

Právní aspekty zpětné analýzy počítačových programů

Lukáš Zbránek

1. Úvod.....	56
2. Zpětná analýza.....	56
2.1 Dekompilace - white box.....	57
2.2 Zkoumání chování programu - black box	57
3. Legislativní úprava zpětné analýzy	58
3.1 Právní úprava v USA	58
3.2 Právní úprava na úrovni EU	60
3.3 Zákonná úprava v ČR.....	61
3.4 Problematické aspekty aktuální úpravy	62
4. Možnosti smluvního vyloučení zpětné analýzy	63
5. Spor mezi SAS Institute Inc. a World Programming Ltd.	65
6. Legální způsoby provedení zpětné analýzy	70
6.1 Americký model.....	70
6.2 Evropský model	71
7. Závěr	72
8. Seznam použité literatury	73
8.1 Monografie.....	73
8.2 Ostatní zdroje.....	73
8.3 Právní předpisy.....	74
8.4 Judikatura	74

Právní aspekty zpětné analýzy počítačových programů

Abstrakt

Článek se zabývá problematikou zpětné analýzy počítačových programů. Jeho cílem je ukázat různé možnosti zpětné analýzy a zhodnotit legálnost takového konání dle platné právní úpravy. Také je posuzována právní závaznost smluv vylučujících zpětnou analýzu. Dosud nevyřešené spory budou základem pro úvahy nad možným budoucím vývojem této oblasti práva.

Klíčová slova

zpětná analýza, software, autorské právo, DRM, dekompilace, black box, white box, EULA, DMCA, EU Software Directive, shrink wrap

Abstract

Article deals with reverse engineering of computer programs. Its aim is to show the different possibilities of reverse engineering and review the legality of such acting in accordance with valid legislation. Also, it is considered legality of contracts prohibiting reverse engineering. Pending litigation will be the basis for consideration of possible future amendments to this field of law.

Key words

reverse engineering, software, copyright, DRM, decompilation, black box, white box, EULA, DMCA, EU Software Directive, shrink wrap

Mgr. Ing. Lukáš Zbránek



lukas.zbranek@gmail.com

Mgr. Ing. Lukáš Zbránek je absolventem Fakulty informačních technologií VUT v Brně a Právnické fakulty MU. V současné době se věnuje programátorské praxi a problematice autorskoprávní ochrany počítačových programů. Mimo jiné také poskytuje poradenství v oblasti právních otázek distribuce softwaru a poskytování hostingových služeb.

1. Úvod

V dnešní době plní informační technologie stále důležitější úlohu v takřka všech oblastech běžného života. V návaznosti na to roste poptávka po počítačových programech různého zaměření, což v konečném důsledku vede k tomu, že konkurenční boj se stává stejnou samozřejmostí, jako ve většině ostatních odvětví lidské činnosti. Tento boj představuje na poli počítačových programů na jednu stranu hybnou sílu pokroku, na stranu druhou s sebou však nese také etické a právní problémy spojené s porušováním autorských práv.

Cílem tohoto článku je tedy nahlédnout do problematiky zpětné analýzy počítačových programů. Jako ve všech jiných odvětvích, kde ji lze aplikovat, se jedná o vysoce specializovanou činnost, která může posunout danou problematiku kupředu, stejně jako může být zneužita k prostému plagiátorství. Jedná se však o poměrně specifické téma, které převážně zaujme pouze osoby pohybující se v oboru tvorby a distribuce software. Budu se však snažit oslovit i širší publikum, neboť se jedná o zajímavou oblast práva, jejíž zpracování dosud obsahuje některá slepá místa.

V úvodu článku se pokusím přiblížit čtenáři samotný pojem zpětné analýzy spolu s jejím rozdělením na její nejběžnější realizace. Nastíním pozitiva a negativa jednotlivých metod zpětné analýzy zejména s ohledem na možnosti jejich praktického využití.

Následovat bude rozbor různých právních systémů a jejich postoj ke zpětné analýze. Použiji metodu komparace pro srovnání, jak je na zpětnou analýzu nahlíženo v právním systému Spojených Států Amerických se zaměřením na federální úpravu, a jak se s touto problematikou vypořádávají státy Evropského společenství prostřednictvím směrnic Evropského parlamentu a Rady. Současně se budu zabývat i tím, jak jsou tyto směrnice implementovány do českého právního řádu a zda neexistují nesrovnalosti mezi komunitárním právem a jeho promítnutím do práva českého.

Zajímavým aspektem samotné existence zpětné analýzy jsou snahy o její smluvní vyloučení. Tato snaha je vedena požadavky autorů na maximální možnou míru utajení vlastního know-how, která však nemusí být slučitelná s právy garantovanými jednotlivými právními systémy. V kapitole věnované smluvnímu vyloučení zpětné analýzy se tak budu zabývat vymahatelností takových smluvních ustanovení ve světle amerických federálních zákonů a judikátů a porovnávat ji se situací v Evropě, zejména pak v České republice.

V praktické části tohoto článku se budu zabývat aktuálním případem společnosti SAS Institute vs. World Programming Limited, který probíhá na anglické půdě a jehož výsledky mohou mít zcela zásadní dopad na vývoj softwaru obecně, zejména pak na oblast open source software, která často nabízí alternativy ke komerčním produktům se zachováním jejich funkcionality a kompatibility s proprietárními datovými formáty. Proberu jak skutkovou podstatu jednotlivých bodů tohoto sporu, tak i jejich relevanci z pohledu komunitárního práva Evropského společenství. Tento případ

je také zajímavý svou povahou, neboť obě strany sporu souhlasí s tím, že žalovaná společnost World Programming Limited neměla přístup ke zdrojovým kódům programu SAS, a to ani v originále, ani ve zprostředkované formě získané pomocí dekompilace, přesto je však veden spor o porušení autorského práva. Dalším důležitým faktorem je, že se jedná o spor týkající se tzv. drop-in náhrady softwaru, tj. zástupného řešení. Ačkoli není program SAS příliš znám běžným uživatelům, spor týkající se drop-in řešení by mohl mít důsledky například pro kancelářské balíky typu Open Office a jiné open source náhrady komerčního softwaru.

Cílem tohoto článku je zpracovat ucelený přehled možností, které stávající právo nabízí v případě praktické aplikace zpětné analýzy. Tento přehled se pokusím namodelovat na hypotetických situacích, které budu posuzovat dle amerických a evropských právních norem. Tato část článku by mohla sloužit jako vodítko při snaze o legální postup při zpětné analýze počítačových programů.

Vzhledem k rychle se vyvíjícímu prostředí softwarového práva tento článek trpí nedostatkem publikací věnovaných tématu. Hlavním zdrojem informací tak bude převážně americká judikatura, internetové příspěvky a články sborníků věnovaných počítačovému právu.

2. Zpětná analýza

Zpětná analýza je metoda, kterou se zjišťují principy a funkce již hotového výrobku, ke kterému zkoumající nemá potřebné podklady. Motivy pro takové konání mohou být vědecké a zájmové, či také čistě finanční.

V prvním případě lze o zpětné analýze mluvit např. při verifikaci softwaru, zejména se to týká bezpečnostních programů, zda splňují bezpečnostní normy, kterými se často honosí. Dále lze zpětnou analýzu využít také při výuce pro lepší pochopení funkcí různých procesů.

Druhým motivem mohou být náklady ušetřené při přeskočení dlouhého vývoje a testování, pokud chce nějaký subjekt uvést na trh substitut pro jiný výrobek. Zde již lze narazit na morální i právní hranice, za něž by se tento subjekt neměl dostat. Světlou stránkou je pak rychlejší vývoj, udržování konkurenčního prostředí a zlepšování výrobků. Nové programy obsahují různé inovace, na kterých pak opět staví i programy další. Dále je potřeba vzít i v úvahu, že s každým dalším programem, který je kompatibilní s původním zkoumaným, dále stoupá hodnota zkoumaného programu, jedná se o tzv. síťový efekt¹.

V případě softwaru se však řešení sporů komplikuje, neboť nehmotná povaha softwaru zde vtahuje i autorské právo, které by se v jiných případech neuplatnilo. Zejména tím, že během zpětné analýzy dochází k vytvoření kopie autorsky chráněného díla. V oblasti

1 ZIEMINSKI, C. Game over for Reverse Engineering: How the DMCA and Contracts Have Affected Innovation. In: *Journal of Technology Law & Policy*, Vol. 13, Issue 2 [online]. 2008 [cit. 2011-10-07]. s. 289 – 340. Dostupné z: <database HeinOnline.org>.

softwaru jsou dva nejčastější způsoby zpětné analýzy, a to dekompilace a analýza metodou black box.

2.1 Dekompilace - white box

Během dekompilace softwaru je binární kód předkládán zpátky do lidsky čitelné podoby pro další zkoumání algoritmů programu. K tomuto účelu existují speciální nástroje, ovšem nelze si představovat, že stačí zkoumaný program otevřít v dekompilátoru a člověk získá přehledný zdrojový kód. Během kompilace zdrojového kódu se totiž v rámci úspory systémových prostředků provádí nevratné změny, jako je odstranění všech komentářů kódu, zrušení formátování, překlad některých částí kódu do samotných souborů a jiné.

Výsledkem práce dekompilátoru pak nejčastěji bývá poměrně nepřehledný kód, který na první pohled říká pouze jaké jsou v paměti prováděny instrukce a na jakých adresách jsou data. Zkušený programátor si z toho ovšem může po podrobné analýze udělat celkový přehled o fungování programu.

V případě použití white box zpětné analýzy k vytvoření konkurenčního programu se stejnou funkcionalitou jakou má program zkoumaný, vzniká i riziko kopírování kódu. Pokud již programátor vidí, jak je daná funkce implementována, může být v pokušení danou část kódu převzít, aniž by se pokoušel o vlastní řešení problému. Autorovi programu pak hrozí, že v případném soudním sporu dojde na úplný zákaz distribuce programu, který je postaven z různých okopírovaných fragmentů kódu. Řešením této situace může být tzv. metoda clean room², vyvinutá firmou Phoenix, která měla zajistit, že výsledný program nebude možné napadnout pro porušení autorských práv kvůli kopírování kódu. Principem této metody je rozdělení pracovníků na tým analytiků a tým programátorů. Úkolem analytiků je prozkoumání původního programu a sepsání podrobné zprávy o tom, jak program funguje. Výsledná zpráva však musí být dostatečně abstraktní bez jakýchkoli referencí na programovou implementaci. Tým programátorů tak vůbec nepřijde do styku s původním programem a pouze dostane podrobnou zprávu od analytiků. Na základě těchto informací vytvoří nový program, který má funkcionalitu shodnou s předlohou. Zároveň je tím minimalizováno riziko, že bude možno identifikovat v obou programech stejný fragment kódu. I pokud by se našla část kódu, která je obdobná v obou porovnávaných programech, nejednalo by se o porušení autorských práv, neboť tým programátorů k dané implementaci došel vlastní tvorbou. Obecně se dá říci, že u běžných programů je pravděpodobnost takové shody malá. Ovšem v případě programů, které operují na nejnižších úrovních počítačových systémů, tj. zejména ovladače hardware, kde je potřebná co největší efektivita, se může stát, že více programátorů dojde ke stejné implementaci určité funkce programu. Firma Phoenix tuto metodu využila při vytváření klonu zaváděcího programu BIOS

2 MOY, R. A Case against Software Patents In: Santa Clara Computer & High Technology Law Journal, Vol. 17, Issue 1 [online]. 2000 [cit. 2012-05-12]. Dostupné z: <<http://pop.cctlj.org/sites/default/files/media/articles/v017/v017.i1.Moy.pdf>>.

počítačů IBM v 80. letech a odstartovala tím celou éru IBM kompatibilních počítačů.

Nutno také podotknout, že pouze pomocí white box analýzy lze nalézt některé chyby programu, bezpečnostní rizika nebo funkcionalitu programu, kterou výrobce nikde neuvádí. Příkladem budiž i aféra kolem společnosti Digital Convergence Corporation³. Ta byla výrobcem čteček čárových kódů CueCat, které distribuovala uživatelům jako dárky k předplatnému některých časopisů a dokonce i zdarma v některých obchodech s elektronikou. Pokud uživatele zaujal nějaký výrobek v časopise označený čárovým kódem, stačilo použít čtečku a přiložený software otevřel příslušnou webovou stránku daného produktu. Co už ovšem Digital Convergence nikde neuváděla, byl fakt, že každá čtečka byla vybavena unikátním sériovým číslem⁴, které odesílala spolu s požadavkem na hledaný produkt. Společnost Digital Convergence tak postupně vytvářela databázi obsahující informace o uživatelích a produktech, o které projevíli zájem. Po zveřejnění této informace následoval útok na servery společnosti Digital Convergence, během kterého došlo ke krádeži osobních údajů zhruba 140 000 uživatelů. Po tomto skandálu Digital Convergence zkrachovala a oficiální stránky zajišťující vyhledávání produktů pomocí CueCat byly vypnuty. Díky zpětné analýze se však téměř okamžitě po zveřejnění objevilo i mnoho návodů a programů, které zajišťovaly, že sériové číslo čtečky nebude odesíláno při hledání výrobků.

2.2 Zkoumání chování programu - black box

Druhá z možností zkoumání softwaru se zakládá na pozorování chování programu, tj. jaké generuje program výstupy na základě zadaných vstupů. V tomto případě se nemanipuluje se zdrojovým kódem, ani není nijak odhalován. Testováním různých vstupních hodnot pak zkoumající osoba získá znalosti o vnitřních principech programu, aniž by znala jeho zdrojový kód. Vyjma zkoumání samotného programu se do této metody může řadit také zjišťování informací z dokumentace programu či jiných zdrojů.

Výsledkem této činnosti je ucelený obraz o způsobu práce programu, jeho komponentech a potřebných formátech vstupních a výstupních dat. Zkoumající osoba však neví nic o vnitřní logice programu, nezná jednotlivé funkce nebo konkrétní způsoby jejich implementace. Pokud je tento druh zpětné analýzy použit k vytváření programů se stejnou funkcionalitou jako originál, stále musí tvůrce nového programu vynaložit dostatek práce a schopností, aby principy zjištěné black box zpětnou analýzou dovedl přenést do praxe ve formě korektně fungujícího počítačového programu.

Stejně jako v případě využití techniky clean room white box zpětné analýzy, ani zde není nijak vysoké riziko, že se v novém programu objeví části kódu příliš

3 OLSEN, S. „Cat“ scanning device may track users online. [online]. [cit. 2012-05-12]. Dostupné z: <<http://news.cnet.com/2100-1023-246008.html&ctag=st.ne.1002.tgif.ni>>.

4 tamtéž.

podobné kódu zkoumaného programu, neboť se plně projeví odlišný styl různých programátorů a programátoři nového programu nejsou ovlivněni znalostmi implementace programu původního.

3. Legislativní úprava zpětné analýzy

Legislativní otázky použití zpětné analýzy jsou velmi úzce spjaty s autorským právem, proto je upravena zejména v předpisech týkajících se této problematiky. V této kapitole tedy budou rozebrány relevantní články z amerických i evropských předpisů a také se budu zabývat zhodnocením jejich formulací jak z pohledu technické realizace, tak i z pohledu právního.

Primárním problémem při dekompilaci programu je fakt, že vyjádření programu je zkopírováno. Už to samo o sobě představuje zásah do oprávněných zájmů autora a vztahují se na něj předpisy chránící autorská práva napříč různými systémy. Rozdíly se pak projevují v daných úpravách podle toho, zda je legální možnost dekompilace vázána na konkrétní podmínku, či zda lze dekompilaci vyjmout z protiprávního jednání argumentací poukazující na obecné principy autorskoprávního odvětví.

3.1 Právní úprava v USA

Úprava zpětné analýzy v USA je vlivem samotné podstaty angloamerického právního systému zaznamenána zejména v klíčových sporech, které jako první definovaly některé problémy. Jeden z nich je např. *Computer Associates vs. Altai*⁵, kde byl poprvé uveden třístupňový Abstraction-Filtration-Comparison test. Jeho cílem je odstranit irelevantní části kódu tak, aby bylo možné porovnat pouze ty části, na které se bude vztahovat autorskoprávní ochrana.

Tento test spočívá ve třech krocích - během abstrakce je zkoumaný program nejdříve rozdělen na jednotlivé funkční bloky podle jejich činnosti. Poté je potřeba provést filtraci, která každý funkční blok očistí od prvků, které nejsou pod autorskoprávní ochranou (typicky reprezentace matematických vzorců apod.). Současně se zjišťuje, zda posuzovaný kód spadá pod pravidlo efektivnosti, zda je nezbytný pro běh programu a nakonec zda se nejedná o volné dílo (tzv. public domain dílo). Pokud daný blok splňuje některé z těchto kritérií, je v rámci filtrace vyřazen z porovnávání. Výsledkem předcházejících dvou kroků je určitý výtah počítačového programu, který je v posledním kroku porovnáván s originálním programem. Na tomto výtahu se pak posuzuje, zda došlo k doslovnému či nedoslovnému kopírování. Tento test byl postupně přijat všemi americkými soudy a v poslední době došlo k jeho zařazení mezi posudkové nástroje i u soudů ve Spojeném království a ve Francii.

5 Rozhodnutí Druhého obvodu odvolacího soudu Spojených států ze dne 22. 6. 1992 ve věci *Computer Associates International, Inc. v. Altai, Inc.*, sp. zn. 982 F.2d 693, (2d Cir. 1992). [online]. [cit. 2011-12-15] Dostupné z: <<http://www.bitlaw.com/source/cases/copyright/altai.html>>.

Další spor, který upravoval poměry i pro následující soudní pře týkající se zpětné analýzy, byl *Atari v. Nintendo*⁶. Ten se týkal hardwarové ochrany her (tzv. NES10) vydávaných společností Nintendo pro její konzoli NES. Díky NES10 bylo možné na konzoli NES spustit pouze hry vydané přímo nebo pod licencí Nintendo. Tato licence mimo jiné zahrnovala přísná omezení na maximálně pět her ročně a také zákazy licencování těchto her pro jakékoli další herní konzole. Společnost Atari se dlouhou dobu snažila přijít na způsob, jakým je tato ochrana řešena. To zahrnovalo i zpětnou analýzu, při které byly originální herní nosiče rozebrány a inženýři Atari se snažili zjistit kód přímo z mikročipu, který tuto ochranu zajišťoval. Protože tento proces nebyl stoprocentně úspěšný, kontaktoval zaměstnanec Atari americký Úřad pro autorská práva (US Copyright office) s tvrzením, že Atari je v soudním sporu se společností Nintendo a potřebují zde uložený zdrojový kód NES10 z pro potřeby důkazního řízení. Takto získaný kód pak porovnali s vlastními výsledky zpětné analýzy a doplnili chybějící části. Díky tomuto pak Atari vyvinulo vlastní program *Rabbit*, který simuloval ochranu NES10 a umožňoval použití neoriginálních herních nosičů v konzoli Nintendo NES.

Následovala žaloba firmy Nintendo pro porušení autorské ochrany a patentu. Atari se bránilo, že dekompilace spadá do tzv. fair-use⁷ využití, uvedeného v § 107 amerického federálního zákona US Code: Title 17 (dále jen USC:T17)⁸, které určitým způsobem omezuje držitele autorských práv. Soud však rozhodl, že fair use se nelze dovolávat, pokud sami získali kód NES10 na základě falešných tvrzení o probíhajícím sporu. Součástí výroku bylo, že lze provádět zpětnou analýzu pouze na legálně získaných produktech, tzv. politika čistých rukou. Částečně z tohoto rozsudku tedy vyplývá, že je na uvážení soudu, zda shledá předložené argumenty jako dostatečné pro souhlas s dekompilací jako použití fair use obrany.

Jiným příkladem použití fair use obrany je případ *Sega v. Accolade*⁹. Ten se rovněž týkal herních konzolích, konkrétně *Sega Genesis*, a ochrany, která měla limitovat použití pouze na ty hry, které byly vydány firmou Sega. Společnost Accolade zakoupila uvedenou herní konzoli a tři *Sega* hry, které následně podrobila pečlivému zkoumání. Inženýrům Accolade se podařilo identifikovat 25 bytů dlouhý kód, díky kterému byly herní nosiče rozpoznány a konzole spustila hru na nich uloženou. Následně také vydali manuál, který popisoval

6 Rozhodnutí Odvolacího soudu Spojených států ze dne 10. 9. 1992 ve věci *Atari Games Corp. v. Nintendo of America Inc.*, sp. zn. 975 F.2d 832 (Fed. Cir. 1992) [online]. [cit. 2011-12-15] Dostupné z: <http://scholar.google.com/scholar_case?case=15866317401594691669&eq=975+F.2d+832&hl=en&cas_sdt=2002>.

7 ABBOT, J. Reverse Engineering of Software: Copyright and Interoperability. In: *Journal of Law and Information Science* Vol. 14 [online]. 2003 [cit. 2012-02-07]. s. 7 – 49. Dostupné z: <[database HeinOnline.org](http://database.HeinOnline.org)>.

8 USC : Title 17 - Copyrights. 1998. [cit. 2011-12-15] [online]. Dostupné z: <<http://www.law.cornell.edu/uscode/text/17>>.

9 Rozhodnutí Devátého obvodu odvolacího soudu Spojených států ze dne 20. 10. 1992 ve věci *Sega Enterprises Ltd. vs Accolade, Inc.*, sp. zn. 977 F.2d 1510 (9th Cir. 1992). [online]. [cit. 2011-12-15] Dostupné z: <<http://bulk-resource.org/courts/gov/c/F2/977/977.F2d.1510.92-15655.html>>.

rozhraní konzole Genesis a prvky nutné pro vytváření kompatibilních her.

V následujícím soudním procesu se během prvoinstančního a dalších odvolacích řízení vystříдалo několik protichůdných rozsudků, které se zabývaly možností vytvoření kopie programu, která je nutná pro dekompilaci softwaru. Byly posuzovány i podmínky dané § 107 USC:T17, jako např. komerční zájem na vytvoření produktu a následně obchodní ztráty společnosti Sega. V poslední instanci (Devátý obvod odvolacího soudu - US Court of Appeals for the Ninth Circuit) bylo však rozhodnuto, že *pokud je dekompilace jediným způsobem, jak získat přístup k myšlenkám a principům, na kterých je program založen, a je zde legitimní cíl pro snahu o přístup k nim, pak spadá dekompilace do rozsahu oprávněného využití díla*¹⁰.

V tomto případě bylo zkopírování programových částí konzole Sega považováno za nutný krok v cestě za zjištěním myšlenek a principů, což bylo dle soudu primárním cílem. Vytvoření dalších samostatných her, jejichž rozhraní byla založena na informacích získaných dekompilací, bylo hodnoceno jako odvozenina od samotného procesu zjišťování principů a myšlenek.

Dále se již budu v této části zabývat USC:T17, upraveném novelou známou jako Digital Millennium Copyright Act (dále jen DMCA)¹¹. Tato novela byla schválena v roce 1998 a jejím cílem byla implementace smlouvy Světové organizace duševního vlastnictví o právu autorském (dále jen WIPOCT)¹². Již během svého projednávání se stala tato novela velmi kontroverzním tématem. Mezi její nejdiskutovanější body patřila zejména kriminalizace vytváření a šíření nástrojů pro obcházení ochrany před kopírováním digitálních autorských děl. Toto je také oblast, ve které je v DMCA explicitně zmiňována zpětná analýza, konkrétně se jedná o kapitoly 12 obsahující paragrafy 1201 až 1205.

První sekce této kapitoly hned v úvodu zakazuje obcházení technologických řešení sloužících ke kontrole přístupu k dílu (dále jen DRM). Tato DRM řešení mohou mít formu softwarových ochranných systémů jako CSS či SecuROM, nebo se může jednat o hardwarové klíče, tzv. dongly. V další části prvního odstavce jsou pak zmíněny třídy děl, na které zákaz obcházení DRM nemá vliv. Tyto třídy definuje knihovnický Kongres. Každé tři roky je pak knihovnickem prováděna revize těchto tříd na základě doporučení Registru autorských práv. Aktuálně jsou ze zákazu obcházení technologických ochranných systémů tyto třídy děl:

1. *Legálně získané filmy na DVD chráněné systémem CSS, kdy je jeho obejití potřebné pro vykopírování části filmu pro použití v následujících oblastech:*

i. *Ukázkové použití univerzitními pedagogy*

nebo studenty filmu a mediálních studií.

ii. *Tvorba dokumentárních filmů.*

iii. *Tvorba nekomerčních videí.*

2. *Počítačové programy umožňující spouštět aplikace na mobilních telefonech, pokud je ochrana vyražena právě z důvodu zajištění interoperability mezi aplikací a programovou vybavou telefonu.*

3. *Počítačové programy, které umožní připojení mobilního telefonu do sítě operátora.*

4. *Počítačové hry pro PC, které jsou chráněny systémy pro kontrolu přístupu k dílu. Jejich obcházení je vyňato ze zákazu za podmínky, že je prováděno za účelem testování, zkoumání nebo oprav bezpečnostních chyb a zranitelností pokud:*

iv. *jsou získané informace použity k posílení bezpečnosti majitele počítačového systému či sítě.*

v. *jsou získané informace použity tak, že nedochází k porušování autorských práv.*

5. *Počítačové programy, které umožňují obcházení ochranných systémů pomocí donglů, pokud došlo k poruše donglu a je již zastaralý. Za zastaralý je považován takový dongle, jehož náhrada není komerčně dostupná za vynaložení přiměřených prostředků.*

6. *Knihy v elektronické podobě, jejichž ochranné prostředky brání funkci pro hlasitě předčítání nebo zobrazení textu do specializovaného formátu. Zde lze tuto ochranu obejít jen pokud jsou takto chráněny všechny dostupné distribuce dané knihy.*¹³

Další odstavce §1201 pokračují v duchu zákazu obcházení DRM tak, že zakazují rovněž výrobu a distribuci jakýchkoli nástrojů, jejichž význam spočívá zejména v obcházení DRM.

Následující odstavce se věnují dalším výjimkám, které za určitých okolností vylučují protiprávnost obcházení DRM podle DMCA. Pro tuto práci důležitou částí je pak jasné vymezení, za jakých podmínek je možné legálně obcházet DRM za účelem provádění zpětné analýzy, které je uvedené v odstavci f) § 1201 USC:T17. Dle kontextu se dá usuzovat, že je tím myšlena zpětná analýza ve formě dekompilace, kdy může DRM efektivně bránit k přístupu do paměti obsazené daným programem. Provádět takto zpětnou analýzu je možné pouze za předpokladu, že získané informace jsou potřeba pro zajištění interoperability zkoumaného softwaru s jinými programy. Použití zpětné analýzy ovšem přichází v úvahu až v případě, kdy potřebné informace nebyly veřejně zpřístupněny a nelze je získat jinak.

Při dodržení výše uvedených podmínek je možné i vyvjet a používat nástroje sloužící k odstranění DRM. Stejně tak je možné i zjištěné informace předat dalším osobám. Pouze musí být naplněna podmínka, že důvodem šíření získaných informací je dosažení interoperability zkoumaného softwaru s dalšími programy.

10 Rozhodnutí Devátého obvodu odvolacího soudu Spojených států ze dne 20. 10. 1992 ve věci Sega Enterprises Ltd. vs Accolade, Inc., sp. zn. 977 F.2d 1510 (9th Cir. 1992). [online]. [cit. 2011-12-15]. Dostupné z: <<http://bulk.resource.org/courts.gov/c/F2/977/977.F2d.1510.92-15655.html>>.

11 Digital Millennium Copyright Act. In: PUBLIC LAW 105-304—OCT. 28, 1998.

12 WIPO Copyright Treaty. 1996. [online]. [cit. 2011-12-20]. Dostupné z: <http://www.wipo.int/treaties/en/ip/wct/trtdocs_wo033.html>.

13 Rulemaking on Exemptions from Prohibition on Circumvention of Technological Measures that Control Access to Copyrighted Works. [online]. [cit. 2011-12-20]. Dostupné z: <<http://www.copyright.gov/1201/2010/>>. Překlad autor.

Mezi další činnosti, které vylučují protiprávnost podle § 1201 USC:T17 patří také zkoumání šifer a šifrovacích metod za účelem dalšího vývoje kryptografie, dále obcházení DRM, které kromě ochrany díla také sbírají informace o uživateli, ovšem pouze za účelem zamezení sběru osobních informací a jako poslední oblast jsou uvedeny bezpečnostní testy. Legálnost zkoumání bezpečnosti a hledání slabín ve výše uvedených technických prostředcích je podmíněna dobrou vírou testující osoby, že tak koná v rámci posílení bezpečnosti svého počítačového systému nebo sítě, a je zakázáno jakékoli zneužití informací získaných při testování. Bohužel toto jsou aspekty, které jsou zkoumány až v případném soudním sporu, takže osoba testující bezpečnost nemá apriorní jistotu, že proti ní nebudou provedeny žádné právní kroky.

Již na první pohled jsou uvedené výjimky velmi specifické a znesnadňují i legitimní vědeckou činnost v oblasti softwaru. Zejména nejistota vedla k útlumu vědeckého bádání na poli počítačové bezpečnosti. Současně také dochází ke stavu, kdy díky DRM nemohou nabyvatelé autorskoprávně chráněných děl využívat některá svá vlastní práva. V modelové situaci si majitel hudebního CD nemůže převést toto CD do přenosného MP3 přehrávače, aniž by tím porušil některé z ustanovení uvedených v USC:T17. V literatuře je někdy tento vliv označován jako *chilling effect*. V oblasti sporů se objevilo i označení *Sega era*¹⁴ označující přístup soudů k softwarovým sporům v době před a po účinnosti DMCA.

Oproti původnímu cíli DMCA, což byla implementace WIPOCT, se její formulace stala podstatně přísnější a výrazně utlumila použití zpětné analýzy. Dalším efektem je zřejmě i horší situace v oblasti vytváření konkurenčních programů a programům zajišťujících interoperabilitu s jinými programy, to vše pod hrozbou civilních i trestních sankcí.

Další hrozbou, kterou DMCA přináší je možnost zneužití autorských práv. Koncept výjimek zaručujících legálnost vyřazení DRM je natolik přísný, že až neúměrně posiluje autorská práva. V případě zabezpečení programu pomocí DRM tak dochází k zúžení možností legální dekompile programů. Autoři tak mají možnost program chránit pomocí DRM, čímž dosáhnou toho, že i části programu které by normálně nepožily ochrany autorského práva, budou efektivně tímto právem chráněny. Tím je naplněn princip tzv. *zneužití autorských práv*¹⁵, který ovšem americké soudy stále příliš neakceptují.

3.2 Právní úprava na úrovni EU

Na úrovni komunitárního práva je toto téma zpracováno v několika směrnicích Evropského parlamentu

14 ZIEMINSKI, C. Game over for Reverse Engineering: How the DMCA and Contracts Have Affected Innovation. In: *Journal of Technology Law & Policy*, Vol. 13, Issue 2 [online]. 2008 [cit. 2011-10-07]. s. 289 – 340. Dostupné z: <data-báze HeinOnline.org>.

15 HEINDL, P. TTTF Working Papers - A Status Report from the Software Decompilation Battle: A Source of Sores for Software Copyright Owners in the United States and the European Union? [online]. 2008 [cit. 2012-06-10]. Dostupné z: <www.law.stanford.edu/program/centers/ttlf/papers/heindl_wp1.pdf >.

a Rady. V první řadě je ovšem potřeba zmínit, že směrnice Evropského parlamentu a Rady nelze při komparaci stavět na stejnou úroveň jako normativní právní akt (v podobě DMCA) neboť pouze určuje cíle, kterých se má v právním systému členských států dosáhnout. Jakou formu a doslovný obsah jednotlivé státy zvolí, to je již ponecháno na jejich uvážení.

Otázkami DRM a zákazem jejich překonávání se zabývá směrnice 2001/29/ES (dále jen EUCP)¹⁶. Ta je ve své podstatě obdobně zpracována jako DMCA, kdy stanovuje zákaz obcházení DRM a přidává i zákaz jakýchkoli změn či odebrání informací o elektronické správě práv.

Druhou směrnicí, která se týká již přímo zpětné analýzy je směrnice 2009/24/ES (dále jen EUSD)¹⁷. Ta již adresuje konkrétní otázky spojené s autorstvím počítačových programů, vyhrazených práv autorů a také jejich omezení. Také se zabývá zpětnou analýzou, a to jak ve formě black box, tak ve formě dekompile počítačového programu. Evropská úprava považuje vytvoření kopie programu jeho dekompilací za zásah do autorských práv. Možnost legální dekompile však existuje a je vázána na taxativně uvedené podmínky, pro které není nutný souhlas autora programu s dekompilací.

Výjimky, pro které není při nakládání s programem potřebné svolení autora, jsou popsány v článku 5 EUSD. Kromě běžné činnosti oprávněného uživatele s programem, což zahrnuje vytváření dočasné kopie v paměti počítače apod., sem patří i oprava chyb. To je již ale činnost, pro kterou je většinou potřeba detailní znalost vnitřního fungování programu. Pokud tyto informace neposkytne autor programu, pak uživatel nemá jinou možnost, než použití zpětné analýzy i ve formě dekompile. V případě opravy chyb totiž není dostačující přibližná znalost fungování programu, jakou poskytne black box analýza, nýbrž je potřeba najít konkrétní komponentu, která chybu způsobuje. Tato možnost opravy programu vlastními silami je nicméně podmíněna formulací v úvodu prvního odstavce článku 5 EUSD:

„Pokud nejsou ve smlouvě sjednána zvláštní ustanovení...“¹⁸.

Z mého pohledu toto rozhodně není ideální formulace. Stalo se totiž zvykem, že při instalaci komerčních programů musí uživatel odsouhlasit i licenční podmínky a podmínky užívání programu. A drtivá většina těchto podmínek má v ustanoveních formulaci, kterou zakazuje jakoukoli formu zpětné analýzy. Proto si myslím, že poslanci Evropského parlamentu a Rady měli tuto část kompletně vypustit. Nicméně problematikou těchto restriktivních licenčních ustanovení se budu zabývat v samostatné kapitole a nyní se budu dále věnovat jednotlivým relevantním ustanovením EUSD.

16 Směrnice Evropského parlamentu a Rady 2001/29/ES: o harmonizaci některých aspektů autorského práva a práv s ním souvisejících v informační společnosti. In: L 167/44. 2001.

17 Směrnice Evropského parlamentu a Rady 2009/24/ES: o právní ochraně počítačových programů. In: L 111/16. 2009.

18 Směrnice Evropského parlamentu a Rady 2009/24/ES: o právní ochraně počítačových programů. In: L 111/16. 2009.

V druhém odstavci již není na smluvní ustanovení brán ohled a je přímo stanoveno, že smlouva nesmí oprávněnému uživateli bránit v pořízení záložní kopie, pokud je nezbytná pro provoz programu. Na tomto místě vyvstává otázka, jak je vyřešeno zajištění tohoto práva v případě, že je program chráněn pomocí DRM tak, jak je upravuje EUCP. Důvodová zpráva EUCP jasně říká, že právní ochrana DRM nesmí bránit nebo zakazovat jejich obejití při výkonu práv uvedených v EUSD článku 5 odstavec 3 a v článku 6. V tomto případě se ale jedná o odstavec druhý článku 5, a ustanovení tak budí dojem, že v případě ochrany programu pomocí DRM jej není možné legálně vyřadit z činnosti pro získání záložní kopie. Stejný problém by nastal i v předchozím případě při pokusu o opravu chyby v takto chráněném programu. Naproti tomu článek 5 odstavec 3 se zabývá black box zpětnou analýzou, tedy zkoumáním chování programu. V tomto případě však nemá odstraňování ochran programu žádný smysl, neboť při black box analýze se s programem pracuje dle jeho zamýšleného určení a ochrana dila by při tom neměla tvořit žádnou překážku. Proto se s formulací důvodové zprávy nedá úplně souhlasit. Korektní formulací v duchu tohoto cíle by mělo být, že právní ochrana DRM se neuplatní v případech zmíněných v článku 5, odstavcích 1 a 2 a v článku 6. Důvodová zpráva má však poskytnout jakési vodítko při implementaci uvedených ustanovení, a proto je možné, že v některých členských státech, jsou dané výjimky uvedeny logicky správně.

Jak jsem již uvedl výše, třetí odstavec článku 5 uvádí v podstatě definici analýzy black box, tedy explicitně povoluje bez potřeby autorova souhlasu zkoumání či testování funkčnosti programu vedoucí ke zjištění myšlenek a zásad, na kterých je program postaven. Za povšimnutí stojí, že není uvedeno, jak je možné s takto získanými informacemi nakládat. Dle zásady co není zakázáno, je dovoleno, by pak mělo být možné získané informace použít k vytvoření konkurenčního produktu.

Článek 6 EUSD probírá možnosti dekompilace a podmínky, za jakých je možné dekompilaci provádět bez svolení autora. Vyjma možnosti opravy chyb je podle EUSD jediným dalším legálním důvodem pro dekompilaci získání informací o interoperabilitě daného programu. I při snaze o dosažení interoperability je potřeba naplnit kumulativní podmínky stanovené v prvním odstavci.

Samozřejmostí pro provedení dekompilace je její provedení oprávněným uživatelem dané rozmnožení programu. V případě jednodušších programů pro koncové uživatele je to jednoduché, ale v případě komplexnějšího softwaru, často směřovaného do podnikové sféry, je potřeba mít na paměti i použití správné licence.

Další podmínka staví dekompilaci do pozice posledního možného řešení. Dekompilaci povoluje pouze v případě, že hledaná informace o interoperabilitě nebyla snadno a rychle přístupná. Tím je myšleno zejména, zda požadovaná informace není již někde zveřejněna. Vzhledem k náročnosti celé dekompilace si myslím, že je tato podmínka již nadbytečná. Není pravděpodobné, že by někdo prováděl dekompilaci, aniž by

danou informaci dříve nehledal z důvodu značné úspory vynaloženého času a úsilí. Pokud však nejde o pouhé nalezení informace o interoperabilitě, ale daná osoba činí všechny potřebné kroky k jejímu dosažení, pak stejně bude muset program dekompilovat k provedení potřebných úprav a následně jej znovu zkompile do strojového kódu.

Poslední podmínka klade důraz na minimalizaci zásahu do programu. Takže nelze při úpravě programu zasahovat do těch jeho částí, které nemají žádný vliv na interoperabilitu daného programu s jiným. Tímto výčet podmínek končí, nicméně je potřeba mít na paměti, že při dekompilaci je potřeba splnit každou z nich.

Druhý odstavec článku 6 omezuje nakládání s informacemi získanými pomocí dekompilace programu. V první řadě uvádí obecné zákazy, jako zákaz použití informací k jiným účelům, než dosažení interoperability a zákaz předání informací jiným osobám. Řekl bych, že zákaz použití k jiným účelům, než dosažení interoperability dostatečně vyplývá již z prvního odstavce článku 6, zde se mi zdá tudíž nadbytečný. Naproti tomu zákaz předání informací cizím osobám, tedy těm, které se osobně nepodílí na dosažení interoperability, je zcela na místě. V opačném případě by totiž mohla nastat situace, kdy by jedna skupina osob mohla, po dosažení interoperability, předat získané informace dál, a na jejichž základě by mohl vzniknout konkurenční produkt. Toto by rozhodně nebylo v souladu se zamýšleným účelem výjimky ze souhlasu autora programu jak jej stanoví článek 6 EUSD. To dokládá i poslední ze zákazů uvedených ve druhém odstavci článku 6, a to že získané informace nesmí být použity pro vývoj konkurenčního počítačového programu.

Ke srovnání s DMCA vybízí zejména garantovaná ochrana DRM. Zde lze usuzovat, že ačkoli i EUSD a zejména EUCP respektují nároky autorů na účinnou ochranu svých děl před neautorizovaným šířením, větší důraz je jednoznačně kladen na práva legitimních uživatelů softwaru.

3.3 Zákonná úprava v ČR

Na základě směrnic EUCP a EUSD byl i vytvářen a následně upraven autorský zákon č. 121/2001 Sb. (dále jen AZ)¹⁹. V době vytváření AZ byla i paralelně formována směrnice EUCP. Proto se tvůrci AZ snažili již vytvářet tento zákon v souladu s tehdy aktuálními návrhy EUCP, nicméně průtahy při přijímání EUCP způsobily, že byl vydán až rok po přijetí AZ. Tímto následným vývojem EUCP po přijetí AZ samozřejmě vzniklo pár rozdílů, které bylo později nutné opravit novelizacemi AZ.

Směrnice EUSD byla sice přijata relativně nedávno (na poměry časových období potřebných k promítnutí směrnic do vnitrostátního práva), nicméně tato směrnice z podstatné části kopíruje dřívější směrnici Rady 91/250/EHS z května 1991. EUSD zejména urovnává některé sporné části směrnice Rady 91/250/EHS, které

19 Zákon č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon). In: 121/2000. 2010.

vznikly přijetím pozdějších předpisů (např. směrnice Rady 93/98/EHS). Podstatná část komunitární úpravy problematiky ochrany počítačových programů byla již známa dlouho před sestavováním AZ, a proto je jeho úprava v souladu i s novějším EUSD.

Základní ustanovení AZ týkající se ochrany počítačových programů je uvedeno v § 65, kde je, z pohledu ochrany softwaru, o počítačovém programu a přípravných materiálech uvažováno jako o literárním díle. V dalším odstavci je pak v duchu obecných pravidel ochrany počítačových programů stanoveno, že ochranu dle AZ nelze uplatňovat na myšlenky a principy programu. Na tomto místě bych se pozastavil nad formulací „*myšlenky a principy, ..., včetně těch, které jsou podkladem jeho propojení s jiným programem*“²⁰. Pokud má být tato část v souladu s komunitárním právem, je jí zřejmě myšleno rozhraní programu, tak jak je to uvedeno i v oficiálních překladech směrnice Rady 91/250/EHS i EUSD. Nevím proč zákonodárce nepoužil již zavedené názvosloví, a místo toho se za každou cenu snažil vytvořit originální název, ale v této podobě je to přinejmenším zavádějící. Jakoby se ochrana nevztahovala pouze na rozhraní umožňující propojení s jiným programem. Ale přitom i formát vstupních nebo výstupních dat tak, jak je zadává uživatel programu, je možné považovat za rozhraní programu. A dle výše uvedené formulace by se na daný formát dat již vztahovala ochrana podle AZ, což by ovšem podle mého názoru bylo v první řadě v rozporu s komunitárním právem, v druhé řadě i v rozporu se zdravým rozumem. Rozhodně bych doporučil lépe specifikovat, na co se ochrana dle AZ vztahuje a nevztahuje. Ideální se mi pak jeví změna celé výše uvedené konstrukce na prosté rozhraní.

Stejně jako DMCA, i AZ poskytuje ochranu DRM, ať už generální klauzulí o zákazu obcházení DRM přístupu § 43 odst. 1, nebo také zákazem vytváření a distribuce nástrojů, které jsou primárně určeny k jejich obcházení a nemají jiný významnější účel či využití. Od DMCA se AZ odchyľuje spíše při stanovování výjimek, na které se právní ochrana DRM nevztahuje. Ty jsou uvedeny v § 43 odst. 4 a zahrnují např. možnosti využití chráněného díla či jeho části pod úřední a zpravodajskou nebo knihovnickou licenci.

Pro téma tohoto článku je však nejdůležitější částí AZ §66, který obsahuje omezení rozsahu práv autora k počítačovému programu. Jejich výčet je uveden v odstavci prvním, a uvádí tyto situace:

- a) *rozmnnožování, překlad, zpracování, úpravy či změny počítačového programu, pokud je to nezbytné při využívání legálně nabytého programu či při opravě jeho chyb.*
- b) *zhotovení záložní rozmnoženiny programu, je-li nezbytná pro jeho užívání.*
- c) *zkoumání a studium programu za účelem zjištění myšlenek a principů, na nichž je kterýkoli prvek programu postaven. Tuto činnost může provádět sám*

oprávněný uživatel nebo jím pověřená osoba, ovšem může to provádět pouze takovým užíváním programu, ke kterému je oprávněn. Opět usuzuji, že přesná konstrukce uvedená v AZ §66 odst. 1 písmeno d) je popisem black box zpětné analýzy.

- d) *dekompilace programu a s ní spojené operace se strojovým a zdrojovým kódem, které jsou potřebné pro zajištění interoperability s jinými programy.* I toto může vykonávat oprávněný uživatel nebo jím pověřená osoba, ovšem až v případě, že informace získané dekompilací nejsou dostupné jiným, snadnějším a rychlejším postupem.

Odstavec čtvrtý §66 pak zakazuje jakékoli jiné využití informací získaných dekompilací, zejména k vytvoření konkurenčních produktů.

Součástí § 66 je v odstavci pátém odkaz na ustanovení, že výjimky a omezení autorského práva lze využít pouze ve zvláštních a zákonem stanovených případech, přičemž tato omezení nesmí představovat nepřiměřený zásah do oprávněných zájmů autora a musí být v souladu s běžným využitím díla. Zkoumání myšlenek a zásad rozhodně za nepřiměřený zásah nepovažují, stejně tak je logické, aby měl uživatel možnost opravy jakýchkoli chyb používaného programu.

V souvislosti s tím osmý odstavec § 66 odebírá garanci právní ochrany DRM právě v případě, že DRM brání využití práv oprávněného majitele kopie programu ke zkoumání myšlenek a principů zásad, na kterých program staví, a v případě zajišťování vzájemné interoperability s jinými programy. Problematické je, že po těchto dvou situacích není explicitně uveden i případ, kdy uživatel musí dekompilovat program při snaze o opravu chyby. Zde je totiž větší pravděpodobnost kolize s ochranou programu, která mu tuto činnost ztěžuje, než při prostém pozorování chování programu v rámci black box analýzy. Současně však tento odstavec i nařizuje autorovi programu zpřístupnit program oprávněnému uživateli v celém rozsahu uvedených výjimek. Dále má autor v případě použití DRM povinnost označit takto chráněný program kontaktními údaji osoby, na kterou se má oprávněný uživatel obrátit v případě uplatňování svých práv podle § 66 odstavce prvního.

AZ tedy respektuje ustanovení uvedená v EUSD a ve stávající podobě je s ním ve shodě. Určité úpravy by byly vhodné zejména pro zpřesnění definic, především jednoznačné určení právní ochrany rozhraní. Při porovnání s právní úpravou panující v USA lze pozorovat zejména větší důraz na spotřebitelská práva, kdy jsou tato práva spotřebitelům garantována i v případě, že by účinkovala proti snaze autorů o maximální zabezpečení svých děl proti zneužití.

3.4 Problematické aspekty aktuální úpravy

Jedním ze základních kamenů úrazu aktuální úpravy je absence přesných definic, se kterými by se dalo pracovat při posuzování konkrétních případů. Je sice přínosné, že uživatel může vyvinout úsilí ke zjištění myšlenek a principů, na nichž program staví, avšak již chybí informace,

²⁰ Zákon č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon). In: 121/2000. 2010.

do jaké míry je možné uvedené myšlenky a principy reprodukovat.

To se týká například rozhraní programu – EUSD říká, že *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*²¹. Ale jaký je tedy status ochrany rozhraní? Rozhraní je integrální součástí počítačového programu, bez něj sice může program existovat, ale nikdo (člověk ani jiný program) se nedozví co program provádí, čímž vlastně popře důvod své existence daný utilitaristickou povahou softwaru. Výše uvedená definice staví ochranu rozhraní na stejnou úroveň jako ochranu samotného programu, ale proč je v takovém případě rozhraní explicitně uvedeno? A pokud by se na rozhraní programu vztahoval jiný režim, pak je uvedená formulace rovněž špatná, protože nic takového neříká.

Z pohledu programátorského i z pohledu právního jsem toho názoru, že ochrana rozhraní programu by měla být explicitně upravena, ovšem jinak, než je její stávající podoba. Pokusím se tento problém popsat na následujícím modelovém případě. Mám program A, který má velmi striktní pravidla pro formát vstupních dat, a z důvodu snahy o maximální efektivitu je napsán v nízko úrovněovém programovacím jazyce assembler. Pokud vytvořím program B, který bude akceptovat stejné vstupy a rovněž bude napsán v assembleru, pak budou relevantní části obou programů velmi podobné, ne-li stejné. Rázem se tak dostávám do situace, kde mi hrozí žaloba z porušení autorských práv. A výsledek případného sporu je velmi obtížné předem určit, protože bude na posouzení konkrétního soudce, zda došlo či nedošlo k podstatnému rozmnožení kódu programu A.

Na základě výše uvedeného tedy zastávám názor, že rozhraní programu by měla požívat jednoznačně nižší úroveň autorskoprávní ochrany než samotný program, a hlavně rozsah této ochrany by měl být přesně definován. To by bylo přínosem z hlediska právní jistoty a mohlo by to přispět k dalšímu rozvoji navzájem interoperabilních programů.

Evropskému právnímu prostředí by dále pomohlo jednoznačné určení vztahu autorskoprávních ochran softwaru a právních ochran DRM. Jde mi zejména o možnost uživatele opravit chybu v programu, která je však zcela bezzubá v případě ochrany programu pomocí DRM.

Co by však bylo vhodné upravit na celosvětové úrovni je naprosto přehnaná ochrana DRM samotného. DRM jakožto prostředek snahy o zamezení neoprávněného šíření softwaru by měl mít svou funkci a ochranu pouze pro tento účel. Faktický stav je však takový, že právní ochrana DRM, prosazená pravděpodobně pouze díky nátlaku zájmových skupin, je zneužívána k nepřiměřené ochraně programů daleko nad rámec ochrany, kterou poskytují autorské právo.

4. Možnosti smluvního vyloučení zpětné analýzy

Je samozřejmě v zájmu autorů komerčních počítačových programů, aby dosáhli co nejsilnějšího postavení na trhu ve svém oboru. Logickou součástí tohoto cíle je snaha o ochranu vlastního know-how. Proto se autoři snaží zajistit, aby zdrojové kódy programu nebyly snadno rozluštěitelné. Prvním krokem k dosažení tohoto cíle je již samotná podoba, ve které jsou programy distribuovány, tj. drtivá většina programů je k dispozici pouze jako již zkompileované spustitelné soubory. Tím, že se ke koncovému uživateli dostanou programy již ve formě strojového kódu, se prakticky eliminuje možnost, že by se ke zdrojovým kódům snažil dostat běžný uživatel. S distribucí programu včetně zdrojových kódů se můžeme setkat spíše u open source softwaru. Zde se ovšem předpokládá, že uživatelé zkoumající kód využijí svých schopností k dalšímu zdokonalování původního softwaru.

Strojový kód ovšem nezastaví před odhalováním zdrojového kódu úplně všechny. Jak jsem již uvedl, dekompilece není triviální záležitost, nicméně v některých případech bude její použití levnější než vývoj vlastního softwaru od samého počátku. Na evropské úrovni je pro autory situace zjednodušena tím, že dekompilací se vytváří v podstatě nelegální kopie autorského díla, a je ji tedy možno využít pouze v zákonem předvídaných situacích. Nehmotná povaha softwaru a jeho snadná distribuce tak může mít za následek to, že ochrana dostačující v Evropě, nemusí stačit v USA nebo kdekoli jinde na světě. Jelikož je zde možnost bránit dekompilaci jako použití fair use, začaly být programy často chráněny pomocí DRM, které kontroluje spuštění a běh programu, a jejichž cílem je dekompilaci zabránit. Díky DMCA, která poskytuje silnou ochranu DRM a možnosti jejich vyřazení omezuje na konkrétní výčet situací, se legální možnosti dekompilece omezily zhruba na úroveň, jaká panuje v Evropě.

Poslední obrannou linií pro uchování tajemství kódu programu se tak stalo smluvní právo. Jedním ze způsobů, jak si vynutit vlastní podmínky, je zřetelný nápis na obalu produktu, který upozorňuje na zákaz dekompilece zakoupeného produktu. Svůj souhlas dává zákazník najevo porušením obalu a použitím programu. Pokud však zákazník odmítne tyto podmínky, následuje informace, že program v neporušeném obalu má vrátit prodejci a převzít zpět kupní cenu programu. Tento postup je možný pouze u programů, které byly zakoupeny na fyzickém médiu. V případě digitální distribuce se lze setkat se situací, kdy je odsouhlasení smluvního ujednání nutnou podmínkou pro samotné stažení zakoupené kopie programu.

Nejčastějším případem je ovšem zakomponování licenčního ujednání přímo do instalátoru samotného programu. Během instalace, kdy jsou nastavovány parametry programu, je přidán krok, ve kterém je zobrazena tzv. End User License Agreement (dále jen EULA). Zde je uživatel vyzván, aby si veškeré podmínky podrobně přečetl a až s jejím odsouhlasením je vůbec

21 Směrnice Evropského parlamentu a Rady 2009/24/ES: o právní ochraně počítačových programů. In: L 111/16. 2009.

možné pokračovat v instalaci programu. Obvykle obsahuje rovněž ustanovení, že v případě nesouhlasu má uživatel přerušit instalaci a vrátit program prodejci, kde obdrží zpět náklady vynaložené na koupi programu.

Jako ukázkou EULA z reálného prostředí jsem vybral licenční podmínky vztahující se na balík kancelářských programů Office 2010 společnosti Microsoft. Pro přehlednost uvádím pouze odstavce, které zmiňují zpětnou analýzu:

7. *ROZSAH LICENCE. Software se neprodává, pouze se uděluje licence k jeho užívání. Tato smlouva vám poskytuje pouze určitá uživatelská práva k funkcím obsaženým v licenciované verzi software. Všechna ostatní práva si vyhrazuje společnost Microsoft. Pokud vám rozhodně právo bez ohledu na tato omezení neposkytuje více práv, smíte software užívat pouze způsobem výslovně povoleným v této smlouvě. Současně musíte dodržovat veškerá technická omezení v software, která umožňují jeho užívání pouze určitými způsoby. Nesmíte:*

- *překračovat žádná technická omezení software,*
- *provádět zpětnou analýzu software, jeho dekompilaci ani převod ze strojového kódu s výjimkou a v rozsahu takových aktivit, které jsou výslovně povoleny rozhodným právem bez ohledu na toto omezení,*
- *vytvořit více kopií software, než je určeno v této smlouvě nebo povoleno rozhodným právem bez ohledu na toto omezení,*
- *zveřejnit software, aby jej ostatní mohli kopírovat,*
- *užívat software jakýmkoli způsobem, který je v rozporu se zákonem,*
- *pronajímat software, půjčovat jej nebo poskytovat na leasing ani*
- *užívat software pro komerční hostitelské služby.²²*

V první řadě tato smlouva stanovuje, že oprávněný uživatel se nestává vlastníkem kopie balíku Office 2010, pouze jeho uživatelem. Tato diferenciací by měla význam v případě, že by práva přiznaná kogentními normami byla vázána pouze na vlastníka kopie programu. Naštěstí obvykle normy upravující zpětnou analýzu hovoří o oprávněném uživateli, tedy není zde podmínkou vlastnictví programu. Např. AZ v § 66 odstavec 6 uvádí oprávněného uživatele jako „*oprávněného nabyvatele rozmnoženiny počítačového programu, který má vlastnické či jiné právo k rozmnoženině počítačového programu, a to za účelem jejího využití, nikoli za účelem jejího dalšího převodu, dále oprávněný nabyvatel licence nebo jiná osoba oprávněná užívat rozmnoženinu počítačového programu.*“

Dále je licence vedena co možná nejrestriktivnějším způsobem v zájmu autora programu – uživatel licence je oprávněn pouze k činnosti stanovené smlouvou. Tím se autor programu chrání např. před novými způsoby zpětné analýzy neznámé v době vydání programu. Současně se snaží uživatele omezit v jeho právech na nejnižší možnou míru, kterou mu dává stát, jehož právem se řídí. Ve většině států se s největší pravděpodobností

22 Microsoft Software License Terms (MSLT) for Microsoft Office 2010. [online]. [cit. 2012-05-07]. Dostupné z: <<http://www.microsoft.com/downloads/cs-cz/details.aspx?displaylang=cs&FamilyID=2b37ad5e-dc64-400b-a00d-88d982292b07&CTT=5&origin=HA101817777>>

uplatní normy na ochranu spotřebitele a právní vztah mezi uživatelem programu a jeho autorem se budou řídit právem dle sídla uživatele.

V této EULA je i vymezen vztah mezi jejími ustanoveními a kogentními normami státu uživatele programu. Práva daná oprávněnému uživateli programu ustanoveními z EULA mohou být navýšena pouze v případě, že uživateli kogentní normy daného státu přiznávají vyšší práva, žádné jiné navýšení není možné. Z uvedených zákazů jsou nejzásadnější první dva, tj. zákaz obcházení DRM a zákaz zpětné analýzy.

Na americké půdě by se řešil rozpor mezi licenčním ujednáním řídicím se právem daného státu a federální úpravou. Obecná představa je, že federální právo má aplikační přednost před státními právy a smluvními ujednáními. Ovšem nelze to brát tak, že automaticky jsou veškerá omezení zpětné analýzy v nevynutitelná. Jak uvádí Elizabeth Morris²³, federální právo bude mít přednost před státním pouze v případě konfliktu v určitém bodě licenčního ujednání.

Zde by takový konflikt mohl představovat zákaz zpětné analýzy podle licenčního ujednání versus obrana zpětné analýzy jako využití fair use dle § 107 USC:T17. Ovšem přiznání obrany zpětné analýzy jako fair use není automatické²⁴. § 107 uvádí několik prvků, na jejichž posouzení závisí, zda bude fair use sloužit jako prvek ospravedlňující zpětnou analýzu. Mezi takové patří třeba zhodnocení, zda je zpětná analýza použita k vytvoření komerčního produktu, a jak moc zasáhne jeho uvedení na trh do hodnoty původního programu apod. Tento princip se ukázal v případě Bowers v. Baystate Technologies²⁵. Baystate Technologies dekompilovala Bowersův doplněk pro CAD programy a na základě výsledků dekompilace vytvořila vlastní konkurenční program. Když byla žalována Bowersem za porušení autorských práv a smluvních podmínek, snažila se Baystate Tech. argumentovat rovněž tím, že díky využití fair use nejsou smluvní podmínky vynutitelné. Její obrana fair use ovšem před soudem neobstála, a proto musela krom škody na autorských právech uhradit rovněž částku za porušení smluvních podmínek, které zakazovaly dekompilaci. Tedy ačkoli smluvní podmínky na rozdíl od federálního práva nepřiznávaly žádné povolené případy dekompilace, přesto byly tyto podmínky vynutitelné, protože nebyly naplněny podmínky nutné pro uplatnění výjimek stanovených federálním právem.

Další možnou obranou zpětné analýzy by byla doktrína zneužití autorských práv²⁶. Tu je možné využít

23 MORRIS, E.M.N. Will Shrinkwrap Suffocate Fair Use. In: Santa Clara Computer & High Technology Law Journal, Vol. 23, Issue 2 [online]. 2006 [cit. 2011-10-07]. s. 237 – 270. Dostupné z: <[databáze HeinOnline.org](http://www.heinonline.org)>.

24 BRESSMAN, S. Restricting Reverse Engineering through Shrink-Wrap Licenses: Bowers v. Baystate Technologies, Inc. In: Boston University Journal of Science & Technology Law, Vol. 9, Issue 1 [online]. 2003 [cit. 2011-11-07]. s. 185 – 193. Dostupné z: <[databáze HeinOnline.org](http://www.heinonline.org)>.

25 Rozhodnutí Odvolacího soudu Spojených států ze dne 29. 1. 2003 ve věci Bowers v. Baystate Technologies, sp. zn. [2003] 320 F.3d 1317. [online] [cit. 2011-11-18]. Dostupné z: <http://scholar.google.com/scholar_case?case=11442789824223925516>.

26 ANDREWS, J. A. Reversing Copyright Misuse: Enforcing Contractual Prohibitions on Software Reverse Engineering. In: *Houston Law Review*, Vol. 41, Issue 3 [online]. 2004 [cit. 2011-10-07]. s. 975 – 1016. Dostupné z: <[databáze Houston Law Review](http://www.houstonlawreview.com)>.

v případě, že nositel autorských práv zneužije svého postavení, aby autorskoprávní ochranu rozšířil i na části díla, která na tuto ochranu nemají nárok. V případě softwaru se tak použije EULA, aby bylo uživateli zabráněno v zobrazení principů a myšlenek, na kterých program stojí, a kterým nepřísluší autorskoprávní ochrana. Bohužel tuto obranu nelze brát jako jednoznačně garantovanou, a protože tato otázka není zatím uspokojivě vyřešena, bude záležet na posouzení konkrétního soudu, jak se k otázce doktríny zneužití autorských práv postaví.

Zákaz překračování technických omezení softwaru, jak jej stanoví výše uvedená EULA balíku MS Office 2010, lze vykládat více způsoby. Na jednu stranu by se mohlo jednat o zákaz jakýchkoli programových úprav směřujících k rozšíření funkčnosti programu – typicky prodloužení data expirace testovací verze programu, která bývá zdarma. Druhým možným významem tohoto ustanovení je, že je zakázáno jakékoli obcházení DRM. To je zakázáno i federální úpravou, která stanovuje pouze taxativní výčet výjimek, pro které tento zákaz neplatí. Pokud by k obejití došlo, je pak na uživateli, aby dokázal, že jeho konání spadá do některé z výjimek uvedených v § 1201 USC:T17. V tomto bodě tedy federální úprava nahradí tu smluvní.

Pokud tedy budeme uvažovat druhý význam, tak z pohledu evropského práva je možnost legálního vyřazení DRM ponechána na zákonodárství jednotlivých států tak, jak je uvedeno v článku 6 odstavci 4. EUCP. V případě českého AZ se tyto výjimky uplatní pouze při výkonu práv oprávněného uživatele stanovených v § 66 odstavec 1., písm. d) a e) AZ. Autor je však povinen oprávněnému uživateli zpřístupnit celý program podle § 66 odstavce 1. Ostatně tato EULA i uvádí, že v závislosti na sídle uživatele mu může být místními normami garantován širší okruh práv.

Co se týká zákazu provádění zpětné analýzy a dekompilace programu, tak v tomto ohledu je EUSD poměrně jednoznačná. V článku 8 se píše, že jakákoli smluvní ustanovení, která jsou v rozporu s odstavci 2 a 3 článku 5 a celého článku 6, jsou od počátku neplatná. Česká implementace EUSD v § 66 AZ stanoví minimální rozsah uživatelských práv, který nelze smluvně omezit. Jakékoli smluvní ustanovení, které je v rozporu s AZ tak rovněž pozbyde platnosti. Promítl se tak silný princip smluvní volnosti v českém právu.

Jiným příkladem sporného ustanovení, které se ovšem běžně vyskytuje v licenčních ujednáních počítačových programů, je pokus autora o zřeknutí se odpovědnosti za škodu způsobenou daným počítačovým programem. Záměrně uvádím, že se jedná o pokus, neboť většina států zřejmě neuzná vynutitelnost takového ustanovení. V českém právním řádu by se uplatnil § 574 odstavec 2. občanského zákoníku²⁷, který nepřipouští vzdání se práv, které mohou teprve vzniknout. Již citovaná EULA kancelářského balíku programů MS Office toto upravuje následovně: *Omezení a vyloučení náhrady škody. Od společnosti Microsoft a jejich dodavatelů můžete získat pouze náhradu za přímé škody až*

do výše částky, kterou jste za software zaplatili. Nezískáte náhradu za žádné jiné škody, včetně následných škod, ušlého zisku a zvláštních, nepřímých nebo náhodných škod. Současně je zde ale i uvedeno, že v některých státech nelze tuto odpovědnost smluvně vyloučit a proto se toto ustanovení smlouvy nemusí na daného zákazníka vztahovat.

Ačkoli je tedy patrné, že výrobci počítačových programů mají logickou snahu o potlačování dekompilace různými způsoby včetně smluvního vyloučení, přesto narážejí na kogentní normy, které tuto možnost garantují. Kombinaci různých metod směřovaných k zamezení zpětné analýzy je možné pozorovat v praxi na aktuálním případě uvedeném v následující kapitole.

5. Spor mezi SAS Institute Inc. a World Programming Ltd.

Jedním, ze zrovna probíhajících sporů, je případ americké společnosti SAS Institute Inc. (dále jen SAS Inc.) proti anglické World Programming Ltd. (dále jen WPL), který probíhá ve Spojeném království. Tento spor je zejména důležitý tím, že si soudce Sir Richard D. Arnold vyžádal stanovisko ESD k některým článkům Evropské směrnice 2009/24/ES (dále jen EUSD), které by mohlo mít fundamentální význam pro všechny budoucí spory s obdobným obsahem. Dalším zajímavým prvkem tohoto sporu je množství otázek a posuzovaných principů autorské ochrany počítačových programů, zejména pak hranice mezi myšlenkou a jejím fyzickým vyjádřením nebo také úroveň vynaložené práce a schopností potřebných k vytvoření programu. Na následujících řádcích se tedy budu zabývat skutkovým stavem sporu, jehož fakta jsem čerpal ze spisu, který je dohledatelný pod kódem [2010] EWHC 1829 (Ch)²⁸. Dále se zaměřím na problematiku otázek, které budou řešeny Evropským soudním dvorem, a také se je pokusím posoudit na základě vlastního pohledu na věc, ovlivněného jak právem, tak i znalostmi z oboru informačních technologií.

Předmětem sporu je domnělé porušení autorských práv ke statistickému softwaru SAS při tvorbě konkurenčního programu společností WPL. Program SAS je tvořen různými moduly a jeho základní komponenta Base SAS podporuje také tvorbu uživatelských skriptů psaných v SAS Language, tj. specifickém programovacím jazyku, jehož syntaxi uměl zpracovat parser obsažený v SAS. Během mnoha let svého působení na trhu si SAS našel mnoho uživatelů, kteří v SAS Language vytvořili velké množství vlastních skriptů. Některé z nich však svou složitostí již připomínaly samostatné programy, jejichž tvorba zabrala hodně úsilí a času. Problémem byla jejich nepřenositelnost, takže mohly být provozovány pouze ve spojení se SAS, což mnohdy převážilo onen pomyslný jazýček vah při rozhodování o přechodu na jiný statistický software. Přepis některých velmi složitých skriptů by znamenal další dodatečné náklady a čas strávený testováním, takže někteří

báze HeinOnline.org>.

27 Zákon č. 40/1964 Sb., občanský zákoník, ve znění pozdějších předpisů.

28 SAS Institute v World Programming [2010] EWHC 1829 (Ch) / Case C-406/10.

zákazníci již zůstávali u programu SAS pouze z nemožnosti jiné volby.

Této situace se rozhodla využít společnost WPL, která přišla s ideou vytvořit konkurenční produkt, který by umožňoval spouštění a zpracování skriptů vytvořených pro SAS. Jejich cílem bylo tedy vytvořit samostatný produkt, který bude na zadané vstupní hodnoty podávat naprosto stejný výstup jako konkurenční SAS. Tento produkt pojmenovali World Programming System, zkráceně WPS. K dosažení tohoto cíle se rozhodli postupovat zpětnou analýzou samotného programu SAS a důkladným zkoumáním velice podrobného manuálu programu SAS. Samozřejmě neexistoval žádný zákonný důvod, který by dovoľoval dekompilaci programu, takže WPL zvolilo metodu black box, během které nechali SAS zpracovat množství vzorových skriptů a pečlivě studovali jeho výstupy.

Proti tomu se společnost SAS Inc. ohradila žalobou, ve které uvádí tato čtyři základní tvrzení:

- 1) tvrzení, že při tvorbě WPS společnost WPL zkopírovala manuály SAS, čímž porušila autorská práva k manuálům SAS,
- 2) tvrzení, že při kopírování manuálů SAS společnost WPL nepřímou zkopírovala programy tvořící komponenty SAS (doplňkové moduly k Base SAS), čímž porušila autorská práva ke komponentám SAS,
- 3) tvrzení, že společnost WPL použila učební edici (tzv. „Learning Edition“) programu SAS, čímž porušila licenční podmínky a tím pádem i relevantní smlouvy a autorská práva k učební edici SAS,
- 4) tvrzení, že společnost WPL zkopírovala manuály SAS při vytváření manuálu k WPS a stručných návodů k WPS.²⁹

Pro správné posouzení uvedených tvrzení je nutné nejdříve určit, jaké právní normy budou použity. Základní mezinárodní smlouvou pro řešení autorskoprávních sporů je Bernská úmluva o ochraně literárních a uměleckých děl (dále jen „BÚ“), která slouží jako podklad pro samotné hodnocení, zda se o kopírování jedná či ne. Relevantními články jsou zejména 2, 9 a 20, pro přehlednost zde uvedu z mého pohledu nejpodstatnější článek 2, odstavce 1 a 5:

(1) Výraz „literární a umělecká díla“ zahrnuje všechny výtvořiny z literární, vědecké a umělecké oblasti, bez ohledu na způsob nebo formu jejich vyjádření jako: knihy, brožury a jiná díla písemná; přednášky, proslovy, kázání a jiná díla téže povahy; dramatická nebo hudebně dramatická díla; choreografická díla a pantomimy; hudební skladby s textem nebo bez textu; filmová díla, jímž jsou postavena na roveň díla vyjádřená způsobem obdobným filmu; díla kreslířská, malířská, architektonická, sochařská, rytecká, litografická; fotografická díla, jímž jsou postavena na roveň díla vyjádřená způsobem obdobným fotografií; díla užitého umění; ilustrace, zeměpisné mapy; plány, návrhy a plastická díla zeměpisná, místopisná, architektonická nebo vědecká.

29 SAS Institute v World Programming [2010] EWHC 1829 (Ch) / Case C-406/10.

(5) Sborníky literárních nebo uměleckých děl jako jsou encyklopedie a antologie, které způsobem výběru a uspořádáním obsahu jsou duševními výtvořiny, jsou samostatně chráněny bez újmy autorského práva ke každému z děl, která tvoří části takových sborníků.³⁰

Aby bylo možné aplikovat BÚ na tento případ, je potřeba zohlednit specifika počítačových programů. K tomuto účelu byla přijata dohoda o obchodních aspektech práv k duševnímu vlastnictví Světové obchodní organizace (dále jen TRIPS). Ta v článcích 9 a 10 upravuje rozsah ochrany poskytované počítačovým programům a dále se odkazuje na BÚ:

Článek 9:

1. Členové se přizpůsobí článkům 1 až 21 Bernské úmluvy (1971) a příloze k ní. Členové však nebudou mít práva nebo povinnosti podle této Dohody, pokud jde o práva udělená podle článku 6 bis uvedené Úmluvy nebo práva z něho odvozená.
2. Autorskoprávní ochrana nebude poskytnuta myšlenkám, postupům, výrobním metodám nebo matematickým pojmům jako takovým, nýbrž jejich vyjádření.

Článek 10:

1. Počítačové programy, ať již ve zdrojovém nebo strojovém kódu, budou chráněny jako literární díla podle Bernské úmluvy (1971).
2. Sestavování údajů nebo jiných materiálů, ať schopné zpracování počítačem nebo v jiné formě, které buď z důvodu výběru nebo uspořádání jejich obsahu představují duševní výtvořiny, bude chráněno jako takové. Tato ochrana, která nebude rozšířena na samotné údaje nebo materiály, nebude na újmu jakéhokoli autorského práva, spočívajícího v samotných údajích nebo materiálech.³¹

WIPOCT zachází s počítačovými programy obdobným způsobem jako výše uvedená dohoda TRIPS, tj. definuje počítačový program jako autorsky chráněné dílo a dále vymezuje rozsah ochrany na záznam programu v jakékoli formě, nikoli na principy a myšlenky, na kterých program staví. Následně se rovněž odkazuje na BÚ.

Čtvrtým zásadním právním předpisem věnujícím se tématu ochrany počítačových programů je EUSD, která kromě stejné koncepce autorskoprávní ochrany přidává i další upřesnění. Tím je odepření ochrany nejen myšlenkám a zásadám samotného programu, ale i rozhraní programu. Toto rozšíření se týká i hodnocení výstupů programu WPS, které byly rovněž napadány, pro porušení autorských práv.

Během řízení pak bylo nutné vyřešit, kromě výše uvedených tvrzení, i jiné sporné body, na jejichž zhodnocení závisela aplikovatelnost právní úpravy. Mezi ně patřil také fakt, zda syntaxe příkazů SAS Language naplňuje znaky programovacího jazyka. Zástupci SAS Inc. se totiž snažili SAS Language popsat jako doménově specifický jazyk a uplatňovat na něj autorskou

30 Bernská úmluva o ochraně literárních a uměleckých děl. *Sbirka zákonů 133/1980*, částka 32, s. 604-605. [online]. Dostupné z: <<http://aplikace.mvcr.cz/archiv2008/sbirka/1980/sb32-80.pdf>>.

31 Příloha 1C Dohody o zřízení Světové obchodní organizace (WTO) – TRIPS. *Sdělení MZV č. 191/1995 Sb.* str. 367-396. [online]. Dostupné z: <<http://www.autorskepravo.cz/soubory/1199281460.doc>>

ochranu. Nám může jako vodítko také posloužit např. definice programovacího jazyka z publikace Moderní slovník softwaru: „Umělý jazyk určený pro zápis počítačového programu. Syntaxe a sémantika jsou přesně definovány“³². Navzdory snahám zástupců společnosti SAS, ukazuje syntaxe a stavební prvky SAS Language, že jej lze bez obav považovat za programovací jazyk.

Po ujasnění tohoto faktu soudce Arnold odložil další řešení problematiky ochrany programovacích jazyků s tím, že bude potřeba vznést dotaz k ESD s jeho určením, zda jsou podle článku 1 odst. 2 EUSD programovací jazyky vyjmuty z autorskoprávní ochrany. K této otázce se také již vyjádřil soudce Pumfrey J. v případě *Navitaire v. EasyJet*³³, aniž by se dotazoval ESD. Já mám za to, že informace na tuto otázku je dostatečně zodpovězena i v 11. bodu odůvodnění EUSD, které výslovně uvádí: *V souladu s touto zásadou autorského práva se ochrana podle této směrnice nevztahuje na myšlenky a zásady, na kterých je založena logika, algoritmy a programovací jazyky.*³⁴ Obdobně se k této otázce vyjadřuje i generální advokát ESD Yves Bot, který ve svém stanovisku přirovnává programovací jazyk k jazyku přirozenému a poukazuje na to, že se jedná o prostředek k vyjádření myšlenek, nikoli vyjádření samotné³⁵. ESD se ve svém rozsudku³⁶ držela stanoviska generálního advokáta, kdy senát usoudil, že programovací jazyk není formou vyjádření programu, který tento jazyk zpracovává, a tudíž nemůže požívat ochrany podle článku 1 odst. 2 EUSD.

V dalším kroku bylo potřeba určit relevantní zdroje informací pro tvorbu WPS, které mají vliv na posouzení bodů žaloby. V první řadě je potřeba uvést, že při tvorbě WPS neměli zaměstnanci společnosti WPL přístup ke zdrojovým kódům programu SAS. Na tomto faktu se shodly obě strany sporu. Dle tvůrců WPS byla jeho tvorba postavena zejména na:

1. studiu SAS manuálů
2. studiu výsledků vygenerovaných studijní verzí SAS
3. zákaznických referencí ohledně výsledků vygenerovaných plnou verzí SAS
4. názorech zkušených uživatelů programu SAS
5. znalostech a publikacích z oboru statistiky

32 VITOVSKÝ, A. *Moderní slovník softwaru : vykladový anglicko-český a česko-anglický*. Vyd. 1. Praha : Antonín Vitovský - AV software, 2006. 588 s. ISBN 8090142885.

33 Rozhodnutí Chancery Division ze dne 30. 7. 2004 ve v2ci *Navitaire Inc v Easyjet Airline Co. & Anor*, sp. zn. [2004] EWHC 1725 (Ch) (30 July 2004). [online]. Dostupné z: <<http://www.bailii.org/ew/cases/EWHC/Ch/2004/1725.html>>.

34 Směrnice Evropského parlamentu a Rady 2009/24/EC ze dne 23. dubna 2009 o právní ochraně počítačových programů.

35 Stanovisko generálního advokáta Yvese Bota přednesené dne 29. listopadu 2011 Věc C 406/10 *SAS Institute Inc. proti World Programming Ltd.* [online]. [cit. 2012-02-10]. Dostupné z: <<http://curia.europa.eu/juris/document/document.jsf?docid=115484&pageIndex=0&doclang=CS&mode=lst&dir=&occ=first&cid=1088404>>.

36 Rozsudek soudního dvora (Velkého senátu) ze dne 2. května 2012 (žádost o rozhodnutí o předběžné otázce *High Court of Justice (Chancery Division) - Spojené království - SAS Institute Inc. v. World Programming Ltd (Věc C-406/10)*). [online]. [cit. 2012-05-20]. Dostupné z: <<http://curia.europa.eu/juris/document/document.jsf?text=&docid=122362&pageIndex=0&doclang=CS&mode=req&dir=&occ=first&part=1&cid=154228>>.

6. znalostech technologických platforem, na kterých měl být WPS provozován
7. znalostech programů, se kterými SAS komunikuje³⁷

Na základě žalobních nároků se zdá být nejdůležitějším použitím manuálů SAS a studium fungování studijní verze SAS. Zástupci SAS uvedli množství příkladů, které ukazují shody mezi manuály SAS a samotným kódem WPS. Mezi ně patřily, z mého pohledu, poměrně banální, jako například použití stejných jmen procedur a funkcí. Je samozřejmě pravda, že takto přílišná shoda se SAS manuály nebyla nezbytná. Nemyslím si však, že to má z pohledu autorské ochrany programu SAS valný význam.

Je však potřeba zdůraznit, že už kvůli samotnému důvodu vzniku WPS, tedy aby se stal možným ekvivalentem k SAS s možností použití SAS skriptů, bylo nutné zajistit stejné chování programu WPS jako SAS. Příkladem budiž jeden z argumentů zástupců SAS, kterým se snažili dokázat dle jejich slov otročké kopírování manuálů SAS do WPS, aniž by zkopírovanému objektu porozuměli, popřípadě i když věděli o lepším řešení. V uvedeném případě se jednalo o zpracování datumů před rokem 1900. Volně přeložený komentář v kódu WPS hovoří o zvláštní transformaci datumů před rokem 1900, což je zřejmě chyba programu SAS, ovšem protože programátoři neví co s tím, zkopírují toto chybné chování. S přihlédnutím k důvodu vzniku WPS je však jasné, že důvodem takového postupu byla snaha o maximální zachování výsledků shodných s výsledky podávány programem SAS. Pokud by přistoupili k optimalizační procesů a opravě chyb, je možné, že výsledný produkt by byl pokročilejší než SAS, nicméně by nemohl sloužit původnímu účelu jako náhrada za SAS pro zpracování SAS skriptů.

Argument potřeby stejného chování se uplatní i v případě většiny ostatních podobností manuálů SAS a programu WPS. Týká se to i stejné volby výchozích parametrů, které se použijí pro výpočet, pokud uživatel tyto parametry nedefinuje. Tvůrci WPS ani neměli jinou možnost, než použít stejné výchozí hodnoty, přesto tento fakt zástupci SAS představují jako zneužití jejich manuálů.

Trochu složitější je situace v případě výstupních logů. Na samotnou funkci programu rozhodně nemá vliv úprava komentářů a výsledků operací. Může to mít ovšem vliv na interoperabilitu s jinými programy, které očekávají výstup v určitém formátu. Pak by bylo možné log programu považovat i za rozhraní, na které se autorskoprávní ochrana nevztahuje. Dle mého názoru je pak potřeba rozlišovat, zda je účelem posuzovaného grafického prvku zajistit možnost interoperability s jinými programy, či zda má pouze evokovat vzhled původního programu. Podle těchto kritérií je pak možné rozhodnout, zda došlo k porušení autorských práv nebo ne.

Další výhrady zástupců SAS směřovaly k použití datového formátu SAS7BDAT. Ty byly postaveny na tvrzení, že uvedený datový formát je definován ve zdro-

37 *SAS Institute v World Programming* [2010] EWHC 1829 (Ch) / Case C-406/10.

jovém kódu programu SAS, a že vytvořením funkcí a procedur ve WPS došlo k porušení autorského práva. Dále se také zástupci SAS odvolávali na formulaci předmětu ochrany v EUSD, a to konkrétně na článek 1 odstavec 2, kde je uvedeno doslova:

„*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.*“³⁸

Dle zástupců SAS, podle tohoto ustanovení nelze generalizovat, že pokud nejsou chráněny myšlenky a principy, na kterých je založeno rozhraní, automaticky je právní ochrana odepřena také samotnému vyjádření rozhraní. S tímto dle mého názoru nelze souhlasit. Rozhraní programu má co do ochrany nesrovnatelně slabší pozici, než samotný program. To dokládá i legální možnost dekompilece stanovená v článku 6 EUSD. Zároveň lze záměry tvůrců EUSD zjistit i z původního návrhu³⁹ k předchůdci EUSD, tj. ke směrnici Rady 91/250/EHS. V tomto návrhu se v sekci 3.7 důvodové zprávy výslovně uvádí, že ochrana je poskytována pouze individuálnímu vyjádření myšlenky a dává tím volnost dalším autorům vytvářet podobné nebo dokonce stejné programy, za předpokladu, že se zdrží kopírování kódu. Tato myšlenka je rozvedena i v sekci 3.12, kde je dána konkurentům volnost ve vytváření dalších programů, které staví na myšlenkách a principech zjištěných analýzou daného programu. Na myšlenku jako takovou nemůže být monopol, pouze bude chráněno vyjádření myšlenky ve formě programu. Dále se sekce 3.13 zmiňuje i o ochraně kódu programu v případě, že kvůli limitaci rozhraní není možné vytvořit další program s odlišnou implementací přístupu k tomuto rozhraní. V takovém případě by nešlo o porušení autorských práv, protože by došlo k tzv. *splynutí myšlenky a jejího vyjádření*⁴⁰. Co se prvního tvrzení týče, tak není nezbytně pravda, že datový formát musí být definován ve zdrojovém kódu SAS, ani toto tvrzení nebylo doloženo částí kódu, který by tento fakt prokázal. Co v kódu být musí, jsou na druhou stranu pouze metody a funkce, které zajišťují zápis do souboru ve formátu SAS7BDAT. A i v případě, že by tyto funkce byly příliš podobné těm v programu SAS, tak i z výše uvedeného vyplývá, že by se nemělo jednat o porušení autorských práv.

Soudce Arnold však nenabyl přesvědčení, že otázka právní ochrany rozhraní je ve světle EUSD jednoznačná, a proto se obrátil na ESD i s touto otázkou. Na tuto otázku se Yves Bot vyjádřil poněkud nepřesně, neboť se zaobírá otázkou možnosti dekompilece rozhraní, ačkoli v otázce položené soudcem Arnoldem bylo jasné stanoveno, že k dekompilaci nedošlo. Ve svém stanovisku se přiklonil na stranu zástupců SAS, když souhlasil, že vyjádření rozhraní může být poskytnuta ochrana podle

EUSD, ovšem bude na soudci, aby posoudil, zda tato implementace rozhraní tvoří podstatnou část díla a má tudíž požívat autorskoprávní ochrany. Další části odpovědi na tuto otázku jsou pro případ irelevantní, protože se zabývají možnou protiprávností dekompilece, ke které nedošlo. ESD však ve svém rozsudku⁴¹ jednoznačně stanovil, že formát datových vstupů nepředstavuje formu vyjádření programu, která by požívala ochrany podle EUSD. Dále pak i rozvádí myšlenku generálního advokáta pro případ, že by si určitá osoba obstarala záznam zdrojového nebo strojového kódu určeného pro obsluhu rozhraní, a tento kód zkopírovala do svého programu. Pak by záleželo na posouzení, zda se jednalo o podstatnou část díla, a z toho následně vyplývající možnost ochrany podle EUSD.

Jednou z nejzásadnějších otázek v celém sporu je však rozhodně možná protiprávnost vytvoření programu, který kopíruje funkcionalitu programu jiného. Zástupci SAS v tomto ohledu poukazovali na to, že k porušení autorských práv k programu SAS došlo vytvořením některých programových modulů WPL, které byly naprogramovány podle manuálů SAS. Pokud na program nahlédneme jako na literární dílo, bude autorskoprávní ochrana dle EUSD garantována pouze zdrojovým kódům programu, veškerým jeho přípravným materiálům a strojovému kódu. Ten sice není pro člověka čitelný, nicméně dekompilací z něj lze získat jednotlivé metody a funkce programu v lidsky čitelné podobě. V souvislosti s tímto se zástupci SAS odvolávali i na důvodovou zprávu návrhu směrnice Rady 91/250/EHS, která zmiňuje i originalitu díla jako jedno z kritérií pro získání ochrany podle této směrnice. Nabyl jsem však dojmu, že zástupci SAS si zde mylně vyložili onu originalitu díla jako označení, že daný program je originální co do tématu, které zpracovává. Avšak originalita zmiňovaná v důvodové zprávě směrnice Rady 91/250/EHS by se podle všeho měla projevit v tom, jak je daný program napsán, jak jsou jednotlivé součásti pospojovány do fungujícího celku či jaké optimalizace byly v programu použity. To je také důvod, proč byly zaváděny testy jako Abstraction-Filtration-Comparison, který zobrazí porovnávané zdrojové kódy jako logické celky, na kterých lze onu originalitu porovnat. Otázka originality by tedy byla na místě v případě, že by došlo ke zkopírování zdrojových kódů společnosti WPL.

Zástupci SAS také předložili argument, podle kterého může dojít k porušení autorských práv při zkopírování zápletky literárního díla, aniž by byl v novém díle použita jediná věta z originálu. Soudce Arnold však správně odmítl toto přirovnání funkcí programu k zápletce literárního díla, neboť již sama zápletka je vyjádřením myšlenek autora. Naproti tomu funkce programu se stávají vyjádřením až ve chvíli jejich implementace.

38 Směrnice Evropského parlamentu a Rady 2009/24/ES: o právní ochraně počítačových programů. In: L 111/16. 2009.

39 Proposal for a Council Directive on the legal protection of computer programs. COM (88) 816 final, 17. 3. 1989. [online]. [cit. 2011-12-17]. Dostupné z: <[http://aei.pitt.edu/13138/1/COM_\(88\)_816_final.pdf](http://aei.pitt.edu/13138/1/COM_(88)_816_final.pdf)>.

40 Proposal for a Council Directive on the legal protection of computer programs. COM (88) 816 final, 17. 3. 1989. [online]. [cit. 2011-12-17]. Dostupné z: <[http://aei.pitt.edu/13138/1/COM_\(88\)_816_final.pdf](http://aei.pitt.edu/13138/1/COM_(88)_816_final.pdf)>.

41 Rozsudek soudního dvora (Velkého senátu) ze dne 2. května 2012 (žádost o rozhodnutí o předběžné otázce High Court of Justice (Chancery Division) - Spojené království) - SAS Institute Inc. v. World Programming Ltd (Věc C-406/10). [online]. [cit. 2012-05-20]. Dostupné z: <<http://curia.europa.eu/juris/document/document.jsf?text=&docid=122362&pageIndex=0&doclang=CS&mode=req&dir=&occ=first&part=1&cid=154228>>.

Na základě všech dostupných informací však soudce Arnold rozhodl, že řešení této otázky není jednoznačné a obrátil se na ESD s otázkou, zda se jedná o porušení autorského práva k prvnímu programu, pokud soutěžitel vytvoří jiný program, který rozmnožuje funkce programu prvního. Důležitým znakem je, že při vytváření druhého programu neměl soutěžitel přístup ke zdrojovým kódům programu prvního, a to ani přímo, ani díky jinému postupu jako např. dekompilací.

Generální advokát se ve svém stanovisku⁴² rovněž zaobírá originalitou díla a dále tuto myšlenku rozvádí, že tato originalita představuje jakési know-how, se kterým jsou jednotlivé komponenty programu zkompletovány do funkčního celku. Také jasně uvádí, že podle něj nemohou být funkce programu předmětem autorskoprávní ochrany, přičemž se také odvolává na zde již uvedené body důvodové zprávy návrhu směrnice Rady 91/250/EHS. Dle mého názoru je pak nejzásadnější větou celého posudku:

„Připustit, že by funkce počítačového programu jako taková mohla být předmětem ochrany, by znamenalo umožnit monopolizaci myšlenek, a to na úkor technického pokroku a průmyslového rozvoje.“⁴³

ESD následně judikovala v souladu s tímto stanoviskem, že funkce programu autorskoprávních ochrany podle EUSD požívat nemohou.

Další průběh sporu se týkal licence programu SAS použité pro zkoumání principů a následný vývoj programu WPS. Jak je výše uvedeno, společnost WPL použila pro zkoumání učební edice programu SAS. Její instalace je podmíněna odsouhlasením licenčních podmínek, tzv. click wrap licencí, které s sebou nesou určitá omezení. Kromě technických omezení týkajících se limitů v počtu zpracovaných prvků, je tato edice označena slovy „pro osobní výuku“ a „pouze k nekomerčním účelům“, je nepřenosná a je omezena na jednoho uživatele, který odsouhlasil licenční ujednání. Výhodou, která převáží uvedené nedostatky, je jednoznačně cena. Ta se pohybuje ve zlomcích ceny plné edice.

Prvním problémem použité licence bylo rozhodnutí, zda bude za zákazníka ve vztahu k SAS považován ten konkrétní uživatel, který odsouhlasil licenční ujednání, anebo firma, která danou kopii zakoupila a předala ji k užívání svému zaměstnanci. Pro první variantu hovoří úvodní část licenčních podmínek, která za zákazníka označuje jednotlivce, který odsouhlasil licenční podmínky kliknutím na patřičné tlačítko. Pro druhou verzi hovoří sekce 1.1 licenčních podmínek SAS, která uvádí, že zákazník získává uživatelská práva za daných podmínek k programu SAS výměnou za platbu všech poplatků. V případě, kdy je plátcem zaměstnavatel uživatele, který licenční ujednání odsouhlasil, by tato formulace naváděla k postavení onoho zaměstnavatele do pozice zákazníka ve vztahu se SAS. Ovšem dle této logiky by zaměstnavatel mohl zakoupit pouze jednu

licenci programu, na které by se podle jeho pokynů střídalo více jeho zaměstnanců. To by bylo v přímém rozporu se specifickými učitelskými edicemi, které výslovně omezují počet uživatelů na jednoho a zakazují přenosnost licence na jiné uživatele. Proto lze považovat za pravděpodobnější, že zákazníkem je myšlen jednotlivec, který odsouhlasil licenční podmínky. Na základě tohoto faktu zřejmě došlo k porušení smluvních podmínek ze strany WPL, protože učební edice byla využívána větším množstvím jejích zaměstnanců.

Druhým sporným bodem týkajícím se učební edice bylo její využití při zkoumání výstupů SAS a jejich porovnávání s WPS. Podle zástupců SAS je v tomto bodě relevantní zaměření licence na nekomerční účely, což je v rozporu s účelem vytvoření konkurenčního produktu. Stejně jako zástupci společnosti WPL si i já myslím, že komerční účel je v tomto případě myšlen jinak. Toto omezení má zabránit použití učební licence pro komerční zpracování uživatelských dat. Tomu má samo o sobě zabránit i omezení maximálního množství zpracovávaných prvků. Avšak pro výpočty na menších vzorcích by se toto omezení nemuselo uplatnit, a proto bylo pro učební edici zakázáno komerční využití smluvně.

Soudce Arnold sumarizoval způsoby využití učební edice SAS zaměstnanci WPL, aby následně mohl posoudit, zda spadaly do rozsahu licenčních podmínek. Využití bylo následovné:

1. zjištění způsobů fungování SAS, opakovaně porovnávání výstupů SAS s výstupy WPS,
2. porovnání výkonnosti za účelem zlepšení efektivity WPS,
3. verifikace výsledků WPS oproti výsledkům SAS,
4. zkoumání logů SAS,
5. zkoumání datového formátu SAS7BDAT,
6. generování seznamu poštovních směrovacích čísel, který byl následně přidán do WPS.⁴⁴

Toto využití soudce Arnold hodnotil jako překročení rozsahu licence. Já se s jeho názorem nemohu ztotožnit, protože dle mého mínění licence pouze specifikuje, jak s programem může uživatel nakládat, nikoli však za jakým účelem. Kromě posledního bodu se ve všech ostatních bodech jednalo o běžnou činnost s programem, tj. jeho spouštění, zavádění do paměti, poskytnutí vstupních dat a analýza výstupu. Samotný účel, tedy zkoumání funkcí programu, je pak garantován v článku 5 odst. 3 EUSD, který výslovně uvádí:

„Osoba oprávněná užívat rozmnoženinu počítačového programu může při zavádění, zobrazování, provozu, přenosu nebo ukládání do paměti počítačového programu, k němuž 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.“⁴⁵

Uvedený článek EUSD byl také hlavním argumentem zástupců WPL. Soudce Arnold tuto argumentaci

⁴² Stanovisko generálního advokáta Yvese Bota přednesené dne 29. listopadu 2011 Věc C 406/10 SAS Institute Inc. proti World Programming Ltd. [online]. [cit. 2012-02-10]. Dostupné z: <<http://curia.europa.eu/juris/document/document.jsf?docid=115484&pageIndex=0&doclang=CS&mode=lst&dir=&occ=first&cid=1088404>>

⁴³ tamtéž

⁴⁴ SAS Institute v World Programming [2010] EWHC 1829 (Ch) / Case C-406/10.

⁴⁵ Směrnice Evropského parlamentu a Rady 2009/24/ES: o právní ochraně počítačových programů. In: L 111/16. 2009.

akceptoval a byl jí i nakloněn, ovšem cítil se povinen rovněž tuto otázku předložit ESD. Stanovisko generálního advokáta na tuto otázku je opět poněkud nejasné, neboť se uchyluje k posouzení možnosti přístupu uživatele ke zdrojovým kódům programu na základě využití práv uvedených v článku 5 odst. 3 EUSD. Senát ESD však koriguje toto stanovisko a zabývá se samotným využitím práva na zkoumání principů a zásad, na kterých je program založen. Ve svém rozhodnutí ESD poukazuje i na důvodovou zprávu EUSD, ve které je uvedeno, že oprávněnému uživateli nesmí být bráněno zkoumat program, pokud tím neporušuje autorská práva k danému programu. Porušením práv tedy rozhodně není zavádění do paměti a spouštění testovacích skriptů. ESD také uvádí, že nelze uživateli upírat právo zkoumat principy a funkce programu s odvoláním na licenční ujednání. S ohledem na výše uvedené jsem tedy přesvědčen, že licenční ujednání omezující např. komerční využití programu nemají možnost zabránit uživateli ve zkoumání jejich funkcí a principů, na kterých stojí.

V závěru tohoto sporu byly řešeny otázky porušení autorských práv k manuálům a příručkám WPS. Na základě porovnání manuálů SAS a WPS došel soudce Arnold k závěru, že v tomto případě k porušení autorských práv došlo. Příručky WPS obsahovaly pouze kompilaci klíčových slov jazyka SAS s poznámkou, zda je daná funkce v programu WPS podporována či ne. V tomto případě bylo rozhodnuto, že k porušení práv nedošlo.

Celé řízení bylo přerušeno kvůli předložení některých fundamentálních otázek ESD. Teď, když jsou známy odpovědi, bude potřeba vyčkat až anglický soud vynese finální verdikt. Nic však nenasvědčuje tomu, že se měl nějak lišit od rozhodnutí vydaným ESD. Názory soudce Arnolda se v podstatných záležitostech shodovaly s rozhodnutím ESD, a předložené otázky byly důležité pro potvrzení jeho úvah a pro minimalizaci dalších průtahů v řízení při případném odvolání některé ze stran.

Problém společnosti WPL představuje zejména přílišná inspirace manuály SAS při tvorbě vlastních manuálů k programu WPS. Já osobně totiž v jejich počínání nikde jinde nespátřuji rozpor s právní úpravou autorské ochrany počítačových programů. Ke kopírování zdrojového kódu SAS nedošlo a ani z podstaty věci dojít nemohlo, program SAS nebyl dekompileován a ani jiným způsobem se programátoři společnosti WPL ke zdrojovým kódům SAS nedostali. Program WPS tedy představuje výsledek jejich vlastních schopností a práce, který je sám hoden autorskoprávní ochrany.

6. Legální způsoby provedení zpětné analýzy

Ve světle skutečností uvedených v této práci se v této kapitole budu zabývat praktickými možnostmi dekompileace počítačových programů. Nemám tím na mysli technický postup a prostředky, které jsou při této činnosti potřebné, ale spíše rozbor situací a podmínek, které

musí nastat, aby nebyla dekompileace právně postižitelná.

Pro přehlednost budu popisovat fiktivní postavu programátora, který má k dispozici program A, s nímž bude provádět určité operace. Veškerou činnost s programem budu nejdříve popisovat dle amerického práva a následně dle evropského, konkretizovaného v české implementaci komunitárního práva. Podmínkou pro jakoukoli manipulaci s programem je jeho legální pořízení. Toto je základní prerekvizita, pokud se má programátorovi dostat jakékoli ochrany. Dále je potřeba si pečlivě pročíst EULA, pokud je u programu přítomna a zejména je-li podmínkou pro samotnou instalaci programu. Nyní již projdu samotné zacházení s programem dle různých právních řádů.

6.1 Americký model

Pokud se tedy programátor rozhodne, že bude provádět jakékoli operace s programem A, bude muset nejdříve zjistit, zda je tento program chráněn pomocí DRM. Pokud nikoli, nebude se muset zabývat problematikou řešenou v USC:T17 § 1201 a následující.

Dalším krokem bude podrobná kontrola ustanovení uvedených EULA, pokud je u programu přítomna a zejména pokud je nutný souhlas při instalaci programu A. Jestliže by EULA obsahovala velmi restriktivní ustanovení, mohlo by to samozřejmě velmi zúžit programátorovy možnosti.

Je-li cílem programátora vytvořit konkurenční program B a EULA programu A zakazuje dekompileaci, pak může programátor zvolit dvě varianty řešení. První možností je pro zkoumání programu dekompileaci použít s vědomím, že se dostane do sporu s EULA. Během případného sporu by pak musel argumentovat zásadou použití fair use tak, jak tomu bylo např. v případě *Sega v. Accolade*. Pro programátora příznivý výsledek takového sporu ovšem není možné s jistotou garantovat. Záleží samozřejmě i na způsobu vytváření programu B – pokud by byly některé části nápadně podobné, či přímo stejné jako v programu A, pak by mohl být odpovědný také za porušení autorských práv. Proto lze i dekompileaci v tomto případě doporučit striktně pro zjištění principů, na kterých je zkoumaný program založen a nikoli pro získání inspirace při tvorbě nového programu. V souvislosti s tím může přijít k užítku i dříve zmíněná technika *clean room*.

Druhou možností je použití zpětné analýzy formou metody *black box*. Ta jediná připadá v úvahu i v případě, že je program A chráněn pomocí DRM. Jak uvádí i Petra Heindl ve své zprávě⁴⁶, přijetí DMCA totiž vytvořilo určitý paradox. Ačkoli je možno dekompileaci bránit dle principu *fair use*, pokud je dekompileovaný program chráněn pomocí DRM, tak jeho vyřazení do tohoto principu nespadá. Podle § 1205 a následujících USC:T17 je možnost legálního vyřazení DRM striktně omezená na zákonem předvídané situace. Takže i kdyby

46 HEINDL, P. TTLF Working Papers - A Status Report from the Software Decompilation Battle: A Source of Sores for Software Copyright Owners in the United States and the European Union?[online]. 2008 [cit. 2012-06-10]. Dostupné z: <www.law.stanford.edu/program/centers/ttlf/papers/heindl_wp1.pdf>.

samotná kopie programu A vzniklá jeho dekompilací nebyla v rozporu s americkým právem, cesta k jejímu vzniku by v rozporu s DMCA již byla.

Nabízí se však otázka, jak lze vykládat ustanovení §1201 odstavec c) USC:T17, které nám říká, že zákazy definovanými v § 1201 USC:T17 nejsou dotčena ostatní práva, omezení vyplývající z autorskoprávní ochrany a možné metody obhajoby proti nařčení z porušení autorských práv, jako např. možnost využití obrany principem fair use. Buď lze dané ustanovení posuzovat tak, že právní ochrana DRM by neměla bránit ve výkonu ostatních práv garantovaných USC:T17, nebo je smyslem uvedeného ustanovení striktní oddělení nároků vyplývajících z autorských práv a jejich možného porušení a nároků vyplývajících z porušení některého z ustanovení § 1201 a následující USC:T17. Já bych se přikláněl k první verzi významu tohoto ustanovení, nicméně z americké soudní praxe spíše vyplývá, že soudy se přiklánějí k verzi druhé. Důvod pro provedení dekompilace tedy není relevantní ve vztahu k posouzení možné protiprávnosti překonání DRM.

Možnost dekompilace se v takovém případě omezuje na její využití pro získání potřebných informací o rozhraní programu A, neboť interoperabilita je jednou z výjimek pro vyřazení DRM za předpokladu, že potřebné informace nebyly již dříve zveřejněny.

Pokud by se nejednalo o snahu programátora vytvořit srovnatelný produkt, ale našel by v programu A chybu, pak opět neposkytuje DMCA programátoru žádnou explicitní výjimku, kterou by mu garantovala, že vyřazení DRM nebude právně postižitelné. Nicméně pokud programátor narazí v programu A na chybu, poskytne mu to dostatek důvodů pochybovat o bezpečnosti takového programu. Tím se dostává do oblasti, kterou DMCA již předvídá, a tou je bezpečnostní testování. S tímto odůvodněním je možné DRM programu vyřadit a dále se zabývat samotným programem. Pokud chybu programátor nalezne, může ji i opravit, ovšem informace získané při testování a opravě chyb by neměly být následně využity pro tvorbu samostatného programu.

Dá se tedy říci, že před příchodem DMCA bylo americké právo poměrně liberální k využití zpětné analýzy i ve formě dekompilace. DMCA ovšem nastavila mnohem restriktivnější podmínky, které celé odvětví doslova svazují. Z právního hlediska je tedy v případě použití DRM jedinou bezpečnou formou zpětné analýzy pouze black box analýza.

6.2 Evropský model

Pokud hypotetického programátora prohlásíme za občana České Republiky, řešení výše naznačených situací se bude mírně lišit. Projeví se tak evropská právní úprava implementovaná v českém právním řádu.

Pokud má programátor legálně získaný program A a chystá se podle něj vytvořit program B, musí si nejprve ujasnit, jaký typ zpětné analýzy by rád použil. Pokud se rozhodne pouze pro black box zpětnou analýzu, nic mu nebrání ve zkoumání programu A a zjiš-

ťování principů, na kterých je založen. Situace se však komplikuje ve chvíli, kdy chce programátor využít techniku dekompilace. Zde není primárním určujícím prvkem ani smluvní ustanovení nutné pro instalaci programu A, ani jeho případná ochrana pomocí DRM, neboť evropské právo garantuje minimální standard, které má oprávněný uživatel při zacházení s programem. Jakékoli smluvní ustanovení či DRM tuto základní sadu oprávnění nemůže snížit. Pro lepší možnost porovnání s americkou úpravou však budu uvažovat, že program A je chráněn pomocí DRM.

Evropské pojetí považuje vznik kopie programu při jeho dekompilaci jako automatické porušení autorských práv. Ačkoli metoda dekompilace není naprosto vyřazena z legálních způsobů zacházení s programem, je vázána na splnění explicitních podmínek. Ty jsou v českém právu definovány v § 66 AZ. Jednotlivé výjimky jsem probral v části zabírající se AZ, vytváření obdobně fungujícího programu však rozhodně mezi tyto výjimky nepatří. Situace je tedy poměrně jednoznačná – v takovém případě dekompilace rozhodně legálně použít nejde a programátor bude muset zůstat u black box techniky při odhalování funkčnosti programu A.

Sporná situace nastává při řešení rozhraní obou programů. Pokud programátor vytvoří program B, který bude mít obdobnou funkcionalitu, jako program A a případně ještě nějakou navíc, bude potřebovat, aby jeho program akceptoval stejné vstupy a generoval stejné výstupy, jako program B. Toto je v podstatě popis interoperability mezi programy. Evropská úprava dává programátorovi možnost dekompilace za účelem získání informací vedoucí k dosažení interoperability, ovšem jedním dechem dodává, že takto získané informace nesmí být použity k vývoji programu, jehož vyjádření je ve své podstatě podobné. Přiznám se, že sám si nejsem jistý přesným významem této formulace. Je tím snad myšleno, že lze dekompilovat rozhraní programů pouze za účelem tvorby nových programů, které rozšiřují jejich funkcionalitu a nikoli k tvorbě programů, které mohou ty původní nahradit? A nebo má spojení „vyjádření ve své podstatě podobné“⁴⁷ znamenat vytvoření programu, jehož zdrojový kód je více než podobný původnímu programu, který byl pod záminkou dosažení interoperability dekompilován celý? Já se přikláním k druhému významu, ovšem to by snad nemuselo být takto explicitně uváděno, neboť by takový program byl posuzován jako porušení autorského práva ať už by byl interoperabilní s originálem či nikoli. Shodou okolností v tomto výkladu může pomoci vyjádření⁴⁸ generálního advokáta ESD Yvese Bota. Ten se ve svém vyjádření zabíral odpovědí na otázku, která mu ovšem nebyla položena.

Zabíral se otázkou, kterou si kladu i já, a to zda je možné dekompilovat rozhraní programu (v případě

47 Zákon č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon). In: 121/2000. 2010.

48 Stanovisko generálního advokáta Yvese Bota přednesené dne 29. listopadu 2011 Věc C 406/10 SAS Institute Inc. proti World Programming Ltd. [online]. [cit. 2012-02-10]. Dostupné z: <<http://curia.europa.eu/juris/document/document.jsf?docid=115484&pageIndex=0&doclang=CS&mode=lst&dir=&occ=first&cid=1088404>>.

SAS v. WPL se jednalo o datový formát vstupních dat) a takto získané informace použít pro zajištění interoperability s konkurenčním programem. Ve své analýze generální advokát tvrdí, že dekompilace rozhraní je bez svolení autora možná, ale že článek 6 odstavec 1. nezakládá právo rozmnožit kód získaný dekompilací ve svém vlastním programu. Tato argumentace nahrává spíše druhému významu sousloví „vyjádření ve své podstatě podobné“, tj. pouze zákaz použití dekompilovaného kódu, ale možnost implementace rozhraní do vlastního programu, byť bude představovat konkurenci pro program původní.

Navzdory těmto šedým místům právní úpravy, i já zastávám názor, že tuto část programu A může programátor bezpečně dekompilovat a takto zjištěné informace pak využít v programu B. V tomto bodě se tak setkává americká i evropská úprava, byť každá zde došla trochu jinou cestou – ve Spojených státech je to výjimka opravňující k vyřazení DRM, v Evropě je to pak výjimka pro dekompilaci, přičemž právní ochranu DRM v takovém případě nelze uplatnit.

Pokud tedy budeme posuzovat otázku snahy o pouhé dosažení interoperability, je nutné ještě podotknout, že dekompilace je brána jako *ultima ratio*, tj. poslední možný způsob zjišťování informací. Programátor by měl tedy postupovat v následujících krocích. Nejprve se pokusí zjistit informace o rozhraní z různých veřejných zdrojů, tím mám na mysli typicky stránky se specifikacemi programu A, internetová fóra nebo informace o rozhraní, které odkryla a zveřejnila jiná osoba, která rovněž zajišťovala interoperabilní program s programem A. Pokud tento postup nevede ke kýženému výsledku, měl by se programátor obrátit přímo na autora programu A s žádostí o poskytnutí informací potřebných pro implementaci vlastního rozhraní kompatibilního s programem A. Pokud programátorovi autor programu A odpoví a poskytne potřebné informace, prakticky mu tím zavře legální cestu k dekompilaci. Pokud ale bude jeho žádost ignorovat nebo ji zamítne, tak pak může programátor konečně přistoupit k dekompilaci.

Avšak i při jejím použití si musí programátor dávat pozor, aby došlo k dekompilaci pouze těch částí programu A, které obsahují informace k rozhraní. Povolení dekompilace pro zajištění interoperability totiž není volná propustka k dekompilaci celého zkoumaného programu. Je ale také pravda, že mnohdy není možné bez přesnějších znalostí exaktně určit, které části programu mohou nést informace o jeho rozhraní. Pokud tedy při dekompilaci programátor odkryje kód, který nemá žádnou souvislost s rozhraním, pak by tuto informaci neměl nikde použít. Zde by se projevil výhody clean room techniky, tedy že zkoumající tým odkryje programátorům pouze relevantní informace a výsledný program tak nebude čelit hrozbě žaloby za porušení autorských práv.

Odlíšný americký a evropský přístup se ještě více projeví i v případě chyby programu A, kterou by si programátor rád svépomocí opravil. Ve Spojených státech je potřeba stavět obranu na teorii bezpečnostního testování a opravě chyb, aby se přes výjimky pro obejít DRM

dostal až k dekompilaci. V Evropské úpravě je tato situace trochu nejasná, neboť ke vztahu právní ochrany DRM a autorskoprávní ochrany samotných programů se staví pouze důvodová zpráva EUCP, která tomto ohledu dává přednost EUSD, ovšem v ne zcela ideálním výčtu případů. Je tedy na zákonodárství jednotlivých států, jak vyřeší právo na svépomocnou opravu chyb v programu, který je chráněn pomocí DRM. V českém právu toto řeší § 66 odstavec 8. AZ, který odejímá ochranu DRM v případě, že brání snaze o dosažení interoperability. Pro možnost uživatele k opravě chyb pak ukládá autorovi programu povinnost zpřístupnit oprávněnému uživateli dílo v rozsahu uživatelských práv stanovených v § 66 odstavec 1. AZ.

Je tedy vidět, že ani v evropském podání není možnost legální dekompilace nikterak jednoduchá. Navzdory odlišným směrům, ze kterých postupuje úprava ve Spojených státech a ta v Evropě se díky snahám o sjednocení dostaly oba právní systémy na obdobnou úroveň autorskoprávní ochrany, kterou počítačovým programům poskytují.

7. Závěr

Jak bylo již předestřeno v úvodu tohoto článku, jejím cílem bylo uvedení do problematiky zpětné analýzy počítačových programů, dopadů její aplikace na vývoj softwaru a také řešení právních otázek s ní spojených. Zpětná analýza rozhodně má své opodstatnění, zejména díky úsporám nákladů na vývoj produktů či pozitivnímu vlivu na technologický vývoj. Na druhou stranu vzhledem k nehmotné povaze softwaru s sebou nese její použití i určitá rizika z hlediska možného ohrožení zájmů chráněných autorským právem.

Praktické metody zpětné analýzy byly představeny ve druhé kapitole. Součástí tohoto rozboru bylo i posouzení možných zásahů do autorských práv. Black box analýza sice neposkytuje tak konkrétní informace o zkoumaném programu, ale vynechává nejproblematictější aspekt dekompilace, kterým je nutné vytvoření rozmnoženiny originálního díla. Dále jsem se v této kapitole zabýval metodou pro minimalizaci byť neuvědomělého rizika kopírování kódu originálního programu do programu nového.

Následující kapitola, která byla věnována normativní úpravě zpětné analýzy počítačových programů na území USA a Evropy, přináší porovnání rozdílných přístupů k řešení základní otázky možnosti použití zpětné analýzy. Důvodem pro stanovení určitých pravidel je snaha o vytvoření rovnováhy mezi zachováním autorských práv na straně jedné a práv uživatelů softwaru na straně druhé. Právní jistota a účinná možnost prosazení autorských práv dává ekonomickou motivaci pro tvorbu nových programů, ovšem příliš silné postavení autorů a možnost úplného „zamknutí“ díla, jak technickými, tak právními prostředky, by mohlo naopak vést k recesi celého trhu s počítačovými programy a tvorbě monopolů pro jednotlivé oblasti informačních technologií. Základní rozdíl v přístupu k právnímu posou-

zení dekompilece mezi USA a Evropou tkví v tom, že v USA lze použít dekompilaci a domoci se účinné obrany proti obvinění z nedovoleného rozmnožování díla, ale v Evropě je na dekompilaci automaticky nahlíženo jako na akt porušení autorských práv a existují pouze taxativně vymezené podmínky pro její provedení.

Je ovšem nutné podotknout, že tato rovnováha byla poměrně zásadně narušena neospravedlnitelně vysokou právní ochranou DRM. Tato ochrana tak může snadno vést ke zneužívání autorských práv na úkor technického pokroku a hospodářské soutěže, a ani zákonná garance určitých uživatelských práv bez ohledu na formulace EULA licencí na tomto faktu nic nezmění.

Toto porušení rovnováhy považují za jeden z problémů aktuální právní úpravy. Zejména v evropské legislativě by prospělo jednoznačné určení vztahů mezi autorskoprávní ochranou počítačových programů a právní ochranou DRM. Další návrhy možného vývoje práva v této oblasti jsou předestřeny ve čtvrté části třetí kapitoly.

Praktickou částí tohoto článku je vlastní rozbor případu SAS Institute v. World Programming Limited, který přinesl do problematiky zpětné analýzy natolik zásadní otázky, že jejich řešení muselo být přeneseno na Evropský soudní dvůr. Jak bohužel ukázal tento případ, tak ani perfektní formulace otázky není zárukou jasněho stanoviska ESD. V souvislosti s tímto případem šestá kapitola ukazuje aktuální možnosti zpětné analýzy s důrazem na podmínky, které jsou dány normativními předpisy. Opět je prováděna komparace mezi americkými a evropskými limity, které by neměl programátor překročit, chystá-li se zpětnou analýzu provádět.

Důležitou úlohu v rychle se rozvíjícím oboru informačních technologií bude představovat i budoucí judikatura, která teprve osvětlí některé sporné body v této oblasti. Díky tomu se zde otvírají široké možnosti pro další studium.

8. Seznam použité literatury

8.1 Monografie

Software protection - a comparative perspective. Editor Josef Donát, Martin Maisner, Radim Polčák. München: Medien und Recht Verlag, 2011, xii, 266 s. ISBN 978-393-9438-151.

VITOVSKÝ, A. Moderní slovník softwaru : výkladový anglicko-český a česko-anglický. Vyd. 1. Praha : Antonín Vitovský - AV software, 2006. 588 s. ISBN 8090142885.

TELEC, I., TŮMA, P. Autorský zákon. Komentář. 1. vydání. Praha: C. H. Beck, 2007.

8.2 Ostatní zdroje

ABBOT, J. Reverse Engineering of Software: Copyright and Interoperability. In: Journal of Law and Information Science Vol. 14 [online]. 2003 [cit. 2012-02-07]. s.7 – 49. Dostupné z: <dataabáze HeinOnline.org>.

ANDREWS, J. A. Reversing Copyright Misuse: Enforcing Contractual Prohibitions on Software Reverse Engineering. In: Houston Law Review, Vol. 41, Issue 3 [online]. 2004 [cit. 2011-10-07]. s. 975 – 1016. Dostupné z: <dataabáze HeinOnline.org>.

BRESSMAN, S. Restricting Reverse Engineering through Shrink-Wrap Licenses: Bowers v. Baystate Technologies, Inc. In: Boston University Journal of Science & Technology Law, Vol. 9, Issue 1 [online]. 2003 [cit. 2011-11-07]. s. 185 – 193. Dostupné z: <dataabáze HeinOnline.org>.

HEINDL, P. TTTF Working Papers - A Status Report from the Software Decompilation Battle: A Source of Sores for Software Copyright Owners in the United States and the European Union? [online]. 2008 [cit. 2012-06-10]. Dostupné z: <www.law.stanford.edu/program/centers/ttlf/papers/heindl_wp1.pdf >.

JOLISH, B. D. Rescuing reverse engineering. In: 14 Santa Clara Computer & High Technology Law Journal [online]. 1998 [cit. 2011-10-15]. s. 509 – 513. Dostupné z: <dataabáze HeinOnline.org>.

JOHNSON-LAIRD, A. Software Reverse Engineering in the Real World. In: University of Dayton Law Review, Vol. 19 [online]. 1994 [cit. 2011-10-07]. s. 843 – 902. Dostupné z: <dataabáze HeinOnline.org>.

MCMANIS, C. R. Intellectual Property Protection and Reverse Engineering of Computer Programs in the United State and the European Community. In: High Technology Law Journal, Vol. 8, Issue 1 [online]. 1993 [cit. 2011-10-07]. s. 25 – 100. Dostupné z: <dataabáze HeinOnline.org>.

MORRIS, E.M.N. Will Shrinkwrap Suffocate Fair Use. In: Santa Clara Computer & High Technology Law Journal, Vol. 23, Issue 2 [online]. 2006 [cit. 2011-10-07]. s. 237 – 270. Dostupné z: <dataabáze HeinOnline.org>.

MOY, R. A Case against Software Patents In: Santa Clara Computer & High Technology Law Journal, Vol. 17, Issue 1 [online]. 2000 [cit. 2012-05-12]. Dostupné z: <http://pop.chtlj.org/sites/default/files/media/articles/v017/v017.i1.Moy.pdf>.

OLSEN, S. „Cat“ scanning device may track users online. [online]. [cit. 2012-05-12]. Dostupné z: <http://news.cnet.com/2100-1023-246008.html&tag=st.ne.1002.tgif.ni>.

SAMUELSON, P., VINJE, T. C., CORNISH, W. R. Does Copyright Protection Under the EU Software Directive Extend to Computer Program Behaviour, Languages and Interfaces?. European Intellectual Property Review [online]. 2012 [cit. 2012-02-03]. Dostupné z: <http://ssrn.com/abstract=1974890>.

SPOOR, J. H. Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive. In: University of Dayton Law Review, Vol. 19 [online]. 1993 [cit. 2011-10-07]. s. 1063 – 1086. Dostupné z: <dataabáze HeinOnline.org>.

ZIEMINSKI, C. Game over for Reverse Engineering: How the DMCA and Contracts Have Affected Innovation. In: Journal of Technology Law & Policy, Vol. 13, Issue 2 [online]. 2008 [cit. 2011-10-07]. s. 289 – 340. Dostupné z: <dataabáze HeinOnline.org>.

Microsoft Software License Terms (MSLT) for Microsoft Office 2010. [online]. [cit. 2012-05-07]. Dostupné z: <http://www.microsoft.com/downloads/cs-cz/details.aspx?displaylang=cs&FamilyID=2b37ad5e-dc64-400b-a00d-88d-982292b078&CTT=5&origin=HA101817777>.

8.3 Právní předpisy

Bernská úmluva o ochraně literárních a uměleckých děl. Sbírka zákonů 133/1980, částka 32, s. 604-605. [online]. Dostupné z: <<http://aplikace.mvcr.cz/archiv2008/sbirka/1980/sb32-80.pdf>>.

Dohoda o zřízení Světové obchodní organizace (WTO). Sdělení MZV č. 191/1995 Sb. str. 367-396. [online]. Dostupné z: <<http://www.autorskepravo.cz/soubory/1199281460.doc>>.

Dohoda o obchodních aspektech práv k duševnímu vlastnictví (TRIPS) ze dne 15. 4. 1994.

Smlouva Světové organizace duševního vlastnictví o právu autorském ze dne 20. 12. 1996.

Směrnice Evropského parlamentu a Rady 2001/29/ES ze dne 22. května 2001 o harmonizaci některých aspektů autorského práva a práv s ním souvisejících v informační společnosti.

Směrnice Evropského parlamentu a Rady 2009/24/EC ze dne 23. dubna 2009 o právní ochraně počítačových programů.

Zákon č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon). In: 121/2000. 2010.

Zákon č. 40/1964 Sb., občanský zákoník, ve znění pozdějších předpisů.

USC : Title 17 - Copyrights. 1998. [cit. 2011-12-15] [online]. Dostupné z: <<http://www.law.cornell.edu/uscode/text/17>>.

Rulemaking on Exemptions from Prohibition on Circumvention of Technological Measures that Control Access to Copyrighted Works. [online]. [cit. 2011-12-20]. Dostupné z: <<http://www.copyright.gov/1201/2010/>>.

Proposal for a Council Directive on the legal protection of computer programs. COM (88) 816 final, 17. 3. 1989. [online]. [cit. 2011-12-17]. Dostupné z: <[http://aei.pitt.edu/13138/1/COM_\(88\)_816_final.pdf](http://aei.pitt.edu/13138/1/COM_(88)_816_final.pdf)>.

8.4 Judikatura

Rozhodnutí Druhého obvodu odvolacího soudu Spojených států ze dne 22. 6. 1992, sp. zn. 982 F.2d 693 (2d Cir. 1992.) [online]. [cit. 2011-12-15] Dostupné z: <<http://www.bitlaw.com/source/cases/copyright/altai.html>>.

Rozhodnutí Odvolacího soudu Spojených států ze dne 10. 9. 1992, sp. zn. 975 F.2d 832 (Fed. Cir. 1992) [online]. [cit. 2011-12-15] Dostupné z: <http://scholar.google.com/scholar_case?case=15866317401594691669&q=975+-F.2d+832&hl=en&cas_sdt=2002>.

Rozhodnutí Devátého obvodu odvolacího soudu Spojených států ze dne 20. 10. 1992, sp. zn. 977 F.2d 1510 (9th Cir. 1992). [online]. [cit. 2011-12-15] Dostupné z: <<http://bulk.resource.org/courts.gov/c/F2/977/977.F2d.1510.92-15655.html>>.

Rozhodnutí Odvolacího soudu Spojených států ze dne 29. 1. 2003, sp. zn. 320 F.3d 1317 [online] [cit. 2011-11-18]. Dostupné z: <http://scholar.google.com/scholar_case?case=11442789824223925516>.

Probíhající spor *SAS Institute v. World Programming*, sp. zn. [2010] EWHC 1829 (Ch) / Case C-406/10.

Rozhodnutí Chancery Division ze dne 30. 7. 2004, sp. zn. [2004] EWHC 1725 (Ch) (30 July 2004). [online]. Dostupné z: <<http://www.bailii.org/ew/cases/EWHC/Ch/2004/1725.html>>.

Stanovisko generálního advokáta Yvese Bota přednesené dne 29. 11. 2011, sp. zn. C 406/10 [online]. [cit. 2012-02-10]. Dostupné z: <<http://curia.europa.eu/juris/document/document.jsf?docid=115484&pageIndex=0&doclang=CS&mode=lst&dir=&occ=first&cid=1088404>>.

Rozsudek soudního dvora (Velkého senátu) ze dne 2. 5. 2012, sp. zn. C-406/10. [online]. [cit. 2012-05-20]. Dostupné z: <<http://curia.europa.eu/juris/document/document.jsf?text=&docid=122362&pageIndex=0&doclang=CS&mode=req&dir=&occ=first&part=1&cid=154228>>.