一个设计良好的网页应当是“**自文档化**”的，犹如一个编码风格良好的程序也应当自文档化一样，不需要任何附加的帮助文档即可充分理解各项功能的含义并在需要实现某项操作时快速准确地使用相应的功能。用户不会为了一个功能去查阅厚厚的用户帮助手册，也不会依次点击各个按钮进行试验，他们只会放弃操作，另寻他处。因此，功能（按钮、链接、输入框等）的布局、样式、图案及必要的说明文字都需要认真推敲。

例如，用户注册界面的输入框应该有必填项和选填项的区分，即使是用户名和密码这样简单易懂的输入框都应附有相应的简短说明，标明此项目的含义及系统的限制。如果用户不慎操作失误，最差的用户体验是什么都不提示，直到用户发现无法登录才想到没注册成功；稍好一点的是冷冰冰地提示一条“系统错误”并让用户重新填写表单；再好一点的是只要求重新填写上次填错的项目；可以称得上用户友好的设计是用醒目的颜色和图形标明错误发生的位置，要求重新填写的同时在旁边显示**操作建议**，犹如一个隐形的助手在用户身边指导如何操作。不幸的是，尽管在用户注册界面上多数有一定规模的网站都已经达到了“用户友好”的要求，但对其他操作可能就没花那么多心思认真处理了，“系统错误”往往成为网站打发走一位用户的最好借口，出了错误还什么都不提示更是让用户如堕五里雾中。

用户是不喜欢思考的，他们喜欢网站已经把所有可能的情况都考虑到，并对每种情况做出人性化的合理回应。尽管考虑所有情况在理论上是不可能的，但“自文档化”地应对至少95%的可能情况并不是很难的任务。为了做到这点，需要认真调研**普通用户**，而不是专业开发人员，浏览网页和使用网站功能的习惯。很多时候开发人员觉得很显眼的功能被多数人“无视”甚至“骑驴找驴”，开发人员觉得很清楚明白的说明文字在菜鸟眼里如同天书。

## 2.5 调试与评测

大型程序系统的开发过程有两种：自上而下和自下而上。**自上而下**的方式意味着开始编写的系统就是一个“能用”的系统，但其中的功能都是假的（模拟出来的）；**自下而上**的方式犹如制造汽车，每个零件工人都在不知道汽车全貌的情况下开发、组装。计算机软件各组件之间耦合复杂、逻辑严密的特征决定了汽车制造式的完全“面向对象”开发是不现实的，同时也容易挫伤开发人员的积极性，容易引入接口和规范的不一致性，为整体组装带来极大的麻烦。

“**自顶向下，逐步求精**”是大型程序系统常用的开发方式。具体到Web应用程序，以下的开发模式是不错的：

·用纸笔画出每个页面的大致框架，并确定程序流程；

·用HTML静态网页的形式实现，并完成CSS等页面设计；

·对每个功能用程序实际实现；

·将静态模板与动态程序结合在一起，成为可用产品。

其中第一步是编程开始前高层决策人员做出**整体规划**、明确设计文档的阶段。第二步和第三步可以同时互不影响的进行，第二步由页面设计与用户体验师完成，第三步由程序设计师完成，这个阶段也是开发流程中耗时最长的阶段，由于**前台页面设计**与**后台程序设计**两类性质不同的任务被分开，在提高任务并行性的同时减少了页面和程序人员对不熟悉的对方领域的盲目探索。如果第一步文档明确，设计合理，采用了**逻辑与显示分离**的策略，第四步将是一个很快的过程。

程序开发完毕，需要多少人来对其测试？开发者自身能否充当测试者的角色？第二个问题的回答是否定的。开发者往往考虑的是程序方面的整体性能，编程复杂度及应用某些“高级”方法的自我实现感觉；而用户关注的则是2.4节所述的易用性和可用性。因此普通终端用户是必须成为内部测试阶段的主力军的，且自视清高的开发者必须放下架子来认真听取他们的意见，根据用户价值取向作出改进，直到大部分用户满意。

网站的内部测试阶段并不像想象中的那样，需要很多人来测试。研究显示，5名负责任的普通用户就能发现80%的问题，而这80%的问题正是网站竞争力低下的关键因素。只要能把这些影响用户体验的**瓶颈**消除，剩余的问题其实并无大碍。

如果一个网站有某个功能非常难用，或存在严重的BUG，就可能出现**屏蔽效应**。用户在遇到这个非常不爽的严重问题后直接丧失了对网站的兴趣，从而也就无法发现后续的小错误了。例如学术网站如果没有对数学公式等必要工具的支持，学术人员在上面讨论不方便，自然不会在此久留；即使用户积分成长体系存在严重的不合理之处，也不会被用户发现。因此，评测应该**分阶段进行**，每一阶段找出影响最严重的“瓶颈”，解决后再组织下一阶段的评测，找出新的瓶颈并加以解决，如此循环往复，直至没有明显影响用户体验的瓶颈。

测试人员虽然不需要很多，但样本应有足够的**代表性**。不必说，测试人员应该属于网站**潜在用户群**，一个时尚女性网站让老大爷评测显然就不合适。应当细分网站的潜在用户群，每位测试人员具有一定的代表性，“各司其职”，例如针对大学生的网站，测试人员应该既包括重点大学的学生，又包括普通大学的学生；包括理科、工科、文科等各专业方向的学生；包括不同年级的学生；包括不同性别的学生。另外，针对不同的网络速度（宽带，拨号），不同的设备（电脑，手机）进行区别测试（甚至区别开发）也是应当考虑的。

在约请普通用户进行评测时，“**考试法**”是较为有效的。网站开发者充当“主考官”，在考试开始前出好涵盖网站主要操作的试卷，其中基本操作与高级操作应各占一定比例，题目的顺序要符合普通用户的操作逻辑，题目描述要清晰、符合用户习惯，但不对具体操作进行任何提示。考试时，采用屏幕录像机等方式记录用户的操作过程，同时主考官在一旁默默观察，发现不愉快的表情或动作时及时记录，这很有可能是用户感到困惑的地方。用户遇有问题，主考官不应予以解答，而要反问道“你觉得应该如何操作呢？”让他凭感觉去操作，同时密切关注这个用户体验不好的功能模块。考试后，不要忘了让用户说说自己的操作感受。统计每道题（每个操作）的完成时间、点击次数、误操作次数、停留过长或感觉不舒服的页面、让用户眼前一亮的页面等。

在大型网站的竞争中，技术是次要的，理念与用户体验才是主要因素。一套网站系统在正式发布前，至少要经历与开发时间相同的调试与测试。一般来说，大型系统的设计时间占整个开发流程的1/3，编码时间仅占1/6，模块测试占1/4，系统测试占1/4，切不可把编码当作开发的全部，更不能认为编码的结束就是开发的结束，导致上线的系统漏洞百出，威胁系统安全，激发用户不满。

# 第三章 系统结构

## 3.1 需求和设计目标

格物致知网的设计目标是面向大中学生和科教工作者的一站式网络服务平台。

用户群的需求包括：

·很多学生组织希望建立网络上的主页，拥有一套组织内外信息发布的便捷平台，但缺乏技术实力；

·大中学校的信息发布平台多为官方主办，信息的组织性、完整性、群众参与性都远远没有满足校内外人士获取详细、真实校园信息的需要；

·大中学生、科教工作者希望有一套功能完善的网络导学系统，但如今这样的导学系统价格昂贵，且属于单向灌输，缺乏互动与交流；

·希望在网络上方便地输入和传输数学、物理、化学等自然科学的公式、图像等；

·希望在网络上建立属于自己的结构化主页，但不拘泥于博客的简单形式；

·希望以更加真实和可信的形象出现在社交网络中，但希望以更加匿名的方式参与公众话题讨论；

·高级用户希望切身参与到网站的开发中去，而不仅仅是被动的使用者；

·组织与个人希望从网络中不仅获取精神上的自我实现，还能获得适当的物质利益。

为解决网民日益增长的网络交流需要与现有网站结构与运营方式僵化之间的矛盾，格物致知网大胆采取开放开发的战略：

·以用户为主线串联网站各大功能；

·普通用户能够方便地定制个性化主页与功能；

·组织可以方便地创建和个性化自己的网站；

·允许高级用户参与开发网站新功能。

为方便这个程序项目的开发，格物致知网底层需要一套完善的公共管理机制：

·为各功能模块的数据结构提供一种统一而又可扩展的定义和控制机制；

·提供统一的安全访问机制；

·在无需修改已有系统代码的情况下，加入新的功能模块；

·提供一组标准的API来对对象执行各种操作；

·提供一组方便的系统函数实现常用的逻辑功能；

·提供一种模板以分离逻辑和显示；

·方便地使用国际化语言包；

·强大、方便的程序调试系统；

·方便的版本控制系统；

·一份详尽、易读的开发文档和使用说明文档。

## 3.2 系统模型

格物致知网系统为实现完善、可扩展的公共管理机制，独创了层次化体系结构模型：

|  |  |  |
| --- | --- | --- |
| 内部显示 | 外部应用 | Application |
| 内部逻辑 | 外部API | Logic |
| 对象抽象层 | Abstraction |
| 系统内核 | Kernel |
| 版本控制与调试 | Develop |

·Layer 0：**开发层**，负责版本控制系统（Version Control System），调试器（Debugger），方便开发人员发现与定位程序中的错误，保存与控制测试版、发布版及历史版本，简化程序开发与调试流程。

·Layer 1：**内核层**，负责数据库、文件、网络等接口和开发者权限控制（Access Control），使得上层应用不需要考虑数据库、文件、网络的具体实现方式；为开放开发提供统一的数据库、文件、网络权限检查，避免危险的、越权的操作和应用。

·Layer 2：**抽象层**，负责管理和控制对象和模板，提供统一而可扩展的API，提供一组方便的逻辑功能系统函数，预设有用户、目录、文章-回复三种常用抽象模型；模板分离了逻辑与显示，使得前台界面设计师和后台程序师各行其是，并行开发。

以上三层均为内核模式（Kernel Mode），为保证效率，不受安全检查限制，程序的正确性和鲁棒性需要内核开发人员保证。

·Layer 3：**逻辑层**，负责程序流程逻辑，是开发者编写的最重要、最大的模块。对于外部应用，指外部调用的API。

·Layer 4：**应用层**，负责控制与显示。在请求处理端，充当分发器（dispatcher）的作用，接受用户输入并调用相应的逻辑层模块完成处理；在输入显示端，充当输出数据的页面显示机制，如HTML代码的生成和提示信息的显示。

以上两层为用户模式（User Mode），受到权限检查控制，只能访问受限的资源集合。被加入信任列表的用户模式程序，可以不受权限检查控制。

## 3.3 关键系统模块

## 3.4 数据结构

系统信息表：

·common\_priv\_db

·common\_model

·common\_style

·common\_setting

·common\_session

·common\_banip

·common\_credit\_rule

·common\_friendlink

·common\_advice

·common\_announcement

·common\_cron

·common\_smiley

公共信息表：

·common\_badword

·common\_district

·common\_university

·common\_highschool

·common\_middleschool

·common\_primaryschool

全局统计表：

·common\_stat

·common\_stat\_user

·common\_log\_sysop

用户角色表：

·common\_role

用户群组表：

·common\_group

用户数据表：

·user

·user\_password

·user\_puppet

·user\_friend

·user\_friend\_tag

·user\_follow

·user\_role

·user\_group

·user\_message

·user\_count

·user\_log

·user\_profile

·user\_setting

·user\_stat

对象数据表：

·stat

·content，privilege，style，language，config，credit，handler，hook

·draft

·backup

·approve

·msg\_in

·msg\_out

对象包括list，post

预设object model包括：

基本：id（序号），author（缓存作者名，以authorid为准），authorid，content，create\_time，modify\_time，privilege（访问控制信息）

Privilege是serialize的array，由15个priv的key=>value组成，value为二级数组，元素为若干条规则。

消息型扩展：dest（目标用户ID）

回帖型扩展：postid（主贴ID），floor（楼层）

文章型扩展：title，display\_style，tag（标签），last\_reply\_time（最后回复），view\_stat，reply\_stat，rate\_good（顶），rate\_bad（踩），stamp（图章），operation（操作记录）；

Open\_time（开启时间），close\_time（关闭时间），read\_perm（阅读权限），allow\_find（未达阅读权限是否允许查看标题），price（售价），advanced\_priv（高级权限控制是否启用）；

Bitwise Assess：highlight，digest，sticky（本版置顶），deleted（删除），blocked（屏蔽）；

Bitwise Settings：html，bbcode，parseurl

版块型扩展：moderator，short\_desc，description，display\_style，sticky\_post（置顶主题列表），recommend（推荐主题列表），follower（订阅者，便于推送更新）

## 3.5 版本控制系统

版本控制系统：网页根目录下建立四个文件夹：release、beta、code、history。Release是当前发布的正式版本程序源代码；beta是测试版本的程序源代码；code是所有历史版本的源代码记录；history是历史版本的程序源代码。Gewu.ustc.edu.cn指向WWWROOT/release/，而gewu.ustc.edu.cn/beta指向WWWROOT/beta/，gewu.ustc.edu.cn/code指向WWWROOT/code，code目录下即为版本控制系统。

版本控制系统首页为开发者登录界面，所有开发者均从这里登陆。开发者分为三级：管理组（admin），信任组（trusted），用户组（user）。管理组成员有权访问所有代码和开发文档，有权对信任组和用户组代码加以修改，有权管理开发人员；信任组成员有权访问所有代码和开发文档，但无权修改其他人的代码；用户组成员仅有权访问自己的源代码、公开的源代码和开发文档。

管理组成员个人有权任免信任组成员，对信任组和用户组成员执行禁止访问、屏蔽代码等操作，生成并保存当前发布版或测试版；两名管理组成员联手，可以任免其他管理组成员，修改其他管理组成员代码，对其他管理组成员执行禁止访问、屏蔽代码等操作；三名管理组成员联手，可以将当前测试版与发布版交换（测试版变发布版），将某个历史版本统一设为测试版或发布版；三名管理组成员联手，可以将某管理组成员或信任组成员的程序代码加入信任列表（trusted list），信任列表内的程序不受权限检查限制。将程序加入信任列表相当于程序在内核模式运行，任何恶意操作或错误均危及整个系统安全。程序取消安全检查确实有助于大幅提高系统效率的，经过严格的审核后方可加入信任列表。

History文件夹存放各历史版本，其下以日期命名的文件夹存放生成保存的每个历史版本，每个版本按照实际文件路径存放。Code文件夹存放用户每次修改的文件，按照实际文件路径+修改时间存放文件，依据数据库中的权限设置决定当前登录用户是否有权限查看、修改、删除文件。在文件夹视图中，显示release版和beta版；点击文件可进入历史版本页。对于有修改权限的文件，用户可以添加新版本、删除历史版本、将某个版本设为release或beta版。添加的新版本自动成为beta版，方便调试。

所有程序文件的上传统一通过开发平台网页接口进行，并在上传时进行必要的安全检查。提供文件批量上传功能，方便本地开发。事实上，上传文件即为添加某个文件的新版本。将某个版本设为release或beta时，将code目录中的相应文件改名放入release或beta目录，直接解析访问，不再通过版本控制系统。

## 3.6 安全访问系统

用户模式程序，除非被加入信任列表，均需通过内核层公共接口访问系统资源，尽量不要自行使用数据库、进行文件操作、发起网络连接等，不得使用安全访问系统认为有风险的函数、变量等。对函数、变量等的语法层校验一般由版本控制系统在上传时进行校验，而通过合法的内核API调用访问系统资源的，需要在内核层经过安全访问系统的权限检查。

对于数据库操作采用表级锁的机制，文件采用文件锁的机制，每个数据表有三种权限：查询（select），修改（insert，update，delete），设定表结构（create，alter，drop，index）；每个文件有三种权限：读（r），写（w），执行（x）；分别对所有者（owner）、所属开发组（group）、所有人（all）设置权限。类似UNIX系统的文件权限，每个表、每个文件的权限可以表示为rwxrwxrwx形式。较为开放的表、文件，权限可以设为764；较为隐私的表、文件，权限可以设为740。

注意，为了可扩展性和组策略的方便性，文件权限系统不是Linux文件权限系统，也没有使用数据库的grant、revoke语句，而是采用自己的技术实现的。

每个开发者属于一个主组，且可以属于若干扩展组，以各组中权限最高者决定开发者的程序是否有权限访问。对文件的访问将在文件上传时加以验证，而对数据库的访问检查则是在第一次访问时查询组策略记录，并缓存入内部数组。为了方便授权与控制，程序广泛应用了类的概念和private、protected属性，因此程序需要在PHP 5以上环境运行。

模仿UNIX系统，我们引入了sudo（superuser do）函数，仅对信任列表中的程序生效。sudo函数的作用是加入root权限标记，使得数据库操作不再需要验证。用完sudo操作后，需及时调用exitSudo函数关闭超级权限。程序只有本着尽可能少、尽可能安全的原则严格遵守sudo的用法才能通过信任列表审核。

系统中最敏感的用户表权限为700，所有者为root（根用户），所有针对用户表的修改与访问均需通过内核API进行。内核API验证调用者权限后，调用sudo函数提权后执行敏感操作，操作完成后恢复原有权限。其余内核API的实现原理也大致如此，因此如果对一个文件或数据库表没有访问或修改权限，应查找相应模块的API并注意当前用户是否有权限使用此API。

## 3.7 调试器

为方便在线调试程序，我们开发了断点调试系统。在release版本中，调试功能会被自动关闭，以提高效率。在beta版通过浏览器的调试过程中，可以在程序引用common.inc.php（负责建立DEBUG类）前加入全局变量：$DEBUG\_OFF = true; 则不再建立DEBUG类，与release版相同。

清空屏幕：$DEBUG->flushOutput()，清空输出缓存。

清空跟踪：$DEBUG->flushTrace()，清空所有跟踪信息重新记录。

调用跟踪：$DEBUG->trace()，将自动echo所有内核调用的文件、行号，和调用时间（以开始时为0），之后继续执行。若需达到最好的调试效果，最好结合使用清空屏幕，并在显示后退出程序，相当于断点。

变量记录：$DEBUG->record(varname)，以字符串方式传入变量名，会自动记录varname变量的值、调用文件和行号。若需记录多个变量，可多次调用此函数。不加任何参数，则只记录调用文件和行号，类似调用跟踪的功能。

记录显示：$DEBUG->showRecord(varname)，以字符串方式传入变量名，将自动echo所有记录过的varname变量值、调用文件和行号、调用时间。若需显示所有记录过的变量，不加任何参数即可。

显示所有：$DEBUG->halt()，先后执行flushOutput、trace、showRecord、exit。

结合使用调用跟踪和变量记录，既避免了单步执行的缓慢与复杂，又能尽快定位到问题所在。一般来说，每次调试一个模块，每次解决一个BUG是最合理的。BUG之间往往藕断丝连，最好在作出改动后重新调试，根据新的结果展开下一步调试。

## 3.8 注册表

Windows注册表形式的树形结构对于存储而言是较好的选择。

注册表里有HKEY\_LOCAL\_MACHINE，是计算机的全局设置，我们搞个全站设置，这个是管理员设置的默认值。

针对每个用户，还有HKEY\_CURRENT\_USER，指HEKY\_ USERS中的当前用户，是当前用户的设置，这样每个Windows用户可以充分个性化配置自己的桌面等，我们也可以搞一套出来。

HKEY\_CLASSES\_ROOT就是HKLM和HKCU中两个classes分支的合并。我们可以类似的搞个合并，有用户设置就载入用户设置，没有就载入全局设置。

HKEY\_CURRENT\_CONFIG是硬件配置，我们可以用作数据库配置、文件系统配置、网络配置等涉及操作系统的全局信息。

HKEY\_DYN\_DATA是控制硬件用的动态数据，暂时没有必要引入。

HKEY\_PERFORMANCE\_DATA不是真正的注册表值，而是系统运行时的动态数据，类似linux中的/proc，我们可以采纳其“虚拟目录”方式。

注意协调好哪些是允许用户自定义配置的，哪些是系统强制的。

我们的目录结构：

·SITE，全站设置，包括组策略

·USER，当前用户设置，未登录视为guest特殊用户

·CONF，涉及操作系统的全局信息，其中有一些是动态生成的虚拟目录，显示当前环境信息

·DATA，非设置类数据信息

在数据存储方面，采用用户分表的方式，为每个用户建立一个数据表，表中包括完整路径、所属位置、数据值。由于PHP是松类型语言，暂不设置数据类型，全部采用字符串。完整路径、所属位置建立索引，方便搜索某个特定项，或某个目录下的所有项。完整路径后带有“/”表示是子目录，不带“/”表示是数据项。

注册表与配置文件的区别，本质上反映了Windows与Linux的设计理念区别。Windows注册表存储于几个特定的二进制格式系统文件中，采用专门的算法和数据结构，相当于文件系统中的文件系统。Windows提供了一套标准的API使得信息能够有条理地存储到数据库中，适用于大型集中式开发，查询效率较高；Linux采用相对松散的配置文件方式，适用于分散的合作开发，但查询效率较低。早期的Windows也曾采用ini文件的形式存储系统配置，但后来被注册表取代了。Linux由于分布式开发、没有统一管理的特性，很难采用集中度高的注册表机制。但Linux机制的优点在于不易残留冗余信息（垃圾），方便外部应用开发等。

当然，采用注册表机制并不意味着像Windows一样采用封闭开发模式，我们完全可以开放注册表实现文档和代码，这与“注册表与配置文件”之争无关。

# 第四章 应用程序接口

4.1 内外API格式

 4.2 开发者权限

4.3 用户

 4.3.1 修改用户表数据结构

 4.3.2 添加、修改、删除用户

 4.3.3 条件搜索用户

 4.4 目录抽象模型

 4.4.1 建立、修改目录数据结构

 4.4.2 添加、删除、修改目录对象

 4.4.3 目录对象显示模型

4.5 文章-回复抽象模型

 4.5.1 建立、修改文章数据结构

 4.5.2 添加、删除、修改文章对象

 4.5.3 条件搜索文章对象

 4.5.4 文章对象显示模型

 4.6 自定义对象

 4.6.1 建立、修改对象数据结构

 4.6.2 添加、删除、修改对象

 4.6.3 条件搜索对象

 4.6.4 自定义对象显示模型

 4.7 安全验证与加密

 4.8 模板与语言包

 4.9 字符串处理辅助函数

 4.10 杂项函数

# 第五章 用户机制

## 5.1 设计原则

用户机制是Web 2.0网站的核心，好的用户激励机制能够促进大家共同建设网站，积极参与到网站的各项讨论活动中。格物致知网的用户机制需要：

·满足用户在不同环境下实名或匿名的需要；

·方便地进行用户管理的各项基本操作；

·满足用户按一定类别建立人际关系网络的需要；

·鼓励为网站作出贡献的用户；

·适当惩罚破坏网站氛围的用户；

·让用户在网站上感到有尊严，自我价值得到实现。

为实现上述原则，格物致知网创新了一套具有浓厚理科色彩的复杂用户体系和奖惩体系，但显得多而不乱，每项设置都有供普通用户使用的默认值，不希望过问用户体系的用户完全可以像使用普通网站那样使用格物网；高级用户则可以尽情发掘生动有趣的用户体系带给网络世界的青春活力。

格物网的用户分为两种：个人用户、组织用户。

## 5.2 注册与登录

作为Web 2.0网站，用户注册与登录几乎成为网站各项行为的前提与基础。因此，需要在网站首页的显要位置添加用户登录框和注册链接，明确引导用户先加入这个大家庭再使用。

网站登录显然是输入用户名和密码即可完成，登陆后默认进入的就是格物桌面。

而注册流程则需要简明快速，即使对计算机不甚了解的初级用户也能在指导下不费力地完成。

1. 填写真实姓名、用户名、密码；若不填写则不进行以下步骤。

2. 选择目前身份（工作，老师，博士研究生，硕士研究生，本科生，高中，初中，其他），选择省份、城市、学校或单位，填写学号或老师编号（可选，对于老师，推荐填写老师编号，已经工作的用户，推荐填写在校期间的学号）。

3. 学生选择班级，教师选择系别，选择好友。（可选）

未填写实名的，无法使用空间功能，因此强烈不建议。

用户统一标识符是用户的唯一身份标识。

对于校内学生，空间号由学校的英文缩写和校内的学号或老师编号构成，格式为“学校英文缩写-学号或老师编号”，对于无法提供校园学号的用户，使用“姓名-重名索引”作为用户统一标识符，方便统一查找和管理。

用户可以设置自己的用户统一标识符对外的可见性，对于不可见的用户，使用“用户统一标识符”无法访问到该用户的空间，此时用户可以绑定其他不与用户统一标识符格式冲突的字符串作为空间地址。

## 5.3 多密码权限

一些用户由于上网不便或其他因素，希望其他用户帮忙进行操作，但又不希望他看到或修改所有内容，因此有必要设置多个不同权限的密码，采用不同的密码加以识别。

用户注册时填写的密码为主密码，拥有用户的所有权限；可以在高级功能中设置其他密码，并为每个密码任意指定在网站不同模块的读/写权限。

## 5.4 多用户名

为满足用户在社交网络中真实可信的需求与在公众讨论中保护隐私的需求，格物网提供绑定昵称（马甲）的功能，除了真实姓名，每个人可绑定至多10个用户名，且这些用户名还可以注销、修改、被别人重用。由此可见，注册时用户名的概念被淡化了，很大程度上成为了用户的马甲与昵称。当然，已被绑定的用户名不能再被其他人绑定了。

对每个绑定的用户名可以设置是否链接到个人主页，即在网站中点击这个用户名是否能够链接到他的个人主页。由此，可以设置一些用户名“断绝”与真实身份的联系。

对每个用户名，提供注销和修改功能。注销是清除用户以此账号名义发表的全部信息，需要输入主密码确认。修改是将用户名修改为另一个尚未被他人使用的用户名，以原用户名名义发表的全部信息将改为新用户名。如果将用户名修改为同一用户的另一用户名，相当于将两用户合并。注销或修改后的原用户名不再归原用户所有，可以被其他用户使用，但将提示“这个用户名是被其他用户使用过后注销的，注册时间×××，注销时间×××，是否继续使用”。

用任意一个用户名、邮箱地址、用户唯一标识符均可登录网站，密码相同，多密码权限适用不同账户。工具栏显示“切换用户名”按钮，点击可切换至另一用户名，以另一用户名的身份浏览网站、发表内容。

在聊天大厦中，更可以在无需申请绑定的情况下独立设置在每个聊天室的昵称，方便用户在特定群体中的身份需要。

更牛的特色在于，在昵称管理中，可以为每个别人的用户名添加备注（例如姓名、班里的职务），这样看到新鲜事里乱七八糟的昵称，在显示用户名时将显示为“用户名（备注）”样式，再不用费心记忆昵称背后的TA到底是谁了。

## 5.4 好友，标签与权限控制标记

·用户分组：为了方便用户有条理的组织其他用户，用户可以对其他用户进行不同的分类，即将用户加入相应的用户组。一个用户可以属于不同的用户组。用户可以自己创建用户组，并设置其与其他用户组以及与标签的相互作用，比如用户可以创建用户组“男同学”和“女同学”，并且设置其他用户只能属于这两个用户组中的一个。

·目录：用户可以在任意用户组下创建目录及其子目录，以方便对一个用户组的用户进行进一步分类，这里的目录有点类似文件系统的目录，一个用户组中的用户只能同时属于一个目录。

·标签：用户可以为其他用户设置标签，方便管理，标签可以采用树形管理，即某个标签是其他标签的子标签。一个用户被加入子标签以后自动被隐式加入父标签，所有对父标签的权限控制自动应用于子标签。

·权限控制标记：权限控制标记实际上是一种特殊的标签，之所以说他特殊，在于它不会像其他标签一样在用户列表中显示，而是只是在设置页面显示。之所以把它同普通标签独立出来，是为了满足用户仅仅想对某些用户设置一定的权限，而又不想使用标签一面使得自己原有的分类显得混乱的需求。

用户组，标签以及权限控制标记的相互作用包括如下内容：

·不能属于（用户组）

·必须属于（用户组）

·必须具有（标签）

·不能具有（标签）

·必须具有（权限控制标记）

·不能具有（权限控制标记）

相互作用冲突的否决，自动添加与自动放弃原则：

1.名词解释：

·目标用户组：在添加到用户组操作中，要将某用户添加到的用户组

·目标标签：在添加标签的操作中，要为某用户添加的标签

·目标权限控制标记：在添加权限控制标记的操作中，要为某用户添加的权限控制标记

·被操作用户：要添加到用户组，添加标签，或添加权限控制标记的用户

·操作目标：目标用户组，目标标签以及目标权限控制标记的统称。

2.处理规则：

·若被操作用户属于目标的相互作用所设置的不能属于的用户组，或具有目标的相互作用所设置的不能具有的标签或权限控制标记，则操作被否决，并对用户作出相应提示。

·若被操作用户属于的用户组，具有的标签或者权限控制符的相互作用设置要求不能属于目标用户组或具有目标标签或格式控制符，并且被操作用户不符合上一中情况，则提示用户是否自动放弃原有的相冲突的用户组，标签或格式控制标记。

·若操作目标要求被操作用户属于特定用户组，或者具有特定标签或者格式控制符，并且被操作用户不符合上两种情况，则提示用户是否自动添加到相应用户组，添加相应标签以及相应权限控制标记。

有如下用户组归系统管理，用户不能对其删除，修改，如果用户不使用此功能，可以将其设为隐藏：

·收藏夹：为了方便用户对可以将用户添加到收藏夹中，以便方便地与其进行交流，进入空间等操作，加入收藏夹不需要通过对方审核。

·好友：加入好友后，就会接收到对方的新鲜事。只有加入好友，才能收到好友动态。

用户可以将其他用户添加为好友，用户添加其他用户为好友的时候需要对方审核，好友添加是单向的。为了防止用户加入好友的申请等待时间过长，用户加入别人为好友的时候，被加好友的人会自动成为我的好友用户组的成员，并标注上“等待验证”标签，该标签是系统标签，用户不可以手动删除。默认情况下，等待验证的好友的动态会被接受，如果用户不希望这样，可以更改设置。对方登录时将提示进行验证，并提示是否将对方加入自己的正式好友。若A已经将B加为好友，B再将A加为好友时不需A再次审核。一般来说，用户登录进行其他操作前，会先审核新加的好友，因此不会发生过多隐私问题。

同班同学自动成为好友，并发系统消息对涉及到的用户进行提示，可以根据空间号、真实姓名、用户名三种方式查询好友，并申请加入。为保护隐私，用户可以分别设置本班级、本校、所有人能否通过真实姓名找到我。

黑名单：同一用户不可以同属于黑名单和好友，被加入黑名单的用户发的站内信，以及聊天会被自动屏蔽。

用户可以为其他用户填写自定义称谓，这就使得不熟悉的昵称更方便查找了，同时也推广了自定义称谓这个功能。

在发送短消息等页面，点击用户列表中的人名或组名可以将其加入收件人列表中。

## 5.5 权限控制

格物致知网是以用户为中心的，因此给用户提供了强大的权限控制系统，允许用户对自己发表的，百科，导学，以及网站中以GPL，LGPL等共享协议发布的内容以外的一切包括论坛帖子，聊天记录，博客文章，状态等内容进行权限控制，用户可以自由地选择同意或者拒绝其他用户对他的帖子浏览，空间访问以及消息发送。由于百科，导学系统遵循知识共享 署名-相同方式共享 3.0协议，不支持权限设置。由于论坛是公共讨论区，过多地对帖子设置权限会影响讨论氛围，降低用户讨论积极性，因此普通用户对帖子设置访问禁止权限需要管理员或有权利的版主审核。用户对其他用户的权限控制按照以下方式进行：

用户发布帖子，文章，状态等内容时，在编辑器的下方总是会出现一个简单的权限选项，并附上一个高级权限控制按钮，点击按钮即可进入高级控制选项。

在简单选项里仅有三个按钮：公开，好友可见，私人，默认选中公开。

高级选项对权限的控制类似于一款优秀的防火墙Look 'n' Stop，采用按顺序从上到下按顺序的过滤器模式，用户可以以任意顺序添加规则，规则的格式如下“对象：用户，用户组，标签或者权限控制符；措施：禁止，允许”。有一个默认允许或默认禁止（Policy Accept/Deny）选项，以保证所有用户都能在通过过滤器以后确定下来是允许还是禁止。

权限包括9种：

对象自身权限：

·READ

·WRITE

下属对象权限：

·SELECT 查找对象

·INSERT 创建新对象

·DELETE 删除对象

·UPDATE 修改对象内容，包括其权限

·MANAGE 管理对象（如加精华、推荐等）

·APPROVE 审核下属对象

过滤器：

OPERATION PRIVILEGE CONDITION VALIDATION EXPIRATION FLAG

OPERATION：ACCEPT/PENDING/DENY （允许、审核、拒绝）

PRIVILEGE：9种权限，逗号隔开

CONDITION：条件，可以使用已定义的任何变量，但为保证简洁，只能使用变量与常量间的等式或不等式，如 userid = 5，roleid = 1，level >= 5。

VALIDATION：生效时间，用-v指定

EXPIRATION：过期时间，用-e指定

FLAG：从过滤器中跳出时，FLAG将被置为此处的字符串，用于后续分支判断处理。用-f指定。

处理逻辑基本在权限控制系统完成。

游客（未登录用户）权限另外设置并记录，以提高执行效率。

下属对象权限是可以继承的，即无论下属多少层，仍然拥有相关权限。

查询某种权限时从根对象开始，自上而下依次匹配下属对象权限规则，如果匹配到明确的ACCEPT/DENY规则，立即执行；若未匹配到，则按照下层对象的规则匹配，以此类推，直到当前对象；当前对象按照自身权限规则匹配，若仍未匹配到，则按照当前对象的DEFAULT设置决定是否拥有权限。如此即可实现白名单、黑名单、上级管理下级等策略。

权限采用“写时复制”策略，默认继承上级权限，仅当发生修改时才复制一份，采用当前权限。有若干种默认权限模板供初始化使用。

## 5.6 找回密码

如果忘记了格物网“多密码权限”中设置的其他密码，可以直接用主密码登录重置之，不需要找回密码功能。

如果忘记了格物网主密码，有两种解决途径：密码重置邮件、人工申诉。

推荐方法：发送密码重置邮件到用户资料中指定的邮箱，用户点击邮件中的密码重置链接，通过认证后即可设定新密码。

若未设置有效的电子邮件地址，可以人工申诉，须填写申请表单，包括：

·基本资料（姓名，详细地址，身份证号，当前身份）

·联系方式（电子邮件，临时登录密码），用于通知申诉是否通过，如果通过将附带密码重置链接。可以用原有用户名和临时登录密码登录格物网，但通过此密码登陆后只能看到用户申诉最新进展，不能进行正常操作。

·用过的密码（包括“多密码权限”中的其他密码）

·最常用格物网地点

·在格物网曾经填写的未公开个人信息

·其他说明文字

以上各项均非必填，但填写的项目越多，越有助于申诉成功。申诉是人工处理的，将根据资料完整性、真实度审核是否用户所有者本人。

人工申诉通过审核后，将暂时锁定账号，原有密码失效，只能使用临时登录密码登录此用户，并显示重置主密码页面，防止可能的密码窃取者进一步破坏。

密码重置链接仅在生成后7天内有效，逾期需重新申请重置密码。

## 5.7 积分规则

## 5.8 用户组

## 5.9 勋章

## 5.10 道具

## 5.11 处分条例

# 第六章 网络服务结构

## 6.1 总体设计

格物致知网站采用基于模块的统一的构架实现大部分功能，每个功能本质上都是一个模块，模块以空间为载体，这类似于操作系统的桌面，用户可以把在桌面上添加各种小插件来实现一些功能，实际上web操作系统是一些人对web3.0的定义 。用户进入格物致知网就相当于进入了一个web操作系统的桌面，只是这个操作系统不是用于文件处理的，也不是用来娱乐的，他的一切核心在于：为所有数学、物理、计算机等科学的爱好着提供一个交流的平台，由于格物致知网秉承以用户为中心、创新每个人的设计理念，用户是网站的设计者，自然网站界面的决定权也就归用户，格物网的用户主界面犹如一个“桌面”，贯穿于整个网站，用户在网站的大部分功能都是在格物桌面中实现的。

格物桌面功能包括提供了一些系统模块，同时也对外开放API，支持用户自己开发API。格物桌面提供的系统模块有如下几个：

·好友列表：类似QQ的好友列表，采用类似资源管理器的树形模式，按照用户组对用户进行分类。

·我的订阅

·好友动态

·主显模块：格物网的几大功能，采用页首主导航的模式，默认有首页，论坛，个人主页，聊天大厦，百科，导学，组织，七个导航供用户切换。默认与当前所在导航不同的在新页面打开，用户可以更改设置。用户点击用户工具栏中的相应按钮，以及页底的发牢骚等均自动显示浮动窗口，在窗口内相应内容。

·站内信

·我的论坛

·我的百科

·我的导学

·我的组织

·今日热点

·留言板

·聊天大厦

·小字报

·漂流瓶

格物桌面是格物网的核心，用户对网站一切功能的访问均通过格物桌面，应当下最多的功夫予以优化。

## 6.2 界面设计

格物桌面的页头使用统一的简洁样式，从左至右包括网站LOGO（左上角）、主要功能导航（顶部）、用户工具箱（右上角）。

用户工具箱从左至右包括：用户名（点击进入个人中心）、用户组与积分（点击进入我的贡献）、我的空间（点击查看空间新提醒等）、我的论坛/百科（根据所处位置显示）。用户点击用户工具箱中的按钮或链接，自动在主显模块中创建新选项卡（如果已有，则自动切换到相应的选项卡，而不是新建选项卡）

在网站所有界面，界面默认布局如表所示：

|  |
| --- |
| 页头 |
| 主显模块 | 小字报 |
| 好友动态 |
| 我的论坛 |
| 好友列表（浮动） |
| 页底 |

格物桌面底部信息统一，均设置“提建议”，“发牢骚”，提建议和发牢骚会自动提交到相关页面的管理团队，发牢骚便于定位用户产生问题、感到不满的功能并加以改进。提建议则为用户为网站发展出谋划策提供了便捷的途径。待管理团队答复之后会有相应的消息提示用户，要求牢骚必须关于我们网站，这样有利于我们收集用户体验。

例如：

好卡，一直登陆不上去……

论坛的搜索功能好难用……

XXX板块的图标太难看……

网站应尽量少用下载速度慢的图片，多用CSS等技术简洁明快地实现各类效果，不走网页复杂化、花哨化的老路，最大限度提升访问速度和简单便捷的用户体验，多给用户畅想与创造的空间。

## 6.3 配置设置

由于格物网采用了“个性化网站”的设计理念，用户有着大量的个性化设置，但这也为设置的管理带来了不便。因此，格物网提供了“备份与还原”功能。

备份：用户可以对整个网站或某一模块的设置进行备份，创建还原点。

还原：用户可以将整个网站或某一模块的设置恢复到包含相应模块的任一还原点。例如对整个网站进行了备份，可以只把某个论坛的相应设置恢复到这个还原点，而保持其他模块设置不变。

恢复默认设置：用户可以将整个网站或某一模块的设置恢复到系统默认状态。

为防止误还原，在还原或恢复默认设置前将提示用户是否备份当前设置，并要求用户输入主密码。

允许备份和还原的“独立模块”包括：

·一个相对独立的论坛

·一个相对独立的百科

·一个相对独立的导学

·一个组织主页

·个性化首页（即导航中的“首页”，用户入口和操作核心）

·个人主页

·聊天大厦

## 6.4 聊天大厦

空间中设立专门的“聊天大厦”，楼层数为基本分类，如学术、科技、人文等；所开的聊天室即为房间号，按照开放顺序自动分配。只有在聊天大厦中显示的聊天室才拥有房间号。

聊天室有四种开放形式：

·开放，所有人可以任意加入讨论，这是默认创建的类型；

·加入需审核，所有人都可以看到讨论内容但发言前需先经聊天室管理员审核加入此聊天室；

·旁观需审核，欲看到并加入讨论必须经过聊天室管理员审核；

·私密，对外不可见，也不在聊天大厦中显示，只通过聊天室管理员手工邀请。

聊天大厦模块首页犹如一座夜间灯火通明的大厦，每个方块象征一个聊天室（房间），凡是有人正在其中参与讨论的显示为点亮。方块内注明聊天室名称、人数和简介。

创建聊天室，必须填写聊天室名称，选择聊天室所属分类和开放形式，选择聊天室是否在聊天大厦中显示，可以填写聊天室简介作为聊天第一帖和聊天室名称的补充说明。

聊天室的创建者自动成为聊天室的管理员。聊天室的管理员可以设置其他人为管理员，同样两名聊天室管理员也可以联名免除一名聊天室管理员。若被免除的聊天室管理员为聊天室的创建者，操作需要管理员审核。

聊天室采用Ajax技术构建，力求简洁、方便，在不刷屏的前提下达到最好的聊天体验。

聊天室的聊天内容可由聊天室管理员推送至论坛，当然在推送时可以选择聊天发言的一部或全部予以发表，选择时可使用发表人、长度、关键字等过滤规则，左侧是全部发言，右侧是选中发言，鼠标拖放筛选希望推送至论坛的聊天记录。

一个人新进入聊天室时，默认显示最近10条聊天记录，当然聊天室管理员可以修改这个默认选项，进入聊天室的新人也可以显示更多的聊天记录。

聊天室管理员有权在聊天进行的任意时刻屏蔽之前的部分或全部发言，这个功能适用于讨论到某些私密问题而不希望以后新来的人看到的情况。

聊天室是一个可扩展性很强的应用，例如每个公告通知、好友新鲜事，都附带一个开放、在聊天大厦中不显示的聊天室，创建人是公告通知或新鲜事的发表者。

尽管聊天大厦的设计目的是公用聊天的场所，但若要实现点对点的二人聊天，也可以使用聊天大厦得“私密”功能，只邀请对方加入就行了。为方便常用的二人聊天，点击好友列表中的一个人，默认动作就是与他聊天，在其他模块的用户名上停留，也会出现进入空间、查看个人资料、与TA聊天等选项。

## 6.5 小字报

网络的传播快速性能够让一件奇闻轶事在一夜间迅速走红，这种放大效应是传统媒体所不具备的。格物致知网进一步利用网络的放大效应，专门开辟了“小字报”模块，默认在网站所有页面的右边栏显示，所有用户都可以将最值得分享的内容发上“小字报”与全站浏览者共享。

默认情况下，小字报模块显示最新10条小字报，小字报的字数限制在25字以下，若需更多解释，可以发表详细内容，浏览者点击小字报链接可进入查看。点击小字报，可进入相应的评论区对这条小字报发表评论。

小字报采用循环滚动方式显示，默认每10秒滚动一条。为防止滥发小字报，限制发表小字报的时间间隔不能短于1分钟，同时处于循环滚动列表的小字报不能超过10条。

发表小字报时，可以选择循环滚动的三种方式：只显示一次，累计播出若干次（不超过100次），自发表之时起累计播出若干小时（不超过24小时）。达到次数或时间限制后，小字报退出循环滚动列表。

发表小字报可以选择投放范围，目前包括整个网站、全校两个选项，默认为全校。

今后访问量较大后，可以对发表小字报进行权限和积分上的必要限制。

在技术实现上，小字报采用循环链表，指针指向当前位置，每10秒下移一格。添加新小字报时，插入到当前位置与下一小字报之间。

为防止访问冲突，采用了数据库互斥锁和忙等待的策略，在添加和删除小字报时进入Critical Section，防止链表指针冲突。

每次调用小字报时，获取Last\_Display，若差值大于10秒，则当前位置指针移动至下一个小字报，并更新Last\_Display。若下一小字报已经过期，则删除过期的小字报。位置指针的循环滚动与添加删除小字报之间不会发生严重冲突。

由于小字报是全站相对稳定（每10秒更新一次）的内容，访问量巨大，采用缓存技术，当不需要更新时直接读取缓存。今后将采用服务器端Daemon的方式，由持续运行的守护进程负责小字报列表的维护，降低数据库压力。

小字报内容由负责版主监管，未来人数较多后，发表小字报可以加入权限限制，须经审核。

## 6.6 今日热点

一些共同关注的话题因讨论人数众多，容易引起大家积极参与的兴趣。因此格物致知网管理团队每天选择一个热点事件设为今日热点，以开放聊天室的形式供大家自由交流。

今日热点默认在全站首页、个人空间、聊天大厦显示，当然其他应用也可以添加这个模块。

今日热点一般选择有一定争论性的话题，选材非常广泛，可以设置正反双方或开放投票以激起大家兴趣。

## 6.7 漂流瓶

为满足同学们随机交友的愿望，格物致知网开通了“漂流瓶”功能，可以给指定范围内的一个随机用户发去一条消息。

可以指定的范围包括：学校（默认是同校），班级，性别，家乡，兴趣爱好。这样，只要对方在个人中心填写了相关个人信息且对外开放且开启了漂流瓶功能，就有机会接收到漂流瓶，进而找到与自己志同道合的人。

## 6.8 自定义模块

·RSS订阅功能，订阅网站外部消息

·To be continued…

# 第七章 消息系统

## 7.1 总体设计

消息是系统提示用户相应状态变更的便捷方式，也是用户之间相互交流的便捷方法，是用户最常使用的功能之一，人性化的消息提示系统能给用户带来好的体验，相反，不好的消息提示会使用户摸不着头脑。消息分为用户消息和系统消息两大类。

格物网登录后的默认首页由页头、主体和右边栏构成：

|  |
| --- |
| 页头 |
| 短消息 | 好友新鲜事 | 小字报 |
| 发送短消息 | 发表新鲜事 | 今日热点 |
| 我的组织 | 我的论坛 | 我的百科 | 好友列表 |
| 组织导航 | 论坛导航 | 百科导航 |

“好友新鲜事”是好友发表的最新动态、好友个人主页的最新文章。

好友新鲜事下方的“发表新鲜事”输入框使得用户可以迅速与朋友共享自己的生活点滴。

在好友新鲜事模块，点击消息进入讨论，作为一个私密聊天室（见6.3节）。

“我的组织”是所加入社团、班级、组织及的最近更新和通知信息。组织可以通过发布公开或内部的信息来向成员发布动态、通知（见第8章）。

“我的论坛”是我加入关注的讨论区的更新和动态，及自己所发帖子的被回复、被执行管理操作的提醒，及自己加入收藏的论坛帖子。

“我的百科”是我加入关注的百科分类、百科词条的更新，及自己所参与编辑的百科词条被编辑的提醒。

在关注模块点击消息进入被关注的页面。

组织、论坛、百科导航使得用户可以设置一些常用区域的快速链接。组织导航默认包括所属班级、院系、学校和加入的社团组织；论坛导航默认是论坛的一级分类；百科导航默认是所属学校百科。

五大模块都可以设置信息来源及何种类别的动态（新文章、新回复、管理操作）才“值得”推送给用户。

本着由用户选择的精神，所有模块都可以添加、移动、改变大小、删除，犹如一个个显示在桌面上的对话框。为完善用户体验，模块采用JavaScript + Ajax技术构建，及时更新最新消息，减少刷屏和数据流量。

每条消息前面设一个红绿图标，使用不刺眼的红色表示未读，绿色表示已读，在用户点击消息后自动变为已读，用户也可以点击图标直接切换未读和已读的状态。点击消息的头像或用户名，进入TA的个人主页。

## 7.2 用户消息

如果是在论坛、百科、导学中直接点击发送消息链接填写的短消息，点击发送消息的时候自动在输入框的第一行加入“消息发送自\*\*\*\*\*（链接）”，如果用户不需要可以自己删掉。

若为用户发来的消息，点击消息出现快速回复对话框，可以迅速回复短消息或发起聊天。

短消息下方的“发送短消息”输入框包括目标用户名或用户组，标签，目录（好友列表中的分组）、消息内容、发送信息和发起聊天两个按钮，使得用户可以迅速向某个用户或用户组中的每位成员发短消息或发起聊天。多个目标用户名或组名用空格或逗号隔开，可以实现短消息群发。

## 7.3 系统消息

若为系统消息，点击消息出现相应详细内容，如用户组升级消息，点击进入积分界面；网站活动通告，点击进入活动页面。消息提供相关链接：比如联系管理员，联系版主（版主帖子操作的时候，谁操作的联系谁）。

不同消息的显示格式：（消息前面均注明发送时间）

用户组升级：您的用户组升级为<a href=用户组页面>小学一年级</a> 联系管理员（链接，给管理员发消息）

好友发表日志：<a href=boj的个人空间>boj</a>发表了日志<a href=日志链接>格物致知网</a> 与他聊天（链接）

帖子或词条被管理操作：您的帖子<a href=帖子链接>帖子名称</a>被<a href=版主的个人空间>版主用户名</a>加为精华，操作理由：原创内容 给他发消息（链接）我要上诉（如果是删帖关闭等负面操作，链接，给上一级版主或管理员发信息）

网站举行活动：活动说明 + 查看详情（链接）

好友申请：<a href =XXX的资料>XXX</a>申请添加你为好友，同意，拒绝，拒绝并加入黑名单（三个链接）

审核通过：您的<a href=帖子链接>帖子名称</a>已通过审核，审核人：<a href=版主的个人空间>版主用户名</a> 给他发消息（链接）

审核拒绝：您的<a href=帖子链接>帖子名称</a>没有通过审核，审核人：<a href=版主的个人空间>版主用户名</a> ，理由：XXXXXX 给他发消息（链接） 我要上诉 （链接）

评论： <a href=boj的个人空间>boj</a>对你的日志<a href=日志链接>日志名称</a>发表了评论：XXXXXXXX 去看看（链接） 与他聊天 （链接）

C语言online judge状态更改：您对<a href=问题页>XXX问题</a>提交的程序测评结果：<a href=结果页面>XXXXXX</a> 排名：<a href=排名页面，注意跳转到自己位置所在的页面，而不是第一页>XXXXX</a>（针对不同项目进行的排名各做一个链接）

# 第八章 个人主页模板

## 8.1 总体介绍

个人主页，即属于个人的网站，是用户在网上展示自我的平台。目前各大网站的个人主页服务多以博客形式，博客的特点是发表迅速，管理方便，但弱点是可定制性不强，难以添加模块和栏目，一眼就能看出是哪家提供商的博客。

格物致知网个人主页吸取了博客快捷方便的优点，着力改进可定制性不强的缺陷，提出了“结构化博客”的理念。在结构化博客中，用户可以在无需修改代码的情况下方便地添加、修改动态页面模块；提供更多一键修改的样式，让自己的博客看起来像内容发布系统（CMS），论坛，百科或其他。页面的所有元素在强大的后台管理平台都可以自行修改，不再受到统一CSS样式表的限制，甚至可以上传制作好的静态网页、符合网站接口规范的模板，真正让个人主页的外观和样式随心所欲。

## 8.2 界面结构

用户点击个人主页的界面设置的时候，会打开一个新网页，供用户进行界面设置。

个人主页目前提供如下个性化应用（模块，可添加在页面任何位置，采用图层、漂浮等技术）：

·时钟

·日历

·成长简史，可以写写个人简历

·新鲜事，即空间的本人新鲜事

·留言簿（涂鸦板），供访客快速留言

·投票

·辩论，采用公共聊天室形式（见6.3节）

·随机文章，随机显示几篇博客文章的链接

·最新文章，最近几篇博客文章的链接

·最热文章，几篇点击数最高，评论最多，被“顶”次数最多的博客文章

·推荐文章，用户自己推荐展示的几篇优秀文章

·相册

·音乐播放器，点几首歌给大家听

·个人简介，一段放在任意位置的文字

·博客导航，博客的结构化目录

·访问统计

·我发表的组织主页文章

·我发表的论坛帖子

·我编辑的百科词条

·我的C语言测评排行榜，显示名次比较靠前的排行榜，具体显示排名优于多少的排行榜由用户指定

·将来根据导学系统的变化增加，各种榜都可以拿出来秀

·我发表的精品帖子

·我发表的推荐文章

在分类方面，结构化博客允许标签采用树形管理，即某个标签是其他标签的子标签。一个用户被加入子标签以后自动被隐式加入父标签，所有对父标签的权限控制自动应用于子标签。由此，标签可以形成一个分类目录，一篇文章由于可以设置多个标签，也可以同时出现在分类目录的多个位置。

在分类目录页，可以显示文章的一部或全部，也可以只显示文章标题，更可以模仿信息发布网站采用分类显示的格局，有助于浏览者了解博客的定位和方向。

在文章显示页，有三种样式可供选择：较为严肃的信息发布型，普通活泼的博客评论型，平等开放的论坛讨论型。

发表文章时，可以：

·使用传统的编辑器、附件和图片上传

·添加各类应用模块

·指定文章摘要以在分类目录页显示恰当范围的内容（现在博客的摘要往往是从一段中间拆开了）

·根据标签和标题自动生成相关文章，或手工指定相关文章，使得浏览者看完一篇后又被相关文章吸引

·关闭或打开评论，评论是否需要博主审核

·将论坛帖子、百科词条等作为文章发表 ，摘要如其他文章一样显示，但点击跳转到原来的发表链接。便于用户将在论坛、百科中发表的内容一键转到自己的个人主页集中展示。

用户文章发表后再次编辑时，可以指定关键字，读者点击关键字就可跳转到具有相同关键字的文章，关键字由系统自动根据文章生成。

默认的个人主页界面：

右侧边栏自上而下是个人资料、访问统计、博客导航、最热文章、新鲜事、留言簿。

在目录界面中，左侧主要区域默认是传统博客样式的摘要显示型文章列表；在文章显示界面中，左侧主要区域默认上部是标题、标签、文章正文，下面是阅读统计、“顶”按钮、相关文章链接、上一篇下一篇链接，之后是分页显示的评论内容。

如前所述，所有组件均可增删、调换位置。

## 8.3 访问地址

个人主页的访问地址可以使用“用户统一标识符”，也可以绑定其他不与用户统一标识符格式冲突（由字母、数字和下划线构成，首个字符不能是数字）的地址作为个人主页的访问入口。每个用户在“用户统一标识符”外至多绑定一个地址。如果设置了“用户统一标识符”不可直接访问，则绑定的地址成为个人主页的唯一入口。

# 第九章 百科模板

## 9.1 总体设计

格物百科的宗旨是建立一个由全体网民共同编辑维护的开放校园信息平台。百科的生命力来源于社区共同贡献的活力，生长点在于社区共同了解且感兴趣的公众话题。

根据百科的性质，百科适合发表公众性强、稳定性强的知识性信息，不适合

·个人信息、随感、杂记，应放在个人主页；

·时效性强的新闻，应放在组织主页；

·讨论话题，应放在论坛或聊天室。

·学术信息，应放在导学系统。

格物百科首页由科大故事、科大新闻、科大教学、科大生活四大板块构成。

·科大故事：校史、人物、奇闻轶事等。

·科大新闻：每人都可以发表的科大新鲜事，采用网站小字报系统，滚动播出。

·科大教学：院系、专业、教授、课程、教学成果、科研成果、奖学金与出国保研政策等。

·科大生活：社团活动、生活信息、校园文化、校园地图、宿舍、食堂等。

## 9.2 分级控制体系

格物百科采用三级管理体系：网站管理委员会、校园百科管理委员会、普通用户。

网站管理委员会是整个格物百科的开发团队和管理机构，负责：

·开发、优化百科各项功能

·维护百科系统正常运行

·管理各校园百科

校园百科管理委员会是一个相对独立的校园百科的管理机构，负责管理校园百科内部事宜，由核心管理团队、分区管理员组成。

核心管理团队由一名总管理员牵头，仅有总管理员有权任免核心管理团队成员并划定职责范围。核心管理团队拥有校园百科的完全控制权，可以修改各项系统设置，进行增删版块、调整界面等操作。当然，核心管理团队成员只有在总管理员划定的职责范围内拥有管理权限。

分区管理员是校园百科某个一级或二级分类的“版主”，有权修改所负责分类的界面显示方式和各项设置，进行审核、锁定词条等管理操作。

## 9.3 词条与评论

一个词条包括以下元素：

·词条名称

·词条分类，词条所属的唯一树形分类

·词条状态，是否被锁定，未被锁定的显示编辑词条链接

·词条名片，以结构化模板的方式显示一些常规信息

·词条简介，用简短的文字介绍词条的基本内容、定义等（正文第一个标题前的内容）

·词条目录，若词条正文中一二级分类总数达到五个，则在词条最前面加入方便导航的目录。

·词条正文，推荐以一二级标题的形式将其结构化。

·参考资料，在正文中可以添加引用参考资料（文字或网址）的标记

·开放分类，不超过五个进一步描述词条属性的标签（tag）

·此词条对我有帮助，点击给词条加一票

·词条统计，被访问次数等信息

·历史版本，词条被修改的历史记录

·词条评论，对此词条内容或其描述的对象的评论或问答。采用二层结构，即可以对评论再评论，但不准添加第三层评论，以防盖楼。提问的第二层评论就是回答了。

词条评论包括两类：提问和普通评论。未解决的提问按照从新到旧的顺序将放在其他所有评论之前。其他评论也按照从新到旧的顺序排列。回答问题后，建议其将回答写入百科词条，这样就可以在用户需求的激励下不断更新需要的内容了。

提问者和版主可以在未解决提问的每个回复中看到“设为最佳答案”链接，在提问中看到“关闭此提问”链接。被设为最佳答案将有一定积分奖励。

词条右侧具有边栏，自上至下默认为：

·网站小字报

·访问统计，点击次数，顶、踩次数，编辑次数

·历史版本，显示创建者和最近10条编辑记录（用户名、时间、字数）。

·相关词条，展示与此词条关系最密切的10条词条。默认情况下为系统根据词条名称、分类、标签自动生成，当然鼓励手工设置、添加。

·百科活动，百科管理委员会举办的活动信息

·需要您解决的问题，展示与此词条关系最密切的设为“提问”的10个词条评论

·优秀词条随机展示，随机展示10条由百科管理委员会评定的优秀词条

词条正文默认的显示方式：

·左上角词条标题，右上角词条状态

·词条名片，右侧是词条简介（即正文第一个标题前的内容）

·词条目录（如果存在的话），点击跳转到相应章节。

·格式化显示的词条正文，一级标题附有“编辑本段”链接。

·添加段落按钮；支持反对、加关注、申请保护等链接；展开/折叠按钮，点击展开/收起词条正文。

·词条评论框，供发表评论，发表按钮旁边设置“这是一个提问”复选框，提问将放在普通评论之前。

·树状显示的词条评论，采用逐级缩进的新闻网站评论样式。

对于知识性、权威性较强的词条，上述内容自上而下顺次显示。对于娱乐性、争议性较强的词条，上述内容折叠显示，即默认展开/收起按钮状态为收起；争议词条本身不应该有鲜明的立场，而是应该保持中立，用户可以通过标题并列的形式体现这种中立性。这是为了将评论内容更多展现在浏览者面前，吸引大家去评论。

多数分类默认展开显示词条正文，但奇闻轶事等分类默认收起词条正文以吸引评论。编辑词条时可设置自动、展开或收起的显示方式。

对于词条显示方式为自动，词条内容和评论内容均达到300字，且评论内容字数多于词条内容的，意味着评论火爆，将自动采用收起正文方式显示；反之采用展开正文方式显示。

## 9.4 分类目录

格物百科是一个结构化展示校园信息的平台，因此合理的分类目录是不可或缺的。格物百科的分类目录采用树形结构，一级分类为四大板块，二级分类为管理团队设置的内置分类，三级以下分类允许用户自行创建。

一个分类可以既是目录，又是词条。进入这样的分类时，显示的是目录页面，但在标题下方显示词条内容被折叠的提示，点击展开可以在标题和目录间显示词条内容。

分类目录可以自定义个性化显示，目前采用加强版HTML代码形式，基本形式是若干矩形块（div），参数在后台予以解析：

·style，即div矩形块的css样式

·display，展示方式，包括标题链接式（默认）、标题+摘要式、标题+图片式、标题+摘要+图片式等。

·listyle，li列表的css样式，据display默认样式不同。

·title，标题名称

·num，显示条数，标题链接式默认为10，其余默认为1。

·item，人工选取的词条名称，多个词条用逗号隔开。item<num时才根据source和order自动选取词条。

·source，自动选取来源，默认为当前目录，支持category（分类）和tag（标签）的与或逻辑表达式。（如来自“格物致知社”分类且标签为“网站”，或来自“格物致知网”分类，表达为cat=格物致知网 OR cat=格物致知社 AND tag=网站）

·order，排序依据，包括最新、最热、被推荐、最好评、浏览数最多、编辑次数最多、随机等。

分类目录目前主要靠编辑个性化代码设置，今后会考虑可视化编辑功能。

## 9.5 保护与审核

对一些较为敏感、权威或私人的词条，有时不希望让别人随意修改，因此格物百科设置了“锁定”与“保护”功能。

未被锁定或保护的词条，词条底部会出现“申请保护”链接，实名用户可以说明理由，申请保护词条。一般需要个人申请保护的词条都是与本人相关的。

被锁定的词条只有版主以上管理人员可以改动。

被保护的词条改动只有版主以上管理人员有权直接进行，其他人进行的改动必须通过上述人员之一审核方可生效。

保护词条的方式为修改审核，默认的审核条件为“任何改动”，但有些词条可以允许小幅改动而无需审核。可以设置的不需审核的修改方式包括：

·添加内部链接

·添加引用（参考资料）

·修改版式

·小幅内容修改（按照字符串贪心匹配算法计算原有内容占新内容的比例，可以设置一个阈值）

## 9.6 编辑冲突

对于热点类词条，编辑操作十分密集，很可能会出现编辑冲突事件，如一个用户正在对一个词条进行修改，另一个用户又点击了编辑按钮，如果不加控制，就会导致后提交的用户覆盖先提交用户的修改。

为减少此种问题，推荐用户使用“编辑本段”或“添加段落”功能，至少不同段落间的同时编辑不会发生冲突。为判断每段编辑是否冲突，数据库中除记录整个词条的最后修改时间外，还用两个数组分别记录每段的最后修改时间（修改正式生效时间）、每段的最后编辑时间（点击“编辑”按钮或自动保存有修改的副本的时间，已提交生效的编辑除外）。

若词条的一段正在被编辑（最后编辑时间距当前时间小于5分钟），则点击“编辑本段”后提示“本段落正在被编辑，建议您稍后再编辑”。如果已经点击“编辑”按钮的用户在5分钟内没有任何操作（自动保存的草稿无变化），将视为未正在编辑。

若在同一段中已经发生编辑冲突，对于提交时已经有新版本的词条，将提示“词条在您的编辑过程中已经被修改过，建议您查看最新版本以避免编辑冲突。”类似地，在“草稿箱”中出现的编辑冲突，也将在草稿箱中进行提示。

智能合并功能将在今后开发。

## 9.7 同义词

一个相同含义的词可能有多种说法，如“中国科大”“中科大”“中国科技大学”“中国科学技术大学”就是相同的意思。遇有此种情况，可以设置同义词重定向，将其他词都定向到最标准的词条，如“中国科学技术大学”。

设置同义词重定向需通过版主审核，一个词含有多种含义的（如“科大”）应在词条中设置多个义项分别链接，不予设置重定向。

如果发现已有的同义词重定向不合理，可以点击词条底部的“举报”说明理由，由版主酌情处理。

## 9.8 历史版本

在当前词条页点击一个历史版本，进入相应的历史版本词条页，但在顶部醒目注明是历史版本。由恢复到历史版本功能按钮。

在当前词条页点击历史版本标题或“比较历史版本”链接，进入历史版本列表页（含当前版本），用户可以选中一个新版本，一个旧版本点击“比较历史版本”，左右两栏分别显示旧版本、新版本，增删的字句高亮显示。

## 9.9 编辑词条

用户只有登陆后才能进行编辑操作，但允许匿名评论。为防止恶意编辑，每个用户15秒内只能编辑一次。

词条编辑器采用网站通用的发帖器，外加插入词条名片等功能。在编辑过程中，适当的人性化提示尽量引导用户编辑结构清晰的词条。

编辑器为用户提供撤销，恢复功能，以方便用户。

## 9.10 我的百科

用户在百科中的个人功能包括：草稿箱、我的百科词条、我的百科评论、我关注的词条、我保护的词条、意见箱。

·草稿箱，发表词条时有两个按钮“发表词条”“存草稿”，方便用户进行大型编辑。为减少编辑冲突，每个词条只能存放一个版本的草稿，因此对草稿箱内的一个词条进行编辑时，其实是在对草稿箱内的版本进行编辑。

为了防止用户在编辑的过程中浏览器崩溃，电脑死机，系统会每隔1分钟将用户的编辑自动保存入草稿箱，注明“自动保存”。

·我的百科词条，包括已通过、审核中、未通过、被删除。审核中和未通过针对需要审核的条目。词条名称旁边由编辑链接。审核中的词条点击编辑后自动撤销原版本的审核请求，待提交新版本后再提交审核，以免出现编辑冲突。

·我的百科评论，包括已回复的提问、已回复的评论、已解决的提问、暂无回复。在已回复的提问中，将提示用户设置最佳答案。

·我关注的词条，指在词条页面点击了“加关注”的词条，会显示词条被编辑、有回复等最近更新。

·我保护的词条，申请保护成功后拥有保护权限的词条，在此可以与版主一同审核词条的改动。

·意见箱，相当于词条底部的举报，不过此处要输入词条名称。提交的意见会列表于此，并显示处理结果。

## 9.11 后台管理

格物百科后台管理平台供版主以上管理人员使用，包括审核词条、锁定词条、保护词条、搜索词条、删除词条、禁止用户、受理意见箱。

·审核词条，审核受到保护词条的修改、重定向申请、保护词条申请。

审核词条修改时，提供版本对比功能，左右两栏显示当前版本和待审核版本，当前版本被删除的字句、待审核版本新增的字句高亮显示，便于审核员评价此次修改的质量。

审核若不通过，应说明理由。

·锁定词条，将一个词条封闭编辑入口，禁止编辑。

锁定词条应当慎重，如无特殊需要，应使用“保护词条”功能。

·保护词条，设定词条的某些编辑需要经过审核。（见7.5节）

·搜索词条，按照创建时间、修改时间（默认）等排序依据，按照创建人ID、修改人ID、分类（category）、标签（tag）等筛选依据显示特定范围的词条列表，供管理人员审核用。

·删除词条，对违反格物致知网管理规定的词条及历史版本进行删除，使得外部不可见，但数据库中留存记录备查。

·禁止用户，禁止一个用户的创建词条、编辑词条、评论等操作。

·受理意见箱，对用户提交的意见进行回复、处理。

对于分区管理员以上级别管理人员，还有任免版主功能。版主是一个确定分类的管理人员，可以不止一个，负责所管理分类的后台管理。

## 9.12 百科活动

格物百科举行一些定期活动和不定期活动，旨在鼓励用户参与到百科的编辑中，并得到自我实现的乐趣。

# 第十章 信息发布模板

## 10.1 总体设计

很多组织机构希望拥有自己的信息发布平台，但苦于没有技术、没有空间。格物致知网为大家提供了这样一个便捷的平台，可以快速建立属于自己的组织主页，既有相对独立的管理权限和数据程序，又与格物致知网用户数据库对接，可以方便地向用户发送公告通知等，大大简化了组织内部联系。

组织主页默认采用CMS（信息发布系统）模板，当然也可以采用博客、论坛模板。高级一些，可以修改template模板文件，结合使用手册，模板语法可以很快掌握。对于专业程序员，甚至可以通过API自行编写程序，开发自己的网页程序。充分保证了组织主页的易用性和可用性，为菜鸟和专业人员都提供了发挥才能的平台。

信息发布系统采用树形结构，由分类目录和文章评论构成，每个分类目录中既可以有子目录，又可以有文章。文章中可以附带评论。

由于各大门户网站已是信息发布系统的极好范例，此处不再赘述。

## 10.2 分级管理体系

格物组织主页的特色在于支持分级控制与管理。例如学生社团要受到校学生会、校团委的管理，那么校学生会、校团委的组织主页管理团队理应对学生社团的组织主页有管理权限。因此，不同机构组织主页之间形成了一个层次结构（而非树形结构），上层组织有权对权力范围内的下层组织进行控制，但一个下层组织可以同时被若干上层组织控制，这就是树形结构与层次结构的不同。

每个组织机构主页相对独立，与百科类似，管理委员会由管理员和分区编辑组成，管理员可有若干人，由一名总管理员牵头，并给其余管理员分配“势力范围”（默认是全部权限）；分区编辑就是组织机构相应分类的编辑。

对规模较小的组织主页，建议不设分区编辑，以免引入过多不必要的限制；有一定规模后，对每个分类设置相应的编辑，使得分工明确。

## 10.3 可视化编辑

## 10.4 访问权限

与传统信息发布网站不同，格物致知网是以用户为中心的，可以针对不同的用户群体设置不同的访问权限。

一篇文章发表时默认是对全网公开的，但可以设置访问范围为全校、组织成员、自定义用户组等，可以同时选择多个有交集的范围，方便针对性信息通告。这样，不同用户看到的网站内容可以有所不同。凡是订阅了组织主页（不需审核）的用户都是订阅者，能够收到有权限查看的主页更新提示，但成为组织成员需要申请并由组织主页管理团队批准。

自定义用户组是组织成员的进一步细分，用户组为树形结构，如“软件与网络工程部”组可以下辖“网络工程部”组，一个用户可以同时属于多个用户组。

巧妙利用访问权限，可以方便地实现对组织成员的信息通知。例如，格物致知社的组织主页希望对网络工程部的所有成员发送一则通知，则可以设定权限范围为“网络工程部”用户组。

组织主页的信息发布与网站短消息群发的区别在于，发布的信息虽然推送到用户的个人空间上了，但信息本身还是存放在组织主页上，用户还可以充分利用评论功能实现沟通交流。另外，新闻发布平台的编辑器功能强大，方便发布一些结构复杂的公告通知。更重要的是，信息发布平台拥有统一的用户组机制，可以方便地维护组织内部的人事关系，不需要每个信息发布者维护、更新用户组，大大简化了信息发布者的操作。因此，建议使用组织主页而不是站内短消息发布组织内部新闻。

# 第十一章 论坛模板

## 11.1 总体设计

格物致知网的论坛对传统的论坛理念大举革新，不再是一个自上而下的大型论坛结构，而是将论坛的开办权“下放”给用户，由用户来管理、维护自己感兴趣的论坛。

论坛模块由学术类、生活类等多个大分类构成，每个相对独立的论坛都属于其中一个分类。开办论坛，需经网站管理委员会审核通过。

每个论坛相对独立，与百科、组织主页类似，管理委员会由管理员和版主组成，管理员可有若干人，由一名总管理员牵头，并给其余管理员分配“势力范围”（默认是全部权限）；版主就是传统意义上每个讨论版的管理人。

综合性论坛应拆分成多个小论坛，一个论坛应集中一个主题，如“数学”或“经济学”，但既包含数学又包含物理的论坛就是不合适的。

## 11.2 讨论版与版内分类

每个论坛由若干分区组成，每个分区可设置若干讨论版。为避免论坛内容太多太杂，讨论版个数不宜超过30个。所有讨论版在首页可以横排或竖排显示，可以调整显示的元素，如名称、图片、主题数、帖子数、最后发表、版主等。

讨论版可以对查看帖子列表、查看帖子内容、发表帖子、回复帖子分别设置访问权限。由此可以实现不少特殊效果，例如“举报申诉”类不希望非管理人员看到的讨论版，就可以允许发表，但不允许查看。

权限设置提供“静默隐藏”和“提示禁止”两种模式以供选择，对于提示禁止模式，没有权限的用户会看到“此贴需要XXX权限才能查看”；在静默隐藏模式中，相应内容对没有权限的用户是彻底不可见的。每个版块可以设置默认的权限和隐藏模式，而发帖时也可以自行设置权限和隐藏模式。

帖子列表可以按照发表时间、回复时间排序，按照作者、帖子类型筛选。

对一些内容需要进一步细分的讨论版，提供了“版内分类”功能，若干分类以标签的形式列在帖子列表页顶部。

每个帖子可以设置一个或多个标签，在发表时除从标签列表中选择外，还可以填写自定义标签，避免了帖子分类困难的问题。除非分类困难，建议每个帖子只设置一个标签，尽量从现有的标签列表中选取。帖子列表页的主题名称前显示帖子标签。

管理团队要定期合并同义标签（如：“线性代数”和“线代”），将其加入同义词系统，今后发表的标签会自动正式化。（如“线代”自动转为“线性代数”）

点击标签，可以显示讨论版内所有以此为标签的帖子，方便了同类内容查找。

## 11.3 边栏

论坛**帖子列表页**设置边栏，默认情况下自上而下为：

·网站小字报（全站）

·讨论版活动（版主举办）

·精华帖子（最新若干条精华）

·活跃会员（一定时间段内回复帖子最多的会员）

·最热帖子（回复数最多或访问量最大）

·今日热点（全站）

如果内容为空，则隐藏相应版块。

论坛主页与帖子内容页由于版面紧张，暂不设置边栏。

## 11.4 帖子类型

论坛帖子内置有以下几种类型：

·普通

·资料，主体是一个资料文件，外加一些可选的说明

·悬赏，回答问题被选为最佳答案，奖励悬赏的积分

·投票

·辩论

·活动，大家一起来参加线上/线下活动，版主和用户均可举办。

“资料”类型帖子的发表与其他帖子不同，“资料”类型帖子的发表采取向导模式，当用户点击下载按钮后，就会出现一个向导，第一步先让用户选择文件，点击下一步进入详细信息界面，此时文件已经开始上传，并通过进度条实时向用户反映进度，在本界面中用户需要填写一个表格，表格第一栏为资源类型，系统会自动根据文件类型为用户选择好，详细信息如下：

若文件格式为doc、pdf、djvu，则默认为书籍类，出现标题（自动填写），作者，译者等信息表格；

若文件格式为rar、zip、tar、gz、exe，则默认为软件类，出现软件授权类型、软件语言、操作系统、开发公司等信息表格。

除“资料”类型本身就是附件外，各种帖子类型均可上传附件。

上传的图片自动在帖子中显示，上传的Word、Excel、PPT自动转化成flash在线显示，上传的PDF文档自动将前10页以Flash的方式预览显示。方便用户不下载附件即可快速了解文件内容。

## 11.5 帖子内容页

帖子内容页的基本格局由楼主帖和回帖组成，暂时设置每页显示20帖。

楼主帖顶部设快速分享按钮，底部设帖子操作按钮、帖子导航：

收藏、分享、支持、反对按钮：

·收藏将在个人空间“我的论坛”版块加入一条“我收藏了……”，并设“打开收藏夹”链接，显示所有自己收藏的帖子。

·分享将发表一个好友新鲜事“我分享了……”

·支持、反对即为“顶”“踩”，不管是否登录均可操作，按钮下显示当前支持、反对的人数。登录用户对同一个帖子只能“顶”“踩”一次。登录的用户点击“顶”，帖子在列表的位置在当天有回复的帖子中提高1，若为顶部不再升高；“踩”，帖子在当天有回复的帖子中降低1，若为底部不再降低。

帖子导航：

分为两列，左列为五条相关帖子，根据帖子标题的相关度加以判断，引导用户查看论坛其他帖子；右列为五条楼主热帖，显示楼主最热的五个帖子。楼主热帖对发帖人本身也是一种宣传作用。

帖子模仿Gmail的显示方式，一个框就是一个帖子，由用户信息栏、内容区、操作栏构成。

每个帖子右上角是楼层数，底部操作栏包括点评、回复链接和发表时间。点评是在帖子内部快捷回复的简短评价。

帖子左侧是用户信息栏，包括传统的用户名、用户头像和积分，下面是加为好友、与TA聊天链接。点击用户名进入其个人博客，在头像上停留显示其用户签名和TA的最新3条帖子。

## 11.6 版主与帖子操作

版主拥有所管理版块的操作管理权限，可对版内帖子进行

（一）鼓励：提升、置顶、高亮、精华、图章、图标、推送

（二）抑制：删除、下沉、关闭、警告、屏蔽

（三）编辑：编辑、移动、分类、复制、合并

如果管理团队要对帖子进行补充、澄清或说明，有特殊权限在帖子内部加入“版主批注”。

楼主可以对帖子进行“关闭”或“解除关闭”，但被管理团队关闭的楼主无权解除关闭。

通过“提升”“下沉”，管理团队可以将帖子移至当前帖子列表的任意位置。

# 第十二章 导学模板

## 12.1 总体设计

目前网络上的在线学习系统几乎都是收费教学，且价格极不合理，而信息社会的发展又迫切需要网络成为学习的重要工具。怀揣着互联网改变教育的愿景与梦想，格物致知网创立了在线导学系统，借用wiki开放维护的理念，团结一切可以团结的力量，动员广大网友在为导学系统贡献自己力量的同时，也从与高手的切磋、编写文档的思考中收获知识与乐趣。

利用互联网优势，最容易开展并取得广泛市场的就是C语言在线评测系统。C语言在线评测系统的灵感来自信息学竞赛中常用的ACM Online Judge，但又与ACM不同。ACM系统侧重的是算法与数据结构，选手的目的是参加编程竞赛，题目一般具有相当难度，并不适合C语言初学者。

我们的主要用户群是C语言初学者。目前广大C语言初学者编写程序后难以确定程序的正误，于是很多同学就简单的输入一组普通数据，结果正确就行了。这样的“评测”不能保证程序的正确性，更违背了计算机语言逻辑思维的严密性，从长远来看，对计算机学习具有负面影响。

我们的在线评测系统则是用若干组（一般是10组）涵盖多种输入可能情况的测试数据来评测用户提交的程序，全部正确则视为Accepted（通过），未全部通过则提示Wrong Answer，并允许用户查看正确与错误的输入输出，从对比发现程序的问题，修改后再次提交评测，以此类推。

## 12.2 关卡模式

故事模式 ，让用户过关，一关比一关难，各个关卡之间用故事穿插。为了给枯燥的学习增加一些乐趣，特地设立关卡模式。关卡模式以故事作为线索始终。有点类似一些RPG游戏，故事的情节会要求用户编写一些相应的程序，并对程序作出各种苛刻的要求，用户编写的程序只有符合相应要求才能继续故事剧情。有的关卡只有一次机会，用户失败了只能从头再来，更增加了游戏的趣味性与刺激性。在剧情的进展中，有时候后出现分支剧情，用户选择不同的分支会对故事今后的发展产生巨大的影响。用户只能选择一个分支继续剧情，通关后，用户可以选择任意关卡进行游戏，刷榜。

## 12.3 题库

C语言在线测评系统题库由基础篇和提高篇组成，基础篇难度不超过大学传统C语言课程的程序设计题目，面向C语言初学者，是测评系统的主要用户群和努力方向；提高篇类似ACM Online Judge，面向参加编程竞赛的选手。

基础篇题库来源于C语言程序设计课本和练习册，题目按照书本的章节组织，便于用户随课程学习进度练习。同时每道题目有一个分类（其实类似书本的章节），再加上一些开发团队设计的题目和热心用户提交的题目，便于用户对某方面知识点进行巩固练习。

## 12.4 题目

每个题目（Problem）的主要部分包括：

·题目ID，在评测系统中唯一的快速索引编号

·题目名称

·作者介绍，若题目为作者原创，允许作者在此对自己做简明的介绍，以示对作者的尊敬

·题目描述，需要解决的具体问题

·输入格式，输入的具体格式

·输出格式，输出的具体格式，程序输出必须严格遵循输出格式才能通过

·样例输入，一组输入样例，便于理解输入格式

·样例输出，一组输出样例，便于理解输出格式

·来源，如某本书某章节，某个用户自编，某项竞赛

·测试数据，一般为10组，包括正确的输入输出

·附加限制，包括运行时限、内存限制，均有默认值。

·智能评分程序（可选）

·特殊标记，如关卡模式中的第N关，必须在通过前N-1关后才能进入

题目的辅助部分包括：

·提示，命题人编写的简要解答提示，解出该题的也可以发表提示

·题解，命题人编写的题目详细解答，解出该题的也可以发表题解

·讨论，用户可以在此对题目进行讨论，采用比论坛简洁的缩略样式

·统计，题目不同通过状态和通过率的统计

·排名，按照运行时间、内存、代码长度的顺序对所有通过的程序排序，激发用户优化程序、争取排名的热情。

辅助部分分别用链接的形式设置在问题页面下方。

在所有页面，设一个“我要纠错”按钮，用于对题目或解答进行纠错，信息发送至发表人（命题人或题解作者）和管理团队。

用户可以提交自己的题目，通过管理团队审核后予以上线。提交题目必须包括除题目ID外的所有主要部分，尽量包含一份有正确源代码的题解。

## 12.5 评测

用户提交代码后，先进行安全检查，安全检查不通过的，提示返回修改。

检查通过后由计算机自动编译执行，为防止恶意代码，GCC库已做过必要的修改。

执行时间超过限制或使用内存超过限制的，立即停止程序运行。程序中途发生错误停止运行的，记录错误信息。

结果正常输出的，与标准答案进行逐字节比对，完全正确的通过；不正确的去掉所有空格和换行与标准答案逐字节比对，正确的为“表达错误”；存在测试点结果不正确的不通过，跳转到错误分析页面。

对于需要特殊的误差处理或对输出方案进行某种计算后给分的，系统设置有智能评分程序的接口，可以在提交的程序运行结束后调用评分程序而不是逐字节比对系统进行评测。

错误分析页面包含出错的测试点的输入数据、用户程序的输出和标准输出数据，供用户参考修改。

一次提交的状态包括：

·Accepted，全部测试点通过

·Presentation Error，去掉空格和换行后全部正确，但逐字节比对不正确

·Wrong Answer (Score)，存在测试点不正确，括号内注明得分

·Time Limit Exceeded，超过时间限制

·Memory Limit Exceeded，超过内存限制

·Runtime Error，运行时程序错误

·Compile Error，编译未通过

·System Error，系统错误，遇有此种情况请联系管理员。

提交时有超过10个程序在排队等待运行时，自动跳转到“状态”页面，查看所有提交的状态，并实时更新以查看当前状态。

## 12.6 其他题型

C语言程序设计中还有选择、填空、读程序写结果、程序填空等题型，且多数是可以通过计算机验证的，因此其他题型的题库也要逐渐构建，形成一套完整的C语言程序设计辅助平台。

阅读程序题、有关运算符优先级的题、涉及复杂声明的题，可以编写后台程序，按照一定的规则产生题目，例如常数、变量名等都是很容易由计算机指定的。

## 12.7 模拟题

模拟期末考试，待其他题型准备完全后可以在线限制解答一套C语言阶段测试题或期末测试题，并立即评分，针对各部分得分自动给出学习提示。

## 12.8 比赛

喜爱编程的同学们都喜欢参加编程比赛一显身手，因此我们组织三个级别层次的比赛：

·初学组，针对C语言课程学习者的比赛

·普及组，信息学竞赛的初级水平

·提高组，信息学竞赛的高级水平

用户在准备好、提交好试题和测试数据后，可以申请举办在线比赛，管理团队审核通过后与申办用户协商比赛时间等具体事宜，后在网站上公开通告。

给定比赛时间到，试题自动开放上线，参赛选手需在规定时间内提交程序，但比赛中程序不予评测，比赛结束后统一评测，程序多次提交的，以最后一次提交的程序为评分标准。

评测结束，按照总得分、总运行时间、总占用内存、总代码长度的顺序对参赛选手给出排名，并对前若干名给予适当奖励。

## 12.9 今后研发方向

·数据结构、算法课程的分章节习题

·PHP程序开发大赛，直接采用网站核心API

·数据库课程在线评测系统，练习数据库存取

·汇编语言在线评测系统，为保证安全采用模拟执行指令（虚拟机）的方式

·安全传输与加密技术大赛，看谁的最先被攻破

# 第十三章 站内搜索引擎

## 13.1 总体设计

在全站页面顶端，统一设置搜索框并设两个按钮“搜索××”“搜索全站”，其中××指当前所处区域，如论坛、百科等。

结果页面显示方面，模仿百度就可以了。查询语法采用国际通行语法，可以采用AND、OR、NOT等简单逻辑。

面对海量的数据，每次都搜索一遍全部数据库显然是不现实的，必须建立快速索引。

站内搜索引擎由分词器、索引器、查询器构成：

·分词器，将连续的中文自动分解成具有语言意义的词。

·索引器，将网站内发生改变的页面按照一定算法计算其对每个关键词的权值，并记录到索引中以便快速搜索。

·查询器，根据用户的查询找到索引中最具相关性的条目显示出来。

每个查询分为四个步骤：

·分词，将查询句子分为若干有语义的关键词；

·查找索引，取出每个关键词排名靠前的索引内容

·合并索引，将不同关键词的索引内容进行归并排序

·显示，将搜索到的条目按照一定格式（如高亮关键字）生成HTML

## 13.2 中文分词

采用最长逆向匹配算法。不再详述。

## 13.3 排名算法

排名算法是搜索引擎的核心技术。

页面相对于某个关键词的权值 =（页面权值 × 新鲜度 + 用户得分）×关键词权值

关键词权值

页面权值

用户得分

关键词权值：太大众化的关键词意义不大，因此权值要低一些；越是专业化的关键词越受到重视。在多个关键词的联合搜索中有效。

PageRank：暂时用常数的方法指定，如百科词条的PageRank大于论坛帖子的PageRank。

页面权值中的常数c：过短的内容一般

新鲜度：由上次更新距当前时间而定，以鼓励新内容。今天有更新的为1，1天历史的为0.95，2天历史的为0.9，……，1年以上的为0.5，采用近似log的函数，控制影响因子不能太大。

用户得分：反映了用户在搜索结果中点击的偏好，是一个小的修正项。对新加入的内容昨天用户得分可设为平均值（而不是0）以平衡新内容未被点击过的弱势。

页面区域权值：针对不同的区域设置不同的重要度，如标题、标签、分类中关键字的权值大于正文的权值，正文中开头的权值大于中间部分的权值。标签>标题>摘要>正文>评论。

字体权值：正文中相对较大字体的权值大于普通字体的权值。加粗、一二级标题、特殊HTML标记等也按照特殊权值计算。

区域加权长度：区域每个字与字体权值之积的和。因为字体是相对而言的，一个通篇都是大字体的文章并不比通篇小字体的文章重要。

采用不同区域加权的方式是为了避免一个区域影响权值过大，如“标题党”。

## 13.4 实现

针对每个关键词建立一个索引表，包括页面地址、最后更新时间、页面权值、用户得分、总权值，用MySQL对总权值建立索引，以便高效排序。

发表或更新内容时，对发生改变区域的所有关键词进行重新评分，新评分 = 旧评分 – 原有内容的评分 + 现有内容的评分。之所以不直接计算整个页面的评分是为了提高效率，例如添加一条评论就没有必要把整个文章重新评分。

同时更新最后更新时间，以新鲜度为1重新计算总权值。

每日凌晨2:00，根据24小时内搜索引擎点击情况，计算所有页面和关键字的用户得分；根据最后更新时间判断新鲜度，计算新的页面总权值。

查询时，采用分词算法将查询语句分成若干关键词，分别取出索引表中的前10条，再按各关键词之和归并排序，取其中前10条作为最终结果。当然，如果要查询第10~20条，则取出各索引表中的前20条归并排序，取其中11~20条作为最终结果，以此类推。

由于多数用户查询只是针对1或2个关键词，因此句子各处相关性问题并不突出。语义匹配问题有待研究新算法。