

Horáková, Jana

Softwarové umění: programování excessu

Musicologica Brunensia. 2012, vol. 47, iss. 2, pp. [23]-47

ISSN 1212-0391

Stable URL (handle): <https://hdl.handle.net/11222.digilib/125887>

Access Date: 16. 02. 2024

Version: 20220831

Terms of use: Digital Library of the Faculty of Arts, Masaryk University provides access to digitized documents strictly for personal use, unless otherwise specified.

JANA HORÁKOVÁ (BRNO)

SOFTWAREVÉ UMĚNÍ: PROGRAMOVÁNÍ EXCESU

Anotace

V diskurzu umění nových médií se setkáváme s nadprodukcí pojmenování této umělecké praxe. Produkce neologismů je pro tuto disciplínu natolik specifická, že můžeme mluvit o **tekuté identitě diskurzu nových médií**. Příspěvek je věnován specifikaci pojmu **softwarové umění** ve vztahu k jiným označením umění využívajícího digitální média, konkrétně **počítačové umění** a **počítačem generované umění**. Softwarové umění představíme jako disciplínu propojující matematické, poetické a metafyzické chápání komputace (A. Lovelace), jako diskurz osvobozující software z logiky čisté funkcionality ve prospěch jeho metaforické funkce (A. Turing), a jako uměleckou tvorbu zkoumající limity lidské i strojové racionality a imaginace, kterou můžeme nazvat **extrémní programování** nebo **programování excessu**.

Úvod

Umění využívající digitální a síťová média, které se skrývá pod různými názvy, ale představuje poměrně jasně definovatelnou praxi a rozvinutý teoretický diskurz, hledá argumenty pro legitimitu svého požadavku na autonomní existenci mimo hranice tradičních dějin umění. S odvoláním na skutečnost, že teritorium této tvorby leží mezi „dvěma kulturami“¹, neboť propojuje oblasti umění, vědy a techniky po stránce konceptuální i po stránce tvůrčích strategií², jsou vznášeny požadavky na vytvoření pojmového aparátu, který by transdisciplinární povahu této kreativní praxe rozvíjel také v rámci teoretické reflexe. Důsledkem těchto snah je nadprodukce neologismů a z toho vyplývající jistá terminologická nestabilita.

Domnívám se, že tendenci k produkování nových pojmů a hledání nových přístupů k tvorbě v prostředí digitálních médií nemůžeme vysvětlit pouhým kon-

¹ Snow, C. P. *The Two Cultures and the Scientific Revolution*. Cambridge, U.K.: Cambridge University Press, 1959.

² Shanken, E. A. *Historicizing Art and Technology: Forging a Method and Firing a Canon*. In: O. Grau (ed.). *Media Art Histories*. Cambridge – London: MIT Press, s. 43–70. Shanken v uvedené studii navrhuje, aby byla ustavena nová disciplína AST (art, science, technology) která bude adekvátní platformou pro výzkum umění nových médií.

statováním, že se jedná o jev přechodný, spjatý s ranou fází každého (nového) média, pro kterou je charakteristické hledání svého místa/funkce mezi ostatními, již etablovanými uměleckými a technickými médii a technologiemi.³ Počítač už není nové médium, vždyť první experimenty s počítačem jako uměleckým nástrojem spadají minimálně do 60. let minulého století a od té doby můžeme sledovat kontinuální vývoj tohoto umění. Spíše můžeme říci, že teoretici umění digitálních médií, s ohledem na vývoj myšlení o jiných druzích médií⁴, přijali terminologickou nestabilitu a s ní spjatou flexibilitu svého teritoria jako něco daného, co je třeba afirmovat a dále s tím pracovat. Jakákoli snaha vedoucí k mediálnímu purismu by byla krokem vydělujícím teorii umění digitálních médií ze současného společenskovešedního a uměnovědného diskurzu.

Také v prostředí tradičních uměnovědných disciplín se v posledních desetiletích častěji setkáváme se zaváděním neologismů, které je motivováno snahou o rozkolísání pevné mřížky pojmového aparátu ve prospěch zavádění nových pohledů a filtrů do poznání o nich. Tato mřížka však současně plní funkci stabilizujícího pozadí, vůči kterému se nové pojmy mohou jasněji vymezit a formovat. V této souvislosti můžeme proto mluvit o procesu rozvíjení, spíše než petrifikaci, zjemňování vnímání, spíše než schematizaci konceptů tvořících základní pilíře oborových diskurzů. V případě tak mladého oboru jakým je umění digitálních médií je terminologická nestabilita nejen strategií umožňující pohyb myšlení napříč více či méně daným teritoriem, ale stala se jeho podstatným charakteristickým rysem. Můžeme proto mluvit o **tekuté identitě oboru umění digitálních médií**.

Příkladem dokládajícím „tekutost“ této disciplíny je také skutečnost, že dosud nepanuje shoda ani ve věci názvu oboru. Setkáme se s pojmenováním nová média, umění nových médií, umění a elektronická média, nová digitální média, počítačové umění, internetové umění, digitální umění, multimédia. Teoretici umění nových médií produkují nové a nové neologismy, které neaspírají na obecnou a pokud možno trvalou platnost, ale mají být co **nejpřesnějším výrazem**, ve kterém by se protнула rovina definiční (Co?) s hlediskem (Jak?), které do dané problematiky vnáší autor, umělec nebo teoretik.

I Softwarové umění vs. počítačové umění

Dobrym příkladem tvrzení, že různé pojmy asociují různé přístupy k umění v prostředí počítačů je vztah pojmů **počítačové umění** (computer art) a **poči-**

³ Například film se v prvních dvaceti letech svého vývoje identifikoval s médii jako fotografie, ilustrovaná hudba, vaudeville, opera, komiks, rozhlas, fonograf, telefon. Viz Altman, R. *Silent Film Sound*. New York: Columbia University Press, 2007.

⁴ Například filmová studia v posledních letech rehabilitovala význam rané fáze filmu, revidují tradiční dějiny filmu vyhledáváním intermediálních vztahů k jiným médiím, kulturním praktikám a institucím protomediální i postmediální doby. Viz Szczepanik, P. (ed.) *Nová filmová historie*. Praha: Herman & synové, 2004.

tačem generované umění (computer generated art) neboť oba se historicky váží k počáteční fázi průzkumu počítače jako kreativního média (zhruba 60. až 70. léta 20. století). Označení počítačové umění zahrnuje počítačovou grafiku, poezii i hudbu, které rozvíjejí myšlenku ne-lidské strojové kreativity a estetiky artikulovanou v tvůrčích procesech ‚rozehrávajících‘ možnost, že počítače jsou schopny tvořit autonomně, mimo omezení subjektivního pohledu, intencí a zásahů umělce. Tento koncept koresponduje jak s greenbergovským pojetím umělecké avantgardy jako směřování k autonomii uměleckých prostředků a potažmo k autonomii díla samotného, tak i s dobovým zájmem umělců o princip náhody, indeterminismus či nezáměrnost v umění.⁵ Peter Weibel se dokonce domnívá, že strojové a s pomocí strojů vytvořené umění, jehož počátek spojuje s vynálezem fotografie, není jen logickou konsekvencí vývoje výtvarného umění, ale že předjímá vývoj moderního umění směrem k autonomii výrazových prostředků, barvy, linie atd., logicky vedoucí k autonomii díla samotného. Weibel napsal, že: „Autonomie fotografického stroje byla prvním modelem ‚autonomie‘, který spustil logiku moderního umění spočívající v progresivním vývoji jeho autonomních prvků.“⁶ Koncept autonomie uměleckého díla dovádí do logických, i když radikálních důsledků, když uvažuje o umělecké tvorbě využívající stroje jako o **umění strojů**. Tvrdí, že „(s)kandál umění tvořeného stroji, od fotografie, videa, po počítač, odhaluje falešnost představy, že umění je lidskou záležitostí, místem lidské kreativity, jedinečné individuality.“⁷ Umění tvořené pomocí strojů podle něj poukazuje k celému autonomnímu světu strojů, který se přihlásil ke svým vlastním hodnotám a principům, nezávislým na hodnotách a principech lidské kreativity. Anglický název computer art, běžně překládaný do českého jazyka jako počítačové umění umožňuje, abychom tomuto typu tvorby rozuměli v intencích úvah Petera Weibela, tedy jako **umění počítačů**, ve smyslu výrazu strojové kosmologie, spíše než lidské kultury.⁸

Název počítačem generované umění (computer generated art) v sobě aspekt ne-lidské kreativity nenese. Váže se spíše ke konceptuálnímu chápání umění, k představě, že skutečně umělecké jsou myšlenky a jejich realizace je jen dodatečnou, mechanickou činností, kterou můžeme přenechat třeba i automaticky pracujícím strojům. Takovou argumentaci najdeme u konceptuálního umělce

⁵ V této souvislosti je třeba upozornit, že počítače vlastně negenerují náhodné výstupy, neboť počítač nefunguje na principu náhody. Viz komentář Ulrike Gabriel, softwarové umělkyně, který cituje Florian Cramer ve studii *Concepts, Notations, Software Art*. 23. března 2002, poznámka 1. Dostupné on-line: http://www.netzliteratur.net/cramer/concepts_notations_software_art.html (rev. 15. 10. 2012).

⁶ Weibel, Peter. *The Apparatus World – a World unto Itself*. D. Dunn (ed.) *Eigenwelt der Apparate – Welt. Pioneers of Electronic Art*. Katalog výstavy kurátorů W. Vasulky a S. Vasulky. Linec: Ars Electronica, 1992, s. 15 – 20, s. 18.

⁷ Ibidum, s. 18.

⁸ Srovnej s konceptem kulturního překódování Lva Manoviche, kerý tvrdí, že strojová logika, logika programovatelných médií, bude stále více ovlivňovat logiku kulturní. Viz Manovich, Lev. *The Language of New Media*. Cambridge: The MIT Press, 2002.

Sol LeWitta, který napsal, že „[m]yšlenka se stává strojem, který tvoří umění.“⁹ Můžeme proto sledovat analogie mezi počítačem generovaným uměním a konceptuálním uměním, v tvorbě spočívající ve vytváření skriptů, tedy algoritmů, které mají být důsledně a přesně následovány, ať už strojem (počítačem) nebo recipientem. Název počítačem generované umění sice poukazuje ke generativním procesům jako prostředkům vytváření umění, ale neasociuje onen magický a poetický aspekt hravé kreativity, s/tvoření ve stroji, který v sobě skrývá název umění počítačů.

Inženýr a pionýr počítačového umění Michael Noll¹⁰ je jedním z těch, kteří se snažili argumentovat ve prospěch tvrzení, že počítač je aktivní a kreativní médium, které můžeme využít i v oblasti umělecké tvorby. Napsal:

„Počítač je nepochybně elektronický nástroj schopný realizovat pouze ty operace, ke kterým byl explicitně instruován. To obvykle vede k představě, že počítač je mocný nástroj, který však není schopen žádné opravdové kreativity. Avšak pokud kreativita je chápána omezeně jako produkce nekonvenčního nebo nepředvídaného, potom by počítač měl být považován za kreativní médium – aktivního a kreativního spolupracovníka umělce.“¹¹

Noll představil počítač jako exaktně fungující, generativní nástroj nebo pomůcku, se kterou může umělec za jistých okolností dosáhnout esteticky hodnotných výsledků. Jeho názory jsou ve srovnání s Weibelovou vizí autonomního světa strojů umírněnější, ale nevybočují z dobového pojetí počítače jako ‚černé skříňky‘, kterou poznáváme a hodnotíme na základě výstupů, obrazů, zvuků či textů, které produkuje. O tom svědčí také jeho známý *The Mondrian experiment*¹², který měl dokázat, že počítač je schopný tvořit esteticky stejně působivé obrazy jako umělec.

Noll uskutečnil experiment zhruba napodobující Turingův test strojové inteligence, v němž využil *Kompozici v čarách* Pieta Mondriana (1917) a počítačem generovaný obraz tvořený pseudonáhodnými prvky, ale podobný svojí celkovou kompozicí Mondrianově malbě. Zatímco Mondrian umístil na svém obraze vertikální a horizontální čáry velice promyšleně a organizovaně, na počítačem generovaném obraze byly čáry umístěny podle pseudonáhodného generátoru čísel a s pomocí statistiky, která určila průměrnou hustotu, délku a šířku čar na Mondrianově obraze.

⁹ LeWitt, Sol. Paragraphs on Conceptual Art. A. Alberro – B. Samson (eds.). *Conceptual Art. A Critical Anthology*. Cambridge-London: MIT Press, 2000, s. 12 – 16, s. 12.

¹⁰ Michael Noll patří spolu s Friederem Nakem a Georgem Neesem k průkopníkům raného počítačového umění.

¹¹ Noll, Michael. *The Digital Computer As a Creative Medium*. IEEE SPECTRUM, Vol. 4, No. 10, October 1967, s. 89–95, s. 90.

¹² Noll, Michael. Human or Machina: A Subjective Comparison of Generated Picture. *The Psychological Rec.*, Vol 16, January 1966, s. 1 – 10.

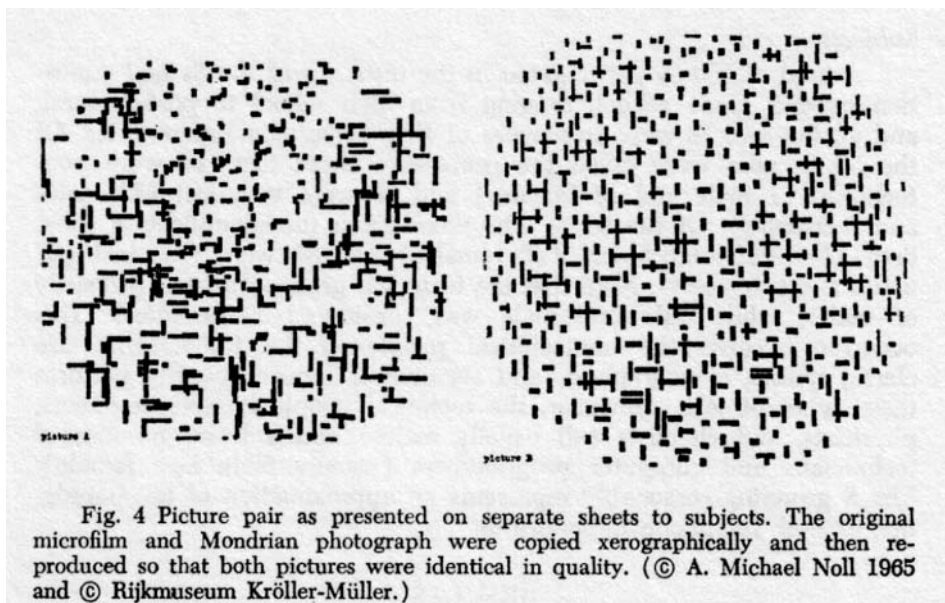


Fig. 4 Picture pair as presented on separate sheets to subjects. The original microfilm and Mondrian photograph were copied xerographically and then reproduced so that both pictures were identical in quality. (© A. Michael Noll 1965 and © Rijkmuseum Kröller-Müller.)

Obr. 1. Ukázka xerokopií Mondrianova obrazu (vpravo) a počítačem generovaného obrazu, které byly využity při tzv. **Mondrianově experimentu**.

Xerografické kopie obou obrazů byly následně předloženy stovce respondentů, jejichž vzdělání sahalo od středoškolského po postdoktorské studium. Tyto subjekty byly dotázány, kterému obrazu dávají přednost a který z dvojice obrazů podle nich namaloval Mondrian. Padesát devět procent dalo přednost počítačem generovanému obrazu a jen dvacet osm procent respondentů správně určilo obraz vytvořený Mondrianem. Dotazovaní lidé spojovali náhodnost počítačem generovaného obrazu s lidskou kreativitou, zatímco systematicky uspořádané čáry na Mondrianově obraze jim připadaly jako vytvořené strojem. Noll na základě výsledků experimentu dospěl k názoru, že počítač můžeme prohlásit za kreativní médium.¹³

Pro tuto etapu vývoje počítačového umění je příznačné, že místo, aby Noll seznámil čtenáře své studie s parametry programu, který je vlastně algoritmickým vyjádřením Mondrianova stylu, a s povahou procesů, jež tento program iniciuje, obrací pozornost recipientů ke konečnému výstupu, tedy k vytištěnému obrazu, jako důkazu kreativity počítače.

¹³ Noll, Michael. *The Digital Computer As a Creative Medium*. IEEE SPECTRUM, Vol. 4, No. 10, October 1967, s. 89–95, s. 91 – 92.

II Počítačové umění vs softwarové umění: 60s vs 90s

Na začátku 21. století, přesně v roce 2001¹⁴, se etabloval pojem **softwarové umění** jako označení určité oblasti umělecké tvorby využívající počítače a počítačovou síť přibližně od 90. let 20. století. Upřesnění definice a významu neologismu softwarové umění se od té doby věnuje řada předních teoretiků nových médií. Přesto, že někteří považují snahy o zavedení nového pojmu za redundantní, vnášející spíše zmatek do zavedené a dostatečně přesné terminologie¹⁵, domnívám se, že pozornost, kterou pojem softwarové umění mezi teoretiky vyvolal, nelze přehlížet. Naopak je zapotřebí, abychom se zaměřili na změny v pohledu na uměleckou tvorbu využívající počítač a na kritéria hodnocení, jež se s příklonem k pojmu softwarové umění dostávají do popředí.

Označení **softwarové umění** je výrazem změny v pojetí tvorby v prostředí digitálních médií, ke které došlo od 60. do 90. let dvacátého století. Názvy, jako počítačové umění a počítačem generované umění, poukazují k poetice typické pro první etapu vývoje umění digitálních médií spadající do šedesátých a sedmdesátých let 20. století. Počítačovní umělci a teoretici v tomto období nahlíželi na procesy uvnitř počítačů jako na metody či prostředky tvorby, nikoli jako na nedílnou součást díla nebo dokonce jako na díla sama o sobě. Počítačové umění mělo v tomto období blízko ke konceptuálnímu umění v tom ohledu, že těžiště práce umělce spočívalo ve vytváření konceptů (programů), ale na rozdíl od konceptuálního umění, se legitimita počítačového umění odvozovala od výsledků práce uměleckých programů, které se objevovaly na monitorech počítačů, ale ještě častěji vytištěné na papíře (viz *Mondrian experiment* Michaela Nolla). Právě tyto finální artefakty sloužily jako důkazy, na základě kterých se počítačové umění ucházelo o místo v rámci světa umění. Inspirační zdroje raného počítačového umění nacházíme v geometrické abstrakci, konstruktivismu, op-artu, minimalismu, konkretismu. Umělci využívající počítač jako kreativní médium však nereflektovali jiné dobové formy umění zaměřené na komunikační a politizující akce, jako happening, event, akce nebo performance. Lieser v této souvislosti připomíná:

¹⁴ O institucionálním etablování pojmu „softwarové umění“ můžeme mluvit od okamžiku, kdy byla v rámci berlínského uměleckého festivalu *transmediale.01* zavedena cena za umělecký software. V této souvislosti definovali softwarové umění dva členové poroty pro tuto cenu, Florian Cramer a Ulrike Gabriel, třetím porotcem byl John F. Simon Jr. Viz: Cramer, F. – Gabriel, U. *Software Art*. 15. srpna 2001. Dostupné on-line: http://www.netzliteratur.net/cramer/software_art_-_transmediale.html (rev. 15. 10. 2012).

¹⁵ Např. Weiss setrvává u pojmu počítačové umění, který považuje za historicky etablovaný a dostatečně výstižný. Viz Weiss, M. *Microanalyses as a Means to Mediate Digital Art*. In A. Bentowska-Kafel, T. Cashen, H. Gardiner (eds.) *CHArt Yearbook 2006: Futures Past: Thirty Years of Arts Computing*, Vol. 2, Bristol: Intellect Books, 2006, s. 13 – 24. Dostupné on-line: <http://www.chart.ac.uk/chart2004/papers/weiss.html> (rev. 15. 10. 2012).

Softwarové umění je většinou chápáno jako podkategorie digitálního umění. Viz např.: Lieser, W. *Digital Art*. China: h.f.ullmann, 2009.
Paul, Ch. *Digital Art*. New York – London: Thames & Hudson, 2008.

„První počítačové grafiky a animace byly tvořeny vědci nebo alespoň ve spolupráci s vědci. Jen velmi málo umělců užívalo k umělecké tvorbě počítač. Přiřazení umělce, že generuje umělecká díla na počítači dlouho znamenalo téměř jeho degradaci. Na začátku 70.let byl Manfred Mohr, který v roce 1971 uspořádal první výstavu počítačového umění v Musée d'Art Moderne de la Ville de Paris nazvanou *Une Esthétique Programmée*, zasypán rajčaty při své přednášce na Sorboně, protože používal ‚kapitalistický vojenský nástroj‘. V době, kdy byly v módě drogy a psychedelické umění bylo na vzestupu, konceptuální počítačové umění Manfreda Mohra bylo protipólem subjektivistického přístupu ostatních uměleckých forem.“¹⁶

Přibližně od 90.let mluvíme o **softwarovém umění**, které se vymezuje vůči počítačovému umění 60. a 70.let. Umělci a teoretici hlásící se k softwarovým studiím vytýkají pionýrům počítačového umění, že se podíleli na prezentaci počítače jako prostředku/médiu úniku, a zbavení se intencionality, jako nástroje k produkování neočekávané, arbitrární a nevyčerpatelné diverzity, odehrávající se jakoby vně lidské kultury¹⁷, a nevedli recipienty k otázkám po povaze procesů, které způsobují, že na obrazovce vidíme obrazy, slyšíme zvuky nebo čteme text. Právě v této skutečnosti nachází Weiss důvod toho, že současní umělci využívající počítače neodkazují na své předchůdce a průkopníky využití počítačů v umělecké produkci a mají potřebu se od této fáze vývoje odlišit dokonce i terminologicky.¹⁸

Počítačové umění a počítačem generované umění jsou pojmy svázané s tvorbou zaměřenou na výsledky generativních procesů, která je prezentována ve formě finálních produktů na obrazovce počítače, případně vytištěných pomocí plotterových a laserových tiskáren na papíře. Softwarové umění se naopak vyznačuje odklonem pozornosti od výsledků práce naprogramovaného média směrem k procesům spuštěným v počítači programovým kódem. Jinými slovy, software se stává nikoli prostředkem či nástrojem umělecké kreativity, ale je na něj nahlíženo jako na umělecké dílo nebo alespoň jako na jeho plnohodnotnou, z hlediska interpretace klíčovou, součást. Tilman Baumgärtel v článku *Experimentelle Software* shrnul změny v přístupu umělců k počítačům, které vyjadřuje pojem softwarové umění.

¹⁶ Lieser, W. *Digital Art*. 2009, s. 28.

¹⁷ Arns, I. *Read_me, run_me, execute_me*. Code as Executable Text: Software Art and its Focus on Program Code as Performative Text. In Tjark Ihmels – Julia Riedel (eds.) *Generative Tools*. 2004. Dostupné on-line: http://www.medienkunstnetz.de/themes/generative-tools/read_me/scroll/ (rev. 15. 10. 2012)

¹⁸ Viz Triebe, M. – Jana, R. *New Media Art*. Uta Grosenick (ed.) Köln: Taschen, 2006. Autoři této publikace umístili počátek umění nových médií do roku 1993 a spojili jej s dílem „jodi.org“, dekonstruovanou webovou stránkou autorů Joan Heemskerkové a Dirka Paesmanse. Umění nových médií chápou spíše jako hnutí, než jako umělecký druh definovaný zvolenou technologií. Za předchůdce umění nových médií označili dadaismus, readymades Marcela Duchampa, pop art a video art. Pionýry počítačového umění ze 60. a 70. let nezmiňují.

„Toto dílo není umění vytvořené pomocí počítače, ale umění, které se odehrává v počítači, není to software naprogramovaný umělcem k tomu, aby tvořil autonomní umění, ale software, který je sám uměleckým dílem. V případě těchto programů není důležitý výsledek, ale proces, který spustí v počítači (a někdy na monitoru).“¹⁹

V rámci tohoto přístupu je počítač chápán především jako programovatelné médium pracující v reálném čase, jehož podstatou je kód, umělý jazyk programovacích jazyků, a jeho zpracování, tedy počítačové procesy²⁰, jedním slovem software. **Software** je v kontextu softwarového umění interpretován jako kulturní produkt, jako jistý typ notace, partitury či skriptu, který je třeba analyzovat a esteticky hodnotit z různých hledisek: sám o sobě, jako výsledek práce a intencí programátora, v procesu zpracování, stejně jako z hlediska jeho vztahu k reálným i potenciálním realizacím a účinkům, které se neomezují jen na obrazovku počítače, ale pronikají do sféry kultury.

Kittler napsal, že „[k]ódy, svým jménem a svojí materialitou, jsou tím, co nás dnes určuje a tím, co musíme artikulovat, abychom zabránili úplnému zmizení pod nimi.“²¹ V intencích jeho požadavků vynášejí umělci a teoretici hlásící se k softwarovému umění principy fungování počítače na povrch, aby je podrobili analýze z hlediska estetického, stejně jako z hlediska jejich kulturních implikací a vyvolali o procesech odehrávajících se v a skrze počítače veřejnou debatu. Softwarové umění je tedy nejen aktivitou, na kterou můžeme nahlížet z estetického hlediska, ale i nástrojem zkoumání místa a role počítačů v naší společnosti.

Pandánem softwarového umění v oblasti teorie jsou softwarová studia. Vztah mezi softwarovým uměním a softwarovými studii není pouhou tautologií, ale přináší s sebou hledání nových metod výzkumu²² a nový definiční a historický rámec umění tvořeného v prostředí počítačů formovaný kolem pojmu software a principů reprezentace a produkce, ke kterým poukazuje.

¹⁹ Baumgärtner, T. Experimentelle Software. *Telepolis*. October 28, 2001. On-line: <http://www.heise.de/tp/artikel/9/9908/1.html> (rev. 15. 10. 2012).

²⁰ Komputace znamená fázi „systémové analýzy následující po kvantifikaci matematického modelu a zahrnující procesy spojené s prováděním výpočtů, zvláště na počítačích.“ Petráčková, V. – Kraus, J. a kol. *Akademický slovník cizích slov*. Heslo komputace. Praha: Academia, 2000, s. 406.

²¹ Kittler, F. Code (or, How You Can Write Something Differently). In M. Fuller (ed.), *Software studies \ A Lexicon*. Cambridge – London: The MIT Press, 2008, s. 40 – 47, s. 40.

²² Například Matthew Fuller se věnoval metodologii výzkumu kulturní historie nových médií ve studiích: Fuller, M. *Softness: Interrogability; General Intellect; Art Methodologies in Software*. Skriftserie. Center for Digital Aestetik-forskning. Nr. 13, 2006. Dostupné on-line: <http://darc.imv.au.dk/wp-content/files/13.pdf>(rev. 15.10.2012). Fuller, M. Faulty Theory. M. Goddard – J. Parikka (eds). *Fibreculture Journal, 17: unnatural ecologies*. 2011. Dostupné on-line: <http://seventeen.fibreculturejournal.org/> (rev. 15. 10. 2012).

III Software

Software je většinou chápán z čistě technického hlediska. Tomu odpovídá i definice pojmu software, kterou najdeme v *Akademickém slovníku cizích slov*. Zde se uvádí, že software je „vybavení počítače programovacím jazykem a programem, zkráceně programové vybavení počítače, na rozdíl od technického vybavení, které bývá označováno jako hardware.“²³ V *Oxford Dictionaries* se dočteme, že software jsou „programy a jiné provozní informace užívané počítačem“.²⁴ Je nám však nabídnuta i definice sociálního softwaru jako „počítačového softwaru, který umožňuje uživatelům interagovat a sdílet data.“²⁵, která se již od čistě formálního, technického popisu odpoutává směrem k charakteristice kulturní praxe spojené s rozšířením osobních počítačů. Pro naše účely postačí, když budeme software definovat jako soubor všech počítačových programů, procedur, algoritmů a dat používaných počítačem. Rozlišujeme systémový software, který zajišťuje chod samotného počítače a jeho interakci s okolím, a aplikační software, se kterým pracuje uživatel nebo propojuje počítač s nějakým jiným strojem.²⁶

Software / hardware

Význam pojmu software bývá často formulován jako opak hardwaru, tedy hmotných částí počítače, jako elektronických obvodů, skříní apod. Avšak ostrou hranici mezi systémovým softwarem a hardwarem můžeme vést pouze v rovině jazykové, fakticky je oddělit od sebe nelze. Florian Cramer argumentuje ve prospěch požadavku na zrušení myšlení v dichotomii nehmotného softwaru a materiálního hardwaru. Tvrdí, že „jestliže je zapotřebí zpochybnit dualitu softwaru a hardwaru, potom musí být zpochybněna i představa nehmotného softwaru vs. hmotného hardwaru.“ Podle Cramera je třeba rozpoznávat

„[r]ozdíl mezi materiálností a nehmotností v rámci softwaru samotného: algoritmus je hmotný v uložené, kódované formě a ve většině svých kulturních uplatnění. [...] Jestliže však umístění nebo dokonce existence některého softwaru není zcela jasná, jestliže softwarové dílo není kódem spuštěným ve stroji, protože se vyskytuje jako pseudokód v knize nebo dokonce není funkčním algoritmem, jako například většina Perl poezie, potom je technická definice softwaru příliš ome-

²³ Petráčková, V. – Kraus, J. a kol. *Akademický slovník cizích slov*. Heslo software. Praha: Academia, 2000, s. 699.

²⁴ *Oxford Dictionaries*. Oxford University Press, 2012. Hesla: software. On-line: <http://oxforddictionaries.com> (rev. 15. 10. 2012).

²⁵ Ibidum. Heslo: social software.

²⁶ Pro podrobný a dobře srozumitelný popis softwaru viz: *Wikipedia. The Free Encyclopedia*: heslo software v české a anglické mutaci. On-line: <http://cs.wikipedia.org/wiki/Software> a <http://en.wikipedia.org/wiki/Software> (rev. 15. 10. 2012).

žená. Můžeme říci, že ‚software‘ a ‚komputace‘ nemohou být striktně od sebe odlišeny. Kulturní historie softwaru je kulturní historií komputace.“²⁷

Oddělení softwaru a hardwaru však není jen důsledkem hry s významy těchto pojmů, ale odpovídá historickému vývoji počítačů, který vedl k emancipaci softwaru od hardwaru a jejich definování jako dvou na sobě nezávislých komerčních produktů.

Již od 60. a 70. let se v rámci odborného diskurzu kolem softwaru, stejně jako v subkultuře amatérů, vytvářely podmínky k proměně softwaru z marginálního, doslova bezcenného, doplňku počítačového hardwaru v jedno z nejprogresivnějších se vyvíjejících odvětví trhu s výpočetní technologií. V prosinci roku 1968 došlo k jedné z přelomových událostí tohoto trendu, když se firma IBM rozhodla pod tlakem vlády U.S.A., která jí hrozila antimonopolní žalobou, rozdělit svůj hardware a software. První odděleně prodávaný software byl CICS, Customer Information Control System, užívaný IBM ke zpracování finančních transakcí. Do té doby, přesně do roku 1969, firma poskytovala svůj software zdarma zákazníkům, kteří si koupili nebo pronajali její počítač. Hardware byl v této době velice drahý a software byl k němu přidáván jako součást jednoho balíku služeb.

Klíčovým momentem a současně precedentem v procesu osamostatnění softwaru od hardwaru byl rok 1985, kdy v rámci soudního sporu známého jako *Digidyne vs. Data General* Nejvyšší soud Spojených států amerických potvrdil rozhodnutí soudu nižší instance, že „odmítnutí Data General corp. licencovat jejich autorizovaný počítačový software těm, kteří nevlastní jimi vyrobený hardware bylo nezákonné.“²⁸ Soudce označil omezování licence softwaru operačního systému pouze pro DG hardware za nelegální chování, které znevýhodňuje konkurenci, omezuje volný trh a vede k jeho monopolizaci. Software od tohoto momentu představuje samostatnou entitu, komoditu vlastněnou společnostmi, které ji produkují. Tato událost je mezníkem, který vývoj softwaru posunul do další fáze umožňující rozvoj softwarových aplikací.²⁹

Software: genealogie

První doklad o užití slova software máme podle nejnovějších výzkumů již z roku 1850, kdy se vyskytuje ve zcela jiném smyslu a kontextu, než v jakém je používán dnes. Slova „soft-ware“ a „hard-ware“ se v této době používala pro roz-

²⁷ Cramer, F. *Words Made Flesh. Code, Culture, Imagination*. Rotterdam: Piet Zwart Institute: 2005, s. 124. On-line PDF kniha: <http://www.netzliteratur.net/cramer/wordsmadefleshpdf.pdf> (rev. 15. 10. 2012).

²⁸ Viz článek: Myers, G. Tying Arrangements and the Computer Industry: Digidyne Corp. v. Data General Corp. *Duke Law Journal* Vol. 1985, No. 5 (Nov., 1985), pp. 1027-1056. Published by: Duke University school of Law, s. 1027. Dostupné on-line: <http://www.jstor.org/stable/1372482> (rev. 15. 10. 2012).

²⁹ Více k vývoji softwaru: Philipson, G. *A Short History of Software*. Routledge: 2004. Dostupné on-line: <http://www.thecorememory.com/SHOS.pdf> (rev. 15. 10. 2012).

lišení dvou typů odpadu: soft-ware byl organický odpad, který se může rozložit a hard-ware všechno ostatní.³⁰ První užití slova software ve smyslu počítačových programů je datováno do roku 1958, kdy jej užil John W. Tukey v článku *The Teaching of Concrete Mathematics* publikovaném v časopise *American Mathematical Monthly*³¹. Tukey zde popisuje význam matematických a logických instrukcí pro elektronické počítače a výhody využívání výpočetních pomůcek při výuce matematiky. Slovo software se u Tukeye objevuje v podkapitole nazvané *Attitudes toward computation* v tomto kontextu: „Dnes ‚software‘, tvořený pečlivě plánovanou výkladovou praxí, překladači a dalšími aspekty automatického programování, je pro moderní elektronické kalkulátory minimálně stejně důležitý jako ‚hardware‘ trubic, tranzistorů, drátů, pásek a podobně.“³²

Software – poznámka G

První počítač v moderním smyslu slova byl ENIAC, Electronic Numerator, Integrator, Analyzer and Computer, navržený v roce 1942 Johnem W. Mauchlym a vyvinutý v U.S.A. v posledních letech 2. světové války. První software však datujeme již o sto let dříve, do Anglie 19. století. Jeho autorka, Augusta Ada King, hraběnka z Lovelace, zkráceně Ada Lovelace, dcera anglického romantického básníka George Gorgona Byrona, jej navrhla pro analytický stroj (1834) Charlese Babbage, považovaný za první koncept počítače.³³

Na rozdíl od předchozích strojů usnadňujících provádění matematických výpočtů, například Babbageova diferenčního stroje (1822), byl analytický stroj programovatelný, jinými slovy, byl schopen samočinně plnit série instrukcí zaznamenaných na děrovacích štítcích. Tento způsob ovládání stroje převzal Babbage od Francouze Josepha Marie Jacquarda, který v roce 1804 dokončil svůj tkalcovský stav řízený sledem děrovacích štítků.³⁴

-
- 30 Peterson, I. Software's Origin. *Science News*, 11.3. 2011.
On-line časopis: http://www.sciencenews.org/view/generic/id/698/title/Softwares_Origin (rev. 15. 10. 2012).
- 31 Tento údaj zveřejnil knihovník Yale Law School v New Havenu, Conn., Fred R. Saphiro, který hledal slovo ‚software‘ v elektronicky archivovaných akademických časopisech v databázi JSTORE (Journal STORage). Dle Peterson, I. Software's Origin. *Science News*, 2011.
- 32 Tukey, J. W. The Teaching of Concrete Mathematics. *American Mathematical Monthly*. Vol. 65, No. 1 (Jan., 1958). Vydavatel: Mathematical Association of America. pp. 1–9, s. 2.
On-line: URL: <http://www.jstor.org/stable/2310294> (rev. dne 15. 10. 2012).
- 33 Viz např. Augarten, S. *Bit by Bit: Illustrated History of Computers*. HarperCollins Publishers Ltd, 1985, s. 44.
- 34 Sám Babbage vlastnil mechanicky tkaný hedvábný portrét, zobrazující Josepha M. Jacquarda držícího kružítka, symbol matematických výpočtů, a sedícího před malým modelem svého stavu, na jehož výrobu bylo užito 20 000 štítků.
Dle: Batchen, G. *Elektricity Made Visible*. Wendy Hui Kyong Chun, Thomas W. Keenan (eds.). *New Media, Old Media: A History and Theory Reader*. New York – Oxon: Routledge, 2006, p. 27–44, s. 31.

Ada Lovelace se s Babbageem setkala přibližně v roce 1833, když se svojí matkou Lady Anabellou Byron navštívila jeho proslulý londýnský salón. Tato návštěva ji údajně natolik okouzila, že se zaměřila na studia matematiky a později se také stala Babbageovou osobní asistentkou a překladatelkou.³⁵

V roce 1842 vydal italský matematik Luigi Federico Menabrea článek ve francouzštině, ve kterém shrnul hlavní rysy Babbageova konceptu analytického stroje. Ada Lovelace tento text přeložila do angličtiny a opatřila jej na Babbageovu žádost poznámkovým aparátem, který byl nakonec třikrát delší než Menabreův text a stal se, spolu s konceptem analytického stroje, významným přínosem k historii výpočetní techniky.³⁶

Těžiště příspěvku Ady Lovelace k historii softwaru se skrývá zejména v poznámce G, ve které popsal metody výpočtů a diagram algoritmu, který nastaví analytický stroj tak, aby počítal Bernoulliho čísla. Tato pasáž je považována za první popis počítačového programu, tedy za první software. Lovelace zde také vysvětluje, proč si pro svoji ukázkou počítačové síly analytického stroje zvolila právě Bernoulliho čísla. Uvádí, že je to sice „poměrně komplikovaný příklad [...], ale vhodný, neboť nám umožní ukázat rozdíl mezi Babbageovým předchozím strojem, diferenčním strojem, jako pouhým počítacím strojem (calculation machine), a jeho vylepšeným analytickým strojem, který je univerzální počítačový stroj (computing machine).“³⁷ Výjimečnost analytického stroje rozpoznala v tom, že vyznačil přechod od výpočtů (kalkulace) k obecnému computingu (general-purpose computing), přechod od strojů schopných pouze uspořádat čísla k programovatelnému univerzálnímu stroji, schopnému manipulovat se symboly podle příkazů a generovat cokoli jiného, ať je to hudba, poezie, nebo obrazy.³⁸ Představovala si, že analytický stroj „splétá algebraické vzorce tak jako Jacquardův stav splétá květiny a listy.“³⁹

Někteří teoretici zpochybňují skutečný přínos Ady Lovelace k rozvoji výpočetní technologie. Nemohou jí však upřít schopnosti rozpoznávat poetickou a metafyzickou dimenzi fungování stroje schopného automaticky provádět složité výpočty. Přínos této „kouzelnice čísel“, jak ji nazýval Babbage, k historii výpočetní techniky je dnes spatřován především v rozvíjení úvah, které procházely napříč matematikou, vynálezectvím, poezií, teologií, obory, které chápala jako součást jednoho celku, tzv. „poetické vědy“.⁴⁰

³⁵ Ibidum.

³⁶ Menabrea, L. F. *Sketches of the Analytical Engine Invented by Charles Babbage*. A. Lovelace trans. and notes. *Scientific Memoirs* 3, 1843, s. 666 – 731.

³⁷ Citováno dle: Krysa, Joasia. *Ada Lovelace. Introduction / Einführung*. DOCUMENTA (13). 100 Notes – 100 Thoughts, No. 055. Ostfildern: Hatje Cattz Verlag, 2012, s. 3.

³⁸ Ibidum, s. 3.

³⁹ Ibidum, s. 3. Viz: Menabrea, L. F. *Sketches of the Analytical Engine Invented by Charles Babbage*. A. Lovelace trans. and notes. *Scientific Memoirs* 3 (1843), pp. 666 – 731, Note A, pp. 691, s. 694.

⁴⁰ Krysa, Joasia. *Ada Lovelace. Introduction / Einführung*. 2012, s. 4.

„Přínosem studia matematiky [...] je nesmírný rozvoj představivosti: a to natolik, že pokud

Meta-software

První teoretický popis fungování počítačového programu, neboli softwaru, který bezprostředně ovlivnil podobu moderních počítačů napsal Alan M. Turing ve studii *On Computable Numbers with an Application to the Entscheidungsproblem* publikované v roce 1936 a v upravené verzi v roce 1937. Popisuje v ní teoretický koncept Univerzálního stroje, později nazvaného Turingův univerzální stroj nebo prostě Turingův stroj, schopného zpracovat jakýkoli matematicky vyjádřitelný příkaz. Jeho základem je nekonečně dlouhá páska rozdělená na políčka, do kterých mohou být zapisovány, čteny a mazány matematické symboly. V kapitole nazvané *Computing machines* tento stroj popsal:

„[...] je vybaven ‚páskou‘ (podobnou papíru), která jím prochází, a je rozdělená do částí (zvaných ‚rámečky‘) schopných něco ‚symbolizovat‘, vždy je jen jeden rámeček [...] ‚ve stroji‘. Můžeme tento rámeček nazvat ‚skenovaný rámeček‘. Symbol na skenovaném rámečky můžeme nazvat ‚skenovaný symbol‘. Stroj si je ‚vědom‘ pouze právě skenovaného rámečku, ale je schopen si také ‚zapamatovat‘ některé symboly, které ‚viděl‘ před tím.“⁴¹

Koncept univerzálního stroje výrazně ovlivnil rané úvahy o povaze komputace a podobu prvních elektronických počítačů. Turingovy symboly byly vlastně matematické funkce, přičti, odečti, vynásob atd., a jeho představa, že můžeme zredukovat jakoukoli složitou operaci na sérii jednoduchých kroků je podstatou veškerého programování.

Turing prokládá své exaktní matematické argumenty metaforami, které naznačují možnost prostupnosti hranice mezi konceptem lidské mysli a konceptem počítače. Napsal, že: „[L]idská paměť je nutně omezená. Můžeme srovnat člověka, který počítá reálná čísla se strojem, který je schopen pouze konečného množství stavů.“⁴² A dále dodává, že stroj si je “přímo vědom”, případně, že “stroj si může efektivně pamatovat některé symboly, které viděl [...] dříve“ nebo hovoří o „možném chování strojů“.⁴³ Počítač tedy chápe jako model inspirovaný činností lidské mysli ve stavu „počítání“, computingu, který je schopen podávat mnohem vyšší a přesnější výkony než jeho předobraz, člověk. Ve svém dalším

dokončím svá studia, mohu se v příhodný čas stát básnířkou. Takový výsledek se může zdát zvláštní, nikoliv však pro mne. Věřím, že jasně chápu jeho příčiny a souvislosti.“píše Ada Lovelace v dopise své matce. Citováno dle: Batchen, G. *Electricity Made Visible. New Media, Old Media: A History and Theory Reader*. 2006, p. 27- 44, s.32.

V dopise Babbageovi tuto myšlenku opět rozvádí: „Nevěřím, že můj otec byl (nebo kdy mohl být) takovým Básníkem, jakou já budu Analytičkou (a Metafyzičkou); protože pro mne je obojí spolu nerozlučně spjato.“ Ibidum, s. 32.)

41 Turing, A. M. *On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society*. Ser. 2, Vol. 42, 1937, s. 230–265, s. 231.

42 Ibidum, s. 231.

43 Ibidum, s. 231.

textu, příznačně publikovaném ve filozofickém časopise *Mind*⁴⁴, se Turing již zcela otevřeně ptá „Mohou stroje myslet?“, čímž jen rozšiřuje otázku „Mohou stroje počítat?“, které se vlastně věnoval v předchozí studii, o další typicky lidské dovednosti. V tomto textu se tedy zakládá kybernetický diskurz rozvíjející vztah analogie mezi člověkem a informační technologií, založený na předpokladu, že lidské myšlení, paměť, kreativita a rozhodování mohou být modelovány počítačovými programy.

Z analogií, které Turing užívá, vyplývá, že předobrazem moderních počítačů, není jakýkoli člověk, ale je to „počítající člověk“ (he/she copmuter). Je to účetní, knihovník, nebo úředník. Warren Sack popisuje povahu aktivity, kterou takový člověk vykonává: „pobíhá po účtárně, přehazuje stohy mřížkovaného papíru, čte, píše a maže čísla v malých okénkách.“⁴⁵ Paměť tohoto úřednického, škatulkujícího typu, třídícího data do složek, adresářů, seznamů, atd., je hlavní oblastí výzkumu a vývoje hardwaru a softwaru. Jedná se tedy o poměrně specifický druh paměti a kognitivních operací, který je v paměti počítače modelován.

Freaks of Number

Myšlení v analogiích propojujících aktivity počítačové a lidské myslí si osvojil také Mathew Fuller, který napsal, že počítač je, stejně jako jeho uživatelé, „freak of number“.⁴⁶ Zatímco Babbage nazýval Adu Lovelace svojí „kouzelnicí čísel“, jako výraz obdivu k její schopnosti splétat matematické vzorce s poetickými a mystickými vizemi, Fuller přirovnává aktivitu počítačů k podivínskému chování zvláště nadaných lidí, spočívajícímu v monomaničtém, abnormálně rychlém a extrémně přesném počítání. V této souvislosti upozorňuje na knihu Maurice d’Ocagne *le Calcul simplifié par les procédés mécaniques et graphiques* z roku 1894, s podtitulem „Historie a popis nástrojů a strojů na výpočty, tabulky, abakusy a nomogramy“. Jedná se podle něj zřejmě o jeden z nejranějších případů kritického zkoumání a popisu fungování softwaru neboli meta-softwaru. D’Ocagne totiž vytvořil komparativní portrét všech v jeho době dostupných počítačích přístrojů a technik. Vedle objevů Pascala, Babbage, Leibnitze nebo Napiera zde najdeme i soubor každodenních pomůcek, jako registrační pokladny, aritmetiky atd.

V úvodní pasáži publikace, ve které se autor snaží přesvědčit čtenáře o významu aritmetiky pro různé obory a disciplíny, je zařazen také seznam geniálních počtářů, které můžeme nazvat „freaks of number“. Většinou se jedná o „zázračné děti“, pastevce ovcí nebo otroky, u kterých byly tyto zvláštní schopnosti roz-

⁴⁴ Turing, A. Computing Machinery and Intelligence. *Mind*, LIX (236), 1950, p. 433-460.

⁴⁵ Sack, W. Memory. In M. Fuller (ed.), *Software Studies \ A Lexicon*. Cambridge - London: The MIT Press, 2008, s. 184-192, s. 189.

⁴⁶ Fuller, M. Freaks of Number. In G. Cox & J. Krysa (eds.). *Engineering Culture: 'The Author as (Digital) Producer'*. New York: Autonomedia (DATA browser 02), 2005. On-line: <http://www.spc.org/fuller/texts/freaks-of-number/> (rev. 15. 10. 2012).

poznány. Seznam podivínů je zvláštním prologem ke knize věnované prvnímu pokusu o taxonomii automatických výpočetních zařízení.

Fuller tuto konstelaci interpretuje jako moment exkluze jevu, kterého by si měl být čtenář knihy vědom, žasnout nad ním, ale současně jej odmítat. Zřící se těchto podivínů ve prospěch „zkrocené“ síly matematiky v podobě transparentních mechanismů počítačích strojů. Zatímco původ schopností geniálních matematiků zůstává záhadou, „anatomii počítačích strojů“, jak svoji metodu nazval d’Ocagne, se můžeme věnovat do nejmenších detailů. V důsledku přemístění výjimečné dovednosti „záračných počtářů“ do počítačích strojů tato funkce hypertrofuje, a to jak z hlediska intenzity, zrychlení a zpřesnění výpočtů, tak i z hlediska její extenze, neboť přestává být výjimečnou záležitostí ze společnosti vydělených jedinců a šíří se společností spolu s rozšířením využívání výpočetních strojů.

Matematicko-materiální síla počítačů

V úvodním gestu d’Ocagnovy knihy rozpoznává Fuller zárodek „matematicko-materiální síly“⁴⁷, která s nástupem průmyslové revoluce v 19. století začala vládnout nad věcmi i lidmi. Mluvíme o zrození statistiky a o snaze s její pomocí regulovat a kontrolovat populaci z hlediska zdraví, kriminality, porodnosti, úmrtnosti, sňatků, stejně jako třeba produkci zboží.⁴⁸ Postindustriální společnost můžeme potom definovat jako vývojovou etapu, ve které dochází k fúzi praxe statistik, jako nástroje řízení společnosti, vedoucí k obrovskému bujení aktivit typu shromažďování faktů, tabelace a zaznamenávání, s výpočetní silou počítačů převádějících data do řádu jedniček a nul.

Fuller charakterizuje povahu této matematicko-materiální síly jako spojení dvou sil: numerace, tedy číselné reprezentace vlastností skutečných věcí, dynamik nebo objektů, vedoucí k jejich abstraktnímu ztělesnění, s chováním světa hmotných objektů. Píše:

„Matematicko-materiální síly jsou poprvé generovány v momentu, když hmota je formována podle svého vědecky formulovaného matematického modelu. Jedním z důsledků této objektivizace je Standardní Objekt, modulární komponent typický pro globalizovaný trh. [...] Všechno, od hranolků po pizzy, je [...] předmětem přísných smluv a procesů standardizace.“⁴⁹

Standardizace produktů je typickým důsledkem průmyslové výroby. V další fázi, tzv. postindustriální společnosti, dochází k abstrahování procesu standardizace ve jménu zvyšování produkce. Produkt, který má být vyráběn, je nejdříve skenován, abstrahován a multiplikován s pomocí počítače. V momentu, kdy se

⁴⁷ Ibidum, s. 2.

⁴⁸ Ian Hacking užívá slovní spojení „lavina čísel“ k popisu okolností vzniku statistiky a snah mapovat a řídit různé aspekty vývoje populace industriálních společností. Viz: Hacking, I. *The Taming of Chance*. Cambridge: Cambridge University Press, 1990.

⁴⁹ Fuller, M. *Freaks of Number*. 2005, s. 2.

předmět promění v čísla zaznamenaná jako vzorec, odděluje se od svého tvůrce a tento vzorec je dále zpracováván strojem. Dochází k obrovské akceleraci produkce v rámci standardizovaných parametrů.⁵⁰

Svědnost programování, hraní počítačových her, zpracování textů, obrazů a zvuků v prostředí počítačů, spočívá v nesmírně snadné možnosti manipulovat světem uzavřeným v počítači. Fuller však upozorňuje, že možnost manipulovat objekty v počítači není výrazem svobody uživatele, ale jedná se o předem naprogramovaný soubor možností, fontů písma, kontrastu, zakřivení linií, který programuje způsob chování uživatele.⁵¹

Softwarové umění můžeme chápat, s ohledem na výše napsané, jako průzkum limitů svobody/kreativity uživatelů počítačů v rámci souboru pravidel a možností, které výpočetní technika nabízí. V této souvislosti můžeme mluvit o softwarovém umění také jako o extrémním programování nebo o programování excesů ve smyslu pohybu přes hranice pravidel, funkcí, či limitů vepsaných do softwaru i hardwaru počítače.

IV Softwarové umění: Extrémní programování: Programování excesů

V rámci softwarových studií je kreativní aktivita uživatelů počítačů chápána širě nebo radikálněji než pouhé „pavlovovské“ klikání myši a tvorba v mezích určených uživatelským rozhraním. Wardrip-Fruin navrhuje kriticky zkoumat operační logiku systémů.⁵² Fuller navrhuje, abychom apropriovali počítačové programy a aktivně se podíleli na jejich vytváření a přetváření.⁵³ Cramer zase poukazuje k bohaté kulturní historii kódu a kombinatorických technik, aby tak zpochybnil nároky na monopol kódu a programu počítačového a současně naznačil možnosti jeho transgrese směrem k metafyzickým a poetickým dimenzím.⁵⁴

Úlohou softwarového umění je poukazovat k fungování softwaru ve smyslu formálních instrukcí a jejich uskutečnění, které jsou podstatou digitálních médií. Umělci často pracují v hraniční zóně programování, neboť právě hranice fenoménu nám dovolí nahlížet jeho vlastnosti v podobě čisté funkcionality principů, které ztělesňuje. Příklady softwarového umění, které dále představím, poukazují k praxi softwarového umění jako k pohybu v limitních oblastech programování.⁵⁵

⁵⁰ Ibidum.

⁵¹ Ibidum,

⁵² Wardrip-Fruin, N. *Expressive Processing. Digital Fiction, Computer Games, and Software Studies*. Cambridge – London: Themis Press, 2009.

⁵³ Fuller chápe softwarové umění v těsném vztahu s tendencemi, které dostávají jména jako free software, open source software, navazujícími na hackerský étos pionýrů kyberprostoru.

⁵⁴ Cramer, F. *Words Made Flesh. Code, Culture, Imagination*. 2005.

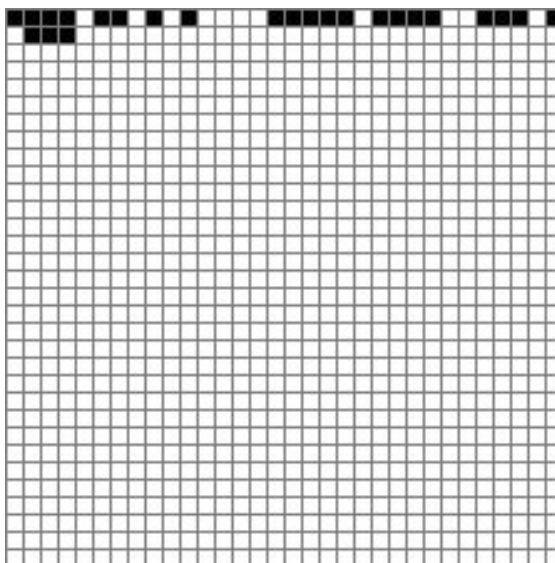
⁵⁵ Různé podoby softwarového umění jsou uloženy na stránkách *Runme.org* založené v roce 2003 a koncipované jako repozitář softwarového umění. Stránky spravují Amy Alexander, Olga Gorjunova, Alex McLean a Alexej Šulgin. Dostupné on-line: <http://runme.org/> (rev. 15. 10. 2012).

Every Icon

Dnes již klasické dílo z roku 1996/97 Johna F. Simona Jr., jednoho z členů komise pro softwarové umění berlínského Transmediale v roce 2001. *Every Icon* je počítačovou interpretací myšlenky Paula Klee, který si zaznamenal svoji snahu prozkoumat kombinatorické možnosti vzorců a struktur s pomocí zaplňování čtverců mřížky nakreslené na papíře.⁵⁶ Simon přepsal tuto myšlenku do jednoduchého programu, který může být spuštěn na webovém prohlížeči:

Given: A 32 X 32 Grid
Allowed: Any element of the grid
to be black or white
Shown: Every Icon

Every Icon se vyvíjí prostřednictvím výpočtů. V počátečním stavu vidíme mřížku, jejíž čtverce jsou všechny bílé. Software postupně prezentuje všechny kombinace černých a bílých prvků až do okamžiku, kdy bude celá mřížka tvořena jen černými čtverci. Celkové množství černých a bílých čtverců v mřížce 32 x 32 je 1.8×10^{308} . To znamená, že při rychlosti 100 čtverců za sekundu, což je typická rychlost osobního počítače, bude trvat 1.36 roku, než se zobrazí všechny varianty prvního řádku mřížky. Vyčerpání všech variant kombinací s druhým řádkem bude trvat exponenciálně déle, 5.85 bilionů let.



Obr. 2. John F. Simon Jr. *Every Icon*, 1997.
<http://www.numeral.com/eicon.html>

⁵⁶ Triebe, M. – Jana, R. *New Media Art*. Uta Grosenick (ed.) 2006, s. 86.

Simon se vzdává možnosti výběru několika vygenerovaných obrazů na základě estetického soudu, jak tomu bylo v případě počítačového umění šedesátých let, ve prospěch představení všech možností, virtualit, které v sobě tato mřížka 32 x 32 a barvy černá a bílá nesou. Napsal, že: „V protikladu k prezentování jednoho obrazu jako intencionálního znaku, představuje Every Icon všechny možnosti“.⁵⁷

Every Icon je příkladem jakési monstrózní numerické grotesky, kdy na základě jednoduchého příkazu je spuštěno kombinatorické šílenství. Every Icon je programem, který, vzhledem k trvanlivosti lidských výtvorů, nebude nikdy plně realizován. Dílo můžeme konceptuálně ocenit, ale jeho extrémní trvání nám neumožňuje je plně smyslově vnímat ani plně realizovat. Jde tedy také o příklad konceptuálního programování.

Právě v oblasti konceptuálního umění můžeme nalézt řadu předchůdců softwarového umění. Známým příkladem, užívaným v této souvislosti je *Composition 1961 No. 1, January 1* od současného skladatele a bývalého člena hnutí Fluxus La Monte Younga, která je považována za první dílo minimalistické hudby a jednu z prvních partitur performancí příslušníků hnutí Fluxus.

„Draw a stright line and follow it.“

Toto dílo se konceptuálně váže k softwarovému umění, neboť se jedná o formální instrukci neboli algoritmus. Je rovněž extrémní ve svém estetickém důsledku, v implikaci nekonečného prostoru a času, kterým se máme pohybovat. Na rozdíl od většiny notační hudby a napsaných divadelních her, tato partitura není esteticky oddělena od své performance, stejně jako počítačové programy. Čára, která má být nakreslena, může být dokonce chápána jako instrukce druhé úrovně určená k následování. Avšak, stejně jako Every Icon, i tuto partituru/program je prakticky nemožné realizovat fyzicky. Stejně jako Simonovo dílo, je to dílo nejen konceptuální, poukazující k limitům lidského vnímání, ale i dílo metafyzické.

Self

Jako další příklad extrémního programování jsem vybrala dílo umělce a teoretika médií Floriana Cramera *Self.pl.* z roku 2001. Jedná se o minimalistickou uměleckou partituru napsanou v jazyce Perl, která je současně počítačovým programem i básní. (obr. 3).

Algoritmus programu *Self.pl* můžeme vyjádřit v přirozeném jazyce angličtiny jako: „open it self while it push at it close it open it self print it join at it close it.“⁵⁸ Tuto softwarovou báseň můžeme celkem dobře číst, protože obsahuje řadu prvků

⁵⁷ Autorský popis konceptu a realizace díla od Johna Simona, jr.: <http://www.numeral.com/articles/paraicon/paraicon.html> (rev. 15. 10. 2012).

⁵⁸ Horst, Ph. Code Poetry's Aesthetic WW(WEB). In Freitag, K. – Vester, K. (eds.) *Another Language: Poetic Experiments in Britain and North America*. LIT Verlag, 2008. s. 105.

```
#!/usr/bin/perl
open (IT, "< self");
while (<IT>) {
  push @it, $_}
close (IT);
open (IT, ">> self");
print IT join ("\n ", @it);
close (IT);
```

Obr. 3. Florian Cramer: *Self.pl*, 2001.

<http://www.uweb.ucsb.edu/~mweismann/art7d/software.html>

přirozeného jazyka. Současně jde však o zápis příkazů adresovaných počítači. Když tento konkrétní program/partituru spustíme v počítači, vytvoří prázdnou složku, kterou nazve „self“; následně program přečte obsah této složky, která je však prázdná, zavře ji a znovu otevře, aby zapsal, co právě přečetl, tedy nic. Nakonec vytiskne prázdnou stránku. Kódovou báseň *Self.pl* můžeme interpretovat také jako literární autoportrét, jako performance „self“, které píše sebe sama. Současně je to však minimalistický program, který svojí jednoduchostí a poetičností kontrastuje se sofistikovanými aplikacemi typickými pro počátky umění virtuální reality, stejně jako je polemikou s komerčními počítačovými aplikacemi skrývajícími svoji kódovou podstatu za obrázkovým rozhraním.

Cramer tvrdí, že kódovou báseň/partituru můžeme číst třemi způsoby: jako báseň napsanou v angličtině, jako sekvenci počítačových příkazů a, v momentu kdy je performována počítačem a produkuje text, jako novou báseň v angličtině.⁵⁹ Není to tedy jen poetický text, ale jedná se o performanci sebe-reflexivity kódu odhalující performativní principy fungování počítačových programů, které jsou scénářem a jeho uskutečněním současně.

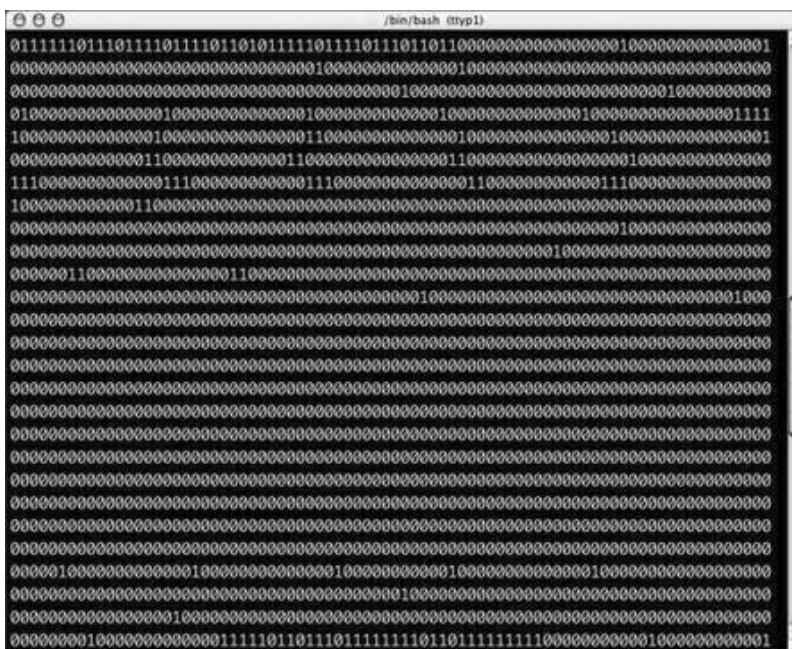
Forkbomb

Ve slovníku hackerů označuje slovo „forkbomb“ program, který pracuje na principu větvení, kdy jeden spuštěný proces generuje spuštění dalších procesů. Spouštění dalších a dalších procesů je tak rychlé, že brzy dochází k přetížení operačního systému počítače a k jeho zamrznutí. Něco podobného znají běžní uživatelé, kteří spustí ve svém počítači příliš mnoho aplikací najednou, a počítač se zasekne. „Forkbomb“ je jedním z nejoblíbenějších žánrů hackerské produkce zaměřené na psaní elegantních, tj. jednoduchých a efektivních, kódů, který se stal velmi populárním někdy v polovině 90.let minulého století. Jedna taková „softwarová bomba“ byla v roce 2002 na festivalu *transmediale.02* v Berlíně oceněna

⁵⁹ Ibidum, s. 103.

v sekci softwarové umění.⁶⁰ Byla to *Forkbomb* (2001/2005) Alexe McLeana napsaná v jazyce Perl.⁶¹

Výstupem asi třiceti řádkového programu *forkbomb.pl* jsou binární data, jakoby náhodně uspořádané jedničky a nuly na černém pozadí, vytvářející dojem jakési minimalistické partitury (viz obr. 4). Podobu výstupu na monitoru určuje jednak algoritmus reprezentovaný v kódu a částečně operační systém, ve kterém je program spuštěn. Počítačový operační systém je ve stavu neustálé změny, proto tento scénář bude produkovat vždy jiné výstupy. McLean interpretuje tyto výstupy jako výraz uměleckého dojmu z daného systému ve stavu přetížení.



Obr. 4. Alex McLean: *Forkbomb.pl*, 2001/2005.

Zřejmě nejelegantnější *forkbomb* napsal italský hacker, politický aktivista a autor free softwarů, který si říká Jaromil. Jeho *forkbomb* tvoří pouze třináct znaků (viz obr. 5):

⁶⁰ Viz: Vyhlášení výsledků sekce softwarové umění na stránkách festivalu: <http://archive.transmediale.de/en/02/awardnom.php?sect=2> (rev. 15. 10. 2012).

⁶¹ Program a popis projektu od A. McLeana: <http://backpan.perl.org/authors/id/F/FO/FO-OCHRE/forkbomb.pl> (rev. 15. 10. 2012).



Obr. 5. Jaromil: *ASCII Shell forkbomb*, 2002.⁶²

Tyto znaky vypadají na první pohled jako řada emotikonů nebo prostě jen chaotický zápis. Pokud je však zapíšeme do operačního systému typu Unix, spustí řetězovou reakci, forkbomb. Autor toto dílo poprvé představil spolu s textem *Digital Boheme*⁶³ v rámci výstavy *I Love You*⁶⁴ (2002, 2003).

ILOVEYOU

Pro Jaromila je softwarové umění především výrazem krásy kódu a mistrného ovládnutí umění programování. Toto extrémní programování chápe jako aktualizovanou podobu literární tradice nesené prokletými básníky nebo beatniky. Ke své tvorbě napsal:

„Chápu zdrojový kód jako literaturu, popisují [počítačové] viry jako by byly typem básní psaných Verlainem, Rimbaudem a dalšími proti těm, kteří prodávají síť jako bezpečnou oblast přímé společnosti. Vztahy, síly a zákony ovládající digitální sféru se liší od těch přirozených. Digitální sféra produkuje formy chaosu – což je nevhodné, protože je to neobvyklé a produktivní – kterým mohou lidé surfovat. V tomto chaosu jsou viry spontánní uspořádání, která podobně jako lyrické básně spouštějí nedokonalosti ve strojích a reprezentují vzpuru našich digitálních nevolníků (serfs).“⁶⁵

Přirovnání počítačového kódu a literatury je funkční jen do určité míry. Můžeme sice říci, že digitální média jsou doslova psána. Tyto zápisy, kódy, jsou však

⁶² Více o tomto díle zde: http://jaromil.dyne.org/journal/forkbomb_art.html (rev. 15. 10. 2012).

⁶³ Jaromil: Digital Boheme. In *I LOVE YOU*, 2002. Katalog výstavy. On-line: http://web.archive.org/web/20081205025500/http://www.digitalcraft.org/index.php?artikel_id=292 (rev. 15. 10. 2012).

⁶⁴ Výstava *I love you – computers_viruses_hacker_culture* kurátorky Franzisky Nori byla nejdříve prezentována v Muzeu užitého umění ve Frankfurtu nad Mohanem od května do června roku 2002. Rozšířená verze výstavy byla představena v „Haus der Kulturen der Welt“ v rámci berlínského *transmediale.03* (31. 1. 2003 – 6. 2. 2003). Katalog výstavy je dostupný zde: http://web.archive.org/web/20091025190232/http://www.digitalcraft.org/index.php?artikel_id=284 (rev. 15. 10. 2012).

Výstava *I LOVE YOU_computer_viruses_hacker_culture* byla nazvaná podle prozatím nejagresivnějšího počítačového viru šířeného elektronickou poštou od 5. května 2000 ještě několik dalších dní. Virus maskovaný jako valentýnské přání, napadl několik desítek milionů počítačů vybavených operačním systémem Microsoft Windows. Více o viru: <http://en.wikipedia.org/wiki/ILOVEYOU> (rev. 15. 10. 2012).

⁶⁵ Jaromil: Digital Boheme. In *I LOVE YOU*, 2002. Katalog výstavy.

```

filename="LOVE-LETTER-FOR-YOU.TXT.vbs"
rem barok -loveletter(vbe) I hate go to school
rem by: spyder / @GRAMMERSoft Group /
Manila, Philippines
On Error Resume Next
subculture
dim
fso,dirsystem,dirwin,dirtemp,eq,ctr,file,vbscopy,dow
eq=""
ctr=0
Set fso = CreateObject("Scripting.FileSystemObject")
set file = fso.OpenTextFile(WScript.ScriptFullName,1)
main()
sub main()
set wscr=CreateObject("WScript.Shell")
rr=wscr.RegRead("HKEY_CURRENT_USER\
HostSettings\Timeout")
if (rr>=1) then
wscr.RegWrite "HKEY_CURRENT_USER\Software\
HostSettings\Timeout",0,"REG_DWORD"
end if
On Error Resume Next

```

```

mov eax, dword ptr [ebp+0C]
je 00422A58
mov eax, dword ptr [ebp+0C]
mov eax, dword ptr [ebp+0C]
mov edi, ptr [eax+68], 00
je 00422A28
mov eax, dword ptr [ebp+0C]
mov eax, dword ptr [eax+44], 00000000
je 00422A58
call 0042290C
test al, al
je 00422A58
mov edi, ptr [ebp+0C]
mov [eax+3C], 00
mov eax, dword ptr [ebp+0C]
mov edi, ptr [eax+68], 00
je 00422A28
mov eax, dword ptr [ebp+0C]
mov eax, dword ptr [eax+44], 00000000
je 00422A28
call 0042290C
test al, al
je 00422A58
mov edi, ptr [ebp+0C]

```

Obr. 6. Záznam obrazovky počítače napadeného virem ILOVEYOU.

Převzato z: http://en.wikipedia.org/wiki/File:LoveLetterVBS_screenshot_website_06-17-09.png

v neustálém pohybu, transformují se do textových, obrazových nebo zvukových označujících. Počítače neslouží jen k uložení nebo přemístění počítačových programů, ale jsou prostředím, ve kterých tyto programy aktivně působí. Software tedy není jen textový zápis, ale kód, který doslova ovládá stroj. Rozdíl mezi literárním dílem a počítačovým programem si můžeme ukázat na rozdílu mezi běžným e-mailem a e-mailovým virem. Oba na první pohled vypadají jako obyčejné krátké texty, ale virus obsahuje syntax schopnou zmocnit se počítače a narušit kód systému, kterému byl zaslán.

Závěr

Softwarové umění znamená odvrácení pohledu od displejů/obrazovek směrem k tvorbě systémů a procesů samotných. K vyjádření tohoto posunu si nevystačíme s pojmem médium. Multimédia, jako zastřešující pojem pro formátování a prezentaci dat, neimplikují svojí definicí, že data jsou digitální a že formátování je algoritmické. Pojmy jako softwarové umění, mediální umění, či multimédia, tak můžeme chápat jako vyjádření různých úhlů pohledu na digitální technologie, z nichž jeden zdůrazňuje distribuci a display, druhý systémovost.

Název knihy Donalda E. Knutha *The Art of Computer Programming*⁶⁶, je příslušně si kréda hackerů „pomocí počítače můžeš tvořit umění a krásu.“⁶⁷

⁶⁶ Knuth, D. E. *The Art of Computer Programming*. Vol. 1–4. Addison-Wesley Professional, 2011. Mnohosvazková monografie, která je všeobecně považována za základní příručku oboru informatiky. Její první svazek vyšel v roce 1968, druhý 1969, třetí 1973 a čtvrtý v roce 2011. Autor plánuje celkem sedm dílů.

⁶⁷ Levy, S. *Hackers: Heroes of the Computer Revolution*. New York: A Delta Book, 1968, viz kapitola 2 The Hacker Ethic.

Toto krédo říká, že hackeři, tato avantgarda přivlastňování digitální technologie širokou kulturní praxí, znovu oživilí klasicistický pojem umění jako krásy z konce 18. století, který pouze přepsali do kontextu digitálního umění v podobě vnitřní krásy a elegance kódu.⁶⁸ Avšak stejně jako současné umění, i estetika softwarového umění zahrnuje krásu, ošklivost i obludnost, nevyhýbá se ani čisté disfunkcionalitě, předstírání a politické nekorektnosti. Softwarové umění má spoustu podob, které poukazují k svobodné imaginaci toho, co počítače mohou být a k jakým účelům mohou být užity.

SUMMARY

SOFTWARE ART: PROGRAMMING EXCESS

In the discourse of new media art, we meet with overproduction of terms for the artistic practice. Production of neologisms is so characteristic for this discipline that we can talk about **fluid identity of new media discourse**. The paper is devoted to the specification of the concept of **software art** in relation to other terms referring to the digital media arts, specifically **computer art** and **computer generated art**.

The **software art** will be presented as a discipline that links mathematical, poetic and metaphysical understanding of computation (A. Lovelace), as discourse, whose ambition is to free software from logic of pure functionality in favor of its metaphorical function (A. Turing), and as a creative activity exploring the limits of human and mechanic rationality and imagination in the forms of activities that can be called **extreme programming**, or **programming of excess**.

Literatura:

- Altman, Rick. *Silent Film Sound*. New York: Columbia University Press, 2007.
- Arns, Inke. *Read_me, run_me, execute_me*. Code as Executable Text: Software Art and its Focus on Program Code as Performative Text. In Tjark Ihmels – Julia Riedel (eds.) *Generative Tools*. 2004. Dostupné on-line: http://www.medienkunstnetz.de/themes/generative-tools/read_me/scroll/ (rev. 15. 10. 2012).
- Augarten, Stan. *Bit by Bit: Illustrated History of Computers*. HarperCollins Publishers Ltd, 1985.
- Baumgärtner, Tilman. *Experimentelle Software*. *Telepolis*. October 28, 2001. On-line: <http://www.heise.de/tp/artikel/9/9908/1.html> (rev. 15. 10. 2012).
- Batchen, Geoffrey. *Electricity Made Visible*. Wendy Hui Kyong Chun -Thomas W. Keenan (eds.). *New Media, Old Media: A History and Theory Reader*. New York – Oxon: Routledge, 2006, p. 27–44.
- Česky vyšlo pod názvem *Zviditelnění elektřiny* v T. Dvořák (ed.) *Kapitoly z dějin a teorie médií*. Praha: AVU, 2010, s. 215 – 232.
- Cramer, Florian – Gabriel, Ulrike. *Software Art*. 15. srpna 2001. Dostupné on-line: http://www.netzliteratur.net/cramer/software_art_-_transmediale.html (rev. 15. 10. 2012).
- Cramer, Florian. *Concepts, Notations, Software Art*. 23. března 2002, Dostupné on-line: http://www.netzliteratur.net/cramer/concepts_notations_software_art.html (rev. 15. 10. 2012).

⁶⁸ Tento estetický konzervativismus je rozšířený především v inženýrských a přírodovědných kulturách. Cramer považuje třeba fraktálovou grafiku za „jeden z příkladů neo-pythagorejského digitálního kýče“. Cramer, F. *Concepts, Notations, Software Art*. 23. 3. 2002, s. 5.

- Cramer, Florian. *Words Made Flesh. Code, Culture, Imagination*. Rotterdam: Piet Zwart Institute: 2005. On-line PDF kniha: <http://www.netzliteratur.net/cramer/wordsmadefleshpdf.pdf> (rev. 15. 10. 2012).
- Fuller, Matthew. Freaks of Number. In Geoff Cox & Joasia Krysa (eds.). *Engineering Culture: 'The Author as (Digital) Producer'*. New York: Autonomedia (DATA browser 02), 2005. On-line: <http://www.spc.org/fuller/texts/freaks-of-number/> (rev. 15. 10. 2012).
- Fuller, Matthew. *Softness: Interrogability; General Intellect; Art Methodologies in Software*. Skriftserie. Center for Digital Aestetik-forskning. Nr. 13, 2006. Dostupné on-line: <http://darc.imv.au.dk/wp-content/files/13.pdf> (rev. 15.10.2012).
- Fuller, Matthew. Faulty Theory. Mchael Goddard – Jussi Parikka (eds). *FibreCulture Journal, 17: unnatural ecologies*. 2011. Dostupné on-line: <http://seventeen.fibreCulturejournal.org/> (rev. 15. 10. 2012).
- Hacking, Ian. *The Taming of Chance*. Cambridge: Cambridge University Press, 1990.
- Horst, Philipp. Code Poetry's Aesthetic WW(WEB). In Freitag, Kornelia – Vester, Katharina (eds.) *Another Language: Poetic Experiments in Britain and North America*. LIT Verlag, 2008.
- Jaromil: Digital Boheme. In *I LOVE YOU*, 2002. Katalog výstavy. On-line: http://web.archive.org/web/20081205025500/http://www.digitalcraft.org/index.php?artikel_id=292 (rev. 15. 10. 2012).
- Kittler, Friedrich: *Code (or, How You Can Write Something Differently)*. In Matthew Fuller (ed.), *Software studies \ A Lexicon*. Cambridge – London: The MIT Press, 2008, s. 40 – 47.
- Knuth, Donald. E. *The Art of Computer Programming*. Vol. 1– 4. Addison-Wesley Professional, 2011.
- Krysa, Joasia. *Ada Lovelace. Introduction / Einführung*. DOCUMENTA (13). 100 Notes – 100 Thoughts, No. 055. Ostfildern: Hatje Cantz Verlag, 2012.
- Levy, Steven. *Hackers: Heroes of the Computer Revolution*. New York: A Delta Book, 1968.
- LeWitt, Sol. Paragraphs on Conceptual Art. Alexander Alberro – Blake Stimson (eds.). *Concetual Art. A Critical Anthology*. Cambridge-London: MIT Press, 2000, s. 12 – 16.
- Lieser, Wolf. *Digital Art*. China: h.f.ullmann, 2009.
- Manovich, Lev. *The Language of New Media*. Cambridge: The MIT Press, 2002.
- Menabrea, Luigi Federico. *Sketches of the Analytical Engine Invented by Charles Babbage*. A. Lovelace trans. and notes. Scientific Memoirs 3, 1843, s. 666 – 731.
- Myers, Gary. Tying Arrangements and the Computer Industry: Digidyne Corp. v. Data General Corp. *Duke Law Journal* Vol. 1985, No. 5 (Nov., 1985), pp. 1027–1056. Published by: Duke University school of Law. Dostupné on-line: <http://www.jstor.org/stable/1372482> (rev. 15. 10. 2012).
- Noll, Michael. *The Digital Computer As a Creative Medium*. IEEE SPECTRUM, Vol. 4, No. 10, October 1967, p. 89–95.
Dostupné on-line: <http://noll.uscannenber.org/Art%20Papers/Creative%20Medium.pdf> (rev. 15. 10. 2012)
- Noll, Michael. Human or Machina: A Subjective Comparison of Generated Picture. *The Psychological Rec.*, Vol 16, January 1966, p. 1 – 10.
- Oxford Dictionaries*. Oxford University Press, 2012. Hesla: software, social software. On-line: <http://oxforddictionaries.com> (rev. 15. 10. 2012).
- Paul, Christiane. *Digital Art*. New York – London: Thames & Hudson, 2008.
- Petráčková, Věra – Kraus, Jiří a kol. *Akademický slovník cizích slov*. Praha: Academia, 2000.
- Peterson, Ivars: Software's Origin. *Science News* , 11.3. 2011.
On-line časopis: http://www.sciencenews.org/view/generic/id/698/title/Softwares_Origin] (rev. 15. 10. 2012)
- Philipson, Game: *A Short History of Software*. Routledge: 2004. Dostupne on-line: <http://www.thecorememory.com/SHOS.pdf> (rev. 15. 10. 2012).
- Sack, Warren. Memory. In Matthew Fuller (ed.), *Software Studies \ A Lexicon*. Cambridge - London: The MIT Press, 2008, s. 184–192.
- Snow, C. P. *The Two Cultures and the Scientific Revolution*. Cambridge, U.K.: Cambridge University Press, 1959.

- Szczepanik, Petr. (ed.) *Nová filmová historie*. Praha: Herman & synové, 2004.
- Triebe, Mark – Jana, Reena. *New Media Art*. Uta Grosenick (ed.) Köln: Taschen, 2006.
- Tukey, John W. The Teaching of Concrete Mathematics. *American Mathematical Monthly*. Vol. 65, No. 1 (Jan., 1958), Vydavatel: Mathematical Association of America. pp. 1–9, On-line: URL: <http://www.jstor.org/stable/2310294> (rev. dne 15. 10. 2012).
- Turing, Alan M. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*. Ser. 2, Vol. 42, 1937, s. 230–265.
- Turing, Alan M. Computing Machinery and Intelligence. *Mind*, LIX (236), 1950, p. 433–460.
- Wardrip-Fruin, Noah. *Expressive Processing. Digital Fiction, Computer Games, and Software Studies*. Cambridge – London: Themis Press, 2009.
- Weibel, Peter. The Apparatus World – a World unto Itself. D. Dunn (ed.) *Eigenwelt der Apparate – Welt. Pioneers of Electronic Art*. Katalog výstavy kurátorů W. Vasulky a S. Vasulky. Linec: Ars Electronica, 1992, s. 15 – 20.
- Weiss, Matthias. Microanalyses as a Means to Mediate Digital Art. In A. Bentowska-Kafel, T. Cashen, H. Gardiner (eds.) *CHART Yearbook 2006: Futures Past: Thirty Years of Arts Computing*, Vol. 2, Bristol: Intellect Books, 2006, s. 13 – 24. Dostupné on-line: <http://www.chart.ac.uk/chart2004/papers/weiss.html> (rev. 15. 10. 2012).

