

软件反向工程：合理利用与结果管制

曹伟

内容提要：由于传统版权法基于“思想表达二分法”的理念，版权体系本身并不保护作者在作品中蕴含的思想，而只注重保护作品的形式。软件作品在版权法意义上具有若干种存在形式，从编写到运行中间经历了若干次的变形。通过对软件进行反向工程所了解到软件的设计思路或算法安排等“思想”，并不处于传统版权法的护佑之下，软件的设计者无法对反向工程的结果主张权利，于是便转而寻求对反向工程这一技术手段加以禁止。欧美的相关立法说明，反向工程应当被界定为一种中立的使用作品的方式，应当管制对反向工程结果的不正当使用，而不是限制反向工程行为本身。

关键词：软件 反向工程 功能性作品 权利限制

在软件行业的发展中，对现有软件进行反向工程，藉此提高自身的研发能力正在日益成为一种重要的研发手段。在目前我国大力发展软件行业的背景下，软件反向工程理应引起我们更多的关注。

一、软件反向工程的概念、由来及功用

在此前很长一个历史过程中，在版权领域里并没有反向工程这个事物的存在。反向工程的概念最早出现是在集成电路和布图设计保护领域，原意是指对他人“掩膜作品”的布图设计进行分析、评价，然后根据分析、评价的结果创造出新的布图设计。此后，反向工程进入软件版权保护的视野，开发人员试图通过对软件的功能进行解析，以此了解软件的编写代码，从而实现不同软件的兼容或者加以借鉴开发新的软件。

一般而言，软件反向工程通常是指对他人软件的可执行程序进行逆向的解剖、分析，从软件的目标代码入手，通过反汇编等编译方式尽量还原源代码，推导出他人软件产品的功能、组织结构、处理流程、算法、界面等设计要素。

普通作品以图文等通常的形态表现出来，基

本可以满足广大使用者（即一般意义上的读者）的使用需求，作品的使用者并不需要对作品进行特殊处理就可以对作品内涵的信息进行透彻的了解和充分的共享。一般而言，绝大部分作品的最终形式与其最初形式往往是一致的，因此，普通作品通常涉及不到反向工程的问题。

但就软件这种功能性作品而言，软件在版权法意义上有着若干种存在形式。从最初的源代码文本，经过编译变成对应的目标代码，再经过多次调试形成最终的可执行程序；软件的存在形态是一个连续的变化过程，中间经历了若干次的变形。最终的结果是，软件呈现在最终使用者面前的形式已经与最初设计师面前的形式发生了极大的变化。软件存在形态的变化导致从软件的最后存在形态（即可执行程序）中无法准确的获知软件的设计信息。对于普通作品而言，读者乃至社会公众都可以通过阅读、个体复制等多种方式对作品加以合理的使用，并获得对作品信息的充分共享。而软件的使用者和潜在的社会公众则无法通过正常的使用方式来合理共享软件作品内在的信息。

在软件存在形态发生改变的过程中，各种表

作者简介：曹伟，法学博士，西南政法大学知识产权学院副教授，主要从事知识产权法领域的研究。

现形式之间存在紧密的内在联系,一种表现形式转化为另一种表现形式,要借助特定技术才能实现。这种转换技术本身是客观中立的,也是双向的,既可以转换过去,同样还可以转换回来。因此,在通常的阅读方式不能满足对作品信息进行合理共享的要求时,就需要提供一种新的途径,将软件作品的最终形态尽量还原成最初形态,以便利社会公众在信息共享方面的需求。而这往往需要通过技术途径对软件的形态变换进行复原和追溯。这种需求直接促使软件反向工程的出现,也使得软件反向工程被看作一个重要的问题,并专门加以分析。

一般意义上,版权制度保护的是创意性的表达,既然表达是公开的,那么就无所谓反向工程。然而,软件的技术性、功能性等特征使其不同于传统的文字作品,软件作品的目标代码与源代码是可以分离的,而通常软件的功能性使用也不以获得源代码为要件,软件作品公开表达的不是源代码,这就使软件作品与反向工程产生了特定的联系。

软件的反向工程在软件研发领域,对于开发其它软件具有特别重要的借鉴价值。总体说来,软件反向工程有助于大幅度降低软件研发成本,并充分激励软件的研发热情。美国学者Bernard A. Galler曾经将反向工程的运用与传统行业中的拆解机器作了一个形象的对比。例如,在商业竞争中,竞争对手拆解一块手表,他可以看到手表中的全部零件以及这些零件如何被组装到一起,但是他却不可能了解到生产制造过程背后的设计信息,即为什么会按照特定的方式来组装排布这些零件,所以,拆解手表等机器设备往往无法为拆解人提供改进生产效率的商业信息。竞争对手如果要生产同类手表就必须建立一个手表生产厂,并且承担全部的建设开办支出。然而,对软件进行反向工程却可以使该软件程序相关的各种设计信息被充分的暴露和了解,因此,竞争对手往往得以在生产同类软件的时候,以极小的代价和很短的时间延误获得一个明显的优势地位^①。

此外,反向工程还将使软件的兼容性增强,

重复的、无序的、低效率的软件开发得以最大程度的减少,软件开发的总体成本随之下降。同时,允许反向工程还将有助于限制平台软件开发商的垄断优势,从而打破恣意上涨的许可费用,降低具有广泛兼容性的应用软件的开发成本,而同类应用软件的竞争也会最终使应用软件的成本进一步降低。如果反向工程不被允许,则应用软件的开发者无法获得平台软件各种接口的兼容信息,这样一来,平台软件开发商的垄断地位在较长时间内无法被打破,他们就可以利用垄断优势提高定价或者联合定价,使软件的价格偏离合理范畴。因此,不支持软件反向工程就很可能要接受高价软件带来的成本激增。

从创新的角度来看,技术创新的热情往往依赖于技术竞争的可行性,反向工程意在打破技术垄断,而打破技术垄断恰恰是技术竞争的基础。反向工程为技术攻击和防御提供了可能,使技术更新和竞争成为现实。同时,反向工程作为一种经济高效的手段,为一些新入场的小公司了解开发信息从而开发一些小巧、实用的应用软件提供了可能并能够藉此获利。随着时间的推移,这样的竞争格局最终将带来技术市场的良性变化,也只有在这样的市场氛围下,才有可能最大程度地激励技术创新和开发的热情,从而推动整个社会科技水平的不断提高。反之,技术垄断在一定时期内可能会推动技术进步,但最终由于缺乏竞争,垄断者必然丧失技术创新的原动力而停滞不前^②。

二、软件反向工程法律地位解析及立法改进

由于传统版权法基于“思想表达二分法”的理念,并不保护作者在作品中蕴含的思想,而是只保护作品的表现形式,因此,通过对软件进行反向工程所了解到的软件的设计思路或者算法安排等“思想”并不处于版权法的护佑之下。由于软件的设计者无法对反向工程的结果主张权利,于是便转而寻求对反向工程这一技术手段加以禁止。在这一点上,各个国家的态度存在着微妙的差别。

^① (美) Bernard A. Galler: 《Software and Intellectual Property Protection: Copyright and Patent Issues for Computer and Legal Professionals》, Quorum Books, 1995.

^② 胡震远: 《软件反向工程的合法性探讨》, <http://www.ipr lawyers.com/iprtml/0804/2006-3/6/20060306170456967.html>, 2010年7月12日访问。

美国之前的判例倾向于认可反向工程的合法性,并认为反向工程在一定条件下构成对软件的合理使用并被允许^③。从这些判例分析,美国法院的判决并不仅仅考虑反向工程行为是否带有商业目的,而是认为反向工程行为是接触程序中不受版权保护的要素(例如设计思想)的唯一方法或者必要手段时,这种反向工程行为属于合理使用。同时,美国法院还认为,此类反向工程行为增加了独立设计的电脑游戏程序的数量,客观上有力地促进了创新,完全符合版权法立法的宗旨。如果判决对计算机程序进行反汇编是非法的,将会赋予版权人对程序中不受保护的思想和功能性概念享有实质性的垄断权^④。美国司法实践中的案例对于各国关于反向工程合法性的理解形成了重大影响。

根据美国司法实践的经验,满足一定条件的反向工程行为构成合理使用,而这些条件通常包括:1.对目标程序的使用属于合法使用,而不是未经授权的使用;2.行为的目的仅限于实现开发程序与目标程序的兼容。这些判例中总结出来的判解观点得到了美国版权法的确认。1998年10月28日,时任美国总统的克林顿签署了《千禧年数字版权法》(DMCA),该法案对美国版权法做出了重大调整。修改后的美国《版权法》第1201条(f)款明确规定了反向工程属于不构成版权侵权的例外情形,允许软件的合法使用者在《版权法》允许的范围内,对限制非法使

用软件的技术措施进行规避或者进行类似的技术处理,以识别该软件必要的组成部分,从而使该软件与其他软件兼容^⑤。美国加州伯克利大学的Pamela Samuelson教授和Suzanne Scotchmer教授合作发表的关于反向工程的研究专著《反向工程的法律和经济》(《THE LAW ECONOMICS OF REVERSE ENGINEERING》),被认为是美国对反向工程里程碑式的研究总结。其开篇第一段话是:“反向工程是一种从人造物品中提取技巧和知识的过程,这种做法被接受和实践已经有很长时间。法学家和经济学家都认可反向工程是获取这些信息的恰当手段,即使其意图是制造一种产品并从被反向工程的厂商手里夺取客户。既然有这一共识,过去几十年里反向工程所遭受的非议是令人吃惊的。”这表明美国主流学术界已经开始接受有条件的反向工程。由此不难发现,美国对于软件反向工程的态度非常明确:允许限定条件下(即实现软件兼容为目的)的反向工程,同时,对于其他目的的反向工程行为严格加以禁止。

在欧洲,1991年5月14日通过的《欧盟关于计算机软件保护的指令》在全世界范围内首开先河明确肯定反向工程的法律地位。该《指令》第6条规定,为获得必要的信息来独立开发兼容的软件,合法用户可以对软件进行复制和编译,而无须经过权利人的同意^⑥。一般认为,欧盟对反向工程做出了以下限制:1.进行反向工程的主

^③ 1992年,Atari Games Corp. vs Nintendo of America Inc.一案中,Atari公司根据Nintendo公司在美国版权局办理版权注册时留存的样本开发了一套针对Nintendo软件的破解程序,并加以应用。美国联邦上诉法院就该案做出的终审判决中认为出于兼容目的,对合法使用的目标程序进行反向工程构成对作品的“合理使用”。此后不久,在Sega Enterprises Ltd. vs Accolade Inc.一案中,被告Accolade通过反汇编等手段对原告Sega公司开发的游戏软件进行了反向工程,最终成功地使得Accolade的游戏软件在未经Sega授权的状态下也得以运行,并在市场上独立销售,与Sega公司的产品和经Sega公司授权的第三方开发者的产品形成竞争。美国加州地区法院一审判决被告侵权成立,其理由是被告进行反汇编的目的具有商业性质,不属于合理使用。该案上诉后,美国联邦第九巡回上诉法庭最终判定,Accolade公司有权“合理使用”Sega公司的游戏软件,以便学习如何为其游戏平台开发兼容软件。

^④ 张晋静:《软件反向工程合法性的经济学分析》,载《电子知识产权》2003年第7期。

^⑤ 曹伟:《1998年〈千禧年数字版权法〉评述》,《知识产权理论与实务》法律出版社2003年版,第272页。

^⑥ 《欧盟关于计算机软件保护的指令》的第6条分3款对反向工程的条件、排除情形及解释限制作了如下规定:

“1.若(本《指令》)第4条第(a)、(b)项3规定的代码复制及其形式转换系获取信息所必需,而该信息又是实现独立创作的计算机程序与其他程序兼容性所必需,则无需获得权利持有人的授权,但须同时满足下列条件:

- (a) 此等行为系由被许可人或者有权使用该程序复制品的人或经授权代表该有权使用人从事该行为者所完成;
 - (b) 实现兼容性所必需的信息不被(a)项所指的人事先轻易获得;
 - (c) 此等行为限于实现兼容性所必需的源程序的相应部分。
- 2.依前款规定所获的信息不得适用于下列任一情形:
- (a) 用于实现独立创作的计算机程序兼容性以外的目的;
 - (b) 给予他人,除非实现独立创作的计算机程序兼容性所必需;
 - (c) 用于开发、复制或营销与该计算机程序实质相似的程序或其他任何侵犯著作权的行为。

3.根据《保护文学艺术作品伯尔尼公约》的规定,本条规定不得解释为允许适用本规定不合理地损害权利持有人的合法利益或与计算机程序的正常使用相冲突。”

体限于软件的合法用户；2. 通过反向工程了解的“必要的信息”应当无法从其他途径轻易取得；3. 只能对生产兼容程序所必需的那部分计算机程序进行反向工程；4. 通过反向工程获得的信息不能用于除实现软件兼容以外的其他目的；5. 反向工程不得合理地损害权利人的正当利益或妨碍计算机程序的正常使用。显然，欧盟在反向工程问题上与美国可谓血脉相通，他们都规定只有软件的合法用户才能进行反向工程，反向工程的目的也偏重于要求实现计算机程序的兼容，对除此以外的其他反向工程行为都加以禁止。1991年11月WIPO关于拟定《伯尔尼公约》议定书的专家委员会第一次会议的备忘录第37条和第38条中也涉及到反向工程，其观点与欧盟《指令》是一致的。此后，1993年《俄罗斯联邦著作权和邻接权法》第25条第2款和第3款作出了与欧盟《指令》相似的规定。

我国作为一个软件研发能力相对滞后的国家，应当允许甚至鼓励软件设计人员大胆借鉴国外先进的软件设计思想，以便实现我国软件行业的跳跃式发展，这就要求我们的相关立法要创设一套适宜的体制。明确说，即应该对软件的反向工程设定较低的条件。然而令人遗憾的是，我国现行的《著作权法》和《计算机软件保护条例》缺乏关于反向工程的明确规定。在《著作权法》中，只能从其第22条中寻觅到一丝软件反向工程的踪迹：因科学研究使用已经发表的作品属于合理使用的范围^⑦。如果对这里规定的“科学研究”作扩大解释，可以认为反向工程就是一种开展软件科学研究的重要手段，因此，软件反向工程构成合理使用。这种推导虽然没有问题，但苦于《著作权法》中对科研使用的形式做了明确的表述：限于为学校课堂教学或者科学研究，翻译或者少量复制已经发表的作品，供教学或者科研人员使用。因此，直接把反向工程纳入科学研究的范畴存在明显的障碍^⑧。

此外，在《软件保护条例》中也可以依稀寻觅到软件反向工程的身影。该《条例》第17条规定：“为了学习和研究软件内含的设计思想和原

理，通过安装、显示、传输或者存储软件等方式使用软件的，可以不经软件权利人许可，不向其支付报酬。”这个条文属于未穷尽的列举，我们可以大胆的推论“为了学习和研究软件内含的设计思想和原理”的一切方式都是被允许的，其中也包括以反向工程的方式合法使用软件。

值得注意的是，如果这一推论成立的话，那么我国对反向工程的立法与前述美欧的立法就存在较大的差异。虽然，我国也要求反向工程的前提是对软件的合法使用，但却并没有对反向工程的目的加以限制，一切“为了学习和研究软件内含的设计思想和原理”都构成正当的目的，而不只仅限于实现软件兼容。应当说，这拓宽了反向工程适用的空间，对于发展我国的软件产业是有益的。但是，这种欲语还休的做法却不值得称道。

我们在立法规定上只要达到国际条约中的最低保护标准即可，在此前提下根据自己的国情做出具体选择完全无可厚非。美欧地区软件研发水平较高，保护水平相应也水涨船高。我国的现实国情决定了我们没有必要盲目跟风，其中自然也包括关于反向工程的规定。

还有一个问题需要注意，即反向工程是否是获取信息的唯一手段。对此，有两种观点：一种观点认为，若通过其他方法可以获得相关技术信息，则实施反向工程即为非法；另一种观点认为，如果反向工程是获得该等信息的最佳手段，即应允许反向工程。笔者基本同意后一观点，前一观点虽然更为简明和确定，但却忽视了效益的要求。如果通过其他途径获取信息需要花费大量的精力和财力，而软件开发商仍然不能运用简便易行的反向工程，那么反向工程制度显然形同虚设。因此，欧盟规定的相关条件是反编译的实施者在实施反编译之前不能轻易地获取兼容信息。“不能轻易获取”意味着通过其它途径获取兼容信息的努力是需要的，但当反向工程的效益具有明显优势时，对其它途径不应再作苛求。

至此，笔者得出一个结论，即反向工程作为一种深入了解学习软件的技术手段本身是中性

^⑦ 杨婵：《论计算机软件反向工程的合法性问题》，载《法律科学》2004年第1期。

^⑧ 《著作权法》第22条规定，“在下列情况下使用作品，可以不经著作权人许可，不向其支付报酬，但应当指明作者姓名、作品名称，并且不得侵犯著作权人依照本法享有的其他权利：…（六）为学校课堂教学或者科学研究，翻译或者少量复制已经发表的作品，供教学或者科研人员使用，但不得出版发行；……”。

的,应当被允许进行。合法用户都有权通过反向工程深入到软件的内核中去了解软件的构思、结构和原理。此时,新的问题随之出现:使用反向工程的结果是否存在限制?在美国的案例中,为了实现兼容的目的,可以通过反向工程的方法了解软件的思想、算法、原理等不受著作权保护的要素,但是允许进行反向工程并不能做简单化的理解,至少不能理解为允许利用这些信息开发与原软件实质性相同的软件。简言之,利用反向工程了解到的软件内在原理等要素虽然不受版权法的保护,但借此开发相同实质的软件的做法显然是一种不正当的竞争手段,应当受到抑制。

总的说来,未来我国就此问题的立法,有两点必须要充分明确:首先,为了确保公众的利益,防止版权人过分垄断软件的使用,我们应在立法中充分认可反向工程的法律地位,明确赋予使用者进行反向工程的权利。具体而言,可以在软件版权的权利限制这个环节中,将反向工程明确规定为合理使用的一种方式。其次,为了保障版权人的基本利益,对于反向工程的实施必须设定一套严格的限制条件。反向工程接触到的是不受版权法保护的思想层面的东西,但是如果这种接触的结果本身超越一定的限度,仍然会构成不合理而被法律所否定。

三、软件反向工程的实施条件

通过前述分析,反向工程存在的合理性可见一斑。但允许反向工程并不意味着对软件版权人的权利完全不予保护,必要的限制条件仍是不可避免。在这个问题上,欧盟的《指令》殊值借鉴。依笔者看来,满足以下条件时,对软件的合法合理的反向工程才能够成立:

一是反向工程以合法获得软件为前提条件,只能由合法使用人完成。实施反向工程的对象必须合法,若通过非法手段获得软件,随后再行复制、转换仍为合法,则犹如承认毒树之果,知识产权保护的基础就毫无依托了。因此,反向工程行为的对象必须以合法获得为基础;反向工程行为的主体则必须是软件的合法使用者。

二是反向工程不以获取必要的兼容信息为唯一目的。只要是软件的合法使用者,就有权对其

合法使用的软件进行反向工程,并且不应当限制反向工程行为的目的和性质。换言之,即使是非个人的或者商业性的反向工程作为一种合理使用软件的手段都应当得到许可,反向工程的目的也不应局限在软件兼容这一狭窄范围之内。

三是通过反向工程所形成的作品不得进行盈利性使用,具有独创性的新作品除外。虽然行为的目的不做严格限制,但对反向工程行为的结果,则应该严格加以管制,不得擅自用于盈利性目的。软件的版权保护虽然并不排除创作上的雷同,但从美国司法实践中的“净室”理论来判断,允许接触加实质相似无异于允许盗版,整个版权保护制度将不复存在。反向工程的意义在于通过剖析已有软件以获取有用信息,进而创造出独立的软件作品。因此,如果反向工程的结果只是以已有软件作品为基础,对作品内容做出变换表达的作品,顶多只能成为演绎作品。只有当运用反向工程所创造的是具有独创性的新作品时,这一新的软件作品才会得到版权法的肯认和保护。

四、软件反向工程禁止条款有效性辨析

由于反向工程获得法律的认可似乎已成为一种不可逆转的潮流,因此,相当部分的软件版权的权利人变换了新的方式试图持续强化其版权保护的力度和深度。版权人在软件的使用过程中,往往借助其优势地位迫使使用人与其缔结反向工程禁止条款或者协议。藉此约定,使用人不得对软件进行反向工程,从而阻止使用人以及潜在的社会公众对其软件实施反向工程,最终消减、规避,甚至取消上述反向工程例外的适用空间。

那么,这类条款或协议在合约双方(即版权人和使用人)以及通过正当途径接触软件的第三人之间,是否具有法律效力?笔者认为,此类条款是无效的,在合约当事人之间自始就不产生法律效力。

就合同法的基本原理而言,违反公共政策和法律强制性规定的合同条款无效^⑨。前述分析已经充分说明,软件反向工程是软件研发领域的一项重大公共政策,并在立法中得到一定程度的肯定。从这个角度出发,反向工程禁止条款与反向工程豁免这一公共政策相违背,依合同法规定应为无效。

^⑨《合同法》第52条规定:“有下列情形之一的,合同无效:……(四)损害社会公共利益;(五)违反法律、行政法规的强制性规定。”

此外,包括软件版权在内的整个知识产权都是法定权利,不允许权利人擅自创设权利。若允许权利人私设权利,必然使得原有的利益平衡机制被打破。展言之,就极易导致知识产权人的权利滥用,在确立软件版权保护原则的同时,同样也应当充分警惕权利被滥用。应当建立一个“适度而合理”的保护机制,在这个机制中,不应当对其中任何一方面过分放大。而此类反向工程禁止条款打破了软件版权中原有的利益平衡机制,理应加以否定。

过分强化软件的版权保护,还会有损于未来建立的软件专利保护制度。就软件而言,版权法的未来保护应该与专利法的保护衔接起来。专利制度是以短期的独占保护换取发明人的信息完全公开,专利权人在该保护期内享有排他性极强的独占权,具有竞争上的完全优势。版权制度提供的则是力度较弱的保护,但授权条件通常比较容易成就。软件的权利人往往会有这样一种冲动,即一方面希望能够方便地以版权方式获得权利,另一方面又希望强化版权的保护,在权利的行使上获得类似专利保护的强大效能。软件版权人迫使使用者签订反向工程禁止条款,就是强化版权保护的一种努力。软件版权人试图通过私权之间的交易,消除反向工程对软件内在功能、组织结构、处理流程、算法、界面等信息要素进行披露的可能,从而达到尽可能延长保护其内在的设计思想等有价值信息的目的。如果对此类禁止条款的效力予以确认并给予保护,则无异于把版权与专利等同起来,必然造成专利权制度功能的部分丧失,而丧失的代价则是信息的不公开和垄断,这与信息公开、自由传播的知识产权立法初衷是严重违背的。

总之,软件版权人利用其优势地位,以私权形式所达成的反向工程禁止条款已经构成权利滥用,此类条款的效力理应不予认可。在软件的版权保护中,软件的版权人无权阻止社会公众通过反向工程这一正当手段获取软件的内在设计思想等信息。

五、功能性作品之反向工程：版权和专利两极划分的破拆

按照版权法中传统的“思想和表达二分法”,要想使自己的思想为他人所感知,就必须把这一思想通过某种外在形式表现出来。而版权法所保护的客体仅限于思想的表现形式,而不能触及抽象的思想本身。此外,由于作品主要满足人的精神需求,而工业产品主要满足人的物质需求,因此版权法中实际上还存在“实用和非实用二分法”。从这个角度出发,可以进一步做一个细分:就思想的表现形式而言,具有功能性和实用性的表现形式归专利法保护,不具有功能性和实用性的表现形式归版权法保护。知识产权法这一两极体制也已经在国际公约的框架中得以确认,并已经存续了一个多世纪。保护文学艺术作品的《伯尔尼公约》和保护工业产权的《巴黎公约》的不同分工,就是这种两极划分的典型表现^⑩。这种划分凝聚着这样的共识:在技术发明创造和艺术作品创作领域有着不同的利益平衡关系,法律应当区别对待技术与艺术,提供不同的调整方式。技术发明的创新,通常是寻求有效解决某一问题的新方案。专利法保护此类客体,对竞争对手的限制非常严厉,因此,有必要采取严格的审查标准。而对于文学艺术作品,在保护条件上则无须进行价值判断^⑪,也不存在妨碍竞争的问题,相应的,版权法给予较宽松的入门条件,给予较长期限的控制。但为了防止出现阻碍创造自由的后果,要求保护仅仅在表达层面,而不能高度抽象。在传统的两极划分体制下,专利法与版权法相互排斥对方所保护的客体。专利法通过印刷物规则^⑫、抽象思想不能获得保护规则等途径,直接否定作品类型的“创造成果”的专利客体地位;而版权法上思想与表达的二分原则、版权不保护功能性要素等原则刻意的将专利

^⑩ J. H. Reichman, “Legal Hybrids between the Patent and Copyright Paradigms”, 94 Colum. L. Rev. 2432, 2451 (1994).

^⑪ 版权法在确定作品保护界限的时候,往往只进行事实判断,即对是否具有独创性这一事实进行判断,而无需对作品本身的艺术价值进行价值判断。存在独创性的作品即可获得版权保护,即使这个作品本身在艺术价值上乏善可陈。

^⑫ 印刷物规则(Printed-Matter Doctrine)是指,单纯的文字、线条、图案的编排结果并非专利法意义上的客体。如果发明的技术方案仅是将信息印制在物质载体上,而信息同载体之间不存在结构性或功能性的联系,这类发明也不是专利法意义上的客体。Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions”, 39 Emory L. J. 1025, 1037 (1990).

法上的客体排除出版权法的视野^⑬。从这个意义上讲,印刷物规则背后的考虑是作品的文本表达形式已经被版权法所保护,并在保护表达和思想自由之间维持了平衡;如果专利法介入,可能会打破这一平衡。

当版权与专利两极划分体制被确立为知识产权领域公理性的制度安排后,人们在法律思维中将版权作品与专利技术这两类客体的外在形式进一步固定下来。于是,在判断一项新的客体究竟属于专利法客体还是版权法客体时,最重要的就是看该客体到底更类似传统的作品还是更接近传统的技术,然后进行非此即彼的保护模式选择。在绝大多数情况下,形式是第一位的,两极制度背后的立法初衷则被有意无意地忽略。

然而,客观上存在着有一部分特殊的作品,这些作品除了在表达形式上具有价值以外,作品本身还具备一些实用性的功能,这类作品被概括为功能性作品。功能性作品中包括软件、集成电路等类型,它们作为作品有具体的表达形式:软件以代码的形式出现,集成电路则以拓扑图的形式出现。从这一点上分析,其版权性毋庸置疑。同时,软件也好,集成电路也罢,它们本身都具有其对应的特定功能,因此,除了满足人们阅读层面的需求以外,它们与身俱来的技术功效,使得这些作品的外延进一步切入了专利法原来的势力范围,甚至模糊了版权法中“版权保护不延及思想”的基本原则。虽然在面对计算机和集成电路等功能性作品的时候,人们往往倾向于沿袭两极划分的习惯思路,但是这种习惯思路似乎在面对反向工程问题时失去了效用。

版权法中对于作品的保护强调客观表达形式的控制和维护,从这个意义上来说,传统版权法中根本不会产生对于反向工程的诉求。反向工程存在的逻辑前提,即为被反向工程对象的表达形式的不唯一性,换言之,被反向工程对象的外在表达形式与技术内核之间不存在同一性,而是存在对应性,经过形式变换方能最后完成。此时,

版权法对于表达形式的孜孜以求,反而变成了版权法保护此类作品的一个软肋。既然表达形式不唯一,那么对表达形式的控制就变得不确定,传统的控制表达形式开始失效。不难发现,反向工程之于版权,更多的是一个专利制度中的舶来品。专利法强调对思想的保护,同时,对于反向工程等深入了解技术思想的方式进行适度的管制。功能性作品的出现,使得这种既有的制度安排发生了奇妙的交叉,在版权法里开始出现反向工程的空间。诚然,这个舞台还仅限于此前探讨的功能性作品这一特定领域之中,但是功能性作品这种跨越版权和专利的特殊客体,的确使得传统的版权理论遭遇了严重的冲击。

软件作为一种功能性作品,其表达形式的多重性,导致版权法为其提供的形式保护显得苍白无力,因为真正具有价值的既不是源代码形式的作品表达形式,也不是经过编译手段变换的最终的可执行代码,而是隐蔽在这些代码背后的设计思想。那么,这就产生了一个版权向专利过渡的引子。软件作为导火索,迫使我们必须面对并回答这样一个一直以来困扰知识产权法学界的问题:除了具有可版权性以外,软件是否具有可专利性?本文在对软件版权保护反思的基础上,将进一步对其可专利性问题展开探讨,这里暂不展开。笔者在这里想要指出的是,即使为软件提供专利保护,但鉴于专利保护的门槛比较高,因此,能够进入专利保护范围的毕竟只是少数,那么,大多数达不到专利保护门槛的软件,我们仍然将在版权法层面上为其提供保护。

正是基于“版权保护不延及思想”这一版权法基本原则,本文此前的研讨方才得出这样一个结论,即对于这些版权法视线内的软件,反向工程被认为是一种中立的使用作品的方式,属于合理使用的一种。版权法对反向工程的管制,更多的应定位于对反向工程结果的不正当使用的控制上,而不是限制反向工程行为本身。■

^⑬ 崔国斌:《“文字作品”的专利法保护——计算机程序文本的可专利性分析》,载《法律适用》2005年第9期。