

**M A S A R Y K O V A  
U N I V E R Z I T A**

PRÁVNICKÁ FAKULTA

**Právní aspekty  
interoperability  
software**

Diplomová práce

DAVID MYSLIVEC

Vedoucí práce: JUDr. Matěj Myška, Ph.D.

Ústav práva a technologií  
Obor Právo

Brno 2021





## Bibliografický záznam

|                          |   |
|--------------------------|---|
| <b>Autor:</b>            | David Myslivec<br>Právnická fakulta<br>Masarykova univerzita<br>Ústav práva a technologií                                     |
| <b>Název práce:</b>      | Právní aspekty interoperability software  |
| <b>Studijní program:</b> | Právo a právní věda   |
| <b>Studijní obor:</b>    | Právo   |
| <b>Vedoucí práce:</b>    | JUDr. Matěj Myška, Ph.D.  |
| <b>Rok:</b>              | 2021  |
| <b>Počet stran:</b>      | 142   |
| <b>Klíčová slova:</b>    | Interoperabilita, software, počítačový program, rozhraní, API, specifikace, implementace, autorské právo, zpětné inženýrství. |

## Bibliographic record

**Author:** David Myslivec  
Faculty of Law  
Masaryk University  
Institute of Law and Technology

**Title of Thesis:** Legal aspects of software interoperability

**Degree Programme:** Law and Legal Science

**Field of Study:** Law

**Supervisor:** JUDr. Matěj Myška, Ph.D.

**Year:** 2021

**Number of Pages:** 142

**Keywords:** Interoperability, software, computer program, interface, API, specification, implementation, copyright, reverse engineering.

## Anotace

Diplomová práce se zabývá právními aspekty interoperability software. Interoperability software je dosahováno pomocí rozhraní, které je součástí počítačového programu. Počítačový program spadá pod ochranu autorského práva. Diplomová práce se zabývá rozsahem autorskoprávní ochrany rozhraní a možnostmi přístupu k nechráněným rozhráním. V diplomové práci jsou poměřovány zájmy autora na ochranu díla proti zájmům veřejným a navrhnutá řešení, která by vedla k zajištění větší míry interoperability, respektive sdílení informací nezbytných pro její zajištění.

## **Abstract**

This thesis addresses the legal aspects of software interoperability. Interoperability of software is achieved through an interface that is part of a computer program. The computer program is protected by copyright. The diploma thesis deals with the scope of interface protection and the possibilities of access to unprotected interfaces. It measures the interests of the author in protecting the work against the public interests and proposes solutions that would be able to ensure a greater degree of interoperability or sharing the information necessary to ensure it.





## Čestné prohlášení

Prohlašuji, že jsem diplomovou práci na téma **Právní aspekty interoperability software** zpracoval sám. Veškeré prameny a zdroje informací, které jsem použil k sepsání této práce, byly citovány v poznámkách pod čarou a jsou uvedeny v seznamu použitých pramenů a literatury.

V Brně 29. března 2021

David Myslivec



## Poděkování

Rád bych na tomto místě poděkoval vedoucímu mé diplomové práce, panu doktoru Matěji Myškovi, za odborné vedení, pochopení a nespočet rad, které mi při jejím psaní poskytl. Nezměrné díky si zaslouží také rodina, přítelkyně a přátelé, kteří mě při psaní podpořili a motivovali k další práci.



## Obsah

|  |           |
|--|-----------|
| <b>Seznam obrázků</b>                                    | <b>13</b> |
| <b>Seznam tabulek</b>                                    | <b>14</b> |
| <b>Seznam pojmů a zkratk</b>                             | <b>15</b> |
| <b>1 Úvod</b>  | <b>17</b> |
| <b>2 Technické aspekty</b>                               | <b>22</b> |
| 2.1 Počítačový program a software .....                  | 22        |
| 2.2 Zdrojový a strojový kód .....                        | 23        |
| 2.3 Rozhraní .....                                       | 25        |
| 2.4 Zpětné inženýrství.....                              | 29        |
| 2.5 Network effect .....                                 | 34        |
| <b>3 Interoperabilita</b>                                | <b>37</b> |
| 3.1 Vertikální interoperabilita .....                    | 38        |
| 3.2 Horizontální interoperabilita.....                   | 39        |
| 3.3 Výhody interoperability.....                         | 40        |
| 3.4 Nevýhody interoperability.....                       | 43        |
| 3.5 Shrnutí .....  | 44        |
| <b>4 Autorskoprávní ochrana rozhraní a její limity</b>   | <b>46</b> |
| 4.1 Chráněné prvky počítačového programu.....            | 48        |
| 4.2 Nechráněné prvky počítačového programu.....          | 55        |
| 4.3 Specifikace a implementace rozhraní .....            | 69        |
| 4.4 Specifikace rozhraní jako myšlenka .....             | 72        |
| 4.5 Standardizace rozhraní .....                         | 75        |
| 4.6 Směrnice o právní ochraně počítačových programů..... | 76        |
| 4.7 Právní úprava v USA.....                             | 82        |
| 4.8 Shrnutí .....  | 84        |

|          |  |            |
|----------|--|------------|
| <b>5</b> | <b>Přístup k rozhraní</b>                                | <b>87</b>  |
| 5.1      | Směrnice o právní ochraně počítačových programů.....     | 89         |
| 5.2      | Nedostatky právní úpravy zpětného inženýrství.....       | 93         |
| 5.3      | Zpětné inženýrství za účelem interoperability v USA..... | 101        |
| 5.4      | Shrnutí .....  | 112        |
| <b>6</b> | <b>Návrhy de lege ferenda</b>                            | <b>113</b> |
| 6.1      | Doplňující ochrana .....                                 | 114        |
| 6.2      | Explicitní vyjmutí specifikace rozhraní z ochrany.....   | 118        |
| 6.3      | Snížení doby ochrany rozhraní .....                      | 119        |
| 6.4      | Rozhraní jako standard .....                             | 120        |
| 6.5      | Povinné zveřejnění specifikace rozhraní .....            | 121        |
| 6.6      | Nový regulační orgán .....                               | 123        |
| 6.7      | Úprava omezujících ustanovení směrnice EU .....          | 124        |
| 6.8      | Další možná řešení.....                                  | 125        |
| <b>7</b> | <b>Závěr</b>   | <b>127</b> |
|          | <b>Použité zdroje</b>                                    | <b>130</b> |

## Seznam obrázků

|   |    |
|---|----|
| Obr. 1: Část specifikace rozhraní ..... | 28 |
|---|----|

## Seznam tabulek

|   |    |
|---|----|
| Tab. 1: Výhody a nevýhody zpětného inženýrství..... | 31 |
|---|----|



## Seznam pojmů a zkratek

|           |   |
|-----------|---|
| 2d Cir.   | – United States Court of Appeals for the Second Circuit.                  |
| 3d Cir.   | – United States Court of Appeals for the Third Circuit.                   |
| 5th Cir.  | – United States Court of Appeals for the Fifth Circuit.                   |
| API       | – Application Programming Interface.                                      |
| AZ        | – Zákon č. 121/2000 Sb. Autorský zákon.                                   |
| CONTU     | – Commission on New Technological Uses of Copyrighted Works.              |
| ECIS      | – European Committee on Interoperable Systems.                            |
| EPO       | – European patent office.   |
| N.D. Cal. | – U.S. District Court for the Northern District of California.            |
| SAGE      | – Software Action Group for Europe.                                       |
| SCOTUS    | – The Supreme Court of the United States.                                 |
| SEGA      | – Sega Corporation.   |
| SFEU      | – Smlouva o fungování Evropské unie.                                      |
| SDEU      | – Soudní dvůr Evropské unie.  |
| TRIPS     | – The Agreement on Trade-Related Aspects of Intellectual Property Rights. |
| USPTO     | – United States Patent and Trademark Office.                              |
| WIPO      | – World Intellectual Property Organization.                               |
| WPL       | – World Programming Limited.  |



## 1 Úvod

Moderní technologie, množina nepřeborného množství věcí, hmotných i nehmotných, které doprovází život téměř každého jedince dnešního vyspělého světa. Všichni je využívají a spoléhají na ně. Dovolíme si tvrdit, že člověk, který se jejich využívání bude bránit a nepřizpůsobí se jim, brzy narazí na nepochopení a nemožnost zcela soběstačně fungovat, a to nejen v soukromé, ale zejména v pracovní sféře svého života. Jejich čím dál častější používání a téměř nemožnost se jim vyhnout, značně přispívá a urychluje proces, který vede k jejich dalšímu rozšiřování. Využíváme je, aniž bychom měli potřebu zjišťovat, jak fungují. Málokdo vidí na pozadí moderních technologií to, co dnešní, na první pohled samostatné systémy propojuje a umožňuje jim spolu sdílet informace. Mimo vlastní funkcionality software je pro jejich plné využití důležitá právě interoperabilita, tedy schopnost dvou počítačových programů spolu komunikovat, vyměňovat si informace a vzájemně tyto informace používat. Její nedostatek může mít dopad jak na uživatele dané technologie, tak na její autory. Bez interoperability nemusí být možné produkt využívat vůbec nebo to lze jen se značnými komplikacemi. Bez zájmu autorů o interoperabilitu je omezena konkurenceschopnost menších podniků, které spoléhají při vývoji vlastních produktů na interoperabilitu s produkty etablovaných společností. V případě počítačových programů je interoperability dosahováno pomocí rozhraní, konkrétně rozhraní pro programování aplikací, tzv. *API*. Právě *API* je zodpovědné za to, že spolu dva počítačové programy mohou komunikovat. Tam kde v této práci bude zmiňován pojem rozhraní, bude tím myšleno právě *API*<sup>1</sup>.

Počítačový program je chráněn autorským právem jako literární dílo. U rozhraní však situace není zcela zřejmá. Právě to, zda je autorským právem rozhraní chráněno nebo ne, respektive v jakém rozsahu, je pro dnešní trh postavený na vzájemné komunikaci jednotlivých programů zásadní. Kritičnost interoperability si uvědomuje i Evropská komise. Ta cílí na to, aby existující produkty mohly být základním stavebním kamenem, ze kterého mohou vycházet společnosti při vývoji nových

---

<sup>1</sup> Pojem *API* řešíme v kap. 2.3.

produktů.<sup>2</sup> Na základě tohoto tvrzení se může zdát, že výhoda interoperability je pouze na straně menších nebo nových společností.<sup>3</sup> Tomu tak však není. Je důležité zmínit, že právě produkty, které rozšiřují, doplňují či vylepšují původní počítačové programy<sup>4</sup> a jejich funkcionality, mohou pro původní počítačový program představovat vysokou přidanou hodnotu. To je v konečném důsledku výhodné zejména pro autora původního programu, který interoperabilitu umožnil. Takovéto vzájemné výhody a vzájemné ovlivňování lze označit jako *network effect*<sup>5</sup>.

V případě interoperability proti sobě stojí zdánlivě protichůdné zájmy. Na jedné straně se nacházejí majetkové a osobnostní zájmy autora na ochranu jím vytvořeného díla. Na druhé straně se pak nacházejí zájmy společnosti. Pod ně spadá zejména zájem na zajištění interoperability a z ní plynoucích výhod. Je proto nezbytné najít proporcionální vztah mezi právy autora k jeho dílu a mezi zájmy společnosti, která těží z interoperability. Pro dosažení interoperability jsou zásadní informace o rozhraní, respektive jeho *specifikace*. Takové informace může autor poskytnout dobrovolně, a mnohdy tomu tak skutečně je, neboť autor spoléhá na páku, kterou mu přinese *network effect*. Rovněž tak nastávají situace, kdy je autor povinen poskytnout informace nezbytné k zajištění interoperability na základě rozhodnutí příslušného orgánu.<sup>6</sup> Nejdůležitější však je, jak se k interoperabilitě staví právní úprava, respektive do jaké míry je rozhraní umožňující interoperabilitu chráněno, případně do jaké míry lze do autorského práva za účelem zjištění informací umožňující interoperabilitu zasahovat. To, že je počítačový program předmětem autorských práv tedy samo o sobě neznamená, že informace o rozhraní může zpřístupnit pouze autor.

Ačkoli lze na interoperabilitu pohlížet z různých úhlu, cílem této práce je popsání a zhodnocení toho, do jaké míry je rozhraní chráněno

---

<sup>2</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*, Brusel, 2013, s. 3.

<sup>3</sup> Tamtéž, s. 3-4.

<sup>4</sup> Pojem software řešíme v kap. 2.1.

<sup>5</sup> Network effect je známý také jako síťový efekt. Dále je řešen v kap. 2.5.

<sup>6</sup> Např. rozhodnutí Evropské komise v řízení podle článku 82 [ES] a článku 54 Dohody o EHP zahájené proti *Microsoft Corp*, ze dne 24. března 2004, sp. zn. COMP/C-3/37.792 – Microsoft.

autorským právem, respektive jaké možnosti nám autorské právo dává v zájmu zachování interoperability počítačových programů.

V kapitole 2 bude nutné seznámit se s tím, co to počítačový program, respektive rozhraní je, jak funguje a z čeho se skládá. Bude nezbytné definovat si rozdíl mezi zdrojovým a strojovým kódem, stejně tak jako rozdíl mezi specifikací a implementací rozhraní. Jednou z možných cest k získání informací o rozhraní je zpětné inženýrství, proto si představíme, o co se jedná a jak se uskutečňuje. Pochopení těchto technických záležitostí umožní lépe pracovat s právní úpravou a judikaturou týkající se interoperability.

Kapitola 3 se bude věnovat interoperabilitě jako takové, jak si ji představit, jak je definována, jak je jí dosahováno a proč je pro současný technologický svět nezbytná. Vzhledem k tomu, že interoperability je dosahováno pomocí rozhraní, bude v práci věnován prostor úloze rozhraní při jejím zajišťování. Důležité v této části bude vyvažování výhod a nevýhod plynoucích z interoperability, stejně tak jako rozdíl mezi vertikální a horizontální interoperabilitou. Budeme se tedy věnovat konkrétním výhodám a nevýhodám interoperability. Mezi výhody lze zahrnout například rychlejší technologický rozvoj nebo podporu hospodářské soutěže. Mezi nevýhody můžeme zahrnout například možnou nižší bezpečnost interoperabilních systémů.

V kapitole 4 se budeme věnovat autorskoprávní ochraně rozhraní. Rozhraní je součástí počítačového programu a ten je chráněn jako autorské dílo. Vymezíme si tedy, co to autorské dílo je a jaké musí rozhraní splňovat požadavky, aby podléhalo ochraně autorským právem. Zejména bude rozebrána podmínka originality, která je pro ochranu klíčová, a to včetně možných interpretačních problémů a jejich řešení. Nicméně to, že je dílo, respektive počítačový program shledán dílem autorským znamená, že autorovi náleží monopol na veškeré prvky v dílu obsažené. Proto budou v této kapitole probrány nechráněné prvky počítačového programu, tzv. autorskoprávně volné složky díla. Ty v autorském právu slouží k vyvažování zájmů autora a zájmů společnosti. Bude vysvětlena dichotomie rozlišování mezi myšlenkou a vyjádřením<sup>7</sup>, která se objevila jak v právních předpisech, tak v judikatuře řešící spory týkající se rozhraní. Autorské dílo totiž chrání pouze vyjádření, oproti tomu myšlenky, stejně tak jako metody a principy, na kterých dílo stojí, zůstávají veřejně

---

<sup>7</sup> Dále bude v textu označována také zkráceně jako dichotomie myšlenka/vyjádření.

přístupné a ochraně nepodléhají. Neméně důležité jsou způsoby, pomocí kterých, lze odlišit chráněné prvky díla od prvků nechráněných.

Z důvodu dichotomie myšlenka/vyjádření bude zásadní dělení rozhraní na *specifikaci* a *implemetnaci* a jejich konkrétní podřazení pod chráněný nebo nechráněný prvek díla. V této části bude pro lepší pochopení specifik počítačového programu, respektive rozhraní, zmíněna funkcionality a její srovnatelnost s myšlenkou. Podstatná část zde bude věnována směrnici o ochraně počítačových programů<sup>8</sup>, neboť ta je pro autorskoprávní ochranu rozhraní v Evropské unii, a tedy i v našem právním řádu, klíčová. Rovněž tak bude porovnána právní úprava ve Spojených státech amerických, ve kterých je situace vzhledem k uzákonění *idea-expression dichotomy* obdobná. Význam judikatury a právní úpravy ve Spojených státech amerických je značný. Sídlí zde velké množství technologických společností, které mají významný vliv na světový trh. Mnohdy se jedná o společnosti v dominantním postavení, které se dostávají do konfliktů s ostatními společnostmi, kterým neposkytnou informace o interoperabilitě.<sup>9</sup> Zároveň byla a stále je většina významných sporů s dopadem na interoperabilitu řešena právě ve Spojených státech amerických. Spolu s právní úpravou tedy bude akcentována i judikatura, která mnohdy vágní právní úpravu významně doplňuje a upřesňuje.

Kapitola 5 rozvede nedostatky přístupu k nechráněným částem počítačového programu. Pro klasická autorská díla je běžné, že myšlenky v nich nezůstávají nabyvateli licence skryty. U počítačového programu tomu tak není, a i ty části, které jsou shledány jako myšlenky, jsou pro veřejnost skryty ve strojovém kódu. I tehdy, je-li autorská ochrana rozhraní odmítnuta, je pro nabyvatele díla obtížné informace o rozhraní, bez možnosti zpětného inženýrství, získat. Bude tedy nutné definovat možnosti, které se uživateli díla nabízejí k tomu, aby nechráněné informace, nezbytné k dosažení interoperability, obsažené v rozhraní, mohl zpětným inženýrstvím získat. Vzhledem k tomu, že aktuálně probíhá<sup>10</sup> jeden ze zásadních soudních sporů, jehož výsledek může mít na problematiku

---

<sup>8</sup> Směrnice o právní ochraně počítačových programů.

<sup>9</sup> Např. Judgment of the Court of First Instance of 17 September 2007. T-201/04.

*Microsoft Corp. v Commission of the European Communities.*

<sup>10</sup> Stav je účinný ke dni 20. března 2021.

interoperability značný vliv, bude tento spor v závěru této části představen.<sup>11</sup>

V závěru práce, v kapitole 6, se zamyslíme nad tím, zdali je aktuální právní úprava postačující a dostatečně přiléhavá k problematice rozhraní. A jestli by nebylo případně vhodné autorskoprávní režim rozhraní doplnit nebo alternovat jiným právním režimem, např. právy na ochrany hospodářské soutěže.

---

<sup>11</sup> SCOTUS. *Google LLC v. Oracle America, Inc.* Docket for 18-956 [online]. [cit. 26. 2. 2021]. <https://www.supremecourt.gov/docket/docketfiles/html/public/18-956.html>

## 2 Technické aspekty

Cílem této kapitoly je seznámení se základními technickými pojmy, které následně poslouží k analýze právních aspektů zkoumané problematiky. Nejprve bude nutné seznámit se s tím, co je to počítačový program, jak je definován, jak funguje a čím je tvořen. Pro tyto účely bude popsán rozdíl mezi strojovým kódem, který je pro člověka nečitelný a zdrojovým kódem, který je pro člověka čitelný. Bude důležité vymezit rozhraní, které je v počítačovém programu nezbytné pro zachování interoperability, stejně tak jako jeho dvě části, na které jej lze dělit, a to specifikaci rozhraní a jeho implementaci. Neméně důležité je pro interoperabilitu zpětné inženýrství. Zpětné inženýrství je jednou z možností, jak získat informace nezbytné pro interoperabilitu.

### 2.1 Počítačový program a software

Počítačový program je soubor operací, výpočtů a instrukcí, které jsou automaticky prováděny na vybraném médiu zejména právě na počítači.<sup>12</sup> Pomocí souboru instrukcí jsou počítačem uskutečňovány konkrétní úlohy. Počítačový program, tedy sada instrukcí a příkazů, je primárně vytvářen pomocí programovacích jazyků. Programátor pomocí zvoleného jazyka zapíše algoritmy, které má počítač provést.<sup>13</sup> Takto vytvořený zápis se nazývá zdrojový kód. Zdrojový kód tak ve svém celku tvoří samostatný program. Počítač je schopný pomocí procesoru s kódem pracovat a vykonávat žádané operace. Obvykle je však nutné zdrojový kód za tímto účelem převést do strojové podoby, tedy do kódu strojového.<sup>14</sup>

V současnosti není příliš rozlišováno mezi pojmem *software* a pojmem *počítačový program*.<sup>15</sup> Někteří autoři na toto nedostatečné dělení

---

<sup>12</sup> Např. § 101 of *Copyright Law of the United States of America*.

<sup>13</sup> Štědroň, B. *Ochrana a licencování počítačového programu*. Praha: Wolters Kluwer ČR, 2010, s. 7.

<sup>14</sup> Nádběla, J. *Velký počítačový slovník. 1*. Kralice na Hané: Computer Media, 2004.

<sup>15</sup> Např. § 30 zákona č. 108/2006 Sb. Zákon o sociálních službách ve znění pozdějších předpisů: „*Ministerstvo zajišťuje pro provádění zákona na vlastní náklady počítačový program (software)*“.



upozorňují.<sup>16</sup> Nejedná se o synonyma, software je z technického pohledu rozsáhlejší a mnohdy zahrnuje několik počítačových programů. Nejednotná je odborná veřejnost také v tom, zda pod pojem software spadají data, dokumentace a případně další prvky.<sup>17</sup> Například Šavelka dělí počítačový program na hardware a software s tím, že prvky, které nespádají pod pojem hardware jsou dle jeho názoru software.<sup>18</sup> Software plní konkrétní příkazy uživatele počítače, je to sada instrukcí, které musí počítač interpretovat a následně provést. Jednoduše řečeno je software soubor počítačových programů, případně dalších dat a jiných prvků, díky jejichž spolupráci může software, který je zahrnuje, fungovat.<sup>19</sup> Běžně se dělí na software aplikační, který tvoří aplikace využívané ke konkrétní činnosti, se kterými přímo pracujeme a software systémový, který zajišťuje fungování počítače v celku.<sup>20</sup> Pro software je vhodnější použití synonymního pojmu *programové vybavení*, které je pojmu software bližší.<sup>21</sup> Pro účely této práce však striktní dělení těchto pojmů není nutné. V práci bude dále používán pojem počítačový program.

## 2.2 Zdrojový a strojový kód

Jak jsme již zmínili, zápis počítačového programu je obvykle proveden textovou formou tzv. zdrojovým kódem. Ten je pro člověka čitelný, je tedy možné s ním pracovat, psát jej, měnit a číst. Pro jeho zápis

---

<sup>16</sup> Telec, I., Tůma, P. *Autorský zákon, Komentář. 2. vydání*. Praha: C.H. Beck, 2019, s. 7-71, bod 29; Srov. Ballardini, R. M. *Intellectual Property Protection for Computer Programs: Developments, Challenges, and Pressures for Change*. Helsinky: Svenska handelshögskolan, 2012, s. 11 a násl.

<sup>17</sup> Štědroň. *Ochrana a licencování počítačového programu*, s. 6; Srov. Maisner, M. a kol. *Základy softwarového práva. Vyd. 1*. Praha: Wolters Kluwer Česká republika, 2011, s. 2.

<sup>18</sup> Šavelka, J. *Autorskoprávní ochrana funkcionality softwaru* [online]. Rigorózní práce, Brno, Masarykova univerzita, Právnická fakulta, 2013 [cit. 2021-03-19]. <https://is.muni.cz/th/e974x/>

<sup>19</sup> Hlavenka, J. a kol. *Výkladový slovník výpočetní techniky a komunikací*. Praha: Computer Press, 1997. s. 379.

<sup>20</sup> Štědroň. *Ochrana a licencování počítačového programu*. s. 6.

<sup>21</sup> Telec. Tůma. *Autorský zákon, Komentář. 2. vydání*, s. 7-71, bod 29; Srov. Rozhodnutí Nejvyššího soudu ze dne 15. 03. 2005, sp. zn. 5 Tdo 291/2005.

programátoři využívají jeden z mnoha programovacích jazyků. Programovací jazyk je systematicky uspořádaný soubor instrukcí, který mnohdy obsahuje i dodatečné komentáře.<sup>22</sup> Komentáře jsou nezbytné pro usnadnění budoucí práce s programem. Pro programátora nemusí být smysl zdrojového kódu zcela zřejmý, nebude-li doplněn o informace v komentářích.

Počítačový program je možné pomocí interpretu spouštět již ve formě zdrojového kódu, nicméně častější je jeho „překlad“ do podoby strojového kódu, který je pro procesor počítače obvykle uzpůsobený.<sup>23</sup> Pro překlad je používán tzv. *compiler* [překladač], který strojový kód vygeneruje a optimalizuje.<sup>24</sup> Strojový kód je pro člověka v zásadě nečitelný, jedná se o uspořádání čísel v některé z číselných soustav<sup>25</sup>. Na základě takto, pomocí čísel, utvořené instrukce, vykonává procesor přesně specifikované úlohy.<sup>26</sup> Z tohoto důvodu lze uvažovat i o rozdílech mezi autorskoprávní ochranou strojového kódu, který jako takový nelze číst ani nelze identifikovat jeho autora a autorskoprávní ochranou zdrojového kódu, který lze získat ze strojového kódu pomocí zpětného inženýrství. Zdrojový kód získaný zpětným inženýrstvím již lze přečíst a zároveň je snazší identifikovat jeho autora. Pro autora počítačového programu se může zdát bezpečnější nezpřístupnit zdrojový kód současně s programem samotným, neboť tím zabrání jeho zneužití. Takovým postupem je však bráněno v přístupu k myšlenkám autorského díla a informacím nezbytným pro interoperabilitu.

---

<sup>22</sup> Hemmendinger, D. *Computer programming language*. Encyclopædia Britannica [online]. 2015-1-28 [cit. 10.1.2021]. <http://www.britannica.com/technology/computer-programming-language>

<sup>23</sup> *What is the difference between a compiled and an interpreted program?* Indiana University [online]. [cit. 25. 1. 2021]. <https://kb.iu.edu/d/agsz>

<sup>24</sup> PC Mag Staff, *The Computer Language Co Inc. Definition of Compiler*. [online]. [cit. 10.1.2021]. <https://www.pcmag.com/encyclopedia/term/compiler>

<sup>25</sup> Typickým příkladem číselné soustavy je např. binární soustava.

<sup>26</sup> Emmerik, M.V. Waddington, T. *Using a decompiler for real-world source recovery*. 11th Working Conference on Reverse Engineering, Delft, Netherlands, 2004.

## 2.3 Rozhraní

Počítačové programy mohou být tvořeny subprogramy anebo s jinými programy spolupracují a komunikují. Tato spolupráce se však netýká pouze software, je důležitá i pro správné fungování software ve vztahu k hardware a rovněž hardware ve vztahu k hardware.<sup>27</sup> Interoperability mezi jedním a ostatními programy je dosahováno pomocí rozhraní, které je jako takové součástí počítačového programu. Právě rozhraní umožňuje zpracovávat informace poskytované jinými počítačovými programy, odpovídat na ně a obecně je propojovat bez provedení dalších úprav.<sup>28</sup> Díky tomu je umožněno jednomu programu přistupovat k funkcionalitám programu jiného.

Existují například uživatelská rozhraní umožňující uživateli interakci s počítačovým programem, datové rozhraní (např. *.docs*, *.pdf* a další) umožňující ukládání, vyvolávání a práci s jednotlivými soubory a komunikační rozhraní umožňující telekomunikaci (např. *http*).<sup>29</sup> Rozhraní, které je pro interoperabilitu počítačových programů zásadní se nazývá API. API je zkratka pro *Application Programming Interface*, tedy rozhraní pro programování aplikací, zajišťující interoperabilitu ve vztahu jednoho počítačového programu k jinému programu. API je definováno jako rozhraní aplikací, které určuje proceduru a způsob, jakým jsou aplikací požadovaná data, funkce nebo protokoly volány z druhé aplikace, respektive co s těmito daty může tato aplikace dělat.<sup>30</sup> Pomocí API jsou tak zpřístupněna data pouze v předem určeném rozsahu a volají se pouze určitým způsobem. Jedná se tedy o relativně kontrolovaný a bezpečný způsob komunikace software. Díky API nemusí mít počítačový program veškeré žádané funkcionality vepsané ve svém zdrojovém kódu, ale může

---

<sup>27</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*, s. 6.

<sup>28</sup> Johnson-Laird, A. *Reverse Engineering of Software: Separating Legal Mythology from Actual Technology*, 5 *Software L.J.* 1992, s. 339.

<sup>29</sup> Van Rooijen, A. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*. Brussels, 2010, s. 14.

<sup>30</sup> Application Programming Interface. Computerworld [online]. [cit. 25. 2. 2021]. <https://www.computerworld.com/article/2593623/application-programming-interface.html>

využít již existující počítačový program se kterým se pomocí API propojí. Může tak využívat jeho funkcionality a data v něm obsažená.<sup>31</sup>

Budeme-li pohlížet na zdrojový kód jako na základní úroveň abstrakce, rozhraní lze chápat jako vyšší úroveň abstrakce.<sup>32</sup> Rozhraní je výsledkem konkrétní sady instrukcí, jež je zapsána v kódu. Jedná se o funkční prvek počítačového programu, který nekomunikuje s uživatelem.<sup>33</sup> Tato rozhraní mají různé úrovně složitosti, jedná se například o jednoduché odkazy na databáze nebo konkrétní datové sady, ale i podrobnější nastavení.<sup>34</sup>

API tvoří navzájem propojené prvky, které lze z pragmatických důvodů dělit. Jako první je důležité zmínit tzv. *specifikaci* rozhraní, která stanovuje, co rozhraní umí, jaký má účel, jak se chová, jaké informace poskytuje a jaké informace potřebuje k tomu, aby mohl být jeden program propojený s jiným programem. Specifikace rozhraní v podstatě definuje způsoby, jak by mělo být rozhraní implementováno do kódu. Hojně užívanou specifikací je například specifikace OpenAPI<sup>35</sup> (Obr. 1). Specifikace API se často zaměřuje s definicí, respektive dokumentací API. Kód, kterým se rozhraní zapisuje do počítačového programu je tedy *implementace*, jedná se o funkční prvek API, který již zajišťuje samotnou komunikaci (interoperabilitu) mezi programy jako takovou.<sup>36</sup> Implementaci rozhraní tak lze vytvořit na základě informací, které přináší specifikace API. Pro zajištění interoperability je tedy dostačující znalost první části,

---

<sup>31</sup> Freeman, J. *What is an API? Application programming interfaces explained*. [online]. InfoWorld. 8. 8. 2019 [cit. 25. 2. 2021]. <https://www.infoworld.com/article/3269878/what-is-an-api-application-programming-interfaces-explained.html>

<sup>32</sup> Karjala, D. S. *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller Symposium: Copyright Protection and Reverse Engineering of Software*. University of Dayton Law Review. 1993, s. 990.

<sup>33</sup> Tamtéž.

<sup>34</sup> Borgogno, O., Colangelo, G. *Data Sharing and Interoperability Through APIs: Insights from European Regulatory Strategy*. Rochester [online]. NY: Social Science Research Network, 2018 [cit. 9. 11. 2020]. <https://papers.ssrn.com/abstract=3288460>, s. 8.

<sup>35</sup> Github, *RAML Version 1.0: RESTful API Modeling Language*. [online]. [cit. 2. 2. 2021], <https://github.com/raml-org/raml-spec/blob/master/versions/raml-10/raml-10.md/>

<sup>36</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s.15

specifikace rozhraní. Vývojář má běžně možnost navrhnout program, a tedy i rozhraní, které tvoří jeho součást dle své vlastní kreativity. Může zvolit jeden z mnoha způsobů vedoucích k žádané funkcionalitě. Jeho možnosti jsou však v případě rozhraní značně omezené, protože je nutné, aby jím vytvořený program, respektive jeho rozhraní vyhovovalo požadavkům specifikace rozhraní programu, se kterým zamýšlí interoperabilitu.<sup>37</sup>

---

<sup>37</sup> Samuelson, P. *Are Patents on Interfaces Impeding Interoperability?* Minnesota Law Review, Forthcoming, 2009: SSRN [online]. [cit. 25. 2. 2021]. [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1323838](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1323838), s. 5.

```
1  {
2    "openapi": "3.0.0",
3    "info": {
4      "title": "Simple API overview",
5      "version": "2.0.0"
6    },
7    "paths": {
8      "/": {
9        "get": {
10         "operationId": "listVersionsv2",
11         "summary": "List API versions",
12         "responses": {
13           "200": {
14             "description": "200 response",
15             "content": {
16               "application/json": {
17                 "examples": {
18                   "foo": {
19                     "value": {
20                       "versions": [
21                         {
22                           "status": "CURRENT",
23                           "updated": "2011-01-21T11:33:21Z",
24                           "id": "v2.0",
25                           "links": [
26                             {
27                               "href": "http://127.0.0.1:8774/v2/",
28                               "rel": "self"
```

**Obr. 1:** Část specifikace rozhraní<sup>38</sup>

Aby mezi dvěma počítačovými programy byla zajištěna interoperabilita, musí být implementovány na základě stejné specifikace.<sup>39</sup> Bude-li tato podmínka splněna, není důvodné, aby byla totožná i implementace. Protože je nutné, aby se specifikace rozhraní shodovaly, je nutné, aby byla specifikace jednoho rozhraní známa autorovi interoperabilního

---

<sup>38</sup> Swagger, *SmartBear Software OpenAPI Specification, version 3.0.3*. [online]. [cit. 2. 2. 2021], <https://swagger.io/specification/>

<sup>39</sup> Menell, P. S. *Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*. SSRN Electronic Journal. 2017 [cit. 12. 2. 2021]. <http://www.ssrn.com/abstract=2893192>, s. 444.

programu. Případně si musí tuto informaci o interoperabilitě vyhledat pomocí zpětného inženýrství ve strojovém kódu sám.

Pro tyto účely je tedy nutné určit, zda je rozhraní chráněné autorským právem a zda jsou chráněny obě části, na které je rozhraní děleno nebo je chráněna pouze implementace rozhraní. Problém pak nastává v případě dokazování. Autor původního díla musí prokázat, že bylo jeho dílo okopírováno a nejedná se o pouhou podobnost z důvodu technického předurčení. Rovněž je možné, aby několik na sobě nezávislých vývojářů vyvinulo na základě jedné specifikace rozhraní program se stejnou implementací rozhraní. V takovém případě se nemůže jednat o porušení autorského práva. Zpřístupnění API nebo alespoň specifikace API je přímou cestou k interoperabilitě mezi různými systémy a usnadňuje výměnu dat.<sup>40</sup> Je však důležité vyřešit problematické aspekty autorského práva, které API doprovázejí.

## 2.4 Zpětné inženýrství

Z výše uvedené podstaty fungování počítačového programu je zřejmé, že nabyvatel počítačového programu má přístup ke strojovému kódu, málokdy je mu však bez dalšího kroku umožněn přístup ke kódu zdrojovému. Díky tomu jsou skryty myšlenky, na kterých počítačový program stojí. Strojový kód totiž, jak již bylo řečeno, není pro člověka čitelný.<sup>41</sup> Autor si tak sice je schopen lépe chránit své dílo, včetně znalostí a schopností, které do něj vložil, na druhou stranu nedostatek informací o rozhraní zásadním způsobem omezuje interoperabilitu. Nebyl-li předán s počítačovým programem i zdrojový kód, je jednou z možností, jak se k informacím o interoperabilitě dostat, právě zpětné inženýrství. Častým motivem, který vede firmy k provádění zpětného inženýrství je snížení nákladů na vývoj vlastního produktu, ve kterém mohou využít již vytvořený zdrojový kód z jiného počítačového programu.<sup>42</sup> Tato výhoda však nemusí platit u rozsáhlých počítačových programů, jejichž zpětné

---

<sup>40</sup> Borgogno, Colangelo. *Data Sharing and Interoperability Through APIs: Insights from European Regulatory Strategy*, s. 6.

<sup>41</sup> McManis, C. R. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*. High Technology Law Journal. 1993, č. 1, s. 28.

<sup>42</sup> Tamtéž, s. 30.

inženýrství může být nemožné nebo je natolik náročné, že se nevyplatí. Zároveň Andrew Johnson-Laird ve své publikaci<sup>43</sup> zmiňuje, že i přes zpětné inženýrství celého programu se zpětný inženýr nedostane k celé jeho podstatě, protože překlad ze zdrojového kódu do strojového již neobsahuje veškerý důležitý obsah, jako například komentáře.<sup>44</sup> Tuto metodu zjišťování informací o interoperabilitě doprovází další nevýhody, kterých si je vědoma i Evropská unie.<sup>45</sup> Nejdůležitější z nevýhod bude zajisté zdlouhavost, neefektivnost a cena zpětného inženýrství, která vede k hledání jednodušší a levnější cesty. Tou může být například okopírování celého kódu. Této nevýhody si je zajisté vědoma i komunita vývojářů, kteří tak jsou nepřímo tlačeni ke zveřejňování informací o interoperabilitě tak, aby uživatelé nebyli nuceni program dekompileovat. To pak může vést ke zlepšení interoperability bez jakýchkoli nucených zásahů do počítačových programů. Weston však zmiňuje, že většina vývojářů nepovažuje zveřejnění informací o interoperabilitě za důležité.<sup>46</sup> Vzhledem k nákladnosti a složitosti zpětného inženýrství je zarážející, že se jedná o natolik využívanou metodu.<sup>47</sup> Samuelson a Scotchmer shrnuli ekonomický efekt plynoucí z legální možnosti provádět zpětné inženýrství za účelem interoperability přehledně v Tab. 1.

---

<sup>43</sup> Johnson-Laird, A. *Software Reverse Engineering in the Real World Symposium: Copyright Protection and Reverse Engineering of Software*. University of Dayton Law Review. 1993, č. 3.

<sup>44</sup> Tamtéž, s. 896.

<sup>45</sup> Viz. European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*.

<sup>46</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 24.

<sup>47</sup> Samuelson, P., Scotchmer, S. *The Law & Economics of Reverse Engineering*. 11 Yale L.J. 2002. <https://digitalcommons.law.yale.edu/yj/vol111/iss7/2>, s. 1614.



| Social Welfare Criterion           | Reverse Engineering Legal                 | Reverse Engineering Illegal                |
|------------------------------------|---|--|
| Incentives to develop platforms    | Worse (Adequate?)                         | Better (Too high?)                         |
| Incentives to develop applications | Good (Better?)                            | Good                                       |
| Price                              | Ambiguous (short run)<br>Lower (long run) | Ambiguous (short run)<br>Higher (long run) |
| Waisted costs                      | Better?                                   | Worse?                                     |

Tab. 1: Výhody a nevýhody zpětného inženýrství<sup>48</sup>

Zpětné inženýrství v každém případě může a mělo by přinést výhodu jak pro zpětného inženýra, tak pro vývojáře. Oběma totiž může značně snížit náklady, zlepšit postavení na trhu a pomoci s inovací svého produktu. Může se jednat o vhodné řešení. Není však zcela bez rizika, že se osoba provádějící zpětné inženýrství nedostane do rozporu s autorským právem. Zpětné inženýrství je proces, při kterém je program rozebrán tak, aby odhalil základy, na kterých je postaven. Mohli bychom jej definovat jako proces při kterém je analyzován počítačový program za účelem tvorby „kopie“ systému na vyšší úrovni abstrakce<sup>49</sup>, která dopomůže k pochopení jeho konstrukce a složení.<sup>50</sup> Metod zpětného inženýrství je vícero. Některé dbají na autorská práva více než jiné. Nejčastěji využívané metody jsou dvě, *dekompilace* a *black box testing*.<sup>51</sup>

První metodou, jak se ke zdrojovému kódu dostat, je pomocí tzv. *dekompilátoru*. Ten ze strojového kódu získá informace umožňující mu jeho překlad do kódu zdrojového. Jedná se o mnohem invazivnější metodu zpětného inženýrství, než je tomu v případě metody *black-box testing*. Osoba provádějící zpětné inženýrství při jejím použití může dekompilovat i části programu, které k interoperabilitě nejsou potřebné.<sup>52</sup> Nicméně jedná se o metodu méně náročnou, která osobě provádějící zpětné inženýrství ukáže veškerá v programu obsažená rozhraní v podobě

<sup>48</sup> Samuelson, P., Scotchmer, S. *The Law & Economics of Reverse Engineering*, s. 1621.

<sup>49</sup> Na vyšší úrovni abstrakce se nachází např. zdrojový kód.

<sup>50</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 18.

<sup>51</sup> Zbránek, L. Právní aspekty zpětné analýzy počítačových programů. *Revue pro právo a technologie*, 2012, s. 57

<sup>52</sup> Tamtéž.

zdrojového kódu. Nemůže se tak stát, že některé rozhraní bude zpětným inženýrem opomenuto, jako tomu může být v případě metody black-box testing.<sup>53</sup>

Problematickou část *dekompilace*, při které je osobě provádějící zpětné inženýrství viditelný celý zdrojový kód a hrozí tak jeho kopírování, lze částečně řešit metodou „*clean room*“.<sup>54</sup> Při této metodě se získávají informace nezbytné k dosažení interoperability analýzou původního programu prováděnou odděleně od vývoje programu nového. Běžně tedy na zpětném inženýrství pracují dva týmy. Jeden tým program dekompile, a aniž by druhému týmu předal zdrojový kód, informuje jej podrobně o tom, jak program funguje.<sup>55</sup> Takto oddělené činnosti mají za cíl vytvoření programu založeného pouze na získaných myšlenkách, principech a metodách původního programu. Nejedná se o obyčejné kopírování chráněného zdrojového kódu, ale důkladnou studii původního programu a popis jeho funkcionalit. Judikatura ukázala<sup>56</sup>, že i v případě kopírování prvním týmem pro pouhou potřebu studia kódu se nejedná o porušení autorského práva.<sup>57</sup> Jedná se o pouhou interpretaci počítačového programu a vývoj nového programu na základě získaných funkcionalit. Není tedy zasahováno do autorských práv, protože se nejedná o pouhé kopírování zdrojového kódu, nýbrž o jeho zkoumání za účelem nalezení myšlenek.<sup>58</sup> V případě, že nový program bude ve svém vyjádření shodný s programem původním, neznamená to porušení autorského práva bez dalšího. Autor původního díla musí prokázat, že zdrojový kód byl okopírován a že se nejedná o podobnost vzniklou například z důvodu

---

<sup>53</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 18.

<sup>54</sup> Moy, R. *A Case against Software Patents*. Santa Clara Computer & High Technology Law Journal, Vol. 17, Issue 1. <https://digitalcommons.law.scu.edu/cgi/viewcontent.cgi?article=1278&context=chtlj>, s. 72

<sup>55</sup> Zbránek. *Právní aspekty zpětné analýzy počítačových programů*, s. 57.

<sup>56</sup> Např. *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992).

<sup>57</sup> Menell. *Rise of the API Copyright Dead? An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, s. 449.

<sup>58</sup> Např. Čl. 5 odst. 3 Směrnice o právní ochraně počítačových: „...je oprávněna, bez svolení nositele práv tento program zkoumat, studovat nebo zkoušet jeho funkčnost za účelem zjištění myšlenek a zásad, které jsou základem kteréhokoliv z prvků programu.“

předurčenosti funkcionalitami.<sup>59</sup> Je však nutné, aby se druhý tým nedostal do styku s původním zdrojovým kódem. Vyjádření, jak je uvedeno níže je totiž autorským právem chráněno, oproti tomu myšlenky a principy v zásadě nikoli.<sup>60</sup>

Některé prvky obsažené ve strojovém kódu však nelze získat ani pomocí dekompilace. Typicky se jedná o komentáře vložené původně do zdrojového kódu vývojářem.<sup>61</sup> Absence takových informací může působit problémy v pochopení funkcionalit konkrétního kódu a učinit kód pro osobu provádějící zpětné inženýrství nepřehledný. Ani o této invazivní metodě zpětného inženýrství tedy nelze tvrdit, že je schopna odhalit veškeré prvky skryté ve strojovém kódu počítačového programu.<sup>62</sup> Často je nutné doplnit ji následnou analýzou.

*Black-box testing* je druhou z metod zpětného inženýrství, při které však není ke zjištění toho, jak počítačový program funguje nebo jaké rozhraní používá nutný zdrojový kód. Zpětný inženýr pouhým pozorováním původního programu, s jakými vstupy pracuje a jaké výstupy generuje, může získat dostatečné informace ohledně rozhraní, která jsou pro interoperabilitu klíčová.<sup>63</sup> Jedná se však o složitější, časově i finančně náročnější metodu, než je dekompilace. Zpětný inženýr musí vědět jakým způsobem získat dostatečné množství informací a jak takto získané informace správně implementovat do svého zdrojového kódu, aby dosahoval stejných funkcionalit jako původní program.<sup>64</sup> Na druhou stranu je při použití této metody potencionálně nižší riziko porušení autorských práv.

---

<sup>59</sup> Důkazní břemeno řešil soud např. v rozsudku Nejvyššího soudu České republiky sp.zn. 32 Cdo 6/2011, ze dne 26. 1. 2012: „*Nestanoví-li právní předpis jinak, stíhá důkazní břemeno toho účastníka, jemuž je existence příslušné skutečnosti podle hmotného práva ku prospěchu.*“

<sup>60</sup> Bernard A. Galler, *Software and Intellectual Property Protection: Copyright and Patent Issues for Computer and Legal Professionals*. Greenwood Publishing Group, 1995, s. 130.

<sup>61</sup> Zbránek. *Právní aspekty zpětné analýzy počítačových programů*, s. 57.

<sup>62</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 20.

<sup>63</sup> Zbránek. *Právní aspekty zpětné analýzy počítačových programů*, s. 57.

<sup>64</sup> Cuillou, C. M. *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*. Columbia-VLA Journal of Law & the Arts. 1997, č. 4, s. 545.

## 2.5 Network effect

Pro pochopení důležitosti interoperability je vhodné zmínit *network effect*. Jedná se o jev, díky kterému je hodnota určité služby nebo produktu zvyšována nárůstem počtu uživatelů, kteří danou službu nebo produkt využívají. Stejný efekt má také nárůst uživatelů služby nebo produktu, které jsou s původní službou nebo produktem kompatibilní.<sup>65</sup> Tento efekt je viditelný zejména v případě moderních technologií, konkrétně u počítačových programů.<sup>66</sup> Typickým příkladem takového efektu jsou i sociální sítě, například Facebook. Existuje obrovské množství sociálních sítí podobného charakteru, avšak přes 2,7 miliardy lidí na celém světě jsou aktivní uživatelé Facebooku.<sup>67</sup> Ostatní sociální platformy se tomuto číslu stěží mohou rovnat, a proto je pro potencionální uživatele Facebook zajímavější a užitečnější. Uživatelé se totiž mohou spolehnout na to, že na Facebooku budou moci komunikovat téměř se všemi známými a přáteli, a že lze přes Facebook přistupovat k velkému množství interoperabilních služeb. Takový rozsah služeb jim ostatní sociální platformy používané mnohem menším počtem lidí nemohou poskytnout. Příchod nových uživatelů tak ve stejném smyslu zároveň přináší užitek uživatelům, kteří již Facebook používají. Důležité je, že výhody network efektu se projeví i v případě dvou interoperabilních počítačových programů, kdy uživatelé jednoho mohou těžit z rozšíření uživatelské základny a funkcionalit programu druhého.<sup>68</sup>

Trh s počítačovými programy se odlišuje od ostatních trhů tím, že na něm působí množství dominantních soutěžitelů.<sup>69</sup> Důvodem je skutečnost, že jak programátoři, tak zákazníci vyhledávají pozitivní přínos

---

<sup>65</sup> Shapiro, Carl. *Information rules: a strategic guide to the network economy*. Varian, Hal R. Boston, Mass.: Harvard Business School Press, 1999, s. 174-182.

<sup>66</sup> Mylly, U. *An Evolutionary Economics perspective on Computer Program Interoperability and Copyright*. IIC Journal, 3/2010, s. 296.

<sup>67</sup> K polovině roku 2020. Viz: Tankovska, H. *Facebook: number of monthly active users worldwide 2008-2020* [online]. 2021. [cit. 2. 2. 2021]. <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>

<sup>68</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 27.

<sup>69</sup> Menell. *Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, s. 318.

network effectu.<sup>70</sup> Hodnota služby nebo produktu je v těchto případech odvislá od množství jejich uživatelů. Pro vývojáře je zajímavější vyvíjet produkt, který bude kompatibilní s počítačovým program, jehož uživatelská základna je na daném trhu největší. S rostoucím počtem takových produktů je zvyšována i hodnota původního programu, který tak nabízí více funkcionalit. Díky tomu se zvyšuje počet vývojářů, kteří k software chtějí vyvíjet své produkty, což opět vede k většímu zájmu spotřebitelů.<sup>71</sup> Pokud tedy společnost z nějakého důvodu nepoužije zavedený standard a nesnaží se o interoperabilitu s již existujícími produkty, je pro ni mnohem obtížnější prosadit na trhu právě svoje řešení a přesvědčit zákazníky a ostatní soutěžitely, že nový produkt je lepší a že je pro ně výhodné migrovat ze stávajících řešení. Společnostem může přinést větší konkurenční výhodu úspěšná softwarová platforma, nežli cenotvorba a kvalita produktů jako taková. Myšlenka možnosti ochrany rozhraní a budování uživatelské základny, ze které není uživatelům umožněno vystoupit začala být proto zajímavá zejména pro velké společnosti.<sup>72</sup> Ochrana rozhraní těmito společnostem umožní zamezit konkurentům v přístupu k produktu a dovolí budovat uzavřený ekosystém či platformu.<sup>73</sup> Vzhledem k nákladům na migraci je pak pro zákazníka výhodnější zůstat v takovémto ekosystému uzamčen, a to i přes vyšší náklady, které si díky tomu autor může vyžádat.

Tento přístup však nezastávají všechny společnosti na trhu. Mezi společnostmi vyvíjejícími počítačové programy se objevil názor, že chránit rozhraní je nadbytečné a povede k omezování svobody tvorby a nižšímu rozvoji technologií. Jejich přesvědčení vedlo až k založení Evropského výboru pro interoperabilitu systémů. Zajímavé je, že jedním ze zakládajících členů ECIS je společnost Sun Microsystems, dnes patřící pod

---

<sup>70</sup> Menell. *Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, s. 318.

<sup>71</sup> Tamtéž, s. 319.

<sup>72</sup> Poudel, S. *Internet of Things: Underlying Technologies, interoperability, and Threats to Privacy and Security*. Berkeley Technology Law Journal 31, 2016, s. 1012.

<sup>73</sup> Například zneužití dominantní postavení společností Microsoft viz Judgment of the Court of First Instance of 17 September 2007. T-201/04. *Microsoft Corp. v Commission of the European Communities*.

společnost Oracle.<sup>74</sup> Společnost Oracle v současnosti vede jeden z největších sporů týkajících se interoperability, ve kterém stojí v podstatě na druhé straně a bojuje za ochranu rozhraní vyvinutého společností Sun.<sup>75</sup> Společnost Sun měla však při zakládání ECIS za to, že nadměrná ochrana počítačových programů povede k monopolistickému trhu a stagnaci inovací.

Výhody network effectu, respektive network effect sám o sobě, by bez interoperability byly značně omezeny. Společnosti by nemohly využívat uživatelské základny interoperabilních programů k rozšiřování své vlastní uživatelské základny. Interoperabilita jakožto další důležitý pojem, včetně výhod a nevýhod z ní plynoucích bude probrána v další kapitole.

---

<sup>74</sup> Samuelson, P. *The Past, Present and Future of Software Copyright Interoperability Rules in the European Union and United States*. European Intellectual Property Review. 2012, č. 4, s. 229

<sup>75</sup> SCOTUS. *Google LLC v. Oracle America, Inc.* Docket for 18-956 [online]. [cit. 26. 2. 2021]. <https://www.supremecourt.gov/docket/docketfiles/html/public/18-956.html>.

### 3 Interoperabilita

Počítačové programy je na jednu stranu nezbytné chránit před kopírováním a jinými zásahy. Na stranu druhou je důležité umožnit dvěma a více počítačovým programům vzájemnou komunikaci, o jejíž zajištění má společnost silný zájem.<sup>76</sup> Interoperabilita umožňuje obchodním společnostem snadnější a rychlejší vstup na trh a má také vliv na rychlost procesu inovace produktů.

Palfrey a Gasser chápou interoperabilitu jako: „*schopnost přenášet a vykreslovat užitečná data a další informace napříč systémy, aplikacemi nebo komponenty*“.<sup>77</sup> Směrnice Evropského parlamentu a Rady, 2009/24/ES ze dne 23. dubna 2009 o právní ochraně počítačových (dále také jako „směrnice o právní ochraně počítačových programů“) ji pak definuje jako schopnost vyměňovat si informace a vzájemně tyto vyměňované informace využívat.<sup>78</sup> Interoperabilita je tedy schopnost systémů spolupracovat, poskytovat si funkcionality, vyměňovat si informace a rovněž tyto informace správně využívat. Toho je dosahováno volbou konkrétní specifikace rozhraní a jejich vzájemnou podobností.<sup>79</sup> Pro interoperabilitu není nutné, aby rozhraní byla stejně implementována. Interoperabilita vyžaduje, aby vyměňované informace byly svým zápisem koncepčně ekvivalentní. Tedy i jiné implementace rozhraní musí mít společnou specifikaci.<sup>80</sup>

Konkrétně pro software je specifické, že jej lze charakterizovat jako systémový produkt, který spojuje několik prvků, které musí spolupracovat, aby došlo k požadovanému výstupu. Interoperabilita u těchto prvků

---

<sup>76</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*. s. 3-4.

<sup>77</sup> Palfrey, J. Gasser, U. *Interop: The Promise and Perils of Highly Interconnected Systems*. Basic Books, 2012, s. 5. Vlastní překlad.

<sup>78</sup> Preambule, recitál 10 Směrnice o právní ochraně počítačových.

<sup>79</sup> Karjala. *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller Symposium: Copyright Protection and Reverse Engineering of Software*, s. 983–986.

<sup>80</sup> Daughtrey, S. C. *Reverse Engineering of Software for Interoperability and Analysis Note*. Vanderbilt Law Review. 1994, č. 1, s. 145–188; Srov. Samuelson. *Are Patents on Interfaces Impeding Interoperability?* s. 4.

je nezbytná.<sup>81</sup> Existuje mnoho výhod, které z interoperability plynou. Důležité je zmínit alespoň tři hlavní výhody, které budou dále rozvedeny v podkapitole 3.3. Těmito výhodami jsou bezesporu podpora inovací, rychlejší rozvoj technologií, podpora hospodářské soutěže anebo podpora zájmů spotřebitelů. Zásahy do autorského díla pomocí zpětného inženýrství nebo jakkoli jinak nelze ani ve světle těchto výhod provádět bez náležitého posouzení ochrany počítačového programu autorským právem. A to ani za účelem interoperability.

Na tomto místě lze například zmínit první soudní spory související s interoperabilitou. Jedná se o dva spory ze Spojených států amerických, a to *Apple vs. Franklin*<sup>82</sup> a *Whelan vs. Jaslow*<sup>83</sup>. V prvním zmíněném sporu společnost Franklin zkopírovala některé části operačního systému Apple s vidinou toho, že jim umožní spouštět stejné počítačové programy, které byly vyvinuty pro operační systém Apple.<sup>84</sup> Franklin se bránil zejména tím, že kopíroval strojový kód a ten vzhledem k nečitelnosti pro člověka není chráněný autorským právem.<sup>85</sup> Soud zde rozhodl, že operační program je chráněn autorským právem a není možné jej bez dalšího kopírovat.<sup>86</sup>

Pro pochopení interoperability a výhod z ní plynoucích je vhodné v následujících podkapitolách uvést rozdíl mezi vertikální a horizontální interoperabilitou. Rozdíl se projeví zejména v zájmu autorů o interoperabilitu a ve výhodách pro uživatele interoperabilních programů.

### 3.1 Vertikální interoperabilita

Vertikální interoperabilita umožňuje sestavit systém nebo produkt z několika navzájem interoperabilních komponent. Díky tomu se společnosti mohou zaměřit na vývoj pouze jimi dodávané komponenty, více se

---

<sup>81</sup> Mylly. *Harmonizing Copyright rules for Computer Program Interface Protection*, s. 886.

<sup>82</sup> *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983).

<sup>83</sup> *Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222 (3d Cir. 1986).

<sup>84</sup> *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983), 1243.

<sup>85</sup> Tamtéž, 1248.

<sup>86</sup> Tamtéž.



specializovat a vyvíjet kvalitnější programy.<sup>87</sup> Společnosti více inovují a snaží se o vývoj co nejdokonalejšího programu.<sup>88</sup> Nesdílení informací o rozhraní by v takovém případě pro společnost nemělo význam, protože by její produkty nebyly schopny fungovat v systému spolu s jinými komponentami. Může se však objevit i snaha jedné společnosti obsáhnout vývoj všech komponent patřících do jednoho systému. V takovém případě se společnost snaží interoperabilitě ostatních programů zabránit proto, aby v co největší míře byly používány její komponenty.<sup>89</sup> Tyto společnosti pak necílí na vývoj jednotlivých komponent, ale na systém jako celek. Výhodu však může být, že je systém sestaven z interoperabilních komponent. Pro společnost je zajímavá možnost navyšování ceny produktů nezbytných pro fungování systému. Zákazník pak nemá možnost, vzhledem k neinteroperabilitě, přejít k produktu konkurence a navýšenou cenu je nucen akceptovat. Vyšší cena může být vyrovnána tím, že nedochází k problémům v kompatibilitě a jsou lépe ošetřena bezpečnostní rizika. Zároveň nedochází k dalším nákladům na kooperaci vícero společností za účelem zajištění interoperability a proces je rovněž plynulejší.<sup>90</sup>

### 3.2 Horizontální interoperabilita

Horizontální interoperabilita umožňuje uživatelům dvou odlišných počítačových programů vyměňovat si mezi sebou data.<sup>91</sup> Nejviditelnější výhodou této interoperability je zejména výhoda plynoucí společností čerstvě vstupujícím na trh. Je pro ně finančně i technologicky méně náročné vyvinout produkt „postavený“ na kompatibilitě s již existujícím, na

---

<sup>87</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 37.

<sup>88</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*, s. 6-7.

<sup>89</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 38.

<sup>90</sup> Tamtéž, s. 39-40.

<sup>91</sup> Tamtéž, s. 25.

trhu rozšířeným produktem.<sup>92</sup> Nově vzniklý produkt tak nebude v postavení, kdy bude muset přesvědčovat uživatele, aby opustili staré prostředí a začali používat nové.<sup>93</sup> To je samozřejmě výhodné i pro společnost která svá rozhraní zpřístupnila, neboť se rozšíří její uživatelská základna. Horizontální interoperabilita významně snižuje riziko tzv. *lock-in efektu*.<sup>94</sup> Výhodou existující interoperability je rovněž to, že se společnosti nemohou zaměřovat na podmanění celého trhu pomocí tvorby vlastních *standardů*, naopak se více soustředí na kvalitu svých produktů, což jim může přinést lepší postavení na trhu.

### 3.3 Výhody interoperability

Hlavní motivací vedoucí k zájmu o informace týkající se specifikace rozhraní, zajišťující interoperabilitu, je zejména zájem na neustálém technologickém vývoji. Interoperabilita má přímý dopad na inovace.<sup>95</sup> Společnosti mohou pomocí interoperabilních programů neustále přidávat funkce k již zavedeným programům, rozšiřovat je, zlepšovat a jinak upravovat. Díky interoperabilitě je umožněno využívat funkcionality jednoho programu v programu druhém.<sup>96</sup> Jedná se o takzvaný kumulativní vývoj se všemi výhodami z něj plynoucími. Vývojář jednoho počítačového programu navazuje na dřívější pokroky jiných vývojářů a staví na nich svoje dílo. Díky tomu se hromadí nové znalosti na znalostech dřívějších.<sup>97</sup> Vývojáři počítačových programů se díky specifikaci rozhraní mohou

---

<sup>92</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 32-33.

<sup>93</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*, s. 6-7.

<sup>94</sup> Pellegrini, R., Rottmann, P., Strieder, G. *Preventing vendor lock-ins via an interoperable multi-cloud deployment approach*, s. 382.

<sup>95</sup> Gasser, U., Palfrey, J. G. *Breaking Down Digital Barriers: When and How ICT Interoperability Drives Innovation* [online]. Rochester, NY: Social Science Research Network, 2008 [cit. 25. 2. 2021]. <https://papers.ssrn.com/abstract=1033226>, s. 12.

<sup>96</sup> Samuelson, P. *Are Patents on Interfaces Impeding Interoperability?* s. 4.

<sup>97</sup> Mylly, U. *Harmonizing Copyright rules for Computer Program Interface Protection*. University of Louisville Law Review, 2009, s. 886.

vyhnout nutnosti psát kód úplně od začátku, což zvyšuje jejich produktivitu.<sup>98</sup> Interoperabilita totiž umožňuje vývoj po malých částech a dovolí zlepšovat postupně již existující produkty.<sup>99</sup> Bez přístupu k informacím o interoperabilitě musí firma, která chce nabízet jen část programu nebo jedinou aplikaci, vyvinout celou platformu, na které aplikace bude fungovat. Zlepšování na trhu dostupných výrobků je dosaženo i pomocí zrychlení procesu vývoje a s tím související snížení nákladů na vývoj, které interoperabilita umožňuje. Ačkoli se velká část pozornosti při řešení otázky interoperability vztahuje na její výhodu v podpoře inovací a technologického pokroku, Van Rooijen zmiňuje i možný opačný dopad na inovativnost společností. Pokud budou společnosti svázány produkty jiných společností, se kterými musejí být interoperabilní, pak nebudou zcela nezávislé a inovace se jim nemusí vyplatit, respektive jim nemusí být umožněna ostatními společnostmi.<sup>100</sup> Nicméně lze konstatovat, že výhody ve větších inovacích a v technologickém pokroku díky interoperabilitě převažují.

Interoperabilita je důležitá také pro udržování konkurenčního prostředí a zamezení společností v získání monopolu k nechráněným myšlenkám počítačového programu a jeho prvků. Společnosti se často snaží ovládnout jeden trh pomocí svého dominantního postavení na jiném trhu tím, že nezpřístupní informace o interoperabilitě. Typické jsou případy prodejců herních konzolí, kteří neumožní svým konkurentům vyvíjet hry spustitelné na těchto konzolích. Důvodem proč na nich nelze hry spustit je nedostatek informací nezbytných pro interoperabilitu.<sup>101</sup> Subjekty, na které může mít nedostatek interoperability dopad jsou společnosti poskytující produkty nebo služby na stejném trhu, respektive společnosti, které mají zájem o vstup na tento trh. I podle Evropské komise je interoperabilita nutná, neboť umožní vstup nových společností

---

<sup>98</sup> Meng, Steinhardt M. S., Schubert, A. *Application Program Interface Documentation: What Do Software Developers Want?* Journal of Technical Writing and Communication 2018, s. 296.

<sup>99</sup> Mylly. *Harmonizing Copyright rules for Computer Program Interface Protection*, s. 886.

<sup>100</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 9.

<sup>101</sup> Např. *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (nited States Courts for the Ninth Circuit, 1992).; *Srov. Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (United States Court of Appeals for the Federal Circuit. 1992).

na trh.<sup>102</sup> Tyto společnosti obvykle nejsou schopny vytvořit celý ekosystém, který bude poskytovat veškeré požadované služby, bez návaznosti, na již existující produkty. Nejzásadnější vliv na trh mají produkty, které se stanou standardem. Nebyla-li by interoperabilita možná u rozhraní, která lze považovat za standard, mohl by jejich autor stanovit tzv. penetrační ceny, což může vést k rozšíření a přijetí horšího rozhraní.<sup>103</sup> Společnostem by měl být poskytnut prostor na vyvinutí produktu interoperabilního s produkty, které na trhu již figurují. Tím je podporována hospodářská soutěž, inovace a pokrok.<sup>104</sup> Zároveň jsou díky novým interoperabilním produktům existující produkty rozšiřovány, doplňovány a vylepšovány. Interoperabilita je tedy přínosná pro obě strany.

Budeme-li se bavit o programech, jež jsou zaměřeny na spotřebitele, neinteroperabilní produkt nebude pro spotřebitele příliš zajímavý. Zde se projeví výhoda interoperability spolu s výše uvedeným network effectem. Gasser a Palfrey zmiňují jako hlavní výhody interoperabilních systémů pro jejich uživatele autonomii, flexibilitu a možnost volby.<sup>105</sup> Horizontální interoperabilita významně snižuje riziko tzv. *lock-in efektu*. Lock-in efekt je stav při kterém není možné produkt z technických (např. právě z důvodu neinteroperability) anebo právních důvodů rozvíjet, upravovat a spojovat s jiným produktem, za pomoci jiného než původního dodavatele. Uživatel takového produktu je tak nucen využívat služby původního dodavatele, což pro něj může znamenat zvýšení nákladů nad rámec reálné hodnoty produktu.<sup>106</sup> Pokud však bude několik

---

<sup>102</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*.

<sup>103</sup> Sagdeo, P. *Application programming interfaces and the standardization-value appropriation problem*. Harvard Journal of Law and Technology, roč. 32, č. 1, 2018, s. 246.

<sup>104</sup> Gasser., Palfrey. *Breaking Down Digital Barriers: When and How ICT Interoperability Drives Innovation*, s. 14.

<sup>105</sup> Tamtéž, s. 15.

<sup>106</sup> Sagdeo. *Application programming interfaces and the standardization-value appropriation problem*, s. 246

produktů navzájem interoperabilních, má mezi nimi uživatel na výběr a nemusí se vázat pouze na produkty jediného dodavatele.<sup>107</sup>

### 3.4 Nevýhody interoperability

I přesto, že výhody interoperability jsou zřejmé, mohou se objevit také některé její nevýhody. Je proto nezbytné nalézt mezi těmito výhodami a nevýhodami interoperability rovnováhu.

Zákonné umožnění „kopírovat“ počítačové programy, respektive jejich rozhraní, by mohlo vést ke ztrátě motivace autorů vytvářet díla další. Tento efekt může nastat tehdy, bude-li vývojář nucen svoje znalosti a schopnosti vložené do produktu ve velkém sdílet s ostatními vývojáři.<sup>108</sup> Vývoj, veškeré myšlenky, znalosti a financování totiž v takových případech přinese původní autor a zpětný inženýr se na jeho práci pouze přiživuje. Ačkoli je podpora inovací jednou z hlavních výhod interoperability, může být v tomto směru interoperabilita v určitých případech také nevýhodou.

Interoperabilita může mít vliv i na bezpečnost produktů. Čím rozsáhlejší je interoperabilita, tím může být bezpečnostní riziko vyšší. Bezpečnostní hrozby se z velké míry šíří rychleji v prostředích, která jsou spolu nějakým způsobem propojená.<sup>109</sup> Interoperabilita svým způsobem umožní jednodušší přístup do jinak uzavřeného počítačového programu. Pomocí rozhraní bude moci kdokoli přistupovat k datům, což může vést k tomu, že se k nim dostanou i lidé neoprávněni. S tím souvisí i nižší míra ochrany soukromí, tedy přístup neoprávněných uživatelů k osobním údajům.<sup>110</sup>

Pokud chtějí zachovat interoperabilitu, jsou vývojáři částečně omezovali v inovacích a kreativě, protože musejí splňovat kritéria, která interoperabilitu dovolují. Tato nevýhoda se neprojevuje pouze v oblasti

---

<sup>107</sup> Pellegrini, R., Rottmann, P., Strieder, G. *Preventing vendor lock-ins via an interoperable multi-cloud deployment approach*. 12th International Conference for internet Technology and Secured Transactions (ICITST), 2017, s. 382.

<sup>108</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 29.

<sup>109</sup> Tamtéž, s. 29.

<sup>110</sup> Gasser., Palfrey. *Breaking Down Digital Barriers: When and How ICT Interoperability Drives Innovation*, s. 16.

konkurenčních vývojářů. Problematický je i rozvoj technologií v rámci jediné společnosti, která je svázána svými dříve vyvinutými produkty, se kterými si přeje zachovat kompatibilitu. Výsledkem tak může být menší pestrost nabízených produktů.<sup>111</sup> V případě, že se některá z platforem dostane do pozice standardu, ostatní společnosti nebudou mít zcela otevřenou možnost volby jiné platformy. Může se stát, že se inovace omezí na ty, které platforma umožní.<sup>112</sup> Standard, který ovládá jedna jediná společnost na trhu vede ke vzniku monopolního trhu. K takové situaci dojde v případě, kdy většina trhu vyvíjí interoperabilní produkty s produktem, jehož rozhraní se tak stane *standardem*.

Interoperabilita může vést k tomu, že systémy „složené“ z několika interoperabilních programů budou náchylnější na chybovost a také nemusí být rychle možné případnou chybu opravit. S tím souvisí možnost vzniku neshod mezi autory interoperabilních programů.<sup>113</sup> Další nevýhodou mohou být vyšší náklady na vývoj. Společnosti jsou nuceny navzájem kooperovat a nemohou tak plně ovlivňovat metodu a technologii, a tedy ani cenu vývoje. Vývoj se rovněž může prodloužit.<sup>114</sup>

### 3.5 Shrnutí

Z výše uvedeného je zřejmé, že interoperabilita je, a to i přes možné nevýhody, jako je například možnost nižší bezpečnosti systému, nezbytným hybatelem technologického pokroku. Důležité tedy je nalézt rovnováhu mezi těmito motivy k zajištění interoperability a mezi zájmy autora na ochranu jeho díla. Postupy související se zajištěním interoperability musí být v souladu s právním řádem. Jedná se zejména o práva duševního vlastnictví a práva hospodářské soutěže. Nesdílení nebo naopak nadměrné sdílení informací o rozhraní, stejně tak jako jejich svévolné získávání, může být pro společnosti nebezpečné. Není totiž zcela jasné, v jakém rozsahu je možné nebo nutné. Ani kdy jejich jednání nebo

---

<sup>111</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 29.

<sup>112</sup> Gasser., Palfrey. *Breaking Down Digital Barriers: When and How ICT Interoperability Drives Innovation*, s. 16.

<sup>113</sup> Tamtéž.

<sup>114</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 29.

opomenutí může být shledáno jako protiprávní. V tomto smyslu se jeví podřazení počítačového programu pod dílo literární chráněné autorským právem jako nešťastná volba. Změna právní konstrukce už dnes nebude zřejmě možná, protože existující vztahy by se mohly dostat do rozporu s novou právní úpravou. Bylo by složité přizpůsobit se jí a trh by ji zřejmě ani nepřijal. Důležité je, aby vývojáři měli jistotu, že mohou vyvíjet a udržovat počítačové programy, které budou interoperabilní s programy ostatními a zároveň měli jistotu, že jejich počítačové programy budou chráněny autorským právem proti kopírování.

## 4 Autorskoprávní ochrana rozhraní a její limity

Vliv rozhraní na interoperabilitu je zřejmý, avšak pojetí ochrany rozhraní v dnešním světě není jednomyslné. Mimo právo autorské mohou mít na jeho ochranu vliv i jiná práva spadající pod práva duševního vlastnictví. Mohlo by se jednat například o obchodní tajemství, patent, průmyslové vzory, ochranné známky, případně i práva jiná.<sup>115</sup> Ačkoli se ochrana počítačového programu, respektive rozhraní může týkat vícero odvětví, tato práce se zaměřuje zejména na právo autorské. Rozhraní je možné vnímat jako autorské dílo, stejně jako ostatní díla může být výsledkem určitého kreativního postupu. V takovém případě by však bylo nutné, aby rozhraní splňovalo požadavky kladené na autorská díla. Rozhraní se příliš neodlišuje od ostatních částí počítačového programu a obvykle je vnímáno jako jeho součást. Ne všechny prvky počítačového programu však jsou uchráněny. Pro tuto práci je důležité dělení autorského díla na chráněné a nechráněné prvky. Toto dělení se vyvinulo postupem času a je v současnosti využíváno vzhledem k ochraně rozhraní. Dělení, které má svůj původ ve Spojených státech je v zásadě založeno na *idea/expression dichotomy*<sup>116</sup> a s tím související doktríně *fair use*. Pro evropský kontinent se dělení vyvinulo z rozlišování mezi formou díla a jeho obsahem. Tedy na jaké oblasti lidské osobnosti autorské dílo působí a jakým způsobem se individualizované prvky promítají do odvozených děl.<sup>117</sup> Obdobou *fair use* je pro nás pak poměrování kolidujících zájmů neboli test *proportionality*.<sup>118</sup> V současnosti je dělení rozhraní na myšlenku a vyjádření zjevné zejména ve směrnici o právní ochraně počítačových

---

<sup>115</sup> Karácsony, B. *The legal Protection of Application Programming Interface*. Master Thesis, Georg-August-University in Göttingen, Faculty of Law. Unpublished manuscript, 2020, s. 10.

<sup>116</sup> Dichotomie myšlenka/vyjádření.

<sup>117</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 210.

<sup>118</sup> Tamtéž, s. 199.



programů.<sup>119</sup> Je důležité zdůraznit, že mezi uvedenými způsoby dělení není zásadní rozdíl. V průběhu let vzniklo rovněž několik testů, které umožňují určit chráněné a nechráněné části počítačového programu, respektive autorského díla obecně.<sup>120</sup>

Bez ohledu na to, zda se budeme zabývat evropským nebo americkým pohledem, je jisté, že bez informací o rozhraní počítačového programu nemůže být interoperabilita zajištěna. Je to právě specifikace rozhraní, která je pro zajištění interoperability nezbytná. Pokud tedy bude rozhraní, respektive jeho specifikace chráněna, ale zároveň k ní nebude umožněn přístup, bude interoperabilita s daným programem omezena. A to nejen ve vztahu mezi jednotlivými vývojáři, ale také ve vztahu původního vývojáře k uživatelům jeho počítačového programu. Ti tak budou nuceni fungovat pouze v ekosystému autora, který neumožnil interoperabilitu.<sup>121</sup> Výhody network effectu mohou být naplno využity teprve tehdy, pokud je zajištěna dostatečná interoperabilita produktů dostupných na trhu.

Na počítačový program je nutné pohlížet odlišně než na jiná autorská díla a omezit technické překážky zajišťující vývojáři monopol na nezveřejněné myšlenky a principy.<sup>122</sup> V rámci nedokonalého přístupu k myšlenkám, na kterých počítačový program stojí, by měl rozsáhlejší autorskoprávní ochranu než jiná díla.<sup>123</sup> Kdyby ve stejném rozsahu bylo zakázáno přistupovat k myšlenkám a principům v knihách, nebylo by je možné číst a ztratily by tak svoji funkci.<sup>124</sup> Myšlenky, jakožto prvky díla nechráněné autorským právem tak musí být přístupné všem bez

---

<sup>119</sup> Čl. 1 odst. 2 Směrnice o právní ochraně počítačových: „Ochrana podle této směrnice se vztahuje na vyjádření počítačového programu v jakékoliv formě. Myšlenky a zásady na kterých je založen kterýkoliv z prvků počítačového programu včetně myšlenek a zásad, na kterých je založeno jeho rozhraní, nejsou chráněny autorským právem podle této směrnice.“

<sup>120</sup> Faust, M. J. *What Do We Do With a Doctrine Like Merger? A Look at the Imminent Collision of the DMCA and Idea/Expression Dichotomy*. Marquette Intellectual Property Law Review, 2008, Volume 12, Issue 1.

<sup>121</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 20.

<sup>122</sup> Přístup k rozhraní a dekompilace jsou detailně řešeny v kapitole 5.

<sup>123</sup> Daughtrey. *Reverse Engineering of Software for Interoperability and Analysis Note*, s. 177-178.

<sup>124</sup> SCOTUS. *Baker v. Selden*, 101 U.S. 99 (1879), 102.

omezení.<sup>125</sup> Nyní je nucen každý nabyvatel dílo dekompileovat a informace spadající pod myšlenky a principy si získávat sám.<sup>126</sup>

Cílem právní úpravy by však mělo být odstranit tento krok a zpřístupnit specifikaci rozhraní spolu s dílem. Správné nastavení právní ochrany je v tomto pohledu klíčové, neboť umožní autorovi chránit své oprávněné zájmy plynoucí z autorství a zároveň je schopné společnosti dovolit užívat výsledky autorovy práce. Právní režim počítačového programu, respektive jeho rozhraní tak má a bude mít významný dopad na další směřování moderních technologií.

V podkapitole 4.1 se budeme nejprve zabývat požadavky na ochranu počítačového programu a jeho prvků. Proto, aby byla počítačovému programu poskytnuta ochrana autorského práva musí splňovat požadavky kladené na autorské dílo. Zásadní v tomto směru je *originalita* díla. Není však dostačující určit si, kdy dílo spadá pod autorskoprávní ochranu, existují totiž rovněž skutečnosti, které autorskoprávní ochranu vylučují nebo alespoň nějakým způsobem omezují. V podkapitole 4.2 tak určíme nechráněné prvky počítačového programu spadající pod *public domain* a rozvedeme doktríny jako *idea/expression dichotomy*, *merger doctrine* nebo například *scènes à faire*. V návaznosti na tuto podkapitolu bude dále přiblížena *funkcionalita* programu a jeho rozhraní, kterou lze řadit vedle myšlenky na které program stojí. Následně bude v podkapitole 4.3 rozebráno dělení rozhraní na *specifikaci a implementaci*, které je pro režim autorskoprávní ochrany zásadní. Vzhledem k tomu, že v Evropské unii je rozhraní, respektive interoperabilita řešena zejména ve směrnici o ochraně počítačových programů, bude v podkapitole 4.6 věnován prostor rozboru této směrnice. A protože softwarové odvětví je globální, bude v poslední části této kapitoly věnován prostor úpravě ochrany rozhraní ve Spojených státech amerických.

### 4.1 Chráněné prvky počítačového programu

Pokud má být počítačovému programu poskytnuta ochrana autorským právem, musí splňovat požadavky na autorské dílo. Autorské dílo je

---

<sup>125</sup> Daughtrey. *Reverse Engineering of Software for Interoperability and Analysis Note*, s. 179.

<sup>126</sup> Weston, S. *Improving interoperability by encouraging the sharing of interface specifications*. *Law, Innovation and Technology*. 2017, č. 1, s. 81–82.

chápáno jako výsledek tvůrčí činnosti autora, je s ním provázáno a svým způsobem je odrazem jeho osobnosti.<sup>127</sup> Má-li se jednat o dílo ve smyslu autorského práva, musí být vyjádřeno tak, že bude dostatečně přesně a objektivně identifikovatelné.<sup>128</sup> Neméně důležitý je požadavek originality, respektive původnosti. Počítačový program může být chráněn jako dílo autorské, pokud je původní a jedinečný. Původnost vyplývá ze spojení autora a díla, dílo je jeho vlastním duševním výtvořem. Jedinečnost díla, respektive autorskoprávní individualita není splněna u díla, které je schopno vytvořit nezávisle na sobě několik autorů.<sup>129</sup> Takové případy se týkají zejména děl abstraktních, existujících na základě předem daných skutečností, bez autorova individuálního přispění.<sup>130</sup> Rovněž tak nemůže být chráněno dílo vytvořené mechanicky nebo rutinně. Tomu tak je zejména z důvodu nezbytnosti tvůrčího procesu při tvorbě autorského díla. Při absenci tvůrčího procesu nelze na rozhraní pohlížet jako na autorské dílo.<sup>131</sup> Takový tvůrčí proces u běžné specifikace rozhraní nemusí být vůbec přítomen. V případě čistě technické novosti či inovativnosti, bez zásadní formy tvůrčí činnosti může být dílo chráněno patentem, jako užitečný vzor a jinak. I v případě patentů je však nezbytné splnit podmínku vynálezecké činnosti. Vynález nesmí pro odborníka vyplývat zřejmým způsobem ze stavu techniky.<sup>132</sup>

Pro to, aby bylo na počítačový program, včetně jeho součástí, tedy i rozhraní, pohlíženo jako na autorské dílo, je nutné, aby splňoval požadavky autorského práva na počítačové programy. Dlouhou dobu<sup>133</sup> byl českým právním řádem chráněn pouze počítačový program, který splňoval veškeré požadované pojmové znaky autorského díla.<sup>134</sup> Tím bylo značně omezeno množství chráněných programů, neboť tvůrčí úroveň počítačových programů již ze své podstaty nelze srovnávat s díly

---

<sup>127</sup> Koukal, P. *Autorské právo, public domain a lidská práva*, Masarykova univerzita, 2019, s. 33.

<sup>128</sup> Rozsudek soudního dvora ze dne 13. listopadu 2018 ve věci C-310/17, bod 40.

<sup>129</sup> Koukal, P. *Autorské právo, public domain a lidská práva*, Masarykova univerzita, 2019, s. 42-43.

<sup>130</sup> Tamtéž, s. 60-61.

<sup>131</sup> Telec. Tůma. *Autorský zákon, Komentář. 2. vydání*, s. 7-71, bod. 2.

<sup>132</sup> Chloupek, Hartvichová a kol. *Patentový zákon, 1. vydání*. C. H. Beck, 2016, s. 40-48.

<sup>133</sup> Do přijetí AZ (účinnost 1. 12. 2000).

<sup>134</sup> Telec. Tůma. *Autorský zákon, Komentář. 2. vydání*, s. 7-71, bod 25.

uměleckými.<sup>135</sup> Na rozšíření ochrany se zaměřil nový autorský zákon v návaznosti na harmonizační snahy Evropské unie. Margoni shrnuje, že nyní lze na požadavek originality podle Evropského práva pohlížet tak, že je dostačující autorova vlastní intelektuální tvorba.<sup>136</sup> Autorským zákonem jsou nadále chráněny počítačové programy, které splňují pojmové znaky autorského díla stanovené v § 2 odst. 1.<sup>137</sup> V tomto ustanovení nejsou brána v potaz specifika počítačových programů a jejich odlišnost od jiných děl.<sup>138</sup> AZ však nově chrání i počítačové programy, které samy o sobě nenaplnují pojmové znaky a nejsou tak autorskými díly, ale které splňují kritérium původnosti.<sup>139</sup>

Na počítačové programy pohlíží AZ v souladu s mezinárodně uznávaným pohledem, jako na literární dílo.<sup>140</sup> Zmíněné kritérium původnosti je tedy dnes pro zajištění ochrany autorským právem dostačující.<sup>141</sup> Proto aby bylo dílo autorovou intelektuální vlastní tvorbou, musí mít autor možnost dílo vytvořit svými svobodnými a kreativními rozhodnutími s osobním přispěním.<sup>142</sup> To, jestli byl počítačový program vytvořený tvůrčí duševní činností a zda se jedná o činnost tzv. *původní* lze určit na základě jeho struktury a uspořádání (posloupnost příkazů, organizace, uspořádání příkazů aj.).<sup>143</sup> Původnost lze určit i podle toho, jak daný program komunikuje s uživatelem, jak je uživatelem vnímán způsob práce s programem. Nejedná se však o přímo o grafická (GUI), zvuková a jiná vyjádření programu. Ta jsou chráněna jiným způsobem.<sup>144</sup> Nicméně SDEU rozhodl, že i na grafické uživatelské rozhraní se může vztahovat

---

<sup>135</sup> Telec. Tůma. *Autorský zákon, Komentář. 2. vydání*, s. 7-71, bod 25.

<sup>136</sup> Margoni, T. *The Harmonisation of EU Copyright Law: The Originality Standard*. Centre for IT & IP Law (CiTiP), Faculty of Law - KU Leuven; University of Glasgow - School of Law, 2019, s.13.

<sup>137</sup> § 2 odst. 1 AZ.

<sup>138</sup> Telec. Tůma. *Autorský zákon, Komentář. 2. vydání*, s. 7-71, bod 25.

<sup>139</sup> Tamtéž.

<sup>140</sup> Např. Čl. 10 odst. 1 Dohody o obchodních aspektech práv k duševnímu vlastnictví (Dohoda TRIPS).

<sup>141</sup> AZ.

<sup>142</sup> Margoni, T. *The Harmonisation of EU Copyright Law: The Originality Standard*, s.13.

<sup>143</sup> Telec. Tůma. *Autorský zákon, Komentář. 2. vydání*, s. 7-71, bod 26.

<sup>144</sup> Tamtéž.

autorskoprávní ochrana.<sup>145</sup> Původnost počítačového programu je tak obecně vykládána mnohem širěji než u jiných autorských děl. Jak je uvedeno výše, program je obvykle považován za autorské, tedy původní dílo i tehdy, pokud individuální prvek náleží „pouze“ v autorovu uspořádání a struktuře kódu. Kód samotný nemusí být původní.<sup>146</sup>

Požadavek původnosti je například ve směrnici definován jako „*original*“.<sup>147</sup> Originalita se tak stává zásadním konceptem při definování, která rozhraní dosáhnou na ochranu. V minulosti byl požadavek originality vykládán a definován členskými státy odlišně. Některá díla tak v jednom státě chráněna byla, kdežto v jiném nikoli. Například pro dosažení autorskoprávní ochrany v Anglii bylo dostatečné, pokud dílo vycházelo z dovednosti autora a ten k jeho dosažení vynaložil práci (skill and labor).<sup>148</sup> Těmto problematickým situacím se snaží Evropská unie v posledních letech předcházet, definuje ideální úroveň originality a případné výjimky. Někteří autoři se obávali toho, že bude ochrana vynucená právem Evropské unie příliš široká.<sup>149</sup> Na druhé straně se někteří obávali, že většina programů díky zásahům Evropské unie chráněna nebude, protože na trhu je stále častější standardizace a automatizace a originalita je upozad'ována.<sup>150</sup> Konkrétně ve směrnici Evropského parlamentu a Rady 2009/24/ES ze dne 23. dubna 2009 o právní ochraně

---

<sup>145</sup> Rozsudek Soudního dvora ze dne 22. prosince 2010, C-393/09, *Bezpečnostní softwarová asociace – Svaz softwarové ochrany v. Ministerstvo kultury*, výrok rozhodnutí.

<sup>146</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 58.

<sup>147</sup> Directive 2009/24/EC of the European Parliament and the Council of 23 April 2009 on the legal protection of computer programs, Article 1 par. 3: „*A computer program shall be protected if it is original in the sense that it is the author's own intellectual creation.*“

<sup>148</sup> Mylly. *Harmonizing Copiright rules for Computer Program Interface Protection*, s. 891.

<sup>149</sup> Broy, M. Lehmann, M. *Die Schutzfdhigkeit von Computerprogrammen nach dem neuen europitischen und Deutschen Urheberrecht*. 94 Gewerblicher Echissschutz und Urheberrecht, 1992.

<sup>150</sup> Lesshaft, K. Ulmer, D. *Urheberrechtliche Schutzwiardigkeit und Tatsachliche Schutzflthigkeit von Software*. 9 Computer und Recht, 1993.

počítačových programů je originalita již chápána jako původnost:<sup>151</sup> „Počítačový program je chráněn, pokud je původní, v tom smyslu, že je vlastním duševním výtvorem autora. Pro stanovení způsobilosti k ochraně není uplatňováno žádné jiné kritérium.“<sup>152</sup> Ani tato definice však není dostačující. Ačkoli směrnice požadavky svým způsobem harmonizuje, nelze zcela překlenout výklad jednotlivých států a mezery plynoucí z jazykové bariéry.<sup>153</sup> Mylly se domnívá, že směrnice může ohrozit soudržnost a konzistenci vnitrostátních systémů autorských práv. Některé státy se zvýšením či snížením nároků na počítačové programy nesouhlasí a pokračují v jimi zavedené autorskoprávní tradici.<sup>154</sup> V důsledku toho nebylo na evropské úrovni dosaženo harmonizace hladce.<sup>155</sup> Originalita tedy v podstatě není stoprocentně definovaný pojem a záleží na jednotlivých státech, jak originalitu blíže definují a zejména interpretují. Vzhledem k několikajazyčnému vydávání předpisů nelze spoléhat na to, že každý překlad bude korespondovat se skutečným záměrem Evropské unie. Sjednocení vnitrostátních právních řádů je dosahováno postupně zejména rozhodovací praxí SDEU<sup>156</sup> Originalita je pro ochranu rozhraní podstatná hlavně co se týče implementace rozhraní, jak bude zmíněno později. Problémům spojeným s výkladem se snažil vyhnout Soudní dvůr tak, že si pro interpretaci předpisů vybral komparativní a systémový výklad všech jazykových verzí konkrétního předpisu.<sup>157</sup> I přes veškeré snahy nadále existuje rozpor v chápání originality v unijních předpisech. Nadále je originalita vykládána jako původnost, jedinečnost ale i originalita.<sup>158</sup> Některé státy, jako například Francie či Nizozemsko, pojímaly za originální dílo takové, které nemůže být nebo je alespoň nepravděpodobné, že by bylo

---

<sup>151</sup> Zibner, J, Originalita v pojetí práva evropské unie. *Revue pro právo a technologie* [Online]. 2017, č. 15. <https://journals.muni.cz/revue/article/view/6651>, s. 231-232.

<sup>152</sup> Článek 1 odst. 3 Směrnice o právní ochraně počítačových programů.

<sup>153</sup> Cuillou. *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*, s. 542.

<sup>154</sup> Mylly. *Harmonizing Copiright rules for Computer Program Interface Protection*, s. 907.

<sup>155</sup> Tamtéž, s. 880.

<sup>156</sup> Margoni. *The Harmonisation of EU Copyright Law: The Originality Standard*, s.13.

<sup>157</sup> Zibner. *Originalita v pojetí práva evropské unie*, s. 221.

<sup>158</sup> Tamtéž, s. 220.

vytvořeno vícero autory dvakrát ve více či méně identické podobě.<sup>159</sup> Například Anglie, vždy vyžadovala nižší míry osobní kreativity. Na druhé straně pak stojí například právo německé. Spoor zdůrazňuje, že: „*Německé právo týkající se softwaru vyžaduje kreativní vstup, který je nad běžné dovednosti programátora.*“<sup>160</sup> Takto rozdílné požadavky není zcela jistě možné sjednotit přijetím jediné směrnice, která sama o sobě dává značný prostor pro pochybení v implementaci svého znění do právních rádu jednotlivých států. Podle Zibnera se však dnes pojmy originalita a původnost čím dál více sbližují.<sup>161</sup>

Například ve Spojených státech je originalita Nejvyšším soudem pojímána jako „*určitá minimální úroveň kreativity vykazovaná dílem*“.<sup>162</sup>

Jedním z možných způsobů, jak provést test originality je dle Zibnera v prvním kroku izolace funkčních prvků díla. Dále je nutné posuzovat, zda autor vůbec měl možnost vytvořit originální dílo a zda měl možnost svoji svobodu náležitě vyjádřit. Posledním krokem je nutnost posoudit výsledné dílo jako výsledek duševní tvůrčí činnosti autora.<sup>163</sup>

Z výše uvedeného plyne, že rozhraní musí splňovat požadavek originality. Důležité tedy je, jaké kritérium je pro dosažení originality nastaveno. V tomto směru může dopomoci rozhodnutí SAS v. World Programming Ltd, ve kterém Soudní dvůr posuzoval originalitu jak software, tak i příručky k software. Podle SDEU sice pod ochranu autorského práva spadá zdrojový i strojový kód, ale programovací jazyk, funkce ani formát datových souborů nejsou vyjádřením, a pod ochranu nespádají.<sup>164</sup> Soudní dvůr v rozhodnutí odůvodnil, proč je příručka, respektive dokumentace k software chráněna autorským právem. Dle Soudního dvora je důležité v tomto případě naplnit znak originality, čehož může být dosaženo i „pouhým“ výběrem, kombinací a uspořádáním obsahu. Hlavní

---

<sup>159</sup> Spoor, J. H. *Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive Symposium: Copyright Protection and Reverse Engineering of Software*. University of Dayton Law Review. 1993, č. 3, s. 1065.

<sup>160</sup> Tamtéž, s. 1066.

<sup>161</sup> Zibner. *Originalita v pojetí práva evropské unie*, s. 251.

<sup>162</sup> SCOTUS. *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co. Inc.*, 499 U.S. 340, (1991), 359-360.

<sup>163</sup> Zibner. *Originalita v pojetí práva evropské unie*, s. 253.

<sup>164</sup> Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, *SAS Institute Inc. v. World Programming Ltd*, výrok 1).

kritérium dle něj je autorova vlastní intelektuální tvorba.<sup>165</sup> Kód je originální a je výtvozem autora, pokud autor vybral vyjádření z dostatečného počtu možností.<sup>166</sup>

Jedná se o příliš jednoduchý test a obsáhlou množinu originálních děl. Logicky by tak rozhraní, které může být tvořeno z více variant bylo originální. Specifikace rozhraní, která je podřazena pod myšlenky však chráněna být nemůže. Zároveň je volba specifikace rozhraní limitována funkcemi, které rozhraní musí splňovat, stejně tak jako standardy na trhu. Při vývoji programů je zároveň běžně využíváno známých technologických postupů. Díky tomu je při programování obecně snížena úroveň originality.<sup>167</sup> To je ještě znatelnější u rozhraní, které je omezeno požadavky na interoperabilitu.<sup>168</sup> Rozhraní je tedy vytvářeno spíše v souladu s technickými postupy, standardy a bezpečnostními požadavky, které je nutné následovat. Volba autora při vývoji rozhraní je omezena a dána jejich technickou funkcí.<sup>169</sup> V případě jednoduchých rozhraní tak nebude nereálná situace, kdy dva autoři napíší kód rozhraní stejně.<sup>170</sup> Byť by však existovalo vícero možností, jak specifikaci rozhraní vyjádřit, stále bude nutné brát v úvahu, že je někdy nutné vybrat tu konkrétní možnost, která je nutná k dosažení daného účelu. Pokud autor zamýšlí zachovat interoperabilitu, je nutné implementovat rozhraní v souladu s jeho specifikací.<sup>171</sup>

---

<sup>165</sup> Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, *SAS Institute Inc. v. World Programming Ltd*, bod 67.

<sup>166</sup> Gervais D. *The TRIPS Agreement: Drafting History and Analysis (Perspective on Intellectual Property Law)*, 2ND ed. Sweet & Maxwell, 2003, s. 134; Srov. Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 55.

<sup>167</sup> Commission of the European Communities, *Proposal for a Council Directive on the legal protection of computer programs*, 1989, COM(88) 816 final - SYN 183, s. 6.

<sup>168</sup> Santoro, M. a kol. *Web Application Programming Interface: general-purpose standards, terms and European Commission initiatives*. European Commission, JRC, 2019, s. 8, 33, 36.

<sup>169</sup> Rozsudek Soudního dvora ze dne 22. prosince 2010, C-393/09, *Bezpečnostní softwarová asociace – Svaz softwarové ochrany v. Ministerstvo kultury*, bod 48.

<sup>170</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 56.

<sup>171</sup> Tamtéž, s. 58.



Autorské právo je založené na originalitě. U standardních děl je to podmínka vhodná. U rozhraní však místo několika originálních děl přináší společnosti větší užitek zajištění interoperability, a tedy používání neoriginálních, stejných specifikací rozhraní. Funkční charakter rozhraní je důležitější než funkční charakter počítačových programů obecně. Vezzoso jako další argument vnímá obecnou podporu interoperability vyjádřenou směrnicí.<sup>172</sup> Není zcela jisté, jakou hladinu originality je pro rozhraní, zejména pro jeho implementaci, nezbytné splňovat, aby mohlo být shledáno originální a mohlo tak být chráněno autorským právem. U rozhraní, která nelze implementovat mnoha způsoby se myšlenka a vyjádření stávají neoddělitelnými.<sup>173</sup> V případě specifikace rozhraní bude originalita na hraně z důvodu její funkčnosti a standardizace. Proto nelze specifikace rozhraní mnohdy vyjádřit vícero způsoby.<sup>174</sup> Rozhraní, která nesplňují požadavek originality, nemohou být považována za dílo v kontextu režimu autorského práva.

Z výše uvedeného je zřejmé, že implementace rozhraní může být chráněna autorským právem, bude-li originální. Tomu tak nebude zejména v případech malého množství způsobů vyjádření, které autor pro implementaci má. U specifikace rozhraní nebude originalita z důvodu její technické a funkční předurčenosti možná. I tehdy, je-li počítačový program shledán originálním a splňuje veškeré náležitosti autorského díla, nemusí nutně podléhat ochraně. V následné kapitole budou rozebrány nechráněné prvky díla.

## 4.2 Nechráněné prvky počítačového programu

Není dostačující určit si, kdy dílo spadá pod autorskoprávní ochranu, existují totiž rovněž skutečnosti, které autorskoprávní ochranu vylučují nebo alespoň nějakým způsobem omezují. Může se jednat o převážení

---

<sup>172</sup> Vezzoso, S. *Copyright, Interfaces, and a Possible Atlantic Divide*. SSRN [online]. [cit. 25. 2. 2021]. [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2161487](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2161487), s. 159.

<sup>173</sup> Rozsudek Soudního dvora ze dne 22. prosince 2010, C-393/09, *Bezpečnostní softwarová asociace – Svaz softwarové ochrany v. Ministerstvo kultury*, bod 49.

<sup>174</sup> *Novitaire Inc v Easyjet Airline Company*, In: Van Rooijen, A. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 76-77.

zájmů autora nad zájmy jiných subjektů, například veřejným zájmem. V případě rozhraní se jedná zejména o zájem na dalším technickém pokroku a inovacích umožněných interoperabilitou. Pakliže by se v případě počítačových programů jednalo o ochranu celého programu, a to včetně specifikace rozhraní, nebyl by technický pokrok dost dobře umožněn. Proto jsou například z ochrany vyloučeny myšlenky, principy, teorie a vzorce, na kterých počítačový program funguje.<sup>175</sup> Ostatně ani u jiných forem autorských děl tomu není jinak. Z toho plyne, že dílo nemusí z hlediska ochrany tvořit jeden celek, je možné a často i žádoucí jednotlivé části oddělovat a chránit pouze ty, které dle autorského zákona chráněny být mají. Funkcionalitu počítačového programu lze podřadit pod nechráněné myšlenky.<sup>176</sup> Specifikace rozhraní je určena svojí funkcí, a proto ji nelze chránit autorským právem. V opačném případě by mohly být zvýhodněny zájmy autora počítačového programu.

V případě proti sobě stojícího práva autora na ochranu jeho díla a práv společnosti včetně zájmů veřejných, je nutné protichůdná práva vyvažovat.<sup>177</sup> Zájmy autora jsou zřejmé, jedná se většinou o zájmy majetkové. Minimálním cílem autora je zaplacení nákladů, které vynaložil na vytvoření počítačového programu. K nákladům si autoři připočítají zisk přiměřený k individualitě a hodnotě programu pro společnost.<sup>178</sup> Neméně důležité mohou být pro autora zájmy osobnostní, a to zejména spojení jeho osoby s programem samotným. Oproti zájmům autora pak stojí zájem veřejnosti a společnosti na užívání díla.<sup>179</sup> Takovýto veřejný zájem je mnohdy opodstatněn zájmem o vědecký pokrok nebo například podporu umění. V případě rozhraní jde o podporu vývoje počítačových programů a zabránění společností v získání monopolního postavení. Z těchto důvodů jsou zákonodárcem obvykle z autorskoprávní ochrany vyloučeny abstraktní prvky, skutečnosti existující nezávisle na lidském vědomí a zejména pro nás důležité myšlenky a principy počítačových programů a myšlenky jejich rozhraní.<sup>180</sup>

---

<sup>175</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 97.

<sup>176</sup> Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, *SAS Institute Inc. v. World Programming Ltd*, bod 46.

<sup>177</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 154.

<sup>178</sup> Tamtéž, s. 155-156.

<sup>179</sup> Menell. *Rise of the API Copyright Dead: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, s. 464.

<sup>180</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 155.

Je evidentní, že pomyslná miska vah se nesmí příliš naklonit na žádnou stranu. Mohlo by dojít k tomu, že by autoři nebyli dostatečně motivováni nové programy vytvářet, protože by pro ně byl ekonomický přínos vzhledem k rozsahu ochrany zanedbatelný. Naopak přílišné zamezení přiměřeného a odůvodněného obecného užívání by vedlo k neinteroperabilitě a s tím související efekt by byl opět snížený ekonomický přínos pro autora doprovázený negativním vlivem na trh.

Jak je uvedeno v předchozí podkapitole, právní předpisy nám jasně definují požadavky na to, aby na dílo mohlo být pohlíženo jako na dílo autorské. Nejsou-li splněny tyto požadavky, dané dílo není dílem autorským a ochrana autorským právem mu nenáleží. Neméně důležité však v tomto směru jsou výjimky tzv. autorskoprávně volné složky díla a jejich právní režim. Do režimu *public domain* (volná složka díla) podle Koukala spadají abstraktní prvky a skutečnosti existující nezávisle na lidském vědomí. Rovněž sem spadají výjimky stanovené v zákoně. Stejně tak jako myšlenky a principy, na kterých jsou postaveny počítačové programy a jejich rozhraní.<sup>181</sup> Vzhledem k nemožnosti obsáhnout všechny případy dopomáhá interpretaci často soud.<sup>182</sup>

Všechna díla jsou z větší nebo menší části tvořena na základě přírodní a sociální zkušenosti člověka. Pro určení autorskoprávní ochrany je důležitý poměr mezi využitím těchto zkušeností a samotným tvůrčím postupem. Tedy myšlenky, náměty a principy počítačového programu chráněny být nemohou, nebude-li k nim přidána i jakási tvůrčí vrstva, které je doplňuje a rozšiřuje. Do těchto prvků obecné povahy musejí zákonitě spadat i veškeré přírodní, sociální, historické informace a skutečnosti, u kterých je v zájmu společnosti, aby přístup k nim do budoucna nebyl omezován např. právě autorským právem.<sup>183</sup> Pro účely této práce jsou nejdůležitějšími prvky, které se řadí do kategorie *public domain*, kam spadají i myšlenky, principy a zásady na kterých stojí počítačový program, respektive rozhraní. Důvodem, proč jsou některé části díla nebo některá díla celkově nechráněna je u myšlenek, námětů a principů zejména jejich přílišná abstraktnost, kdy nelze nalézt v takovém díle dostatečné množství individuality autora, které je pro ochranu nezbytné. Jsou to díla, která jsou obecná, opakující se, generická apod. která mají

---

<sup>181</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 438-439.

<sup>182</sup> Např. Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, *SAS Institute Inc. v. World Programming Ltd*; Srov. *Oracle America, Inc v. Google Inc.*

<sup>183</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 90-91.

sloužit k obecnému užití, nikoli k prospěchu jednotlivce.<sup>184</sup> V opačném případě by pak např. řecký autor, který přišel s divadelní komedií, omezil ostatní tvůrce v psaní stejného žánru a komedii by se nedostalo takového rozmachu. Oproti tomu někteří autoři zastávali názory, že i myšlenky, principy a nápady jsou výsledkem duševní činnosti a jako takové by měly být chráněny autorským právem.<sup>185</sup>

U počítačových programů je toto dělení ještě konkrétnější, protože společnost prostřednictvím zákonodárce má velký zájem na tom, aby nebyly myšlenky (a tedy i pokrok v oboru IT) počítačového programu drženy v rukou jednotlivců, nýbrž aby bylo možné s nimi pracovat a posouvat se tak technologicky kupředu. To však nebrání tomu, aby mohla být počítačová díla chráněna, neboť počítačový program, odmyslíme-li si od něj myšlenky a principy na kterých je postaven, je dostatečně individuální autorské dílo a ochranu si tak zaslouží ve stejném rozsahu jako jiná díla.<sup>186</sup> I z toho důvodu je zákonem stanovena fikce, která jej připodobňuje k dílu literárnímu.

Soud například navíc stanovil<sup>187</sup>, že některé prvky počítačového programu patří do veřejné sféry, pokud je jejich zdrojový kód, respektive vyjádření v počítačovém průmyslu běžné nebo standardní.<sup>188</sup> Jak soudy, tak zákonodárci se tedy přiklonili k menší míře ochrany ve prospěch inovací a hospodářské soutěži. Základním principem je, že je možné na základě jednoho počítačového programu (jeho myšlenek a funkcionalit) vytvořit počítačové programy jiné, avšak nesmí dojít k přímému kopírování. Dle některých autorů se jedná v podstatě o směnu, kdy společnost nabízí autorovi ochranu vyjádření, tedy jakýsi monopol k němu, a na oplátku požaduje volný přístup k myšlenkám.<sup>189</sup>

Public domain tedy lze chápat jako soubor prvků, které nespádají do skupin soukromých ani kolektivních statků a lze je tak obecně užívat. Jak tvrdí Koukal: „*Jestliže zákonodárce stanoví, že určitý statek může každý užít za*

---

<sup>184</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 201-202.

<sup>185</sup> Např. Diderot. In: Koukal. *Autorské právo, public domain a lidská práva*, s. 123.

<sup>186</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 66.

<sup>187</sup> *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), 709-710.

<sup>188</sup> Mylly. *Harmonizing Copyright rules for Computer Program Interface Protection*, s. 905.

<sup>189</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 42.

*podmínek stanovených zákonem (princip rovnosti), znamená to, že v rozsahu takového užití uživatel realizuje svou svobodu (§ 3 odst. 1 ObčZ) a nezasahuje ani do práv ke statkům soukromým, ani do práv ke statkům kolektivním.*<sup>190</sup>

Zde je příhodné doplnit, že princip volného užití by nebyl naplněn v případě, že by ochrany autorsky nechráněného díla bylo možné dosáhnout jinou formou, např. pomocí obchodního tajemství nebo patentu. Pokud jeden zákon, některé prvky přímo vyloučí z ochrany z důvodu veřejného zájmu, zajisté není ve veřejném zájmu, a tedy ani v zájmu zákonodárce, tyto prvky chránit zákonem jiným. Bez přístupu k takovýmto prvkům by bylo přinejmenším velice obtížné vytvářet nová díla, protože jak již bylo zmíněno výše, tvorba nových děl je vždy založena na zkušenosti získané z obecného fondu a zároveň na autorem přidáných individuálních prvcích.<sup>191</sup>

#### 4.2.1 Dichotomie myšlenky a jejího vyjádření

Mimo to, zdali jsou specifikace rozhraní anebo implementace rozhraní originální, a tedy způsobilé ochrany autorským právem, je pro jeho ochranu zásadní, zda bude konkrétní prvek rozhraní podřazen pod chráněné vyjádření autorova díla nebo nechráněnou myšlenku na které dílo stojí. Důležité je, že na rozdíl od patentu není autorská ochrana tolik striktní<sup>192</sup>, co se předmětu ochrany týče a ve značném rozsahu se v ní uplatňují výše uvedené výjimky na základě tzv. public domain. Díky tomu je nadále umožněn vývoj nových podobných, vylepšených či upravených programů pracujících na stejném principu či myšlence. Již ve Smlouvě Světové organizace duševního vlastnictví o právu autorském byla vnímána potřeba omezit autorskoprávní ochranu počítačového programu, tedy vyjmout z ní myšlenky a principy, které mají být přístupné všem programátorům a mají tak přispět k rozvoji technologií.<sup>193</sup> Z historického pohledu se dá tato problematika řešit rozlišením formy a obsahu. Již podle Fichte je chráněna forma vyjádření myšlenek a obsahu, nikoliv obsah samotný, ten totiž má sloužit celé společnosti. Při přečtení

---

<sup>190</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 144.

<sup>191</sup> Tamtéž.

<sup>192</sup> Tamtéž, s. 216: „*Idea-expression dichotomy se původně vztahovala na vyloučení technických myšlenek, které mohou být chráněny právem patentovým.*“

<sup>193</sup> Tamtéž, s. 214.

knihy si její obsah čtenář ukládá ve své mysli a nelze tvrdit, že tento obsah, tedy myšlenky čtenáře patří stále autorovi díla.<sup>194</sup>

Myšlenky, principy, postupy anebo metody poprvé na mezinárodní úrovni z ochrany vyloučila dohoda TRIPS<sup>195</sup>. Ne náhodou byla tato ustanovení vytvořena právě kvůli problematice počítačových programů. Jedná se však nadále o ne zcela jasné a určité omezení, neboť interpretací mohou jednotlivé státy a jejich soudní orgány dojít k jiným závěrům a poskytovat tak stejným dílům, respektive jejich částem odlišnou ochranu.

Možnou inspirací pro smluvní strany Dohody TRIPS byl čl. 102 b) amerického Copyright Act of 1976<sup>196</sup>, dle kterého myšlenky, postupy, procesy, systémy, metody, koncepty nebo objevy, bez ohledu na formu, v nichž jsou popsány, vysvětleny, znázorněny či zachyceny nejsou chráněny. Existuje ústavní rovina zdánlivě protichůdných práv a doktrína *idea-expression dichotomy* si dává za cíl ji alespoň částečně vyřešit. Pokud tedy Dohoda TRIPS, byť částečně, byla inspirována právě americkým Copyright Act, pak i ona se snaží vyřešit tato protichůdná práva. Tento cíl je patrný z čl. 7 TRIPS: „Ochrana a dodržování práv k duševnímu vlastnictví by měly přispět k podpoře technických inovací a k převodu a rozšiřování technologie, ke vzájemným výhodám výrobců a uživatelů technických znalostí způsobem, přispívajícím k sociálnímu a ekonomickému blahobytu a k rovnováze práv a povinností.“<sup>197</sup>

S podobnou dělbou souhlasí i smluvní strany Smlouvy Světové organizace duševního vlastnictví o právu autorském, která stanovila, že „ochrana autorského práva se vztahuje na vyjádření, nikoli však na myšlenky, postupy, způsoby fungování nebo matematické pojmy jako takové.“<sup>198</sup> Zákonem takto definovaná ochrana v podstatě dává soudům možnost korigovat trh a chránit jej před zneužíváním tržní síly. Je totiž

---

<sup>194</sup> Fichte, J. G. *Beweis der Unrechtmäßigkeit des Büchernachdrucks* [online]. Ein Rasonnement und eine Parabel, 1793. [http://www.copyrighthistory.org/cam/tools/request/showRecord?id=record\\_d\\_1793](http://www.copyrighthistory.org/cam/tools/request/showRecord?id=record_d_1793), s. 450 In: Koukal. *Autorské právo, public domain a lidská práva*, s. 203.

<sup>195</sup> Čl. 9 odst. 2 TRIPS.

<sup>196</sup> Prchal, P. *Limity autorskoprávní ochrany*. Praha: Leges, 2016, s. 58 a násl. In: Koukal. *Autorské právo, public domain a lidská práva*, s. 212.

<sup>197</sup> Čl. 7 TRIPS.

<sup>198</sup> Článek 2 Smlouvy Světové organizace duševního vlastnictví o právu autorském, přijaté v Ženevě dne 20. prosince 1996.

na soudech, aby dichotomii v konkrétních případech uplatňovaly a určily směr kterým se bude ochrana počítačových programů, respektive rozhraní nadále ubírat. Soudy postupem let zdokonalují své postupy při zajišťování ochrany autorského práva. Z jednotlivých soudních sporů vznikly různé srovnávací, filtrační a jiné testy. Často používaná je například tzv. *merger doctrine* probraná v podkapitole 4.2.2. Pokud nejde vyjádření, které by jinak bylo chráněno autorským právem, oddělit od faktů, myšlenek nebo funkcí obsažených ve vyjádření, pak ochrana autorským právem není možná. Tyto dlouhodobé zásady nejenže vymezují předmět a rozsah výlučných práv chráněných autorskými zákony, ale také odlišují ochranu autorských práv od ochrany patentů pro počítačové programy.<sup>199</sup>

V rámci Evropské unie omezila ochranu na vyjádření počítačového programu Směrnice č. 91/250/EHS. Ochrana se týká například právě zdrojového kódu, strojového kódu či jiné formy vyjádření umožňující dílo kopírovat. SDEU takto pojal ochranu například v případě *Bezpečnostní softwarová asociace – Svaz softwarové ochrany proti Ministerstvu kultury*.<sup>200</sup> Samotná směrnice o právní ochraně počítačových programů výslovně odkazuje na dichotomii myšlenky a vyjádření na nichž stojí rozhraní.<sup>201</sup> Ne jinak je tomu ve Spojených státech amerických, kde autorský zákon stanoví, že: „V žádném případě se ochrana původního autorského díla nevztahuje na jakoukoli myšlenku, postup, proces, systém, způsob fungování, koncept, princip nebo objev, bez ohledu na formu, v níž je popsána.“<sup>202</sup>

To, které prvky počítačového programu jsou chráněny a které nikoli bylo na půdě Evropské unie diskutováno zejména ve sporu *SAS Institute v. World Programming Ltd.*<sup>203</sup> Soudní dvůr pracoval s dichotomií a rozhodl o tom, že prvky díla jako programovací jazyk a funkcionality

---

<sup>199</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 41.

<sup>200</sup> Rozsudek Soudního dvora ze dne 22. prosince 2010, C-393/09, *Bezpečnostní softwarová asociace – Svaz softwarové ochrany v. Ministerstvo kultury*, bod 48.

<sup>201</sup> Mylly. *Harmonizing Copiright rules for Computer Program Interface Protection*, s. 889.

<sup>202</sup> Spojené státy americké. Copyright Act, 17 U.S.C. § 102(b) (2006). Překlad vlastní.

<sup>203</sup> Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, *SAS Institute Inc. v. World Programming Ltd.*

počítačového programu nejsou chráněny autorským právem.<sup>204</sup> Funkcionality rozvedl ve svém stanovisku rovněž generální advokát, dle kterého lze použít specifikaci rozhraní jiného autora, aniž by tím bylo porušeno autorské právo. To však pouze tehdy, kdy nepůjde o prosté kopírování zdrojového kódu.<sup>205</sup> Soudní dvůr podle Weston podal účelovou interpretaci směrnice o právní ochraně počítačového programu, podle které funkčnost rozhraní nesmí omezovat interoperabilitu.<sup>206</sup> Soud uvedl, že výběr a organizace programu a implementace jeho funkcí mohou spadat pod ochranu autorského práva.<sup>207</sup> I pokud je dílo shledáno dílem autorským, stále je možné, že některé prvky v díle chráněném autorskými právy mohou ostatní použít. Prvky, které představují myšlenku díla, patří do veřejné sféry. Rozsah ochrany se tak zužuje tím, že se zabrání monopolním právům na myšlenky. To umožňuje konkurenci v odvětví.<sup>208</sup>

Z dichotomie myšlenka/vyjádření plyne, že ochrana je poskytována pouze vyjádření. Myšlenky mohou ostatní používat a rozvíjet bez toho, aby tím porušovali autorské právo. Tímto dělením se autorská práva snaží najít rovnováhu mimo jiné v ekonomických zájmech autora a společnosti. Ačkoli ochrana autorských práv nezahrnuje dílo celé, nýbrž pouze vyjádření díla, rozdíl plynoucí z nemožnosti myšlenku z počítačového programu bez dalšího získat, stále brání spravedlivému vyvážení zmíněných zájmů.<sup>209</sup>

K tomu, do jaké míry lze na část díla pohlížet jako na myšlenku se vyjádřil i Nejvyšší soud České republiky. Ten k § 2 odst. 6 autorského zákona uvedl, že dle něj námět jako takový není autorským dílem, avšak

---

<sup>204</sup> *Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, SAS Institute Inc. v. World Programming Ltd*, výrok 1).

<sup>205</sup> *Opinion of advocate general Bot*. Delivered on 29 November 2011, Case C-406/10, SAS Institute Inc. v World Programming Ltd. [online]. [cit. 25. 2. 2021]. <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:62010CC0406:EN:HTML>, bod. 55–57.

<sup>206</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 81.

<sup>207</sup> Mylly, U.-M. *Intellectual property protection of computer program interfaces and interoperability*. IPR University Center Helsinki, 2013, s. 34.

<sup>208</sup> Mylly. *Harmonizing Copyright rules for Computer Program Interface Protection*, s. 890.

<sup>209</sup> Tamtéž, s. 902.



pouze jedná-li se o prostý, respektive holý námět.<sup>210</sup> V takovém případě jde pouze o myšlenku nebo nápad a ochrana autorským právem není možná. V případě námětu rozvitého, ze kterého je zřetelná i individuální inovativní tvorba autora se o autorské dílo jednat může.<sup>211</sup> Nelze přistoupit na to, aby u prvků počítačového programu nebo u programu samotného, který je svým způsobem obecný, byla omezena jejich přístupnost z důvodu ochrany autorským právem. Již z podstaty nemohou být obecná díla chráněna, neboť jejich tvorba není přímo odvislá od lidského vědomí, jakož spíše od přírodních jevů, skutečností nebo historie a není tak splněna jejich individualita. Autorské právo tedy v podstatě zakazuje prosté kopírování či jiné zpracování vyjádření. Naopak užití myšlenek, na základě kterých chráněné dílo pracuje, je možné.

Koukal má za to, že by ve střeoevropských právních řádech měla být dichotomie myšlenka/vyjádření rozlišována pomocí metody abstrakce konkretizace, tedy rozlišováním individualizovaných a neindividualizovaných částí díla.<sup>212</sup> Stejně tak již Ulmer nesouhlasil s rozlišováním formy a obsahu s tvrzením, že takové rozlišení není vždy možné. Rozlišoval mezi individuálně chráněnými prvky díla a mezi prvky, které jsou určeny k obecnému užití.<sup>213</sup> To v podstatě znamená, že i obsah vytvořený autorem dostatečně individuálně a tvůrčím způsobem je autorským právem chráněn. Koukal se domnívá, že „*bychom měli také v českém právu u autorskoprávně chráněných částí díla rozlišovat mezi vnější formou díla, která je vnímatelná smysly, a individualizovanými prvky díla, které skrze smyslové vnímání působí na lidský rozum.*“<sup>214</sup> Při ochraně autorským právem by tak bylo rozlišováno mezi vyjádřením, respektive formou, kterou lidská bytost vnímá a mezi individualizovanými prvky díla. Rozlišujeme tedy individuální a neindividuální část díla. Obdobně funguje již zmíněné rozlišování fungující v angloamerickém právním řádu. Jedná se o tzv. *idea-expression dichotomy*, mimo jiné obsažené také v TRIPS<sup>215</sup>. Koukal konstatuje, že mezi oběma pojetími není podstatný

<sup>210</sup> Rozhodnutí Nejvyššího soudu ČR. ze dne 16. 5. 2007, sp. zn. 5 Tdo 449/2007.

<sup>211</sup> Telec. Tůma. *Autorský zákon, Komentář. 2. vydání*, 7-71, bod 94.

<sup>212</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 218.

<sup>213</sup> Ulmer, E. *Urheber und Verlagsrecht. 3. vyd.* Berlin, Heidelberg (DE), New York (USA): Springer, 1980, str. 120. In: Koukal. *Autorské právo, public domain a lidská práva*, s. 205.

<sup>214</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 208.

<sup>215</sup> Čl. 9 odst. 2 TRIPS.

rozdíl.<sup>216</sup> Nicméně zdůrazňuje, že tradiční americké instituty je potřeba vnímat v kontextu středoevropských právních tradic.<sup>217</sup>

#### 4.2.2 Způsoby odlišení chráněných a nechráněných prvků díla

Z diskusí ohledně ochrany myšlenek a principů došlo ve Spojených státech amerických rovněž k vytvoření tří přístupů, které zkoumají dichotomii myšlenka/vyjádření, tj. *abstractions test*, *merger doctrine*, *scènes à faire*, na základě, kterých lze odlišit volnou a chráněnou složku počítačového programu. Každou z metod je možné určit, respektive odlišit použití zdrojového kódu jednoho programu jako inspirace a jeho obyčejné kopírování. Přestože se jedná o doktríny používané obvykle ve Spojených státech, jsou obecně použitelné i v našem právním prostředí. Podle Koukala tyto doktríny mají přímý protějšek ve středoevropské autorskoprávní doktríně.<sup>218</sup> *Merger doctrine* posuzuje existence tvůrčí volnosti, *abstractions test* odděluje neindividuální prvky díla a *scènes à faire* obvyklé prvky, které jsou pro konkrétní druh díla typické.<sup>219</sup> Podle Koukala je proto důležité, interpretovat tyto doktríny v souladu se středoevropským pojetím individuality autorského díla a vnitřní struktury obecného fondu.<sup>220</sup>

První přístup je známý jako tzv. *abstractions test*.<sup>221</sup> Ten slouží jako návod soudům, při zkoumání, zda došlo k obyčejnému kopírování počítačového programu nebo byl program pouhou inspirací. Odděluje prvky, které v kontinentální doktríně spadají do oblasti vnitřní formy díla, jedná se například o námět díla, dějová linie nebo charakteristiky postav.<sup>222</sup> Neexistuje totiž dílo, které by bylo v celku původní, lze jen s těžší předpokládat, že by se autor při vytváření díla obešel bez jakékoli vnější

---

<sup>216</sup> Berking, Ch. *Die Unterscheidung von Inhalt und Form im Urheberrecht*. Baden-Baden: Nomos, 2002, s. 69 a násl.; In: Koukal. *Autorské právo, public domain a lidská práva*, s. 210.

<sup>217</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 210.

<sup>218</sup> Tamtéž, s. 218.

<sup>219</sup> Tamtéž, s. 231.

<sup>220</sup> Tamtéž, s. 231.

<sup>221</sup> Tamtéž, s. 219.

<sup>222</sup> Tamtéž, s. 220.

inspirace.<sup>223</sup> Nemusí však jít nezbytně o inspiraci ostatními vývojáři, může se jednat o inspiraci fakty a skutečnostmi existujícími nezávisle na autorovi. I pro tuto doktrínu je důležité rozdělení námětu na holý a rozvitý, které přineslo například rozhodnutí Nejvyššího soudu ze dne 16. 5. 2007, sp. zn. 5 Tdo 449/2007 viz kapitola 4.2.1. Dalším rozhodnutím je *Nichols v. Universal Pictures Corp.*, ve kterém soud dospěl k názoru, že použití základních prvků a motivů díla není v rozporu s autorským právem, nejedná-li se o podstatnou část díla.<sup>224</sup> Tedy náměty děl, které se podobají co do základního příběhu nejsou chráněny, neboť nejsou dostatečně individuální. Stejně tak nejsou chráněny postavy a jejich charakteristika jedná-li se o charakteristiku základní bez konkrétních detailů.<sup>225</sup> Test abstrakce do značné míry přispěl k vytvoření testu známého jako test abstrakce-filtrace-komparace. Nejprve je nutné oddělení obecně známých anebo neoriginálních prvků od ostatních prvků. Až zbylé části jsou porovnávány a je v nich hledána podobnost.<sup>226</sup>

Druhý přístup je znám jako tzv. *merger doctrine*. Ten má za cíl vyloučení technicky a funkčně předurčených prvků.<sup>227</sup> Existují dva pohledy na myšlenku *merger doctrine*. Americký přístup, který staví do pozice mimo ochranu autorského práva prvky programu, které by v případě jejich ochrany mohly překážet v tvorbě jiným autorům. A kontinentální, který vylučuje typicky prvky, u nichž není možné, aby byly součástí tvůrčí činnosti.<sup>228</sup> Pouze v případech, ve kterých měl autor dostatek volnosti tvořit totiž lze poskytovat autorskoprávní ochranu. Tato volnost se však nenachází v dílech technicky či funkčně předurčených, neboť autor nemůže provést volbu.<sup>229</sup> Podle Koukala se ve své podstatě jedná o poměrování kolidujících zájmů.<sup>230</sup> Merger doctrine v podstatě vede k tomu, že je-li možné nějaký obsah nebo funkcionalitu vyjádřit dostatečně individuálně, je možné, aby program byl chráněn autorským právem. Je-li obsah nebo funkcionalita technicky, přírodně nebo například funkčně natolik předurčena a autor se jí musí řídit, nelze na takové prvky pohlížet

---

<sup>223</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 219.

<sup>224</sup> Tamtéž, s. 219-220.

<sup>225</sup> *Nichols v. Universal Pictures Corp.*, 45 F. 2d 119 (2d Cir. 1930).

<sup>226</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 221.

<sup>227</sup> Tamtéž.

<sup>228</sup> Tamtéž, s. 221.

<sup>229</sup> Tamtéž, s. 221, 222.

<sup>230</sup> Tamtéž, s. 224.

jako na autorská díla. Autor musí použít jedno konkrétní vyjádření. V takovém případě se myšlenka spojí s vyjádřením a takové vyjádření nelze chránit autorským právem.<sup>231</sup>

Tato doktrína je z pohledu rozhraní zajímavá. Pracuje totiž s tím, že jak vyjádření, tak myšlenka mohou být spojeny v jedno, jde-li o přímý vliv myšlenky na podobu vyjádření, bez kterého by myšlenka ztrácela smysl. Hlavním hodnotícím kritériem není originalita ani individualita, nýbrž to, zda by nevznikla postavením takového prvku do pozice chráněného autorského díla neopodstatněná újma ostatním autorům. Lze tedy dílo zkoumat z toho pohledu, kterým se snažíme najít jiné možné vyjádření pro jednu konkrétní myšlenku. Pokud danou myšlenku, na které stojí rozhraní nepůjde ztvárnit jinak než v blízké podobnosti s původním rozhráním, nemůže rozhraní podléhat autorskoprávní ochraně, neboť by se taková myšlenka stala vlastnictvím jediné osoby. Pokud ji půjde ztvárnit více způsoby, nicméně pouze s nepodstatnými rozdíly, pak se rovněž nemůže jednat o autorské dílo.<sup>232</sup> Je třeba podotknout, že specifikace rozhraní může mít jen limitované množství vyjádření. Pokud není možné vyjádřit myšlenku, respektive dosáhnout funkcionality pomocí alternativního vyjádření, pak se myšlenka a vyjádření spojí v jedno a tato myšlenka pak nemusí být autorským právem chráněna.<sup>233</sup> Značné množství počítačových programů, které operují na nejnižší úrovni a jejichž rozhraní musí obsahovat předem dané specifické prvky budou mít více či méně podobný zdrojový kód. Originalita specifikace rozhraní bude v tomto smyslu na hraně.<sup>234</sup> V opačném případě by mohla nastat situace, kdy se autor, který konkrétní rozhraní vyvinul jako první dostane do monopolního postavení. Takový autor bude mít k myšlence prakticky monopol, protože danou funkci/myšlenku rozhraní nebude možné napodobit odlišným způsobem vyjádření, tedy zápisem zdrojového kódu.

Už CONTU dříve určilo, že nejen myšlenky, ale i prvky obsahující myšlenky, které nelze vyjádřit vícero způsoby a jsou tedy funkčně

---

<sup>231</sup> Mylly. *Harmonizing Copiright rules for Computer Program Interface Protection*, s. 906.

<sup>232</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 223-224.

<sup>233</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 40.

<sup>234</sup> *Navitaire Inc v Easyjet Airline Co. & Anor*, sp. zn. 2005 E.C.D.R. 16 (UK High Court of Justice, 2004)

předurčené, nepodléhají autorské ochraně.<sup>235</sup> To znamená, že zdrojový kód, který nelze napsat jinak, aniž by se změnila jeho funkce, nelze chránit autorským právem. V tomto směru zabránil soud monopolu na myšlenky obsažené v jinak chráněném vyjádření například ve věci *Levola Hengelo BV v. Smilde Foods BV*. Podle SDEU nelze mít za to, že dílo, jehož zhotovení je dáno technickými úvahami, určitými pravidly nebo jinými omezeními v jejichž důsledku nezůstal žádný prostor pro tvůrčí svobodu, vykazuje nezbytnou originalitu.<sup>236</sup> Na druhou stranu může být originalita shledána u díla, které sice bylo částečně předurčeno, ale autor měl nadále možnost, aby v díle odrážel svou osobnost a tvůrčí svobodu.<sup>237</sup> Obdobně rozhodl soud ve věci *Computer Associates v. Altai*.<sup>238</sup> Soud usoudil, že existují důvody, proč může existovat pouze omezený počet účinných implementací pro jednu funkcionalitu. Lze tedy dojít k závěru, že pokud nedojde k přímému kopírování, ale i tak jsou implementace totožné, nedojde k porušení autorského práva. Soud dospěl k závěru, že prvky, jež jsou předurčeny funkcionalitou, nelze zahrnout do prvků srovnávaných v rámci podobností, neboť se jedná o nechráněné prvky.<sup>239</sup> Počítačová rozhraní, která jsou zásadní pro interoperabilitu jsou prvky, ve kterých se efektivně spojuje funkce či myšlenka a vyjádření.

Specifikace rozhraní určují, jak musí počítačové programy fungovat, aby byla zajištěna interoperabilita s původním programem. Specifikace rozhraní tedy významně omezuje možnosti vývoje dalších programů, respektive jejich rozhraní.<sup>240</sup> Funkcionality a vyjádření jsou u specifikace rozhraní spojeny. Proto by tato rozhraní neměla být chráněna autorskými právy. Rovněž proto, že rozhraní musí být navrženo tak, aby dostatečně identifikovalo své funkcionality. Rozdíl mezi specifikací rozhraní a jeho implementací je tedy viditelný. Autorské právo může chránit

---

<sup>235</sup> National Commission on New Technological Uses of Copyrighted Works (CONTU). Final Report on the National Commission on New Technological Uses of Copyrighted Works, 3 Computer L.J. 53 (1981) str. 20.

<sup>236</sup> Rozsudek soudního dvora ze dne 13. listopadu 2018 ve věci C-310/17, bod 24.

<sup>237</sup> Tamtéž, bod 26.

<sup>238</sup> *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992).

<sup>239</sup> Mylly. *Harmonizing Copyright rules for Computer Program Interface Protection*, s. 905.

<sup>240</sup> Samuelson, P. *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*. *SSRN Electronic Journal*. 2015 [cit. 12. 3. 2021]. <http://www.ssrn.com/abstract=2667740>, s. 1266.

kód, který implementuje rozhraní, splňuje-li požadavek originality. Nechrání však specifikaci rozhraní, která je předurčena v rámci zachování svých funkcí, zejména interoperability.<sup>241</sup>

Doktrína *scènes à faire* staví do pozice nechráněných prvků počítačového programu takové prvky, které pro něj jsou charakteristické, jsou obecně známé a lze je u něj očekávat nebo je dokonce musí obsahovat.<sup>242</sup> Tyto běžně používané prvky není třeba porovnávat, protože nejsou dostatečně individuální a zároveň jsou pro konkrétní programy typické. Je třeba porovnat prvky na nich postavené, kterými autor dostatečně individuálně přispěl. V případě, že se budou krýt i tyto individuální prvky můžeme hovořit o porušení autorského práva.<sup>243</sup> Podle Koukala nepřináší *scènes à faire* nic, co by nebylo vlastní rozsahu autorskoprávní ochrany ve středoevropském prostoru.<sup>244</sup> Je chráněn námět individualizovaný, který je ztvárněním námětu obecného.<sup>245</sup> Díky tomu je možné odlišit příliš abstraktní prvky, které nenesou autorskoprávní individualitu.<sup>246</sup> I v tomto případě lze zmínit soudní spor ve věci *Computer Associates v. Altai*.<sup>247</sup> Soud pomocí této doktríny odfiltroval nechráněné prvky, které jsou podmíněny vnějšími faktory, a které omezují možnosti vývoje.<sup>248</sup> Podle soudu se jedná o následující prvky: 1) mechanické specifikace počítače, na kterém má být určitý program spuštěn; 2) požadavky na kompatibilitu jiných programů, s nimiž je program navržen, aby pracoval společně; 3) konstrukční normy výrobců počítačů; 4) poptávky obsluhovaného průmyslu; a 5) široce přijímané programovací postupy v počítačovém průmyslu.<sup>249</sup> Se zohledněním těchto požadavků tak soud rozhodl, že specifikace rozhraní nemůže být chráněna autorským právem.

---

<sup>241</sup> Samuelson, P. *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, s. 1266.

<sup>242</sup> Koukal. *Autorské právo, public domain a lidská práva*, s. 225-226.

<sup>243</sup> Tamtéž, s. 228.

<sup>244</sup> Tamtéž, s. 227.

<sup>245</sup> Tamtéž, s. 227.

<sup>246</sup> Tamtéž, s. 228.

<sup>247</sup> *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992).

<sup>248</sup> Mylly. *Harmonizing Copyright rules for Computer Program Interface Protection*, s. 905.

<sup>249</sup> *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), 709-710.

### 4.3 Specifikace a implementace rozhraní

Z veškerých poznatků uvedených výše nám plyne, že rozhraní a jeho nezpřístupnění ostatním, má přímý vliv na interoperabilitu. Je proto důležité, aby právní řád našel rovnováhu mezi ochranou rozhraní a jeho zpřístupněním ostatním soutěžitelům.<sup>250</sup> Protože je počítačový program chráněn jako literární dílo, je chráněno jeho vyjádření, respektive jeho kód. Naopak myšlenky, nápady, funkcionality a další abstraktní části, na kterých počítačový program stojí, chráněny autorským právem nejsou. Nemí žádoucí bránit ostatním s danou myšlenkou pracovat a na jejím základě vyvíjet vlastní produkty. Naopak je žádoucí s nimi volně disponovat.<sup>251</sup> *Implementace* rozhraní, tedy konkrétní zápis algoritmů ve zdrojovém kódu lze podřadit v tomto smyslu pod vyjádření, na jehož základě počítačový program komunikuje. Vyjádření takové implementace rozhraní, lze relativně nahradit jiným vyjádřením vykonávajícím tu samou funkci.<sup>252</sup> Pro jedno rozhraní je tedy teoreticky možné použít vícero vyjádření, respektive zápisů ve zdrojovém kódu. I tak však jejich počet není neomezený. Čím stručnější rozhraní je, tím menší je počet alternativních vyjádření, která by jej mohla nahradit. Po vyčerpání všech možností by přes jejich ochranu nebylo možné nové rozhraní vytvořit. Zároveň není vždy vhodné, dokonce ani možné, nějakou funkci či myšlenku vyjádřit alternativně.

Jak jsme si již uvedli, pro zajištění interoperability musí mít oba programy identické *specifikace* rozhraní.<sup>253</sup> Specifikace rozhraní popisuje začátek zdrojového kódu, názvy, vstupy a výstupy a jiné informace nezbytné pro správné fungování rozhraní. Jedná se o technickou informaci, která stanoví požadavky na komunikaci dvou počítačových programů.<sup>254</sup> Požadavky na specifikaci rozhraní jdou proti podstatě autorského práva – originalitě. Pro ochranu počítačového programu je totiž originalita díla

---

<sup>250</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 80.

<sup>251</sup> Štědroň. *Ochrana a licencování počítačového programu*, s. 220.

<sup>252</sup> Mylly. *An Evolutionary Economics Perspective on Computer Program Interoperability and Copyright*, s. 302

<sup>253</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 61.

<sup>254</sup> Mylly. *An Evolutionary Economics Perspective on Computer Program Interoperability and Copyright*, s. 302.

žádoucí, a to včetně originality v rámci rozhraní. Originalita v rozhraní však není v rámci zajištění interoperability možná, protože dvě originální rozhraní by spolu nemohla komunikovat. Specifikace rozhraní by neměla být autorským právem chráněna, protože není jiná možnost, jak zajistit interoperabilitu s jiným počítačovým programem než pomocí stejné specifikace. U specifikace rozhraní se projeví mimo jiné *scènes à faire* nebo *merger doctrine*, neboť jeho funkcionalita je technicky a funkčně natolik předurčena, že rozhraní nelze dostatečně individualizovat. Autor musí použít jedno konkrétní vyjádření, které vede k žádané funkcionalitě.

Oproti tomu by měla být implementace, kterou lze provést různými způsoby, chráněna. Tomu tak není, pokud existuje jeden nebo malé množství možných způsobů, jak implementovat specifikaci rozhraní. Pak nelze chránit autorským právem ani implementaci.<sup>255</sup>

Podle dichotomie myšlenka/vyjádření tak nespádají pod ochranu autorského práva procesy, myšlenky a metody počítačového programu, respektive rozhraní. Zároveň nelze poskytovat ochranu tehdy, nelze-li dosáhnout stejné funkce počítačového programu, respektive rozhraní, pomocí jiného vyjádření. Vzhledem k funkci specifikace rozhraní však nelze dosáhnout dané funkce jinak než kopií textu. Je tedy nepodstatné, že existuje vícero způsobů, jak lze sestavit rozhraní, pokud takové rozhraní nebude plnit žádané funkce.<sup>256</sup>

Zajímavými jsou v tomto ohledu níže uvedené soudní spory. Prvním z nich je spor mezi *Whelan Associates, Inc. a Jaslow Dental Laboratory*. Společnost Jaslow si od společnosti Whelan objednala software, který zajišťoval základní administrativu a chod dentální laboratoře, kterou provozovala. Počítačový program však Jaslow nevyužila pouze na smluvnou činnost, ale některé jeho části použil za účelem vývoje jiného počítačového programu.<sup>257</sup> V tomto případě soud postupoval podle testu, který z ochrany vylučoval myšlenky a funkcionality programu. Myšlenku zde soud spatřil v zajištění administrativního chodu dentální laboratoře.<sup>258</sup>

---

<sup>255</sup> Mylly. *An Evolutionary Economics Perspective on Computer Program Interoperability and Copyright*, s. 302.

<sup>256</sup> Menell. *Rise of the API Copyright Dead: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, s. 305–490.

<sup>257</sup> *Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc.* 797 F.2d 1222 (3d Cir. 1986), 1226-1227.

<sup>258</sup> Tamtéž, 1240.



To značně zužuje rozsah myšlenek, které jsou nechráněné. Zároveň soud naznal, že nelze chránit ty části programu, které nelze vytvořit jinými způsoby.<sup>259</sup> Program, jehož funkcionalitou je zajišťování chodu dentální laboratoře, lze zajisté vytvořit mnoha způsoby a není tak důvodné jakoukoli část programu kopírovat. Takováto judikatura by však zakazovala nabyvatelům díla uplatňovat výjimky se kterými mimo jiné počítá i zákon a vedla by k nemožnosti zajistit si interoperabilitu s dotčeným programem.

K podobnému názoru došel soud v případě *Plains Cotton Coop. Assoc. v. Goodpasture Comput. Serv., Inc.* Ten však dělil na vyjádření a myšlenku jednotlivé části díla a nespokojil se pouze z myšlenou základní.<sup>260</sup> Podobnosti v programech, které existují z důvodu určitých principů a skutečností, které program k danému účelu musí obsahovat jsou soudem považovány za myšlenky, které chráněny nejsou.<sup>261</sup> Soud došel k závěru, že struktura, sekvence a organizace programu není chráněna. Je tedy otázka, zda rozhraní není součástí struktury, sekvence nebo organizace programu.

Výsledky těchto soudních rozhodnutí byly vyvráceny rozsudkem ve věci *Computer Associate v. Altai*. Tento soudní spor vznikl díky zaměstnanci Altai, který však dříve pracoval pro Computer Associates. Vzhledem k tomu, že obě společnosti vyvíjely podobné programy, si zaměstnanec usnadnil práci tím, že okopíroval kód, na kterém zprvu pracoval právě pro Computer Associates.<sup>262</sup> Altai stáhl takto okopírovaný program z trhu a pomocí metody *clean-room* vytvořil jeho náhradu, se kterou opět začal obchodovat.<sup>263</sup> Soud rozhodl, že shodné části programů jsou určeny okolnostmi (např. rozhraní) a pro daný program typickými funkcionalitami a touto podobností nelze porušit autorské právo.<sup>264</sup> V tomto soudním sporu se odvolací soud pokusil o pravidla, podle

---

<sup>259</sup> *Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc.* 797 F.2d 1222 (3d Cir. 1986), 1248.

<sup>260</sup> Samuelson. *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, s. 1229.

<sup>261</sup> *Plains Cotton Co-op. v. Goodpasture Computer* 807 F.2d 1256 (5th Cir. 1987).

<sup>262</sup> *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), 700.

<sup>263</sup> Tamtéž.

<sup>264</sup> Tamtéž, 713-714.

kterých lze určit, zda bylo autorské právo porušeno.<sup>265</sup> Jedná se o tzv. *abstraction-filtration-comparison test*. I díky tomuto testu je možné psát interoperabilní programy, neboť z ochrany vylučuje vybrané prvky díla.

V této podkapitole je definována specifikace a implementace rozhraní, které jsou zároveň podřazeny pod myšlenku nebo vyjádření. Specifikace rozhraní nám určuje, jak musí být rozhraní implementováno, aby plnilo správně své funkce. Implementace je pak přímo zápis konkrétních algoritmů, jež má počítač vykonávat. Autor, který vytváří program, jež má být interoperabilní s jiným, musí použít totožnou specifikaci rozhraní, jinak interoperabilita nebude možná. Implementace rozhraní je v případě dodržení specifikace rozhraní možná obvykle vícero vyjádřeními. Není tomu však vždy. Například v případě jednoduchých rozhraní je vývojář značně limitován. V následující kapitole přiblížíme, proč je na specifikaci rozhraní třeba pohlížet spíše jako na nechráněnou myšlenku.

#### 4.4 Specifikace rozhraní jako myšlenka

Počítačové programy, stejně jako jejich rozhraní, se od ostatních autorských děl liší zejména svojí funkcionalitou. Je to právě funkcionalita, kvůli které si lidé počítačové programy pořizují. Kód, díky kterému program funguje, není v tomto případě nejdůležitější. Důležitá je funkcionalita, kterou kód umožňuje. Právě ta přináší uživatelům žádaný užitek. Počítačové programy jsou ze své podstaty funkční díla a patří do oblasti techniky.<sup>266</sup> Funkcionalita definuje úkoly a operace, které má stroj provádět, definuje jaké vstupy vedou k jakým výstupům.<sup>267</sup> Funkcionalitu počítačového programu, stejně jako myšlenky, nelze monopolizovat, neboť by tím byl omezen technologický pokrok.<sup>268</sup> Galajdová a Zibner upozorňují, že ačkoli je obecně přijímáno stanovisko, že jedné funkcionality

---

<sup>265</sup> *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), 707.

<sup>266</sup> Mylly. *Harmonizing Copiright rules for Computer Program Interface Protection*, s. 878-886.

<sup>267</sup> Lord J. Lewison, Court of Appeal in England(Civil Division): Copyright protection of software manuals - *SAS v. World Programming* (GRUR Int. 2014, 289), decision of 21 November 2013, [2013] EWCA Civ 1482, Marg 74.

<sup>268</sup> Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, *SAS Institute Inc. v. World Programming Ltd*, bod 40, 43; Srov. *Opinion of advocate general Bot*, bod 57.

lze dosáhnout několika způsoby, v rámci počítačových programů je vhodné na situaci pohlížet opačně. Dle jejich názoru je vhodné spíše smýšlení: „*mnoho myšlenek = jedno vyjádření.*“<sup>269</sup> A to, protože počítačové programy jsou definovány funkcionalitou, které mají dosahovat.<sup>270</sup> Rozhraní je pak jakousi nadstavbou již existujícího programu a pracuje s jeho existujícími funkcionalitami. Díky tomu není zcela možné oddělit myšlenky od jejich vyjádření, neboť rozhraní vykonává předem určené funkcionality a jeho vyjádření je tak to té míry omezené. Funkce jako takové tedy nemohou být předmětem ochrany. Množství způsobů, jakými lze dosáhnout stejné funkcionality je omezené, a proto se funkcionality podobají spíše myšlenkám.<sup>271</sup> Stejně chápe funkcionality i generální advokát Bot: „*Funkce nebo kombinace několika funkcí je srovnatelná s myšlenkou, a nemůže být tedy chráněna jako taková autorským zákonem.*“<sup>272</sup> Nejvyšší soud Spojených států amerických rozhodl ve věci *Baker v. Selden*<sup>273</sup> tak, že díla se silnými funkčními prvky, jako jsou účetní knihy, nemají nárok na stejný stupeň ochrany jako faktická díla, respektive nemusí mít nárok na ochranu vůbec.<sup>274</sup> K tomuto názoru se přiklání rovněž Samuelson, podle které prvky, jež jsou nezbytné pro správné fungování například účetního systému, nejsou chráněny.<sup>275</sup>

Specifikace rozhraní určuje, co rozhraní umožňuje a jak je pomocí něj komunikováno. Sama o sobě neumožňuje počítačovému programu vykonávat jeho funkcionality, pouze je definuje. Specifikace rozhraní tedy nelze podřadit pod vyjádření počítačového programu, a proto specifikace rozhraní nejsou způsobilé pro ochranu autorských práv.

---

<sup>269</sup> Galajdová, D. Zibner, J. Nedostatky autorskoprávní ochrany počítačového programu (Deficiencies of computer program copyright protection). *Právní rozhledy*. C.H. Beck, 2018, vol. 26, No 22, s. 784.

<sup>270</sup> Tamtéž.

<sup>271</sup> Vezzoso. *Copyright, Interfaces, and a Possible Atlantic Divide*, s. 155.

<sup>272</sup> *Opinion of advocate general Bot*, parag. 63.

<sup>273</sup> SCOTUS. *Baker v. Selden*, 101 U.S. 99 (1879), 107.

<sup>274</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 69.

<sup>275</sup> Samuelson, P. *Why Copyright Law Excludes Systems and Processes from the Scope of Its Protection Symposium: Frontiers of Intellectual Property: III. Copyright*. Texas Law Review. 2006, č. 7, s. 1974.

Rovněž legislativní historie směrnice o právní ochraně počítačových programů<sup>276</sup> dopomáhá k pochopení toho, co zákonodárce zamýšlel chránit.<sup>277</sup> Chráněno mělo být pouze individuální vyjádření díla tak, aby bylo umožněno ostatním autorům vytvářet podobné nebo dokonce identické programy za předpokladu, že se zdrží kopírování. Směrnice tak zajisté neukládá žádná omezení na opakované použití funkčnosti programu nebo myšlenek a principů, které nabyvatel získá oprávněnou cestou.<sup>278</sup> Autorské právo tak chrání zejména vyjádření počítačového programu před obyčejným kopírováním. Funkcionality však nejsou autorským zákonem chráněny. Jejich ochrana by byla možná spíše jako vynález.<sup>279</sup> Prvky, které mají svoji podobu danou z důvodů funkčnosti, nelze chránit. Jedná se i o případy, kdy se myšlenka a vyjádření spojily (viz merger doctrine).<sup>280</sup> Stejným způsobem se k věci postavily i americké soudy, když zamítly stížnosti na porušení autorských práv založené na kopírování funkcionalit.<sup>281</sup> Zde je dle Van Rooijena problém v tom, že i díla, která nejsou funkční, mohou i tak potřebovat použít stejné vyjádření, protože jinak nebudou sloužit k účelu, ke kterému mají.<sup>282</sup> Příkladem toho může být například soudní spor *Dior v. Evora*.<sup>283</sup>

---

<sup>276</sup> Směrnice je dále rozebrána v podkapitole 4.6.

<sup>277</sup> Commission of the European Communities, *Proposal for a Council Directive on the legal protection of computer programs*, 1989, COM(88) 816 final - SYN 183, s. 11.

<sup>278</sup> Samuelson, P., Vinje, T. C., Cornish, W. R. *Does Copyright Protection Under the EU Software Directive Extend to Computer Program Behaviour, Languages and Interfaces?* Rochester, NY: Social Science Research Network, 2011 [cit. 25. 2. 2021]. <https://papers.ssrn.com/abstract=1974890>, s. 6.

<sup>279</sup> Huet, J., Ginsburg, J. C. *Computer Programs in Europe: A Comparative Analysis of the 1991 EC Software Directive*. Columbia journal of transnational law, 1992, s. 336.

<sup>280</sup> Rozsudek Soudního dvora ze dne 22. prosince 2010, C-393/09, *Bezpečnostní softwarová asociace – Svaz softwarové ochrany v. Ministerstvo kultury*, bod 48.

<sup>281</sup> Např. *Computer Associates Int'l, Inc. v. Altai, Inc.; Lotus Development Corp. v. Borland Int'l* In: Samuelson. Vinje. Cornish. *Does Copyright Protection Under the EU Software Directive Extend to Computer Program Behaviour, Languages and Interfaces?* s. 7.

<sup>282</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 61

<sup>283</sup> Judgment of the European Court of 4 November 1997, C-337/95, *Parfums Christian Dior SA .and Parfums Christian Dior BV v Evora BV*.

Karjala je pak přesvědčený, že neuživatelské rozhraní je celé funkčním prvkem. Nekomunikuje s lidskou bytostí ani ji nebaví a jejich ochrana autorským právem tak není možná.<sup>284</sup>

Na základě výše uvedeného lze dojít k závěru, že autorské právo chrání vyjádření počítačového programu, nikoli jeho funkcionalitu, respektive funkcionalitu rozhraní. Funkcionality totiž lze vnímat jako myšlenky, které jsou na základě dichotomie myšlenka/vyjádření nechráněny. Stejně tak nemohou být chráněny prvky, které mají svoji podobu danou z důvodů funkčnosti, a to i tehdy, kdy se myšlenka a vyjádření spojily. Z těchto důvodů nelze specifikaci rozhraní chránit autorským právem. Vliv na míru ochrany má i standardizace. Prvky, které se stanou standardem nelze dobře nahradit, a proto jsou z ochrany vyloučeny.

#### 4.5 Standardizace rozhraní

Standardizace může mít velice příznivý dopad na trh. Nebyla by však možná, pokud by rozhraní, které se stalo standardem, bylo chráněno jako autorské dílo. Zajímavé pro pochopení standardizace je rozhodnutí *Lotus vs. Borland*.<sup>285</sup> Borland okopíroval jedna ku jedné rozhraní menu Lotus a použil jej v rámci svého produktu. Soud rozhodl, že hierarchické uspořádání příkazů v nabídce menu, včetně okopírování textu nabídky, není porušením autorského práva. Nedošlo totiž ke kopírování kódu programu.<sup>286</sup> Soud dospěl k tomu, že rozhraní Lotusu se stalo standardem a není tak chráněno autorským právem.<sup>287</sup> Standardizace však není to samé, co vyjádření, které je nutné používat stejně, aby byla plněna jeho funkce.<sup>288</sup> V tomto případě byla shledána nutnost zajištění funkce pomocí stejného vyjádření a zároveň bylo toto vyjádření shledáno jako

---

<sup>284</sup> Karjala. *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller Symposium: Copyright Protection and Reverse Engineering of Software*, s. 990.

<sup>285</sup> SCOTUS. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 516 U.S. 233 (1996).

<sup>286</sup> Vasilescu-Palermo, D. *APIs and Copyright Protection: The Potential Impact on Software Compatibility in the Programming Industry*. 16 J. Marshall Rev. Intell. Prop. L. 153, 2016, s. 157.

<sup>287</sup> SCOTUS. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 516 U.S. 233 (1996).

<sup>288</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 61.

*standard*. Bez možnosti použít podobnou nabídku menu by nebylo možné takový program vytvořit a byla by omezena konkurenceschopnost ostatních soutěžitelů. Pokud by menu nebylo standardem, musel by Borland postavit své vlastní. Nadále by k tomu však mohl využít myšlenek a principů rozhraní Lotusu. Vyvinout rozhraní se stejnou funkcionalitou by však nebylo možné, pokud by dané funkce nemohlo být dosaženo jinak než stejným vyjádřením.<sup>289</sup>

Standardem se zabýval také soud ve věci *Apple v. Microsoft*. Společnost Apple tvrdila, že operační systém Windows společnosti Microsoft porušoval autorská práva společnosti Apple ke grafickému uživatelskému rozhraní pro jeho počítačový systém Macintosh. Soud myšlenku takovéto ochrany odmítl, neboť by umožnila společnosti Apple vstoupit do monopolního postavení ke grafickému rozhraní, které je však v podstatě standardem a není k němu adekvátní alternativa.<sup>290</sup>

#### 4.6 Směrnice o právní ochraně počítačových programů

Jak jsme již uvedli, počítačový program je chráněn jako literární dílo. Je tomu tak jak na základě Smlouvy o autorském právu Světové organizace duševního vlastnictví, tak na základě Dohody o obchodních aspektech práv k duševnímu vlastnictví Světové obchodní organizace. Ani jeden z těchto dokumentů se však nezabývá otázkou ochrany rozhraní. Ochrana rozhraní je tak z velké části na uvážení jednotlivých států.<sup>291</sup> Pro české prostředí je relevantní snaha Evropské unie, upravit právní režim rozhraní pomocí směrnice o právní ochraně počítačových programů.<sup>292</sup> Tato směrnice harmonizuje právní úpravu ochrany počítačových programů v rámci Evropské unie. Vyjímá z autorskoprávní ochrany ta rozhraní počítačových programů, která lze podřadit pod myšlenky. Jedná se o ta rozhraní, která určují metody výměny informací, kterým musí nové programy vyhovovat, jestliže chtějí dosáhnout interoperability. Není jí tedy vyloučeno rozhraní jako takové a není jí ani určeno, které rozhraní

---

<sup>289</sup> SCOTUS. *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 516 U.S. 233 (1996).

<sup>290</sup> *Apple Computer, Inc. v. Microsoft Corporation*, 35 F.3d 1435 (9th Cir. 1994).

<sup>291</sup> Mylly. *Harmonizing Copyright rules for Computer Program Interface Protection*, s. 878-879.

<sup>292</sup> Směrnice o právní ochraně počítačových.

spadá pod myšlenky a zásady.<sup>293</sup> Směrnice u rozhraní sice odlišuje myšlenky a vyjádření. Nicméně tím není vyřešena otázka ochrany specifikace rozhraní. Již od počátku debat o směrnici jsou viditelné dva proudy nabízejí odlišné pohledy na interoperability počítačových programů. Zastánce vyjmutí specifikace rozhraní je na evropské úrovni ECIS. Oproti tomu SAGE žádá výslovnou ochranu rozhraní.<sup>294</sup> V rámci těchto dvou proudů bylo nutné najít kompromis a zohlednit zájmy obou skupin.<sup>295</sup> Směrnice v preambuli pod recitálem 10 shrnuje důležitost interoperability:

*„Počítačový program je určen ke komunikaci a spolupráci s ostatními prvky počítačového systému a s uživateli. Za tímto účelem je nezbytné logické a případně i fyzické propojení a interakce umožňující náležité fungování veškerých prvků programového a technického vybavení s jiným programovým a technickým vybavením i s uživateli. Části programu, které toto vzájemné propojení a interakci mezi jednotlivými prvky programového a technického vybavení umožňují, jsou obecně nazývány „rozhraní“. Toto funkční propojení a interakce jsou obecně nazývány „interoperabilita“; tuto „interoperabilitu“ lze vymezit jako schopnost vzájemně si vyměňovat informace a vzájemně vyměněné informace užívat.“<sup>296</sup>*

V recitálu 11 pak pro vyloučení nejasností stanoví, že předmětem ochrany je pouze vyjádření počítačového programu. Naopak myšlenky a zásady, na kterých jsou založeny jednotlivé prvky programu nejsou směrnici chráněny. A to se týká i těch myšlenek a zásad, na kterých jsou založena rozhraní.<sup>297</sup> Směrnice výslovně zmiňuje rozhraní, respektive myšlenky a principy na kterých stojí, což podtrhuje jeho důležitost

<sup>293</sup> Samuelson. Vinje. Cornish. *Does Copyright Protection Under the EU Software Directive Extend to Computer Program Behaviour, Languages and Interfaces?* s. 10.

<sup>294</sup> Palmer, A. K., Vinje, T. C. *The EC Directive on the Legal Protection of Computer Software: New Law Governing Software Development. International Law.* roč. 2, s. 70-71.

<sup>295</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 72; Srov. Palmer, A. K., Vinje, T. C. *The EC Directive on the Legal Protection of Computer Software: New Law Governing Software Development. International Law*, s. 77.

<sup>296</sup> Recitál 10 Směrnice o právní ochraně počítačových programů.

<sup>297</sup> Recitál 11 Směrnice o právní ochraně počítačových programů.

a zároveň nejistotu ohledně toho, jaký právní režim jej následuje. Z textu směrnice podle Van Rooijena spíše plyne, že rozhraní by chráněno být nemělo, jinak by nebylo konkrétně stanoveno, že myšlenky a principy, na kterých rozhraní stojí jsou vyloučeny z ochrany.<sup>298</sup> Z ochrany autorského práva však výslovně není vyjmuta každá specifikace, ale musí být zkoumáno na základě dichotomie myšlenka/vyjádření. Van Rooijen vykládá směrnici tak, že zákonodárce obešel přímé vyloučení specifikace tím, že jej vyloučil nepřímo.<sup>299</sup> Proč by jinak výslovně uváděl rozhraní v souvislosti s dichotomií. Podle Van Rooijena tak je evidentní, že specifikace rozhraní je chápána zákonodárcem jako myšlenka a implementace jako vyjádření.<sup>300</sup> K tomu přichází pomocí výkladu původního návrhu textu směrnice.<sup>301</sup> Van Rooijen předpokládá, že slovní spojení *specifikace rozhraní*, které bylo původně zamýšleno do směrnice uvést, bylo vyjmuta z textu směrnice proto, aby se zabránilo úvahám o tom, zda tedy nemůže být specifikace rozhraní chráněna autorským právem jako vyjádření.<sup>302</sup> Původní návrh obsahoval i *merger doctrine*<sup>303</sup>. Pokud myšlenka týkající se rozhraní jde vyjádřit jen několika málo způsoby, pak se myšlenka a vyjádření spojí.<sup>304</sup> Při práci na směrnici byly ve hře rovněž návrhy na výslovné vyloučení rozhraní z oblasti ochrany.<sup>305</sup> Přijaté znění nakonec nevyklučuje rozhraní obecně, ani mu však nepřiznává ochranu. Odkazuje spíše na obecnou dichotomii myšlenka/vyjádření.<sup>306</sup> Úprava se tedy značně podobá vyloučení specifikace rozhraní ve Spojených státech

---

<sup>298</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 73.

<sup>299</sup> Tamtéž, s. 72.

<sup>300</sup> Tamtéž, s. 72-73.

<sup>301</sup> European Commission, Proposal for Council Directive on the legal protection of computer programs, 17 March 1989.

<sup>302</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 73.

<sup>303</sup> Řešeno v kapitole 4.2.2.

<sup>304</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 74.

<sup>305</sup> Palmer, A. K., Vinje, T. C. *The EC Directive on the Legal Protection of Computer Software: New Law Governing Software Development. International Law*, s. 75.

<sup>306</sup> Graef, I. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* Journal of European Competition Law & Practice. 2014, č. 1, s. 13.



amerických.<sup>307</sup> V podkapitole 4.4 jsme došli k závěru, že specifikace rozhraní definuje funkcionality a ty lze podřadit pod myšlenku. Myšlenka jako taková není chráněna pomocí autorského práva.

K tomuto závěru lze dojít i pomocí rozboru přijatého textu směrnice. Ihned v čl. 1 odst. 2 je vyzdvižnuta nemožnost chránit funkcionalitu počítačových programů. Navíc je výslovně zmíněno rozhraní: „*Myšlenky a zásady, na kterých je založen kterýkoliv z prvků počítačového programu včetně myšlenek a zásad, na kterých je založeno jeho rozhraní, nejsou chráněny autorským právem podle této směrnice.*“<sup>308</sup> Tím je zdůrazněno, že mnoho prvků rozhraní bude spadat spíše pod nechráněné myšlenky než pod chráněné vyjádření. Podle Samuelson je vodítkem rovněž článek 6, na základě kterého je nabyvatelům díla umožněno zpětné inženýrství, pokud je nutné k zajištění interoperability.<sup>309</sup> Směrnice však nedává přesný návod vedoucí k tomu, jaké rozhraní, respektive jeho prvek je z ochrany vyjmut. Soudy tak jsou nuceny v každém případě poctivě zkoumat, zda lze rozhraní podřadit pod vyjádření, a tedy zda jej lze chránit či nikoli.

Judikatura Soudního dvora chápe specifikaci rozhraní jako relativně nechráněný prvek počítačového programu. V rozsudku SAS v. WPL<sup>310</sup> soud shledal jako chráněné vyjádření počítačového programu právě zdrojový kód, strojový kód, podrobnou strukturu kódu a přípravné koncepční materiály.<sup>311</sup> Soud uznal, že vývoj funkcionalit programu vyžaduje tvůrčí úsilí. Myšlenky, procesy, metody operací a matematické vzorce počítačového programu si však ochranu autorským právem nezaslouží ani jsou-li k jejich vzniku vynaloženy značné schopnosti, úsudek a práce.<sup>312</sup> Společnost World programming vyvinula systém, který napodoboval systém vyvinutý společností SAS. Společnost SAS dříve neumožňovala svým zákazníkům žádným způsobem provozovat jejich aplikace

---

<sup>307</sup> Mylly. *An Evolutionary Economics perspective on Computer Program Interoperability and Copyright*, s. 301.

<sup>308</sup> Čl. 1 odst. 2 Směrnice o právní ochraně počítačových.

<sup>309</sup> Samuelson. Vinje. Cornish. *Does Copyright Protection Under the EU Software Directive Extend to Computer Program Behaviour, Languages and Interfaces?* s. 13.

<sup>310</sup> Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, *SAS Institute Inc. v. World Programming Ltd.*

<sup>311</sup> Tamtéž, bod 37.

<sup>312</sup> Samuelson. Vinje. Cornish. *Does Copyright Protection Under the EU Software Directive Extend to Computer Program Behaviour, Languages and Interfaces?* s. 4.

existující v jazyce SAS bez možnosti nepořídít si uživatelskou licenci k modulům SAS.<sup>313</sup> Společnost World Programming Limited tedy měla zájem na vytvoření alternativního počítačového programu, který by byl tyto aplikace schopný provozovat. Za tímto účelem World programming použila formát datových souborů a programovací jazyk společnosti SAS.<sup>314</sup> Nejednalo se však o okopírování kódu, respektive kopírování nebylo prokázáno.<sup>315</sup> Soudní dvůr Evropské unie se přiklonil k názoru, že myšlenky a principy, na kterých je rozhraní postaveno nejsou vyjádřením a nejsou chráněny autorskými právy:

*„Ani funkce počítačového programu, ani programovací jazyk či formát datových souborů užívaných počítačovým programem za účelem využití některých z jeho funkcí nepředstavují formu vyjádření tohoto programu, a v důsledku toho nepožívají autorskoprávní ochrany počítačových programů ve smyslu této směrnice.“<sup>316</sup>*

Autorské právo chrání pouze formy vyjádření programu, tedy zdrojový a strojový kód. Pokud není kopírován přímo zdrojový kód týkající se rozhraní, ale bude znovu napsán pomocí analýzy<sup>317</sup> původního programu, nedojde k porušení duševního vlastnictví.<sup>318</sup> Funkce počítačového programu lze definovat jako veškeré možnosti, které systém pro zpracování informací nabízí, tedy jednotlivé úkony tohoto programu. Jinými slovy jsou funkcemi počítačového programu služby, které od něj uživatel očekává. Podle soudu nemohou být funkce počítačového programu jako takové předmětem autorskoprávní ochrany. Podle Weston poskytl Soudní dvůr účelovou interpretaci směrnice o software, díky které funkčnost rozhraní nesmí omezovat interoperabilitu.<sup>319</sup>

---

<sup>313</sup> High Court of Justice (England & Wales), SAS Institute Inc v World Programming Ltd [2010] EWHC 1829 (Ch) (23 July 2010), 6.

<sup>314</sup> Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, SAS Institute Inc. v. World Programming Ltd, bod 24.

<sup>315</sup> Tamtéž, bod 26.

<sup>316</sup> Tamtéž, výrok rozhodnutí.

<sup>317</sup> Black-box testing.

<sup>318</sup> Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, SAS Institute Inc. v. World Programming Ltd.

<sup>319</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 81.

Český právní řád zabývající se ochranou počítačového programu, respektive jeho rozhraní včetně možnosti zpětné inženýrství je výsledkem implementace Směrnice Evropského parlamentu a Rady 2001/29/ES ze dne 22. května 2001 o harmonizaci určitých aspektů autorského práva a Směrnice Evropského parlamentu a Rady 2009/24/ES ze dne 23. dubna 2009 o právní ochraně počítačových programů. Ochrana počítačového programu je v našem právním řádě uvedena zejména v § 65 a § 66 AZ.<sup>320</sup> Ochrana počítačovému programu je autorským právem přiznávána v souladu se světovým standardem tak, jako by se jednalo o dílo literární. Je tedy nutné, aby počítačový program měl své vnější vyjádření. To může být například ve strojovém nebo zdrojovém kódu.<sup>321</sup> Podle Telce rozhraní není vyjádřením počítačového programu: „Rozhraní počítačového programu neumožňuje rozmnožení daného počítačového programu, ale představuje pouze prvek tohoto programu, jehož prostřednictvím uživatelé využívají funkce uvedeného programu.“<sup>322</sup> S tímto tvrzením lze souhlasit v případě specifikace rozhraní. Implementace rozhraní jakožto zápis ve zdrojovém nebo strojovém kódu rozmnožení umožňuje. Telec však pravděpodobně pod pojem rozhraní ve svém tvrzení řadí pouze rozhraní uživatelské, které pro interoperabilitu není podstatné. I AZ z ochrany vyjímá myšlenky, které jsou podkladem rozhraní. Myšlenky a principy, na nichž je založen jakýkoli prvek počítačového programu, včetně těch, které jsou podkladem jeho propojení s jiným programem, nejsou podle autorského zákona chráněny.<sup>323</sup> Chráněno je pouze tvůrčí vyjádření anebo zpracování takovýchto prvků.<sup>324</sup> Ochranu rozhraní však právní řád oproti směrnici blíže nespecifikuje. Lze dojít k závěru, že implementace a specifikace rozhraní je jím chráněna v souladu se závěry uvedenými výše. Je také vhodné zmínit, že stejně jako v případě unijního práva není vyloučen souběh autorskoprávní ochrany s ochranou proti nekalé soutěži, budou-li splněny pojmové znaky nekalé soutěže.<sup>325</sup>

<sup>320</sup> Štědroň. *Ochrana a licencování počítačového programu*. Praha : Wolters Kluwer ČR, 2010, s. 220.

<sup>321</sup> Telec. Tůma. *Autorský zákon. Komentář. 2. vydání*, s. 714–721, bod 5.

<sup>322</sup> Tamtéž, s. 714–721, bod 5.

<sup>323</sup> § 65 odst. 2 AZ.

<sup>324</sup> Telec. Tůma. *Autorský zákon. Komentář. 2. vydání*, s. 714–721, bod 6.

<sup>325</sup> Tamtéž, s. 721–735, bod 11.

## 4.7 Právní úprava v USA

Obdobná situace dnes panuje ve Spojených státech amerických, a to ačkoli se díky jednomu z prvních soudních rozhodnutí<sup>326</sup> v tomto oboru jevila specifikace jako chráněný prvek počítačového programu, který je součástí struktury, sekvence a organizace.<sup>327</sup> Většina velkých technologických společností pochází právě ze Spojených států amerických. Není tedy náhodou, že se právní úprava počítačových programů začala nejdříve prosazovat tam. Vzhledem k rychlému rozvoji technologií a nově vznikajícím otázkám se nejdříve otázky týkající se ochrany počítačových programů, respektive jejich prvků, řešily na úrovni soudní. Soudní spory tak definovaly některé problémy, které následně zákonodárce mohl řešit na legislativní úrovni. Již v roce 1974 vznikla CONTU. Na základě výsledků jednání této komise bylo doporučeno chránit počítačový program jako autorské dílo.<sup>328</sup> V roce 1976 vznikl zákon o autorském právu (Copyright Act of 1976), který poskytoval ochranu autorským dílům, a ve kterém je stanoveno, že jsou z ochrany vyloučeny prvky díla, které jsou myšlenkami, procedurami, metodami či procesy.<sup>329</sup> Výslovné vyjmutí myšlenek z autorské ochrany naznačuje, že zákonodárce nezamýšlel umožnit autorovi tyto myšlenky skrýt před nabyvatelem licence ani za pomoci strojového kódu. Interoperabilita jako taková v zákoně však nebyla dostatečně řešena. Milníkem v rámci zákonodárství Spojených států amerických je přijetí novely tzv. „*Digital Millennium Copyright Act 1998*“. V ní jsou zohledněny v minulosti řešené spory týkající se rozhraní, funkčních prvků díla, myšlenek a procesů. Autorský zákon chrání vyjádření, nikoli myšlenky, které je tak možné v podstatě dále volně užívat, šířit a jinak s nimi nakládat. Zároveň je dle něj možné užít v určitých případech i vyjádření (viz fair use). Stejně jako v případě evropského práva i v USA je pro ochranu zásadní originalita.<sup>330</sup> Ta je téměř

---

<sup>326</sup> *Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222 (3d Cir. 1986).

<sup>327</sup> Samuelson. *The Past, Present and Future of Software Copyright Interoperability Rules in the European Union and United States*, s. 231.

<sup>328</sup> Menell. *Rise of the API Copyright Dead: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, s. 316.

<sup>329</sup> Samuelson. *The Past, Present and Future of Software Copyright Interoperability Rules in the European Union and United States*, s. 230.

<sup>330</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 36.

na stejné úrovni jako v evropském prostředí a jedná se spíše o původnost díla, tedy nikoli prostou kopií. Obecně tedy právo kopírovat dílo a využívat jej k tvorbě odvozených děl patří výlučně autorovi.

Zásadním je oddíl 102 (b), a to nejen pro ochranu počítačového programu, ale i pro interoperabilitu. Dané ustanovení stanoví, že myšlenky, nápady, metody apod. bez ohledu na formu, ve které jsou popsány nejsou chráněny autorským právem: „*In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.*“<sup>331</sup> Jedná se o negativní výčet prvků, kterým by v žádném případě neměla být poskytována autorskoprávní ochrana. Jsou vyňaty z ochrany bez ohledu na to, v jaké formě vyjádření jsou tyto nechráněné prvky obsaženy. Jedná se o uzákonění již zmíněné idea/expression dichotomie. Již v případě Baker v. Selden soud rozhodl, že se chrání kreativní vyjádření, nikoli metody, myšlenky nebo funkční a užitečné vyjádření. Soud zde v podstatě zavedl dichotomii vyjádření/myšlenka. Podle soudu se inovace (funkční vyjádření) mají chránit pomocí patentového práva<sup>332</sup>, které klade na dílo přísnější požadavky.<sup>333</sup> Takto rozdělená ochrana umožní technologické inovace.<sup>334</sup>

V rámci této dichotomie Graef řadí implementace rozhraní jakožto vyjádření v kódu pod chráněné vyjádření, pokud splňuje požadavek originality. Specifikaci rozhraní, považuje za pravidla a metody, které jsou základem rozhraní a řídí chování programu. Proto podle něj nespádají do rozsahu ochrany autorských práv.<sup>335</sup> Autorskoprávní ochrana tak umožňuje implementovat rozhraní se stejnou funkcí nebo za stejným účelem, pokud nedojde k přímému kopírování nebo nebude jinak narušena ochrana vyjádření.<sup>336</sup> Využití nechráněného rozhraní nesmí být

---

<sup>331</sup> Oddíl 102 (b) U.S Copyright Act.

<sup>332</sup> SCOTUS. *Baker v. Selden*, 101 U.S. 99 (1879), 103.

<sup>333</sup> Vasilescu-Palermo. *APIs and Copyright Protection: The Potential Impact on Software Compatibility in the Programming Industry*, s. 156.

<sup>334</sup> SCOTUS. *Baker v. Selden*, 101 U.S. 99 (1879), 102.

<sup>335</sup> Graef. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* s. 13.

<sup>336</sup> Mylly. *An Evolutionary Economics perspective on Computer Program Interoperability and Copyright*, s. 301.

považováno za porušení autorských práv, stejně jako není použití stejné myšlenky ve dvou knihách.

Je zřejmé, že i Spojené státy americké, respektive zákonodárce zamýšlel chránit veškeré myšlenky a nápady pouze pomocí patentu.<sup>337</sup> V současnosti je účinný spor mezi společnostmi Oracle a Google, ve kterém však v prvním stupni nebyl odvolací soud přesvědčen o spravedlivém použití rozhraní žalovaným a který nyní čeká na rozhodnutí Nejvyššího soudu USA.<sup>338</sup>

### 4.8 Shrnutí

V této kapitole jsme uvedli, v jakých případech lze dílo chránit autorským právem. Aby se mohlo jednat o dílo ve smyslu autorského práva, musí být vyjádřeno tak, že bude dostatečně přesně a objektivně identifikovatelné. Druhým požadavkem, který autorské právo na dílo klade je originalita. U standardních děl je to podmínka vhodná. U rozhraní však místo několika originálních děl přináší společnosti větší užitek zajištění interoperability, a tedy používání neoriginálních, stejných specifikací rozhraní. Rozhraní tak může být chráněno, bude-li v konkrétním případě dílem a bude splňovat požadavek originality. Vzhledem k tomu, že rozhraní je děleno na dva prvky, musí být tyto podmínky zkoumány u každého prvku zvlášť. Je zřejmé, že implementace rozhraní tak může být chráněna autorským právem. Pokud však implementaci není možné vyjádřit několika způsoby, nebude možná ani její ochrana. Specifikace rozhraní nebude obvykle originální z důvodu její technické a funkční předurčenosti.

I tehdy, splňuje-li konkrétní prvek počítačového programu veškeré náležitosti kladené na autorské dílo, nemusí být nutně autorským právem chráněn. Existují skutečnosti, které autorskoprávní ochranu vylučují nebo alespoň nějakým způsobem omezují. Zásadní roli zde hrají autorskoprávně volné složky. Do režimu *public domain* spadají také myšlenky a principy, na kterých jsou postaveny počítačové programy a jejich rozhraní. Pro specifikaci a implementaci rozhraní je tak podstatné, zda jsou podřazeny pod chráněné vyjádření nebo nechráněnou myšlenku.

---

<sup>337</sup> Cuillou. *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*, s. 178.

<sup>338</sup> Oracle v. Google je řešen v podkapitole 5.3.1.

Myšlenky mohou ostatní používat a rozvíjet bez toho, aniž by tím porušovali autorské právo.

V kapitole byly představeny přístupy, které zkoumají dichotomii myšlenka/vyjádření. *Merger doctrine* posuzuje existence tvůrčí volnosti, *abstractions test* odděluje neindividuální prvky díla a *scènes à faire* obvyklé prvky, které jsou pro konkrétní druh díla typické. Ačkoli je možné tyto přístupy používat, je důležité interpretovat je v souladu se středoevropským pojetím individuality autorského díla a vnitřní strukturou obecného fondu. Pomocí uvedených přístupů lze určit, zda je konkrétní prvek rozhraní chráněný autorským právem nebo nikoli. Specifikace rozhraní lze podřadit pod myšlenku počítačového programu, neboť se jedná o technickou informaci, která popisuje začátek zdrojového kódu, názvy, vstupy a výstupy a jiné informace nezbytné pro správné fungování rozhraní. Pro zachování interoperability musejí být specifikace rozhraní dvou počítačových programů stejné. Specifikace sama o sobě neumožňuje počítačovému programu vykonávat jeho funkcionality, pouze je definuje. Specifikace rozhraní tedy nelze podřadit pod vyjádření počítačového programu, a proto specifikace rozhraní nejsou způsobilé pro ochranu autorských práv. Stejně tak nemohou být chráněny prvky, které mají svoji podobu danou z důvodů funkčnosti, a to i tehdy, kdy se myšlenka a vyjádření spojily. I z těchto důvodů nelze specifikaci rozhraní chránit autorským právem. U implementace rozhraní je v případě dodržení specifikace rozhraní možné obvykle více způsobů vyjádření a bude tedy záležet na tom, zda je shledána originální a zda jej lze oddělit od myšlenky.

Pro české prostředí je relevantní snaha Evropské unie upravit právní režim rozhraní pomocí směrnice o právní ochraně počítačových programů. Tato směrnice harmonizuje právní úpravu ochrany počítačových programů v rámci Evropské unie a vyjímá z ochrany rozhraní počítačových programů, která lze podřadit pod myšlenky. I z textu směrnice lze dojít k závěru, že specifikace rozhraní není chráněna. Jedná se o natolik funkční a svou funkcí předurčený prvek počítačového programu, že by jeho ochrana byla na újmu společnosti. Ochrana by byla možná pouze v případě implementace rozhraní, kterou by bylo možné vyjádřit několika odlišnými způsoby. Ačkoli se autorské právo přiklání k tomu, že specifikace rozhraní není chráněna a implementace pouze tehdy pokud je originální a není spojena s myšlenkou, stále není vyřešen problém přístup k těmto informacím nezbytným pro interoperabilitu. Z podstaty nechráněného prvku autorského díla k němu má mít každý umožněný

přístup. Ten je však zcela nedostatečný, viz spor Microsoft v. Evropská komise.<sup>339</sup> Často je tak nutné přikročit ke způsobům, jak informace o rozhraní získat. Tím, který je v autorském právu nejrozebíranějším, je zpětné inženýrství.

---

<sup>339</sup> Judgment of the Court of First Instance of 17 September 2007. T-201/04. *Microsoft Corp. v Commission of the European Communities*.



## 5 Přístup k rozhraní

Přestože stojí informace nezbytné pro interoperability spíše mimo ochranu autorského práva, dochází k problémům ohledně přístupu k nim. Pokud tyto informace nezveřejní přímo autor, není vyjma zpětného inženýrství příliš možností, jak informace získat.<sup>340</sup> Přístup k takto nechráněným prvkům nebo myšlenkám je žádoucí, není však vždy možný bez dodatečné kopie, překladu nebo jiného zásahu do počítačového programu. Pro společnost je u autorských děl, která byla zveřejněna, důležité mít přístup k jejich vyjádření i k myšlence, na které jsou postavené. Počítačový program, včetně jeho rozhraní, je založený na zdrojovém a strojovém kódu. Ani jeden z nich není viditelný pro uživatele, kterým byl program zpřístupněn.<sup>341</sup> Běžný uživatel má přístup pouze k funkcím daného programu. Je na volbě autora, zda myšlenku skrytou v kódu zpřístupní nebo nechá skrytou. Obvykle se tak přístupné stává pouze vyjádření v podobě strojového kódu, nikoli myšlenka počítačového programu.<sup>342</sup> Strojový kód pro získání myšlenek není dostačující.<sup>343</sup> Myšlenka tedy není přístupná okamžitě a bez omezení, jako tomu je u jiných děl. Autorské právo je však založené na přístupu k oběma částem. Oprávněný uživatel počítačového programu by také měl mít přístup ke všem myšlenkám. Počítačový program, který je distribuován ve strojovém kódu toto neumožňuje a myšlenky v něm obsažené nelze bez možnosti zpětného inženýrství studovat a pochopit.<sup>344</sup> Proto se postupem času vyvinuly dva hlavní směry umožňující přístup k takto chráněnému rozhraní. Směr, který vznikl ve Spojených státech amerických umožňuje

---

<sup>340</sup> Graef. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* s. 15.

<sup>341</sup> Karjala. *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller Symposium: Copyright Protection and Reverse Engineering of Software*, s. 992.

<sup>342</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 28-29.

<sup>343</sup> Karjala. *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller Symposium: Copyright Protection and Reverse Engineering of Software*, s. 992.

<sup>344</sup> Committee on Computer Law, Reverse Engineering and Intellectual Property Law, s. 140.

zpětné inženýrství za účelem interoperability pod doktrínou *fair-use*. Evropský směr umožňuje zpětné inženýrství za účelem interoperability na základě zákonných limitací rozsahu autorských práv, které lze najít např. ve směrnici o ochraně počítačového programu.<sup>345</sup>

Vývojáři mají zájem na tom, aby autorským právem bylo zabráněno možnosti kopírovat a překládat kód, a to i přes zájem na interoperabilitě. Přesto však myšlenky nelze autorským právem chránit. Je totiž převažující zájem společnosti na jejich využívání.<sup>346</sup> Pokud by zpětné inženýrství nebylo umožněno, měli by autoři monopol jak na vyjádření, tak na myšlenky, které by zůstaly v počítačovém programu uschovány. Autorské právo by v žádném případě nemělo chránit, ani nepřímo, myšlenky autorského díla. Monopol plynoucí z nemožnosti přístupu k myšlenkám by byl umocněn, neboť doba ochrany je u autorského díla významně delší, než je tomu například u patentu. Je zřejmé, že zákonodárce nezamýšlel chránit myšlenky v žádném rozsahu a bylo by proti smyslu autorského práva, aby poskytovalo nepřímou ochranu těchto myšlenek pomocí nemožnosti se k nim dostat legální cestou, tedy zákazu překládání kódu. Z toho důvodu musí být autorským právem umožněno program překládat, studovat a kopírovat tak, aby z něj mohly být získány nechráněné informace o rozhraní.<sup>347</sup> Znemožnění zpětného inženýrství by bylo proti úmyslu zákonodárce a smyslu autorských zákonů, neboť by bránilo rozvoji technologií a přístupu k nechráněným myšlenkám.<sup>348</sup> Cílem zpětného inženýrství tedy nemá být kopírování díla, ale umožnění technologického rozvoje v softwarovém průmyslu.<sup>349</sup>

---

<sup>345</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 67.

<sup>346</sup> Karjala. *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller Symposium: Copyright Protection and Reverse Engineering of Software*, s. 993.

<sup>347</sup> Spoor. *Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive Symposium: Copyright Protection and Reverse Engineering of Software*, s. 1069-1071; Srov. Graef. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* s. 12.

<sup>348</sup> Daughtrey. *Reverse Engineering of Software for Interoperability and Analysis Note*, s. 179.

<sup>349</sup> Mylly. *An Evolutionary Economics Perspective on Computer Program Interoperability and Copyright*, s. 308.

## 5.1 Směrnice o právní ochraně počítačových programů

Možnost zpětného inženýrství za účelem interoperability zavádí v právním řádu Evropské unie zejména směrnice o právní ochraně počítačových programů. Konkrétně v článku 5 a článku 6 směrnice umožňuje do značné míry dekompilaci a blackbox testing. Pokud má uživatel díla zájem o získání a pochopení v rozhraní skrytých myšlenek a funkcí, mezi které lze podřadit např. právě specifikaci rozhraní, musí mít uživatel možnost rozhraní, respektive počítačový program studovat a analyzovat jeho zdrojový kód.<sup>350</sup> Nabyvatelé, kteří mají zájem o zjištění myšlenek počítačového programu pomocí dekompilace, musí mít nejprve přístup k programu ve formě zdrojového kódu. Dekompilace je upravena právě v čl. 6 směrnice.<sup>351</sup> Veškeré znalosti a vědomosti, které programátor do vývoje programu vložil, na kterých program stojí a které jsou programátory nejvíce ceněny jsou dekompilací mnohem více ohroženy, než by tomu bylo u blackbox testingu.<sup>352</sup> Je tomu tak, protože pomocí dekompilace se osoba provádějící zpětné inženýrství dostane k vyjádření ve formě zdrojového kódu, kdežto blackbox testing se zdrojovým kódem přímo nepracuje. Metoda zpětného inženýrství tzv. blackbox testing je proto umožňována již článkem 5 směrnice.<sup>353</sup> Tento článek umožňuje pozorování počítačového programu za účelem zjištění myšlenek a zásad. Van Rooijen upozorňuje, že i pomocí blackbox testingu lze v některých případech úspěšně dosáhnout interoperability a není tedy vhodné na něj zapomínat a věnovat se pouze dekompilaci.<sup>354</sup>

Dekompilace umožněná čl. 6 je jednou z možností, která může vést k získání informací o interoperabilitě počítačového programu, a proto je jí také v této směrnici věnován značný prostor. Čl. 6 dovoluje dekompilaci bez svolení nositele práv v případě, že je nezbytná pro získání informací o interoperabilitě nezávisle vytvořeného počítačového programu s jinými programy. Z textu směrnice podle Cuillou není zřejmé, jak široce

---

<sup>350</sup> Daughtrey. *Reverse Engineering of Software for Interoperability and Analysis Note*, s. 174.

<sup>351</sup> Čl. 6 Směrnice o právní ochraně počítačových programů.

<sup>352</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 79.

<sup>353</sup> Čl. 5 Směrnice o právní ochraně počítačových programů.

<sup>354</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 81.

je interoperabilita chápána. Na základě sdělení Evropské komise<sup>355</sup> má spíše za to, že je umožněna dekompilace za účelem interoperability jak při vývoji kompatibilního, tak konkurenčního programu.<sup>356</sup> Jediné omezení je zde zákaz podstatné podobnosti ve vyjádření vyvíjeného programu.<sup>357</sup> Článek 6 tedy povoluje dekompilaci za účelem zjištění informací nezbytných pro interoperabilitu, a implementaci těchto informací do nového, nezávisle vytvořeného programu, který umožňuje novému programu spolupracovat s jinými programy.<sup>358</sup>

### 5.1.1 Podmínky dekompilace podle směrnice

Aby bylo umožněno zpětné inženýrství, musí být splněny další podmínky. Jednou z těchto podmínek je, že dekompilaci musí provádět osoba, která má oprávnění dílo užívat, tedy např. nabyvatel licence, oprávněný uživatel rozmnoženiny apod. Dále nesmí být dosahované informace rozkladateli přístupné jinak, rychleji a snadněji, a musí být zasaženo jen do té části díla, která je pro dosažení těchto informací nezbytná.<sup>359</sup> Odst. 1 čl. 6 směrnice tedy míří na případy, ve kterých je dekompilace povolena.

Oproti tomu další odstavce stanoví případy, ve kterých dojde k porušení práv autora i přesto, že budou splněny podmínky pro dekompilaci díla stanovené v odst. 1. Zpětný inženýr nesmí použít získané informace k jiným účelům než k dosažení interoperability počítačového programu vytvořeného nezávislým způsobem. Nesmí informace získané pomocí dekompilace předávat třetím stranám. Toto omezení neplatí tehdy, je-li předání nezbytné k dosažení interoperability nezávisle vytvořeného počítačového programu.<sup>360</sup> Zejména nesmí rozkladatel informace použít

---

<sup>355</sup> European Commission, Communication from the Commission to the European Parliament, 1991 Parl. Eur. Doc. (SEC No. 87), final – SYN 183, para. 4. 7, at 5 (18 Jan. 1991).

<sup>356</sup> Cuillou. *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*, s. 543.

<sup>357</sup> Čl. 6 odst. 2 písm. c) Směrnice o právní ochraně počítačových programů.

<sup>358</sup> Čl. 6 odst. 1 Směrnice o právní ochraně počítačových programů; Srov. Samuelson. Vinje. Cornish. *Does Copyright Protection Under the EU Software Directive Extend to Computer Program Behaviour, Languages and Interfaces?* s. 13.

<sup>359</sup> Čl. 6 odst. 1 Směrnice o právní ochraně počítačových programů.

<sup>360</sup> Čl. 6 odst. 2 písm. b) Směrnice o právní ochraně počítačových programů.

pro vývoj, výrobu ani odbyt počítačového programu, jehož vyjádření bude podstatně podobné původnímu dílu ani k jakémukoliv jinému úkonu porušujícímu autorská práva.<sup>361</sup> Proti těmto omezením se postavil například ECIS.<sup>362</sup> Podle ECIS není důvod takto zpětné inženýrství omezovat, protože u rozhraní, které je myšlenkou nebo principem, je autorskoprávní ochrana tak či tak vyloučena.<sup>363</sup> Noam Shemtov s takovouto myšlenkou souhlasí.<sup>364</sup> Podle něj by mělo být možné zpětné inženýrství kdykoli, jednalo-li by se o prvky programu spadající pod pojmy jako nápad či myšlenka. Neomezoval by jej tedy pouze na rozhraní. K tomu ještě doplňuje, že by možná bylo vhodnější umožnit zpětné inženýrství za účelem oprávněných zájmu (viz fair use v USA) a neomezovat se pouze na umožnění zpětného inženýrství pro zajištění interoperability.<sup>365</sup>

I AZ pracuje s výjimkami umožňujícími konkrétní zásah do počítačového programu.<sup>366</sup> Lze dojít k tomu, že zákonodárce pouze nepoužil pojem rozhraní tak, jak je použit ve směrnici. Místo toho pracuje s pojmem *propojení*, který situaci ještě více znepréhlednil.<sup>367</sup> AZ tedy umožňuje učinit kroky nezbytné k účelnému propojení počítačových programů.<sup>368</sup> Oprávněný uživatel může zasahovat do autorského práva na rozmnožování díla a práva na nedotknutelnost díla. K tomu dochází právě za účelem dosažení interoperability. AZ výslovně uvádí možnost pořídit si rozmnoženinu počítačového programu anebo možnost přeložit formu kódu počítačového programu.<sup>369</sup>

Důležité je, aby rozmnoženinu kódu nebo jeho překlad prováděl oprávněný uživatel nebo jím pověřená osoba. Zároveň musí jít o kroky nezbytné. Lze tedy mít za to, že nesmí existovat jiná možnost, jak

---

<sup>361</sup> Čl. 6 odst. 2 Směrnice o právní ochraně počítačových programů.

<sup>362</sup> Gilbert-Macmillan, K. *Intellectual Property Law for Reverse Engineering Computer Programs in the European Community*. 9 Santa Clara High Tech. L.J. 247 (1993). s. 256.

<sup>363</sup> Tamtéž.

<sup>364</sup> Shemtov, N. *Beyond the Code: Protection of Non-Textual Features of Software*. Oxford University Press, 2017, s. 288.

<sup>365</sup> Tamtéž.

<sup>366</sup> § 66 AZ.

<sup>367</sup> § 66 odst. 1 písm. e) AZ.

<sup>368</sup> Chaloupková, Holý. *Autorský zákon, 5. vydání*, Nakladatelství C.H. Beck, 2017, s. 144–150.

<sup>369</sup> Telec. Tůma. *Autorský zákon. Komentář. 2. vydání*, s. 721–735 bod 6.

dosáhnout propojení. Ustanovení AZ neumožňuje tyto kroky provádět tehdy, je-li informace o propojení dostupná rychle a snadno jinak. Podle Telce jde o situace, kdy jsou informace dostupné z odborné literatury, na trhu nebo například na internetu.<sup>370</sup> Zároveň by se mělo jednat o zásah do programu v co nejmenším rozsahu, který je pro získání informací, nezbytných pro interoperabilitu, dostačující.<sup>371</sup>

Právě díky § 66 AZ může oprávněný uživatel dekompilovat program za účelem zjištění myšlenek počítačového programu.<sup>372</sup> Podle Telce je AZ částečně v rozporu s unijním právem, neboť omezuje rozsah poskytnuté licence. Unijní úprava totiž umožňuje zasahovat do díla každému oprávněnému uživateli, kdežto AZ stanoví, že do počítačového programu může zasahovat oprávněný uživatel, který je k zásahu oprávněn.<sup>373</sup> Tato situace nastala zřejmě z důvodu implementaci několika unijních článků do jednoho paragrafu v AZ. Čl. 6 odst. 1 Směrnice o právní ochraně počítačových programů totiž obsahuje možnost provádět dekompilaci *oprávněným uživatelem* teprve v písm. a), kdežto AZ toto ustanovení přejal do § 66 odst. 1 písm. e) aniž by zohlednil, že podmínka *oprávněného uživatele* je již obsažena v první větě § 66 odst. 1 AZ.

Mimo to, kdy je zásah za účelem propojení dovolený, upravuje AZ rovněž situaci po získání informací o interoperabilitě. Dekompilací získané informace by měly být použity pouze k vytvoření počítačového programu, který bude splňovat podmínku tvůrčí původnosti. V případě vývoje konkurenčního programu by mohly být porušeny oprávněné zájmy autora.<sup>374</sup> Získané informace nesmí být zneužity k vývoji, zhotovení nebo k obchodnímu využití počítačového programu podobného původnímu počítačovému programu v jeho vyjádření. Informace nesmějí být poskytnuty jiným osobám a zároveň AZ zamezuje jejich použití k jinému jednání ohrožujícímu nebo porušujícímu právo autorské.<sup>375</sup> Pro úplnost je nutné zmínit tzv. třístupňový test. Zásah do díla musí splňovat podmínky v něm určené. Nestačí tedy, že zásahy do programu spadají pod některou z výjimek v § 65 nebo § 66 AZ, je třeba, aby zároveň byly splněny podmínky v tomto testu stanovené. Konkrétně se jedná o zákaz užití díla

---

<sup>370</sup> Telec. Tůma. *Autorský zákon. Komentář. 2. vydání*, s. 721–735 bod 6.

<sup>371</sup> § 66 odst. 1 písm. e) AZ.

<sup>372</sup> Telec. Tůma. *Autorský zákon. Komentář. 2. vydání*, s. 721–735, bod 6.

<sup>373</sup> Tamtéž.

<sup>374</sup> Tamtéž, bod 11.

<sup>375</sup> § 66 odst. 4 AZ.

v rozporu s běžným způsobem užití díla a zákaz nepřiměřeného dotčení oprávněných zájmů autora.<sup>376</sup> Chaloupková také správně připomíná, že je nadále u takto získaných informací nutné rozlišovat mezi chráněným vyjádřením a nechráněnými principy a myšlenkami.<sup>377</sup>

## 5.2 Nedostatky právní úpravy zpětného inženýrství

Vzhledem ke kompromisům při přijímání směrnice dochází nyní k několika interpretačním problémům. Mimo jiné může způsobovat komplikace nedefinování mnoha pojmů a jejich následný odlišný výklad.<sup>378</sup> Problémem je zajisté hned definice rozhraní. Vzhledem k jeho ne zcela detailní definici, se při vnímání rozsahu rozhraní může projevit rozdílná interpretace. Rozhraní standardně bývá vyjádřeno zdrojovým kódem. To se však nedá tvrdit o každém rozhraní. Rozhraní, které není objektivně vyjádřeno pak ani nemůže být chráněno autorským právem. Specifikace takto nevyjádřeného rozhraní musí být chápána jako charakteristika určující jeho funkcionalitu ne jako vyjádření. To však nelze tvrdit paušálně o všech rozhráních. Vždy je tedy nutné rozhraní zkoumat individuálně a poskytovat rozhraní ochranu podle toho, zda je spíše vyjádřením nebo myšlenkou.<sup>379</sup>

Možné nedostatky mohou být spatřovány rovněž v článku 6 odst. 1 písm. b), který stanoví pravidlo, podle kterého nesmí být informace o interoperabilitě přístupné jinak. Zpětný inženýr má často omezené možnosti dokázat, že informace o rozhraní mu nebyly zpřístupněny dříve, než provedl dekompilaci. Není zcela jisté, zda to pro autory znamená povinnost na žádost informace poskytnout. Není jisté ani to, zda si má nabyvatel díla nejprve informace vyžádat. Také ze znění ustanovení není zřejmé, kolik úsilí musí nabyvatel díla vynaložit na to, aby se pokusil informace od autora získat, než je možné na informace pohlížet jako na

---

<sup>376</sup> Telec. Tůma. *Autorský zákon. Komentář. 2. vydání*, s. 380–386, bod 6.

<sup>377</sup> Chaloupková, Holý. *Autorský zákon, 5. vydání*, s. 144–150.

<sup>378</sup> Cuillou. *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*, s. 542.

<sup>379</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*.

nepřístupné.<sup>380</sup> Pravděpodobně je zde ponechán prostor pro uvážení autora, zda informace zpřístupní dobrovolně nebo raději nechá uživatele provést dekompilaci.<sup>381</sup> Nejasná situace by mohla vést k případům, ve kterých se budou autoři pokoušet o získání finančního prospěchu za to, že umožní přístup k informacím o rozhraní. Otázkou je, zda by bylo nutné takto zpoplatněné informace přijmout, aby nenastal rozpor s tímto ustanovením. Autor by sice tímto způsobem obcházel ustanovení autorského práva a žádal odměnu za prvek, který autorským právem není chráněn. Nicméně podmínka zpřístupnění informací před provedením dekompilece by mohla být soudy brána za splněnou. Bylo by tedy vhodné, kdyby zákonodárce přiblížil, kde a v jakém rozsahu by měly být informace přístupné. Zda mají být přístupné automaticky nebo na žádost a zda za ně například náleží autorovi licenční odměna.<sup>382</sup>

Další nejasný pojem obsahuje čl. 6 odst. 2 písm. c), ve kterém je zakázáno použít získané informace za účelem vytvoření programu, který se bude podstatně podobat původnímu programu svým vyjádřením. Není však zřejmé, co se podstatnou podobností myslí. Shemtov má za to, že se jedná o podstatnou podobnost v doslovném zdrojovém kódu.<sup>383</sup> Díky tomu je umožněna podobnost ve funkcionalitách mezi původním a novým programem.

Mimo to směrnice neřeší interoperabilitu v celé své komplexnosti, neboť se zaměřuje pouze na interoperabilitu dvou počítačových programů.<sup>384</sup> Důležitá je však rovněž interoperabilita mezi software a hardware. To by mohlo mít za následek nemožnost zjišťování informací o interoperabilitě, nezbytných pro spuštění software v novém hardwarovém prostředí.<sup>385</sup>

---

<sup>380</sup> Cuillou. *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*, s. 544.

<sup>381</sup> Mylly. *An Evolutionary Economics perspective on Computer Program Interoperability and Copyright*, s. 310.

<sup>382</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 82.

<sup>383</sup> Cuillou. *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*, s. 544.

<sup>384</sup> Čl. 6 odst. 1 písm. Směrnice o právní ochraně počítačových programů: „...o interoperabilitě nezávisle vytvořeného počítačového programu s jinými programy.“

<sup>385</sup> Mylly. *An Evolutionary Economics perspective on Computer Program Interoperability and Copyright*, s. 309.



Je-li umožněna dekompilace za účelem nalezení specifikace rozhraní, pak není zcela jasné, zda se může jednat o možnost provádět zpětné inženýrství pouze u konkrétní části nebo u programu celého. Čl. 6 odst. 1 písm. c) dovoluje provádět zpětné inženýrství pouze u té části programu, která je pro interoperabilitu nezbytná. Vzhledem k tomu, že zpětný inženýr pravděpodobně nebude mít informace o tom, v které části má informace hledat, bude nucen dekompilovat celý program.<sup>386</sup>

Směrnice se příliš nezabývá technologií zpětného inženýrství. Technologický pokrok a změna prostředí může vést k tomu, že rovnováha, které se směrnice snaží dosáhnout, bude narušena. Je možné, že v budoucnu bude zpětné inženýrství běžný, levný a rychlý proces a výhoda tak bude posunuta na stranu zpětného inženýra.<sup>387</sup> Může tomu však být i naopak a zpětné inženýrství nemusí být schopné žádné informace vyjevit. Zpětné inženýrství složitých rozhraní může navíc zabrat velké množství času<sup>388</sup> a zároveň vyžaduje od vývojářů značné úsilí a investice.<sup>389</sup> Rovněž není zaručeno, že implementace rozhraní povede k interoperabilitě.<sup>390</sup>

Směrnice zároveň příliš neřeší fáze následující po provedení zpětného inženýrství. Například chybí informace o tom, zda lze implementaci rozhraní, respektive jeho zdrojový kód kopírovat nebo se spokojit se specifikací rozhraní. Ačkoli toto směrnice výslovně neřeší, lze dojít k závěru, že lze použít specifikaci rozhraní, jakožto pro zajištění interoperability,

---

<sup>386</sup> Cuillou. *The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach*, s. 544.

<sup>387</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 87.

<sup>388</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*, s. 34, 14.

<sup>389</sup> Rozhodnutí Evropské komise v řízení podle článku 82 [ES] a článku 54 Dohody o EHP zahájeném proti *Microsoft Corp*, ze dne 24. března 2004, sp. zn. COMP/C-3/37.792 – Microsoft, recitál 36.

<sup>390</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*, s. 34, 14.

dostačující prvek. Implementace rozhraní, která je vyjádřena ve zdrojovém kódu je chráněna autorským právem jako vyjádření.<sup>391</sup>

Naopak tam, kde směrnice řeší fáze následující po provedení zpětného inženýrství, obsahuje velké množství omezujících ustanovení, která brání jednoduššímu přístupu k informacím o rozhraní. Jedním z takových omezení je nemožnost poskytnout získané informace o rozhraní ostatním subjektům na trhu. Každá společnost si tak musí provést zpětné inženýrství samostatně, což navyšuje náklady a prodlužuje dobu, po kterou je rozhraní nepřístupné. Zároveň mohou některé společnosti být z trhu vytlačeny, protože nedisponují dostatečnými prostředky umožňující zpětné inženýrství. Výše jsme si uvedli, že přístup k rozhraní by neměl být jakkoli omezen, stejně jako není u jiných děl, a toto omezení přístupu by nemělo být nadále rozšiřováno nemožností poskytnout získané informace trhu.<sup>392</sup>

### 5.2.1 Technické nedostatky zpětného inženýrství a zájem na dobrovolném licencování

Jak směrnicí o ochraně počítačových programů v Evropské unii, tak doktrínou fair use ve Spojených státech amerických je za účelem interoperability upřednostňována varianta umožňující za jistých okolností zpětné inženýrství. Zpětné inženýrství za účelem zajištění interoperability, stejně jako jiné metody, má své nedostatky. Potřebuje-li nabyvatel počítačového programu provést zpětné inženýrství proto, aby si zajistil interoperabilitu, ani při veškeré opatrnosti nelze vždy dosáhnout toho, že neodhalí části kódu, které k interoperabilitě nezbytné nejsou. Takové části však jsou chráněny.<sup>393</sup>

Zpětné inženýrství není jednoduchý proces, vyžaduje značné odborné znalosti a finanční prostředky.<sup>394</sup> Stejně tak nemá osoba provádějící zpětné inženýrství jistotu v tom, že dosáhne informací, které hledá.

---

<sup>391</sup> Např. Rozsudek Soudního dvora ze dne 2. května 2012, C-406/10, *SAS Institute Inc. v. World Programming Ltd*, bod. 38.

<sup>392</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 86.

<sup>393</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 51.

<sup>394</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 89.

V případě rozsáhlejších, komplexnějších programů pak zpětné inženýrství za účelem zjištění informací o rozhraní nemusí být realizovatelné vůbec.<sup>395</sup> To vše může bránit zájmu zpětné inženýrství provádět. Prokázat skutečnost, že provedené zpětné inženýrství není pro zajištění interoperability dostačující bude problematické. Ještě závažnější může být pro zpětného inženýra, pokud autor provede změny specifikace rozhraní poté, co zpětný inženýr provedl zpětné inženýrství.<sup>396</sup> Je evidentní, že zpětné inženýrství doprovází několik problematických aspektů.

Směrnice tedy neslouží, respektive by neměla sloužit jako primární instrument k zajištění interoperability. Měla by být spíše v roli alternativního instrumentu, selžou-li instrumenty jí předcházející. Tím základním by měla být zejména motivace autorů zpřístupnit specifikaci rozhraní.<sup>397</sup> Ideální varianta je zpřístupnění informací o rozhraní podle stranami nastavených podmínek. V takovém případě by nebyla dekompilace možná. Směrnice totiž zakazuje zpětné inženýrství vedoucí k získání informace nezbytné k dosažení interoperability, která je pro zpětného inženýra snadno a rychle přístupná jinak.<sup>398</sup> Zákodárce při přijímání směrnice možná příliš spoléhal na zájmy autorů a motivaci dobrovolně rozšiřovat svoji síť. Měl za to, že existuje značné množství důvodů, proč by autoři měli své rozhraní sdílet dobrovolně. Ať už jde o zabránění zpětnému inženýrství, rychlejší tvorbu interoperabilních systémů, větší základnu uživatelů a vývojářů nebo jiné motivy. Je přinejmenším vhodné motivovat společnosti k dobrovolnému licencování, díky kterému bude zpětné inženýrství zbytné.<sup>399</sup> Tento cíl však nebyl zcela naplněn.

Evropská komise si je vědoma nedostatků v úpravě interoperability týkající se počítačových programů. Ve svém pracovním dokumentu analyzuje proveditelnost opatření, která by mohla vést významné hráče na trhu k licencování informací o interoperabilitě.<sup>400</sup> Evropská komise se

---

<sup>395</sup> Graef. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* s. 13.

<sup>396</sup> Tamtéž.

<sup>397</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 89-90.

<sup>398</sup> Čl. 6 odst. 1 písm. b) Směrnice o právní ochraně počítačových programů.

<sup>399</sup> Samuelson, P., Scotchmer, S. *The Law & Economics of Reverse Engineering*, s. 1616.

<sup>400</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*.

v dokumentu zaměřuje spíše na řešení situace *ex ante*. Práva na ochranu hospodářské soutěže, pomocí kterých je interoperabilitu také možné zajistit, jsou jakožto právní úprava *ex post* upozadována.<sup>401</sup>

Komise zvažovala legislativní opatření, která nicméně z důvodu nákladnosti, technické obtížnosti a dalších možných komplikací zamítla.<sup>402</sup> Je však vhodné zmínit zvažovaná opatření nelegislativní, která by mohla vést k lepší interoperabilitě a zároveň zachovala důvěru v stabilitu právního řádu. Nelegislativní opatření také budou mít rychlejší dopad na trh, sníží transakční náklady a podpoří kulturu licencování.<sup>403</sup> Evropská komise v *soft law* vidí značný potenciál. Záměrem je snížení bariér bránících ve vstupu na trh menším soutěžitelům a vývojářům. Graef zde upozorňuje, že problémy interoperability se netýkají pouze na trhu významných společností, ale také společností menších.<sup>404</sup>

Prvním návrhem Evropské komise je zavedení osvědčených postupů týkajících se dostupnosti informací o interoperabilitě. V podstatě jde o zavedení standardizovaných procesů usnadňujících komunikaci mezi zájemcem o informace o interoperabilitě a jejich držitelem.<sup>405</sup>

Stejně tak by mohlo pomoci zavedení standardizovaných licenčních podmínek, díky kterým by byly významně šetřeny transakční náklady.<sup>406</sup> Problém by mohl nastat v rychlém zastarání takto vytvořeného licenčního modelu. Na jeho dynamický vývoj však Evropská komise myslela a navrhovala vytvořit tyto licence na základě Creative Commons.<sup>407</sup>

Dalším bodem, na kterém by mohlo podle Evropské komise jednání o licenci ustrnout, je hodnota informací o interoperabilitě. Bylo by tedy

---

<sup>401</sup> Graef, I., Valcke, P. *Exploring new ways to ensure interoperability under the Digital Agenda*. Emerald Group Publishing Limited, 2014, č. 1, s. 2.

<sup>402</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*. s. 14-15.

<sup>403</sup> Graef. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* s. 18.

<sup>404</sup> Tamtéž.

<sup>405</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*, s. 15.

<sup>406</sup> Tamtéž.

<sup>407</sup> Tamtéž.

možné přistoupit k vypracování vzorového modelu, na základě kterého by strany jednání mohly určit hodnotu informací o rozhraní.<sup>408</sup>

Komise v dokumentu navrhuje jako další alternativní možnost zavázat společnost, která obdrží dotaci k tomu, aby informace o rozhraní zpřístupnila.<sup>409</sup> Mimo tento poslední případ se jedná o návrhy, které autory, kteří nemají zájem o zveřejnění informací o interoperabilitě nedonutí k jejich zveřejnění. Soft-law neumožní autoritativní vynucení zpřístupnění informací a jako takové je nedostatečné. Návrhy nejsou vhodné ani z toho důvodu, že specifikace rozhraní je myšlenou, kterou autorské právo nechrání. Valcke se domnívá, že by bylo vhodnější, pokud by se Evropská komise rozhodla vyjasnit při kterých situacích na trhu by informace o interoperabilitě měly být sdíleny. Na druhou stranu uznává, že soft-law umožní Evropské komisi sledovat dopad regulace ex ante a toto následně vyhodnotit a upravit.<sup>410</sup>

Ačkoli není zpětné inženýrství ideální postup k získání informací, je v současné době jeden z mála, pomocí kterého informace lze získat. Interoperabilita dosahována pomocí informací získaných zpětným inženýrstvím, které je umožněno autorským právem, respektive směrnicí by mohla být značně omezena, bylo-li by zpětné inženýrství omezeno jinými právy duševního vlastnictví.

## 5.2.2 Obchodní tajemství a patent

Důležitou otázkou je, zda lze omezit zpětné inženýrství jinými formami právní ochrany. Typicky se jedná o obchodní tajemství nebo patent. Podle Spoor *„některé země dokonce poskytují patenty týkající se rozhraní.“*<sup>411</sup> Důležitost interoperability a výše uvedených článků směrnice se naštěstí projevila rovněž v Dohodě o jednotném patentovém soudu. Ta v článku 27 omezuje účinky jednotného patentu, které se tak nesmí

---

<sup>408</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*, s. 16.

<sup>409</sup> Tamtéž.

<sup>410</sup> Graef. Valcke. *Exploring new ways to ensure interoperability under the Digital Agenda*, s. 8.

<sup>411</sup> Spoor. *Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive Symposium: Copyright Protection and Reverse Engineering of Software*, s. 1073.

vztahovat na „*úkony a používání získaných informací dovolené podle článků 5 a 6 směrnice 2009/24/ES (3), a zejména podle jejich ustanovení o rozkladu a interoperabilitě*“.<sup>412</sup> Patent s unijním účinkem tedy nesmí chránit informace získané podle článků 5 a 6 směrnice o právní ochraně software. Zde je opět viditelný zájem na zpřístupnění specifikace rozhraní a snaha zamezit společnostem omezovat přístup k rozhraní jinými než autorskými právy.<sup>413</sup> Jednotný evropský patent tak má za následek, že jím nebude bráněno v provádění dekompilace za účelem zajištění interoperability. Národní patenty však tímto předpisem nejsou dotčeny. Nestanoví-li národní předpisy obdobnou výjimku pro interoperabilitu, bude nadále možné interoperabilitě v konkrétních případech bránit.<sup>414</sup>

Bráněno zpětnému inženýrství by nemělo být ani obchodním tajemstvím. To potvrdila i Směrnice o obchodním tajemství. Není v ní však řešena souvislost se směrnicí o právní ochraně počítačových programů. Rovněž v ní chybí limitace v používání zpětným inženýrstvím získaných informací, jako tomu je v případě směrnice o právní ochraně počítačových programů.<sup>415</sup> Přijatá verze směrnice na ochranu obchodního tajemství dovoluje zpětné inženýrství tam, kde nabyvatel je bez povinnosti omezit získávání obchodního tajemství: „*Získání obchodního tajemství se pokládá za oprávněné, pokud k němu došlo některým z těchto způsobů: pozorováním, zkoumáním, demontáží nebo testováním výrobku nebo předmětu, který byl zpřístupněn veřejnosti nebo který je oprávněně v držení osoby, která informace získala a nemá žádnou právně závaznou povinnost omezit získávání daného obchodního tajemství.*“<sup>416</sup> Takové právně závazné povinnosti by podle Weston mohl odpovídat právě čl. 6 směrnice, který povinnost zavádí. Zpětné inženýrství by pak bylo v rámci ochrany

---

<sup>412</sup> Čl. 27 písm. k) Dohody o Jednotném patentovém soudu, 2013/C 175/01, EN - EUR-Lex [online]. [cit. 26. 2. 2021]. <https://eur-lex.europa.eu/legal-content/CS/TXT/?uri=celex%3A42013A0620%2801%29>

<sup>413</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 91-92.

<sup>414</sup> Graef. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* s. 15-16.

<sup>415</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 82.

<sup>416</sup> Čl. 3 Směrnice o ochraně nezveřejněného know-how a obchodních informací (obchodního tajemství) před jejich neoprávněným získáním, využitím a zpřístupněním.

obchodního tajemství nelegální. Weston se ale přiklání k tomu, že povolení dekompileovat počítačový program zůstává v souladu se softwarovou směrnicí se všemi jejími omezeními.<sup>417</sup>

### 5.3 Zpětné inženýrství za účelem interoperability v USA

Ačkoli není rozsah ochrany rozhraní americkým právním řádem explicitně řešen, v podkapitole 4.7 jsme si ukázali, že ochrana rozhraní je odvislá od toho, zda konkrétní rozhraní lze podřadit pod myšlenku nebo vyjádření. Podle dosavadní soudní praxe jsou myšlenky rozhraní z ochrany autorských práv vyloučeny. Stejně tak tomu je v oddílu 102 (b) autorského zákona. V úvodu kapitoly 5 jsme si ukázali, že přístup k těmto myšlenkám v případě strojového kódu je zásadně omezen. Ačkoli jsou tedy tyto myšlenky nechráněny není možné se k nim bez dalšího dostat a volně je používat. Proto je nutné zpětné inženýrství. Vyvažování zájmů mezi autory a společnostmi si bylo vědomo již CONTU, dle kterého sice měla autorská práva zakazovat neoprávněné kopírování těchto děl, nic však nemělo bránit jejich oprávněnému použití. Neměl být blokován vývoj a šíření těchto děl a nikdo neměl díky autorskému právu nabýt více ekonomické síly než tolik, aby byl dostatečně motivován k tvorbě.<sup>418</sup> Daughtrey a Cuillou se domnívají, že zákonodárce vzhledem k uzákonění *idea-expression dichotomy* v oddíle 102 autorského zákona<sup>419</sup> zamýšlel poskytovat ochranu myšlenkám pouze skrze patent. Podle Daughtrey a Cuillou: „*Omezení zpětného inženýrství počítačových programů vytvářejí virtuální dlouhodobý patent podle Copyright Act, ale bez přísných požadavků kladených na patenty.*“<sup>420</sup> Ve Spojených státech neexistuje žádné povolení zpětného inženýrství za účelem interoperability podobné tomu,

---

<sup>417</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 83.

<sup>418</sup> Karjala, D. S. *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller Symposium: Copyright Protection and Reverse Engineering of Software*, s. 999-1000.

<sup>419</sup> Spojené státy americké. Copyright act of 1976.

<sup>420</sup> Daughtrey. *Reverse Engineering of Software for Interoperability and Analysis Note*, s. 178 anebo Cuillou. *The Reverse Engineering of Computer Software in Europe and the United States*, s. 178. Vlastní překlad.

jež je obsaženo v právním řádu Evropské unie.<sup>421</sup> Místo toho lze zpětné inženýrství provádět na základě obsáhlejší doktríny *fair-use*.<sup>422</sup> Důležitým ustanovením, které toto umožňuje je oddíl 107 autorského zákona. Díky němu lze provádět u počítačového programu zpětné inženýrství, pouze však pokud jde o poctivé použití a pouze za stanovenými účely. Vzhledem k tomu, že myšlenky, na kterých počítačový program stojí jsou často skryté, je toto jeden ze způsobů, který soudy shledaly jako přiměřený k jejich získání. Musí však sloužit primárně k získání nechráněných částí díla, tedy myšlenek a tyto nesmí být přístupné jinak. Jedná se o metodu použitelnou pouze tam, kde je soudem kopírované dílo shledáno jako chráněné. U děl autorským právem nechráněných postrádá tento test smysl. Soudy při hodnocení *fair use* musí hodnotit minimálně tyto čtyři záležitosti.

*Účel a charakter použití*, kde je soudy preferováno použití za účelem tvorby nových autorských děl (transformativní použití), a tedy takové, které přispívá společnosti.<sup>423</sup> Nemělo by se tedy jednat o pouhé kopírování díla, bez dodání nějakého nového individualizovaného prvku. Stejně tak je důležité, zda je dílo kopírováno za účelem zisku nebo například za účelem vzdělání.<sup>424</sup> Hospodářský účel však nutně neznamená porušení autorského práva. McManis rozlišuje vytvoření konkurenčního programu, který okopíroval chráněné vyjádření a vytvoření konkurenčního programu se stejnými nechráněnými funkcemi a myšlenkami. Dle něj není důvodné omezovat vytvoření konkurenčního díla, pokud nedojde k prostému okopírování.<sup>425</sup> Rovněž je důležité, zda není kopírované vyjádření natolik standardizované nebo předurčené svou podstatou, že jej

---

<sup>421</sup> Směrnice o právní ochraně počítačových programů.

<sup>422</sup> Mylly. *Intellectual property protection of computer program interfaces and interoperability*, s. 303.

<sup>423</sup> Vasilescu-Palermo. *APIs and Copyright Protection: The Potential Impact on Software Compatibility in the Programming Industry*, s. 158.

<sup>424</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 57.

<sup>425</sup> Tamtéž.



nelze vyměnit, respektive vyjádřit jiným způsobem.<sup>426</sup> Těmito body se zabývaly soudy například v případech *Atari* a *Sega*.<sup>427</sup>

Dále soudy řeší *podstatu kopírovaného díla* a také *část (poměr) použitého díla*. Pro *podstatu kopírovaného díla* je důležité, o jaké dílo se jedná. Je přijatelnější kopírování díla postaveného na faktech, které jako takové nepodléhají ochraně, případně podléhají ochraně slabší. Nejvyšší soud rozhodl ve věci *Baker v. Selden*<sup>428</sup> tak, že díla se silnými funkčními prvky, jako jsou účetní knihy, nemají nárok na stejný stupeň ochrany jako faktická díla, respektive nemusí mít nárok na ochranu vůbec.<sup>429</sup> Naopak kopírování díla čistě uměleckého bude pravděpodobně shledáno jako porušení autorského díla. Co se *poměru použitého díla* týče, je vzhledem k podstatě rozhraní, pro pochopení myšlenky, nutné okopírovat jej celé. To by však nemělo být možností zpětného inženýrství na újmu. Kopírování celého díla bylo shledáno přípustné i v jiných případech.<sup>430</sup> Důležitý tak bude spíše poměr okopírované části ku zbytku nově vytvořeného díla. Mimo tento kvantitativní požadavek je třeba také zkoumat jak důležitá a podstatná část je kopírována.

Posledním prvkem je *efekt použití díla na hodnotu kopírovaného díla pro trh*.<sup>431</sup> Zde je hodnocen dopad na možnosti autora profitovat ze svého díla. V podstatě tedy spíše dojde k porušení tehdy, kdy nový program nahradí okopírovaný program na daném trhu.

K tomuto testu lze doplnit rovněž bod zahrnující veřejné zájmy. Ten však v doktríně fair use není výslovně uveden. Podle Daughtreyho se však některé soudy domnívají, že určité veřejné zájmy, jako je přístup k informacím, převažují nad autorským právem k dílu.<sup>432</sup>

<sup>426</sup> Spoor. *Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive Symposium: Copyright Protection and Reverse Engineering of Software*, č. 3.

<sup>427</sup> *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992); Srov. *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (United States Court of Appeals for the Federal Circuit, 1992).

<sup>428</sup> SCOTUS. *Baker v. Selden*, 101 U.S. 99 (1879), 19.

<sup>429</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 69.

<sup>430</sup> SCOTUS. *Sony Corp. v. Universal City Studios*, 464 U.S. 417 (1984), 30.

<sup>431</sup> Spojené státy americké. § 107 of Copyright act of 1976.

<sup>432</sup> Daughtrey. *Reverse Engineering of Software for Interoperability and Analysis Note*, s. 164.

O výklad čtyřstupňového testu se pokoušel například *Committee on Computer Law*<sup>433</sup> nebo *Last Frontier Conference on Copyright Protection of Computer Software*<sup>434</sup>. Lze říci, že konference stanovila možnost studia myšlenek počítačového programu, včetně k tomu nezbytných kopií programu, a to i přesto, že výsledné dílo má být použito komerčně.<sup>435</sup> Prioritou je totiž celkový společenský pokrok a vývoj technologií.<sup>436</sup> Doktrína fair use byla soudy uplatněna také ve věci *Nintendo v. Atari*.<sup>437</sup>

Společnost Nintendo prodávala konzole, které byly šifrované. Společnost Atari chtěla prodávat své videohry pro konzoli od Nintendo a snažila se ji dešifrovat. Atari však nebylo úspěšné a zažádalo Úřad pro autorská práva o zdrojový kód, který mu úřad nesprávně poskytl.<sup>438</sup> Atari však ze zdrojového kódu okopírovala víc, než bylo nutné k interoperabilitě her s konzolí.<sup>439</sup> Soud rozhodl, že Atari měla možnost získat kód legálním způsobem a měla této možnosti využít. Avšak pokud je k pochopení myšlenek a principů, na kterých je program založený, nutný přístup ke zdrojovému kódu, je za těchto podmínek možné provést zpětné inženýrství.<sup>440</sup> Atari se však nedostala ke kódu legální cestou a kód použila nad rozsah nutný k zajištění interoperability. Na základě rozhodnutí bylo společnosti Atari zamezeno ve vývoji a distribuci okopírovaného programu.

Další případ se týkal rovněž videoher. Společnost *Sega* vyvinula konzoli a společnost *Accolade* měla zájem prodávat hry spustitelné na této konzoli.<sup>441</sup> *Accolade* si pomocí zpětného inženýrství získala sama nezbytný kód zajišťující interoperabilitu. K tomu však musela vytvořit několik kopií programu společnosti *Sega*. Z kopií použila pouze nezbytnou

---

<sup>433</sup> Committee on Computer Law, Reverse Engineering and Intellectual Property Law, 44 REC. ASS'N. B. CRRY OF N.Y., 132, 134-35 (1989).

<sup>434</sup> Last Frontier Conference Report on Copyright Protection of Computer Software, Conference Report, 30 Jurimetrics J. 15 (1989).

<sup>435</sup> Tamtéž, s. 25.

<sup>436</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 45.

<sup>437</sup> *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (United States Court of Appeals for the Federal Circuit. 1992).

<sup>438</sup> Tamtéž, 835.

<sup>439</sup> Tamtéž, 843.

<sup>440</sup> Tamtéž, 842-843.

<sup>441</sup> *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992).

část, která zajistila interoperabilitu.<sup>442</sup> Soud rozhodl, že pořizování kopií, které jsou nezbytné pro zjištění myšlenek a principů je tzv. fair-use a funkční požadavky na kompatibilitu s herní konzolí Genesis nejsou chráněny autorským právem.<sup>443</sup> Pro posouzení případu použil soud stejný test jako soud v případě *Computer Associates v. Altai*, který se týkal interoperability software a hardware. Podle soudu nelze přistoupit na to, že funkční specifikace počítačového programu jsou chráněné.<sup>444</sup> Kód nebo postup zajišťující interoperabilitu není chráněn autorským zákonem. Ochrana kódu v takovém rozsahu a nemožnost zpětného inženýrství by mohla vést k monopolu k myšlenkám, principům a funkcím, což je přímo proti principu autorského práva.<sup>445</sup> Soud dospěl k závěru, že autorský zákon měl podporovat růst tvůrčího projevu, založený na šíření dalších tvůrčích děl a nechráněných myšlenek obsažených v těchto dílech.<sup>446</sup>

V dalším sporu týkajícím se videoher vytvořila společnost *Atari* interoperabilní herní kazety. Interoperability dosáhla pomocí zpětného inženýrství počítačového programu. Soud rozhodl, že se jednalo o spravedlivé použití autorského díla, neboť *Sega* úmyslně blokovala ostatní soutěžitele před možností spouštět v jejím programu jiné herní kazety.<sup>447</sup>

V některých sporech však soud spravedlivé použití zamítl jako nedůvodné. Jedním z takových sporů je například *Synops v. ATopTech*. Společnost ATopTech zkopírovala příkazové sady (kombinace jména a syntaxe zahrnující vstupní formáty PrimeTime a GoldTime, včetně formátů výstupních) z programu Synopsu. Soud toto jednání shledal jako porušení autorského práva a zároveň nenašel důvody pro spravedlivé použití příkazových sad.<sup>448</sup>

---

<sup>442</sup> *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992), 1514-1515.

<sup>443</sup> Vasilescu-Palermo. *APIs and Copyright Protection: The Potential Impact on Software Compatibility in the Programming Industry*, s. 59. *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992), s. 1526-1527.

<sup>444</sup> *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), 713.

<sup>445</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 45.

<sup>446</sup> *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992).

<sup>447</sup> *Atari Corp. v. Sega of America, Inc.*, 869 F. Supp. 783 (N.D. Cal. 1994).

<sup>448</sup> *Synopsys, Inc. v. ATopTech, Inc.*, 13-cv-02965-MMC (N.D. Cal. Oct. 24, 2016).

Z autorského zákona Spojených států pak lze zmínit ještě oddíl 117<sup>449</sup>, který umožňuje vlastníkovvi kopie počítačového programu pořízení další kopie nebo úpravy počítačového programu.<sup>450</sup> Umožňuje tedy například převod programu z jazyka vyšší úrovně do jiné úrovně, aby se usnadnilo používání a možnost přidat nové funkce, které program v době získání neobsahoval.<sup>451</sup>

### 5.3.1 Google v. Oracle

Aktuální soudní spor, který se týká rozhraní vede společnost *Sun*, respektive *Oracle* proti společnosti *Google*.<sup>452</sup> Spor probíhá ve Spojených státech amerických. Ačkoli se spolu dnes tyto společnosti soudí, v počátcích vývoje sporné platformy Android byla společnost *Sun* za krok společnosti *Google* vděčná.<sup>453</sup> Platforma Java upadala, oproti tomu trh s mobilními platformami nabíral na síle. Společnost *Sun* se však mylně domnívala, že platformu Android vyvinou se společností *Google* společně.<sup>454</sup> *Google* však nezajímala spolupráce, jeho zájmem bylo pouze použití Java API.<sup>455</sup>

Společnost *Sun* se pokoušela vytvořit z Javy standard používaný v rámci celého trhu. Tomu mělo dopomoci vytvoření balíčků Java API. Balíčky Java API jsou vyjádřeny v předem napsaných třídách provádějících konkrétní metody.<sup>456</sup> Každý balíček Java API je v podstatě soubor specifikací potřebných k vyvolání funkcí. Tyto balíčky umožnily vývojářům využít předem napsaný kód k zabudování určitých funkcí do jejich

---

<sup>449</sup> Spojené státy americké. § 117 of Copyright act of 1976.

<sup>450</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 78.

<sup>451</sup> National Commission on New Technological Uses of Copyrighted Works (CONTU, Final Report on the National Commission on New Technological Uses of Copyrighted Works, 3 Computer L.J. 53 (1981)), s. 61-62.

<sup>452</sup> SCOTUS. *Google LLC v. Oracle America, Inc.*

<sup>453</sup> Menell, P. S. *API Copyrightability Bleak House: Unraveling and Repairing the Oracle v. Google Jurisdictional Mess*. Berkeley Technology Law Journal. 2016, č. 3, s. 344.

<sup>454</sup> Vasilescu-Palermo. *APIs and Copyright Protection: The Potential Impact on Software Compatibility in the Programming Industry*, s. 155.

<sup>455</sup> Menell, P. S. *API Copyrightability Bleak House: Unraveling and Repairing the Oracle v. Google Jurisdictional Mess*. Berkeley Technology Law Journal. 2016, č. 3, s. 364.

<sup>456</sup> Vezzoso. *Copyright, Interfaces, and a Possible Atlantic Divide*, s. 155.

vlastních programů.<sup>457</sup> Balíčky tedy obsahují třídy a tyto třídy se dále člení na metody.<sup>458</sup> Každý balíček je tvořen dvěma typy kódů, kódem implementačním a kódem deklaračním (specifikací rozhraní). Deklarační kód popisuje funkci balíčku.<sup>459</sup> Balíčky Java API se provádí prostřednictvím implementačního kódu vycházejícího ze specifikace rozhraní. Podstatným přínosem balíčků Java API je snazší, rychlejší, efektivnější vývoj programů.<sup>460</sup> Java API bylo společností Sun vytvořené a přístupné z toho důvodu, aby společnosti a programátoři měli jednodušší práci s psáním interoperabilních programů. Programátoři si balíčky API mohou převzít, což jim zajistí základní propojení s platformou Java. Dokument Java Standard Edition (SE), který stanoví rozhraní Java API, je mimo jiné k dispozici ke stažení zdarma na webových stránkách Oracle.<sup>461</sup>

Google v roce 2005 koupil Android a zamýšlel do něj implementovat část Javy. K balíčků Java API však neměl licenci a společnosti se nedohodli na spolupráci. Google tak začal na vývoji pracovat sám pomocí programovacího jazyka Java a nelicencovaných balíčků Java API, respektive převzatých hlaviček, které pojmenovávají a popisují funkce.<sup>462</sup> Specifikace Java API určující, jakým způsobem jsou funkce či metody volány nebo použity musí být totiž pro správnou funkci zůstat nezměněná. Společnost Google použila při vývoji platformy Android programovací jazyk Java a 168 balíčků API, z nichž mnohé se stejnými funkcemi jako Java API.<sup>463</sup> Implementaci API si vytvořil Google vlastní na základě napodobení implementace Javy. Android tak na jednu stranu byl interoperabilní s Javou, na druhou stranu nedošlo k doslovnému kopírování. Kopírování se Google dopustil pouze u názvů a metod, původní organizace, struktury a posloupnosti 37 balíčků Java API, které však pro jejich správnou funkci

---

<sup>457</sup> *Oracle America, Inc v. Google Inc*, 750 F.3d 1339 (Fed. Cir. 2014), 1349.

<sup>458</sup> Tamtéž.

<sup>459</sup> Tamtéž.

<sup>460</sup> Menell. *Rise of the API Copyright Dead: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, s. 305–490.

<sup>461</sup> Viz dokumentace Oracle. Java® Platform. Standard Edition Java Development Kit Version 9 API Specification. <https://docs.oracle.com/javase>

<sup>462</sup> Vasilescu-Palermo. *APIs and Copyright Protection: The Potential Impact on Software Compatibility in the Programming Industry*, s. 167.

<sup>463</sup> *Oracle America, Inc v. Google Inc*, 750 F.3d 1339 (Fed. Cir. 2014); Srov. Vezzoso. *Copyright, Interfaces, and a Possible Atlantic Divide*, s. 155.

nešlo napsat jiným způsobem.<sup>464</sup> Google však i u těchto 37 balíčků použil jinou implementaci.<sup>465</sup> Totožnou implementaci použil pouze u funkce *rangeCheck* a osmi dekompilovaných souborů zabezpečení.<sup>466</sup> Ze sporných bodů je nejdůležitější specifikace rozhraní, okopírované soubory a 37 balíčků API kopírujících původní organizaci, strukturu a posloupnost. Naopak myšlenky, na kterých stojí rozhraní, programovací jazyk Java anebo názvy API a ostatní jednoduché popisné vyjádření předmětem sporu nebyly.<sup>467</sup> Google již od začátku sporu prokazuje, že musel použít konkrétní API, protože jinak by Android nemohl být kompatibilní s Javou.<sup>468</sup> API musí být napsáno určitým způsobem, aby příkaz fungoval.<sup>469</sup> Společnost Google také tvrdí, že použití balíčků API bylo transformativní. Podle společnosti Sun, okopíroval Google spolu se specifikací API také jejich strukturu posloupnost a organizaci.<sup>470</sup>

Výsledkem prvoinstančního rozhodnutí byl závěr, že specifikace rozhraní Java API nelze chránit autorskými právy. Rozhodnutí bylo odůvodněno tím, že „kopírování“ 37 balíčků API se týkalo funkčních, a tedy nechráněných prvků.<sup>471</sup> K dosažení žádané funkce tedy nešlo použít jiný kód. Dalším argumentem soudu byla skutečnost, že Google napsal svůj vlastní implementační kód.<sup>472</sup> Hlavičky Javy totiž dle soudu lze podřadit pod specifikace, které umožňují vyvolání funkcí programu. Funkčnost těchto hlaviček je neoddělitelně propojena s obsaženým vyjádřením podle *merger doctrine*.<sup>473</sup> Okresní soud rozhodl, že pokud je kopírována pouze specifikace API, ale implementace je provedena vlastní, není

---

<sup>464</sup> *Oracle America, Inc v. Google Inc*, 750 F.3d 1339 Fed. Cir. (2014), 1349-1350; Srov. Gratz, J. Lemley, M. *Platforms and Interoperability in Oracle v. Google*. Harvard Journal of Law and Technology, 2018, roč. 31, special issue Spring. s. 609.

<sup>465</sup> Tamtéž.

<sup>466</sup> *Oracle America, Inc v. Google Inc*, 750 F.3d 1339 (Fed. Cir. 2014), 1351.

<sup>467</sup> *Oracle America, Inc v. Google Inc*, 872 F.Supp.2d 974 (N.D.Cal. 2012), 997.

<sup>468</sup> Tamtéž, s. 998.

<sup>469</sup> Tamtéž.

<sup>470</sup> *Oracle America, Inc v. Google Inc*, 750 F.3d 1339 (Fed. Cir. 2014), 1356.

<sup>471</sup> *Oracle America, Inc v. Google Inc*, 872 F.Supp.2d 974 (N.D.Cal. 2012), 976.

<sup>472</sup> Tamtéž.

<sup>473</sup> *Oracle America, Inc v. Google Inc*, 872 F.Supp.2d 974 (N.D.Cal. 2012), 997; Srov. Samuelson. *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*, s. 1278

důvod omezovat vývojáře v možnosti volat funkce Java.<sup>474</sup> Rozdíl mezi specifikací rozhraní a jeho implementací je v oblasti výpočetní techniky podle soudu dlouhodobý.<sup>475</sup> Toto dělení je rovněž viditelné např. v případě *Connectix v. Sony*.<sup>476</sup> Connectix vytvořil PlayStation emulator, čímž v podstatě umožnil uživatelům spouštět hry vytvořené pro PlayStation na stolních počítačích. Podle soudu se jednalo transformační použití. Stejně tak šlo podle soudu o zcela nový produkt, neboť Connectix vytvořil vlastní kód pro implementaci funkcí firmwaru PlayStation.<sup>477</sup>

S takovým rozhodnutím se však neztotožnil odvolací soud a věc vrátil k dalšímu projednání. Odvolací soud se zabýval *merger doctrine*. Podle ní nelze chránit dílo, pokud lze myšlenku v něm obsaženou vyjádřit pouze jedním způsobem.<sup>478</sup> Podle soudu však v tomto případě alternativní způsoby vyjádření existovaly.<sup>479</sup> Dle odvolacího soudu měla být doktrína použita na odfiltrování jednotlivých balíčků. Ty měly být zkoumány individuálně.<sup>480</sup> Doktrína *scène a fair* pak podle soudu neovlivňuje způsoblost deklaračního kódu, ani struktury a posloupnosti sporných balíčků Java API podléhat autorskoprávní ochraně.<sup>481</sup>

Soud zároveň zdůraznil, že se prvoinstanční soud nedostatečně vypořádal s rozdělením ochrany deklaračního kódu a ochrany struktury, organizace a posloupnosti balíčků API.<sup>482</sup> Výsledkem bylo stanovisko, dle kterého je struktura organizace a posloupnost originální a deklarační kód mohl být napsán a uspořádán mnoha způsoby a stále mohl dosahovat stejných funkcí. Proto podle soudu nelze vyjmout balíčky API z autorskoprávní ochrany.<sup>483</sup> Soud nepřesvědčil ani argument Google, že se z balíčků API stal standard.<sup>484</sup> V dalším řízení tak neměla být

<sup>474</sup> *Oracle America, Inc v. Google Inc*, 872 F.Supp.2d 974 (N.D.Cal. 2012), 997.

<sup>475</sup> Samuelson, P., Asay, C. D. *Saving Software Fair Use Future*, Volume 31, Special Issue Spring 2018. roč. 31. str. 545-546.

<sup>476</sup> *Sony Computer Entertainment v. Connectix Corporation*, 203 F.3d 596 (United States Court of Appeals, Ninth Circuit, 2000).

<sup>477</sup> Samuelson. Asay. *Saving Software Fair Use Future*, s. 553-554.

<sup>478</sup> *Oracle America, Inc v. Google Inc*, 750 F.3d 1339 (Fed. Cir. 2014), 1360.

<sup>479</sup> Tamtéž, 1361.

<sup>480</sup> Tamtéž, 1362.

<sup>481</sup> Tamtéž, 1364.

<sup>482</sup> Tamtéž.

<sup>483</sup> Tamtéž, 1368.

<sup>484</sup> Tamtéž, 1371-1372.

projednávána možnost ochrany autorským právem, nýbrž možnosti fair use u použití API.<sup>485</sup> Prvoinstanční soud v následném řízení shledal použití balíčků Java API v souladu s principem fair use.<sup>486</sup>

Proti tomuto rozhodnutí se společnost Oracle odvolala. Odvolací soud se postavil na stranu společnosti Oracle a zamítl stanovisko okresního soudu týkající se fair use.<sup>487</sup> Podle Odvolacího soudu Google při použití Java API nesplnil žádné ze čtyř kritérií fair use. První ze zkoumaných hledisek byl účel a charakter použití. Soud rozhodl, že Google použil balíčky API za komerčními účely a jde tak proti prvnímu faktoru fair use.<sup>488</sup> Stejně tak se soud neztotožnil s argumentem Google, že použití bylo transformativní, a to proto, že kopírování bylo doslovné, sloužilo pro stejnou funkci a účel a pouhá změna platformy ze stolních počítačů na mobilní telefony není dostatečná.<sup>489</sup> Podle soudu rovněž neobstojí argument, že se jednalo o malou část Javy, neboť z kvalitativního pohledu je tato část významná.<sup>490</sup> Soud souhlasil s tím, že Google jednal v konkurenčním boji se snahou zamezit společnosti Sun v rozšíření svých obchodních aktivit a snahou přiživit se na rozmachu Java. Takové chování má nepříznivý dopad na potencionální trh původního díla.<sup>491</sup> Odvolací soud dospěl k závěru, že dva ze čtyř faktorů fair use nebyly splněny, jeden ano a jeden zůstal neutrální.<sup>492</sup> Okresní soud na základě názoru soudu odvolacího měl rozhodnout o výši náhrady škody pro společnost Oracle.<sup>493</sup>

S takovým rozhodnutím se však Google nemohl spokojit a podal návrh k Nejvyššímu soudu USA. Na stranu jednotlivých účastníků sporu se přidávaly postupně různé společnosti. Společnost Google podpořil např.

---

<sup>485</sup> *Oracle America, Inc v. Google Inc*, 750 F.3d 1339 (Fed. Cir. 2014). 1381.

<sup>486</sup> *Oracle America, Inc v. Google Inc*, 3:10-cv-03561 (2016).

<sup>487</sup> *Oracle America, Inc v. Google Inc*, 886 F.3d 1179 (Fed. Cir. 2018).

<sup>488</sup> Tamtéž, 1198.

<sup>489</sup> Tamtéž, 1202.

<sup>490</sup> Tamtéž, 1207.

<sup>491</sup> Tamtéž, 1210.

<sup>492</sup> Tamtéž.

<sup>493</sup> Tamtéž.



Microsoft, IBM a jiní, obávající se dopadu rozhodnutí na informační technologie. Na stranu Oracle se postavila vláda USA.<sup>494</sup>

Nejvyšší soud tak stojí před závažným rozhodnutím, bude se muset vypořádat s tím, zda je API funkční nebo kreativní část počítačového programu.<sup>495</sup> A do jaké míry lze poskytovat ochranu funkčním specifikacím API tehdy, existuje-li více způsobů dosažení stejné funkce. Bylo-li by ponecháno stanovisko odvolacího soudu, mohlo by to značně rozšířit ochranu počítačového programu původně zamýšlenou zákonodárcem.<sup>496</sup> Bude se rovněž muset vypořádat s názorem odvolacího soudu, dle kterého nejsou chráněny myšlenky obecné, avšak myšlenky konkrétní, nižších kategorií ano. Pokud tedy lze dosáhnout konkrétní funkce vícero způsoby je nutné využít tyto jiné funkce a neomezit se na pouhé kopírování. Tento pohled však lze použít v klasickém prostředí autorského práva, nikoli v případě specifikace rozhraní, která z určitých důvodů musí být totožná. Kreativitu nezbytnou pro vytvoření literárního nebo jiného uměleckého díla nelze srovnávat s kreativitou při vývoji API. Soud bude muset správně zhodnotit, která část počítačového programu spadá pod myšlenku, a která pod vyjádření. Je evidentní, že soud nebude vycházet pouze ze zákona a dřívější judikatury soudů. Bude muset zohlednit konkrétní situaci a zejména dopady svého rozhodnutí na trh s informačními technologiemi.

Pokud soud shledá API chráněné autorským právem bude se muset vypořádat s doktrínou fair use. Bude muset projít zmíněné čtyři kroky na základě kterých určí, jestli je nebo není okopírovaná část Java API v souladu s fair use. Z dosavadního průběhu se jeví sporným už krok první, a to transformativní použití.

Rozhodnutí soudu ve prospěch společnosti Oracle by vedlo k omezení vývoje programů se stejnými funkcemi. Soud by v podstatě určil porušení autorského práva v případech, kdy rozdílní autoři pomocí stejných specifikací implementují kód odlišně. To povede k nemožnosti zajistit interoperabilitu a ke konci network efektu, jak je představen výše.

---

<sup>494</sup> SCOTUS. *Oracle America, Inc v. Google Inc.* Docket for 18-956 [online].

[cit. 25. 3. 2021]. <https://www.supremecourt.gov/docket/docketfiles/html/public/18-956.html>

<sup>495</sup> Vasilescu-Palermo. *APIs and Copyright Protection: The Potential Impact on Software Compatibility in the Programming Industry*, s. 62.

<sup>496</sup> Menell. *Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, s. 427.

Naopak se může stát, že verdikt ve prospěch společnosti Google motivuje větší a silnější společnosti k „vykrádání“ produktů společností menších. Zároveň zajisté povede k nárůstu sporů, neboť bude existovat šance, že soud neshledá použití API jako fair use.

### 5.4 Shrnutí

Specifikace rozhraní je nechráněným prvkem počítačového programu. Pokud však není zpřístupněna autorem počítačového programu, musí se osoba, která informaci o interoperabilitě požaduje, spokojit se zpětným inženýrstvím. To je dovoleno jak na unijní<sup>497</sup>, tak na vnitrostátní úrovni<sup>498</sup>. Metoda zpětného inženýrství tzv. blackbox testing je umožněna článkem 5 směrnice.<sup>499</sup> Tento článek umožňuje pozorování počítačového programu za účelem zjištění myšlenek a zásad na kterých stojí. Pro interoperabilitu je zásadní také článek 6 směrnice a § 66 AZ. Obě tato ustanovení umožňující zásah do díla na základě oprávněných zájmů. Povolená dekompilace se vztahuje na případy, ve kterých je nezbytná pro získání informací o interoperabilitě nezávisle vytvořeného počítačového programu. Musí však zároveň být splněny další podmínky. Jednou z těchto podmínek je skutečnost, že dekompilaci musí provádět osoba která má k dílu oprávnění. Dále nesmí být informace nezbytné pro interoperabilitu rozkladateli přístupné jinak rychleji a snadněji a musí být zasaženo jen do té části díla, která je pro dosažení těchto informací nezbytná. Je vhodné dodat, že zpětnému inženýrství nelze zamezit pomocí patentu s unijním účinkem ani ochranou obchodního tajemství. Čl. 6 směrnice a § 66 AZ rovněž omezují způsoby nakládání se získanými informacemi. Informace o interoperabilitě nesmí být předány třetí straně a zejména nesmí být použity pro vývoj, výrobu ani odbyt počítačového programu, jehož vyjádření bude podstatně podobné původnímu dílu. Ani jeden z uvedených právních předpisů se však nezabývá technologií zpětného inženýrství. Zpětné inženýrství nemusí vést vzhledem ke své náročnosti k zajištění interoperability. Vzhledem k tomu, že specifikace rozhraní není chráněna, není zpětné inženýrství dokonalým instrumentem.

---

<sup>497</sup> Čl. 65 a 65 Směrnice o právní ochraně počítačových programů.

<sup>498</sup> § 66 AZ.

<sup>499</sup> Čl. 5 Směrnice o právní ochraně počítačových programů.

## 6 Návrhy de lege ferenda

Poslední část práce se bude zabývat tím, zdali je aktuální právní úprava, zvolená forma a rozsah ochrany rozhraní, možnost zpětného inženýrství a zejména míra interoperability dostatečná. Vzhledem k tomu, že v minulosti již byla interoperabilita zajišťována rovněž pomocí práv na ochranu hospodářské soutěže<sup>500</sup>, bude níže uvedena v souvislosti s aktuální právní úpravou. Autorské právo by však jakožto zvolené právní odvětví, mělo být dostačující, neboť obsahuje vhodné instrumenty k dosažení interoperability a jako takové má vyřešenou otázku ochrany rozhraní. Rozsah ochrany rozhraní je určován dělením rozhraní na myšlenky a vyjádření. Vzhledem k tomu, že specifikace rozhraní spadá pod myšlenku a není chráněna, je autorské právo pro zajištění interoperability vhodné. Otázkou zůstává, zda je dostačující možnost provádět zpětné inženýrství nebo by bylo vhodné informace o interoperabilitě zpřístupnit zájemci jinak. Například pomocí práva na ochranu hospodářské soutěže. To by se však mělo uplatnit pouze v případech, kdy interoperability nemůže být dosaženo pouze pomocí autorského práva. V každém případě by bylo vhodné úpravu zpětného inženýrství modernizovat tak, aby nebylo nadále nutné využívat ex post doplňující instrumenty (např. ochrana hospodářské soutěže, viz případ Microsoft<sup>501</sup>).<sup>502</sup> Není však žádoucí přijetí nového zákona nebo jeho rozsáhle reformace. To by mohlo způsobit nejistotu a destabilizaci současného stavu.<sup>503</sup> Z těchto důvodů bude navrženo několik způsobů, kterými by mohla být autorskoprávní úprava doplněna, upravena nebo nahrazena. Navrhujeme několik možných alternativ, pomocí kterých lze dosáhnout větší míry interoperability.

---

<sup>500</sup> Judgment of the Court of First Instance of 17 September 2007. T-201/04. *Microsoft Corp. v Commission of the European Communities*.

<sup>501</sup> Tamtéž.

<sup>502</sup> Graef. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* s. 18.

<sup>503</sup> Daughtrey *Reverse Engineering of Software for Interoperability and Analysis Note*, s. 145–188.

## 6.1 Doplnující ochrana

Zajištění interoperability může být dnes umožněno pomocí dvou odvětví právního řádu. Ačkoli cílem obou z nich je zajištění interoperability a správné fungování trhu, jsou metody k dosažení takového cíle odlišné. Prvním odvětvím, které má vliv na interoperabilitu je právo duševního vlastnictví, konkrétně autorské právo. Autorské právo vyjímá z ochrany specifikaci rozhraní a umožňuje zpětné inženýrství. Druhým odvětvím je právo hospodářské soutěže, které interoperabilitu vynucuje až v případě porušení práva hospodářské soutěže, například zneužitím dominantního postavení nebo v monopolním postavení.<sup>504</sup> Jednotlivé případy jsou tak hodnoceny individuálně a flexibilně, což může vést ke spravedlivějšímu rozhodování případů s posouzením všech okolností a jejich vlivu na trh a výběr adekvátního řešení nastalé situace. Mohou tak být lépe vyvažována pozitiva a negativa konkrétního případu. Na druhou stranu nemusí být zajištěna dostatečná míra právní jistoty, což může vést k nižší motivaci inovovat.<sup>505</sup>

Najít rovnováhu mezi těmito právy není jednoduché. Ochranu počítačového programu a jeho rozhraní je z obecného pohledu vhodnější upravit od počátku, tedy autorským právem. Tak aby se adresáti těchto norem mohli přizpůsobit. V některých případech, kde tato obecná úprava není dostačující je vhodné pomoci autorskému právu právem na ochranu hospodářské soutěže.<sup>506</sup> Tento stav by však neměl být shledán dostačujícím. To i proto, že ochrana hospodářské soutěže dnes není poskytována příliš flexibilně, případ od případu (*ex post*). Čím dál častěji se vytváří konkrétní doktríny a návody vydávané Evropskou komisí, které se snaží vyřešit hospodářskou soutěž obecně, bez nutnosti zkoumat a balancovat každý jednotlivý případ (*ex ante*).<sup>507</sup> Základním východiskem by proto měla být zcela soběstačná a dostačující úprava autorským právem. Právo na ochranu hospodářské soutěže by se uplatnilo pouze ve výjimečných případech.

---

<sup>504</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 91.

<sup>505</sup> Tamtéž, s. 99-100.

<sup>506</sup> Tamtéž, s. 105.

<sup>507</sup> Tamtéž, s. 120, 142.

### 6.1.1 Hospodářská soutěž

Směrnice o právní ochraně počítačových programů ve své preambuli počítá s uplatňováním pravidel hospodářské soutěže. V bodu 17 stanoví, že: „Ustanoveními této směrnice není dotčeno uplatňování soutěžních předpisů podle článků 81 a 82 Smlouvy, pokud dodavatel mající dominantní postavení odmítne zpřístupnit informace nezbytné k dosažení interoperability ve smyslu této směrnice.“ Články 101 (dříve 81) a 102 (dříve 82) Smlouvy o fungování Evropské unie zakazují jednání podniků zneužívající dominantní postavení a dohody vylučující, omezující nebo narušující hospodářskou soutěž. Čl. 101 SFEU omezuje společnosti ve spojování, které by mohlo vést k omezování soutěže. Jedná se na například o vznik kartelů.<sup>508</sup> Čl. 102 SFEU míří na zneužití dominantního postavení jednou společností. To, zda společnost zneužívá své postavení je nutné zkoumat individuálně a soudy zde mají velký prostor k úvaze.<sup>509</sup> Současně ale článek 102 demonstrativně vypisuje některé příklady zneužití dominantního postavení a poskytuje tak alespoň částečnou právní jistotu pro společnosti v dominantním postavení.<sup>510</sup> Pro rozhraní je významné zejména zneužívání postavení v omezování výroby, odbytu nebo technického vývoje na úkor spotřebitelů. Půjde-li o rozhraní, respektive o přístup k informacím o daném rozhraní, který bude nezbytný pro fungování trhu, může nastat situace, kdy bude společnost povinná informace o rozhraní poskytnout ostatním soutěžitelům. Autor počítačového programu se dostane do rozporu s právem na ochranu hospodářské soutěže zejména tehdy, stane-li se jeho rozhraní tzv. standardem a on neposkytne zájemcům informace nezbytné pro interoperabilitu. Jak jsme již uvedli, směrnice počítá s tím, že se v těchto případech použijí pravidla hospodářské soutěže. Směrnice o právní ochraně počítačových programů se tedy použije v každém případě, bez ohledu na to, zda je autor v dominantním postavení nebo ne. Pokud však v dominantním postavení je, použijí se rovněž ustanovení SFEU a autor musí sám aktivně poskytnout informace o rozhraní ostatním soutěžitelům.<sup>511</sup>

<sup>508</sup> Článek 101 Smlouvy o fungování Evropské unie.

<sup>509</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 115.

<sup>510</sup> Článek 102 Smlouvy o fungování Evropské unie.

<sup>511</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 116.

Ochrana pomoci práva hospodářské soutěže by měla nastat pouze ve velmi výjimečných případech. Veškeré situace, které lze předvídat by měly být řešeny primárně pomocí autorského práva. Pravidla hospodářské soutěže by tak byla použitelná pouze ve zjevném zneužití dominantního postavení a nemožnosti zajistit interoperabilitu pomocí autorského práva.<sup>512</sup> Prvním takovým případem byl *Magill*<sup>513</sup>, který byl potvrzený v rozhodnutí ve věci *IMS Health*. Produkt společnosti IMS se stal standardem na německém trhu. Vzhledem k tomu, že IMS nechtěla svůj formát licencovat, musely konkurenční společnosti používat odlišné, nestandardní formáty.<sup>514</sup> Soud rozhodl, že: „*Odmítnutí podniku v dominantním postavení umožnit přístup k produktům chráněným právem duševního vlastnictví... lze považovat za zneužívající pouze pokud společnost, která o licenci požádala..., hodlá vyrábět nové zboží nebo služby, které vlastník práva nenabízí a po kterých existuje potencionální spotřebitelská poptávka.*“<sup>515</sup> V tomto rozhodnutí se jeví jako zásadní podmínka, podle které musí žadatel mít zájem vytvářet nové zboží, které autor nenabízí a po kterém je poptávka. V opačném případě nemusí jít o zneužití dominantního postavení.

Autorské právo by pro zajištění interoperability mělo být dostačující. Avšak v ojedinělých případech lze použít i právo hospodářské soutěže. Jedná se například o případy, kdy držitel autorských práv má na trhu takové postavení, že například může své rozhraní změnit, čímž omezí přístup i těch soutěžitelů, kteří k rozhraní dříve měli přístup. Dalším takovým případem může být rozhraní natolik komplexní, že je nemožná jeho dekompilace a držitel autorských práv zároveň odmítá dobrovolně informaci o rozhraní zpřístupnit.<sup>516</sup> Takový případ byl například spor s *Microsoftem*. Dle rozhodnutí zneužila společnost Microsoft svého dominantního postavení tím, že odmítla poskytnout konkurentům

---

<sup>512</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 84.

<sup>513</sup> Rozsudek Soudního dvora ze dne 6. dubna 1995., RTE a ITP v. Komise, sp. zn. C-241/91 P a C-242/91 P.

<sup>514</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 124-125.

<sup>515</sup> Rozsudek Soudního dvora ze dne 29. dubna 2004, IMS Health GmbH & Co. OHG proti NDC Health GmbH & Co. KG, sp. zn. C-418/01, bod 49.

<sup>516</sup> Graef. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* s. 7-8.

informace o interoperabilitě.<sup>517</sup> Soud rozhodl o tom, že Microsoft musí zpřístupnit pouze informace o interoperabilitě, nikoli celý kód.<sup>518</sup> A to za účelem rovné soutěže, konkurence, spravedlivé ceny a kvality. Konkurenci pro zajištění interoperability postačuje specifikace rozhraní, nikoli implementace rozhraní. Rozhodnutí soudu tak nepovede k tomu, že společnosti budou kopírovat nebo reprodukovat výrobky Microsoft.<sup>519</sup> V těchto a dalších obdobných případech, které mohou nastat, a které nelze zcela předpokládat není důvodné omezovat použití pravidel hospodářské soutěže. Tato pravidla by se však neměla dostat do pozice alternativního řešení autorských práv k rozhraní.<sup>520</sup>

Právo na ochranu hospodářské soutěže může sloužit jako doplňující úprava ochrany počítačového rozhraní. Jako takové neřeší nikterak ochranu duševního vlastnictví, jeho účelem je zajištění fungování hospodářské soutěže.<sup>521</sup> Oproti autorskému právu poskytují řešení zmíněných problémů až při jejich vzniku. Pokud by tedy autorská ochrana byla dostatečná a obsáhla by veškeré problematické body již od vzniku díla, nebyl by důvod zabývat se jinými formami ochrany. Vzhledem k tomu, že autorské právo se již v několika případech<sup>522</sup> ukázalo jako nedostatečné, neboť jím byly nepřímo chráněny i prvky nespádající pod ochranu autorského práva, je to právě právo hospodářské soutěže, které může přinést rovnováhu mezi autory a trh. Zároveň se může projevit výhoda v individuálním řešení případů. Antimonopolní pravidla hospodářské soutěže tak mohou vyrovnat zneužívání monopolního postavení, do kterého se společnost může dostat na základě autorského práva. Musí však vždy jít o zneužití postavení, pouhé autorské právo k počítačovému programu tedy není dostačující argument pro použití antimonopolních pravidel. Naráží se zde však na jeden problém, a to nemožnost reagovat na situace, kdy je na trhu více společností omezujících interoperabilitu (oligopolní

---

<sup>517</sup> Judgment of the Court of First Instance of 17 September 2007. T-201/04. *Microsoft Corp. v Commission of the European Communities*, bod 1231.

<sup>518</sup> Tamtéž, bod 657.

<sup>519</sup> Tamtéž, bod 657.

<sup>520</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 43-45.

<sup>521</sup> Menell. *Rise of the API Copyright Dead: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software*, s. 465.

<sup>522</sup> Např. Judgment of the Court of First Instance of 17 September 2007. T-201/04. *Microsoft Corp. v Commission of the European Communities*.

trh). Korekce interoperability pomocí práv na ochranu hospodářské soutěže totiž přichází v úvahu pouze a jedině při zneužití dominantního postavení, což se stává primárně v případě monopolního trhu.<sup>523</sup>

Je zřejmé, že cíle obou právních odvětví jsou obdobné a mohou se částečně v jejich dosahování doplňovat. Ochrana právem na ochranu hospodářské soutěže může tak být jedním z korekčních prvků doplňujících autorské právo. Je však evidentní, že nemůže jít o jediné řešení, neboť je značně zdoluhavé a komplexní a lze použít pouze při zneužití dominantního postavení. Zároveň je nutná značná asistence dominantního soutěžitele, který informace musí zpřístupnit.<sup>524</sup>

## 6.2 Explicitní vyjmutí specifikace rozhraní z ochrany

Na základě idea/expression dichotomie se z autorskoprávní ochrany vyjímají části díla, které jsou jeho myšlenkou, nápadem a principem a nejsou tak vyjádřením autorovy individuální tvorby. Specifikaci rozhraní bychom mohli chápat právě jako myšlenku nebo nápad. Specifikace rozhraní sice může být rovněž vyjádřením, protože má konkrétní formu. Jedná se však o natolik funkční prvek počítačového programu, že je možné jej vyjmout z ochrany autorského práva. Jednou z možností je tedy výslovné vyjmutí specifikace rozhraní z autorskoprávní ochrany. Implementace rozhraní by však musela z důvodu zachování autorských práv k dílu zůstat nadále chráněna a nesměla by být kopírována. Část programu by sice byla vyňata z ochrany, nicméně autor by nadále měl nad programem kontrolu.<sup>525</sup> Společnosti by nebyly méně motivovány inovovat a vytvářet autorská díla, neboť by jejich počítačové programy byly nadále chráněny, avšak s výjimkou, která by byla jasně stanovená a pro všechny transparentní. Takovéto vyjmutí z ochrany by však znamenalo změnu směrnice. Nově by musela přesně definovat, jaký prvek počítačového programu chráněn je a jaký není. Problémy může způsobovat to, že není objektivně zcela určitelné, která část počítačového

---

<sup>523</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 84.

<sup>524</sup> Graef. Valcke. *Exploring new ways to ensure interoperability under the Digital Agenda*, s. 3.

<sup>525</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 209-211.



programu je specifikace rozhraní.<sup>526</sup> Zároveň je možné, že vyjmutí rozhraní povede k částečné demotivaci autorů rozhraní vytvářet. Rovněž by mohla nastat situace, kdy společnost využije rozhraní, které je standardem pro získání většího podílu na trhu a poté, co jej získá, rozšíří nebo změní své rozhraní tak, aby vyloučil interoperabilitu konkurentů.<sup>527</sup>

Podle Van Rooijena však pouhé vyjmutí rozhraní z autorskoprávní ochrany situaci zcela neřeší, neboť by nadále nabyvatel produktu musel informace o rozhraní nějakým způsobem získat.<sup>528</sup> Tyto informace by musel být autor povinen zveřejnit. Nebo by si nabyvatel musel tyto informace zajistit sám. Tím by však mohl zasáhnout do části chráněné autorským právem, stejně jako v případě zpětného inženýrství.<sup>529</sup> Ideální se tak jeví explicitní vyjmutí specifikace rozhraní z autorskoprávní ochrany spojené s povinným zveřejněním informací o tomto rozhraní.

### 6.3 Snížení doby ochrany rozhraní

Podle Weston je další možností, která povede k lepší, nikoli dokonalé interoperabilitě, snížení doby ochrany počítačového programu, respektive jeho rozhraní.<sup>530</sup> Lze důvodně předpokládat, že snížení ochrany u počítačového programu jako celku by mělo nepříznivý dopad, neboť autoři by již neměli takový zájem na vývoji nových děl, protože by tato nebyla chráněna po přiměřenou dobu. V případě snížení doby ochrany u rozhraní narážíme na již zmíněný problém s odloučením rozhraní od počítačového programu. Spíše, než tato varianta by byla vhodnější varianta úplného vyjmutí specifikace rozhraní z ochrany.<sup>531</sup> Ačkoli by snížení doby ochrany rozhraní mohlo vyřešit otázku interoperability v rámci

---

<sup>526</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 21.

<sup>527</sup> Sagdeo. *Application programming interfaces and the standardization-value appropriation problém*, s. 257-258.

<sup>528</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 212.

<sup>529</sup> Tamtéž, s. 212-214.

<sup>530</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 95.

<sup>531</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 212.

dlouhodobějšího pohledu, situace v krátkodobém horizontu by zcela vyřešena nebyla. Například pro uživatele je totiž nezbytné mít programy interoperabilní ihned a kontinuálně. Vzhledem k podpoře inovací by však tato varianta mohla být vhodná, neboť soutěžitelé by mohli po nějaké době využít informací o rozhraní a postavit na nich své produkty. Zároveň by však autor měl možnost v rámci zkrácené doby ochrany bojovat o postavení na trhu a vytvoření standardu rozhraní.<sup>532</sup>

## 6.4 Rozhraní jako standard

Příznivý dopad na trh může mít zavedení tzv. standardů rozhraní.<sup>533</sup> Interoperabilita je forma standardizace.<sup>534</sup> To proto, že vývojář musí použít již používanou specifikaci rozhraní a svým způsobem z ní tak vzniká standard. Standardy mohou vést k vyššímu množství investic do inovací, ke zlepšení přístupnosti uživatelů k jednotlivým programům nebo omezení lock-in efektu.<sup>535</sup> Na trhu se pohybuje obrovské množství společností vyvíjejících jednotlivé produkty, které však má zákazník zájem používat komplexně.<sup>536</sup> Společnosti prodávající jednotlivé komponenty mají zájem na tom, aby mezi sebou byly kompatibilní, a proto se na trhu vyvíjejí určité standardy rozhraní, které interoperabilitu zajišťují. Problém je v tom, že není zřejmé, kdo by měl určovat tyto standardy.<sup>537</sup> Standard může určit konsenzus většiny, případně autority (např. stát). Neobvyklá není ani situace, kdy se rozhraní pro trh významné a u spotřebitelů oblíbené stane standardem.<sup>538</sup> Problém však může nastat v případech,

---

<sup>532</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 95-96.

<sup>533</sup> Tamtéž.

<sup>534</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 59.

<sup>535</sup> Shah, R. C., Kesan, J. P. *Lost in Translation: Interoperability Issues for Open Standards*. Rochester, NY: Social Science Research Network, 2008 [cit. 26. 2. 2021]. <https://papers.ssrn.com/abstract=1201708>, s. 116.

<sup>536</sup> Poudel. *Internet of Things: Underlying Technologies, interoperability, and Threats to Privacy and Security*, s. 1010.

<sup>537</sup> Van Rooijen, A. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 208.

<sup>538</sup> Samuelson, P. *Are Patents on Interfaces Impeding Interoperability?* s. 6.

kdy společnosti právě takto vzniklé „standardy“ začnou zneužívat pro svůj prospěch. Standardy by nicméně mohly být vyjmuty z ochrany. Každé rozhraní, které by se stalo standardem by tak bylo automaticky z ochrany vyloučeno a byla by umožněna interoperabilita.<sup>539</sup> Problematické však je hodnocení toho, zda dané rozhraní je standardem nebo není.

Možností je rovněž používání otevřených standardů, které ze své podstaty interoperabilitu umožňují.<sup>540</sup> Touto cestou by se mohla vydat i Evropská komise. Stále však není jasné, kdo a jak by měl standardy v případě rozhraní definovat.<sup>541</sup> Zavedení standardů však není v zájmu velkých společností, které by přišly o výhodu „lock-in“ efektu.<sup>542</sup>

## 6.5 Povinné zveřejnění specifikace rozhraní

Zvažováno by mohlo být například i povinné a automatické poskytnutí informací nezbytných pro zajištění interoperability, případně licence k rozhraní celému. Informace nebo licence k celému rozhraní by byly poskytnuty současně s licencí k počítačovému programu.<sup>543</sup> Licence by dokonce mohla být poskytnuta za poplatek. Ten by mohl být vypočítaný na základě předem stanovené základní sazby. Autoři rozhraní by tak měli kompenzováno používání jejich rozhraní.<sup>544</sup> To by je mohlo motivovat k dalšímu licencování. Zpětné inženýrství je, jak jsme si uvedli, značně náročný proces, který s neustále se zvyšující komplexitou počítačových souborů může být čím dál složitější. Alternativou pro něj je právě povinné sdělování informací o interoperabilitě nebo licencování rozhraní. V případě poskytnutí pouhých informací nezbytných pro interoperabilitu by nadále trvala časová výhoda autora, neboť implementace

---

<sup>539</sup> Palfrey, Gasser. *Interop: The Promise and Perils of Highly Interconnected Systems*, s. 12.

<sup>540</sup> Borgogno, Colangelo. *Data Sharing and Interoperability Through APIs: Insights from European Regulatory Strategy*, s. 21-22.

<sup>541</sup> Tamtéž, s. 35.

<sup>542</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 9.

<sup>543</sup> Palfrey, Gasser. *Interop: The Promise and Perils of Highly Interconnected Systems*, s. 12.

<sup>544</sup> Sagdeo. *Application programming interfaces and the standardization-value appropriation problém*, s. 260-262.

specifikace rozhraní trvá určitý čas.<sup>545</sup> Až v případě nesplnění požadavku na zveřejnění informací nezbytných pro interoperabilitu, by mohlo být umožněno zpětné inženýrství.

V případě poskytnutí licence k rozhraní by to pro zájemce znamenalo dodatečné náklady na odměnu za licenci. Otázkou je, jak by licenční poplatek mohl být definován.<sup>546</sup> Lze však mít za to, že takové zveřejnění by mohlo naplňovat požadavek přístupu společnosti k myšlenkám a principům (viz public domain) a zároveň by dostatečně chránilo autorovo dílo. Nevýhodou je, že v případě zveřejňování rozhraní autorem si autor může vybrat, které části dle jeho názoru jsou rozhraní a sdělí je a které nikoli. To může vést k neúplnosti informací. Naopak uživatel si pomocí zpětného inženýrství může získat více informací o počítačovém programu, než by bylo k zajištění interoperability nutné. V případě obou možností by bylo zcela zásadní si definovat, co přesně rozhraní je a co tedy může nebo musí být získáváno nebo sdělováno.<sup>547</sup> Problematická by také mohla být nepružnost tohoto režimu. Rozhraní se mohou podstatně lišit velikostí, složitostí a dalšími faktory, které ovlivňují náklady na jeho vývoj. Mohlo by tak docházet k nerovnoměrnému a nespravedlivému určování licenčních poplatků.<sup>548</sup>

Tato možnost byla zvažována i v rámci Evropské unie. Nakonec nebyla přijata. Evropská komise dospěla k názoru, že zřízení nových orgánů kontrolujících tuto povinnost není možné.<sup>549</sup> Dokonce i pokud by takový orgán vznikl, neměl by k dispozici žádný instrument umožňující mu adekvátní kontrolu dodržování povinností. Zároveň by byla nutná úprava značného množství vnitrostátních předpisů. Mimo jiné nebyla

---

<sup>545</sup> Van Rooijen, A. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 208.

<sup>546</sup> Borgogno., Colangelo. *Data Sharing and Interoperability Through APIs: Insights from European Regulatory Strategy*, s. 37.

<sup>547</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 93-94.

<sup>548</sup> Sagdeo. *Application programming interfaces and the standardization-value appropriation problém*, s.260-262.

<sup>549</sup> European Commission, Commission staff working document, Analysis of measures that could lead significant market players in the ICT sector to license interoperability information, s. 14-15, 17.

Evropská komise přesvědčena o tom, že je vůbec zmocněna k vydání takového předpisu.<sup>550</sup>

S touto povinností by mohla souviset i nutnost rozhraní registrovat tak, jak tomu je např. u patentu. Rozhraní by nadále bylo chráněné autorským právem, ale každý zájemce by měl možnost nahlédnout do registru a informace o rozhraní v něm získat.<sup>551</sup> Povinnost mít veřejné informace o interoperabilitě by mohla rovněž hrát roli při rozhodování o veřejných zakázkách. Komise by mohla upřednostňovat interoperabilní produkty nebo služby, a tím motivovat účastníky k interoperabilním řešením<sup>552</sup>

## 6.6 Nový regulační orgán

Než vybírat mezi všemi výše zmíněnými přístupy, možná cesta by mohla být ve vytvoření regulačního orgánu s pravomocí v oblasti tvorby pravidel, řešení sporů a monitoringu.<sup>553</sup> Ani zde však není zřejmé, jak by byl orgán vytvořen a zda by se na jeho vytvoření našla shoda v rámci Evropské unie. Bylo by totiž nutné změnit znění směrnice. To se jeví jako komplikované, vzhledem k rozsahu diskuse a množství kompromisů při přijímání aktuálního znění směrnice.<sup>554</sup> Orgán by byl úzce specializovaný a mimo vydávání pravidel by byl zejména orgánem kontrolním, ke kterému by mířily případy, kdy informace o rozhraní není autorem poskytnuta, zpřístupněna v rozsahu stanoveném zákonem. Autoři by tak zřejmě byli motivováni specifikaci rozhraní sdílet dobrovolně. Orgán by využil svých pravomocí až v případě, kdy by tomu tak nebylo. Stejně jako v případě povinného licencování však nemusí mít nový orgán k dispozici

---

<sup>550</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*, s. 16-17.

<sup>551</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 110.

<sup>552</sup> Palfrey. Gasser. *Interop: The Promise and Perils of Highly Interconnected Systems*, s. 12.

<sup>553</sup> Van Rooijen. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 224-225.

<sup>554</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 92-93.

žádný instrument umožňující mu adekvátní kontrolu. Také není zcela jisté, zda tato možnost je v kompetenci Evropské komise.<sup>555</sup>

## 6.7 Úprava omezujících ustanovení směrnice EU

Omezení v odst. 2 čl. 6 směrnice o právní ochraně počítačových programů jsou na první pohled přijatelná, zakládající rovnováhu mezi autorem a uživatelem díla. Rozhraní je zde však chráněno mnohem širěji, než by tomu bylo v případě jeho vyjmutí z autorskoprávní ochrany. V zásadě je zcela zamezeno tomu, aby rozhraní bylo, byť jakožto myšlenka autora, využíváno ostatními v jejich tvorbě a používání. Další možností je tedy úprava nebo vyjmutí druhého odstavce čl. 6 směrnice o právní ochraně počítačových programů. To by umožnilo specializaci firem na zpětné inženýrství počítačových programů, které by neprováděly pro své vlastní potřeby, ale pro potřeby zákazníků. To by vedlo k vytvoření trhu s rozhraním, což by znamenalo zpřístupnění rozhraní i menším firmám, které na vlastní zpětné inženýrství nemají peníze. To však podle Weston není možné.<sup>556</sup> Trh se bránil již přijetí směrnice o právní ochraně počítačových programů a v tomto případě lze očekávat ještě větší odpor.<sup>557</sup> Byla-li by směrnice takto upravena, měly by společnosti pravděpodobně větší zájem zveřejnit rozhraní dobrovolně. Rozklad podle článku 6 by v takovém případě byl dokonce nemožný, protože by nebyl odůvodněný. Informace nezbytné pro interoperabilitu by totiž již byly zveřejněny.<sup>558</sup> Autor by pak měl v podstatě jistotu, že nikdo zpětné inženýrství nezneužije pro jiné účely, než je právě zajištění interoperability. Weston by rovněž přistoupila na upravení odst. 1 čl. 6 směrnice, podle kterého je možné zpětné inženýrství pouze pro nezávisle vytvořený počítačový

---

<sup>555</sup> European Commission, Commission staff working document, *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*.

<sup>556</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 106.

<sup>557</sup> Tamtéž, s. 106-107.

<sup>558</sup> Viz. čl. 6 odst. 1 písm b) Směrnice o právní ochraně počítačových: „*informace nezbytné k dosažení interoperability nebyly dříve pro osoby uvedené v písm. a) snadno a rychle přístupné*“.

program.<sup>559</sup> Tato podmínka totiž jde proti smyslu autorského práva, dle kterého mají mít přístup k myšlenkám a principům díla všichni, a to i bez odůvodnění.

## 6.8 Další možná řešení

Další možností by mohlo být odložení přístupu k informacím o interoperabilitě po určitou dobu. Zájemce o informace o interoperabilitě by měl možnost zvolit si mezi možnostmi nechat si rozhraní licencovat za poplatek a s nějakým časovým odkladem nebo by se mohl pokusit o časově náročnou a nejistou dekompilaci, která by nadále byla umožněna.<sup>560</sup>

Možná je také inspirace směrnicí, kterou se stanoví evropský kodex pro elektronické komunikace.<sup>561</sup> Tato směrnice mimo jiné umožňuje regulačnímu úřadu uložit podnikům s významnou tržní silou povinnost umožnit přístup k prvkům sítě a souvisejícím zařízením. Včetně přístupu k technickým rozhraním, protokolům nebo jiným klíčovým technologiím, které jsou nezbytné pro interoperabilitu služeb.<sup>562</sup>

Možná by mohla být také inspirace průmyslovými vzory nebo ochrana patentovým právem. Karjala dělí díla na ta která nemají jinou funkci než informovat, bavit nebo zobrazovat lidské bytosti a díla která tuto jinou funkci mají, tzn. jsou funkční. První zmíněná by měla být předmětem ochrany autorských práv. Druhá by měla být předmětem ochrany patentů.<sup>563</sup> Vynálezy související s počítačem, které jsou dostatečně nové, užitečné a nejsou zřejmé pro odborníky v oboru tak mohou být

---

<sup>559</sup> Weston. *Improving interoperability by encouraging the sharing of interface specifications*, s. 109.

<sup>560</sup> Van Rooijen, A. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*, s. 83.

<sup>561</sup> Směrnice Evropského parlamentu a Rady ze dne 11. prosince 2018, 2018/1972, kterou se stanoví evropský kodex pro elektronické komunikace.

<sup>562</sup> Graef. Valcke. *Exploring new ways to ensure interoperability under the Digital Agenda*, s. 5; Srov. recitál 153 preambule Směrnice Evropského parlamentu a Rady ze dne 11. prosince 2018, 2018/1972, kterou se stanoví evropský kodex pro elektronické komunikace.

<sup>563</sup> Karjala, D. S. *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller Symposium: Copyright Protection and Reverse Engineering of Software*, s. 997.

předmětem ochrany patentů. Úřad pro patenty a ochranné známky Spojených států ani EPO se nebrání registrování patentů na prvky počítačového programu splňující požadavky patentovatelnosti.<sup>564</sup> Samuelson věří, že možnost patentovat si rozhraní není natolik problematická, jak by se mohlo zdát.<sup>565</sup> Ochrana rozhraní patentem může být menší překážkou interoperability než nevěle autora informace o rozhraní zájemci sdělit. Stejně tak může být přínosnější než provádění nezbytného zpětného inženýrství. Podle ní této možnosti nebude zneužíváno na blokování interoperability. A i kdyby, existují adekvátní možnosti, které toto zneužívání vyřeší.<sup>566</sup>

---

<sup>564</sup> McManis. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*, s. 35; Srov. Karácsony, B. *The legal Protection of Application Programming Interface*. Master Thesis, Georg-August-University in Göttingen, Faculty of Law. Unpublished manuscript, Analysis of measures that could lead significant mar2020, s. 20: „there are four requirements that APIs must comply with to be protected by patent. Such interfaces shall be regarded as an invention and must be new, inventive, and industrially applicable. Invention means a subject-matter having technical character.“

<sup>565</sup> Samuelson, P. *Are Patents on Interfaces Impeding Interoperability?* s. 54-55.

<sup>566</sup> Tamtéž, s. 33.



## 7 Závěr

Tato práce se zabývala právními aspekty interoperability software. Vzhledem ke skutečnosti, že počítačový program je chráněn autorským právem, je interoperabilita počítačových programů řešena primárně ve vztahu k autorskému právu. Cílem práce bylo nalézt odpověď na otázku, jak lze interoperability počítačových programů dosáhnout a jaké možnosti nám v tomto směru dává autorské právo, konkrétně do jaké míry jsou autorským právem chráněny informace, které jsou nezbytné pro zajištění interoperability.

Ve **druhé kapitole** jsme definovali technické pojmy a určili jejich vztah k řešené problematice. **Ve třetí kapitole** jsme odpověděli na otázku, proč je nezbytné zabývat se interoperabilitou, jak lze interoperabilitu definovat a jaké hlavní výhody a nevýhody z interoperability počítačových programů mohou plynout. Interoperabilita je pro počítačové programy zásadní, neboť je výhodné, když spolu programy umějí spolupracovat a komunikovat. Díky interoperabilitě si mohou dva počítačové programy vzájemně vyměňovat informace a navzájem využívat svých funkcionalit. Díky této schopnosti lze rychleji inovovat stávající technologie. Interoperabilita je přínosná i ve vztahu k hospodářské soutěži a ve vztahu ke spotřebitelům. Ačkoli lze nalézt i některé její nevýhody, jako je například nižší motivace autorů vyvíjet počítačové programy nebo možnost nižší bezpečnosti systému, je interoperabilita nezbytným hybatelem technologického pokroku. Interoperability lze obecně dosáhnout pomocí rozhraní počítačového programu. To lze dělit na specifikaci rozhraní a implementaci rozhraní. Pro interoperabilitu je klíčová právě specifikace rozhraní, bez které by propojení nebylo možné. Protože je rozhraní součástí počítačového programu, je nutné nalézt rovnováhu mezi motivy k zajištění interoperability a mezi zájmy autora na ochranu počítačového programu.

Ve **čtvrté kapitole** jsme odpověděli na otázku, jestli je rozhraní autorským právem chráněno nebo nikoli. Nejprve jsme určili předpoklady, které musí dílo splňovat, aby mohlo být chráněno autorským právem. Aby se mohlo jednat o dílo ve smyslu autorského práva, musí být vyjádřeno tak, že bude dostatečně přesně a objektivně identifikovatelné. Zásadní v tomto směru je také podmínka originality. Je důležité dělit rozhraní na specifikaci a implementaci, neboť podmínka originality nebude u obou stejně splnitelná. Specifikace rozhraní vzhledem ke svému

technickému a funkčnímu předurčení bude spíše neoriginálním dílem. Společnosti přináší větší užitek používání neoriginálních rozhraní, kterými je dosahováno interoperability. U implementace rozhraní bude vždy záležet na konkrétních okolnostech. Nejdůležitější se jeví počet možností, které má autor na vyjádření implementace rozhraní. V případě, že dosažení konkrétní funkce nebude možné bez odchýlení se od konkrétního vyjádření, nelze toto vyjádření chránit.

I tehdy, splňuje-li konkrétní prvek počítačového programu veškeré náležitosti kladené na autorského díla, nemusí být nutně autorským právem chráněn. V další části kapitoly jsme se proto zabývali limitací autorskoprávní ochrany, respektive nechráněnými prvky autorského díla. Zásadní roli zde hrají autorskoprávně volné složky. Do režimu public domain spadají myšlenky a principy, na kterých jsou postaveny počítačové programy a jejich rozhraní. Pro specifikaci a implementaci rozhraní tak je podstatné, zda jsou podřazeny pod chráněné vyjádření nebo nechráněnou myšlenku. Ve čtvrté kapitole jsme došli k závěru, že specifikaci rozhraní lze podřadit pod funkcionalitu počítačového programu, neboť se jedná o technickou informaci, která popisuje začátek zdrojového kódu, názvy, vstupy a výstupy a jiné informace nezbytné pro správné fungování rozhraní. Funkcionalita počítačového programu je nechráněnou myšlenkou a specifikace rozhraní tak nemůže být autorským právem chráněna. Stejně tak nemohou být chráněna rozhraní, která mají svoji podobu danou z důvodů funkčnosti. V závěru této kapitoly jsme rozebrali právní úpravu týkající se ochrany rozhraní, kterou zavedla směrnice o právní ochraně počítačových programů. Směrnice harmonizuje právní úpravu a vyjímá z ochrany rozhraní počítačových programů, která lze podřadit pod myšlenky.

Autorské právo nechrání specifikaci rozhraní. Ta však zůstává pro uživatele skryta ve strojovém kódu. Z podstaty nechráněného prvku autorského díla k němu má mít každý umožněný přístup. V **kapitole páté** jsme zjistili, že autorské právo přináší několik možností, jak se lze k nechráněným prvkům dostat. Tyto zásahy jsou dovoleny jak na unijní, tak na vnitrostátní úrovni. Právní úprava umožňuje jak zkoumání a pozorování počítačového programu za účelem zjištění myšlenek, tak také dekompilaci, která je nezbytná pro získání informací o interoperabilitě. Musí však být zároveň splněny další podmínky. Jednou z těchto podmínek je skutečnost, že dekompilaci musí provádět osoba která má k dílu oprávnění. Zároveň nesmí být informace přístupné jinak rychleji a snadněji a musí být zasaženo jen do té části díla, která je pro dosažení

informací nezbytná. Je vhodné dodat, že zpětnému inženýrství nelze zaměřit v těchto zásazích pomocí patentu s unijním účinkem ani obchodním tajemstvím. Právní úprava zpětného inženýrství však obsahuje mnohé nedostatky a nakládání se získanými myšlenkami je nadále omezoováno. Vzhledem k tomu, že je specifikace rozhraní nechráněna, jsou tato omezení nadbytečná a přespříliš omezující. Zpětné inženýrství se tak nejeví jako vhodný institut zajišťující interoperabilitu. Nehledě na to, že se nejedná o jednoduchý proces. Naopak, vyžaduje značné odborné znalosti a finanční prostředky.

V kapitole šesté jsme uvedli několik možných návrhů na doplnění nebo změnu právní úpravy tak, aby více odpovídala požadavkům na zajištění interoperability. Pravidla na ochranu hospodářské soutěže například mohou dopomoci v případech, kdy společnost zneužívá své dominantní postavení a odmítá poskytnout informace o interoperabilitě. Ideální variantou, díky které by nebylo nadále nutné provádět zpětné inženýrství nebo aplikovat pravidla na ochranu hospodářské soutěže, by mohlo být explicitní vyjmutí specifikace rozhraní z ochrany a jeho povinné zveřejnění uživateli počítačového programu. Dalším relativně účinným řešením by mohla být úprava znění ustanovení, která omezují použití informací získaných zpětným inženýrstvím.

V každém případě lze konstatovat, že by bylo vhodné autorskoprávní úpravu ochrany rozhraní modernizovat. Specifikace rozhraní, která není autorským právem chráněna by měla být uživatelům přístupná i bez nutnosti provádět další kroky. Zpětné inženýrství sice může vést k dosažení žádaných informací, může však být značně neefektivní, náročné a nákladné.

## Použité zdroje

1. Abbott, J. Reverse Engineering of Software: Copyright and Interoperability. *14 Journal of Law, Information and Science* 7 [online]. 2003. <https://www.austlii.edu.au/au/journals/JLInfoSci/2003/2.html>
2. ACIS, *Statement of Principles contained in Attachment to Letter from Peter M.C. Choy to Professor Barry E. Carter* [online]. 1992. <https://www.cciacnet.org/wp-content/uploads/2014/10/ACIS-Letter-to-Clinton-Admin-1992.pdf>
3. Aguilar, J. *Intellectual Property – Sega Enterprises Ltd. v. Accolade, Inc.: Setting the Standard on Software Copying in the Computer Software Industry*. Golden Gate University Law Review 23. Intellectual Property. 1993.
4. Anderson K.E. *Oracle v. Google – Interoperability and Copyright Infringement: IP Biz Tech Blog* [online]. 2014. <http://ipbiztech.com/oracle-v-google-interoperability-and-copyright-infringement/>
5. Andruško, M. A. *Internet, informační společnost a autorské právo*. Wolters Kluwer, 9/2016.
6. Application Programming Interface. *Computerworld* [online]. <https://www.computerworld.com/article/2593623/application-programming-interface.html>
7. Ballardini, R. M. *Intellectual Property Protection for Computer Programs: Developments, Challenges, and Pressures for Change*. Helsinki: Svenska handelshögskolan, 2012.
8. Band, J., Katoh, M. *Interfaces On Trial: Intellectual Property And Interoperability In The Global Software Industry*. Avalon Publishing, 1995.
9. Band, J. The Impact of Specific Exceptions on Fair Use: An Update Part III: Copyright Corner. *Journal of the Copyright Society of the USA*. 2016, č. 2.
10. Band, J. The Global API Copyright Conflict Special Issue: Software Interface Copyright. *Harvard Journal of Law & Technology* (Harvard JOLT). 2017, Special Issue.

11. Bernard A. Galler, *Software and Intellectual Property Protection: Copyright and Patent Issues for Computer and Legal Professionals*. Greenwood Publishing Group, 1995.
12. Bhutra, A. *Achieving Interoperability Through Reverse-Engineering: A Defense To Copyright Infringement?* [online]. 8. 6. 2020. <https://lexinsight.wordpress.com/2020/06/08/achieving-interoperability-through-reverse-engineering-a-defense-to-copyright-infringement/>
13. Borgogno, O., Colangelo, G. *Data Sharing and Interoperability Through APIs: Insights from European Regulatory Strategy*. Rochester. NY: Social Science Research Network, 2018.://papers.ssrn.com/abstract=3288460
14. Broy, M. Lehmann, M. *Die Schutzfähigkeit von Computerprogrammen nach dem neuen europäischen und Deutschen Urheberrecht*. 94 Gewerblicher Echissschutz und Urheberrecht. 1992.
15. Butler, S. et al. Maintaining interoperability in open source software: A case study of the Apache PDFBox project. *Journal of Systems and Software*. 2020.
16. Ciancarini, P. et al. *Reverse engineering: a European IPR perspective*. Pisa, Italy: Association for Computing Machinery. 2016. <https://doi.org/10.1145/2851613.2851790>
17. Commission of the European Communities, Proposal for a Council Directive on the legal protection of computer programs, 1989, COM(88) 816 final - SYN 183.
18. Committee on Computer Law, Reverse Engineering and Intellectual Property Law, 44 REC. ASS'N. B. CRRY OF N.Y., 132, 134-35 (1989).
19. Cuillou, C. M. The Reverse Engineering of Computer Software in Europe and the United States: A Comparative Approach. *Columbia-VLA Journal of Law & the Arts*. 1997, č. 4.
20. Dailey, M.E. *Abstraction, Filtration, Comparison: The Difficult Task of Defining and Applying an Appropriate Substantial Similarity Test for Computer Software*. 34 Suffolk U. L. REV. 2001.
21. Daughtrey, S. C. *Reverse Engineering of Software for Interoperability and Analysis Note*. Vanderbilt Law Review. 1994, č. 1.

22. Davey, F. Reverse engineering and copyright in programming languages and data file formats. *Journal of Intellectual Property Law & Practice*. 2013, č. 7.
23. Dokumentace Oracle. Java® Platform. Standard Edition Java Development Kit Version 9 API Specification. <https://docs.oracle.com/javase>.
24. Duan, C. Internet of Infringing Things: The Effect of Computer Interface Copyrights on Technology Standards. *Rutgers Computer and Technology Law Journal*. 2019, č. 1.
25. Ehrlich, M. A. *Fair Use or Foul Play - The EC Directive on the Legal Protection of Computer Programs and Its Impact on Reverse Engineering Notes and Comments*. Pace Law Review. 1993, č. 3.
26. Emmerik, M.V. Waddington, T. *Using a decompiler for real-world source recovery, 11th Working Conference on Reverse Engineering*. Delft, Netherlands, 2004.
27. European Commission. Commission staff working document. *Analysis of measures that could lead significant market players in the ICT sector to license interoperability information*. Brusel, 2013.
28. European Commission. *Communication from the Commission to the European Parliament*. Parl. Eur. Doc. (SEC No. 87). final – SYN 183, 1991.
29. European Commission. *Proposal for Council Directive on the legal protection of computer programs*. 17 March 1989.
30. Faust, M. J. *What Do We Do With a Doctrine Like Merger? A Look at the Imminent Collision of the DMCA and Idea/Expression Dichotomy*. Marquette Intellectual property Law Review, 2008, Volume 12, Issue 1,
31. Fichte, J. G. *Beweis der Unrechtmäßigkeit des Büchernachdrucks. Ein Raisonement und eine Parabel* [online]. 1793. [http://www.copyrighthistory.org/cam/tools/request/showRecord?id=record\\_d\\_1793](http://www.copyrighthistory.org/cam/tools/request/showRecord?id=record_d_1793)
32. Freeman, J. *What is an API? Application programming interfaces explained* [online]. InfoWorld. 8. 8. 2019. <https://www.infoworld.com/article/3269878/what-is-an-api-application-programming-interfaces-explained.html>

33. Galajdová, D. Zibner, J. *Nedostatky autorskoprávní ochrany počítačového programu* (Deficiencies of computer program copyright protection). *Právní rozhledy*. Nakladatelství C.H. Beck. 2018, vol. 26, No 22.
34. Gasser, U., Palfrey, J. G. *Breaking Down Digital Barriers: When and How ICT Interoperability Drives Innovation*. Rochester [online]. NY: Social Science Research Network. 2008. <https://papers.ssrn.com/abstract=1033226>
35. Gervais D. *The TRIPS Agreement: Drafting History and Analysis (Perspective on Intellectual Property Law)*. 2ND ed. Sweet & Maxwell, 2003.
36. Gilbert-Macmillan, K. *Intellectual Property Law for Reverse Engineering Computer Programs in the European Community*. 9 Santa Clara High Tech. L.J. 247 (1993).
37. Github. *RAML Version 1.0: RESTful API Modeling Language* [online]. <https://github.com/raml-org/raml-spec/blob/master/versions/raml-10/raml-10.md/>
38. Graef, I. *How can Software Interoperability be achieved under European Competition Law and Related Regimes?* *Journal of European Competition Law & Practice*. 2014, č. 1.
39. Graef, I., Valcke, P. *Exploring new ways to ensure interoperability under the Digital Agenda*. Emerald Group Publishing Limited. 2014, č. 1.
40. Gratz, J. Lemley, M. *Platforms and Interoperability in Oracle v. Google*. *Harrard Journal of Law and Technology*, 2018, roč. 31, special issue Spring.
41. Harrington, A. J. *Drawing the Line between Idea and Expression in Oracle v. Google: Questioning the Copyrightability of Java's Application Programming Interface Notes & Comments*. *Lewis & Clark Law Review*. 2017, č. 3.
42. Hemmendinger, D. *Computer programming language*. *Encyclopædia Britannica* [online]. 2015-1-28. <http://www.britannica.com/technology/computer-programming-language>.
43. Hinshaw, M. J. *The role of standardization and interoperability in copyright protection of computer software*. *Communication Law and Policy*. 1999, č. 3.

## POUŽITÉ ZDROJE

---

44. Hlavenka, J. a kol. *Výkladový slovník výpočetní techniky a komunikací*. Praha: Computer Press, 1997.
45. Holton, N. A. *Google, Inc. v. Oracle America, Inc.: Supreme Court Declines to Review Reversal of Landmark API Copyright Decision Casenotes*. Loyola Law Review, 2016, č. 1.
46. Huet, J., Ginsburg, J. C. *Computer Programs in Europe: A Comparative Analysis of the 1991 EC Software Directive*. Columbia journal of transnational law, 1992.
47. Chaloupková, H., Holý, P. *Autorský zákon, 5. vydání*, Nakladatelství C.H. Beck, 2017.
48. Chloupek, V., Hartvichová, K. a kol. *Patentový zákon, 1. vydání*. C. H. Beck, 2016.
49. John T. Soma, Gus Winfield & Letty Friesen, *Software Interoperability and Reverse Engineering*. 20 Rutgers COMPUTER & TECH. L.J. 189. 1994.
50. Johnson-Laird, A. *Reverse Engineering of Software: Separating Legal Mythology from Actual Technology*, 5 Software L.J. 1992.
51. Johnson-Laird, A. *Software Reverse Engineering in the Real World Symposium: Copyright Protection and Reverse Engineering of Software*. University of Dayton Law Review, 1993, č. 3.
52. Karjala, D. S. *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller Symposium: Copyright Protection and Reverse Engineering of Software*. University of Dayton Law Review, 1993, č. 3.
53. Koukal, P. *Autorské právo, public domain a lidská práva*. Brno: Masaryk University Press, 2019.
54. Last Frontier Conference Report on Copyright Protection of Computer Software, Conference Report, 30 Jurimetrics J. 15 (1989).
55. Lesshaft, K. Ulmer, D. *Urheberrechtliche Schutzziardigkeit und Tatsachliche Schutzflthigkeit von Software*, 9 Computer und Recht, 1993.
56. Lohmann, F. *The New Wave: Copyright and Software Interfaces in the Wake of Oracle v. Google*. Harvard Journal of Law and Technology, Volume 31, Special Issue Spring 2018.



57. MacCulloch, D. C. *Sega Enterprises LTD. v. Accolade, Inc.: What's so Fair about Reverse Engineering?* 14 Loy. L.A. Ent. L. Rev. 465, 1994.
58. Margoni, T. *The Harmonisation of EU Copyright Law: The Originality Standard*. Centre for IT & IP Law (CiTiP), Faculty of Law - KU Leuven; University of Glasgow - School of Law, 2019.
59. Marty, F., Pillot, J. *Intellectual Property Rights, Interoperability and Compulsory Licensing: Merits and Limits of the European approach*. Journal of Innovation Economics Management. 2012, č. 1.
60. McManis, C. R. *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community*. High Technology Law Journal, 1993, č. 1.
61. Menell, P. S. *API Copyrightability Bleak House: Unraveling and Repairing the Oracle v. Google Jurisdictional Mess*. Berkeley Technology Law Journal, 2016, č. 3.
62. Menell, P. S. *Against Defibrillating the API Copyright Dead: A Response to Advocates of Copyrightability of Software Functional Specifications Special Issue: Software Interface Copyright*. Harvard Journal of Law & Technology (Harvard JOLT), 2017, č. Special Issue.
63. Menell, P. S. *Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software* [online]. SSRN Electronic Journal. 2017. <http://www.ssrn.com/abstract=2893192>
64. Meng, Steinhardt M. S., Schubert, A. *Application Programminf Interface Documentation: What Do Software Developers Want?* Journal of Technical Writing and Communication 2018.
65. Miller, A. R. *Copyright Protection for Computer Programs, Databases, and Computer Generated Works: Is Anything New Since CONTU?* 106 HARV. L. REV. 977, 1993.
66. Moy, R. *A Case against Software Patents* [online]. Santa Clara Computer & High Technology Law Journal Vol. 17, Issue 1, <https://digitalcommons.law.scu.edu/cgi/viewcontent.cgi?article=1278&context=chtlj>

## POUŽITÉ ZDROJE

---

67. MuleSoft, U. S. *A Non-Apocalypse: APIs, Copyright, and Fair Use* [online]. Wired. 2014.<https://www.wired.com/insights/2014/05/non-apocalypse-apis-copyright-fair-use/>
68. Mylly, U. *An Evolutionary Economics perspective on Computer Program Interoperability and Copyright*. IIC Journal, 3/2010.
69. Mylly, U. *Harmonizing Copyright rules for Computer Program Interface Protection*. University of Louisville Law Review, 2009.
70. Mylly, U.-M., IPR University Center. *Intellectual property protection of computer program interfaces and interoperability*. Helsinki, 2013.
71. National Commission on New Technological Uses of Copyrighted Works (CONTU), Final Report on the National Commission on New Technological Uses of Copyrighted Works, 3 Computer L.J. 53, 1981.
72. Nádběla, J. *Velký počítačový slovník*. 1. Kralice na Hané: Computer Media, 2004.
73. *Opinion of advocate general Bot*. Delivered on 29 November 2011. Case C-406/10, SAS Institute Inc. v World Programming Ltd.
74. *Order Partially Granting And Partially Denying Defendant's Mot. For Summary Judgment On Copyright Claim*, Oracle Am., Inc. v. Google Inc., 810 F. Supp. 2d 1002 (N.D. Cal. 2011).
75. Owen, D. R. *Interfaces and Interoperability in Lotus v. Borland: A Market-Oriented Approach to the Fair Use Doctrine*. FORDHAM LAW REVIEW. roč. 64, 1996.
76. Palfrey, J. Gasser, U. *Interop: The Promise and Perils of Highly Interconnected Systems*. Basic Books, 2012.
77. Palmer, A. K., Vinje, T. C. *The EC Directive on the Legal Protection of Computer Software: New Law Governing Software Development*. International Law. roč. 2.
78. Parasidis, E. *A Sum Greater Than Its Parts? Copyright Protection for Application Program Interfaces*. Texas Intellectual Property Law Journal. roč. 14.
79. PC Mag Staff, *The Computer Language Co Inc. Definition of Compiler* [online]. <https://www.pcmag.com/encyclopedia/term/compiler>

80. Pellegrini, R., Rottmann, P., Strieder, G. *Preventing vendor lock-ins via an interoperable multi-cloud deployment approach*. 12th International Conference for Internet Technology and Secured Transactions (ICITST), 2017.
81. Pereira, A. L. D. Software interoperability, intellectual property and competition law — Compulsory licenses for abuse of market dominance. *Computer Law & Security Review*. 2011, č. 2.
82. Polčák, R a kol. *Právo informačních technologií*. Wolters Kluwer, 9/2018.
83. Poudel, S. *Internet of Things: Underlying Technologies, Interoperability, and Threats to Privacy and Security*. Berkeley Technology Law Journal. 2016, č. 2.
84. Prchal, P. *Limity autorskopravní ochrany*. Praha: Leges, 2016.
85. Professional, B. B. *Refusals to License Intellectual Property [online]*. Bloomsbury Publishing. 2011. <https://www.bloomsburyprofessional.com/uk/refusals-to-license-intellectual-property-9781841138732/>
86. Riley, C. *Unpacking interoperability in competition*. Journal of Cyber Policy. 2020, č. 1.
87. Sagdeo, P. *Application programming interfaces and the standardization-value appropriation problem*. Harvard Journal of Law and Technology, roč. 32, č. 1, 2018.
88. Salzberger, E. Elkin-Koren, N. *The Law and Economics of Intellectual Property in the Digital Age: The Limits of Analysis*. Routledge & CRC Press, 2013.
89. Samuelson, P. *Why Copyright Law Excludes Systems and Processes from the Scope of Its Protection Symposium: Frontiers of Intellectual Property: III. Copyright*. Texas Law Review. 2006, č. 7.
90. Samuelson, P. *The Past, Present and Future of Software Copyright Interoperability Rules in the European Union and United States*. European Intellectual Property Review. 2012, č. 4.
91. Samuelson, P. *Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement*. SSRN Electronic Journal. 2015. <http://www.ssrn.com/abstract=2667740>
92. Samuelson, P., Asay, C. D. *Saving Software Fair Use Future*, Volume 31, Special Issue Spring 2018. roč. 31.

93. Samuelson, P. *Are Patents on Interfaces Impeding Interoperability?* MINNESOTA LAW REVIEW. SSRN [online]. [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1323838](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1323838)
94. Samuelson, P., Vinje, T. C., Cornish, W. R. *Does Copyright Protection Under the EU Software Directive Extend to Computer Program Behaviour, Languages and Interfaces?* [online]. Rochester, NY: Social Science Research Network, 2011. <https://papers.ssrn.com/abstract=1974890>
95. Samuelson, P. *The Strange Odyssey of Software Interfaces and Intellectual Property Law*[online]. SSRN Electronic Journal. 2008<http://www.ssrn.com/abstract=1323818>
96. Samuelson, P. a kol., *A Manifesto Concerning the Legal Protection of Computer Programs*. 17 Colum.-VLA J.L. & Arts 51, 1992.
97. Samuelson, P., Scotchmer, S. *The Law and Economics of Reverse Engineering*. The Yale Law Journal. 2002, č. 7.
98. Santoro, M. a kol. *Web Application Programming Interface: general-purpose standards, terms and European Commission initiatives*. European Commission, JRC, 2019.
99. Shah, R. C., Kesan, J. P. *Lost in Translation: Interoperability Issues for Open Standards* [online]. Rochester, NY: Social Science Research Network, 2008. <https://papers.ssrn.com/abstract=1201708>
100. Shahab, A. *Data Portability: Key to Cloud Portability and Interoperability* [online]. SSRN. November 11, 2010. [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1712565](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1712565)
101. Shemtov, N. *Beyond the Code: Protection of Non-Textual Features of Software*. Oxford University Press, 2017.
102. Shapiro, Carl. *Information rules: a strategic guide to the network economy*. Varian, Hal R. Boston, Mass.: Harvard Business School Press, 1999.
103. Sharma, C. *Concentrated Digital Markets, Restrictive APIs, and the Fight for Internet Interoperability*. University of Memphis Law Review. 2019, č. 2.
104. Spoor, J. H. *Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive Symposium: Copyright Protection and Reverse Engineering of Software*. University of Dayton Law Review, 1993, č. 3.

105. Standeford, D. *Will US Follow UK Lead In Case On Copyright And Interoperability?* [online]. Intellectual Property Watch. 27. 2. 2017. <https://www.ip-watch.org/2017/02/27/will-us-follow-uk-lead-case-copyright-interoperability/>
106. Swagger, *SmartBear Software OpenAPI Specification, version 3.0.3*. [online]. <https://swagger.io/specification/>.
107. Štědroň, B. *Ochrana a licencování počítačového programu*. Praha: Wolters Kluwer ČR, 2010.
108. Šavelka, J. *Autorskoprávní ochrana funkcionality softwaru* [online]. Rigorózní práce, Brno, Masarykova univerzita, Právnická fakulta, 2013. <https://is.muni.cz/th/e974x/>
109. Tankovska, H. *Facebook: number of monthly active users worldwide 2008-2020* [online]. Statista.com, 2021. <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>.
110. Telec, I., Tůma, P. *Autorský zákon. Komentář. 2. vydání*. Praha: C.H. Beck, 2019.
111. Ulmer, E. *Urheber und Verlagsrecht. 3. vyd.* Berlin, Heidelberg (DE), New York (USA): Springer, 1980.
112. Ünver, M. B. *Turning the crossroad for a connected world: reshaping the European prospect for the Internet of Things*. International Journal of Law and Information Technology, 2018, č. 2.
113. Van Rooijen, A. *The Software Interface Between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs*. Brussels, 2010.
114. Vasilescu-Palermo, D. *APIs and Copyright Protection: The Potential Impact on Software Compatibility in the Programming Industry*. 16 J. Marshall Rev. Intell. Prop. L. 153, 2016.
115. Vezzoso, S. *Copyright, Interfaces, and a Possible Atlantic Divide*. SSRN [online]. JIPITEC, 2012. [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=2161487](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2161487).
116. Vukovic, M. *Internet Programmable IoT: On the role of APIs in IoT: The Internet of Things (Ubiquity symposium)*. Ubiquity, 2015, November.
117. Weston, S. *Improving interoperability by encouraging the sharing of interface specifications*. Law, Innovation and Technology, 2017, č. 1.

## POUŽITÉ ZDROJE

---

118. Weston, S. E. *The Legal Regulation of Interoperability in an Oligopolistic Market*. Doctorate Thesis. Bournemouth University. 2016.
119. *What is the difference between a compiled and an interpreted program?* Indiana University [online]. <https://kb.iu.edu/d/agsz>
120. Zbránek, L. Právní aspekty zpětné analýzy počítačových programů. *Revue pro právo a technologie*, 2012.
121. Zibner, J, Originalita v pojetí práva evropské unie. *Revue pro právo a technologie* [Online]. 2017, č. 15. <https://journals.muni.cz/revue/article/view/6651>.