

Digitální kompozice jako princip práce s programovatelnými médii

Autorka textu: Michaela Rokosová

Anotace: Východisky k článku *Digitální kompozice jako princip práce s programovatelnými médii* jsou dva texty popisující algoritmickou a uživatelskou funkčnost dvou komerčních softwarových aplikací: *It looks like you're writing a letter: Microsoft Word* Matthewa Fullera (2001) a *Inside Photoshop* Lva Manoviche (2011). Oba se ve svých textech zaměřují na porovnání technik předchozích nedigitálních médií a nových médií na uvedených konkrétních příkladech. Matthew Fuller je k aplikaci *Microsoft Word* poměrně skeptický. Jeho argumentace se soustřeďuje na popis množství nástrojů, jež *Word* nabízí a s nimiž se musí uživatel vypořádat. Oproti tomu Lev Manovich si na začátku svého článku stanovuje jasný cíl a metodu, pomocí níž analyzuje nástroje aplikace *Adobe Photoshop*, a své teze uzavírá jasným odkazem, že klasické pojetí médií se v současnosti proměňuje na pojetí softwaru. V tomto článku se pokusíme porovnat argumentaci obou autorů s přihlédnutím k teorii softwarových studií, přičemž nás budou zajímat odpovědi především na následující otázky. V čem se odlišuje práce se softwarem od práce s předchozími médii? Jakým způsobem ovlivňuje software myšlení jeho uživatelů? Můžeme nazvat obecný princip dnešního vnímání médií jako koncept digitální kompozice?

Abstract: The starting point to the article *Digital composition as the principle of working with programmable media* are two texts describing both the algorithmic and user functionality of two commercial software applications: *It looks like you're writing a letter: Microsoft Word* written by Matthew Fuller (2001) and *Inside Photoshop* written by Lev Manovich (2011). Both of them focus in their texts on comparison of previous non-digital media techniques and those of new media on specific examples. Matthew Fuller is rather skeptical towards *Microsoft Word*. His arguments focus on description of a number of tools which are provided by *Word* and with which ones the user must deal with. In contrast, Lev Manovich establishes a clear goal and methods at the beginning of his article which he uses to analyze the *Adobe Photoshop* tools and he concludes his thesis with a clear reference that the classical concept of media is currently transforming to the concept of software. In this article, we will try to compare both authors' arguments with regard to the theory of software studies where we will be primarily interested in the answers to the following questions: What distinguishes the work with software from work with previous media? How the software influences thinking of its users? Can we call the general principle of today perception of media as a concept of the digital composition?

Klíčová slova: software, aplikace, digitální kompozice, softwarová studia, archeologie médií, nová média, uživatel, digitální kompozice

Keywords: software, application, digital composition, software studies, media archeology, new media, user

Úvod

Nová (digitální) média se od starších médií liší dvěma způsoby. Zatímco analogová média jako fotografie, film, televize, rozhlas či video jsou zprostředkovatelem vždy pouze jednoho typu sdělení (statického nebo pohyblivého obrazového, případně zvukového, odlišujícího se od sebe způsobem výroby i užitými technologiemi). Digitální média mají schopnost simulovat veškeré typy starších médií. Stávají se tak jakýmsi univerzálním strojem, platformou pro přístup ke statickým a pohyblivým obrazům, hudbě, televiznímu a rozhlasovému vysílání, ale i knihám a jiným textům a samozřejmě také k multimodálním [1] typům sdělení, a to při použití jediné technologie.

To, že nová média fungují jako rozhraní k nejrůznějším typům mediálních forem a obsahů, je způsobeno právě digitalitou, tedy číselnou reprezentací a kvantizací dat uložených uvnitř přístroje. Takto uložená data mohou být předmětem algoritmické manipulace. Jinými slovy, nová média jsou **programovatelná**. A právě tato vlastnost nových médií nám umožňuje výše zmíněnou multimodalitu. Jak píše Lev Manovich ve své knize *Principy nových médií*, „[s]rovnáním nových médií s tiskem, fotografií či televizí se nikdy nedobereme celé pravdy. Z jednoho úhlu pohledu zůstávají nová média jen dalším mediálním typem, z úhlu jiného však jsou určitým druhem počítačových dat, něčím uchovávaným v souborech a databázích, obnovovaným a tříděným, závislejícím na algoritmech a výstupních zařízeních. [...] Nová média mohou vypadat jako média tradiční, ale to je jenom povrch.“ (Manovich, 2002, s. 75)

Také **software** musíme vnímat skrze obdobnou logiku dvou úhlů pohledu. Na jedné straně je dnes tento pojem širší veřejností chápán jako programová aplikace, jež uživateli umožňuje práci s nejrůznějšími typy dat. [2] Podle Matthewa Fullera, „[s]oftware je vnímán jako nástroj, něco, s čím se něco dělá“. (Fuller, 2008, s. 3) [3] Ovšem na druhé straně musíme software zároveň vnímat jako rozhraní, prostřednictvím něhož komunikujeme s digitálním médiem. Je přitom důležité si uvědomovat existenci kódů a procesů pracujících na pozadí. Dle nástrojů, jež uživatel použije, jsou spouštěny naprogramované příkazy – algoritmy, které provedou nějakou akci.

Algoritmus je „[...] sled příkazů jednoznačně interpretovatelný vykonavatelem [...] k plnění určitých úkolů nebo uskutečňování postupů řešení problémů. [...] Ve volnějším smyslu slova označuje algoritmus jakoukoli sérii kroků určenou k řešení nějakého úkolu a může být vyjádřen v jakémkoli jazyce: od přirozených jazyků přes notové záznamy až po programovací jazyky“. (Horáková, 2010) Algoritmy tak vlastně zastupují myšlenkové procesy formalizované do série instrukcí, jedná se o logický postup vykonání určitého zadání. Budeme-li chtít například napsat na obyčejný papír nadpis, zvolíme si pravděpodobně jiný styl písma (například verzálky) a poté text barevně zvýrazníme. Totéž můžeme udělat i v počítačových textových editorech. Nástup digitálních médií nám však přinesl i možnosti, jež nebylo předtím možné tak jednoduchým způsobem uskutečnit. Vezměme si například funkci, bez níž se dnes již málokdo obejde – obyčejné mazání a přesouvání textu. To bylo dříve nemožné, aniž bychom jednotlivé kusy textu nerozstříhali a znovu neslepili (nebo celý text nenapsali jednoduše znovu).

Pojďme se nyní detailněji seznámit se dvěma články, které se zaměřují právě na porovnání technik předchozích nedigitálních médií a médií nových, a to na příkladech dvou aplikací: textového editoru *MS Word* ve studii *It looks like you're writing a letter: Microsoft Word* Matthewa Fullera (Fuller, 2001) a editoru pro grafické úpravy *Adobe Photoshop* v Manovichově textu *Inside Photoshop* (Manovich, 2011). Popíšeme algoritmické a uživatelské funkce jednotlivých programů a zaměříme se přitom na komparaci zvolených argumentačních technik a nejdůležitějších tezí autorů studií, se kterými pracujeme.

Při porovnání obou textů je důležité uvědomit si, že Fullerův článek poprvé vyšel v roce 2001, tedy o deset let dříve než studie Lva Manoviche. Vývoj počítačových aplikací byl v té době sice již poměrně daleko, ovšem tehdejší programy zdaleka nedosahovaly takových kvalit a možností jako ty z roku 2011. Dokonce i aplikace *MS Word* je dnes již o notný kus dále a umí nepřeborné množství nových funkcí než verze z roku 2001, kterou Fuller analyzoval. Dalším faktorem, jenž nám může ztížit srovnání obou dokumentů, je povaha popisovaných aplikací – textový editor má jinou historii, funkcionalitu, užití a specifika a vyžaduje odlišný přístup než grafický program. Nám však nepůjde o prostý popis rozdílů jednotlivých aplikací, ale o způsoby formulování obecné logiky používání těchto softwarů. Zkusíme pochopit, do jaké míry nám naprogramované funkce dovolují využít naši vlastní představu o tom, jakým způsobem bychom chtěli software využívat.

Šablona dopisu na rozloučenou zatím zůstává nedokončená...

V úvodu studie Matthew Fuller píše: „[...] program jako *Word* nepopírá autonomní práci či touhu po ní, ale parazituje na ní, ohraničuje ji a válkuje ji zároveň s tím, jak vstupuje a přeskupuje naše chápání uspořádání daného rámce. Obrovské množství nadbytečných funkcí je uskladněno ve vašem počítači i přesto, že je pravděpodobně nikdy nebudete potřebovat, čímž se vyhraňuje oproti možnosti každého uživatele samostatně rozšířit či změnit účel nebo datový typ.” (Fuller, 2001) [4] Vytýká tedy aplikaci existenci pevně přednastavených funkcí, aniž by si je mohl konkrétní uživatel vymezit dle vlastních potřeb. Ten je místo toho postaven před hotový uzavřený produkt. Aby mohl v programu pracovat, byl schopen jím procházet a vybírat si z nabídky nástrojů ten pravý, vhodný pro danou aktivitu, musí o něm předem získat dostatečné teoretické znalosti. Problematická je podle Fullera také integrovaná nápověda. Jedná se o archiv indexově seřazených informací o jednotlivých nástrojích. Aby však uživatel našel odpověď na svůj problém, musí správně definovat dotaz, tedy musí znát přesný název funkce, na niž se ptá. V tomto vidí Fuller popření teorie, že software je determinován daným hardwarem. Program po této stránce funguje jako tradiční médium – nespolupracuje s uživatelem interaktivně v reálném čase, ale nabízí pouze omezené a přednastavené možnosti vyhledávání.

Dále v článku se dozvídáme, že aplikace *MS Word* (stejně jako ostatní aplikace řady *Office*) je vytvořena v objektově orientovaném programovacím jazyku. To znamená, že jednotlivé objekty mají nadefinované neměnné příkazy, které se po zvolení příslušné funkce provedou, a to na pozadí; uživateli jsou tyto procesy skryty. Aby bylo příjemnější a jednodušší s objekty pracovat, jsou rozděleny do více skupin přístupných z nástrojové lišty podle logiky a možnosti použití. „Uživatel se stává jedním z objektů, ale ve specifické pozici v hierarchii objektů ostatních. Je vyloučen z vnitřního přenosu informací a místo toho se nachází na úrovni reprezentace prvků těchto informací, tedy na úrovni rozhraní. [...] Další možnost dotazování programu je mu odeprána.” (Fuller, 2001) [5] Tento způsob samozřejmě není pouze záležitostí *Wordu* a nedá se nezbytně označit za problematický, přesto však by se dalo poukázat na vývoj větší nezávislosti jednotlivých funkcí na úkonech, se kterými jsou spjaty. Důvod přebírání hotových funkcí předešlých verzí programu je očividně ekonomický. *Word* je součástí většího balíku *MS Office* a jednotlivé objekty se prolínají všemi jeho produkty. Je tak jednodušší i rychlejší vzít již hotový objekt než vyvíjet nový. Nehledě na to, že dialogové okno je naprogramováno jednoduchým šesti až osmiřádkovým kódem, zatímco složitější kódy zaberou až 100 příkazových řádků.

Při tak velkém množství funkcí, jež *Word* obsahuje, se může stát, že některé budou v hierarchii nástrojového menu zařazeny tak hluboko, že je uživatel ani nenajde. Fuller uvádí příklad funkce

Animovaný text, kdy uživatel musí zvolit oddíl Formát z nástrojové lišty, vybrat možnost Fonty, pak kliknout na nabídku dostupných formátů a najít mezi nimi možnost animovaného textu. Fuller si myslí, že „[...] u mnoha uživatelů je pravděpodobné, že bude tato možnost tak hluboko ve stromovém výběru, že zcela vypadne z jejich zřetel“: (Fuller, 2001) [6] Není to ovšem pouze problém usnadnění a kvality rozhraní nebo nadbytku méně užitečných funkcí, ale také způsobu, jakým vývojáři komerčního softwaru přiřazují důležitost jednotlivým funkcím, a manipulují tak vlastně uživatele k jejich používání. Fuller se zamýšlí nad tím, „[...] jaké modely ‘práce’ byly vzorem pro Word v tom smyslu, že implementované možnosti úpravy textů nezahrnují tak jednoduché funkce, jako jsou automatické abecední pořadí položek seznamu nebo možnost vytvářet kombinatorickou poezii, stejně jednoduše jako WordArt [7]“ (Fuller, 2001) [8].

Další rovinou, jíž se Fuller zabývá, je rovina textury. Editory typu *Word* v podstatě pracují s textem pomocí textu, neboť i programový kód je systém znaků zapsaných v řádcích, které se při spuštění algoritmu zároveň stávají strojem. Oproti běžnému psaní rukou je však textový editor zaměřen více na zpracování dokumentu než na jeho samotný obsah. Máme tak k dispozici rámeček, „[...] který zůstává stejný, ať už to, co je napsáno, je milostný dopis nebo daňové přiznání“: (Fuller, 2001) [9] To dokazuje i fakt, že aplikace nemá dostatečně vyvinutou sémantickou rovinu. Změníme-li slovosled věty přesunutím již napsaných slov, program nepozná změnu obsahu a nenahradí na začátku věty malé písmeno velkým. Existují tu i šablony, ukázkové dokumenty typu *elegantní fax*, *současný fax*, *formální dopis* apod. Pomocí nich se stávají všichni uživatelé unifikovanými. Jak Fuller vtipně dodává, „[šablona dopisu na rozloučenou zatím zůstává nedokončená“: (Fuller, 2001) [10] Aplikace *Word* tak nemá přímý zájem o obsah informace či komunikace, ale snaží se pouze usnadnit práci při jejich vytváření. Zároveň však způsobuje, že se člověk stává jedním z objektů programu a je jím ovlivňován, aniž by to sám postřehl. Existence takového množství možností formální úpravy textu totiž zahlcuje uživatele, jenž jim podléhá, a ve snaze nalézt pro svůj text ten nejvhodnější rámeček se tak vlastně stává pouhým předmětem obchodu a manipulace společnosti Microsoft. Samotný text – jeho význam a dopad – tak zůstává stranou a je z ekonomického hlediska zcela nepodstatný.

V závěru článku se snaží Fuller zodpovědět otázku, zda je možné poučit se z analýzy této aplikace a zformulovat kritéria pro vývoj softwaru otevřeného uživatelským přáním a požadavkům; tedy zda je možné vytvořit software, který by podpořil možnost otevřenější spolupráce člověk-stroj. Možností, jíž Fuller vidí, je vývoj volně šiřitelných programů. Je ale důležité, aby malí nekorporátní vývojáři pouze nekopírovali funkce, které znají z *Wordu* a dalších podobných komerčních aplikací, ale aby šli dále, začali znovu od začátku, s otevřenou myslí a novými nápady.

Re:

Práce s programem *MS Word* je občas bezesporu „zákeřná“. Všichni jistě známe problémy se zarovnáním, kdy nám po stisknutí tabulátoru odskočí celý odstavec jinak, než jsme si přáli. Také pokročilejší funkce, jako například hromadná korespondence, chtějí již trochu více zkušeností nebo opravdu dobrý návod, protože integrovaná nápověda vždy nestačí a prostředí *Wordu* nebývá ve všech případech zcela intuitivní. Na druhou stranu, balík *MS Office* po svém uvedení na trh přinesl ve své době pokročilé možnosti práce s daty přístupné skrze uživatelsky přívětivé grafické rozhraní, a naznačil tak směr, jakým se svět textových editorů bude ubírat dále. Pro spoustu podobných aplikací, ať už placených či tzv. freewarů, je *MS Word* vzorem, kterému se chtějí podobat. Vzhledem k jeho rozšířenosti se totiž vývojáři bojí toho, že pokud by byl program zcela jiný, běžný uživatel zvyklý na *MS Word* by se v něm

neorientoval a nedal by mu pak přednost. Fullerovu výzvu k vytváření open source programů, jež by začaly „na zelené louce“ a snažily se uživatelům nabídnout nová řešení, tak vidíme jako poněkud utopickou. Existuje celá řada freeware balíčků snažících se obsáhnout nebo i zdokonalit funkcionalitu balíku *MS Office* (například *Kingsoft Office Suite*, *SoftMaker FreeOffice*, *Apache OpenOffice* či *LibreOffice*). [11] Některé z nich čerpají inspiraci přímo od svých uživatelů a pobízí je, aby přispěli svým testováním ke zdokonalení softwaru. [12] Zde vidíme snahu naplňující poselství Fullerovy studie, ovšem když si ony programy otevřeme, vzhled i funkcionalita nám stále silně připomíná analyzovaný *MS Word*. Věříme však, že to není na závadu. Textový editor má být v první řadě programem pro vytváření a úpravu textů, což všechny tyto aplikace splňují. Pokud chce uživatel použít pokročilejší funkce, existují dnes nejrůznější publikace i online manuály, kde může najít odpověď na daný problém. Navíc, pokud není uživatel natolik nadaný v oblasti informačních technologií, většinou ho ani nenapadne pokročilé možnosti použít. A pro uživatele, kteří jsou zběhlejší v práci s počítačem, by neměl být problém nástroje *Wordu* vyhledat a správně použít.

Vítr ve Photoshopu

Na začátku článku *Inside Photoshop* předkládá Lev Manovich dvě schémata, kterými se bude snažit zodpovědět otázku, jak porozumět způsobu, jakým tvarují softwarové aplikace náš svět a naši představivost.

Prvním schématem je rozdělení veškerých softwarových technik na dvě strategie v závislosti na typu dat, jež zpracovávají:

1a) Techniky tvorby, manipulace a přístupu k médiím, jež jsou specifické pro konkrétní typy dat, neboli **mediálně specifické techniky**. Jako příklad uvádí filtry umožňující pracovat se spojitými odstíny obrázku. Tyto filtry můžeme použít na 2D obrázky, nikoli však již na 3D modely.

1b) Nové softwarové techniky umožňující práci s digitálními daty obecně nejsou tedy specifické pro jednotlivá média, jsou **na médiu nezávislé**. Jsou to obecné možnosti manipulace s daty bez ohledu, zda se jedná o obrázky, texty, 3D tvary apod., například hyperlinkování, kopírování a vkládání, vyhledávání či síťové protokoly.

Druhé schéma také rozděluje softwarové techniky na dva druhy, ale jiným způsobem. Co je zde důležité, je vztah mezi softwarovými technikami a technologiemi předchůdců digitálních médií:

2a) **Simulace předchozích fyzických médií rozšířená o nové vlastnosti a funkce**. Užíváme-li počítač pro simulaci předchozích médií, místo abychom použili tato média přímo, je to z důvodu právě nově přidaných vlastností, jež rozšiřují expresivitu a komunikační potenciál média.

2b) **Nová (počítačová) média, která nejsou přímým následovníkem jiného média**. Příkladem je hypertext a hypermédium, interaktivní 3D prostory, internet.

Otázkou ovšem je, zda druhé schéma není určené spíše pro teoretiky, mediální historiky a starší generaci, jež zažila po většinu života pouze předchozí nedigitální média, a zda jí mohou porozumět „digitální domorodci“, jejichž vnímání je již od dětství ovlivňováno všudypřítomností digitálních médií. Manovich proto definuje princip, který si mohou i oni v praxi ověřit a který je společný pro veškeré ryze novomediální techniky softwarových aplikací: „[...] spousta technik, jež nesimulují nic, co existovalo dříve

– nebo alespoň ne v očividné rovině – nabízí **vysokouúrovňovou automatizaci kreativních procesů**. Místo aby uživatel kontroloval každý detail, pouze nastaví parametry a ovládací prvky a uvede nástroj do chodu. Veškeré generativní (nebo také 'procedurální') techniky dostupné v softwarových médiích spadají do této kategorie.“ (Manovich, 2011) [13]

Dále ve svém článku Manovich testuje platnost navržených schémat na dvou technikách dostupných v grafickém editoru *Photoshop* – na **filtrech** a na **vrstvách**.

Názvy mnoha filtrů ve *Photoshopu* odkazují k technikám pro úpravu a tvorbu obrázku dostupným před rozšířením aplikačního softwaru nových médií. Oproti „ručnímu“ kreslení, malování, fotografování atd. však *Photoshop* nabízí pokročilejší možnosti technik prostřednictvím interaktivních posuvníků nebo dialogových oken, kde je možné zadat konkrétní numerickou hodnotu. „*Tyto volby umožňují kontrolovat vizuální efekt filtru s takovým stupněm preciznosti, kterého by nebylo možné docílit při použití odpovídajícího fyzického nástroje.*“ (Manovich, 2011) [14] Navazujeme zde tak na bod 2a) – rozšíření vlastností předchozích médií. Jak však Manovich správně upozorňuje, nemůžeme tuto softwarem umožněnou explicitní kontrolu simulace považovat striktně za lepší, neboť profesionálové klasických uměleckých technik dokážou vytvářet při své práci efekty, mnohdy i nechtěné, jež by nebylo možné vytvořit pomocí grafického editoru.

Oproti filtrům s předobrazem v předchozích médiích zde můžeme nalézt i takové, které odkazují k fenoménům fyzického světa, jež na první pohled nemají co do činění s médii. Jedním z nich je *Wind filter*. Ten můžeme aplikovat na libovolný objekt, tedy například na 2D obrázek. Výsledek je však odlišný od toho, jak vítr skutečně působí na stromy, pole nebo trávu. Spíše nám takto upravený obrázek připomíná fotografii pořízenou ve větru při užití dlouhé expozice. „*Můžeme tak název filtru 'vítr' pojímat jako dvojitou metaforu – pomáhá nám představit si, co tento konkrétní algoritmus dělá s obrázkem – a jako simulaci specifické fotografické techniky (dlouhá expozice). Ve zkratce, ačkoli jeho název poukazuje na fyzický svět, jeho skutečná funkčnost může taktéž odkazovat k předdigitální mediální technologii.*“ (Manovich, 2011) [15]

Dále zde můžeme nalézt filtry typu High-pass, Median, Reduce Noise, Sharpen nebo Equalize. Jsou to filtry pracující přímo s úpravou digitálních informací daného objektu. Můžeme snad tyto považovat za ryzí příklady technik nových médií? Manovich odpovídá, že nikoli, neboť i tyto filtry jsou softwarové simulace společné pro veškeré elektronické, telekomunikační a IT technologie. Navíc nejsou užívány výlučně při zpracování digitálního obrazu, ale lze je použít na jakákoli data – zvuky, televizní přenos, informace zachycené monitorováním životního prostředí či lékařských zařízení apod. „*V rámci programu Photoshop pracují tyto filtry se spojenými tóny obrazů, ale vzhledem k tomu, že mohou být aplikovány také na zvuk a další typy signálů, patří ve skutečnosti do naší kategorie 'na médiu nezávislých' technik (1a).*“ (Manovich, 2011) [16] Podobně jako tyto filtry, také mnohé techniky, jež se zdají být vytvořené až v prostředí digitálních technologií, nebyly vyvinuty specificky pro práci s mediálními daty, ale pro zpracování signálu a informací obecně.

Vraťme se nyní k filtrům, jejichž názvy odkazují k předobrazům fyzického světa, a zaměřme se na způsob, jakým fungují. I přesto, že jakýkoli možný úkon má svého předchůdce v analogovém světě, fakt, že je simulován v počítači, umožňuje rozšířit a znásobit jeho efekt díky číselné reprezentaci počítačových dat a automatizaci algoritmických výpočtů. Ve *Photoshopu* totiž můžeme ručně zadat numerické parametry zvoleného nástroje, a tím ovlivnit výsledný vzhled. Jinak řečeno, při užití stejného filtru můžeme změnou parametrů dosáhnout jak výsledku podobajícího se reálnému tvaru (obsaženému

v názvu filtru), tak abstraktního tvaru nepřipomínajícím nám nic z běžného života. Lev Manovich uvádí příklad filtru *Wave*, který pracuje se sinovou funkcí. Zadáme-li jako parametr funkce sin nízká čísla, výsledný obraz je přetvořen tak, že kontury skutečně připomínají vlny. Zadáme-li ale čísla vyšší, obraz je deformován natolik, že již nelze o vlnách uvažovat ani metaforicky.

A právě tato skutečnost má významné důsledky pro teorii softwaru. Vysokoúrovňová automatizace algoritmu je základem dnes již běžných generativních (neboli procedurálních) softwarových technik používaných například v animacích a pohyblivé grafice či počítačových hrách. Vzhledem k tomu, že většina děl vytvořených pomocí generovaného algoritmu je abstraktní povahy, stává se, že umělci a teoretikové je staví do opozice proti programům jako *Photoshop*, jež jsou ilustrátory a fotografy masově užívány k reálnému a figurativnímu zobrazování. „Navíc, jelikož tyto aplikace simulují starší manuální model práce, jsou často vnímány jako méně 'médiá-specifické' než generativní software. Finální výsledek této kritiky je ten, že software simulující 'stará média' je považován za konzervativní, zatímco generativní algoritmy a díla jsou prezentovány jako progresivní, protože jsou typické pro 'nová média.'“ (Manovich, 2011) [17] Vidíme ale, že i ve *Photoshopu* jsou implementovány procedurální softwarové techniky, které využívají vysokoúrovňovou automatizaci procesů.

I přestože nás předchozí úvahy dovedly k závěrům, že v zásadě všechny softwarové techniky mají předobraz v již existujících fenoménech, zásadní je jejich společná vlastnost, že jsou implementovány jako programy a funkce do počítačového prostředí. Následují tak principy moderního softwarového inženýrství. Jejich rozhraní používají zavedené konvence vestavěné do všech typů aplikačního softwaru. Jsou podrobovány rozsáhlé numerické kontrole, jejich nastavení je možné uložit a později načíst z paměti, jejich použití je zaznamenáváno v okně historie, takže mohou být zpětně vyvolány. „Jinými slovy, nabývají plné funkcionality moderního softwarového prostředí – funkcionality, jež je výrazně odlišná od technik fyzických nástrojů a strojů, které existovaly dříve.“ (Manovich, 2011) [18]

Na závěr článku se Manovich zaměřuje na jinou techniku, již poprvé přinesla právě aplikace *Photoshop* – vrstvy. Náповěda programu definuje tuto techniku jako možnost práce s jakýmkoli prvkem obrázku bez narušení ostatních prvků. „Z pohledu mediální teorie je však technika vrstev více než to. Znovu definuje způsob, jakým jsou obrazy tvořeny, a také skutečný význam 'obrazu'. Co dříve bývalo nerozdělitelné, se nyní stává složeninou samostatných prvků.“ (Manovich, 2011) [19] Každá vrstva může obsahovat libovolné prvky a různé modifikace jednoho obrazu. Designér může kontrolovat průhlednost vrstev, slučovat je dohromady, měnit jejich pořadí či vypínat ty, které se zrovna do výsledného obrazu nehodí. Přesně tento způsob dnes definuje realitu profesionálního designu a vnímání obrazu jako takového. Místo abychom pracovali s celým obrázkem a každou změnu prováděli v reálném čase (kdy v případě fyzického nástroje je změna nevratná), pracujeme s celou kolekcí samostatných prvků, jež můžeme mazat, přidávat, kopírovat, importovat a modifikovat tak dlouho, dokud nebudeme s výsledným obrazem (tedy souhrnem všech těchto jednotlivých elementů ve více vrstvách) dokonale spokojeni. Navíc, i když finální dílo uložíme, stále nám zůstane informace o jednotlivých vrstvách uchována a můžeme se kdykoli vrátit, obraz opět „rozebrat na díly“ a poskládat jinak.

Logika vrstev je opět spjatá s některými dřívějšími technikami. Nalézt ji můžeme ve vizuálních technikách jako animování pomocí filmového pásu, vícenásobná expozice, projekce na pozadí či klíčování videa, ale také v hudební technologii u vícestopého nahrávání zvuku. Zároveň však existuje vodítko, jež spojuje vrstvy s obecnými principy moderního počítačového programování. V roce 1984 dva počítačové vědci, Thomas Porter a Tom Duff, formálně definovali **koncept digitální kompozice** (a to na přednášce

v rámci konference SIGGRAPH): „Zkušenost nás naučila rozložit velká seskupení zdrojového kódu do samostatných modulů s cílem zkrátit čas kompilace. Při chybě v algoritmu tak stačí znovu zkompilovat pouze jeden z modulů, čímž je zaručeno relativně rychlé znovunačtení celého programu. Podobně nás malé chyby v barevnosti nebo designu jednoho objektu nenutí k opětovné kompilaci celého obrazu.“ (Porter – Duff, 1984, s. 253) [20]

Re:

Průkopníci softwaru se snažili rozšířit vlastnosti mediálních technologií s cílem vytvořit nové médium s novými vlastnostmi. Softwarové techniky odkazující se k předchozím fyzickým, mechanickým nebo elektrickým nástrojům a kreativním procesům jsou zároveň také novými médii, protože se chovají velmi odlišně ve srovnání se svými předchůdci. Můžeme tak říci, že **veškeré mediální techniky a nástroje přístupné skrze softwarové aplikace jsou nová média**, nezáleží přitom na tom, do kterého Manovichem navrhnutého schématu spadají, všechny jsou jednotlivými případy jediného typu technologie – **interaktivního aplikačního softwaru**. Jeho vlastnostmi jsou interaktivita, uživatelské přizpůsobení, možnost simulovat jiná média či informační technologie a definovat nová, schopnost zpracovat obrovské množství informací v reálném čase nebo kontrola a komunikace s jinými zařízeními. Všechny tyto funkce moderního softwaru jej odlišují od předchozích médií, informačních technologií a nástrojů vytvořených člověkem.

Shrnutí

Při porovnávání výše uvedených textů jsme si uvědomili, jak zřejmý rozdíl mezi oběma je. Matthew Fuller pojímá analýzu programu *MS Word* poměrně skepticky, zaměřuje se převážně na uživatelský přístup k softwaru, nestanovuje si konkrétní cíle a metody analýzy. Navíc je jeho text psán spíše publicistickým tónem, často užívá metafory a citově zabarvené výrazy, což můžeme vidět i na mezititulcích, jež zvolil (např. „P-O-M-O-O-C“, „Nehodící se škrtněte“, „Sedé prostředí zvyšuje produktivitu vajec u kuřat“ nebo „Psaní se zdviženými pěstmi“). Analýza programu *MS Word* mu slouží jako ilustrace jeho kritického pohledu na nová média, varování před unifikací, „zmasověním“ uživatelů, není však sama o sobě předmětem i cílem jeho výzkumu. Oproti tomu si Lev Manovich hned v úvodu stanovuje jasnou metodiku, pomocí níž bude analyzovat vybrané funkce softwaru, své metody se drží, logicky argumentuje a spíše se snaží o analytický pohled na problematiku. Fuller vyjadřuje především obavy z unifikace výrazu uživatelů v interakci s předem definovaným prostředím aplikace *Word*, zatímco Manovich se opírá o ideologicky neutrální analýzu funkcí nových médií jako nástrojů lidské kreativity, podle nějž nová média následují kulturní formáty médií tradičních, moderních, a pouze mění způsob, jakým mohou být tyto formáty užívány. (Manovich, 2002)

Zdá se, že všechny prvky softwarových struktur mají původ v mediálních a kulturních praktikách moderní společnosti a zároveň ve vývoji počítačových technologií a softwarového inženýrství. Média se tak stávají softwarem, neboť jednotlivé techniky se vzájemně natolik prostoupily, že není možné je od sebe striktně rozlišit. Navzájem se ovlivňují, koexistují, určují náš pohled na svět a aktivity v něm. Libovolná softwarová aplikace je souhrnem nejrůznějších nástrojů, z nichž některé odkazují k již zavedeným kulturním formám analogových médií (koláž, montáž, funkce kamery apod.), a další jsou pak funkcemi specifickými pro digitální média. Obecný princip práce s programovatelnými médii tak můžeme nazvat **digitální kompozicí**.

Poznámky:

[1] Multimodalita a digitální integrace jsou pojmy, které zavádí Pierre Lévy ve svém spise *Kyberkultura*, kde vysvětluje správný význam běžně používaného pojmu multimédium. Multimodální sdělení jsou taková, která „[...] zapojují více smyslových schopností (zrak, sluch, hmat, tělesné pocity)“; zatímco „[...] termín ‘multimédia’ v podstatě znamená to, co používá několik komunikačních nosičů nebo kanálů“ (Lévy, 2000, s. 58–59)

[2] Podíváme-li se na výklad slova software do nejrůznějších slovníků, nalezneme jej v téměř většině případů ve významu počítačového programu. Slovo software je však složeninou slova soft (měkký, jemný, snadný, pohodlný) a ware (zboží) a je užíván od poloviny 19. století ve významu jemného zboží, nebo také rychle se kazícího spotřebního zboží. (Harper, 2012)

[3] Přeložila autorka článku. Originální znění: „*Software is seen as a tool, something that you do something with.*“

[4] Přeložila autorka článku. Originální znění: „[...] *program such as Word doesn't deny autonomous work or the desire for it, but parasites it, corrals and rides it at the same time as entering into an arrangement of simultaneous recomposition of scope. The surplus feature mountain warehoused in your computer is stored against the possibility of your ever needing it, against the possibility of the user's self expanding, or changing purpose or data-type.*“

[5] Přeložila autorka článku. Originální znění: „*The user becomes an object, but at a peculiar position in the hierarchy of others. It is excluded from the internal transmission of information, and instead allocated representations of elements of this information as interface. [...] Further interrogability of the program is denied.*“

[6] Přeložila autorka článku. Originální znění: „*To many users it is likely that this option should be so far down a choice tree that it drops off completely.*“

[7] „*Objekty WordArt tvoří galerii textových stylů, které můžete přidat do dokumentů sady Microsoft Office 2007 a vytvořit tak ozdobné efekty, například text vrhající stín nebo zrcadlený (odrážející se) text.*“ (Přidání nebo odebrání objektu, 2013)

[8] Přeložila autorka článku. Originální znění: „[...] *which models of 'work' have informed Word to the extent that the types of text management that it encompasses have not included such simple features as automated alphabetical ordering of list items or the ability to produce combinatorial poetry as easily as 'Word Art.'*“

[9] Přeložila autorka článku. Originální znění: „[...] *that remains the same whether what is being written is a love letter or a tax return.*“

[10] Přeložila autorka článku. Originální znění: „*Suicide Note Wizard remains uncompleted.*“

[11] *Libre Office a Apache OpenOffice*, 2012.

[12] *Apache OpenOffice*, 2011–2012.

[13] Přeložila autorka článku. Originální znění: „[...] *many of the techniques that do not simulate anything that existed previously – at least, not in any obvious way – offer higher-level automation of creative processes. Rather than controlling every detail, a user specifies parameters and controls and sets the tool in motion. All generative (also called 'procedural') techniques available in media software fall into this category.*“

[14] Přeložila autorka článku. Originální znění: „*These options make possible to control filter's visual effects with a degree of precision that would be hard to achieve when using the corresponding physical tool.*“

[15] Přeložila autorka článku. Originální znění: „*Therefore, we can think of the name 'Wind' both as a metaphor – to help us imagine what a particular algorithmic transformation does to an image – and as a simulation of a particular photographic technique (long exposure). In short, although its name points to the physical world, its actual operations may also refer to a pre-digital media technology.*“

[16] Přeložila autorka článku. Originální znění: „*In their Photoshop implementation, these filters work on continuous-tone images, but since they can be also applied to sound and other types of signals, they actually belong to our 'media-independent' category (1a).*“

[17] Přeložila autorka článku. Originální znění: „Additionally, because these applications simulate older manual model of creation, they are also seen as less ‘new media specific’ than generative software. The end result of both of these critiques is that software that simulate ‘old media’ are thought to be conservative, while generative algorithms and artworks are presented as progressive because these are unique to ‘new media.’“

[18] Přeložila autorka článku. Originální znění: „In other words, they acquire the full functionality of modern software environment – the functionality that is significantly different from that of physical tools and machines that existed previously.“

[19] Přeložila autorka článku. Originální znění: „From the point of view of media theory, however, the Layers feature is much more than that. It redefines both how images are created and what an ‘image’ actually means. What used to be an indivisible whole becomes a composite of separate parts.“

[20] Přeložila autorka článku. Originální znění: „Experience has taught us to break down large bodies of source code into separate modules in order to save compilation time. An error in one routine forces only the recompilation of its module and the relatively quick reloading of the entire program. Similarly, small errors in coloration or design in one object should not force ‘recompilation’ of the entire image.“

Použitá literatura:

Apache OpenOffice: Get involved, 2012. *The Apache Software Foundation* [online]. –2012. [cit. 2013-01-09]. Dostupné z: <http://openoffice.apache.org/get-involved.html>.

FULLER, Matthew, 2003. *Behind the Blip. Essays on the Culture of Software*. Michiganská univerzita: Autonomedia. ISBN 1-57027-139-9.

FULLER, Matthew, 2006. *Software Studies: A Lexicon* [online]. [cit. 2013-01-06]. Cambridge, MA: MIT Press, s. 3. ISBN 97-0-262-06274-9. Dostupné z: [http://www.google.cz/books?hl=cs&lr=&id=LFJ3ashVBuIC&oi=fnd&pg=PR5&dq=FULLER,+Matthew.+Software+Studies.+\(A+Lexicon,+2008\)+...&ots=G12k3I-xgX&sig=qItXhKvDXPnwd9e2v8JlO4UPDrM&redir_esc=y#v=onepage&q=blue&f=false](http://www.google.cz/books?hl=cs&lr=&id=LFJ3ashVBuIC&oi=fnd&pg=PR5&dq=FULLER,+Matthew.+Software+Studies.+(A+Lexicon,+2008)+...&ots=G12k3I-xgX&sig=qItXhKvDXPnwd9e2v8JlO4UPDrM&redir_esc=y#v=onepage&q=blue&f=false).

FULLER, Matthew, 2012. It looks like you're writing a letter: Microsoft Word. *Heise online* [online]. [cit. 2013-01-09]. Dostupné z: <http://www.heise.de/tp/artikel/7/7073/1.html>.

HARPER, Douglas, 2012. Software. *The Online Etymology Dictionary* [online]. [cit. 2013-01-30]. Dostupné z: <http://www.etymonline.com/index.php?term=software>.

HORÁKOVÁ, Jana, 2010. Algoritmus. *Umělecké dílo v době své digitální reprodukovatelnosti: Heslář | Filozofická fakulta Masarykovy univerzity* [online]. [cit. 2013-01-28]. Dostupné z: <http://is.muni.cz/do/rect/el/estud/ff/ps10/dilo/web/pages/heslar.html>.

LÉVY, Pierre, 2000. *Kyberkultura: zpráva pro radu Evropy v rámci projektu Nové technologie: kulturní spolupráce a komunikace*. Praha: Univerzita Karlova. S. 58–59. ISBN 80-246-0109-5.

Libre Office a Apache OpenOffice: V čem se liší, 2012. *Elektronický Ott* [online]. [cit. 2013-01-06]. Dostupné z: <http://www.e-ott.info/2012/05/08/libreoffice-a-apache-openoffice-v-cem-se-lisi>.

MANOVICH, Lev, 2002. Principy nových médií. *Teorie vědy: časopis pro teorii vědy, techniky a komunikace: téma: nová média*. XI, XXIV, 2, 2002, s. 55–76.

MANOVICH, Lev, 2011. Inside Photoshop. *Computational Culture: a journal of software studies* [online]. [cit. 2013-01-10]. Dostupné z: <http://computationalculture.net/article/inside-photoshop>.

MANOVICH, Lev, 2012. Avant-Garde as Software. *University of California, Art Nodes* [online]. [cit. 2013-01-06]. Dostupné z: <http://www.uoc.edu/artnodes/espai/eng/art/manovich1002/manovich1002.html>.

PORTER, Thomas – DUFF, Tom, 1984. Compositing Digital Images. *Keithp* [online]. [cit. 2013-01-12]. Computer Graphics. 1984-07, roč. 18, č. 3. s. 253–259. ACM 0-89791-138-5/84/007/0253. Dostupné z: <http://keithp.com/~keithp/porterduff/p253-porter.pdf>.

Přidání nebo odebrání objektu WordArt – Word – Office.com, 2013. *Microsoft* [online]. [cit. 2013-01-30]. Dostupné z: <http://office.microsoft.com/cs-cz/word-help/pridani-nebo-odebrani-objektu-wordart-HP010079386.aspx>.