

文章编号: 1674 - 5205(2013)05-0187-(008)

# 计算机软件著作权保护对象范围研究

——对美国相关司法探索历程的分析与借鉴

张吉豫

(中国人民大学法学院,北京 100872)

**〔摘要〕** 作为功能性技术产品,计算机软件在我国著作权法下的保护范围仍处于模糊状况。美国司法实践中对软件著作权保护对象范围的界定经过了较长期探索,积累的理论和司法审判结论对世界软件产业及司法实践影响广泛。我国司法可借鉴美国的经验、理论和案例,以立法目的为出发点,考量技术领域的利益平衡关系,在司法实践中对软件著作权保护对象范围做出更加明确的合理界定,从而在激励软件开发与促进技术的传播和进步之间实现最佳平衡。

**〔关键词〕** 软件著作权; 著作权对象; 思想表达二分法

**Abstract:** Computer softwares are functional, technical products. The scope of copyrightable subject matter in computer software is still vague in China. The United States has a long-term judicial exploration experience about the scope of copyrightable subject matter in computer software, and the theories and judicial conclusions have widespread impacts on software industries and judicial practices all over the world. Our judicial practice can learn from American judicial experiences, theories and cases, consider the balance of interests in technical field, and make a more clear and reasonable definition of the scope of copyrightable subject matter in computer software based on the purpose of copyright law legislation, thus to build the best balance between inspiration of software development and promotion of the propagation and progress of technology.

**Key Words:** software copyright; subject matter of copyright; idea-expression dichotomy

中图分类号: DF523.1 文献标识码: A

软件著作权保护对象范围的明晰是调整计算机软件开发和使用中产生的利益关系的重要问题。我国现行《计算机软件保护条例》规定了条例所称“软件”指计算机程序及其有关文档,并进一步规定了体现“思想/表达二分法”及“思想表达合并原则”这些现代著作权法基本原则的条款<sup>①</sup>,对软件著作权保护对象范围进行了限制。然而,思想与表达之间界限的模糊性使得软件著作权保护范围存在许多争议,给法律适用和司法实践带来了一定的困惑。程序流程、数据结构、菜单命令、编程接口等软件成分是否应受到、以及在怎样的条件下可受到著作权法保护?其依据为何?在司法实践中应如何判断软件间的实质相似

性?美国数十年的司法实践给世界各国提供了借以破解这些困惑的经验、理论和典型案例参考。

## 一、我国司法实践中对软件著作权保护对象范围界定的模糊状况

1997年“曾小坚、曹荣贵诉连樟文、刘九发、深圳市帝慧科技实业有限公司计算机软件侵权纠纷案”<sup>②</sup>是我国较早涉及软件著作权保护对象范围争议的案件。一审法院根据鉴定分析专家组的鉴定报告,裁定被告的软件与原告享有著作权的“公安基层业务管理系统”软件具有实质相似性,侵犯了原告的软件

<sup>①</sup> 我国《计算机软件保护条例》第6条规定,对软件著作权的保护不延及开发软件所用的思想、处理过程、操作方法或者数学概念等;第29条规定,由于可供选用的表达方式有限而与已经存在的软件相似的,不构成对已经存在的软件的著作权的侵犯。

<sup>②</sup> 广东省深圳市中级人民法院(1997)深中法知产初字第007号。该案中,原告开发了一套“公安基层业务管理系统”,被告曾向原告提出购买其软件著作权,被原告拒绝,后来原告发现被告销售了其软件似乎相似的软件,于是起诉被告侵犯其软件著作权。

收稿日期: 2013-04-06

基金项目: 中国博士后科学基金资助(2013M530794)

作者简介: 张吉豫(1982—),女,河南南阳人,北京大学计算机系统结构专业理学博士,中国人民大学法学院博士后,研究方向为知识产权法。

著作权。后被告上诉至广东高院,广东高院维持了原判,该案判决中介绍了鉴定报告的意见:两款软件相比,“两系统的数据库、屏幕显示及其有关的基本数据库、数据结构、参数选用,以及数据之间的关系都是相同的,而且多处出现设计中的特征相同(包括其中的差错及失当之处)这些现象,独立设计的软件中是不可能出现的。尽管它所用的开发工具不同,但两系统的软件仍存在实质相似性。而连樟文、刘九发向专家组提供的软件程序是临时修改出来的,不真实的。”<sup>①</sup>连樟文、刘九发不服该判决,委托中国科学技术法学会科技评价委员会组织专家进行了鉴定。鉴定小组给出的《咨询意见》认为两个软件不构成实质相似,二人据此上诉至最高人民法院。<sup>(1)</sup>最高人民法院将该案发回再审查,并发函指出(《鉴定报告》)并未对原被告软件的源程序或目标程序代码进行实际比较,而是通过比较程序的运行参数(变量)、界面和数据库结构,就得出了两个软件实质相似的结论。运行参数属于软件编制过程中的构思而非表达;界面是程序运行的结果,非程序本身,且相同的界面可以通过不同的程序得到;数据库结构不属于计算机软件,也构不成数据库作品,且本案原告的数据库结构实际上就是公安派出所的通用表格,不具有独创性。因此,《鉴定报告》所称的两个软件存在实质相似性,并非著作权法意义上的实质相似性。”<sup>②</sup>

最高人民法院的上述意见为确定软件著作权涵盖范围、区分思想与表达提供了指导意见,但未进一步发展形成关于软件著作权保护范围界定的系统解释。2003年广东省高级人民法院再审后维持了原判,认为一审法院委托的软件侵权鉴定“程序合法,其所出具的鉴定报告具有证据效力”;连樟文、刘九发委托的鉴定小组所出具的《咨询意见》是“单方委托,其送检的软件未经双方质证,故鉴定程序不符合法律规定,该咨询意见不具有证据效力”<sup>③</sup>。除此之外,我国后续司法实践中仍然反映出对软件著作权保护对象范围界定的一定差异和论证依据的模糊。英特尔诉东进公司一案<sup>④</sup>曾在我国引发关于“头文件、软件兼容性与版权保护”的讨论,但因该案和解而未能有进一步的司法意见。一些案例中,被告会以相似之处是不受著作权法保护的部分为理由进行抗辩,法官很少接受此类抗辩,而且对不采信理由也往往缺少适当说明。例如北京惠斯特诉斯惠特一案<sup>⑤</sup>判决认为,(被告辩称)相似之处仅限于参数、变量、界面等并不受到著作权法保护的部分,但菜单定义、数据表对应关系显已超出参数、变量、界面之范围”。但该判决中并未解释在著作权法中,为何区别对待菜单定

义与参数、界面。也有案件判决继承了在美国已不再采用的早期判例的思想和方法。如在长沙青果园诉长沙诚光一案<sup>⑥</sup>中,法官指出“如果程序的思想概念和表现形式不具有同一性,即为了达到一个预期目的存在多种方法,而且所选用的方法对于实现该目的是并非必须的,那么这些方法就属于表现,而不是思想概念。这时,计算机程序的‘结构、顺序与组织’就被认为是程序的表现,受著作权法保护。本案中,原告青果园公司开发的软件所采用的方法并非唯一的,故其表现形式应受著作权法保护。”该观点与美国早期 Whelan 案<sup>⑦</sup>思想一致,但 Whelan 案提出的原则和方法没有充分考虑受保护的程序中也可能包含一些不受保护的代码(程序的局部还可能存在对于实现局部的“思想”所必须的“表达”),在美国已不再被采用。同时,长沙青果园案中被告提出“涉案两软件之间存在部分文件名、数据表名等重复的情况是由于其本人经手原告教材软件开发过程中形成的思维定势和个人偏好所致”,这一抗辩理由实质上指向软件著作权保护对象范围问题,而法院仅以“不符合著作权法中规定的‘著作权合理使用情形’”为由认定该抗辩理由不成立,需要进一步审慎考虑。司法实践中的上述状况反映了我国在软件著作权保护对象范围相关理论不够成熟。美国软件产业发展较早,相应的社会纠纷和司法实践也产生较早。在软件著作权保护范围方面,美国经历了激烈讨论和不断发展,积累了理论和实践经验。梳理和分析美国软件著作权保护对象范围的探索历程,对我国软件著作权保护对象范围的界定具有参考意义。

## 二、美国软件著作权保护对象范围的探索历程

### 美国软件著作权保护对象范围的界定及侵权判

① 广东省高级人民法院(1997)粤知终字第55号。

② 《最高人民法院关于深圳市帝慧科技实业有限公司与连樟文等计算机软件著作权侵权纠纷案的函》[1999]知监字第18号函。

③ 广东省高级人民法院(2003)粤高法审监民再字第59号。

④ 广东省深圳市中级人民法院(2005)深中法民三初字第328-1号。

⑤ 《北京惠斯特科技开发中心诉北京斯惠特科技有限公司等侵犯计算机软件著作权纠纷案》(2007)海民初字第16977号。在该案中,公证下载的被告软件目标程序被进行了加密,只能显示界面。被告既未提供与该软件目标程序相匹配的源代码,又未曾由自己或请求其所称实施加密的第三方将目标程序进行解密。因此法院只能委托鉴定机构对菜单等进行比较。基于由判决书了解的案情来看,本文对裁定被告侵权并无异议,只是认为“菜单定义、数据表对应关系显已超出参数、变量、界面之范围”的表述值得商榷。

⑥ 《长沙青果园科技开发有限公司诉长沙诚光科技开发有限公司等著作权侵权纠纷案》(2005)长中民三初字第147号。

⑦ Whelan Assoc. Inc. v. Jaslow Dental Laboratory Inc., 797 F.2d 1222 (3d Cir. 1986). 本文第二节将对此案做进一步介绍。

定标准并非自始就清晰存在,而是在实践中不断探索、权衡和论证后得到的结果。美国的软件产业发展较早,相应地许多矛盾冲突也出现得较早。20世纪60、70年代起,新兴的软件产业开始快速发展。由于复制软件相比开发软件而言成本极低,软件行业的发展需要法律提供某种形式关于复制的保护。这成为软件进入著作权法保护的最主要理由。在1980年美国正式将软件纳入著作权法保护对象之列以前,美国版权局就已经接受计算机软件的版权注册,也出现了关于侵犯软件著作权的诉讼案件。但当软件被明确写入版权法进行保护之时,法律还未能能在软件中受保护的表达与不受保护的思想之间划出清晰的界限。事实上在1970年,哈佛大学的Stephen Breyer教授就提出,法院难于判断一个程序是否构成了对另一个程序的复制——如果认为细微改动就足以逃避责任,将使著作权法的意义大打折扣;但过宽的保护范围又将限制竞争,或迫使程序员采用非常规的编写方式以降低被指认侵权的风险,这是对社会资源不必要的浪费。<sup>(2)</sup>美国版权作品新技术应用国家委员会(CONTU)在1978年的报告中说明:一定的智力劳动和思想/表达二分法是用来判断程序中受保护部分的原则;著作权法只保护程序的表达,不禁止他人开发功能相同的程序,并且在一种思想仅有有限表达方式之时,该表达亦不受保护;初看起来,在受保护的程序表达形式与不受保护的程序功能之间划分界限很简单,但程序日益丰富的应用会使界限的划分越来越困难,在本报告中尝试建立这样一条界线是徒劳无益的;如果需要进一步制定界限,该界限应该由法官在个案中确定。<sup>(3)</sup><sup>18-23</sup>此后10多年的发展历程也印证了这一点:尽管在1980年时软件已明确成为美国著作权法保护对象,但司法审判中对软件著作权保护范围的更细致的探索从那时起才刚刚开始。

根据软件被纳入著作权法保护的原因,以及著作权法最基本的制度设计——作品的表达受到保护,法官们首先在一系列案件中明确了计算机程序的源代码、二进制代码、微码或固件、应用软件和操作系统软件的文字部分作为软件作品受到著作权法的保护,并且认同程序运行产生的音像输出可以作为视听作品进行考察和保护。<sup>(4)</sup>例如,Stern Electronics v. Kaufman案<sup>①</sup>是关于有用户参与的游戏屏幕显示和声音是否受到著作权法保护的案例。在该案中,法官指出计算机程序的音像输出只要本身具有独创性,满足视听作品的法定要求,就可以作为视听作品进行保护。

在案例中界定了软件著作权保护对象的大体范围之后,法官们遇到的最主要的问题是如何对计算机

软件中不受保护的思想 and 受保护的表达进行更具体的划分。当时已经出现一些案件,其中被控侵权方都接触过原告软件的源代码,并开发了功能类似的软件,虽然并没有直接逐字逐行复制原告软件,但开发出的软件在功能、结构等方面与原告软件相同或相似。在当时,对于文学艺术作品等著作权法保护的傳統对象,美国著作权法在长期实践中已经形成了比较稳定的确定保护范围的理论 and 通识;而对于软件作品,这些基本原则具体如何适用 and 解释,美国法院展开了分析和探索。

20世纪80年代中后期,美国司法界关于软件著作权保护对象范围仍有不同意见。1986年,第三巡回上诉法院在Whelan Assoc. Inc. v. Jaslow Dental Laboratory Inc.一案<sup>②</sup>判决中给出如下意见:1. 肯定了软件著作权保护可以超出文字部分,保护程序的结构、顺序和组织(Structure, Sequence and Organization,简称SSO);2. 针对程序结构是否属于或在何种情况下属于思想这一问题,提出一种软件思想与表达的划分原则:实用作品的目的或功能就是其思想,而为达到该思想或目的所必要的一切就是思想的表达;当有多种方式可以达到目的时,所选择的某种非实现目的所必需的特定方式就是表达而非思想。判决在论证该划分原则时提出的理由可总结如下:思想与表达之间的界限是模糊的,因此在确定划分软件中思想与表达之界限的原则时,必须特别注意在思想/表达二分法和著作权法中蕴含的实用主义考虑。在这点上,必须牢记著作权法的目的是在保护(激励)和信息传播之间取得最有效率、最有生产力的平衡,以促进学习、文化和发展。法院认为判决中提出的这一划分标准有利于实现思想/表达二分法的基本目的,即“保持竞争与专利法和著作权法体现的保护之间的平衡”。这可以从两个角度来分析:1. 计算机程序编制中花费较大、比较困难的是程序的结构和逻辑开发、程序调试等部分,而编码过程是相对简单的部分。这一判断标准可以通过保护程序员最有价值的努力,从而为其提供适当的激励;2. 如果相关主题没有其它表达方法则不受保护,因此该判断标准不会抑制实现同一目的的新软件的开发。

然而在1987年,第五巡回上诉法院却在Plains Cotton Coop. Assoc. v. Goodpasture Computer Serv.,

<sup>①</sup> Stern Electronics, Inc. v. Kaufman, 669 F.2d 852 (2d Cir. 1982).

<sup>②</sup> Whelan Assoc. Inc. v. Jaslow Dental Laboratory Inc., 797 F.2d 1222 (3d Cir. 1986).

Inc. 一案<sup>①</sup>中表示在本案他们将不采用 Whelan 案的意见,而援引了一个早期案件 Synercom Technology, Inc. v. University Computing Co.<sup>②</sup>的判决,认为:1. 程序的输入格式属于思想而非表达;2. 有证据显示两款软件相似处都是由棉花市场的外部属性决定的,正如 Synercom 一案判决中所述,“软件中顺序与形式的可著作权性与其它作品不同”,在此处这些相似处可以视为思想。

在上述两个案件中,被控侵权方都接触过原告软件的源代码,并开发了功能类似的软件。Whelan 案中,涉案软件是两个用于牙科实验室的软件。被控侵权方用另一种编程语言重新编写了与原告软件功能相同的软件。Plains 案中,涉案软件用于查询棉花市场的信息。原告的软件运行在大型服务器上,用户在终端通过电话线接入服务器进行查询,而被控侵权方的软件是用在个人电脑上的。这两个相似案件的判决论述中的不同考虑显示了这一时期软件著作权保护对象范围判断体系的不完善性。Whelan 案的判决中从“著作权法的目的是在保护(激励)和信息传播之间取得最有效率、最有生产力的平衡,以促进学习、文化和发展”出发,认定软件的保护可以在一定程度上延至程序的结构等要素,并且如果表达与思想合并则不受著作权法保护,这一部分的论证思路及其结论得到了广泛的支持。但是,Whelan 案认为实用作品在著作权法意义上的“思想”只有一个,即其中心目的或功能,其它的一切只要不是表达该目的或功能的唯一方式就属于表达,这一观点受到了较多质疑。一些评论认为,这一论断缺乏对软件中实际情况及利益关系的深入考量,担心在这一原则下,对软件著作权保护范围将过于宽泛而可能损害了合理的竞争。Plains 案裁定程序的输入格式以及由外部属性决定的顺序与形式不受著作权法保护,但未进行详细论述,在判决中援引了 Synercom 案中“软件中顺序与形式的可著作权性与其它作品不同”这一比较笼统的判断,似乎有将对软件著作权的保护紧缩向文字部分的倾向,因此也受到了评论人士的批评。

1992 年,第二巡回上诉法院在 Computer Associates v. Altai 一案<sup>③</sup>的判决中提出了“抽象-过滤-比较”三步骤判断方法,对后来的相关理论总结和司法审判产生了重要影响。该方法将涉案软件的实质性相似比较分为三个步骤:1. 抽象:1930 年 Learned Hand 法官在 Nichols v. Universal Pictures Co. 一案判决中阐释了区分作品中思想与表达的“抽象”法<sup>④</sup>。该方法最初主要用于文学作品领域。文学作品的文字部分属于具体表达。Hand 法官解释道“随着情节

被剥离得越多,会有许多更加概括的模式与之相符。最后可能只剩下对作品最概括的陈述,有时可能只包括它的题名。但在这一系列抽象中,存在一个不再受保护的临界点,否则作者可以阻止他人使用其‘思想’,而他的财产权永远不能扩展到对表达以外的思想的保护。”在此,法院将其应用在软件领域。程序由一条条语句或指令组成。在最低层次的抽象上,语句被抽象成若干模块;这些模块由可以被抽象成更高层次的模块。这样逐层向上抽象。最后一层抽象的结果就是软件的最终功能。这一步骤完成之后,程序在不同抽象层次上的结构就更加清晰地呈现出来,以便于寻找该程序中思想与表达的界限以及对属于表达的软件结构进行相似性比较。2. 过滤:这一步骤中,该方法采用了 Nimmer 教授建议的“连续过滤法”,用来将受保护的表达与不受保护的部分区分开来。具体而言,该步骤在每个抽象层次上,审查结构成分来确定其是否属于下列:a) 思想;b) 基于效率考虑,表达该思想所必要的;c) 由程序外部因素所要求的(例如硬件和软件规范的要求、使用该软件行业的要求等);d) 来自公有领域因而不受保护的表达(例如计算机软件行业通用的表达形式等)。在除去这些不受著作权法保护的部分之后,余下部分就是可以保护的表达。3. 比较:这一步骤在可受保护的表达(包括非文字部分)中进行比较,以判断是否构成实质性相似。

“抽象-过滤-比较”判断法的提出有两个主要背景:一是在 20 世纪 80 年代中期到 90 年代初计算机软件侵权判定的司法实践经验积累和学术理论总结;二是 1991 年美国联邦最高法院在 Feist v. Rural 案<sup>⑤</sup>中对美国版权法中关于受著作权法保护对象的“独创性”要件的着重阐释。Whelan 案之后,许多法学评论和案件判决中对软件著作权保护范围进行了讨论。Altai 案判决并不是简单地推翻了 Whelan 案的方法,而是批判地继承了 Whelan 案中对于著作权法保护计算机软件结构的理论,并综合了当时其它案例以及来自学术界的意见,特别是 Nimmer 教授建议

① Plains Cotton Coop. Assoc. v. Goodpasture Computer Serv., Inc., 807 F.2d 1256 (5th Cir. 1987).

② Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003 (N. D. Tex. 1978).

③ Computer Associates International v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992).

④ See Nichols v. Universal Pictures Co., 45 F.2d 119 (2nd Cir. 1930).

⑤ Feist Publications, Inc. v. Rural Telephone Service Co., Inc., 499 U.S. 340 (1991).

的“连续过滤法”将当时比较明确的不应受著作权保护的部分总结归纳起来。可以看到,Altai案判决与许多法律评论一致,肯定了Whelan案中强调的著作权法基本规定和原则在软件领域适用时的如下考虑:著作权保护可以超越文字部分而扩展到程序的结构;软件中思想/表达的界限划分应着重考虑激励作者与保护竞争之间的平衡;思想/表达合并原则和必要场景原则对软件作品同样适用。判决中对Whelan案方法的主要批评在于两个方面:1. Whelan案的判决中认为,在著作权法意义下,软件的思想只有一个,即软件的中心目的或功能。第二巡回上诉法院认为,从软件的结构设计的实际情况来看,一个程序通常由若干互相交互的子程序构成;每个子程序也可以看成一个软件,也可以说具有自己的中心目的或功能,即具有自己的“思想”。应该说,相比Whelan案对思想/表达的划分,Altai案方法表示软件中可能不仅仅包含最终功能这唯一的思想,思想/表达合并原则的作用也相应增大,著作权保护的範圍有所缩减。2. Whelan案强调计算机程序编制中花费较大、比较困难的是程序的结构和逻辑开发、程序调试等部分,而编码过程是相对简单的部分。第二巡回上诉法院则认为,在Feist案中,最高法院一再强调,“著作权的主要目的不是奖励作者的劳动”,因此仅是付出了大量劳动并不应使原本不能取得著作权的作品获得著作权,尽管大量的劳动和费用投入到了计算机程序流程图开发和调试中,但并不总能带来可受保护的表达。那些在思想、操作方法以及由外部因素等决定的程序中的劳动和费用投入并不能受到著作权法保护。

“抽象-过滤-比较”三步骤判断法提出后很快被美国多数法院采纳,并得到了加拿大<sup>①</sup>、英国<sup>②</sup>等多国法院的赞同。该方法不仅提供了一种侵权判定方法,更重要的是,它在软件著作权保护范围还比较模糊并充满争议的当时,总结了多方意见,勾勒出了更加清晰的软件著作权保护范围的界限。“过滤”这一步骤旨在排除软件中不受保护的部分,使保护不会过宽,尽量趋近于著作权法理论上应当保护的上限。而在其之前的“抽象”步骤则能够使不同层次的程序结构更清晰地进入法官判断的视野。是否真正进行逐层抽象并不是最重要的。正如Lemley教授所发表的评论意见所述:(当时)大部分争议“曲解并且不必要地复杂化了抽象这一步骤。问题在于律师和法院过度局限于抽象步骤的字面含义……在文学领域我们都知道抽象层次只是帮助我们做判断的人工构建对象……唯一重要的问题是复制行为被指控发生在哪个层次。一旦该抽象层次被确定,法院就可以继续判

断享有著作权的作品在该层次受到怎样程度的保护。”<sup>④</sup>如果无论程序大小和案情复杂程度都首先对程序整体进行分析和逐层抽象,有时并不必要或者可能耗费大量时间。有法院建议,可以由原告来指出其认为受到著作权法保护但被侵权的部分<sup>③</sup>。从实际意义上来看,“抽象”这一步骤能够使程序结构更清晰地进入判断的视野,尽可能使保护范围不低于理论上的下限——即保护范围至少应包含一定程度的程序详细结构。尽管我们仍然难以在普遍意义上构建出软件著作权保护范围的精确上限和下限,但这一方法有效地综合了现有的公认结论,对于避免保护过窄或过宽有非常积极的意义。

在相同的寻求最佳利益平衡点的原则之下,1995年,第一巡回上诉法院在Lotus Development Corp. v. Borland International一案中,裁定菜单命令层次结构属于“操作方法”,即属于思想,因而不受著作权法的保护。Boudin法官在赞同法院判决的同时提出了自己的附和意见,认为如果赋予Lotus关于菜单命令层次结构的垄断权,用户将因为已经付出的学习使用和编制宏命令的成本而继续选择Lotus;Lotus已经凭借过去的产品质量获得了奖励,如果Borland制造出了更好的产品,应该使用户可以自由选择,从而激励产品的进步,这在很大程度上需要Borland的产品可以兼容Lotus产品的菜单;因而“问题不是Borland是否应该胜诉,而是根据什么理由胜诉”。这体现了其面对此案首先进行价值判断再寻找法律依据的思考方法。一些法律评论在赞同该案关于实现相同的菜单命令层次结构不侵权之结论的同时,对其关于“菜单属于操作方法”的论证过程有所质疑。美国联邦最高法院准予了该案复审调卷令,最终以四比四持平的匿名投票维持了第一巡回上诉法院的判决,但未形成先例意见<sup>④</sup>。此案一方面体现了在软件中区分思想与表达依旧存在困难,另一方面,实际上为美国后续案件中裁定菜单命令、程序外部接口不受著作权法保

① 例如Delrina Corp. v. Triolet Systems Inc., 47 C. P. R. (3d) 1 (Ont. Gen. Div., 1993)。该案中采用了“抽象-过滤-比较”三步骤判断法,并被评述为“在加拿大版权法中建立了‘合并原则’”。参见[http://en.wikipedia.org/wiki/Delrina\\_Corp.\\_v\\_Triolet\\_Systems\\_Inc](http://en.wikipedia.org/wiki/Delrina_Corp._v_Triolet_Systems_Inc)。

② John Richardson Computers v. Flanders & Chemrec Ltd., (1993) FSR 497. See V. K. Unni, “Cases and Materials on Intellectual Property Law”, [http://www.nalsarpro.org/PDFS/IPR\\_Caselaws.pdf](http://www.nalsarpro.org/PDFS/IPR_Caselaws.pdf), p. 19-20.

③ MiTek Holdings Inc. v. Arce Engineering Co., 89 F. 3d 1548 (11th Cir. 1996)。

④ Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F. 3d 807 (1st Cir. 1995), aff'd per curiam, 516 U. S. 233 (1996)。Stevens大法官回避了此案。

护提供了依据。例如,2012年判决的 Oracle America, Inc. v. Google Inc. 一案<sup>①</sup>中,地区法院的 Alsup 法官认为 Java 类库和方法的命名以及组织构成了“一个系统的命令结构或者其应用程序编程接口(API)的操作方法”,因此不受著作权法保护。但其似乎也担心这一理由不足够使人信服,因此在论述了类库和方法的命名及组织属于操作方法因而不受保护之后,还补充说明“从兼容性的要求来看,这些代码不受著作权法保护就更加清晰了”,从这一角度对自己的结论进行补充论述。从 Lotus v. Borland 案中 Boudin 法官的附和意见,到 Alsup 法官在判决中的论述,我们可以看到,对于这些关于软件著作权保护范围的争议问题,两位法官首先根据著作权法的立法目的和利益平衡原则,认同对于实现计算机系统之间兼容性的菜单和命令层次结构或应用程序编程接口不应受到著作权法保护<sup>②</sup>,在此判断之下,再从著作权法律制度中寻找最合适的依据。这体现了明显的目的法学和利益法学的思考方式。

从知识产权法律制度的整体来看,美国软件可专利性与软件著作权保护范围界定的司法实践有着相互影响。在 Altai 一案判决的最后提出,著作权对任何作品一视同仁;有新颖性和非显而易见性要求的专利权“也许更适合于保护这种类型的知识产权”。在此之前,美国联邦最高法院 1981 年在 Diamond v. Diehr 案<sup>③</sup>中已指出,不能仅因为产品或方法涉及计算机软件就认为其不具有可专利性。此后软件可专利呼声日益高涨。在 1994 年的 Alappat 一案中,美国最高法院明确表示“根据专利法条文,没有理由支持算法是不可授予专利的主张”;该发明“将抽象概念(即数学算法、公式或者计算)进行了实际应用,产生了‘一个有用、具体且有形的结果’”,是可以获得专利的对象<sup>④</sup>。自 90 年代中后期开始,美国的软件专利年申请及授权量均呈指数级增长。<sup>(5)15</sup>与之相应的是,软件著作权保护范围方面,法院逐渐重视贯彻落实著作权法不保护思想这一原则。

通过对美国软件著作权保护范围相关司法实践的简要回顾,可以看到软件著作权保护对象范围经历了从模糊到宽泛再到合理缩减的过程,司法中对程序的运行参数、菜单命令组织结构、应用程序编程接口、程序中由外部因素决定的表达、程序中包含的公有领域代码等不受著作权法保护逐渐有了比较统一的认识。正如拉伦茨所说“大部分的法律都是经过不断的司法裁判过程才具体化,才获得最后清晰的形象,然后才能适用于个案。”<sup>(6)20</sup>这一系列司法审判逐渐建立起了比较清晰的软件著作权保护边界,在这一过

程中,立法目的条款及其蕴含的利益平衡原则在划分软件中思想与表达的界限时起到了重要指引作用。美国软件产业的发展成就全世界有目共睹,美国立法和司法实践为在鼓励竞争与提供保护激励之间寻求最有效的平衡而付出的努力,也应该得到认同和借鉴。

### 三、美国软件著作权保护对象范围相关司法实践对我国的借鉴意义

美国司法实践中所提出的上述软件著作权侵权判定方法(“抽象-过滤-比较”判断法),以及关于程序运行参数、菜单命令组织结构、行业通用的表达形式等不受著作权法保护的结论可供我国参考。更重要的是司法中对于法律中未进行明晰规定之处,可以借鉴美国重要判例中体现的法律方法,首先是明确软件著作权设置的目的与价值取向,继之深入考察各方利益及冲突关系,在此基础上依据著作权法的立法目的与价值取向对所体察的利益冲突进行司法调整。

(一) 保护范围的界定需符合著作权法的立法目的

通过美国及我国司法实践中的诸多案例可以看到,思想与表达的区分是困难的。Learned Hand 法官在 Nichols v. Universal Pictures Corp. 案判决中曾写道“没有人曾解决了思想与表达的界限问题,未来也不会有。”诚然,思想/表达之界限划分,如果从哲学层面解释很难得到结论。然而我们在这里遇到的不是一个哲学层面的问题,而是一个法律上的问题。法律是用来调节利益关系的。思想/表达二分法的提出,本来也是依据法律上调节利益冲突的需要,以保护“表达”来实现对作品创造和传播的激励,以不保护“思想”来强调和保护由思想的传播和使用带来的社会公共利益。在遇到思想与表达的界限难于判断之时,司法者仍然需要做出裁决,而界限如何划分更有利于实现著作权法的立法目的就成为了论证的重

<sup>①</sup> Oracle America, Inc. v. Google Inc., 2012 U. S. Dist. LEXIS 75896. Alsup 法官似乎担心人们对这一结论质疑。他在论述了属于操作方法因而不受保护之后,还补充说明从兼容性的要求来看,这些代码不受著作权法保护就更加清晰了。

<sup>②</sup> 关于软件接口相关代码的可著作权性问题,可参见张吉豫:《软件接口代码可著作权性研究——兼评〈著作权法〉第三次修改草案“反向工程条款”》,《吉林大学社会科学学报》2013年第1期,第87-92页。

<sup>③</sup> Diamond v. Diehr, 450 U. S. 175 (1981). 涉案发明是一种合成橡胶的方法,其中使用了计算机软件来接收元件自动检测的温度并根据一个公式来计算橡胶制品在硫化过程中的正硫化时间(合适的加工时间),以此判断是否应终止硫化。美国联邦最高法院裁定该发明可以被授予专利。

<sup>④</sup> In re Alappat, 33 F. 3d 1526 (Fed. Cir. 1994).

要依据,也成为了所需考虑的焦点。

著作权法的正当性基础有着多种哲学解释,目前最为人们所广泛接受的是著作权法通过激励作者创作和传播作品,以促进文化科学的传播和发展,从而促进社会公共利益的功利主义观点。美国著作权法的功利主义性质十分明显。美国宪法授权国会“为促进科学和有益技术的进步,可授予作者和发明人在有限期限内对其作品和发明所享有的专有权”。这奠定了美国著作权法立法和法律解释的基础,即著作权法的根本目的在于公共利益,作者所拥有的关于作品的有限专有权是实现这一目的的途径。因此,美国著作权法中尽管也体现了其它哲学观点(如自然权利理论),也可以被视为主旨在于实现一种最佳的平衡:既可以激励作品创作,又能够最大程度地使用和传播这些作品;<sup>(7)419</sup>而这一最佳平衡点应选取在哪里,则取决于从长远、整体来看,如何能够更有利于促进文化科学进步的公共利益。在美国司法实践中,宪法及其所确定的这一利益平衡原则被反复引用,也成为了裁定软件著作权保护范围的基础。

我国《著作权法》第1条为立法目的条款,说明了“保护著作权”、“鼓励作品创作和传播”、“促进文化和科学事业的发展”三项目的。这三项目在条文中并列存在,并没有表明其间的关系。《计算机软件保护条例》的立法目的条款同样采用的是并列列举的方式。这可能会使支持不同观点的人们在解释和适用法律时产生分歧。例如,自然权利学说的支持者更可能倾向于为作品提供较宽的著作权保护;功利主义的支持者会倾向于认为赋予作者的权利不宜超过使其从事创作所必要的激励限度。我国司法中显现出对著作权法保护范围存在一定分歧。对立法目的进行更加清晰的阐释可以更好地指导法律的解释和适用。我国有学者从解释论的角度论证了中国立法保护智力成果,并非出于对所谓自然权学说的默认规则的尊重,而是出于一系列功利主义的考虑。<sup>(8)</sup>我国《计算机软件保护条例》第一次修改期间,当时国家领导人提出“在切实保护知识产权的同时,按照市场规律,使知识产权的保护范围、保护期限和保护方式,有利于科技知识的扩散和传播,有利于各国共享科技进步带来的利益”。<sup>(9)</sup>这也印证了国家倡导的立法原则是以功利主义为基础的。因而在我国立法和司法实践中,应以此为基准确定软件著作权保护对象范围,对相关法律概念和法律规范进行解释,及在必要时对法律进行补充规定。

(二) 保护范围的界定需考量技术领域的利益平衡关系

为做出符合上述立法目的的裁决和解释,需要对具体问题所涉及的在先软件开发者、后继软件开发者及公众等多方的利益进行审慎考察。计算机软件属于功能性的技术产品。这不仅是一个单纯的分类,更是提醒我们在著作权法框架下对软件领域进行利益考量和价值判断之时,需要特别注意作为功能性的技术产品的软件与文学艺术作品的区别。

一方面,软件作为功能性的技术产品,其创作更多的时候属于一种工程开发,所产生的表达受到产品功能需求、产品所在领域的客观约束、产品所应用的软硬件环境、用户体验、产品性能、工程经验和惯例等多方面约束。计算机业存在大量广为接受的常规的编程方式,也存在许多代码自动生成工具<sup>①</sup>,<sup>(10)</sup>教师们通过课程、培训影响着大量的程序员,鼓励他们采用成熟的、高效的、不易出现错误的以及容易被理解的方式编写代码,从而开发出高效率、低错误且易于扩展和维护的软件产品。如果将这些由外部约束或常规编程模式所决定的代码包含在著作权法保护范围之内,将严重束缚后来的软件开发者,限制软件产业的发展。在进行实质相似性判断时,可依据思想/表达二分法、思想表达的合并原则或不符合独创性要求等原理加以解释,将相应代码排除在保护范围之外。

另一方面,在技术领域,产品间的竞争具有非常重要价值,是开发者提高产品质量、改善经营水平、降低成本和产品价格的重要动力。社会愿意鼓励计算机软件产品间的公平竞争。然而计算机软件市场具有明显的网络效应(network effects)。网络效应指一个产品对于消费者的价值随着使用该产品的用户人数增长而增长。<sup>(11)</sup>数据格式、传输协议等标准体现了直接的网络效应。平台软件则展示了间接的网络效应:一个操作系统的价值随着可在其上运行的软件数量的增加而增加;编程人员并不希望为没有几个用户使用的操作系统来编写应用程序,因此,随着操作系统被越来越广泛的使用,会有越来越多的可以在该操作系统上运行的软件被编写;而更丰富多样的应用软件相应地也会带来更多的用户。网络效应虽然使得许多企业开放产品兼容性信息,也使许多企业希望凭借对产品兼容性的控制,使自身在市场竞争中获得更大的优势。出于对促进竞争和消费者利益的考虑,美国案例中将菜单命令组织结构、应用程序编程接口排除在

<sup>①</sup> 可参见:参考文献(10)第319-321页。其中展示了用户编写不同程序时,使用微软 Visual Studio 开发环境创建标准 Windows 表格时 Visual Studio 自动生成的两份源代码几乎完全相同。

软件著作权保护范围之外。我国司法实践中可以依据思想表达的合并原则,将使用类似的菜单组织和编程接口解释为实现兼容性要求所必要的表达。

司法实践中在确认原告对作品拥有著作权之后,通常依照“接触+实质性相似”的判断框架,考察被控侵权方是否可能接触过原告作品,以及被控侵权方作品中存在多少与原告作品中受著作权保护的部分构成实质性相似之处。基于上述软件作品的特点和所涉利益关系与文学艺术领域的差异,在软件侵权判定的过程中进行实质性相似判断时,法院和鉴定机构需要特别意识到软件中可能包含多种不受著作权法保护的部分。由此,需要格外注意不能简单因为证明了存在相似就认定构成侵权,而是要确认相似之处是否属于受到著作权法保护的部分,尤其应对被控侵权方提出的原告作品中不受著作权法保护的部分和理由进行审慎考察。具体而言,我国司法实践中可以借鉴前述美国 Altai 案中提出的“抽象-过滤-检验”三步骤判断方法中所体现的对软件著作权保护范围的界定及其理由。一方面,应认定在一定程度上程序的结构也可以属于“表达”,对软件著作权的保护不能只停留在逐字逐句的文字层面,否则后来者可以通过简单替换就得到不侵权的软件,成本极低,这将使在先开发者无法与后来者进行竞争和获利,从而失去

继续从事软件开发的必要激励;另一方面,应将软件中的思想、由效率或外部约束决定的代码以及公有领域的代码等排除到保护范围之外,以避免使著作权人在事实上垄断了相关思想技术,避免不必要的对竞争的限制以及避免对公有领域的侵蚀。

#### 四、结语

美国软件著作权保护范围的探索历程对我国立法和司法实践具有借鉴意义。在软件著作权保护对象范围模糊之处,应以促进技术和产业发展从而促进公共利益为目标,注意到软件与文学艺术作品的不同,仔细考察保护范围的不同界定对于著作权人的激励以及对市场竞争和公众利益的影响,寻找最能够促进这一根本目的的各方利益的平衡点。在这一原则下进行思想与表达的划分,才是与软件著作权设置目的相符的划分方式。在我国司法鉴定及审判中,特别需要明确软件中可能存在多种不受著作权法保护的代码片段,在进行实质相似性判断时应予以过滤。同时,本文建议我国司法判决中对软件相似性鉴定情况进行更详细的介绍,并对被告的相关抗辩是否采信予以更加具体的论述,以形成更好的案例交流和讨论平台,推动我国软件著作权保护对象范围相关理论和法律制度的发展和完善。

#### 参 考 文 献

- (1) 陈雪. 软件作品的著作权司法鉴定——以“公安基层管理系统”著作权纠纷案为例 (J). 华南师范大学学报(社会科学版), 2005, (3): 24 - 30.
- (2) Stephen Breyer. The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, and Computer Programs (J). Harvard Law Review, 1970, (2): 281 - 351.
- (3) Final Report of the National Commission on New Technological Uses of Copyrighted Works (R). U. S. Government Printing Office. July 31, 1978.
- (4) Mark A. Lemley. Convergence in the Law of Software Copyright? (J). High Technology Law Journal, 1995, (1): 1 - 34.
- (5) James Bessen. A Generation of Software Patents (R). Working Paper No. 11 - 31, Boston University School of Law, June 21, 2011. Available at: <http://www.bu.edu/law/faculty/scholarship/workingpapers/Bessen-Generation-Software-Patents.html>.
- (6) [德] 卡尔·拉伦茨. 法学方法论 (M). 陈爱娥, 译. 北京: 商务印书馆, 2003.
- (7) Robert P. Merges, Peter S. Menell and Mark A. Lemley. Intellectual Property in the New Technological Age (M). 5th Ed. New York, USA: Aspen Publishers, 2010.
- (8) 崔国斌. 知识产权法官造法批判 (J). 中国法学, 2006, (1): 144 - 164.
- (9) 寿步. 中国计算机软件著作权保护的回顾与展望 (J). 暨南学报(哲学社会科学版), 2010, (6): 1 - 8.
- (10) Robert Zeidman. The Software IP Detective's Handbook: Measurement, Comparison, and Infringement Detection (M). Boston, U. S. A.: Pearson Education, Inc., 2011.
- (11) Mark A. Lemley and David McGowan. Legal Implications of Network Economic Effects (J). California Law Review, 1998, (3): 479 - 611.

(本文责任编辑 焦和平)