

Mariusz Pisarski

Literatura «open source» : raport z eksperymentu

Teksty Drugie : teoria literatury, krytyka, interpretacja nr 3 (153), 96-116

2015

Artykuł został opracowany do udostępnienia w internecie przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego. Artykuł jest umieszczony w kolekcji cyfrowej bazhum.muzhp.pl, gromadzącej zawartość polskich czasopism humanistycznych i społecznych.

Tekst jest udostępniony do wykorzystania w ramach dozwolonego użytku.

Literatura *open source*: raport z eksperymentu

Mariusz Pisarski

Niniejszy artykuł dotyczy cyfrowych utworów literackich, które czerpiąc z narzędzi i pojęć stosowanych w informatyce i w kręgu inicjatyw propagujących wolny dostęp do zasobów kultury za pośrednictwem Internetu, świadomie rzucają wyzwanie tradycyjnym kategoriom literackim, takim jak nadawca, komunikat, granice dzieła, rola czytelnika, utrzymując przy tym nie tylko aspiracje, ale i – w większości przypadków – walory literackości. Biorąc za punkt wyjścia najbardziej reprezentatywne literackie projekty *open source* mam na celu przybliżenie problematyki uwalniania autorskiego tekstu, zaproponowanie ogólnej typologii otwartości dzieła oraz wstępną ocenę dotychczasowych prób ośmielenia czytelnika do tworzenia własnych utworów na podstawie udostępnionego źródła.

Open source, czyli w tłumaczeniu dosłownym „otwarte źródła”, to globalna inicjatywa promująca ideę otwartego oprogramowania dla firm, instytucji publicznych i osób prywatnych¹. Kojarzenie jej z literaturą może wydać się

Artykuł powstał w ramach projektu „Literatura polska po roku 1989 w świetle teorii Pierre’a Bourdieu”, sfinansowanego ze środków Narodowego Centrum Nauki przyznanych na podstawie decyzji nr DEC-2011/01/D/HS2/05129

Mariusz Pisarski – dr, członek Pracowni Badań Intersemiotycznych i Intermedialnych UW, redaktor w pismach „Ha!art” i „Techsty”; producent, tłumacz i teoretyk literatury cyfrowej, stypendysta SAIA w Słowackiej Akademii Nauk. Autor książki *Xanadu. Hipertekstowe przemiany prozy* (2013). Ostatnio opublikował też *Autor i dzieło w kulturze cyfrowej (na przykładzie twórczości Nicka Montforta, „Rocznik Kulturoznawczy”, 2014)*, *Od znaczenia do działania i z powrotem: nowe funkcje słowa w cyberpoezji i hipertekście* („Przegląd Humanistyczny”, 2013). Kontakt: mariusz.pisarski@techsty.art.pl

1 Historię idei i terminu *open source* najlepiej opisują sami aktywiści otwartego oprogramowania na stronach opensource.org. Analizę kulturową i ekonomiczną zjawiska można znaleźć m.in. w książce Stevena Webera *The Success of Open Source*, Harvard University Press, Cambridge–London 2004.

zabiegiem retorycznym, a sam termin *literatura open source* – oksymoroniczną metaforą, bowiem zderzenie „otwartości” oprogramowania z tradycyjną „zamkniętością” dzieła literackiego zdradza raczej niemożność pogodzenia tych dwóch światów niż ich symbiozę². Tak jednak nie jest. Prawie 30-letnia historia literatury elektronicznej, nie tyle digitalizowanej, ile „rdzenie cyfrowej”³, a zatem opartej na kodzie komputerowym, czyniącej zeń swój budulec i środek ekspresji, dowodzi, że „otwieranie źródeł” dzieła literackiego stanowi praktykę o swojej własnej tradycji, historii i o różnorodnym zasięgu. Niniejszy artykuł, biorąc za punkt wyjścia najbardziej reprezentatywne literackie projekty *open source*, ma na celu przybliżenie problematyki uwalniania tekstu autorskiego, zaproponowanie wstępnej typologii otwartości dzieła oraz próbę oceny omawianych tu praktyk literackich.

Założenia wstępne

Uważny obserwator socjotechnologicznych przemian w praktykach piśmienniczych, dokonujących się na naszych oczach za sprawą narzędzi cyfrowych, mógłby powiedzieć, że zjawisko pisarstwa otwartego jest dużo bardziej powszechne, niż się powyżej sugeruje. Po pierwsze, dzięki ustanowieniu domeny publicznej i dzięki Internetowi, który pozwala na masowe rozpowszechnianie jej zasobów, literatura dostępna jest dziś jak nigdy wcześniej. Po drugie, popularne narzędzia społecznościowe, takie jak „Oś Czasu” serwisu Facebook, automatycznie czynią z aktywności użytkownika rodzaj otwartej księgi, w której odnajdziemy drobiazgową historię jego peregrynacji w obrębie tegoż portalu i poza nim. Otwarte zdają się zatem zarówno tradycyjne formy narracyjne, jak i te przynależące do paradygmatu bazy danych, która – według

2 „Otwartość” rozumiana jest w tym artykule bardziej dosłownie niż nie tylko w *Dziele otwartym* Umberta Eco, ale też we wczesnej teorii literackiego hipertekstu. Dzieła otwarte według Eco to – poza muzyką – np. dramaty Brechta, zaprojektowane tak, by potęgować możliwe interpretacje. Dzieła otwarte według Michaela Joyce’a, autora pierwszej powieści hipertekstowej *popołudnie, pewna historia*, to takie, które zachęcają czytelnika do aranżacji materiału tekstowego według własnego uznania. W obu przypadkach nie ma mowy o odsłonięciu „źródła” tekstu. Zob. U. Eco *Dzieło otwarte*, przeł. J. Gałuszka [i in.], Czytelnik, Warszawa 1994, s. 36; M. Joyce *Of Two Minds: Hypertext Pedagogy and Poetics*, University of Michigan Press, Ann Arbor 1996, s. 41.

3 Zob. K.N. Hayles *Literatura elektroniczna: czym jest?*, przeł. M. Pisarski, „Techsty” 2011 nr 7 (1), http://www.techsty.art.pl/magazyn/magazyn7/literatura_elektroniczna_czym_jest_1.html (20.10.2014).

Lva Manovicha – paradygmat narracyjny zastępuje⁴. Zastąpienie to przejawia się dziś tendencją algorytmów społecznościowych do tego, by ukazywać dane i zdarzenia tak, aby mówiły same za siebie, zanim ktokolwiek przerobi je na formę narracyjną⁵. Rozumienie idei *open source* powinno być jednak węższe. Nie bez powodu propagatorzy otwartego oprogramowania odróżniają je od oprogramowania darmowego (*free software*). Encyklopedia Internetu i nowych technologii definiuje *open source* jako nieopatentowany i chroniony opłatą licencyjną „sposób tworzenia i dystrybucji oprogramowania oparty na udostępnianiu go wraz z kodem źródłowym”. Nieodłącznym elementem definicji otwartego oprogramowania jest aspekt społeczny tej praktyki⁶:

Specyfika dystrybucji produktów Open Source umożliwia analizę, samodzielną kompilację (utworzenie wersji wykonywalnej) oraz wprowadzanie modyfikacji do kodu przez niezależnych programistów [...]. Nieodpłatność produktów sprzyja rozwojowi idei, lecz według jej autorów pełni funkcję marginalną. Zasadniczym celem Open Source jest bowiem wolność przepływu idei i myśli ludzkiej, których nośnikiem jest między innymi oprogramowanie komputerowe.

Analizowane w niniejszym artykule przykłady otwartej literatury elektronicznej, *Toroko Gorge* Nicka Montforta, *Erica T. Carter* Jima Carpentera oraz seria polimedialnych poetyckich projektów Jasona Nelsona na wortalu Netpoetics,

-
- 4 Zapowiedziane przez Manovicha zastępowanie narracji przez bazę danych jest dziś na gruncie literatury cyfrowej szczególnie aktualne. Baza danych to ważny element składowy literackich generatorów i każdego utworu, który prezentuje swoją treść na zasadach algorytmicznych: losowo, warunkowo lub według zmiennych kryteriów wyznaczanych przez czytelnika. W przypadku serwisów społecznościowych bazy informacji o aktywności użytkowników są własnością dostawcy serwisu, podobnie jak reguły prezentacji tych informacji. Zob. L. Manovich *Język nowych mediów*, przeł. P. Cypryjański, WAIp, Warszawa 2006.
- 5 Roberto Simanowski sugeruje, że Oś Czasu, jako symboliczna forma bazy danych, jest technicznym rozwiązaniem ważnego filozoficznego problemu, przewycięża bowiem antropocentryzm refleksji humanistycznej: *jako technologia stworzona po to, by przechowywać dane wytwarzane przez człowieka bez włączania w to ludzkiego punktu widzenia, wydaje się ona (Oś Czasu, przyp. M.P) obiecywać nowy pozytywizm*. Zob. R. Simanowski *The Compelling Charm of Numbers: Writing for and Thru the Network of Data*, w: *Remediating the Social*, ed. S. Biggs, Montgomery Litho, Edinburgh 2013, s. 26.
- 6 Zob. *Netopedia. Encyklopedia Internetu i nowych technologii*, online, http://web.archive.org/web/20050502072326/http://www.ws-webstyle.com/cms.php/netopedia/open_source/open_source (14.06.2015).

charakteryzują się dwoma najważniejszymi wyznacznikami otwartości. Po pierwsze – udostępniają swój kod źródłowy, a po drugie – tworzą ramy dla aktywności czytelników, którzy w konsekwencji (w zróżnicowanej skali) stają się autorami. Wymienione cechy mocno odróżniają te utwory od zjawisk i praktyk spokrewnionych, takich jak utwory z domeny publicznej, wydelegowane tam przez autorów współczesnych, czy też utwory powstałe na podstawie tekstów pochodzących z wolnych (darmowych) zasobów literackich⁷.

Dzięki temu, że omawiane eksperymenty są rdzennie cyfrowe, a zatem ich podłożem są głębsze warstwy cyfrowości tekstu niż powszechnie się ją rozumie, np. przypisując cyfrowość do literatury zdigitalizowanej, fenomen komunikacji literackiej zostaje w ich przypadku zasadniczo przewartościowany. Najbardziej oczywistą konsekwencją są przesunięcia na linii autor – dzieło i autor – czytelnik. Ich zasięg i znaczenie w kontekście teorii literatury przedstawiam na końcu omówień każdego z trzech utworów, ukazując jego specyfikę, potencjał i stopień realizacji nadbudowującego się nad przykładowymi modelem literatury otwartych źródeł.

Przy okazji poszczególnych refleksji poddaję próbie dwie tezy, o których prawdziwości – na obecnym, wstępnym etapie rozwoju literatury cyfrowej – nie sposób przesądzać, lecz które wyłaniają się w trakcie obserwacji najlepszych, czy też najszcześliwszych, realizacji modelu otwartości. Pierwsza teza zakłada, że autor w utworze programowalnym i jednocześnie otwierającym swój program do użytku publiczności nie tyle zamienia się miejscami z czytelnikiem, jak to widziała pionierska teoria literatury cyfrowej⁸, lecz raczej wręcza mu część swoich dawnych kompetencji, usadawiając się jednocześnie o stopień wyżej i rozszerzając swoją rolę o taką, którą można porównać z rolą

7 Różnicę między tekstami otwartymi a wolnymi, analogiczną do różnic między oprogramowaniem wolnym a otwartym (*open source software*) najbardziej plastycznie ilustruje zestawienie działalności serwisu Wolne Lektury, którego celem jest darmowe udostępnienie tekstów znajdujących się w domenie publicznej z którymiś z omawianych w tym artykule przykładów. *Taroko Gorge* jest utworem zarówno wolnym, jak i otwartym, choć nie należy do domeny publicznej; *Dziady* Adama Mickiewicza są utworem darmowym, ale historycznie i strukturalnie zamkniętym; z kolei *Śmierć księżki. No Future Book* (2008) Łukasza Gołębińskiego – książka oddana przez autora do domeny publicznej, a zatem darmowa, nie zapewnia ram otwartości i zachęty do twórczego działania na bazie udostępnionego utworu, które charakteryzują prezentowane przeze mnie przykłady literatury *open source* rozumianej w sposób dosłowny, jako literatury „otwartego źródła”.

8 Zob. m.in rozdział o redefiniowaniu kategorii czytelnika w klasycznym opracowaniu George'a Landowa *Hypertext 3.0: Critical Theory and New Media in An Era of Globalization*, Johns Hopkins University Press, Baltimore 2006.

architekta budowli literackiej, reżysera czytelniczo-autorskiego spektaklu czy też konstruktora platformy kolektywnej aktywności literackiej. Teza druga zakłada, że spadek integralności utworu literackiego, a nawet jego jakości, nie jest w procesie uwalniania kodu źródłowego regułą. *Taroko Gorge* Nicka Montforta udowadnia, że możliwe są projekty, które właśnie dzięki recyklingowi tego samego kodu źródłowego, w którym zawarte są reguły kompozycji i dystrybucji semantycznej, oraz dzięki wprowadzeniu regulujących ram kolektywnej twórczości na bazie uwolnionego źródła, potrafią wręcz pielęgnować integralność i jakość.

Historia praktyk open source

Praktyka dzielenia się źródłem programu komputerowego, bez względu na podziały własnościowe czy ideowe, sięga lat 50. Tworzenie instrukcji dla prymitywnych, olbrzymich komputerów, które zaczęły trafiać do instytucji rządowych, jak choćby zaprogramowanie komputera IBM 704, tak aby wyświetlał obraz wysyłany przez radary, wymagało napisania 80 000 linii kodu. Sposobem na przyspieszenie tego zadania było kolegalne pisanie kodu źródłowego (kompilatora) przez programistów, bez względu na ich zakontraktowanie w konkurujących ze sobą firmach, po to by dopiero na tej kolaboratywnej podstawie tworzyć projekty osobne. Jeśli ta – jak nazywa ją Steven Weber – „pre-konkurencyjna” współpraca⁹ rozpoczyna historię programowania komercyjnego i obronnego, to tym łatwiej odnaleźć ją w polu niekomercyjnym. Wokół komputerów i poszczególnych platform zaczęły formować się społeczności: zrazu programistów i hakerów, a następnie artystów i literatów. Za datę początkową twórczych eksperymentów z komputerami uznaje się rok 1958, kiedy komputer TX-0 trafia do Laboratorium Badań nad Elektroniką w MIT. Trzy lata później dołącza do niego komputer PDP-1. Wynikiem programowania „rekreacyjnego”, stworzonego przez studentów, często nieformalnie i dla popisu, był m.in. pierwszy procesor tekstowy, pierwsze generatory wzorów graficznych oraz pierwsze gry, takie jak *Mouse in the Maze* czy – nieco później – *Spacewar*¹⁰. Programowanie – jak podkreśla Nick Montfort – od zawsze było aktywnością społeczną i kulturalną¹¹.

9 S. Weber *The Success of Open Source*, s. 21.

10 N. Montfort *Programming for Fun, Together*, w: *Remediating the Social*, s. 16.

11 Tamże, s. 15.

Technokultura wyrosła po 1984 roku, częściowo za sprawą firmy Apple, która swoje komputery dostarczała jako systemy zamknięte¹², zapominając o korzeniach programowania kreatywnego. Powstanie Internetu, gwałtowny rozwój branży informatycznej i tak zwanych *dot-comów* sprzyjały pogłębieniu tej niepamięci. Znamienne, że sygnał sprzeciwu wobec komercjalizacji Internetu i twórczości cyfrowej wyszedł ze środowiska artystów skupionych wokół platformy Macromedia Flash, programu do tworzenia animacji internetowej, który był zarówno efektem, jak i – jako narzędzie marketingowe – siłą napędową internetowej bańki. W 2001 roku Joshua Davis, jeden z liderów tej społeczności¹³, udostępnił za opłatą zawartość swojego twardego dysku, zawierającego 397 folderów, 3637 plików oraz 462 megabajty „surówki danych”¹⁴. Stanowiły ją gotowe do modyfikacji pliki graficzne i tekstowe animacje. *Praystation hard drive* – tak nazywał się zbiór Davisa – podkreślił, że twardego dysku artysty jest w stanie pomnożyć zarówno kapitał kulturowy, jak i ekonomiczny swojego właściciela, podkreślając przy tym ważny aspekt idei *open source*, a mianowicie, że udostępniany powszechnie utwór nie musi być darmowy, aby był otwarty¹⁵. Choć wydanie twardego dysku amerykańskiego artysty zapisało się w historii sztuki internetowej, to *Praystation* nie wywołało burzliwego, otwartego dialogu derywatów z tekstem

12 Twory zaliczane do pierwszej generacji literatury elektronicznej, a także dzieła prekursorskie – jak cyfrowa poezja bpNichola – powstawały na komputerach osobistych Apple’a. Programy Hypercard i Storyspace, na które pisano animowane tomiki poezji i hipertekstowe powieści, podobnie jak systemy Macintosh, były programami zamkniętymi i komercyjnymi. Powieści napisane w Storyspace, np. wydana w Polsce *popołudnie, pewna historia*, sprzedawane były jako artefakty drukowane, z okładką i z dołączoną dyskietką. Oferowanie powieści za darmo, wraz z udostępnianiem jej w Internecie (np. *The Unknown* Scotta Rettberga, Dirka Strattona i Williama Gillespie, zwiastowało nadejście nowych generacji e-literatury i to one dopiero odkryją uroki *open source* wraz z jego tradycją. Zob. m.in K.N. Hayles *Literatura elektroniczna: czym jest* (20.10.2014); historię „zamykania” systemów komputerowych opisuje też Lori Emerson w swojej książce *Reading Writing Interfaces. From the Digital to Bookbound*, University of Minnesota Press, Minneapolis 2014, s. 64-85.

13 Więcej na temat środowiska grafików tworzących w programie Flash zob. D. Leishman *Out of Place. Digital In-grouping*, w: *Remediating the Social*, s. 131-132.

14 *Praystation harddrive*, strona *Praystation Year 2* w katalogu wydawnictwa Eastgate Systems, <http://www.eastgate.com/catalog/Praystation.html> (20.10.2014).

15 Steven Weber, wyjaśniając różnicę między „darmowym” a „wolnym”, po angielsku na określenie obu terminów używa się słowa „free”, posługuje się anegdotą programistów, którzy zachęcają, by mówiąc o *open source*, myśleć nie o darmowym piwie, ale o swobodzie wypowiedzi (*think „free speech” not „free beer”*). Zob. S. Weber *The Success of Open Source*, s. 5.

źródłowym¹⁶. Jak się okazało, udostępnienie zasobów źródłowych, nawet gotowych do komercyjnego wykorzystania, nie oznacza automatycznie lawiny naśladownictw. Kryje się za tym szerszy problem praktyk *open source*, jaki może stwarzać chęć pozostawienia autorskiej „pieczęci” na udostępnianym dziele, oraz brak sygnałów otwartości na poziomie poetyki utworu, który utrudnia twórcom odzew odbiorcy.

Jim Carpenter i otwieranie poezji

Ukazanie się w 2006 roku pierwszego tomu *Electronic Literature Collection*, zawierającego 60 wybranych przykładów literatury elektronicznej, jest ważną cezurą w historii rodzącego się, autonomicznego pola e-literatury¹⁷. Darmowy i dostępny zarówno online, jak i offline tom prezentował utwory nowej generacji, zbudowane za pomocą powszechnie dostępnych narzędzi, takich jak html, css, javascript i Flash. Stało to w jawnym kontraście wobec klasycznych powieści hipertekstowych, wydawanych przez Eastgate Systems, dystrybuowanych odpłatnie na zamkniętych platformach¹⁸. Jednak dla niektórych obserwatorów nawet tak odważny gest, jak darmowa dystrybucja literatury, nie był wystarczająco radykalny. Kolekcja zawierała głównie utwory nieudostępniające swoich kodów źródłowych. Wytknął to autorom antologii Jim Carpenter, uznając to za brak wrażliwości autorów na ideę otwierania źródeł, za przejaw własnościowego sposobu myślenia i „nieudolną próbę trzymania się czegoś, co oprogramowanie komputerowe czyni niemożliwym: obecności oryginału”¹⁹.

16 Utwór ten był pokazywany np. na wystawie „Digital Revolutions” w Barbican Centre w Londynie, 3.06-14.09.2014.

17 *Electronic Literature Collection vol. 1*, ed. K.N. Hayles, S. Rettberg, N. Montfort, S. Strickland, Electronic Literature Organization, 2006, <http://collection.eliterature.org/1/> (20.10.2014).

18 Darmowa kolekcja literatury elektronicznej nie powstałaby, gdyby Mark Bernstein, szef wydawnictwa Eastgate Systems, przeciwny darmowym publikacjom, nie zrezygnował z funkcji członka zarządu Electronic Literature Organization – przyznał Scott Rettberg. Zob. S. Rettberg *Developing an Identity for the Field of Electronic Literature Reflections on the Electronic Literature Organization Archives*, „Dichtung Digital” no. 41, <http://www.dichtung-digital.de/en/developing-an-identity-for-the-field-of-electronic-literature-reflections-on-the-electronic-literature-organization-archives/> (20.10.2014).

19 Carpenter zapewniał w rozmowach o generatorze o kilku przypadkach przyjęcia sztucznie wytworzonych wierszy do publikacji w amerykańskich czasopismach poetyckich. Efekty poetyckiej produkcji generatora są dziś dostępne m.in. w internetowej bazie literatury elek-

Carpenter pisze nie tyle z pozycji krytyka, ile jako praktyk. W chwili wydania kolekcji literatury elektronicznej amerykański poeta, programista i nauczyciel angielskiego przygotowywał się do sieciowej publikacji najnowszej wersji swojego *opus magnum* – innowacyjnego generatora poezji *Erica T. Carter*. Dotychczasowe generatory produkowały tekst najczęściej na zasadzie losowego rozdania lub przez emanacyjną, nieinteraktywną autoprezentację generowanych wersów i strof. Tymczasem program tworzył wiersze na podstawie ustanowionych przez użytkownika haseł-kluczy i tematów wyjściowych. Zamiarem autora było nadanie procedurom generatywnym bardziej ludzkiego, naturalnego charakteru. Miało temu służyć wydzielenie semantyki i gramatyki z pliku źródłowego i oparcie ich na osobnych bazach danych, z jednej strony – zasobów słownych, pochodzących od różnych autorów z domeny publicznej, a z drugiej – z bazy struktur gramatycznych i retorycznych. Modularny generator – mimo wytwarzania wielu utworów nie do końca udanych²⁰ – był w stanie wyprodukować wiersze na tyle „ludzkie”, że można je było pomylić z przykładami awangardowej poezji współczesnej, napisanej przez poetów z krwi i kości²¹. Głównym celem twórcy było tworzenie wierszy, które są w stanie przejść poetycki test Turinga, czyli zakwalifikowanie do publikacji w czasopiśmie literackim:

tronicznej ELMCIP w postaci autorskiej kompilacji: <http://elmcip.net/sites/default/files/files/attachments/work/erica.t-project-cw.pdf> (20.10.2014). Oto, dla przykładu, wiersz doceniony przez badaczy analizujących gramatykę komputerowej generatywności:

A kind of background

A kind of wonder
A kind of mica
A sort of background

Cyt. za: M. Hvidtfeldt Christensen *Grammars for Generative Art (part III)*, Syntopia. Generative art, 3D fractals, creative computing, 13.01.2009, <http://blog.hvidtfeldts.net/index.php/2009/01/grammars-for-generative-art-part-iii/> (20.10.2014).

- 20 Zobaczyć np. krótki raport z testowania generatora Carpentera przez Nicka Montforta w: tegoż *Erica T. Stirs*, „GrandTextAuto”, <http://grandtextauto.org/2007/01/29/erica-3-stirs/> (20.10.2014).
- 21 Carpenter wyjaśnia przyczyny swojej decyzji na przestrzeni kilku, pisanych w coraz dłuższych odstępach wpisów na blogu. Ostatecznie staje się jasne, że uwolnienie kodu jest niemal równoznaczne z zaprzestaniem pracy nad generatorem. Zob. J. Carpenter *A couple of things*, „The prosthetic imagination”, 12.10.2008, http://theprostheticimagination.blogspot.co.uk/2008_10_01_archive.html, [20 X 2014].

Wiersze Eriki znaczą nie dlatego, że Erica jest kimś znaczącym, lecz z tej racji, że sam język znaczy. Wszystko co tu powstaje uznajemy za domenę publiczną. Traktujcie te teksty jako swoje, mieszajcie ich fragmenty z własnymi próbkami. Rozkładajcie je na części i układajcie w nowe konfiguracje. Wysyłajcie wyniki tych działań do czasopism poetyckich i małych gazetek. Jeśli wam się uda – szepnijcie mi słowo. Dotrzymam tajemnicy, jeśli tego chcecie, lub rozgłoszę wszem i wobec.²²

Fragmenty kodu odpowiedzialnego za produkcję wierszy *Eriki* poeta-programista udostępniał stopniowo, by w siedem lat po rozpoczęciu projektu uwolnić kod źródłowy. Konsekwencje otwarcia źródeł, podobnie jak w przypadku Davisa, okazały się minimalne. Nie pomógł w tym nawet bezprecedensowy wydawniczy happening. W październiku 2008 roku poeta i redaktor awangardowego portalu Ubu Web Kenneth Goldsmith ogłosił publikację „pirackiej”, gargantuicznej w rozmiarach antologii *Issue 1*²³. Redaktorzy antologii, Stephen McLaughlin i Jim Carpenter, na jej 3785 stronach zamieścili ponad 3000 wierszy, które podpisano nazwiskami dawnych i żyjących poetów. Jak można się domyślać, wiersze zostały napisane przez *Erikę*, co nie od razu i nie przez wszystkich zostało zauważone. Reakcje poetów, którym bez pytania przypisano po jednym wygenerowanym wierszu, były mieszane, podobnie jak ocena jakości zawartych w antologii wierszy²⁴. Sam fakt, że część tej poezji podobała się publiczności, wypada uznać za sukces generatora. Sieciowa publikacja *Issue 1* była łabędzim śpiewem *Eriki T. Carter* i jej uwolnionego kodu. Dwa lata później, po wyłączeniu serwera, na którym znajdowała się baza danych potrzebna generatorowi do wytwarzania nowych tekstów, utwór zniknął z sieci. Modularny charakter stworzonego przez Carpentera systemu, a zatem rozpisanie generatora na osobne bazy danych (semantyka i gramatyka), okazało się zarówno siłą, jak i słabością projektu. Gdy zabrakło jednego z ogniw, całość przestała działać. Technologia, która stała za sukcesem generatora, przyczyniła się zatem do klęski przedsięwzięcia spod znaku open source!

Generator wierszy Jima Carpentera to dzieło bezprecedensowe. Maszynę do produkcji „akceptowalnej” jakości utworów autor poszerzył o dodatkowy

22 J. Carpenter *Erica T. Carter*, [https://web.archive.org/web/20080608105103/http://etc.wharton.upenn.edu:8080/Etc3beta/Erika.jsp\(20.10.2014\)](https://web.archive.org/web/20080608105103/http://etc.wharton.upenn.edu:8080/Etc3beta/Erika.jsp(20.10.2014)) [tłumaczenie tego i kolejnych cytatów ze stron www własne – M.P.]

23 Zob. komentarze do wpisu Kennetha Goldsmitha, tamże.

24 Zob. komentarze do wpisu Kennetha Goldsmitha, tamże.

wymiar, udostępniając jej kod oraz zachęcając do przeróbek i udoskonaleń. Czytelnik mógł być „współtwórcą” w wymiarze dotąd niepraktykowanym. W zaproponowanym przez Espena Aarsetha modelu cybertekstu, w którym aktywność czytelnika podzielona zostaje na interpretacyjną, eksploracyjną, konfiguracyjną i tekstoniczną²⁵, implikowany odbiorca *Eriki T. Carter* dostępuje najwyższej roli: żywiąc maszynę produkującą wiersz swoim własnym materiałem wyjściowym, personalizuje nie tyle odczytanie wiersza i jego kompozycję, ile samą zawartość tekstową. Otwarcie kodu źródłowego potęguje tekstoniczną funkcję czytelnika i zezwala na ingerencję w wewnętrzne mechanizmy programu, jego bazy tekstowej i reguły składniowe. Choć do tak zaawansowanych praktyk nie doszło²⁶, generator Carpentera zachęca do rewizji, poszerzeń i uściśleń podstawowych pojęć używanych w teorii odbioru, takich jak „współpraca” i „aktywność”, „aktualizacja” i „konkretyzacja”, „spotkanie”, „dialog” czy – *notabene* – „interakcja”²⁷. Walor ten projekt ten dzieli z dwoma kolejnymi, które tu omawiam.

Jason Nelson: uwalnianie instrumentów tekstowych

Lori Emerson w książce *Reading Writing Interfaces. From the Digital to Bookbound* odświeża na gruncie refleksji literackiej zaproponowaną przed Theodora Nelsona kategorię *thinkertoy*²⁸. Jest to rodzaj ponadmedialnego interfejsu, odnoszącego się zarówno do zjawisk cyfrowych, jak i do odnajdywanych na przestrzeni całej historii literatury, w którym procesy pisania, czytania, działań autorskich i działań odbiorcy odgrywają równoprawną rolę²⁹. Owa

25 E. Aarseth *Cybertekst. Spojrzenia na literaturę ergodyczną*, przeł. M. Pisarski, P. Schreiber, D. Sikora, M. Tabaczyński, Korporacja Ha!art – Miejskie Centrum Kultury, Kraków–Bydgoszcz 2014, s. 75.

26 Poważniejsze przekształcenia strukturalne czekałyby *Erikę T. Carter* przede wszystkim w odmiennym obszarze językowym, np. w ramach „przekładu” generatora na język polski. Programistyczne i językowe zmiany musiałyby uwzględnić wymianę zasobów leksykalnych, wzorów gramatycznych i frazeologicznych, a także odpowiednie poszerzenie kodu, aby zasobami tymi rozdysponować. Choć sam autor mógłby nie rozpoznać tak odległego potomka *Eriki*, idea *open source* miałyby w takim projekcie mocne potwierdzenie.

27 Henryk Markiewicz *Odbiór i odbiorca w badaniach literackich*, w: *Wymiary dzieła literackiego*, Universitas, Kraków 1996, s. 250-253.

28 Th.H. Nelson *Computer lib / dream machines*, w: *The New Media Reader*, ed. N.-W. Fruin, N. Montfort, MIT Press, Cambridge–London 2003, s. 330.

29 Zob. L. Emerson *Reading Writing Interfaces...*, s. 150-152.

konstrukcja zakłada przemieszczenia, ponowne konfiguracje, majsterkowanie przy tekście, by – jak zakładał twórca terminu „hipertekst” – „wyobrażać sobie złożone alternatywy”. *Thinkertoy* to zatem rodzaj maszyny budowanej ze słów, której użycie zwykło się oddelegowywać w sferę procesu twórczego poprzedzającego publikację. Emerson podaje jako przykład praktyki Emily Dickinson, która niektóre z wierszy układała na fiszkach papieru w kilku wariantach, przypinając do nich dodatkowe, alternatywne wersy³⁰. Media cyfrowe w ogóle, a literatura elektroniczna w szczególności, majsterkowanie przy tekście wysunęły z cienia, przeniosły w sferę odbioru i uczyniły głównym punktem interaktywnej lektury.

Rozwiniętym przykładem tekstowych aparatów dających sobą manipulować są drobne graficzno-dźwiękowe gry literackie, tworzone przez australijskiego artystę Jasona Nelsona. W *This is How You Will Die* z 2005 roku Nelson zachęca czytelnika, by – posługując się interfejsem przypominającym losowanie w automacie hazardowym na żetony – poznał scenariusz swojej własnej śmierci³¹. Wydzielone tekstowe segmenty przypisane są do miejsca zdarzenia, sposobu odejścia z tego świata czy konsekwencji, jakie śmierć czytelnika-gracza przynosi (np. „sprzedaż twoich tomików poezji wciąż nie jest w stanie opłacić pojedynczego obiadu”). W *with love, from a failed planet* z 2011 roku Nelson zaprasza do manipulacji wolno obracającym się, półprzezroczystym globusem wyklejonym znanymi flagami i markami światowych korporacji. Każdy z elementów graficznych skrywa tekst krótkiego, postapokaliptycznego opowiadania, które wprowadza czytelnika w możliwy świat niedalekiej przyszłości. Twórczość Nelsona obfituje w podobne utwory. Są one krótkie, polisemiotyczne i stanowią hybrydę elementów ludycznych i narracyjnych. Większość, jako utwory stworzone w omawianym programie Flash, z definicji nie otwiera przed użytkownikiem swojego kodu źródłowego. Nic dziwnego, że w sześć lat po debiucie, na łamach kolaboratywnego serwisu krytyczno-literackiego Netpoetic, Nelson ogłasza uwolnienie całej serii swoich *thinkertoy*s³². Każda z sześciu literackich minigier, oprócz linku do kodu źródłowego, została opatrzona szczegółową instrukcją modyfikacji pliku. Jednak raz jeszcze, podobnie jak w przypadku generatora Jima Carpentera, entuzjazm autora udostępniającego czytelnikowi rezultat swoich

30 Tamże, s. 139.

31 J. Nelson *This is How You Will Die*, w: *Electronic Literature Collection vol. 2*, 2011, <http://collection.eliterature.org/2/> (20.10.2014).

32 Zob. <http://www.netpoetic.com>.

programistycznego i poetyckich eksperymentów nie szedł w parze z entuzjazmem odbiorców. Sam autor nie odnotował żadnego odzewu ze strony potencjalnych „korekcjonerów” (termin Edwarda Balcerzana). Wizualno-dźwiękowo-poetyckie programy były wielofazowe, nie miały oczywistych reguł interakcji, wymagały od użytkownika pewnych kompetencji oraz zakupu komercyjnego oprogramowania³³. Czyżby zatem uwolnione poetyckie animacje miały wpisaną w sobie niemożność czytelniczej przeróbki? A może sam fakt, że praktyka otwierania źródeł odbyła się w cieniu zamkniętej, komercyjnej i sprzecznej z ideą otwartości platformy, jaką do dziś pozostaje program Flash³⁴.



Ilustracja 1. *layeringFollow* – jedna z uwolnionych literackich instalacji Jasona Nelsona, wykonanych w programie Flash.

Przyczyn może być wiele, ale w przypadku Jasona Nelsona jedna wydaje się dominująca. We wstępie do *reDimensionalCube*, w którym tekst rozkłada się wokół centralnego, modułowego sześciangu przypominającego kostkę

- 33 Ten ostatni element, w przypadku dzieł ludycznych, lokujących się na granicy gier i literatury, stanowi *sine qua non* poetyckości dzieła w kontekście odbioru wyznaczonym przez jasne reguły komunikacji na linii system – gracz w grach komercyjnych. Literatura cyfrowa zawsze będzie zatem iść pod prąd konwencji ogólnie przyjmowanych przez technokulturę.
- 34 Byłaby to wspólna i dla Nelsona, i dla Davisa przyczyna względnego faska uwalniania kodu. Widoczną w ostatnich latach tendencją w gronie cyfrowych literatów jest odchodzenie od programu Flash jako narzędzia literackiego. Powodem jest nie tylko zamknięty charakter platformy, lecz niemożność odtworzenia utworów powstałych we Flashu na urządzeniach mobilnych. Tematowi temu poświęcono osobną sesję (*Electronic Literature After Flash*) na dorocznej konwencji MLA w 2014 roku, jednym z najbardziej prestiżowych spotkań literaturoznawców w USA, organizowanych przez Modern Language Association. Zob. http://www.mla.org/cfp_detail_6103 (20.10.2014).

Rubika, australijski poeta pisze: „Od dawna fascynuję się sześcianami, być może dlatego, że większość wierszy drukowanych nadbudowana jest na kratkach, w rytmie opadających wzdłuż kartki linijek. Sześcian transponuje tę linearną kratkę na wielowymiarową przestrzeń”³⁵. Nie każdy musi podzielać zainteresowanie autora oryginału geometrią poetycką, a tym bardziej chcieć nanosić własny wiersz na cudzą, na dodatek wyraźnie autorską, konstrukcję. Przy każdym z uwolnionych eksperymentów Nelson umieszcza swój wyraźny podpis, podkreślając autorski charakter nie tylko idei, ale i wykonania. Owa autorska pieczęć na utworze jest odcisnięta na wielu – prawdopodobnie zbyt wielu – poziomach dzieła. W konsekwencji, przy uwzględnieniu wymienionych aspektów technicznych, próg trudności w naturalizacji utworu jest podniesiony za wysoko. Jednym ze środków zaradczych mogłaby być zmiana kontekstu odbioru, np. przez umieszczenie uwolnionych źródeł i ich przeróbek w ujednoczonej ramie wystawienniczej, przykładowo tak, by obok każdego z dzieł oryginalnych umieszczać linki do derywatów. Duch współzawodnictwa szedłby wówczas w parze z możliwością publikacji i z poczuciem przynależności do podobnie zorientowanej społeczności twórczej. Odkrycie opartego na takich właśnie przesłankach klucza do działań *open source* historia literatury elektronicznej przypisze innemu utworowi.

Wąwóz Taroko Nicka Montforta

W tym samym roku, w którym Jason Nelson udostępnia kod źródłowy swoich ludycznych, multimedialnych eksperymentów literackich, Nick Montfort publikuje prosty wiersz generatywny *Taroko Gorge*, którego tytuł i temat nawiązują do wąwozu w jednym z parków narodowych na Tajwanie. Jak poprzednie generatory Montforta (np. *Duet*³⁶), *Taroko Gorge* jest utworem świadomie oszczędnym, zarówno pod względem językowym, jak i programistycznym. Kod strony www wyświetlającej wiersz składa się ze 135 linijek i nieco ponad 50 słów. Zasoby leksykalne rozdzielono między zbiory semantyczne i gramatyczne, np. jeden zawierał elementy krajobrazu znajdujące się „ponad” punktem widzenia obserwującego podmiotu (mgła, las, grzbiet), z kolei zbiór „pod” zawierał rzeczowniki desygnujące dolne partie

35 J. Nelson *reDimensionalCube*.

36 N. Montfort *Duet*, przeł. A. Małecka, P. Marecki, http://www.ha.art.pl/nick_montfort/duet.html (20.10.2014).

wąwozu (basen, żyła, kamień). Sytuację liryczną dynamizował zbiór imperatywów, takich jak „biegnij, wejdź, nadepnij”, które miały kierować uwagę obserwatora w określoną stronę, pogłębiając sugestywność tematu głównego – spaceru po wąwozie³⁷. Odpowiedzialny za dystrybucję słów w wersach, i wersów na ekranie, był prosty kod javascript, w którym Montfort ustalił tryb wyświetlania się wiersza i sposób łączenia członów zbioru w układy wersowo-syntaktyczne. Linijki wiersza pojawiają się jedna po drugiej, a gdy na dole zabraknie dla nich miejsca – znikają u góry ekranu. Temporalny tryb czytania się wiersza praktycznie nie ma końca, a losowy dobór segmentów każdego ze zbiorów zapewnia względną różnorodność i niepowtarzalność wersów i strof.

Taroko Gorge nie był wierszem *open source*, choć jak większość generatywnej poezji Montforta jego kod źródłowy można było odsłonić z poziomu przeglądarki internetowej. Prawdopodobnie byłoby tak dalej, gdyby nie „przywłaszczenie” i przeróbka kodu źródłowego, których dokonał Scott Rettberg – przyjaciel Montforta, twórca e-literatury, autor m.in. hipertekstu *The Unknown* (1998) oraz stworzonej wspólnie z autorem *Taroko Gorge* powieści wlepkowej *Implementation* (2006). Niedługo po opublikowaniu wiersza Montfort otrzymał od Rettberga zmienioną wersję, zamieszczoną na prywatnej stronie tego ostatniego. Nazwisko Montforta zostało tam przekreślone i zastąpione nazwiskiem kolegi. Składnię i zbiory leksykalne zachowano, lecz temat i semantyka zostały radykalnie zmienione. Swoją dywersję Rettberg wyjaśnił następująco:

Nigdy nie byłem fanem poezji natury, natomiast zawsze kusilo mnie, by przerobić któryś z utworów Nicka. Otwierając program w przeglądarce i wybierając proste polecenie „pokaż źródło”, bardzo szybko uzyskałem dostęp do tekstonek *Taroko Gorge*. Zacząłem się przyglądać konstrukcji programu i jego parametrom. Bez pytania autora o zdanie przepisałem to dzieło na nowo, najpierw przez podmianę, moim zdaniem, zbyt oszczędnego słownictwa na zdecydowanie bardziej bogate. Ciekaw byłem, czy uda mi się – wychodząc od podstawowej kompozycji – zastąpić ten minimalistyczny wiersz o porządku natury, w którym człowiek jest prawie nieobecny, wierszem maksymalistycznym o chaotycznym mieście,

37 Szczegółowy opis struktury *Taroko Gorge* można znaleźć w dysertacji doktorskiej Talana Memmota *Digital Rhetorics and Poetics. Signifying Strategies in Electronic Literature*, Malmö University, Malmö 2011, s. 119-128.

w którym ludzie są prawie wszędzie [...]. I tak oto w ciągu jednej nocy z *Taroko Gorge* powstał *Tokyo Garage*.³⁸

Montfort potraktował rzucone swojemu tekstowi wyzwanie tak, jak przewidział Rettberg: jako literacki happening i próbę przetestowania postawy *open source*. W rezultacie – po przyjęciu przez tego pierwszego roli gospodarza i uwzględnieniu remiksu na stronie tytułowej oryginału oraz po opisanu zdarzenia na wspólnym blogu GrandTextAuto³⁹ – *Taroko Gorge* i jego wielkomięjska przeróbka *Tokyo Garage* wywołały falę działań derywatywnych, które trwają do dziś. Brytyjska poetka J.R. Carpenter stworzyła trzy różne remiksy, zmieniając temat m.in. na krajobraz morski Kornwalii; Talan Memmott napisał turpistyczny *Toy Garbage*; Kathleen Inman Berens umieściła sytuację liryczną w kuchni; Leonardo Flores, w remiksie metatekstualnym, opisuje zmagania nad *Taroko Gorge* autorów dotychczasowych przeróbek. Utwór Nicka Montforta ma też polską wersję⁴⁰. W sumie program wywołał 24 remiksy wykorzystujące, z drobnymi zmianami, ten sam kod źródłowy.

Taroko Gorge wypada uznać za wzorowy przykład przełożenia praktyk „otwartych źródeł” na działalność literacką. Pojedynczy utwór, dzięki udostępnieniu kodu źródłowego, był w stanie wywołać niekończącą się serię twórczych przeróbek, z których każda odnosi się do oryginału i jednocześnie wchodzi w relację z przeróbkami poprzednimi. Ciekawie rysuje się przede wszystkim los oryginału. Gdyby nie inicjujący gest Rettberga i następujące po nim pospolite ruszenie na rzecz tworzenia coraz to nowych remiksów, *Taroko Gorge* mogłoby pozostać jednym z wielu generatorów Montforta. Tymczasem trawestacje i przekłady uczyniły zeń najczęściej omawiane dzieło amerykańskiego programisty-poety. Kulturowa ekonomia *open source* rządzi się niekonwencjonalną logiką. Wyrzeczenie się praw do kodu oraz oddanie dzieła na własność literackiej publiczności podnosi rangę utworu i symboliczny kapitał autora. Impakt praktyki otwierania źródeł oddziałuje też poza dziełem. *Taroko Gorge* aktywizowało i zorganizowało wokół

38 S. Rettberg *A Response to Nick Montfort's „Programming for Fun, Together”*, http://elmcip.net/sites/default/files/files/attachments/criticalwriting/a_response_to_nick_montfort_o.pdf (20.10.2014).

39 S. Rettberg *Tokyo Garage*, <http://grandtextauto.org/2009/03/17/tokyo-garage/>, 17.03.2009 (20.10.2014).

40 N. Montfort *Wąwóz Taroko*, przeł. P. Marecki, http://www.ha.art.pl/nick_montfort/wawoz_taroko.html, 2014 (20.10.2014).

siebie powiększającą się literacką społeczność, przemieniając czytelników w autorów, autorów w czytelników, i krytyków – często po raz pierwszy – w autorów.

Ów optymalny wręcz scenariusz nie rozegrał się przypadkowo. Montfort i Rettberg zapracowali na niego, tworząc silną publiczność literacką i stabilny kontekst. Blog GrandTextAuto, na którym ogłaszane były kolejne trawestacje *Taroko Gorge*, swoją długoletnią działalnością dopracował się rangi salonu literackiego, którego członkowie – wśród nich Montfort, Rettberg i J.R. Carpenter – dzielili się z nowościami ze świata literatury, próbkami własnej twórczości i własnymi refleksjami. Teoretycznoliterackie dyskusje z czytelnikami w komentarzach nie były rzadkością. Po tym, gdy Rettberg zrobił swój pierwszy remiks i prowokacyjnie przekreślił nazwisko Montforta we własnej wersji generatora, kolejny odzew, tym razem ze strony Carpenter, był naturalnym przedłużeniem praktyki blogowej. Z kolei odzew czytelników, w większości innych cyfrowych literatów, został ułatwiony przez sposób powstania pierwszego remiksu. Prowokacja Rettberga była wyzwaniem rzuconym autorowi oryginału, zaproszeniem innych do rywalizacji, a nawet czymś na kształt otwarcia literackiego konkursu, przypominającego np. literackie zabawy w pisanie limeryków, w których na polskim gruncie brali udział Stanisław Barańczak i Wisława Szymborska. Na dodatek *Taroko Gorge* to utwór prosty, jego kod jest czytelny i szybko przyswajalny dla laików. Co ważne, w działaniu na otwartym kodzie programu Nicka Montforta wpisana była obietnica nagrody w postaci umieszczenia nazwiska autora remiksu na stronie oryginału. Aspekt nobilitacyjny był niewątpliwą dźwignią przedsięwzięcia. Większość z wymienionych wyżej elementów była obecna także w projektach Jasona Nelsona i Jima Carpentera, jednak nie wystąpiły one tam w proporcji i w natężeniu, które obserwujemy w *Taroko Gorge*.

Taroko Gorge to modelowy przykład uwolnienia utworu wraz z jego kodem, kiedy to działanie autora nie ogranicza się do oddania dzieła czytelnikowi, choćby z instrukcją obsługi (Carpenter, Nelson), lecz przechodzi do dalszych faz. Montfort aktywnie zachęca do tworzenia derywatów, pełni rolę kuratora i wydawcy gotowych przeróbek, bierze udział w toczącej się – zainicjowanej przez Rettberga – refleksji nad rezultatami spontanicznie powstałej inicjatywy poetyckiej, swoistego rodzaju „turnieju”, którego stawką jest prezentacja własnej realizacji oryginalnego pomysłu na generator obok uznanych i wpływowych, a także aspirujących twórców literatury elektronicznej.

Problematyka wartościowania

Wskazane w tym miejscu jest pytanie o jakość poszczególnych prac oraz o wartościowanie przedsięwzięć spod znaku *open source*⁴¹. Jeśli uznać, że generatywny wiersz Montforta jest utworem udanym, a nie ma podstaw, by mu tego odmawiać (autora publikują wydawnictwa o światowej renomie), to powstałe na jego bazie utwory pochodne, nie tylko ze względu na społecznościowy, turniejowy kontekst, ale także dzięki ściślemu określeniu dystrybucji motywów i określeniu relacji przestrzennych wiersza, mają szansę jakość tę zachować. Poetycki derywat określa się tu wobec oryginału raczej jako utwór partnerski, będący pełnoprawnym uczestnikiem po dialogu, niż jako wiersz ucznia wobec utworu mistrza czy epigona wobec prekursora. Problem jakości literatury otwartych źródeł musi być rozpatrywany w każdym przypadku osobno i najczęściej zależy od roli, jaką po uwolnieniu utworu określi dla siebie autor oryginału. Kwestią bardziej uniwersalną, dotyczącą zarówno przedsięwzięć *open source*, jak i literatury cyfrowej w szerszym zakresie, pozostaje wartościowanie. Dla literaturoznawcy wyrosłego na metodologiach XX wieku i instytucjonalnych ramach tradycyjnej humanistyki może stanowić ono nie lada wyzwanie. Jak bowiem oceniać dzieło, które jest zarówno wierszem, jak i programem komputerowym, jeśli kompetencjom literaturoznawczym nie towarzyszy znajomość informatyki? Jeśli wiążące pozostają dwa wymiary wartościowania⁴², historyczny (biorący za punkt wyjścia horyzont odbioru, środowisko i konteksty z czasu powstawania dzieła) i ponadhistoryczny (przykładający dzieło do

41 Jeśli zgodzimy się, że samo to pytanie jest dobrze postawione. Można je uznać za zabarwione Turingowską koncepcją informatyki, która zdominowała pierwszą fazę refleksji nad rolą i znaczeniem komputera w kulturze. W jej centrum znajdowała się problematyka sztucznej inteligencji, a celem było stworzenie komputerów, które będą zachowywać się na podobieństwo człowieka. Stąd pierwotny test Turinga. Rozwój technologii informacyjnych przewartościował te aspiracje, odpodmiotowiając komputer, który dziś jest przede wszystkim narzędziem realizującym zamysł programisty, a nie jego autonomicznym partnerem. Jest to ważne dla koncepcji autorstwa, która w tej drugiej perspektywie zostaje nobilitowana, czyniąc autora konstruktorem (reżyserem) poszerzonego doświadczenia lekturowego, zakładającego interakcję w obrębie interfejsu, przeszukiwanie bazy dzieła itp. Siłą rzeczy kompetencje i warsztat autora wychodzą daleko poza tradycyjny akt przelania idei na papier czy obmyślenia kompozycji narracyjnej. Por. m.in. J.D. Bolter, D. Gromala *Windows and Mirrors. Interaction Design, Digital Art, and The Myth of Transparency*, MIT Press, Cambridge 2003, s. 46-48; P. Celiński *Postmedia. Cyfrowy kod i bazy danych*, Wydawnictwo UMCS, Lublin 2013, s. 19-21.

42 Zob. J. Sławiński *Analiza, interpretacja i wartościowanie dzieła literackiego*, w: *Prace Wybrane*, t. 6: *Próby teoretycznoliterackie*, Wydawnictwo Literackie, Kraków 1976, s. 36-37.

zmiennych warunków odbioru), to ów pierwszy, zapraszający badacza *Taroko Gorge* w progi MIT, zachęca do przyjrzenia się kulturze komputerowych laboratoriów uczelni, z których wyrasta generator Montforta, wydaje się być poza zasięgiem badacza literatury. Nic bardziej mylnego! Rozwijana od lat teoria literatury cyfrowej, ufundowana na inicjatywach wysoce interdyscyplinarnych i której pionierzy posiadali kompetencje z obu dziedzin⁴³, wypracowała własne, przyjazne humanistom narzędzia do badania nowych, poszerzonych o kod komputerowy form literackich. Radykalne i wprowadzające podziały deklaracje, zabraniające badaczom, którzy nie potrafią napisać kodu komputerowego, określania się mianem „cyfrowego humanisty”, są przesadzone⁴⁴. „Dziś wszyscy jesteśmy badaczami cyfrowymi” – stwierdza Jane Winters⁴⁵. Fenomen literatury *open source* jest jednym z wielu symptomów wspierających tę tezę.

Open source po polsku

Literatura „otwartych źródeł” to zjawisko pojemne. Sugerowaną miarą powodzenia przedsięwzięcia pod szyldem *open source* jest społeczny, kreatywny oddźwięk w postaci dzieł derywatywnych, które gest otwarcia kodu inicjuje. W 2008 roku Łukasz Gołębiowski przekazał w domenę publiczną książkę o przyszłości czytelnictwa *Śmierć książki. No Future Book*⁴⁶. W tym samym roku propagator wolnej kultury Jarosław Lipszyc opublikował *Mnemo techniki* – zbiór wierszy będących remiksami wybranych źródeł Wikipedii, udostępnionych w Internecie w wersji edytowalnej na otwartej licencji

43 Jednym z fundatorów teorii literatury cyfrowej i nowych mediów w ogóle jest np. Jay David Bolter, twórca koncepcji remediacji i programista odpowiedzialny za powstanie Storyspace, programu do tworzenia powieści hipertekstowych. Przykładem interdyscyplinarnych inicjatyw badawczych są organizowane od 1989 roku przez ACM – Amerykańskie Stowarzyszenie Informatyków – konferencje Hypertext, SIGWEB i podobne, które w jednym miejscu skupiały i informatyków i humanistów.

44 Zob. S. Ramsay *Who's In and Who's Out*, panel „History and future of digital humanities”, MLA 2011, <http://stephenramsay.us/text/2011/01/08/whos-in-and-whos-out/> (20.10.2014),

45 J. Winters *How I Learned to Stop Worrying and Love the Digital*, wystąpienie w trakcie warsztatów „Digital Humanities at Oxford Summer School”, Oxford 2015, <http://dhoxxs.humanities.ox.ac.uk/2015/programme.html> (20.10.2014).

46 Mimo że książka dostępna jest za darmo w repozytoriach polskich uniwersytetów, to rynkowi daleko od jej uwolnienia. Zbiór refleksji Gołębiowskiego dostępny jest odpłatnie zarówno w tradycyjnych księgarniach, jak i w księgarniach e-booków.

GNU FDL⁴⁷. Jednak zarówno aspekt społeczny, jak i poetyka otwartości źródeł są tu inne niż w prezentowanych wcześniej przykładach. Książka Gołębiewskiego to uwolniona, ale wciąż zamknięta i skończona całość. Tomik Lipszycy opiera się na źródłach i eksponuje je – każdy z wierszy powstał na bazie dwóch stron Wikipedii – jednak konwencja edytowania remiksu jest nie do końca dowolna, choćby z tego powodu, że każda zmiana – według reguł sieciowej encyklopedii – musi zostać zaakceptowana przez autora strony, co jest warunkiem koniecznym publikacji. Pieczęć autorska, którą zdiagnozowałem jako jedną z przyczyn braku przeróbek minigier Jasona Nelsona, tutaj zamienia się w redaktorską, jeśli nie autorską, kontrolę⁴⁸. Nawet gdyby istniała ona tylko na poziomie systemowym, a w rzeczywistości Lipszyc nie ingerowałyby w żadną ze zmian dokonywanych przez czytelników, obecność tej formy kontroli wpisana jest w dzieło.

Utwory Carpentera, Nelsona i Monforta dzieli od polskich przykładów wspomagana algorytmicznie, wariantywna ekspresja⁴⁹. Utwory polskie funkcjonują jako przykład i zachęta do postaw spod znaku wolnej kultury. Te pierwsze są dodatkowo matrycami do wytwarzania tekstów pochodnych, własnych animowanych, graficznych i generatywnych wierszy każdego, kto zdecyduje się wziąć udział w proponowanej przez autora swobodnej, kreatywnej, literackiej zabawie⁵⁰.

47 J. Lipszyc *Mnemotechniki*, Wikizródła, 2008, <http://pl.wikisource.org/wiki/Mnemotechniki> (20.10.2014). Nazwisko Jarosława Lipszycy widnieje pod tytułem „Mnemotechniki”, w wersji online nie sygnalizuje się funkcji redaktora, jak w wydaniu drukowanym, gdzie autor przechodzi na pozycję redaktora. Zob. *Mnemotechniki*, red. J. Lipszyc, Wydawnictwo Krytyki Politycznej, Warszawa 2008.

48 W trakcie prac nad niniejszym artykułem, w marcu 2014 roku, dokonałem próbnej ingerencji w dwóch miejscach jednego z wierszy *Mnemotechnik* („Bezpieczeństwo publiczne”). Nie doczekała się ona autorskiej akceptacji do czerwca 2015 roku, czyli do chwili oddania niniejszego numeru „Tekstów Drugich” do druku.

49 Są to zatem cyberteksty, utwory dynamiczne, o „zmienniej ekspresji”, w których tekst jest wynikiem uprzedniej, wywoływanej przez użytkownika lub algorytm, kalkulacji. Zob. E. Aarseth *Cybertekst. Spojrzenia na literaturę ergodyczną*, przeł. M. Pisarski, P. Schreiber, D. Sikora, M. Tabaczyński, Korporacja Ha!art – Miejskie Centrum Kultury, Kraków–Bydgoszcz 2014, s. 14.

50 Swobodna gra, a zatem – paradoksalnie – wolność w obrębie wyznaczonych reguł, to jeden z podstawowych warunków kreatywności. Wszystkie z omawianych utworów, oprócz książki Gołębiewskiego, warunek taki wprowadzają. Zob. R. Achmituw *Embodied Algorithms: On Space and Mobility as Structural Metaphors*, w: *Remediating the Social*, s. 89.

Podsumowanie

Jeśli zgodzić się z diagnozą Lva Manovicha, według której dzisiejsza kultura definiowana jest przez software, który ją przekształca i reguluje, czyniąc to samo ze społeczeństwem (badacz określa je mianem *software society*)⁵¹, to dzieło literackie będące jednocześnie programem komputerowym przestaje być jedynie awangardową zabawą formalną. Staje się narzędziem do badania zarówno granic literatury, jak i do eksplorowania algorytmicznych mechanizmów, które coraz częściej, w sposób niewidoczny, próbują kształtować nasze postrzeganie świata i wzorce zachowań. Kto wie, czy instrumenty tekstowe Jasona Nelsona i generatory Nicka Montforta nie przygotowują swoich odbiorców do dużo sprawniejszej orientacji w terenie kulturowym i społecznym niż niejedna XIX-wieczna saga. Wbrew pozorom jest to dobra wiadomość dla literatury. Programy komputerowe omawianych autorów to pełnoprawne przykłady literatury XXI wieku nie dlatego, że do miana literatury aspirują, lecz z choćby z tej racji, że oprócz światów wewnątrztekstowych potrafią wytworzyć zjawiska literackie, mobilizować literacką publiczność, generować krytykę oraz – przede wszystkim – przesuwać granice między zastanymi kategoriami literackimi.

Uwalnianie kodu źródłowego przez programistę-literata staje się rodzajem wspólnego z odbiorcą czytania współczesnej kultury, a jednocześnie sztuką podnoszenia algorytmicznych, a zatem także kulturowych, kompetencji. Było to zadaniem komunikacji literackiej od dawna, dziś jednak statyczna, monokanałowa, związana z tradycyjnym medium druku literatura zdaje się wypełniać je coraz gorzej.

Literatura *open source* odzwierciedla przemiany kulturowe dokonujące się za sprawą, i w obrębie, tzw. zwrotu cyfrowego⁵², zwłaszcza te, które dotyczą ontologii dzieła literackiego, zmieniających się stylów odbioru oraz przekształceń i przemieszczeń (aczkolwiek nie przetasowań) na osi autor – odbiorca. Choć idea otwartych źródeł odnosi się przede wszystkim do programów komputerowych, omawiane tu utwory, w tym polski przykład *Mnemotechnik* Jarosława Lipszyca, udowadniają, że uwolnieniu i otwarciu na modyfikację poddaje się także tekst literacki. Paradoksalnie, i wbrew oczekiwaniom wywoływanym choćby przez sam termin „otwarte źródła”, literackie projekty spod tego znaku dowartościowują instancję autorską. Cele oprogramowania – weźmy jako

51 L. Manovich *Software Takes Command (International Texts in Critical Media Studies)*, Bloomsbery, London–New York 2013, s. 39.

52 *Zwrot cyfrowy w humanistyce*, red. A. Radomski, R. Bomba, Wydawnictwo UMCS, Lublin 2013.

przykład edytor tekstu Open Office, w którym powstaje niniejszy artykuł – są najczęściej czysto funkcjonalne. Sygnatura autorska zamazuje się w otwartym, kolaboratywnym procesie ich realizacji. Inaczej jest w przypadku otwartej literatury. Za nią wciąż stoi pojedynczy autor lub wyraźnie wyodrębniona grupa. Choć wycofuje się on za kulisy, pozostawia pole gry czytelnikowi, to wciąż wydaje się sprawować pieczę nad toczącym się – według reguł określonych przez niego – spektaklem *open source*. Mimo wszelkich cyfrowych przemian i zwrotów, widmo nieszczęsnego pytania „co autor miał na myśli?“, zdaje się wciąż krążyć nad eksperymentami spod znaku otwartych źródeł. Mam nadzieję, że udało mi się zjawisko to zaprezentować.

Abstract

Mariusz Pisarski

UNIVERSITY OF WARSAW

Open Source Literature: A Report on an Experiment

In this article Pisarski attempts to classify, analyse and provide an initial evaluation of the literary practice of open source – practices that have made possible by the code layer of digital poetry and prose. Made accessible in the digital era, this layer invites transformations; literary culture is opened up to competition; it raises authorial competence and facilitates collaborative creativity. Pisarski explores e-literary works that are “genuinely digital”. When the creator of a code opens up the computer code and encourages the creation of further works within a given framework, the reaction is not always as expected. Pisarski compares Jasona Nelsona’s digital poetry, Jim Carpenter’s Erica T. Carter project and Nick Montfort’s generative poem Taroko Gorge – one of which Pisarski considers a fully successful attempt; the digitalised reasons include references to the literary tradition of collective action, the choice of an appropriate poetics for the work (such as a narrative point of view), a clear indication of exchangeable elements and referential framework. Summing up, Pisarski sketches out the potential for open source literary practices and their future development.

Keywords

electronic literature, digital poetry, open source, collaborative literature, literary games, hypertext, Internet, Internet literature.