

开源许可证相关知识与相容性定义

开源逐渐成为全球信息技术创新和软件产品创新的重要模式，基于开源组件的开发方式也越来越流行，通过这种方式可以极大地缩短软件开发周期，节约开发成本。开源组件中通常包含一种或多种开源许可证，用于描述该开源组件的使用者需遵循的条款和规定。因此，合法进行软件重用和修改的前提是对开源许可证进行相容性分析以及条款风险分析，可以有效避免因侵犯知识产权而带来法律上的风险。

1 开源许可证相关知识

1.1 开源许可证定义

软件许可证是一种具有法律性质的合同或指导，目的在于规范受著作权保护的软件的使用或散布行为。通常的授权形式允许用户使用该软件的一份或多份副本，如果因为未经授权而擅自使用软件，将违反版权法给予软件开发人员的专属保护。从效用上来说，软件许可证是软件开发人员与其用户之间的一份合约，用来保证在匹配授权范围的情况下，用户将不会受到控告。

开源许可证是软件许可证的一种特殊形式，其授权一般是针对软件的开放源代码。它是一个软件组件的作者和用户之间合法的并且有约束力的合同，声明该软件可以在特定条件下用于不同的目的。同时它也是将代码转换为开源组件的工具，如果没有开源许可证，该软件组件是不能被其他人所使用的。开源许可证允许用户在承认软件原作的著作权的基础上对软件源代码的使用、修改、私用等权益，还规定了用户的义务和限制，以保证开源软件能够合法地被众多软件开发者优先共享和使用。

目前，全球范围内通过 OSI（Open Source Initiative）认证的开源许可证已上百余种。尽管大多数开源许可证的出发点都是一致的，即允许用户免费使用、修改软件，但在具体细节上差别很大。例如，使用 GPL 许可证的代码二次分发时必须提供源代码，否则属于违约行为，使用者对开源软件的复制、修改、再发布就会造成侵权。而 BSD 和 MIT 等宽松型许可证则没有上述要求，允许使用者对再发布软件时闭源，但其并未明确声明专利授权条款，用以授权用户使用软件所包含的相关专利，开源使用者很可能被开源贡献者提起专利诉讼并收取专利许可费。

1.2 开源许可证主要条款

在法律上，一般采用知识产权法对软件进行保护，开源软件许可证在法律上通常被认为是合同，一般都遵照《合同法》进行判别，即遵照合同本身声明的权益进行判别。

Choosealicense 网站上列举的 13 项条款覆盖了绝大多数许可证声明的内容，如表 1.1 所示：

表 1.1 开源许可证条款及描述

序号	条款类型	条款	描述
1	权利 (Permissions)	商用 (Commercial use)	该许可是否允许开发人员将该软件及其衍生版本用于商业目的
2		分发 (Distribution)	该许可是否允许开发人员将该软件或代码分发
3		修改 (Modification)	该许可是否允许开发人员修改代码
4		专利授权 (Patent use)	该许可是否提供了来自贡献者的明确专利权，是否允许工程项目进行专利申请
5		私用 (Private use)	该许可是否允许开发人员将该软件私下修改和使用
6	义务 (Obligations)	公开源码 (Disclose source)	该许可是否要求开发人员分发软件时必须 是源代码可用，即开源
7		放置协议与版权信息 (License and copyright notice)	该许可是否要求该软件必须附加许可证和 版权声明的副本
8		使用网络分发 (Network/SaaS)	该许可是否要求通过网络/云服务与软件 交互必须使源代码可用，即开源
9		使用相同协议 (Same license)	该许可是否要求开发人员分发软件时必须 以相同的许可证发布修改后的软件
10		声明变更 (State changes)	该许可证是否要求记录对代码所做的更改
11	限制 (Limitations)	使用商标 (Trademark use)	该许可是否允许开发人员使用商标
12		不承担责任 (Exclude liability)	该许可是否允许开发人员不承担责任
13		提供担保 (Provide warranty)	该许可是否向开发人员提供保证

笔者认为在分析许可证的文本内容时还应考虑到链接（Link）和子证书授予（Sublicense）这两项条款声明的内容。

链接（Link）：该许可是否允许在开发人员在编程的过程中链接受其它开源协议约束的库文件。

子证书授予（Sublicense）：该许可是否允许开发人员在源码中引入其它的开源协议，或者是在该协议的基础上添加额外条例

1.3 SPDX-License-Identifier

License A **OR** License B:

使用该开源项目必须遵循 License A 或 License B。

License A **AND** License B:

使用该开源项目必须同时遵循 License A 和 License B。

License A **WITH** License B:

使用该开源项目必须遵循 License A，License B 为该项目提供了额外的附加条款。

1.4 开源许可证分类

现有开源许可证都有的共同特点是，都允许用户免费地使用、修改、共享源码，但是都有各自的使用条件。根据使用条件的不同可以将开源许可证分为两个类别：

（1）Permissive（宽松型）许可证：

Permissive（宽松）许可证：对用户没有使用限制，不提供担保，要求披露原始作者。常见的许可证有 BSD、MIT、Apache-2.0，它们都允许用户任意使用代码，区别在于要求用户遵守的条件不同。

（2）Copyleft（著佐权型）许可证：

Copyleft 是理查德·斯托曼发明的一个词，作为 Copyright（版权）的反义词。Copyleft 许可证相比 Permissive 许可证有更高的限制，核心在于 Copyleft 许可证下的项目必须开源，且项目的修改和拓展版本也必须开源。我们根据代码的修改版本再发布时是否需要以相同许可协议发布，将 Copyleft 许可协议分为强 Copyleft 许可协议（Strong Copyleft）、弱 Copyleft 许可协议（Weak Copyleft）以及更弱的 Copyleft 许可协议（Weaker Copyleft）。

强 Copyleft 许可协议：无论以何种方式使用开源代码，强 Copyleft 许可协议明确声明修改版本须以同一许可协议发布，如 GPL-3.0-only、EURL-1.2、AGPL-3.0-only 等。

弱 Copyleft 许可协议：以使用源代码的方式时，弱 Copyleft 许可协议明确声明修改版本须以同一许可协议发布；以使用库方式时，可以在其他开源许可协议下发布。如 MPL-2.0、EPL-2.0、LGPL-3.0-only 等。

更弱的 Copyleft 许可协议：无论以何种方式使用开源代码，均可以在其他许可协议下发布。

1.5 Copyleft 许可证传染性

Copyleft 许可协议通常具有一定的传染性，笔者认为不同 Copyleft 许可协议的传染形式有所不同。当一个使用 Copyleft 许可协议的项目合并新的开源组件或代码时，新并入的开源组件或代码需要遵循该许可协议的各项条款，即该许可协议对新并入的开源组件和代码发生了“传染”，我们称这种传染方式为**自上而下的传染**；当一个项目包含了一个 Copyleft 许可协议下的部分代码时，完全发布时必须作为整体适用该许可协议，即该许可协议对整个项目的代码进行了传染，我们称这种传染方式为**自下而上的传染**。除 GPL 许可协议之外，其他大部分 Copyleft 许可协议仅具有自上而下的传染性，它们会对使用该许可协议项目下的其他开源组件或代码进行传染。GPL 许可协议区别于其他 Copyleft 许可协议，具有自上而下以及自下而上两种传染性，只要项目中包含 GPL 许可协议下的部分代码，则整个项目的代码都会被 GPL 所传染。

当 Copyleft 许可协议的传染性发生时，我们认为被传染的开源组件或代码受到的是双许可协议的限制，双许可协议的关系为 **AND**。

2 开源许可证相容性

2.1 相容性定义

适用不同许可证的两个开源程序合并成一个较大的程序，或者把其中之一的代码合并入另一个时，如果各个许可证的限制或条件产生了冲突，即两个许可证的义务或限制无法同时满足，则该种合并不被允许，我们称这些许可证是不相容的。反之，若两个许可证的义务和限制能够同时满足，则该种合并是被允许的，我们称这些许可证是相容的。

2.2 开源许可证相容性通用判定方法

在未考虑各开源许可证中有关分发和相容性的独特性要求的情形下，我们可以根据开源组件使用方式、开源许可证的类别以及 Copyleft 强弱，对不同开源许可证的相容性进行初步判断。许可证中声明的公开源码（Disclose Source）、使用

相同协议(Same license)这两项义务,是判断许可证是否为 Copyleft 以及 Copyleft 强弱的主要依据。

开源组件的使用方式可以分为使用源代码方式和使用库方式:使用源代码方式指的是从要组合的代码中取出整体/部分代码,修改或不修改都可以,然后把它添加到你的代码中构成一个作品;使用库方式指的是在编译或运行时通过链接、导入或其他典型的机制(例如静态与动态链接)把要组合的开源代码绑定在一起。

判断不同许可证之间相容与否的一般性规则如下,相容性关系如图所示:

License Category		I want to license my code under:			
		Strong Copyleft	Weak Copyleft	Weaker Copyleft	Permissive
I want to copy code under:	Strong Copyleft				
	Weak Copyleft				
	Weaker Copyleft				
	Permissive				

图 2.1 使用源代码方式

License Category		I want to license my code under:			
		Strong Copyleft	Weak Copyleft	Weaker Copyleft	Permissive
I want to use a library under:	Strong Copyleft				
	Weak Copyleft				
	Weaker Copyleft				
	Permissive				

图 2.2 使用库方式

注：图中红色区域代表冲突，即不相容；绿色区域代表无冲突，即相容。

2.2.1 Permissive 许可协议与其他类型许可协议相容性

Permissive 许可协议之间、Copyleft 与 Permissive 许可协议之间通常是相容的。

2.2.2 Copyleft 许可协议相容性

Copyleft 许可协议之间的相容性判定更为复杂，以不同方式引入开源组件时，相容性也有所不同。

（1）使用源代码方式：

强 Copyleft 与强 Copyleft、强 Copyleft 与弱 Copyleft、弱 Copyleft 与弱 Copyleft 许可协议之间通常是通常是不相容的。主要原因在于强 Copyleft 许可协议和弱 Copyleft 许可协议中均包含 Same license 义务，即要求代码再分发时以相同许可协议发布。当二者以某种方式组合后，因无法同时满足二者的 Same License 义务，无论以何种许可证再发布，都违背了另一许可证的要求，所以这种情况下许可证之间大多是不相容的。

（2）使用库方式：

强 Copyleft 与强 Copyleft 许可协议之间仍不相容。在弱 Copyleft 许可协议下的项目中使用强 Copyleft 许可协议下的库时也是不相容的，而在强 Copyleft 许可协议下的项目中使用弱 Copyleft 许可协议下的库时是相容的，此时组合后的作品整体遵循强 Copyleft 许可协议，使用的弱 Copyleft 许可协议下的库需同时遵循原弱 Copyleft 许可协议和该项目强 Copyleft 许可协议。

在两种使用方式下，强 Copyleft 与更弱的 Copyleft、弱 Copyleft 与更弱的 Copyleft、更弱的 Copyleft 许可协议与更弱的 Copyleft 许可协议之间通常是相容的，主要原因在于更弱的 Copyleft 许可协议通常允许使用其他许可协议再发布，组合后的作品整体遵循 Copyleft 更强的许可协议。

2.2.3 开源许可证相容性判定例外情况

上一小节中介绍了在未考虑各开源许可证中有关分发和相容性的独特性要求的情形下，各开源许可证相容性判定的通用方法。该方法对于原文中未对相容性做出明确说明的不同许可证而言，是适用的。那么对于原文中包含明确的相容许可协议清单或有关条款的部分开源许可证来说，在判断其与其他开源许可证之间是否相容时，我们主要依据该说明进行判断，上一小节中介绍的方法用于辅助判断。

开源许可证中有关其相容性的表述和条款包含如下示例：

Relicense: 重新授予许可。某些 Copyleft 许可协议明确声明了在与其兼容的 Copyleft 许可协议组合时，允许变更许可。如 EUPL、CECILL、RPSL-1.0 等。

Secondary License: 次级许可协议。某些 Copyleft 许可协议包含一个次级许可协议清单，只要该清单上列举了要组合项目的开源许可协议，即可判定为相容。如 MPL-2.0、EPL-2.0 等。

笔者对 60 余种常见开源许可协议进行阅读，并主要对 30 余种 Copyleft 许可协议的相容性描述进行了归纳汇总，同时借助上节中总结的相容性判断通用方法，制成两种使用方式下开源许可协议相容性图表（见附件），初步完成了对常见开源许可协议冲突和相容关系的归纳和总结。借助该图表，对开源软件的许可证进行相容性分析的流程如图 2.3 所示：

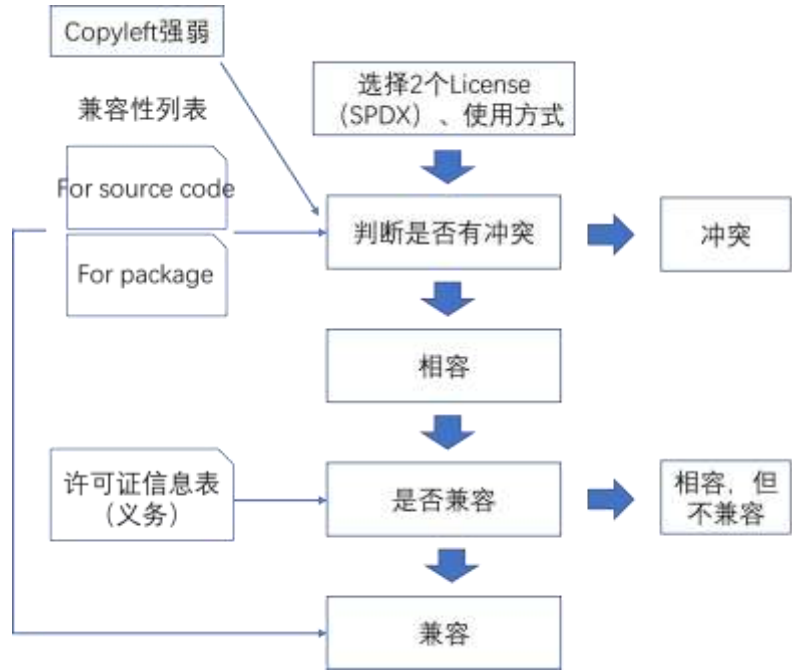


图 2.3 相容性分析流程

由于笔者并非专业律师，因此并未对许可协议中有关专利、版权、法律等相关条款的内容进行分析，因此笔者对相关冲突的忽略表示遗憾。