

Komputer i jego znaki*

Dość długo obróbka żelaza pozostawała wierna stylowi i modelom odziedziczonym po epoce brązu, podobnie jak epoka brązu podtrzymywała początkowo kształty i stylistykę kamienia łupanego.

Mircea Eliade, *Kowale i alchemicy*

Rzut oka w przeszłość

Co zainspirowało człowieka tysiące lat temu do pozostawiania znaków? Czy kształt stopy odcisnięty w miękkiej glebie? Czy ślad dłoni, przybrudzonej barwnikiem, pozostawiony na skale? Czy rysy, które pojawiały się w sposób naturalny na tworzywie pierwszych narzędzi? Trudno powiedzieć. Jednakże moment, w którym człowiek zaczął pozostawiać znaki celowo, skłonny byłbym uznać za narodziny cywilizacji. Znak, rozumiany jako nośnik przekazu, jest nieodłącznym i niezwykle ważnym atrybutem kultury. To wynalazek znaku utworzył drogę wynalazkowi pisma, bez którego trudno sobie wyobrazić rozwój cywilizacyjny.

Trwająca od kilku wieków eksplozja technologiczna wywarła istotny wpływ na rolę znaku w kulturze. Wprowadzenie maszyn zdolnych precyzyjnie powtarzać tę samą czynność zaowocowało tym, że znak zatracił swoją pierwotną indywidualność, zaczął ulegać – jak byśmy to dziś powiedzieli – standaryzacji. Równocześnie człowiek, twórca idei znaku, odsuwany był przez postęp technologiczny coraz dalej od fazy nadawania znakowi ostatecznej postaci.

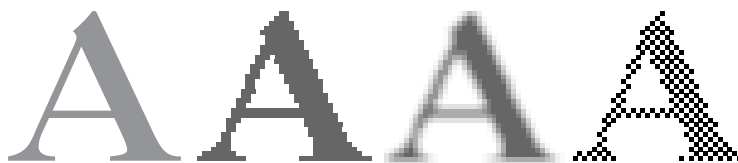
Twórcy paleolityczni najpewniej uczestniczyli w całym procesie tworzenia dzieła – od wyszukiwania i przygotowywania barwników, aż do naniesienia ich własnoręcznie na skałę. Ten sposób tworzenia przetrwał bardzo długo, przez tysiąclecia proces przygotowywania narzędzi i tworzywa był udziałem twórcy. Tak było jeszcze w początkowej fazie powszechnej industrializacji. Na przykład artysta projektujący czcionkę sam wycinał jej kształt w metalu, sam składał tekst, sam parął się drukiem.

Później pojawiła się niezbędna w świecie podporządkowanym technologii specjalizacja, która spowodowała, że twórca nie miał szans, by brać udział w całym procesie powstawania znaków. Produkcją farb i pędzli zajęły się fabryki, projektant pisma przestał być drukarzem. Człowiek został oddzielony od swojego znaku.

* Artykuł został zainspirowany Jesiennymi Spotkaniami Polskiego Towarzystwa Informatycznego w Mrągowie i w Wiśle, których autor był wielokrotnym uczestnikiem.

Znak w dobie komputerów

Mniej więcej w połowie XX wieku wkroczył do akcji rzekomy sprzymierzeniec, a w istocie, jak się miało okazać, wróg znaku w jego dotychczasowej formie – komputer. Otóż znak, który widzimy na ekranie komputera lub na wydruku, to tylko czubek góry lodowej. Pod powierzchnią kryje się nader skomplikowana struktura. Podobnie jest zresztą ze znakiem narysowanym na kartce papieru, tylko że w przypadku znaku komputerowego struktury nie można w żaden sposób pominąć – i twórca, i użytkownik znaku o tej strukturze musi wiedzieć i umieć z tej wiedzy korzystać. W przypadku znaku narysowanego na kartce papieru wystarczy wiedzieć, że papieru nie powinno się niszczyć – drzeć, wkładać do pieca czy moczyć w wodzie. W przypadku znaku komputerowego potrzebna jest wiedza dużo głębsza.



Rys.1. Ten sam znak (tu: szara litera „A”) może być interpretowany przez program komputerowy na wiele sposobów, w szczególności mogą mu odpowiadać dwie liczby: literze „A” na ogół odpowiada liczba 65, zaś 50-procentowej szarości – liczba 128 lub 0,5

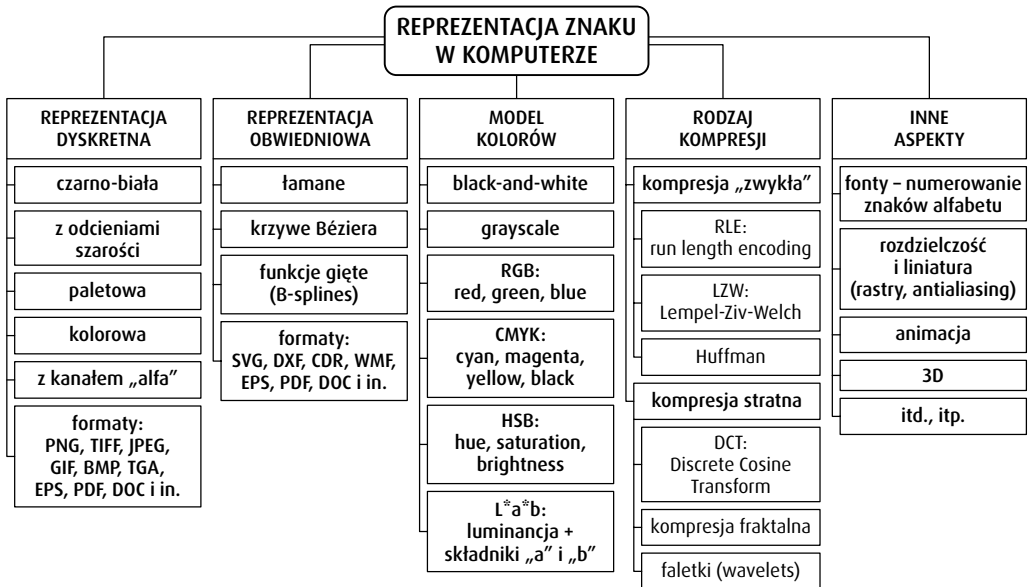
Najpospolitszym przykładem wspomnianej wyżej struktury jest ponumerowanie liter alfabetu. Skądinąd ważne osiągnięcie intelektualne i technologiczne, dziś okazuje się hamulcem rozwoju i niewyczerpanym źródłem kłopotów. W ten sposób powstała bowiem dwoistość reprezentacji danych: dane mogą być zapisane bądź w postaci graficznej, odzwierciedlającej jedynie kształt znaku, bądź w postaci „ponumerowanej”, zwanej tekstową lub znakową.

Dwoistość taka okazuje się niezrozumiała dla tych, którzy z komputera korzystają sporadycznie i nie są zainteresowani zdobywaniem zaawansowanej wiedzy o komputerach. Nazwijmy umownie takich użytkowników *nieinformatykami* w odróżnieniu od *informatyków* wtajemniczonych w arкана wiedzy komputerowej.

Informatyka dziwi, że nieinformatyk nie zdaje sobie sprawy z tego, iż tekst wydrukowany na papierze można wprowadzić z powrotem do komputera jedynie za pomocą skanowania i poddania analizie OCR (*Optical Character Recognition*) i że lepiej tekst przekazywać w postaci elektronicznej. Dziwi niesłusznie, bo powinno dawać do myślenia.

Jeżeli chodzi o ilustracje, to nieinformatyk nie ma szans zrozumieć, skąd się bierze podział postaci graficznej na obwiedniową (ciągłą), tzn. zapisaną w postaci matematycznie przedstawionych krzywych, i rastrową (dyskretną), w której obraz jest pamiętany jako tablica pikseli¹, gdzie każdemu pikselowi przypisany jest kolor. Pamiętajmy też, że grafika rastrowa daje się dalej podzielić na czarno-białą, szarą, paletową, RGB, CMYK i inne rodzaje, a jakby tego było mało, to w odwodzie po-

[1] Słowo *piksel* jest akronimem utworzonym ze słów *picture element*, czyli element obrazu.



Rys. 2. Świat aspektów związanych z reprezentacją znaku w komputerze jest zaskakująco bogaty (powyższy diagram jest oczywiście daleki od kompletności; określenia angielskie zostały pozostawione celowo) – każdy z tych aspektów może dać o sobie znać w przykry sposób przy próbie przekazania materiału tekstowego i graficznego między komputerami

zostają różne formaty graficzne (TIFF, PNG, JPEG) i różne sposoby kompresji danych (RLE, LZW, DCT). Dlaczego tak samo wyglądający obrazek może zająć kilka kilobajtów lub kilka megabajtów? Dlaczego jeden obrazek daje się dowolnie skalować, a na drugim po przeskalowaniu pojawiają się jakieś wzorki? Dlaczego wydrukowany obrazek jest nieostry, ba, nawet wyraźnie widać na nim jakieś kwadraciki, poszarpane brzegi (por. rys. 1), a na ekranie wszystko było dobrze?

Tego rodzaju pytania nieinformatyka mogą się wydać naiwne informatykowi, który wie, co to jest *kompresja*, *mora*, *antyaliasing*, *rozdzielczość*, *raster*. A jednak są to pytania trafiające w sedno sprawy – przecież użytkownik komputera, a zwłaszcza nieinformatyk, ma prawo oczekiwać, że komputer ułatwi tworzenie i używanie znaku, a nie, że je utrudni.

Zestawy znaków, czyli fonty

Już pobieżne zaznajomienie się z problemem struktury pojedynczego znaku w komputerze (por. rys. 2) może przyprawić o zawrót głowy. Nietrudno sobie wyobrazić, co może oznaczać „struktura” w przypadku zestawów znaków zwanych fontami. Kłopoty związane z reprezentacją pojedynczych znaków w całości przenoszą się na fonty, w szczególności trwa zamieszanie związane z podziałem na fonty obwiedniowe i, odchodzące właściwie w przeszłość, ale wciąż stosowane do wyświetlania na ekranie, fonty rastrowe (tzw. bitmapowe). Do tego dochodzą kłopoty specyficzne dla fontów.

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	W	X	Y	Z
ă	â	ă	ç	š	ć	ó	ű	ž	ž	ó						3	4	5	6	7	8	9	o
À	Ā	Ĉ	Ċ	Ď	Š	Ě	Ě	Ě	Ĉ	Æ	Œ	J	U	Ł	ı	2							
A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	W	X	Y	Z
â	é	î	ô	ú	ń	ś		ć	ą	ę	—				.						Ź	Ż	
ä	ë	ï	ö	ü	fl	é		s	t	u	r				w	y	z				ž	ž	
à	è	ì	ò	ù	fi	ó																v	x
á	é	í	ü	ff	k			m	n	i	o				p	-	.	!	;			Pół	fi
Ā	Ē	Ń	Ń	Ń	Ń	Ń	Ń								q	:	:	?	(Firety
ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	a	Spacje	e	d				h	,		pół	kwa	draty			Kwadrat
æ	œ	J	U	Ł	j	c									g	f		konokr-	dzansy				ty

Rys. 3. Podobieństwo układu kaszty zecerskiej (górną) i tablicy znaków wyświetlanej przez program do obróbki fontów (dół) przywodzi na myśl podobieństwo pierwszych samochodów do pojazdów konnych i zdaje się świadczyć o niezbyt dalekim zaawansowaniu technologii komputerowej w tym względzie

ki nim można w szerokim zakresie kontrolować pozycjonowanie i podmianę znaków. Należy jednakże podkreślić, że sposób interpretacji danej dyrektywy nie jest precyzyjnie zdefiniowany w foncie – zależy on od programu przetwarzającego font. W szczególności dyrektywy niezrozumiałe dla danego programu mogą być ignorowane. Istotną zmianę wprowadzono (półoficjalnie) w fontach OpenType około połowy pierwszej dekady obecnego stulecia – do struktur fontowych została dodana tablica MATH. Tablica ta, zaproponowana przez firmę Microsoft, wzorowana na koncepcji zastosowanej w systemie TeX przez Donalda E. Knutha (por. [3]), zawiera informację niezbędną do składu formuł matematycznych. Fonty z tablicami MATH (por. np. [4] oraz [9]) są obsługiwane przez najnowsze wersje pakietu Microsoft Office oraz niektóre warianty TeX-a (LuaTeX, XeTeX).

Pozostałości z przeszłości

Fonty odziedziczyły – zgodnie z powtarzającym się od pradziejów schematem – wiele cech po swoich przodkach, czyli kasztach drukarskich. W szczególności dotyczy to zestawu znaków, który jest znacznie bogatszy niż zestaw oferowany

Font jest odpowiednikiem zecerskiej kaszty – zbiorem znaków do drukowania tekstów (por. rys. 3). Dodatkowo font zawiera informacje o znakach – o ich rozmiarach, o pozycjonowaniu par znaków, o ligaturach itp. Informacje te zależą oczywiście od rodzaju (formatu) fontu.

Najmłodniejszy obecnie jest format OpenType – wspólny projekt firm Adobe Systems i Microsoft, w istocie będący „skrzyżowaniem” postscriptowego² formatu Type 1 oraz windowsowego formatu TrueType. W zamierzeniu format OpenType miał zastąpić te właśnie formaty. Na razie jednak wszystkie trzy formaty zgodnie koegzystują, aczkolwiek należy przyznać, że popularność formatu OpenType rośnie, a pozostałych dwóch – maleje.

Fonty OpenType oprócz poprawionych zabezpieczeń przed wykorzystaniem niezgodnym z prawem wyposażone zostały w struktury zwane *features* (cechy). *Features* można by określić po polsku jako „dyrektywy zecerskie”. Dzięki

[2] Język PostScript, i jego następcą PDF (*Portable Document Format*), to produkty firmy Adobe Sys-

tems, będące światowym standardem opisu stron graficzno-tekstowych.

przez maszyny do pisania, po których komputery odziedziczyły z kolei klawiatury. Takie niejednorodne dziedzictwo prowadzi nieuchronnie do zamieszania – nie może być inaczej, skoro drukarze dopracowali się np. czterech kresek poziomych różnej długości (są to: dywiz, pauza półfretowa, matematyczny minus oraz pauza fletowa), natomiast na klawiaturze maszyny do pisania znalazła się tylko jedna kreska. Stąd w przytłaczającej większości tekstów przygotowywanych za pomocą komputerów pojawia się – wbrew możliwościom komputerów i tradycji drukarskiej – tylko najkrótsza z kolekcji kresek.

Dziedzictwo drukarskie w połączeniu z początkowo prymitywnymi implementacjami fontów zrodziło oprócz zamieszania także problemy natury estetycznej. Otóż znak we współczesnych fontach jest niejako „zamrożony” – jego kształt zasadniczo nie podlega modyfikacji.

Niemniej jednak łatwość realizacji w komputerze przekształceń afinicznych spowodowała, że takie zabiegi, jak skalowanie, zwężanie czy pochylanie znaków, weszły na stałe do repertuaru opcji programów do przetwarzania tekstów (tzw. edytorów formatujących). Mistrzowie typografii oczywiście nie chcą słyszeć o poddawaniu znaków transformacjom afinicznym i traktują czcionkę jak konterfekt (por. rys. 4).

Nieprzypadkowo w języku angielskim słowo *typeface* (dosł. twarz czcionki) oznacza krój pisma. Według kanonów sztuki typograficznej, nawet zmianie rozmiaru znaku powinna towarzyszyć delikatna zmiana proporcji jego elementów.

Tymczasem popularność, jaką cieszą się edytory formatujące, pchnęła ich twórców nawet dalej – niektóre edytory potrafią utworzyć z fontu podstawowego odmianę pogrubioną w razie jej braku, czasem po prostu przez obrysowanie znaku obwódką. Skutki estetyczne są przeważnie opłakane, ale dzięki temu użytkownik nie musi się zastanawiać, dlaczego w danej rodzinie kroju pisma brakuje odmiany pogrubionej.

W tym kontekście godną odnotowania próbą było opracowanie przez firmę Adobe Systems formatu fontów postscriptowych Multiple Master. Font typu Multiple Master zawiera w istocie kilka osobno zaprojektowanych odmian danej rodziny, np. normalną, wąską i szeroką. Użytkownik systemu potrafiącego obsługiwać fonty Multiple Master może zażyczyć sobie, by jego dokument został złożony fontem będącym dowolną kombinacją liniową fontów³ stanowiących bazę fontu Multiple Master. Dzięki temu można było uzyskać font lekko poszerzony lub zwężony



Rys. 4. Trudno sobie wyobrazić, żeby w poważnej publikacji ilustrator nie dość że *Dziewczynnie z perłą* Jana Vermeera ująłby kolorów (transformacja taka jest na razie dopuszczalna ze względów technicznych), to jeszcze przekształciłby ją afinicznie

[3] Algebraiczne pojęcie kombinacji liniowej daje się zastosować do krzywych Béziera, z których zbudowane są znaki w fontach postscriptowych,

daje się więc przenieść na znaki, a tym samym także na fonty.

z zachowaniem stosownych zmian proporcji elementów znaku. Był to niewątpliwie słuszny krok w stronę „odmrożenia” znaku w foncie – wraz z rosnącą mocą komputerów znaki mogą być opisywane w coraz bardziej złożony sposób. Fonty Multiple Master były jednak skomplikowane w produkcji, dość niewygodne w obsłudze i niestety ograniczone, jeśli chodzi o możliwości. Stąd zapewne ich umiarkowana popularność i – w konsekwencji – decyzja Adobe o poniechaniu tej technologii i skupieniu się na rozwijaniu technologii OpenType, które w ograniczonym zakresie umożliwiają skalowanie „optyczne”, tzn. uwzględniające zmianę proporcji elementów znaku.

Inne próby „ożywienia” znaków w foncie, takie jak na przykład delikatna zmiana kształtu szeryfów przy niektórych zestawieniach liter, nie wyszły poza laboratoria typograficzne. Animacje tekstowe w programach prezentacyjnych (np. OpenOffice Impress czy Microsoft PowerPoint) czy cudactwo zaimplementowane w starszych wersjach edytora Microsoft Word, mianowicie „fonty animowane”, nie mają oczywiście nic wspólnego z „ożywianiem” znaków, o którym tu mowa.

Najbardziej czywista koncepcyjnie technologia, traktująca znaki w komputerze jako programy (np. obiekty z odpowiednimi strukturami danych i metodami mogące się kontaktować ze sobą i z systemem składu), nie doczekała się jak na razie realizacji, z wyjątkiem fontów postscriptowych określanych jako Type 3. W fontach tych znaki mogły być niemalże dowolnymi programami zapisanymi w języku PostScript (por. [7]). Niestety, postscriptowe fonty Type 3 nie stały się popularne, jako że są zrozumiałe jedynie dla interpreterów języka PostScript (ale już nie dla interpreterów PDF-ów), a większość programów graficznych i edytorów tekstu nie ma stosownego modułu pozwalającego interpretować PostScript.

„Standardy” kodowania, czyli wieża Babel

Perturbacje wywoływane zawiłym rodowodem komputera to pół biedy. Aspekty estetyczne też zawsze można zrelatywizować. Gorzej, że na scenie pojawia się koń trojański numerowania znaków. Przez długie lata liczba znaków w foncie – mocą tradycji, podpieranej argumentem rzekomej efektywności – nie przekraczała magicznej wartości 256. Póki technologia komputerowa rozwijała się w krajach, które nie używają znaków diakrytycznych, problem pozostawał w ukryciu. Niestety szybko okazało się, że inne kraje też chcą używać komputerów i muszą w związku z tym przyporządkować numery swoim znakom. Nagle w 256-znakowej tablicy zrobiło się za ciasno dla wszystkich. Jak grzyby po deszczu zaczęły więc pojawiać się „standardy” kodowania, uwzględniające to te, to tamte znaki, nigdy jednakże nie zaspokajające w pełni potrzeb użytkowników.

Istnienie setek schematów kodowania kojarzy się nieodparcie z wieżą Babel, aczkolwiek w tym przypadku „pomieszenie języków” okazało się (na razie) mniej brzemienne skutki. Mimo powszechnego zastosowania komputerów do komunikacji można się jakoś dogadać, ot, najwyżej polskie znaki zostaną pozbawione elementów diakrytycznych albo zamienione na dziwne symbole, zwane popularnie „krzaczkami”. Aż korci, by podkreślić, że pióro i papier nie przysparzały aż takich kłopotów.

Babelopodobne zamieszanie nie byłoby szczególnie niepokojące, gdyby nie trendy wskazujące na umacnianie się wadliwej od początku koncepcji numerowania znaków w sposób uniwersalny. Skoro jeden bajt nie wystarcza, to użyjmy dwóch bajtów do numerowania itd. – proste, prawda? Za proste, żeby mogło zadziałać.

Koncepcja numerowania znaków (aktualnie dopuszczalne są numery dwudziestobitowe) uzyskała rangę normy ISO/IEC 10646 i znana jest pod nazwą standardu Unicode. Szczegółową analizę standardu Unicode w kontekście zastosowań typograficznych przedstawił w poprzednim numerze „Acta Poligraphica” Piotr Strzelczyk [8]. Tu poprzestaniemy zatem na zwięzłym przedstawieniu niektórych istotnych kłopotów, jakie sprawia Unicode w typografii.

Po pierwsze, standard Unicode ukierunkowany jest na znaki (*characters*), a nie na formy (*glyphs*). W związku z tym takie twory, jak ligatury, jako nie mające tożsamości semantycznej, nie należą do obszaru zainteresowań Unicode. Ligatury tradycyjne („fi”, „ff”, „fl”, „ffi” czy „ffl”) znalazły się wprawdzie w tabelach Unicode, ale dziś znawcy tematu coraz bardziej skłaniają się ku uznaniu tej decyzji za błędną. Tymczasem typografia, zwłaszcza wytworna, nierozzerwalnie wiąże się z wykorzystaniem różnorodnych, niekoniecznie typowych ligatur. Zatem brak miejsca na formy tego rodzaju w standardzie Unicode oznacza, że jest to w najlepszym razie adekwatny standard językowy, ale nie typograficzny. A zważmy, że dzięki gwałtownemu rozwojowi systemów DTP zastosowania typograficzne są – i wolno przypuszczać, że pozostaną – niezwykle popularne.

Po drugie, Unicode w sposób zdecydowany odcina się od języków „nietekstowych”, takich jak notacja muzyczna czy taneczna. Jeśli więc Unicode zaplanuje jako standard wymiany dokumentów, ważne obszary zastosowań, w których wykorzystywane są znaki „nietekstowe”, w sposób sztuczny zostaną wyniesione poza standard. Utrudni to życie tym, którzy mają nieszczęście zajmować się notacją muzyczną czy taneczną. Notacja muzyczna i taneczna, a także pismo kaligraficzne to niezwykle ciekawe przykłady języków posługujących się w istocie *nieskończoną* liczbą znaków. Nie dziwota, że fonty zbudowane z „zamrożonych” znaków i różne systemy numerowania znaków nie pasują do tych języków. Nie oznacza to jednak, że komputer jako taki jest niedobrym narzędziem do składu nut czy zapisu kroków tanecznych.

Po trzecie, języki takie jak tolkienowski Tengwar czy język klingoński z serialu „Star Trek” z trudem mieszczą się w zakresie zainteresowań standardu Unicode. Oba wymienione języki być może znajdują się w standardzie Unicode (por. [2]), co z jednej strony wydaje się niezupełnie poważne, a z drugiej strony długotrwała z natury procedura włączania nowych znaków do standardu Unicode po-



Rys. 5. Zestaw znaków w foncie nie musi mieć nic wspólnego z jakimkolwiek językiem; trudno sobie wyobrazić, aby dało się ponumerować w wygodny sposób wszystkie możliwe do pomyslenia symbole

woduje, że twórcy nowych języków, np. fantastycznych, mogą się pożegnać z myślą o wygodnym posługiwaniu się standardem Unicode.

Po czwarte wreszcie, sporo fontów nie zawiera liter, tylko znaki ozdobne lub ornamenty (por. rys. 5). Sporo takich znaków zostało uwzględnionych w standardzie Unicode, ale chyba nikt przy zdrowych zmysłach nie spodziewałby się, że wszystkie znaki – i te już wymyślone, i te, które dopiero ktoś wymyśli – można ponumerować. Znaki spoza standardu Unicode powinny znaleźć się w obszarze przeznaczonym na użytek prywatny (PUA – *Private Use Area*). Zarezerwowane są na to zakresy od $u+e000$ do $u+f8ff$, od $u+f0000$ do $u+ffffd$ i od $u+100000$ do $u+10ffffd$ (137 468 znaków), gdzie można umieszczać swoje własne znaki pod własnymi nazwami i kodami. Jest to ewidentna próba załatwienia problemu i to niezbyt skuteczna, bo istniejące narzędzia nie pozwalają na łatwe i wygodne wprowadzanie takich znaków z klawiatury. Nie ma bowiem uniwersalnego, dobrze zdefiniowanego mechanizmu swobodnego definiowania znaczenia klawiszy czy też kojarzenia klawiatury z fontem lub wybranymi znakami. Co gorsza, umieszczanie znaków w obszarze PUA z definicji wyklucza możliwość uniwersalnej wymiany dokumentów, czyli w istocie zaprzecza idei, która przyświeca Unicode Consortium: ułatwić w jak najwyższym stopniu wymianę dokumentów elektronicznych.

Jakim prawem?

Wraz z rozwojem oprogramowania komplikował się opis znaku. I o ile w przypadku grafiki rastrowej można było mówić o znaku jako o danych do programu wyświetlającego lub drukującego znak, to obecnie nie jest jasne, czy znak to program, czy dane. W przypadku znaku opisanego za pomocą języka PostScript raczej trzeba by mówić o programie, chociaż z kolei przygotowywanie znaku za pomocą programu takiego, jak np. CorelDRAW trudno uznać za programowanie. Skoro jednak PostScript jest językiem programowania, to można sobie wyobrazić znaki opisywane za pomocą programów.

Podział na dane i programy dodatkowo się komplikuje w przypadku znaków animowanych, ostatnio niezwykle popularnych za sprawą Internetu. Czy jest to program, czy dane? O ile w przypadku animowanych obrazków zapisanych w formacie GIF jeszcze można by się zgodzić, że chodzi o interpretowanie sekwencji obrazów jako zbioru danych, o tyle w przypadku wyświetlanej za pomocą programu javascriptowego serii ilustracji przygotowanych w formacie SVG zasadniczych wątpliwości co do charakteru zapisu uniknąć się nie da.

Rodzi to problemy natury prawnej związane z ochroną praw autorskich. Rozwój pojęć prawnych wyraźnie nie nadąża za rozwojem informatyki – świat software'u przestał się dzielić na programy i dane. Tymczasem prawo inaczej chroni programy, a inaczej pozostałe utwory autorskie. Konieczne jest zatem ustalenie, czy dany utwór (w naszym przypadku znak) jest programem czy też nie, gdyż to wyznacza ramy prawne ochrony dzieła.

Jeszcze trudniej jest z określeniem, czym jest z prawnego punktu widzenia współczesny font obwiedniowy (na przykład OpenType). Dane? Nie. Program? Nie. Biblioteka programów? Nie. Baza danych? Niezupełnie. Wydaje się, że można pró-



Rys. 6. Co robić, gdy brak znaków diakrytycznych? Zrzuty ekranu dla programów CorelDRAW wer. X6 (po lewej) i Microsoft Word wer. 2013 (po prawej); użyte fonty (od góry): Calibri, AR DECODE, Bradley Hand ITC

bować określić font jako zbiór programów opisujących kształty i sposób pozycjonowania poszczególnych znaków. Ale wolno wątpić, czy jest to określenie satysfakcjonujące z prawnego punktu widzenia.

W sumie okazuje się, że w praktyce najłatwiej jest chronić nazwę (sic!) fontu, co zakrawa na paradoks, bo trudno uwierzyć, że w większości przypadków główny wysiłek intelektualny jest wkładany w wymyślanie nazwy.

Mimo niejasnej merytorycznie sytuacji prawnicy skłaniają się ku uznaniu fontów za dzieła podlegające podobnej ochronie prawnej jak programy (polskie prawo nie uznaje fontów za programy, lecz za „utwory komputerowe, chronione prawem autorskim”), niemniej jednak co jakiś czas podnoszone są wątpliwości, czy analogia między fontem a programem jest istotnie pełna. Wątpliwości dochodzą do głosu szczególnie wyraźnie w przypadku kwestii praw autorskich do znaków dodanych lub zmodyfikowanych: co powinien oferować program obsługujący fonty w razie braku odpowiednich znaków diakrytycznych, skoro licencje na ogół zabraniają użytkownikowi jakiegokolwiek modyfikowania fontu. Powszechne stosowane rozwiązania (por. rys. 6) trudno uznać za satysfakcjonujące.

Bardzo poważnym zagadnieniem, zresztą nie tylko z prawnego punktu widzenia, jest dołączanie grafiki lub fontów do dokumentu. Obecnie w większości przypadków odbiorca dokumentu jest w stanie wyłuskać z przesłanego materiału zarówno font, jak i grafikę. W przypadku grafiki sytuacja prawna na ogół jest dość jasna, gdyż grafika jest chroniona na zasadach ogólnych. Font natomiast, jako „utwór komputerowy”, podlega specjalnym obwarowaniom prawnym i jeżeli odbiorca dokumentu nie nabył fontu, to nie ma prawa do jego posiadania, co uniemożliwia praktycznie legalne przekazanie dokumentu z dołączonym fontem. Z kolei niedołączenie fontu bardzo często prowadzi do katastrofy – takiego dokumentu może się nie dać wydrukować bądź wyświetlić poprawnie. Tak się dzieje wtedy, gdy odbiorca ma inaczej skonfigurowany komputer niż twórca dokumentu. Dotyczy to zwłaszcza dokumentów sformatowanych (PDF), aczkolwiek dokumenty, w których formatowanie odgrywa drugorzędną rolę (HTML), również mogą płatać figle na skutek użycia niezgodnych fontów (np. różnych wersji) przez odbiorcę i twórcę dokumentu.

Niektóre aplikacje tworzące dokumenty PDF dołączają fonty w postaci fragmentarycznej (*partial downloading*), tzn. dołączane są jedynie znaki wykorzystywane w dokumencie. W większości przypadków font taki po wyłuskaniu jest praktycznie nieużywalny, co stanowi niejaką ochronę. Wspomniany format OpenType określa (na zasadzie zalecenia), czy font może zostać dołączony do dokumentu, czy może zostać dołączona jedynie fragmentaryczna informacja zezwalająca na wydruk i podgląd, czy też dokument może zawierać kompletny font w wersji instalowalnej. Jest to niewątpliwie krok w stronę ochrony prawnej fontów – na ile przydatny (społecznie) w praktyce, wciąż trudno powiedzieć.

Pewnym antidotum na problemy z komercyjnymi licencjami fontów jest systematycznie rosnący zasób fontów udostępnianych na zasadach *openware*, tzn. z liberalnymi licencjami pozwalającymi na swobodne użytkowanie fontów (por. np. [5]). Wbrew pozorom nie znaczy to, że twórcy takich fontów nie mają problemów prawnych z licencjami. Ale przynajmniej ten problem został zdjęty z barków użytkownika.

Rzut oka w przyszłość

Jaki los zgotuje znakom nadchodzące tysiąclecie? Bacznie obserwując znaki na niebie i w komputerach, można dostrzec dwa scenariusze.

W pierwszym scenariuszu znaki byłyby całkowicie odseparowane od człowieka, tzn. człowiek byłby biernym użytkownikiem gigantycznej liczby *clipartów*, zrobionych przez nie wiadomo kogo, nie wiadomo na jaką okazję, zapisanych w ściśle tajnym formacie. Styczność użytkownika ze znakiem sprowadzałaby się do naciśnięcia klawisza myszy (pleniące się w telefonii komórkowej i w komunikatorach w rodzaju Facebooka niedefiniowalne i niemodyfikowalne w żaden sposób emotikony dają przedsmak tej sytuacji). Jeśli chodzi o fonty, to ten wariant jest już właściwie realizowany – użytkownik dostaje font w postaci czarnej skrzynki, niekonfigurowalnej, dającej się używać w jedynie słuszny sposób przewidziany przez producenta fontu i programu operującego fontem.

Symptodem mogącym wskazywać na ten kierunek biegu wydarzeń jest niska jakość dokumentacji fontów OpenType. Pełna dokumentacja formatu OpenType (por. [6]) to sprawiające wrażenie luźnego zbioru notatek archiwum zawierające ok. 200 plików różnego rodzaju (GIF, HTML, PDF, DOC, TXT), najwyraźniej tworzone „ewolucyjnie”. Fonty OpenType nie są dużo bardziej skomplikowane od fontów postscriptowych Type 1, których dokumentację mimo pewnych nieścisłości i niejasności można stawiać za wzór (por. [1]), w każdym razie nie na tyle bardziej skomplikowane, by uzasadniać aż taki wzrost poziomu skomplikowania dokumentacji. To, że powstawanie obecnej specyfikacji formatu OpenType odbywało się przyrostowo, nie jest zaskakujące, natomiast to, że żadna z firm zaangażowanych w ten projekt nie podjęła próby stworzenia jednolitej dokumentacji, trochę dziwi, a trochę nawet niepokoi.

Według drugiego scenariusza człowiek znalazłby się ponownie nieco bliżej znaku. Specjalizacji przeskoczyć się nie da, ale można sobie wyobrazić, że świadomy użytkownik potrafiłby dokonać stosownej rekonfiguracji znaku czy też fontu

(legalnie, zgodnie z zamierzeniami twórcy), aktywnie uczestnicząc w procesie tworzenia. Patrząc optymistycznie, można oczekiwać, że wyklucze się kiedyś standard opisu dynamicznych („inteligentnych”) znaków, które nie byłyby zamrożoną formą, ale zmieniałyby na życzenie użytkownika swój kształt w ściśle określonym zakresie w zależności od potrzeb. Zbudowany z takich znaków dynamiczny („inteligentny”) font byłby po prostu zestawem programów opisujących zgodne wizualnie znaki, a zarazem – dzięki „inteligencji” znaków – czymś w rodzaju małego systemu składu kontaktującego się w sposób konfigurowalny z klawiaturą i z systemem, nie mniej inteligentnym od fontu. Idąc dalej, można sobie wyobrazić technologię, w której dzięki temu, że podstawowy element, czyli znak, byłby programem, tworzenie hierarchii: *znaki – słowa – zdania – akapity – rozdziały – dokumenty*– itd. mogłoby się odbywać w sposób logicznie jednolity, a tym samym koncepcyjnie prostszy niż obecnie.

Brzmi to trochę jak science fiction.

Aby nie popaść w przesadny entuzjazm, zauważmy, że obecna technologia jest w prostej linii potomkiem tradycji czcionki z ołowiu i ubogiej klawiatury maszyn do pisania. Tradycja, jak już zostało wspomniane, bywa w technologii niezwykle silnym hamulcem rozwoju. Wydaje się, że upowszechnienie modelu fontu zbudowanego ze znaków o stałych kształtach, sięgającego korzeniami technologii Gutenberga, jest takim właśnie czynnikiem hamującym. Nie sprzyjają rozwojowi też inne przyjęte powszechnie w świecie komputerowym rozwiązania technologiczne: numeracja znaków, skomplikowana reprezentacja grafiki, zawiłości związane z tworzeniem i używaniem fontów, nie wspominając o tendencji twórców oprogramowania, by promować niemyślenie jako podstawowy tryb działania użytkownika...

Nie należy więc oczekiwać w najbliższym czasie rewolucyjnych zmian. Może to i dobrze, byle tylko ewolucja się nie zatrzymała. Poza tym, mimo wielu objawów ślepej ewolucji widocznych we współczesnym świecie komputerowym, nie jesteśmy na nią skazani. Możemy działać celowo, opierając się na starannej analizie możliwości komputerów i dostosowując do tego projekty przyszłych rozwiązań, zamiast wciąż powracać do przestarzałych pomysłów technologicznych.

Niemożliwe jest przewidzenie dalszego rozwoju cywilizacji. Warto jednak przypomnieć, że jedyna dotąd rewolucja o porównywalnej wadze, jaką było odkrycie uprawy ziemi, spowodowała wstrząsy i zapaści duchowe, których znaczenie trudno nam sobie wyobrazić. Czcigodny świat, świat wędrownych myślicieli, odchodził wraz ze swoimi religiami, mitologiami, koncepcjami moralnymi. Trzeba było tysięcy, by ostatecznie wygasły skargi przedstawicieli „starego świata”.

Mircea Eliade, *Kowale i alchemicy*

Bibliografia

1. *Adobe Type 1 Font Format*. http://partners.adobe.com/public/developer/en/font/T1_SPEC.PDF, dostęp 23.09.2013.
2. *ConScript Unicode Registry*. http://en.wikipedia.org/wiki/ConScript_Unicode_Registry, dostęp 23.09.2013.
3. Knuth Donald E., *The TeX book, Computers & Typesetting vol. A*. Reading Massachusetts 1986, Addison-Wesley.
4. *Open-source mathematical typefaces*. http://en.wikipedia.org/wiki/Category:Mathematical_OpenType_typefaces, dostęp 23.09.2013.
5. *Open-source typefaces*. http://en.wikipedia.org/wiki/Category:Open-source_typefaces, dostęp 23.09.2013.
6. *OpenType specification (full)*. <http://download.microsoft.com/download/E/5/B/E5B69A0B-D636-46C1-8E08-97AE4CE09CAF/otspec160.zip>, dostęp 23.09.2013.
7. *PostScript Language Reference Manual, Third Edition*. <http://www.adobe.com/products/postscript/pdfs/PLRM.pdf>, dostęp 23.09.2013.
8. Strzelczyk Piotr, *Standard Unicode w typografii*. „Acta Poligraphica” 2013 1(1), s. 41–50.
9. Vieth Ulrik, *The state of OpenType math typesetting*. 2011. http://www.gust.org.pl/bachotex/2011-en/presentations/Vieth_1_2011, dostęp 23.09.2013.
- . Eliade Mircea, *Kowale i alchemicy*, Warszawa 2007, Wydawnictwo Aletheia.

Abstract

Computer and its signs

The article attempts to analyse the phenomenon of signs as implemented in nowadays computers with a cautious look to the future, keeping in mind the historical development and role of the sign in the culture. The drawbacks of the current computer typesetting technology are pointed out. In particular, the insufficiency of Unicode numbering for typography applications, and, most importantly, the “frozen” nature of nowadays glyphs, while such applications as music or calligraphy necessitate either infinite number of glyphs or fundamentally different approach. In his paper, the author supports the latter solution suggesting that the glyphs should be just programs capable of communicating with each other.