

# Slova se stávají tělem: Recenze knihy Floriana Cramera

**Autorka textu: Eva Čajková**

**Anotace:** Cramer vychází z předpokladu, že spustitelné (executable) kódy, takové, jaké se začaly vědecky reflektovat až s informačními technologiemi, existují už celá staletí; v magii, kabale, hudebních kompozicích, experimentální poezii atd. Kultura je prostě „komputerizovaná“ na bázi algoritmů a kniha pak rekapitulací „*kulturní historie imaginativní komputerizace*“. Následný text je recenzí shrnující základní teze Floriana Cramera, které autorka propojuje s Austinovou teorií řečových aktů.

**Abstract:** Cramer works on the presumption that executable codes, which have begun to be reflected by scientists not until information technologies, exist the centuries yet; in magic, kabbalah, music compositions, experimental poetry etc. Culture is simply computed on base of algorithms and the book is the recapitalization of „*cultural history of imaginative computation*“. Following text is review summarizing fundamental thesis of Florian Cramer linked to Austin's speech act theory.

**Klíčová slova:** Florian Cramer, Words Made Flesh, recenze, softwarová studia, John Austin, teorie řečových aktů, komputace kultury

**Keywords:** Florian Cramer, Words Made Flesh, review, software studies, John Austin, speech acts theory, computation of culture

V roce 2005, v době zvyšujícího se zájmu akademiků z oblasti humanitních a společenských věd o formulování nového pohledu na tzv. nová média, zdůrazňujícího jejich specifickou vlastnost, totiž jejich programovatelnost a práci v reálném čase, která se následně zformovala pod názvem softwarová studia, byla vydána kniha *Words Made Flesh. Code, Culture, Imagination* autora Floriana Cramera. Ten patří společně s Lvem Manovichem, Matthew Fullerem a Noahem Wardrip-Fruinem mezi stěžejní představitele zmíněného přístupu. Cramer ve svých esejích definoval formu softwarového umění jako konceptuálního umění, v němž se osamostatňuje zdrojový kód a stává se svébytnou výrazovou formou.

Ve stočtyřicetistránkové knize *Words Made Flash* se čtenář nedočká nijak zvlášť důkladné analýzy fenoménů, které autor popisuje. Tyto fenomény (pythagorejská matematika, kabala, fluxus a další živá témata) uvádí jako příklady pro aplikaci svých tezí. Zvolený postup je pochopitelný, jelikož by delší popis daných jevů byl vzhledem k jejich složitosti spíše na škodu. V záplavě informací by se pravděpodobně rozostřilo, ne-li přímo ztratilo, přemostění mezi nimi i záměr autorovy argumentace, tedy podat důkazy o dlouhé kulturní historii softwaru. Sám název napovídá, že se autor chystá v knize rozebírat téma s historicky hlubokými kulturními kořeny. *Words Made Flesh* je citát přejatý z Bible, konkrétně z Janova evangelia (1:14): „*The Word became flesh and made his dwelling among us.*” (Nová Bible, 2013) [1]

## Autonomní software

Ačkoliv se termín software etabloval již v roce 1957, do kolektivního povědomí se vepsal jako nástroj, který jde ruku v ruce s hardwarem. Hardware a software jsou považovány za neoddělitelné jednotky jako například tělo a duše, rub a líc. Cramer tento stereotyp systematicky nabourává a software povyšuje nad úroveň nástroje a přiznává mu konceptuální ráz. V této publikaci svou tezi utvrzuje zařazením digitálního kódu softwaru do historického kontextu od antiky napříč středověkem až do současnosti, přičemž ukazuje, jak se v magii, pythagorejské matematice, kabale, hudební kompozici, experimentální poezii atd., používá slovo jako kód, který má sílu (záměr i účinek) programovat lidské myšlení, dotvářet vědomí i nevědomí určité fungující společnosti.

„*This booklet attempts to show that algorithmic code and computations can't be separated from an often utopian cultural imagination that reaches from magic spells to contemporary computer operating systems.*” (Cramer, 2005, s. 8) [2]

V závislosti na tezi, která je koncentrována v tomto citátu, lze vyvodit, že Cramer chápe kód coby označující znak. Ten funguje mimo klasické, zavedené sémantické modely, ať už jako dyadický Ferdinanda Saussura či triadický C. S. Peirce. To znamená, že nelze oddělovat zdrojový kód od toho, co označuje, co reprezentuje, protože (zatím!) člověk-laik není s to spojit si kód a jeho konkrétní výstup:

```
<!doctype html><html itemscope="itemscope" itemtype="http://schema.org/WebPage"><head><meta  
itemprop="image" content="/images/google_favicon_128.png"><meta id="mref" name="referrer"  
content="origin"><title>Google</title><script>(function(){ [3]
```

Nelze jej označit ani za zavádějící znaky (záměrně odkazující k mylnému obsahu), jak chápal Roland Barthes rovinu reprezentace populárních mýtů.

Ač je to pouhá domněnka nepodložená kvantitativním výzkumem, zkušenost ukazuje, že většina společnosti (včetně autorky textu) není schopná nijak identifikovat zdrojový kód. Nicméně nesmíme přehlížet fakt, že cílem iniciátorů i šířitelů softwarových studií je studium digitálních médií z perspektivy

jak technické, tak humanitní (v tomto případě sociální a kulturní), což lze přeložit jako nezbytnost participátorů médií (odborníků i laiků) vzdělávat se v obou oborech, a kriticky pak média vnímat.

Pokud bychom přece jenom chtěli najít lingvistickou teorii, která je blízká Cramerovu konceptu autonomního kódu, jistě bychom se dostali k Austinově teorii řečových aktů.

### Austinovy performativy

V úvodu knihy (souboru přednášek) *Jak udělat něco slovy*, kde Austin své myšlenky vykládá, poznamenává, že se vymezuje proti starým (již přežitým) dogmatům gramatiky, kdy se předpokládalo, že všechny věty jsou nějakým tvrzením a musí být verifikovány. Taková tvrzení, která nemůžeme potvrdit, ani vyvrátit, přesto mají za jasných, známých podmínek smysl, označuje Austin jako performativy. Ty vymezuje vůči konstantivům, což jsou řečové akty, které mají výpovědní hodnotu, jež se dá označit za pravdivou, nebo nepravdivou.

Například výrok „má hlad“ je do jisté míry konstativ, pakliže to má být sdělení tohoto holého faktu, že „není najeden“. Ten stejný výrok však může sehrát i roli performativu, a to například tehdy, když je to mířeno na někoho, kdo by měl pochopit, že by měl dotyčného nakrmit. Aby to bylo pochopeno, musí být splněny podmínky, které Austin všem performativům určuje (zjednodušeně):

- existuje nějaká přijatá konvenční procedura
- existují osoby a okolnosti, které proceduru realizují
- procedura je provedena všemi účastníky správně a úplně
- záměrem účastníků je chovat se podle procedury
- účastníci se následně podle dohodnuté procedury chovají.

Příklad „mám hlad“ by pravděpodobně ve většině případů skončil jako konstativ.

Austin ukazuje na řadu případů, kdy dochází k selhání performativní výpovědi. To nastane, když není splněna některá z podmínek, např. chci-li někoho oddat, řeknu přesně to, co říct mám, ale nejsem kompetentní zmocněnec, například kněz v případě křtu.

*„[...] tam, kde se neuplatňuje nárok na kompetenci, ba ani se nepředstírá uplatňování takového nároku, tam žádná přijatá konvenční procedura neexistuje; a jde pak o frašku, jako je třeba sňatek s opicí.“*  
(Austin, 2000, s. 39)

Ze sady příkladů, které Austin používá (např. performativ „Odsuzuji Vás...“, „Křtím Tě jménem...“ atd.), může vzejít mylný dojem, že performativy jsou jakási institucionalizovaná tvrzení. Performativy však mají mnohem širší okruh použití v praxi, jak o tom svědčí např. aplikace této teorie jinými teoretiky na literaturu a divadlo v 70. letech. Pravdou je, že tyto teorie působí poněkud eklekticky. Na softwarové umění se nicméně hodí skvěle.

K ucelení teorie rozvinul Austin ještě další neologismy a implicitně koncepty; tři kategorie mluvních aktů: akty lokuční, ilokuční a perlokuční. Jednotlivé kategorie se dělí podle stupně akce, tedy podle akce konané slovy. Lokuční jsou takové, které se dají označit jako buďto pravdivé, nebo nepravdivé. Alokuční

jsou takové, kdy mluvčí zároveň dělá to, co také říká (např. soudce: „Odsuzuji Vás...“). „*The message and execution come together [...]*“ (Arns, 2005) [4] Perlokuční akty jsou takové, které spouští řetěz následků. Ilokuční akty mají efekt okamžitě bez prodlení, perlokuční s jistým zpožděním a v závislosti na okolnostech. Ty výroky, které nemají okamžitý efekt, se dějí svou performativitou, která je efektivní tak, že spouští řetězec událostí. A právě v tom lze spatřit ekvivalent s programovými kódy.

## Algoritmy kultury

Přistupme k přehledu tezí Floriana Cramera, jak je sám shrnuje:

„*Software is a cultural practice made up of (a) algorithms, (b) possibly, but not necessarily in conjunction with imaginary or actual machines, (c) human interaction in a broad sense of any cultural appropriation and use, and (d) speculative imagination. Software history can thus be told as intellectual history, as opposed to media theories which consider cultural imagination a secondary product of material technology.*“ (Cramer, 2005, s. 124) [5]

**(a) algoritmy:** Cramer na řadě příkladů dokazuje, že ke komputaci kultury pomocí čísel docházelo běžně ve filozofiích a jejich psaných dokumentech, které utvářely kulturu, dávaly společnosti řád tak, že jí předepisovali, jak se chovat. Příkladem je *Sefer Jecira*, doslovným překladem z hebrejštiny „Kniha stvoření“, pozdně starověký kabalistický spis.

„*Dvacet dvě základní písmena:*

*Tři matky, sedm dvojitých a dvanáct jednoduchých.*

*Tři matky: A, M, Š,*

*jichž základem jest miska nevinu a miska viny,*

*a jazýček jako kolísavý*

*řád mezi jimi.*“ (Griese, 1921, s. 9)

Kabala kombinuje pythagorejskou ideu světa postaveného na číslech a proporcích s magickým jazykem jako činitelem, jenž ovlivňuje materii (Cramer, 2005, s. 29).

„*The religious sublimation of this concept exists in the idea that God created the world through language and that even humans possessed the power of influencing things through the Adamic language spoken in paradise.*“ [6]

**(b) možné, ovšem nikoliv nezbytné spojení s pomyslnými nebo existujícími stroji:** tímto Cramer deklaruje své přesvědčení, že software (kód) je schopný existovat nezávisle na strojích, na hardwaru, což se však nedá říct naopak. Hardware je bez softwaru jen pouhou schránkou, formou bez obsahu, zatímco software je obsahem, jenž se může stát i formou. Autor důkladně rozebírá imaginární stroje (návrhy mechanických pomůcek) *Ars Magna Ramona Llulla* (též Raimundus Lullus) [7], většinou soustředné kruhy, jejichž otáčením se vytvářely pojmy filozoficko-teologického charakteru.

„It provides, as a symbolic code, an abstract ‘alphabetum’ or artificial metalanguage for the entries of the table. The letter ‘A’ is omitted in this artificial language, expressing the taboo of representing god as the absolute beginning and therefore the first letter in the alphabet. Lull’s letter B–K may be the first example of what computer science calls the ‘semantics’ of a programming language.” (Cramer, 2005, s. 37) [8]

**(c) lidská interakce v širokém smyslu kulturního osvojení a užití:** to souvisí s již zmiňovaným apelem badatelů softwarových studií na potřebu vnímat software jako kulturní formu, která proniká každodenním životem a neodmyslitelně se vštěpuje do lidských mozků. V této kategorii uvedme některé umělce (jako zástupce artifiční kultury): Tristan Tzara, John Cage, William S. Burroughs, Oulipo, Jodi, mez, I/O/D group, Netochka Nezvanova nebo Adrian Ward a spousta dalších, bez omezení kulturně-uměleckými epochami.

**(d) spekulativní obrazotvornost (speculative imagination):** komputace je jedinečná svou schopností generovat obrovské (až nekonečné) množství informací a jejich kombinací podobně, jako je tomu v Borgesově povídce *Babylonská knihovna*. Vyčerpávající výčet příkladů a systematickou kategorizaci komputací zde nemusíme sami koncentrovat, protože to udělal už Cramer:

„It [symbolická forma komputace] encompasses opposites, algorithms as a tool versus algorithms as a material of aesthetic play and speculation, computation as inner workings of nature (as in Pythagorean thought) or God (as in Kabbalah and magic) versus computation as culture and a medium of cultural reflection (starting with Oulipo and hacker cultures in the 1960s), computation as a means of abolishing semantics (Bense) versus computation as a means to structure and generate semantics (as in Lullism and Artificial Intelligence), computation as a means of generating totality (Quirinus Kuhlmann) versus computation as a means of taking things apart (Tzara, cut-ups), software as ontological freedom (GNU) versus software as ontological enslavement (Netochka Nezvanova), ecstatic computation (Kuhlmann, Kabbala, Burroughs) versus rationalist computation (from Leibniz to Turing) versus pataphysical computation as the parody of both rationalist and irrationalist computation (Oulipo and generative psychogeography), algorithm as expansion (Lullism, generative art) versus algorithm as constraint (Oulipo, net.art), code as chaotic imagination (Jodi, codeworks) versus code as structured description of chaos (Tzara, John Cage).” (Cramer, 2005, s. 125) [9]

## Shrnutí: komputace kultury

Florian Cramer používá termín komputace jako označení schopnosti člověka číst kódy své kultury, celou řadu kódů, kulturní software. Aluzí k tomuto konceptu může být neurolingvistické programování, komunikační metoda alternativních psychologií, nicméně Cramer o kulturních kódech nemluví ani tak jako o něčem, co lidskou mysl přenastavuje (rozhodně to nepopírá, jen to není perspektiva knihy *Words Made Flesh*). Spíš vnímá lidský mozek jako hardware schopný kódy přečíst; ke spoustě úkonů proto není třeba dalších strojů, jak tento fakt dokazuje množstvím historických příkladů. Cramerova komputace odkazuje k překladu přirozených dějů do jazyka kódu.

Jak bylo zmíněno na začátku této recenze, Cramer si pro svou metodu vybral složité systémy, což volá po hlubší analýze toho, jak interaguje kulturně podmíněná mysl a symbolický systém. Takový výzkum by naplňoval požadavek softwarových badatelů na kooperaci humanitních a technických věd.

**Poznámky:**

[1] „A to Slovo se stalo tělem a přebývalo mezi námi.“ Viz *Nová bible kralická*, 2013.

[2] „Tato příručka se pokouší ukázat, že algoritmický kód a komputace nemůžou být odděleny od často utopické kulturní imaginace, která dospívá od magického zařikávání až k počítačivému operačnímu systému.“ (Cramer, 2005, s. 8) Překlad: Eva Čajková

[3] Ukázka zdrojového kódu.

[4] „Zpráva a její provedení přichází zároveň.“ (Arns, 2005) Překlad: Eva Čajková

[5] „Software je kulturní praktika tvořená: (a) algoritmy, (b) možným, ovšem nikoliv nezbytným spojením s pomyslnými nebo existujícími stroji, (c) lidskou interakcí v širokém smyslu kulturního osvojení a užití, (d) spekulativní imaginací. Historii softwaru tak může být nazývána jako intelektuální historie, jako opak k mediálním teoriím, které braly kulturní imaginaci za sekundární produkt materiální technologie.“ (Cramer, 2005, s. 124) Překlad: Eva Čajková

[6] „Náboženská sublimace tohoto konceptu existuje v ideje Boha, který tvoří svět pomocí jazyka, a posléze dokonce lidi posedlé silou ovlivňovat věci skrze Adamův jazyk v ráji.“ (Cramer, 2005, s. 29) Překlad: Eva Čajková

[7] Žil v letech 1232–1315.

[8] „Zajišťuje to, jako symbolický kód, abstraktní 'abecedu' nebo umělý metajazyk. Písmeno 'A' je v tomto jazyce vypuštěno, aby vyjádřilo tabu znázornění boha jako absolutního začátku, a právě proto počáteční písmeno abecedy. Llullovo písmeno B-K může být prvním příkladem toho, co počítačová věda nazývá 'sémantikou' programovacího jazyka.“ (Cramer, 2005, s. 37) Překlad: Eva Čajková

[9] „Zahrnuje [symbolická forma komputace] opozita: algoritmy jako nástroj versus algoritmy jako materiál estetické hry a spekulace, komputace jako vlastní práce přirozenosti (jako v pythagorejských myšlenkách) nebo Bůh (jako v kabale nebo v magii) versus komputace jako kultura a médium kulturní reflexe (začínající u Oulipa a hackerské kultury 60. let), komputace jako význam mizející sémantiky (Bense) versus komputace jako význam struktury a generativní sémantiky (jako llullismus a umělá inteligence), komputace jako význam generativní totality (Quirinus Kuhlmann) versus komputace ve smyslu rozebírání věcí (Tzara, cut-upy), software jako ontologická svoboda (GNU) versus software jako ontologická zotročení (Netochka Nezvanova), exstatická komputace (Kuhlmann, Kabbala, Burroughs) versus racionální komputace (od Leibnitze po Turinga) versus patafyzikální komputace parodie obojího, racionální i iracionální komputace (Oulipo a generativní psychogeografie), algoritmus jako expanze (lullismus, generativní umění) versus algoritmus jako omezení (Oulipo, net.art), kód jako chaotická imaginace (Jodi, codeworks) versus kód jako strukturovaná deskripce chaosu (Tzara, John Cage).“ (Cramer, 2005, s. 125) Překlad: Eva Čajková



**Použitá literatura:**

ARNS, Inke, 2005. Read\_me, run\_me, execute\_me. Code as Executable Text: Software Art and its Focus on Program Code as Performative Text. In: FRIELING Rudolf – DANIELS Dieter (eds.) *Media Art Net 2: Key Topics*. Wien/New York: Springer, s. 197–207 (engl.). Dostupné z: [http://www.mediaartnet.org/themes/generative-tools/read\\_me/scroll/](http://www.mediaartnet.org/themes/generative-tools/read_me/scroll/).

AUSTIN, John Langshaw, 2000. *Jak udělat něco slovy*. Praha: Filosofia.

CRAMER, Florian – GABRIEL, Ulrike, 2001. Software Art. *Netzliteratur* [online]. [citováno dne 1. 2. 2013]. Dostupné z: [http://www.netzliteratur.net/cramer/software\\_art\\_-\\_transmediale.html](http://www.netzliteratur.net/cramer/software_art_-_transmediale.html).

CRAMER, Florian, 2002. Concepts, Notations, Software, Art. *Netzliteratur* [online]. [citováno dne 1. 2. 2013]. Dostupné z: [http://www.netzliteratur.net/cramer/concepts\\_notations\\_software\\_art.html](http://www.netzliteratur.net/cramer/concepts_notations_software_art.html).

CRAMER, Florian, 2005. *Words Made Flesh*. Rotterdam: Piet Zwart Institute.

GRIESE, Otakar, 1921. *Sefer Jecira. Kniha o utváření*. Přerov: Ústřední nakladatelství okultních děl [online]. [citováno dne 1. 2. 2013]. Dostupné z: [http://knihovna.vzahrade.net/knihy/Sefer\\_Jecira-Kniha\\_o\\_utvareni.pdf](http://knihovna.vzahrade.net/knihy/Sefer_Jecira-Kniha_o_utvareni.pdf).

Nová bible kralická, 2013. *Evangelická teologická fakulta* [online]. [citováno dne 25.3.2013]. Dostupné z: <http://www.etf.cuni.cz/~rovnanim/bible/nbk/J1.php>.