

开放源代码软件的应用研究



江 何 周华春 (北方交通大学电信学院网络管理研究中心 100044)

摘要: 目前, 开放源代码的软件开发模式已经越来越多的被采用, 其基本思想是通过公开软件的源代码使得不同的开发人员可以互相交流, 发现错误从而提升软件质量。在某些环境下运行的软件由于对可靠性的高度要求, 采用开放源代码的模式进行软件开发将能最大化的做到这一点。本文试从商业模型、法律规范、实现技术这三个角度论述了开放源代码软件的应用。

关键词: 开放源代码 商业模型 法律规范 实现技术



1 前言

近年来, 开放源代码软件已经逐渐不再是新鲜名词, Linux 和 Apache 所取得的巨大成功使得开放源代码软件受到人们的注意。事实上, 著名的大企业如 IBM、AT&T、Alcatel 已经在不同的级别上采用了开放源代码的软件开发模式, 并且融入其商业理念中。微软也承认开放源代码软件在操作系统领域对其市场领导地位构成了威胁, 而且还对此进行了秘密研究。开放源代码软件最重要的优点就是可靠性, 这听起来似乎不合理, 然而, 正是因为开放源代码独立于各厂商, 软件产品也就没有众多的附加条款, 这些条款其实含有极大的不安全因素, 而且软件开发人员和使用者可以在合法的情况下自由测试、修改, 不会象往常那样感觉软件是个黑盒子, 这可以避免原始开发人员在软件中开后门。对于系统与网络软件来说, 可靠性与安全性尤为重要, 因此, 引入开放源代码的软件开发模式具有极大潜力, 下面主要从商业模型、法律规范、实现技术三个角度说明如何应用, 首先介绍开放源代码的具体含义。



2 什么是开放源代码软件

开放源代码软件首先是用户要能看到源代码, 然而这是不够的, 开放还要体现出软件用户的某些自由性, 基本的如下:

- 通过学习源代码知道软件的工作流程并且能自由地应用到自己的软件开发中;
- 能够自由地重新分发软件拷贝;
- 能够自由地修改程序提升质量, 并且将自己所作的优化公诸于众使大家受益;
- 能够以各种用途如私人的或商业的去运行软件。

互联网的使用使开放源代码软件得以广泛传播, 就像在一个自由集市一样, 大家通过邮件列表可以自由交流, 但是软件被自由的修改分发并不意味着开放源代码的软件开发处于无政府状态, 实际上大部分的开放源代码项目都是组织良好, 管理严格的。例如流行的 GPL (GNU Public License) 就明确规定软件的原始开发者拥有软件产权, 他/她也被称为“善意的决策者”, 只有产权所有者才能对软件进行重新分发, 其他人员对软件错误的修正必须通过产权所有者进行, 当然, 产权所有者可以授权他人重新分发而且对于善意的修改是应该会接收的。Linux 系统就是一个典型例子, 其最初的开发人 Linus Torvalds 拥有知识产权, 他决定了什么样的修正能被加入到新的发布包中, 开放源代码并不意味着丧失对软件的控制权, 它意味着能保持软件向更好的方向发展, 吸引更多的人对其进行完善。

一般说来, 开放源代码具有如下一些优点: 因为可以学习并修改源代码, 软件人员就

可能验证原始软件某些算法的正确性甚至应用更好的算法优化软件, 提高性能, 大家在基于互联网上的交流也使得开放源代码的软件能够得到更快的发展, 有经验的专家可以自由的加入进来提出更好的意见, 最终使软件具有高度的可靠性与稳定性。对于软件用户来说, 他们不必再担心从软件开发商那里买回来一个什么也不懂的黑盒子, 而且从长远来看, 软件产品受到广泛关注不断被升级更新也降低了开发与维护费用, 在都遵守开放源代码规则的大框架下, 软件产品也比较容易兼容。

开放源代码软件的某些缺点也是需要考虑的。比如并不能保证所有的开放源代码软件都能吸引大家去完善它, 这样软件质量便无法得到提升。开放源代码软件的服务与技术支持体系也不是很完善, 目前为止还是大多利用邮件列表, 多数软件用户会很习惯甚至不相信它, 这种情况正随着越来越多的软件咨询服务站点的建立而得到改善, 另外, 如果不能得到有效的承认与合理的报酬, 开放源代码软件开发者的热情也可能会收到打击, 失去了众多的参与者, 开放源代码软件的种种优势都是无法想象的。尽管如此, 近年来开放源代码软件所取得的成功让人有理由相信开放源代码利大于弊, 至少在某些领域是可以完全取代传统的软件开发模式的, 例如需要保证计算机在程序运行时不会因开发人员留下的安全漏洞而泄密的场合。



3 从商业模型的角度探讨开放源代码模式的应用

理解了开放源代码软件的概念后会发现这的确是提高软件质量的一种良好模式, 但是对于软件开发商来说, 他关心的是利润; 对于软件用户而言, 比如在通信运营环境下, 如果得到的仅仅是寄希望于再修改再完善的产品, 那么是无法让人放心的, 下面介绍几种实践过或

尚待实践检验的开放源代码软件开发的商业运作模式,从这些模式可以看出开放源代码与获取商业利润并不矛盾。

(1) 服务支持提供商模型: 在这种模型里, 软件开发商免费提供软件, 但是收取服务支持费用, 通俗的说就是“卖服务”。著名的RedHat(红帽子)公司就是一例, 从小型嵌入式设备到实用的高端Web集群式服务, RedHat都通过采用GPL的方式开放了软件源代码, 其正式Linux版本可以通过各种途径获取, 如直接从公司网站上下载, 从合作商那里获取等, 每个发布包都含有特定的服务支持包, 据此RedHat提供软件更新支持(如30到180天内), 安装支持(如30到90天内)并且收取相应的费用, RedHat公司的收费服务还包括诸如软件移植, 优化以及开发特定功能等。

(2) 成本缩减模型: 此时, 使用开放源代码软件的目标是降低软件的相关成本。电信运营商常有这样的经验, 购买了某厂商的产品后为了发挥其应有性能就陷入无尽的升级换代中, 软件越来越贵, 效率越来越低, 最终耗费极大。开放源代码软件则是另一种情况, 软件不再是黑盒子, 提供商需要尽力提高软件质量才能在竞争中取胜, 这导致使用者的

相关费用降低, 当硬件自然换代时, 软件用户可以自己修改软件以适应硬件的变化而不用再买新的发布包。

(3) 部件混和模型: 这种模型适用于硬件厂商, 它实际上包含了上面两种模型, 因为现在软件对于硬件来说不可缺少, 从长远考虑, 硬件厂商采用成本缩减模型, 使用开放源代码软件, 同时, 软件开发商作为服务支持提供商保证了软件的高可靠性与稳定性, 这种模型相对较新, 目前还没有十分成功的样例, 不过开放源代码软件联盟之一的OSI(Open Source Initiative)相信这种模型将取得巨大成功, 因为有许多硬件厂商已经逐渐转向使用开放源代码软件了。

(4) 私有软件转换模型: 这种模型是由AT&T公司提出的, 由于私有软件的某些模块落后低效或者其他什么原因, 软件提供商可以将部分模块转换成开放源代码的软件模块, 这样做有两个好处: 一是可以分享开放源代码软件种种好处; 二是有可能使公开的部分成为行业标准, 还有稍微自私的考虑是希望让用户在长远的使用中感受开放源代码软件的一些劣势如服务有限从而最终转回购买私有软件。

(5) 技术进步模型: 与私有软件转换模型

类似, 但是在这个模型里软件商将相关领域的所有软件都公开成为开放源代码软件而不只是那些旧的或者有局限性的部分, 其目的是获取技术进步, 而厂商也寄希望于在该技术进步带动的相关领域销售软硬件赚钱。目前流行的Java技术实际上就采用了这种模型, Sun将Java的相关技术公开也带动了其“网络就是计算机”理念的传播, 为其构想的嵌入Java智能卡的数字世界打下基础。

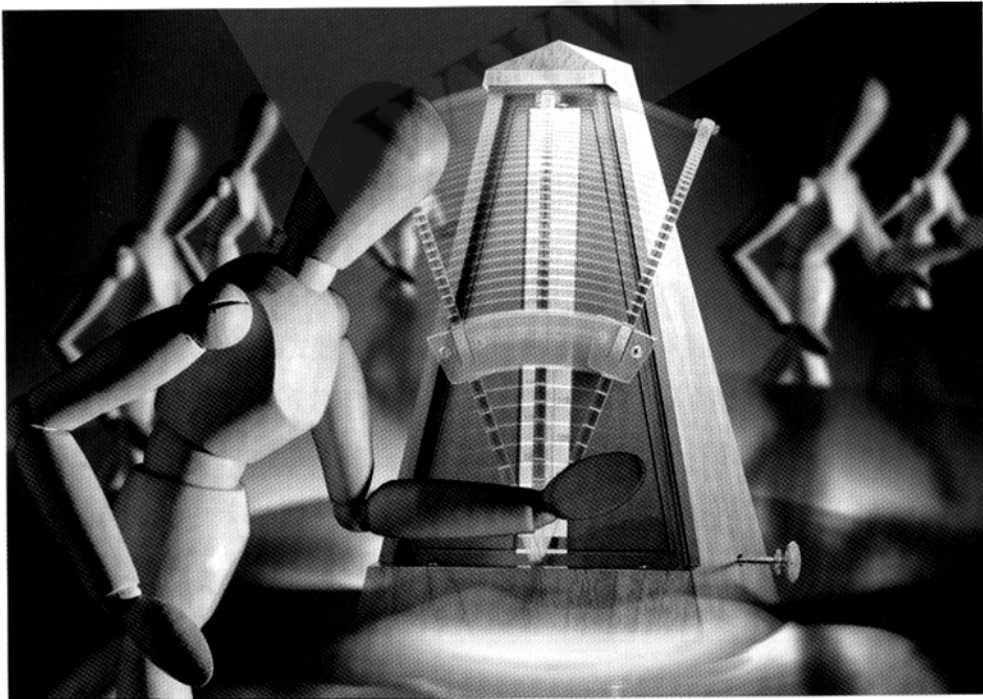
(6) 开发代理模型: 许多公司了解开放源代码的种种好处并希望利用开放源代码软件, 但是又不想自己去网罗自由软件的开发者, 而且由于公司的目标定位也不是软件开发商, 因此高价聘用优秀开发人员不合算。在这种情况下, 象SourceForge(<http://sourceforge.net>)和CollabNet(<http://www.collab.net>)这样的开放源代码软件网络中介就出现了, 在这里, 公司把研发项目计划放到网上, 自由软件开发人员寻找自己感兴趣的项目, 他们通过中介互相联系, 由于开发的软件是开放源代码的, 这样就充分利用了开放源代码的优势, 同时, 网络中介作为一个整体的代表与公司打交道, 可以严格的要求规范约束开发人员的行为, 使之看起来又像传统的软件开发商一样, 这种模型有点类似于保安公司, 显然, 客户会比较信任保安公司, 出了问题也比较容易找到责任人。事实上, CollabNet所取得的巨大成绩已经使许多大公司消除了对开放源代码软件的疑虑, 从某种意义上说, 这是目前最成功的开放源代码软件商业模型。

还有许多有用的商业模型, 例如开发者许可证模型、技术倡导模型等, 后者在Apache项目中得到成功应用, 限于篇幅不再一一介绍。



4 从法律规范的角度考虑开放源代码模式的应用

从法律规范的角度来考虑开放源代码的应



用主要是指在发布与管理维护开放源代码软件时需要遵循的一些规则,包括许可证与专利制度。开放源代码软件的发布者只有遵循相应的规则,得到开放源代码软件联盟比如OSI或FSF(Free Software Foundation)认证后,才可以以开放源代码的形式发布自己的软件并享受相应的权利。通常需要遵循的规则有:

(1) 软件可以被自由分发而不必向原开发者付费;

(2) 如果软件的源代码没有随软件一起发布的话,它应该能以合理的花销从给定地方获取;

(3) 软件的衍生产品应该遵循相同的许可证要求;

(4) 软件的发布者本身应该尽量保证软件以完整的方式发布,除非要发布必要的补丁,否则某些恶意的修改再发布会损害原作者的声誉;

(5) 软件许可证不应该有歧视行为;

(6) 软件许可证不应该对其他软件构成妨碍。

除此之外,开放源代码软件的两大组织OSI和FSF对开放源代码的含义有各自的理解。FSF认为所有开放源代码软件的衍生软件都应该遵循与原始软件相同的许可证,这实际上就不可能在它的基础上发布私有软件了,它的许可证采用版权保留(CopyLeft)模式,前面介绍过的GPL就是其一,还有一种LGPL(Less GPL)允许与非自由软件模块的集成。OSI的要求稍微宽松一点,它允许在开放源代码软件的基础上发布私有软件,但是必须包含原有的许可证信息,它的许可证是无限制模式的,例如用户可以只重新发布修改的部分代码模块而不必附上原始条款。主要有BSD和MIT两种许可证。

从实际角度看,两者都保证了用户有权修改源代码,因此没有本质上的区别。

关于专利问题目前还有争议。因为专利保护的是某种技术、思想,任何他人在未得到授

权的情况下即使采用同样的技术进行复制也是违法的,软件界广泛应用的版权则只是具体的保护某件作品,它规定了如何使用这件作品而不涉及其他的问题。显然,任何软件设计都包含大量的设计思想,如果也施行专利的话,那么可以想见其巨大的数量,这样他人将无法有效借鉴先进的设计思想与方法。



5 从实现技术的角度探讨开放源代码模式的应用

开放源代码软件目前在许多新技术领域都有涉足,特别是与通信网络相关的如Web/XML服务器,智能移动终端以及多级代理系统等。对于希望应用开放源代码模式的企业来说应该在如下几个方面深入:

(1) 结合实际情况尽力推进前述几种商业模式在本企业的应用;

(2) 对于经试用发现合适的模型应该启动相应的研发项目作进一步研究,因为有些商业模式是还没有经过实践检验的;

(3) 可能的话在本企业推行开放源代码模式的相关服务。

另外开放源代码软件也同样可以利用当前热门的PKI(Public Key Infrastructure)研

究成果对自己进行安全包装以适应上述通信软件对安全可靠的高要求,这样私有软件声称代码不公开所具有的安全优势就不存在了,事实上,后者极有可能被开发人员留后门导致更不安全。

应用开放源代码模式还要注意文档的整理,同样需要按照软件工程的要求完成相应的文档,而且,好的技术支持文档是软件成功的重要基础。



6 总结

本文主要从商业模式、法律规范和实现技术三个方面论述了开放源代码的应用,总的看来,开放源代码的种种好处使得现有的软件开发模式已经逐渐发生了改变,前述着重论述的几个主要商业应用模型也表明开放源代码在遵循相应的法律规范情况下同样具有经济效益,因此它是有发展潜力的,国内的软件商如中科红旗软件技术有限公司在Linux基础上开发出红旗Linux就是良好的例子。在开放源代码的模式下开发软件也有助于争取自主知识产权,掌握核心技术,相信开放源代码软件会有较大发展空间。 ■

The application research of open source code software



参考文献

- 1 《A strategic study on the use of Open Source in a telecommunications operator's environment》,Aspects of Open Source in Telecommunications,Brendan de Bruijin, Veit Vogel.
- 2 《Open source —— An analysis from a telecommunication operator's perspective》,Aspects of Open Source in Telecommunications, Brendan de Bruijin, Lothar Grimm.
- 3 《The Open source definition》 <http://www.opensource.org/docs/definition.html>.
- 4 《The Cathedral and the Bazaar》 <http://www.linux.org.tw/CLDP/doc/Cathedral-Bazaar.html>.