

分布式开源社群治理机制研究综述

任 重

(杭州电子科技大学 经贸学院,杭州 310018)

摘 要:作为一种新兴的生产方式,开源软件的特点是志愿者为软件开发做出贡献而不追求任何个人的经济回报。这一现象引起了经济、法律、社会、心理以及计算机科学等领域的学者们的持续关注。但已有文献主要集中在分析参与者的动机上,因而忽视了对开源社群组织和制度结构的研究。事实上,开源软件奉行的是一种分布式的治理机制,即项目的创始人或者领导者让渡部分权责给他所信任的子系统负责人以建立核心层,在没有等级制度的情况下实现对整个社群的控制和协调。

关键词:开源软件;认知共同体;治理;分布式创新;共同对等生产方式

中图分类号:F270

文献标志码:A

文章编号:1672-8580(2012)05-0070-06

一、引 言

近年来,开源软件(F/OSS)已经渐次演绎成为了一个传奇。这不仅仅表现在吸引了越来越多的软件工作者和爱好者投身其中,也不仅仅表现在它甚至赢得了直接竞争对手如微软这样的商业软件巨头的承认和尊重,更重要的在于它对包括经济、法律、社会、心理和计算机科学在内的学术界传统分析范式的全面颠覆。例如,著名的布鲁克斯法则认为“对于未能按期开发出的软件项目,增添人手只能使其出炉得更慢”,即更多并不意味着更好,对一个项目而言,个人的才华比群体扩大所引致的规模效应更具决定作用。有趣的是,F/OSS的成功与否,恰恰主要取决于参与的人数^[1-3]。再如团队作业中众所周知的“短板”现象在F/OSS社区中也是不复存在的,F/OSS的分布式创新模式决定了其开发进度是由“长板”决定的:技术路线是非事前(Ex-ante)规制的,无相互关联关系的成员开展的是一种竞赛式的平行研究,谁率先取得突破,谁的模块最有可能被系统接纳。与此同时,F/OSS奉行一种“尽早发表,经常发

表”(Release early, Release often)的宗旨,可以最大程度地防止平行研究所带来的资源浪费。更为重要的是,F/OSS的源代码完全开放、任意下载和无限限制修改等理念直接挑战了西方古典经济学中的“理性人”假设,因此引来了学者们井喷式的密集研究。这些研究无非主要是竭力证明开源软件现象并没有违反经济学逻辑,例如Lerner和Tirole提出的所谓“信号效应”,意指F/OSS参与者虽然没有直接的金钱回报,但却可以通过贡献源代码赢得职业声誉,然后为获取雇主或风险资本的青睐铺平道路^[4]。当然如果信号效应成立,F/OSS无疑褪化为追逐私利前一项必要的投资而已,也就不足为奇了。事实上,一系列的实证研究表明,能真正通过信号效应受益的仅仅是少数人,大多数人参与F/OSS还是出于一种奉献的利他主义精神,但这一结果由于不能作为维护经济学理论权威的佐证,学者们往往会“选择性失明”。

然而,如果只把注意力落脚在F/OSS的“免费”和“开放”上,就很容易把视角局限于参与者的动机分析而逐渐走进死胡同。综合已有的研究不难发现,无论F/OSS

收稿日期:2012-07-13

基金项目:浙江省哲学社会科学规划项目(10CGGL13YB);浙江省高校人文社科重点研究基地资助项目(RWSKZD02-201006);浙江省教育厅科研项目(Z201119527)

作者简介:任 重(1972-),男,陕西商州人,副教授,博士后,硕士生导师,研究方向:国际贸易理论和政策、创新理论。

网络出版时间:2012-08-24 **网络出版地址:**<http://www.cnki.net/kcms/detail/51.1676.C.20120824.0917.001.html>

项目的内涵还是参与者的个人特性都是高度异质的,因此很难得到所谓“一统江湖”式的结论。实际上,要更好地理解 F/OSS,应不仅仅纠结于软件开发本身,而应把 F/OSS 看作一种新兴的组织革命,从而能透过源代码和许可证制度等表象去探寻其内在的分工和治理机制。

二、F/OSS 治理的基本法则

在一本奠基之作中, Raymond 以 Linux 为例,把开源软件称为“集市”风格的生产模式(Bazaar-style):开放式建设,开发者每一步的改进工作都是完全公开的;把用户当成协作者,随时准备倾听来自同行和用户的评论和建议;无中央决策机构预先锁定技术发展路径;开发者自我选择与自身能力和兴趣匹配的 F/OSS 项目;结构比功能更为重要,合作者和用户的人数对结构的稳健性起决定作用,而源代码的开放性又为吸引足够多的合作者和用户提供了一定的保证^[9]。可以看出, F/OSS 既不同于商业软件的运作模式,也不同于对合作创新的惯常理解。

Von Hippel 和 Von Krogh 指出, F/OSS 尽管具有经济学意义上的公共品的部分特征,但却非完全的公用品,即从表面上看,没有知识产权的保护, F/OSS 的开发者似乎很难避免被搭便车,但实际上由于 F/OSS 属于用户创新,其代码段是高度客服化的,这决定了社群外的搭便车者能够享受的只能是“截短”的收益, F/OSS 的开发者并不必然因被搭便车而遭受到很大损失^[6]。另一方面,缺乏剩余索取权和自上而下的严格监督机制, F/OSS 又如何防止社群内部成员偷懒,搭其他合作者的便车呢?根据 F/OSS 的基本法则,这一点似乎是不用担心的:由于必须做出实际贡献(发布源代码)才能获取所需的声誉,显然任何 F/OSS 开发者都没有投机取巧的激励和可能。但引发的问题是如何保证是来自同行的评价是真实可靠的呢?事实上,在 F/OSS 社群中,这一点同样不用担心:参与者的进入和退出是自我认知和选择的结果,位于顶层的项目主持者如果不能做到公正客观去评价社区成员的工作,将直接导致社群公信力的丧失,参与者可以选择“用脚投票”,转向其他的 F/OSS 社群。成员的大量流失无疑将造成社群的消亡,这是任何项目主持者都无法承受的严重后果,因此类似“人情社会”之类的文化传统在 F/OSS 中是没有立足之地的。

Benkler 提出了“共同对等生产方式”(Commons-based peer production)来描绘类似于 F/OSS 的分布式知识生产方式的基本法则^[7]。相对于等级制度(Hierarchy)

或者市场化的生产方式, F/OSS 的信息成本更低:个体和社群之间通过互联网这种低成本的通讯方式,自我搜寻、自我选择和自我匹配,既不需要复杂的薪酬制度充当价格信号,也不需要由中央集权机构规定做什么和谁来做,因此可以规避不完全契约和市场交易所衍生的各种组织成本和交易费用。但这种分布式或者共同对等的生产方式易于遭受到商业利益的威胁, F/OSS 问世之初,曾被视为对商业软件企业的巨大威胁,一开始这些企业利用各种知识产权保护政策竭力将 F/OSS 扼杀于摇篮之中,后来看到共同对等生产方式的发展趋势不可阻挡,他们由阻挠改为浑水摸鱼,试图私有化本已开放的源代码,从中谋利。如果让这些企业得偿所愿,志愿者分文不取的奉献就将沦为个体谋取私利的工具,奉献与回报的利益平衡机制将被破坏(对那些期待获取声誉的志愿者的打击尤其大),长此以往,也就无人愿意做这样的免费奉献了。特别值得注意的是,仅靠社群内部人员的自律或者相互之间的承诺是无法克服这种危险的,道理很简单,内部自律或者承诺是无法约束社群外的商业企业的,因此是不可置信的。F/OSS 的解决办法是实施通用性公开许可证(GPL, General Public License)或非盈利版权(Copyleft), GPL 或者类似的许可证制度(如 Facebook 所采用的 CPAL, Common Public Attribution License 以及 MPL, Mozilla Public License 等)强制参与者开放源代码和防止私有化(与传统的许可证制度刚好相反)。GPL 同时也对受信号效应驱动的参与者形成一种事前(Ex-ante)的可置信承诺,即他们的奉献不会被埋没,从而可以最大限度地保留这部分参与者的热情。

三、F/OSS 社群治理: 用户创新网络还是认知共同体?

开源软件的一个重要特点是用户可以根据自身的需要任意修改源代码。据此, Von Hippel 把 F/OSS 定义为用户创新社区,特别重点强调了粘着信息(Sticky Information)的作用。 von Hippel 定义粘着信息是从一个地方转移单位信息到另一个地方以供一个给定的信息需求者使用所发生的增量成本,成本越高,则粘性越强。信息粘性主要来源于其自身的性质,某些隐性知识对环境的依赖性较强,编码的成本高昂,因此不可能完全显化,也很难被转移^[8]。 von Hippel 进一步指出,由于成功的创新所需要的两种主要信息,即用户的需求信息和厂商的解决方案因附着在不同的主体上而相互分离,信息的粘性越强,则两种信息的桥接越困难,用户自己完成创新的倾向也就越强,也更有效率^[9]。因而,

von Hippel 认为开源软件所代表的这种分布创新模式实际源自于用户需求的特殊性,F/OSS 社群即是一种用户间的横向创新网络(Horizontal innovation network)^[10]。而这种用户创新网络的优势在于:首先,由于这种网络是基于客户化定制的,可以充分满足个性化的要求,因此不难吸引到足量的合作者参与设计和测试,F/OSS 社群生存和发展所需要的成员基础能够得到满足;其次,用户直接干预整个生产流程,可以保证项目有的放矢,避免盲目投资造成的浪费和效率低下;最后,体现了用户意志的产品也更容易推广,进一步延伸网络的边界,最终形成良性循环。

但这种基于用户创新的理论依然有些盲点亟需解释:如果网络内各元素是同质的,商业软件不能有效满足不同用户特定需求方面的弱点就消失了,那么与之相比,F/OSS 的相对优势又在哪里?如果网络内各元素是异质的但是互为补充的,满足某个用户的全部要求意味着其他用户必须再花费成本去除其中他们并不需要的功能,这无疑是非效率的,而且容易造成系统的不稳定。如果网络内各元素是异质的并且相互冲突,显然,必须有人决断那些元素会被优先考虑,哪些元素会被剔除。问题在于,应该赋予谁终极权力做出这样的决断,即使能找到这样的人,又如何保证他的决断是一贯理性和正确的?况且这也违背了 F/OSS 无中央决策机构预制技术路线的基本法则。

Edwards 试图用“认知共同体”(Epistemic Communities)、情境学习(Situated Learning)和合法的边缘性参与(Legitimate Peripheral Participation)来阐释 F/OSS 的协调和治理机制^[11]。认知共同体是由于外部环境动态演化所催生的群体,其主要功能是为决策机构提供知识以克服信息的不完全性、环境的不确定性和系统的不稳定性。认知共同体的成员拥有近似的知识架构、从业经验和爱好,分享共同的信仰、方法论和价值观,奉行公认的行为准则和判断标准^[12]。例如在 F/OSS 社群中,志愿者都崇尚资源的非私有化和利他主义精神,重视个人声誉,热衷于团队合作、信息与成果的公开和分享,信任合作者,并有条理清晰的连贯规则来评价和保证志愿者工作的质量。认知趋同有助于在保证效率和质量的前提下,降低信息的交易费用和通讯成本。从这个角度考虑,分布式创新与传统的团队生产有本质的不同:它是集体合作的结果,但它的工作又非常个人化,地理上的差异使得网络实时通讯很难实现。如果我们理解合作就是持续的信息和知识的流动,那么在 F/OSS 社群中,这种无间断的流动发生的机率是非常低的。无独有偶,Krishnamurthy 对 F/OSS 在线平台 Source

-forge 上的 100 项成熟产品的实证研究结果令人惊讶:样本项目中的绝大部分是由个人开发的,一般的情况下成员也没有超过四人,而且在这些 F/OSS 项目中,志愿者相互之间讨论和交流的机会并没有理论上演绎的那么多,因此 Krishnamurthy 认为 F/OSS 本质上仍然是“洞穴”(Cave)而非“集市”(Bazaar)^[13]。

Edwards 认为,F/OSS 志愿者初期的目的确实是为了满足自身的特定需求,但加入后对职业声誉的渴求逐渐取而代之成为主要的驱动力量,例如对 F/OSS 操作系统 FreeBSD 的调查显示,那些已经做出了重要贡献的志愿者往往执着于保持自己的好名声,最不能忍受自己影响力的衰减和地位的下降,因而往往会自发地继续努力改进自己开发的源代码。但另一问题是,既然认知共同体的治理需要成员有共同的信仰和价值观,也要有共同的知识背景,但如前文所述,共同体内部的交流手段十分有限,那么对于思维模式异质,主要受用户需求驱动的新加入者,如何帮助他们尽快实现社群化呢?Edwards 认为只能依靠情境学习和合法的边缘性参与解决这一难题。传统学习理论仅仅把学习抽象地看成知识从一个个体到另一个体的传播,而忽视了知识只有在特定的情况下才会发挥作用,从而把知识和其植根的社会和历史背景割裂开来。而情境学习理论认为,学习不仅仅是个体心理和知识的建构过程,而更是一个集社会化、实践化和资源差异化为一个不可分割整体的参与过程。知识的真实意义以及学习者自身的理念与认知都是在学习者和学习情境的互动、学习者之间的互动过程生成的,简而言之,情境学习是指在要学习的知识、技能的应用情境中进行学习的方式^[14]。“合法的边缘性参与”则是情境学习的核心概念和基本特征。基于情境的学习者必须首先得到共同体中现有成员的集体认同,方能成为“合法”的参与者,而不是被动的观察者。同时现有成员也承认边缘性参与者的学习地位,对他们在学习期间所做的工作和努力的不完善性给予宽容和尊重。另一方面,边缘性参与者必须承认和接受他们在共同体中的学习地位(类似于学徒),其学习过程也应该始终在共同体工作的情境中进行。边缘性参与者主要通过“自学”(即积极参与共同体的集体事务)来了解该认知共同体的制度、理念、编码方法和结构。总之,所谓“边缘性参与”是基于这样一个事实,即作为新手的学习者部分地、不完全地参与共同体的活动。“参与”意味着学徒(或新手)应该在知识产生的真实情境中,通过与核心层的专家、同伴的互动,学习他们建构共同体知识和价值观的方法和路径。通过这种渠道,最终逐渐转化为共同体的核心成员。

无论是用户创新网络还是认知共同体,都没有解决两个问题:一是 F/OSS 的初始动力问题。作为受信号效应或者用户需求驱动的志愿者,如果参加已经启动,正在运行的 F/OSS 项目,无疑将更快帮助他们达到目的,类似于企业总是喜欢招聘有工作经验的员工一样。试想如果每个企业都不给没有经验的新人以工作机会,工作经验的员工又从何而来?以此类推,如果每个志愿者都只愿意参加成熟项目,那么 F/OSS 还有机会成熟吗?但这种理论上的推演在现实中并没有发生,原因何在?为什么会有那么多志愿者愿意为前途未卜的 F/OSS 项目承担机会成本和风险?二是 F/OSS 社群的内部治理机制问题,无论是用户创新网络还是认知共同体并没有完全打开这一黑盒子,例如在认知共同体中,合法的边缘性参与者在向正式成员转化过程中的文化和价值观震荡,用户创新网络成员之间在参与动机不一致或者对技术发展路线观点不一致时的群体冲突等,如何妥善地解决这些分歧和矛盾以便维持 F/OSS 赖以生存的协调机制?

四、分布式权责： F/OSS 的治理和协调机制

共同的信仰和价值观并不等同于在整个开源社群内部一团和气,毫无争议。事实上,任何一个 F/OSS 项目从初生到成熟的发展历程中,总是充满了思想的分歧、矛盾甚至争端。例如对不同技术路线绩效的评定很可能会导致达尔文主义,这就会触及某些成员的直接利益(例如主要为获取声誉的参与者,如果他的技术路线最终不被社群采纳,显然他无从实现私人目标),再如不同身份背景和认知的成员在达成社群共识前的文化和理念冲突。这些分歧、矛盾和争端如果不妥善解决,会引起整个 F/OSS 社群的震荡甚至分崩离析。因此,即使是 F/OSS 这样的分布式创新模式,也要求有精密的内部治理和协调机制来处理这些事务,保证整个系统的稳定。如前文所述,F/OSS 社群有民主投票机制来尽量避免主观、专断和偏袒,但受制于有限理性,仍不可能保证任何结果都是绝对公正和客观的。另一方面,无论是用户还是受声誉驱动的参与者都有取而代之的激励:对用户而言,如果能让自己的想法成为今后技术发展的主导思路,无疑对更好地满足自己的特定需求是有利的;而对于后者,当他的声誉和知识积累超越了项目的创始者,即“功高震主”,掌控权的更迭就是顺理成章的事情了,何况赢得更大的发言权乃至登上金字塔尖对任何人来说都能带来成功的愉悦和自我满足。

那么,在个体高度自治的 F/OSS 社群中,实践中为什么很少出现对“舵手”(即系统集成者)的挑战,或者说试图背离原程序创立新进程的“分叉”(Forking)行为呢?Raymond 试图从文化和习俗的角度来解释这一 F/OSS 社群的自我稳定现象^[5]。Raymond 认为,F/OSS 社群的创始者或者舵手(类似于 Linux Torvalds 在 Linux 操作系统中的地位)通过知识创造的先占性取得话语权,开拓了所谓“知识领土”。按 Raymond 的理解,创始者或者舵手对其知识领土的管辖权是其“产权”的延伸,并通过戒律、规则和惯例来实现。对创始者或者舵手管辖权的挑战正是 F/OSS 社群严禁的行为之一,当然由于 F/OSS 的开放性质,维护这些戒律、规则和惯例还是主要依靠个人的自觉和合作态度。F/OSS 社群可能的惩罚措施只能是道义上的谴责或者对违规者实行禁入(即和普通人一样,他仍然可以自由获取源代码,但被排除在 F/OSS 社群的团队合作以外)。应该说,前者并无实质意义,后者本身就具有公用品的性质,很容易被搭便车(很难完全禁止所有成员私自与违规者接触),执行成本过高。

McGowan 借用了 Grossman—Hart—Moore 模型中资产专用性的概念,认为 F/OSS 社群开发者的人力资本具有专用性,即需相互结合才能发挥作用,禁入意味着获取其他成员知识、技能和经验的通路被阻断,与对源代码的阻断相比,这是一个可置信的威胁,因此开发者有接受被管辖的天然倾向^[6]。

在 Lerner 和 Tirole 的研究中,舵手则被赋予了更大的权限,他们不仅是创建者,也需要规划项目发展的愿景(例如展示对群的整体驾驭能力和系统集成能力)。此外,舵手还对社群内部的具体分工负指导之责,即负责把整个项目分割成若干个模块,并开发出超越阈值的初始代码以吸引合作者加入^[7]。但 Edwards 并不同意这一观点,他认为舵手显然没有这样的能力,只能扮演系统维护者的角色,提供初始的源代码,志愿者或者用户在此基础上根据自己的需要和理解各自开发愿景,换言之,其地位类似于街头卖艺者,刚开始只能靠他的高超技艺来吸引路人的眼球,如果他的技艺有足够的魅力,自然会有大量的观众愿意贡献金钱,此时他就具有了选择权,可以专门为馈赠多者表演^[8]。那么,如果舵手仅仅是系统维护和协调者,又如何行使对系统的管辖权和影响力呢?

在一些有代表性的 F/OSS 社群中,Linux 实行的是—种被称为“良善的统制”(Benevolent dictatorship)的治理机制,Linux 之父 Torvalds 对程序开发拥有最终的决定权。随着 Linux 的发展,结构越来越庞杂,整个系

统又被划分成若干个子系统。由若干通过认证的组长分别负责管理和维护与其他子系统的界面,这样也减轻了 Torvalds 本人的压力。Torvalds 和合作者的关系以及在系统中的地位是建立在绩效的基础上,并奉行“权利越大,责任越大”的原则。Perl 的治理机制则被称为“轮庄统制”(Rotating dictatorship),即系统的控制权是间或在少数高级成员中间传递的。Apache 则采取投票制度(Voting committee),即委员会成员通过电子邮件或者 BBS 投票,只要达到法定最低人数赞成,决议就可以通过。无论是哪种治理结构,可以看出 F/OSS 社群绝非完全的“无政府主义”,领导和治理依然对结构的健壮性和系统的稳定性举足轻重,但 F/OSS 领导者显然没有商业软件企业中 CEO 的权限大,后者可以任意解雇员工,影响下属的升迁,F/OSS 领导者并没有这样的权威,就如总设计师不可能干预到某个零部件的采购这一层面一样,这种治理结构可称为是一种分布式的权责模式。

当然,在这种分布式的权责模式中,舵手的地位不能只靠 Raymond 所主张的知识领土来维持。原因在于随着系统外延的不断扩张和日益分解成若干子系统,舵手的掌控能力也在边际递减,其他志愿者尤其下属小集团的负责人可以通过“干中学”逐渐弥补与舵手的知识差距,而且具体到某个特定模块,舵手更不可能和直接负责人一样熟悉。随着知识领先优势的逐渐丧失和子系统的逐渐壮大,分歧、争论甚至分裂将无可避免。舵手避免这种最坏情况出现的方法是“分而治之”,在分布式权责治理结构中,舵手的部分放权绝不是随意的。正如 Torvalds 曾表达的那样,他选择子系统负责人的标准是理解他本人思路、能够在他所制定的愿景上开发新功能,换言之,他能够完全信任和放手的志愿者才能够进入 Linux 的核心领导层(通过源代码认证实现,源代码的内涵可以体现开发者对社群规则的理解和决定)。轮庄统制和投票制度也基本上是基于同样的思路。举例而言,如果有两条不同的技术发展路线分别来自核心层成员和外部普通志愿者,舵手不会花费时间和其他成本去具体评估两条路线可能的绩效,核心层成员由于其可信赖的身份在竞赛中自动获胜。通过这种方式,舵手把整个社群牢牢地驾驭在自己手中,有异心的核心层以外人员很难渗透,这造成他们即使有另起炉灶之心,也无另起炉灶之力,其维稳效果可以达到近似等级制度(Hierarchy)的程度。

五、结 语

本文探讨了开源软件的基本性质和治理机制,发

现其治理机制是介于等级制度和完全开放之间的一种中间状态。开源软件向公众提供免费资源并完全开放源代码,具备公用品的部分性质,但其治理机制又具有等级制度的部分特征,并非完全依赖共同的信仰、价值观或者对声誉的偏好来维持系统的稳定。志愿者可以通过卓越的工作、持续参与的热情并贡献与系统领导者意愿相吻合的技术路线通过认证以获取进入核心层的机会,核心层由各子系统的实际控制者组成,系统领导者让渡部分权利给他们,通过他们驾驭整个社群。核心层以外的普通志愿者和用户实际上无法对社群决策产生实质影响,因此开源软件的治理机制具有分布式的特征,并不完全归属于用户创新的范畴(用户创新是一个结构松散的横向网络,且用户需求在其中发挥重要作用)。由于开源软件的治理机制弱化了用户的功能,需要中介机构作为桥梁,这也是类似 Red Hat 这样的商业企业介入开源软件市场化的原因之一。这些企业的介入反过来又激发了更多的志愿者加入 F/OSS 社群,道理显而易见:作为志愿者,总是希望自己的馈赠能更好发挥作用,服务于大众,而有良好市场运作经验的企业无疑将给志愿者们这样的希望。这可能也是众多动机不同的个体能够和谐地并存于 F/OSS 社群中,并保持 F/OSS 项目正常运转的重要原因。当然,必须注意到商业企业是以追逐直接的经济利益(金钱)为目的机构,和慈善机构毕竟有本质不同,那么如何在治理机制上避免对志愿者馈赠行为的“挤出”效应,是今后理论和实证研究中需要解决的问题。

参考文献:

- [1] von Hippel E. Innovation by User Communities: Learning from Open-Source Software [J]. MIT Sloan Management Review, 2001, 42(4): 82-86.
- [2] Johnson J. Open Source Software: Private Provision of a Public Good [J]. Journal of Economics & Management Strategy, 2002, 11(4): 637-662.
- [3] Harhoff, D., Henkel, J., von Hippel E. Profiting from Voluntary Information Spillovers: How Users Benefit by Freely Revealing Their Innovations [J]. Research Policy, 2003, 32(10): 1753-1769.
- [4] Lerner, J., Tirole, J. Some Simple Economics of F/OSS [J]. Journal of Industrial Economics, 2002, 52(2): 197-234.
- [5] Raymond, E. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary [M]. Sebastopol, CA, O'Reilly and Associates, Inc, 1999.
- [6] von Hippel E., von Krogh The Private-Collective Innovation

- Model in F/OSS Software Development: Issues for Organization Science[J]. *Organization Science*, 2003, 14(2): 209–223.
- [7] Benkler, Y. Coase's Penguin, or Linux and the Nature of the Firm[J]. *Yale Law Journal*, 2002, 112(3): 369–446.
- [8] von Hippel E. Sticky Information and the Locus of Problem Solving: Implications for Innovation[J]. *Management Science*, 1994, 40(4): 429–439.
- [9] von Hippel E. *Democratizing Innovation*[M]. Cambridge: MIT Press, 2005.
- [10] von Hippel E. Horizontal innovation networks—by and for users[J]. *Industrial and Corporate Change*, 2007, 16(2): 293–315.
- [11] Edwards, K. Epistemic Communities, Situated Learning and Open Source Software[R/OL]. <http://opensource.mit.edu/papers/kasperedwards-ec.pdf>, 2001.
- [12] Haas, P. Introduction: Epistemic Communities and International Policy Coordination[J]. *International Organization*, 1992, 46(1): 1–35.
- [13] Krishnamurthy, S. Cave or Community? An Empirical Examination of 100 Mature F/OSS Projects[R/OL]. http://firstmonday.org/issues/issue7_6/krishnamurthy/index.html. 2002.
- [14] Lave, J., Wenger, E. *Situated Learning: Legitimate Peripheral Participation*[M]. Cambridge University Press, 1991.
- [15] McGowan, D. Legal implications of open-source software[J]. *University of Illinois Law Review*, 2001, (1): 241–304.
- [16] Edwards, K. When Beggars Become Choosers[R]. http://www.firstmonday.dk/issues/issue5_10/edwards/index.html. 2000.

责任编辑: 陈于后

Literature Review on Governance of F/OSS Communities of Distributed Mode

REN Zhong

(School of Economics and Trade, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: Free and Open Source Software is an emerging production mode, with which many volunteers contribute to development of computer software without expecting to receive any individual rewards at all. This phenomenon has raised continuous concerns of scientists in economics, law, psychology, sociologist, and computer, etc. Nevertheless, the existing literatures mainly focus on explaining the motivation of participants, thereby neglecting the institutional and organizational aspects of F/OSS communities. This paper analyzes governance structure of F/OSS as a distributed regime where the one initiator or leader of F/OSS project distributes some rights and responsibilities to some trusted responsible person of sub-systems to form a core. In this way, he can control and coordinate the whole community without hierarchy.

Key words: F/OSS; epistemic communities; governance; distributed innovation; commons-based peer production