

Uniwersytet Komisji Edukacji Narodowej w Krakowie
Wydział Nauk Humanistycznych
Instytut Socjologii



Język kodu źródłowego programów komputerowych w ujęciu
hermeneutyki dyskursu Paula Ricœura i filozofii form
symbolicznych Ernsta Cassirera

Anna Bajer

Praca doktorska napisana pod kierunkiem
Profesor dr. hab. Katarzyny Gurczyńskiej-Sady

Kraków 2025

Spis treści

Wstęp	3
1. Zarys Problematyki	10
1.1. Pytanie o język - dwie perspektywy	10
1.2. Język w hermeneutyce dyskursu Paula Ricœura	15
1.3. Język jako forma symboliczna w filozofii Ernsta Cassirera	33
1.4. Spojrzenie na język Ricœura i Cassirera	48
1.5. Języki programowania w kontekście języków sztucznych	50
1.6. Perspektywy humanistycznych badań nad kodem źródłowym	65
2. Ustanowienie przestrzeni publicznej w języku	71
2.1. Wymiar ekspresji	73
2.2. Wymiar ujawniania	96
2.3. Wymiar konstytuowania	110
3. Artykulacja indywidualności	125
3.1. Wymiar ekspresji	127
3.2. Wymiar ujawniania	137
3.3. Wymiar konstytuowania	155
4. Rozpoznawanie standardów	174
4.1. Wymiar ekspresji	177
4.2. Wymiar ujawniania	192
4.3. Wymiar konstytuowania	206
Zakończenie	226
BIBLIOGRAFIA	235
Spis rysunków	265

Wstęp

Kod źródłowy programów komputerowych tworzony jest z zasady w celach użytkowych. W tym zakresie, może podlegać ocenie jakościowej, w sposób przyjęty w inżynierii jakości oprogramowania. Co jednak może zaskakiwać, pogłębionej analizie tego rodzaju tekstu można dokonać również z perspektywy humanistycznej. Celem niniejszej pracy jest zbadanie kodu źródłowego w oparciu o refleksję filozoficzną, a konkretnie – poprzez pryzmat dwóch podejść: hermeneutyki dyskursu Paula Ricœura oraz filozofii form symbolicznych Ernsta Cassirera.

Przyjęcie perspektywy hermeneutycznej pozwala postawić główną tezę pracy: język, którym posługują się programiści, umożliwia ujawnianie relacji jednostki z samą sobą, z innymi oraz ze światem. Staram się zatem uchwycić, w jaki sposób w kodzie źródłowym programów komputerowych można konstruować, wyrażać i odsłaniać treści istotne dla człowieka. Przyjęcie wspomnianych ujęć Ricœura i Cassirera umożliwia zaprezentowanie takiego podejścia.

Kod źródłowy to zapis instrukcji stworzony w języku programowania. Traktowanie go jako tekstu poddającego się interpretacji wywodzi się z obszaru badań znanego jako *Critical Code Studies* (CCS). Badania te stanowią jednak jedynie punkt wyjścia dla niniejszej dysertacji. Ujęcie problematyki języka z perspektywy filozoficznej pozwala spojrzeć na kod źródłowy w szerszym, kulturowym kontekście. W tym sensie jest próbą uporządkowania humanistycznego myślenia o programie komputerowym jako artefakcie kultury.

Ze względu na obranie hermeneutycznej perspektywy, język kodu źródłowego programów komputerowych zostaje tu potraktowany jako „działanie językowe”¹, a więc twórcza aktywność podejmowana przez programistów. Wychodząc poza jego deskryptywny wymiar zwracam więc uwagę na twórczy sposób, w jaki się nim posługują. Dyskurs Ricœura stanowi tu przestrzeń dokumentowania owego działania językowego, zaś Cassirerowska koncepcja form symbolicznych pozwala umieścić je w szerszym kontekście kulturowym. Celem pracy nie jest zatem porównanie obu filozofii,

¹ Ch. Taylor, *Teoria znaczenia*, [w:] *Język, dyskurs, społeczeństwo*, red. L. Rasiński, Państwowe Wydawnictwo Naukowe, Warszawa 2009, s. 154.

lecz wykorzystanie ich podejść do ukazania kodu źródłowego jako zjawiska kulturowego.

Ukazanie w sposób usystematyzowany różnych działań językowych programistów pozwala wyjść poza prezentowanie tychże jako zaledwie ciekawostek. Kod źródłowy przedstawiam jako artefakt kultury poddający się interpretacji, a język – nie jako narzędzie, ale jako przestrzeń twórczego bycia. Zarówno refleksja Ricœura, jak i Cassirera przynależy do nieanalitycznej tradycji filozofii języka. Odwołując się do rozróżnienia zaproponowanego przez Charlesa Taylora, zaliczam je do rodziny teorii konstytutywno-ekspresywnych (zw. HHH)².

Struktura pracy, inspirowana teorią znaczenia Taylora, opiera się na układzie problemowym. Składają się na nią cztery rozdziały, z czego pierwszy stanowi wprowadzenie teoretyczne. Pozostałe trzy odpowiadają trzem aspektom działania w języku: ustanawianiu przestrzeni publicznej, artykulacji indywidualności oraz rozpoznawaniu standardów. Każdy z nich podzielony jest na trzy podrozdziały, odpowiadające wymiarom: ekspresji, ujawniania oraz konstytuowania. Wprowadzenie takiego podziału za Taylorem, nadaje pracy przejrzystą strukturę. Stanowi ono zarazem odniesienie do budowy macierzy 3x3, nawiązując w ten symboliczny sposób do wykorzystywania macierzy w programowaniu.

Praca obejmuje cztery rozdziały. W rozdziale pierwszym przedstawiam podstawy teoretyczne pracy. W ramach filozofii języka prezentuję dwie dominujące perspektywy i za Taylorem sytuuję swoje rozważania w obszarze jednej z nich – rodzinie teorii konstytutywno-ekspresywnych. Na tej podstawie wyodrębniam trzy obszary tematyczne: przestrzeni publicznej, artykulacji indywidualności i standardów moralnych. W powyższych aspektach działania w języku rozpatrywane są, w kolejnych rozdziałach wymiar ekspresji, ujawniania i konstytuowania. Przybliżam również problem języka w hermeneutyce dyskursu Ricœura oraz jako formy symbolicznej w filozofii Cassirera. Zakończenie rozdziału stanowi omówienie języków programowania w kontekście języków sztucznych oraz przedstawienie perspektywy humanistycznych badań nad kodem źródłowym (*Critical Code Studies*).

²*Ibidem*, s. 154.

O działaniach językowych programistów traktują trzy kolejne rozdziały pracy. Rozdział II przybliży zagadnienie używania języka w kontekście przestrzeni publicznej. Na początku odnoszę się do kwestii więziotwórczego potencjału ekspresji językowej, porównując możliwości wyrażania się w kodzie źródłowym z uwzględnieniem różnych języków programowania. W dalszej części podrozdziału omawiam temat ekspresji artystycznej realizowanej w eksperymentalnej twórczości poetów posługujących się językami programowania, a także zagadnienie wartości estetycznych obecnych w tekście kodu. Choć ani Cassirer, ani Ricoeur nie zajmowali się bezpośrednio sztuką czy estetyką, obaj dostrzegali rolę ekspresji w procesach komunikacyjnych.

W kolejnej części rozdziału II podejmuję refleksję nad sposobami ujawniania się wspólnotowości w języku. Kod źródłowy okazuje się nośnikiem kwestii istotnych dla środowiska programistów, takich jak przeciwdziałanie dyskryminacji czy troska o zachowanie pamięci. Zestawiam zatem pojawiającą się w tym kontekście zmianę świadomości społecznej z równoczesną potrzebą utrzymania ciągłości historycznej. Analizy tę prowadzę wykorzystując „okrężną drogę” hermeneutycznego rozumienia, lecz same rozważania organizuję wokół Cassirerowskiej kategorii czasu.

Rozdział kończy się refleksją nad konstytuowaniem wspólnot na gruncie różnych języków programowania. Odnoszę się do możliwości tworzenia dyskursu w zależności od przyjętego paradygmatu programowania. Ta kwestia, w powiązaniu z problematyką tzw. fenomenów bazowych, pozwala ukazać kulturotwórczą rolę relacji międzyludzkich. Między innymi bowiem właśnie w oparciu o komunikację w językowej przestrzeni publicznej możliwe jest traktowanie kodu źródłowego jako artefaktu kulturowego.

Rozdział III obejmuje zagadnienia związane z indywidualnym aspektem działań programistów w języku. Tezą przyjętą w tej części pracy jest przekonanie, że artykulacja własnego sposobu widzenia świata jest możliwa również poprzez kod źródłowy. Można ją dostrzec w wyborze stylu programistycznego, używanego słownictwa, a także w samym akcie kreacji ezoterycznych języków programowania. Te właśnie aspekty stanowią przedmiot rozważań w części poświęconej ekspresji językowej. Przedstawiam w niej refleksje Cassirera dotyczące wyrażania afektu oraz Ricoëura na temat komunikowania doświadczenia.

W kolejnym podrozdziale podejmuję problem indywidualności w kontekście tego, jak rzeczy stają się znaczące dla programistów. W symbolach i komentarzach pojawiających się w kodzie dostrzegam przejawy artykulacji indywidualności oraz wyraz osobistych potrzeb jednostki. Konfrontując hermeneutykę dyskursu z myślą psychoanalityczną, ukazuję możliwość samoujawniania się programistów w tekstach, które tworzą.

Twórczość programistyczną, będącą wyrazem indywidualności, ujmuję w kolejnym podrozdziale jako element konstytuujący tożsamość. Rozdział III domyka więc analiza sposobów kształtowania się tożsamości programistów-artystów, programistów-rzemieślników oraz programistów-inżynierów. W duchu filozofii form symbolicznych zostają oni określani przez swoją aktywność, co koresponduje z podejściem prezentowanym przez Ricœura. Proces tworzenia tożsamości programistów przedstawiam zatem poprzez ich działania językowe, w ramach hermeneutyki dyskursu i hermeneutyki narracji.

W ostatnim, czwartym rozdziale uwaga skupia się na wymiarze etycznym tego, jak programiści posługują się językiem. Rozważana jest możliwość wyrażania wartości za pomocą kodu źródłowego. Na przykładzie ekspresji językowej hakerów podkreślam znaczenie świadomości standardów moralnych w środowisku programistycznym. Refleksja nad ich dążeniami etycznymi prowadzona jest w odniesieniu do mitu „kowboja klawiatury”, który odsłania system wartości charakterystyczny dla tej tożsamości. Analizując działalność hakerów, posługuję się trójskładnikową strukturą działań etycznych według Ricœura oraz Cassirerowską koncepcją ujarzmiania siły mitów.

Dalsza część rozdziału poświęcona jest namysłowi nad źródłem działania programistów jako podmiotów etycznych. Omawiam etyczne standardy, które wyrastają z wyznawanych przez nich wartości – troski o zdrowie, piękno i otwartość. W tym kontekście ruch *open source* rozpatrywany jest przez pryzmat fenomenologii człowieka działającego autorstwa Ricœura, natomiast kształtowanie się wokół tego ruchu subkultury – w świetle teorii form symbolicznych Cassirera.

Rozdział zamyka konfrontacja działań podejmowanych w języku przez programistów z aktywnością sztucznej inteligencji. Porównanie kompetencji językowych człowieka i AI uwypukla zdolność programistów do posługiwania się

językiem jako nośnikiem dyskursu. Tylko działania językowe podejmowane przez wolne, etyczne podmioty umożliwiają tworzenie wymiaru etycznego w języku kodu źródłowego. Ochrony przed zdominowaniem kultury przez formę nauki, czerpiąc z myśli Cassirera, warto szukać właśnie w obszarze refleksji etycznej.

Rozprawę kończy podsumowanie, w którym hermeneutyka dyskursu oraz filozofia form symbolicznych przedstawione są jako drogowskazy dla dalszych rozważań – nie tylko nad językiem, lecz także nad kulturą kodu źródłowego programów komputerowych. Kierując się tymi dwiema perspektywami, możliwe staje się satysfakcjonujące ujęcie kulturowego krajobrazu towarzyszącego językowym działaniom programistów. W obu ujęciach człowiek jawi się jako istota nastawiona na poznanie i osadzona w kulturze. Istotą działania językowego pozostaje sposób myślenia w języku: u Cassirera – ustrukturalizowany, u Ricœura – rozumiejący.

W zakończeniu powracam do dwóch przewodnich motywów niniejszej pracy: refleksyjności i działania. Przypominam również, w sposób skrótowy, kluczowe zagadnienia analizowane w rozprawie: ekspresję, ujawnianie oraz konstytuowanie w języku. W nawiązaniu do założeń metodologicznych – a przede wszystkim filozoficznych – uzasadniam decyzję o użyciu pierwszej osoby we wstępie i zakończeniu pracy. Całość domykam wskazaniem możliwych kierunków dalszych badań, zwłaszcza w kontekście wyzwań, jakie niesie ze sobą rozwój sztucznej inteligencji.

Obu perspektyw filozoficznych nie da się uznać za podobne czy komplementarne – okazały się wszelako przydatne, każda w odmiennym zakresie. Niniejsza praca nie rości sobie pretensji do wyczerpującego omówienia wszystkich zagadnień związanych z językową działalnością programistów w obszarze kodu źródłowego. Celem było raczej możliwie pełne ukazanie jej w trzech kluczowych wymiarach: przestrzeni publicznej, sferze indywidualnej oraz w kontekście standardów etycznych. Do każdego z nich odwołują się zarówno Cassirer – poprzez swoją filozofię form symbolicznych – jak i Ricœur – w ramach hermeneutyki dyskursu.

Na zakończenie warto poczynić kilka uwag natury technicznej. Wymaga wyjaśnienia kwestia sformułowania użytego w tytule pracy: „język kodu źródłowego programów komputerowych”, a nie – „język w kodzie źródłowym”. Takie ujęcie wydaje się trafniejsze i pozwala szerzej ukazać problem. Ograniczenie się do języka w

kodzie źródłowym mogłoby przywoływać skojarzenia z obszarem językoznawstwa, a nawet prowadzić do mylnego wrażenia, że przedmiotem rozważań pozostaje jedynie język, który pojawia się w treści instrukcji programów komputerowych. Tymczasem język jest tu rozumiany znacznie szerzej – jako dyskurs oraz forma symboliczna. Analizie poddano nie tylko kod źródłowy wraz z komentarzami, lecz także język, którym programiści posługują się opisując własne działania.

Przyjęte w pracy cezurę dotyczące twórczości Ricœura mają szeroki zakres – obejmują różne etapy jego działalności filozoficznej. Zwornikiem tych odniesień pozostaje pojęcie dyskursu. Uwagę czytelnika może zwrócić obecność terminologii zaczerpniętej z informatyki oraz odniesienia do fragmentów kodu źródłowego wybranych programów komputerowych. Dla zachowania przejrzystości oraz ułatwienia śledzenia wywodu, specjalistyczne terminy zostały objaśnione w przypisach dolnych, natomiast przywoływane w treści fragmenty kodu zamieszczone są w formie ilustracji opatrzonych podpisami.

Wśród źródeł istotnych z perspektywy tej pracy znajdują się oryginalne fragmenty kodu pochodzące z repozytoriów, podręczników, blogów i forów dyskusyjnych. Niekiedy odwołuję się również do towarzyszących im komentarzy programistów – będących wyrazem indywidualnych opinii bądź głosem części środowiska. W trakcie zbierania materiału oraz opracowywania zagadnień napotkałam trudności wynikające z braku jednoznacznych definicji niektórych pojęć wśród samych programistów i powoływania się na „fakty powszechnie znane”.

Choć tematyka niniejszej pracy mogłaby pozornie wpisywać się przede wszystkim w obszar filozofii informatyki, to ze względu na przyjęcie perspektywy hermeneutyki dyskursu Ricœura oraz analizę form symbolicznych w ujęciu Cassirera, trafniejsze wydaje się zaklasyfikowanie jej do nurtu kontynentalnej filozofii języka. Rozważania prowadzone są metodą hermeneutyczną, z uwzględnieniem języka jako formy symbolicznej. Warto podkreślić, że przyjęte ujęcie ma wyraźnie interdyscyplinarny charakter i wykracza poza ramy samej filozofii czy informatyki. Inspiracje czerpane są również z psychologii, socjologii, językoznawstwa oraz kognitywistyki.

Dysertacja ta może okazać się interesująca także z perspektywy badań nad antropologią współczesności, ukazuje bowiem fragmentaryczny, ale znaczący obraz

kultury kształtowanej przez społeczność programistów. W tym sensie może stanowić punkt wyjścia do dalszych badań nad językiem kodu źródłowego. Ujęcie problematyki w ramach rzadko przywoływanej dziś koncepcji dyskursywności Ricœura oraz wykorzystanie mniej popularnej współcześnie filozofii form symbolicznych Cassirera przynosi satysfakcję badawczą. Poza motywacją wewnętrzną i popularyzatorską kieruje mną również przekonanie, że praca ta może mieć pozytywny wpływ na rozwój filozofii informatyki. Nie bez znaczenia pozostaje także ukazanie w niniejszej rozprawie możliwości podejmowania zagadnień z obszaru nauk ścisłych z perspektywy kontynentalnej filozofii języka.

W tym miejscu chciałabym bardzo podziękować mojej promotorce Profesor Katarzynie Gurczyńskiej-Sady za danie mi przestrzeni do realizacji moich pomysłów oraz okazaną pomoc w trakcie pisania pracy doktorskiej. Dziękuję również moim bliskim za wyrozumiałość i wsparcie.

1. Zarys Problematyki

1.1. Pytanie o język - dwie perspektywy

Filozoficzny namysł nad językiem ma bardzo bogatą tradycję. Kwestię języka ujmowano jednak na tak różne sposoby, że trudnym było klasyfikować rozważania mu poświęcone jako po prostu - filozofię języka. Ponadto, rozmyślenia te nierzadko stanowiły zaledwie marginalny wątek dorobku filozofów. Zauważalna stała się natomiast z czasem pewna tematyzacja pojawiających się treści. Dostrzec da się rozróżnienie na problematykę skupiając się wokół tego, jak za pomocą języka można ujmować rzeczywistość i zagadnienia jak człowiek myśli w języku³. Przemyślenia nad poznaniem rzeczywistości dokonujące się dzięki porządkującej funkcji języka wiążą się z rozważaniami jednego nurtu, natomiast refleksja nad swoistym związkiem człowieka z językiem, drugiego. Te dwie tendencje myślenia o języku wyodrębniły się jednak stosunkowo niedawno: instrumentalne podejście do języka w XVII wieku, zaś antropologiczne od XIX wieku⁴. Kolejno odpowiadają podstawom, z których wyewoluowały analityczna i kontynentalna filozofia języka.

Wyodrębnienie się filozofii analitycznej nastąpiło w sposób bardziej czytelny niż filozofii kontynentalnej. XX-wieczny, wyrosły na krytyce metafizyki, ruch analitycznych filozofów języka dawał odpór tradycyjnej filozofii⁵ i to z takim rozmachem, że zdominował na kolejne dziesięciolecia sposób w jaki filozofowano o języku⁶. Analitycy dążyli przede wszystkim do uzyskania przejrzystości rozumowań i aby sprostać temu zamiarowi ujmowali język, jako narzędzie oraz przedmiot analizy⁷. To, by tak rzec, „rachunkowe” podejście do niego miało gwarantować ściśle wyrażanie myśli. Powstanie analitycznej filozofii języka, która swoje centra intelektualne przeniosła⁸ poza Europę kontynentalną, pociągnęło zaś za sobą wyodrębnienie się

3 P. Prechtel, *Wprowadzenie do filozofii języka*, Wydawnictwo WAM, Kraków 2007, s. 9.

4 *Ibidem*, s. 9.

5 *Ibidem*, s. 69.

6 Trudno wytłumaczyć dość często spotykane, a mylące uproszczenie polegające na, używaniu nazwy „filozofia języka” dla określenia filozofii analitycznej języka. Jest to tym bardziej osobliwe, gdy wziąć pod uwagę, że analiza filozoficzna jako sposób filozofowania ma mieć na uwadze klarowność.

7 J. Odrowąż-Sypniewska, *Przewodnik po filozofii języka*, Wydawnictwo WAM, Kraków 2016, s. 8.

8 Działalność wczesnego Wittgensteina, Koła Wiedeńskiego i szkoły lwowsko-warszawskiej miała miejsce na kontynencie, ale dalszy rozwój filozofii o analitycznym podejściu, już poza nim. Miało to związek z przymusem ucieczki z hitlerowskiej Europy wielu przedstawicieli filozofii analitycznej. Za A. Pawelec, *Kontynentalna filozofia języka: wybrane stanowiska*, [w:] *Współczesna filozofia języka. Inspiracje i kierunki rozwoju*, red. P. Stalmaszczyk, Primum Verbum, Łódź 2013, s. 15.

nazwy dla kontynentalnej filozofii języka. Bliskość geograficzna i czasowa oraz koncentracja refleksji wokół języka byłyby jednak zbyt słabym powiązaniem, aby móc ująć pod jednym szyldem tak różnych reprezentantów myśli europejskiej. Do jej przedstawicieli zalicza się przecież między innymi strukturalistów, hermeneutów, fenomenologów czy też postmodernistów, wszystkich tych, którzy problematyzują kwestie związane z językiem.

W uproszczonym porównaniu, analityczny sposób filozofowania przeciwstawia się myśli kontynentalnej, czymkolwiek miałyby ona być. Analitycy zajmować się mieli poprawnym użyciem języka, aby adekwatnie opisać rzeczywistość⁹, toteż poruszali się w swoich rozważaniach niejako w szerz problemu¹⁰. Kontynentalni filozofowie języka kierowali swoją uwagę na rozumienie jakie się dzięki niemu uzgadnia¹¹. Dążenie do osiągnięcia przejrzystości przez pierwszych bywało kontrastowane z zarzucaną drugim mętnością wywodu, jak chociażby we fraszce Tadeusza Kotarbińskiego:

„Do jasnych dążąc głębin – nie mógł trafić w sedno
Śledź pewien, obdarzony natura wybredną.
Dokądkolwiek wędrował, zawsze nadaremno:
Tu jasno, ale płytko – tam głębia, lecz ciemno.”¹²

W wierszu, ironicznie, refleksję nurtu analitycznego nazywa się niepełną, acz jasną, zaś przedmiot badań kontynentalnych filozofów języka uznaje się za niezgłębialny, a i być może namysł nad nim za prowadzony w sposób niejasny¹³. Twierdzenie, że powstały dwie rozłączne drogi¹⁴ byłoby jednak zbyt prostym uproszczeniem.

Nie zagłębiając się dalej w rzeczywiste, czy też fikcyjne animozje, trzeba zaznaczyć, że istnieją uzasadnione wątpliwości wobec, tego czy da się zgrupować i

9 *Ibidem*, s. 17.

10 Tłum. własne słowa *lateral*, wydające się lepiej wyjaśniać istotę niż dosłowne tłumaczenie „boczny”. Andrzej Pawelec w tekście *Kontynentalna filozofia języka...* tłumaczy słowo jako *wymiar poziomy*. D. Morris, *Philosophy of Mind*, [w:] *The Edinburgh Companion to Twentieth-Century Philosophies*, red. C.V. Boundas, Edinburgh University Press, Edinburgh 2007, s. 531.

11 *Vertical difficulties*, *Ibidem*, s. 531.

12 A. Nowaczyk, *Polawianie sensu w filozoficznej głębi*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2006, s. 5.

13 Taki zarzut mógłby paść na przykład wobec stwierdzenia, że poznanie odbywa się poprzez „rozczłonkowanie rzeczywistości w języku”. E. Kowalska, *Wprowadzenie*, [w:] *Rozmaitość języków a rozwój umysłowy ludzkości*, W. Humboldt, przekł. i wstęp E. Kowalska, KUL, Lublin 2001, s. 41.

14 Obecnie pojawiają się próby nawiązywania przez filozofów postanalitycznych do myśli filozofów kontynentalnych np. przez Donalda Davidsona, Roberta Brandoma czy Johna McDowella, za *Postanalytic and Metacontinental Crossing Philosophical Divides*, red. J. Reynolds, J. Chase, J. Williams, E. Mares, Wydawnictwo Continuum, New York 2010.

według jakich kryteriów, kontynentalnych filozofów języka. Bez względu, czy przyjmiemy za zasadne odgraniczenie przez definicję negatywną, czy może za Simonem Glendinningiem¹⁵ uznamy, że idea wyodrębnienia jest myląca i można co najwyżej stworzyć konturową mapę uwzględniając zainteresowania kontynentalistów¹⁶, to jednak dostrzec da się bliskość duchową i intelektualną pewnych ich refleksji. Jest to o tyle ważne, że rozważania Paula Ricœura przyjmujące język za horyzont ontologii hermeneutycznej, sytuują je w ramach tego, co zwykle się nazywać tradycją kontynentalną. Również przedstawienia języka jako jednej z form symbolicznych w ujęciu Ernsta Cassirera nie zaliczymy do ówczesnego nurtu analitycznego. Koncepcje tych filozofów daje się lepiej sklasyfikować dzieląc filozofię języka na hermeneutyczną i transcendentálną^{17,18}.

Przywołano już powody, dla których wyodrębnianie filozofii kontynentalnej w sposób dotychczas czyniony, okazuje się daremne. Wspomniane trudności przezwyciężył Charles Taylor w swojej filozoficznej teorii języka dzieląc powstałe dotychczas koncepcje na: instrumentalne i ekspresyjne. Za punkt wyjścia nie obrał tzw. zwrotu lingwistycznego¹⁹ Rortyego, czyniącego z filozofii kontynentalnej niezidentyfikowaną resztę pozostałą po wyodrębnieniu nurtu analitycznego i pozbawiając ją tym samym statusu samodzielności. Ujęcie Taylora zapobiega odcięciu kontynentalistów od ich wspólnych korzeni – zainteresowania aktywnym charakterem języka, *energeia*²⁰. Co szczególnie istotne z punktu widzenia tej rozprawy, taki podział umożliwia zgrupowania refleksji filozofów języka na kontynencie, pomimo zróżnicowania ich przemyśleń. W podobnym do Taylora koncyliacyjnym tonie rozważania prowadził nad tym tematem Jürgen Habermas przedstawiając jako komplementarne dwie odmiany zwrotu lingwistycznego: analityczną i hermeneutyczną²¹. Biorąc jednak pod uwagę zakotwiczenie jego rozważań w teorii

15 S. Glendinning, *The Idea of Continental Philosophy*, Edinburgh University Press, Edinburgh 2006, s. 2.

16 *Ibidem*, s. 20, s. 66.

17 A. Bronk, *Rozumienie, dzieje, język. Filozoficzna hermeneutyczna H.-G. Gadamera*, Redakcja Wydawnictw KUL, Lublin 1988, s. 284, s. 283-288.

18 T. Nowak, *Język w świetle odkryć nauki*, Petrus, Kraków 2011, s. 54.

19 R. Rorty, *Introduction*, [w:] *The linguistic turn: recent essays in philosophical method*, The University of Chicago Press, Chicago; London 1968, s. 1-39.

20 Ch. Taylor, *Teoria znaczenia*, [w:] *Język, dyskurs, społeczeństwo*, red. L. Rasiński, Państwowe Wydawnictwo Naukowe, Warszawa 2009, s. 154.

21 J. Habermas, *Filozofia analityczna i hermeneutyczna – dwie komplementarne odmiany zwrotu lingwistycznego*, [w:] *Język, dyskurs, społeczeństwo*, red. L. Rasiński, Państwowe Wydawnictwo Naukowe, Warszawa 2009, s. 244.

działania komunikacyjnego, trudno byłoby dla filozofii form symbolicznych wskazać miejsce w którejś z tych tradycji.

W swojej propozycji Taylor zdefiniował w sposób pozytywny, nie zaś przez negację, dwie rodziny teorii. W pierwszej, zwanej deskrypcyjno-instrumentalną²² lub deskrypcyjną (inaczej HLC)²³, koncentruje się uwagę na reprezentowaniu za pomocą języka niezależnej rzeczywistości. W ramach tego zespołu koncepcji, w których traktuje się język jako dane narzędzie, znalazłaby się zatem filozofia analityczna. W drugiej rodzinie teorii, nazwanej przez Taylora konstytutywno-ekspresywną (inaczej konstytuujące podejście do języka, rodzina teorii HHH)²⁴, wychodzi się poza funkcje opisywania – eksponuje się twórczy aspekt języka. W języku ludzie formułują myśli, w taki sposób, że je sobie uświadamiają²⁵. Poprzez swoją artykulację konstytuuje on zatem ludzką świadomość, zarazem kształtuje też życie społeczne²⁶. Ten właśnie ukryty w nim potencjał sprawia, że człowiek wchodzi w relacje ze światem i może troszczyć się o siebie i innych. Jako istota językowa ma możliwość tworzyć i stosować się do standardów zachowania, a to właśnie pozwala na uznanie człowieka za podmiot moralny.

Do typu teorii HHH, zwanych przez Taylora „romantycznymi”, zaliczyć można to, co przyjęło się nazywać kontynentalną refleksją nad językiem, gdy język traktowany jest jako „locus różnego rodzaju ujawnień”²⁷, a zarówno Ricœur jak i Cassirer takie właśnie prezentują podejście. Praca ta dotyczy tego, co dzieje się w języku i jaka jest natura tej aktywności. To zaś pozwala na wykorzystanie zaczerpniętego od Taylora podziału działania językowego według aspektu: artykulacji, ustanawiania przestrzeni publicznej i rozpoznawania standardów²⁸. W ramach tych funkcji można dokonać dalszych użytecznych rozróżnień: wyrażania w języku (wymiar ekspresji), ujawniania

22 Tłum. własne słowa „designative-instrumental”, Ch. Taylor, *The Language Animal. The Full Shape of the Human Linguistic Capacity*, The Belknap Press Cambridge, London 2016, s. 4.

23 Trzej myśliciele Hobbes, Lock i Condillac stworzyli zręby tej tradycji myślenia, stąd też inną proponowaną przez Taylora nazwą dla tego typu teorii jest akronim HLC.

24 Teorie konstytutywno-ekspresywnym powstał ze zrębów refleksji tworzonej przez myślicieli: Johann Georg Hamanna, Johann Gottfried Herdera i Wilhelm von Humboldta (stąd skrótowiec HHH) i wiązały się z wzrostem zainteresowania tematyką okołojęzykową w drugiej połowie XVIII wieku.

25 Ch. Taylor, *Teoria znaczenia...*, s. 155.

26 *Ibidem*, s. 167.

27 *Ibidem*, s. 169.

28 *Ibidem*, s. 154.

w języku (wymiar artykulacji fundamentalnej) oraz stwarzania w języku (wymiar konstytuowania)²⁹. Podziały te zostaną wykorzystane w niniejszej pracy.

Konstytuujące ujęcie języka daje możliwości bycia „duchową orientacją”³⁰, nie redukując wyjaśnienia języka do kwestii kategorii funkcjonalnych, tak jak robi się to w teoriach instrumentalnych (HLC)³¹. W teoriach konstytutywnych wprowadzanie przez innych ludzi do świata znaczeń - „wychowanie do trwającej wciąż rozmowy”³² - umożliwia podmiotowi poruszanie się we wspólnej przestrzeni moralnej i duchowej³³. Język, jak to przedstawia Taylor, staje się tym samym źródłem człowieczeństwa, natomiast w teoriach typu instrumentalnego służy procesom poznawczym³⁴.

Od tekstów kodu źródłowego programów komputerowych wymaga się jednoznaczności, więc można by je uznać za wyraz spełnienia oczekiwań wobec języka z perspektywy deskrypcyjno-instrumentalnego podejścia. Ponieważ w ramach twórczości programistycznej pojawia się jednak wiele elementów burzących wyobrażenie o „czystości przedstawienia” - aprioryczności języków programowania, w pracy tej wybrano konstytutywną koncepcję języka dla opisanego technicyzowanego fragmentu rzeczywistości do którego przynależą. To, jak w języku konstytuuje się potrzeba bycia z sobą, bycia z innymi i jak język stanowi medium wyrażania standardów moralnych - przybliżą nam dwa ujęcia: Ricœura i Cassirera. Kwestie te zostaną to zbadane w kolejnych rozdziałach poprzez prześledzenie wymiaru ekspresji, ujawniania i konstytuowania w wybranych tekstach kodu źródłowego programów komputerowych.

29 *Ibidem*, s. 161-169.

30 Ch. Taylor, *Źródła podmiotowości. Narodziny tożsamości nowoczesnej*, Państwowe Wydawnictwo Naukowe, Warszawa 2001, s. 69.

31 A. Pawelec, *Romantyczna teoria języka i metafory*, [w:] *Charlesa Taylora wizja nowoczesności. Rekonstrukcje i interpretacje*, red. Ch. Grabowski, J. Hudzik, J. Kłos, Oficyna Wydawnicza Łośgraf, Warszawa 2012, s. 263.

32 Ch. Taylor, *Źródła podmiotowości. Narodziny tożsamości nowoczesnej*, Państwowe Wydawnictwo Naukowe, Warszawa 2001, s. 69.

33 *Ibidem*, s. 68.

34 A. Pawelec, *Romantyczna teoria ...*, s. 262-163.

1.2. Język w hermeneutyce dyskursu Paula Ricœura

Zagadnienia dotyczące filozofii języka stanowią integralną część stworzonej przez Ricœura hermeneutyki filozoficznej i nie sposób sztucznie oddzielić ich od teorii rozumienia jego autorstwa. Na dodatek wraz z ewolucją myśli filozofa zmieniała się też jego koncepcja języka. Rozważając ją trzeba tym samym czerpać z całego dorobku tego myśliciela. Rozważania nad kwestią języka poprzez wpisanie ich w mgliście zarysowaną teorię, na przykład, hermeneutyki humanistycznej³⁵ nie miałyby zbyt dużych szans powodzenia. Wyjaśnianie w oparciu o niejasny koncept jest procedurą mało użyteczną. Pomocne staje się w związku z tym wyszczególnienie kolejnych okresów w twórczości Ricœura.

Jean Grondin podzielił twórczość Ricœura na sześć okresów³⁶. W dużej mierze zaproponował ich nazwy na podstawie głównych tematów zawartych w kolejnych pozycjach wydawniczych. Wyszczególnił kolejno: „hermeneutykę symboli”³⁷, „hermeneutykę zaufania”³⁸, „hermeneutykę innowacji semantycznej”³⁹, „hermeneutykę rozumienia i wyjaśniania”⁴⁰, „hermeneutykę czasu i narracji”⁴¹ oraz „hermeneutykę siebie”⁴². Mniej dokładnie niż Grondin, za to jak się zdaje bardziej dogłębnie, twórczość Ricœura ujęła Alison Scott-Baumann. Badaczka podzieliła jego działalność filozoficzną na trzy okresy, dzięki czemu przejrzyste staje się to, jak zmieniała się koncepcja rozumienia u tego filozofa. Kolejno mamy zatem: hermeneutykę I zwana „hermeneutyką podejrzeń” skoncentrowaną na interpretacji mitów i symboli, następnie

35 Przyporządkowanie Alwina Diemera Za A. Bronk, *Rozumienie, dzieje, język*, Lublin 1988, s. 33.

36 J. Grondin, *Do Gadamer and Ricœur have the same understanding of hermeneutics?*, [w:] M. Xie, *The Agon of Interpretations: Towards a Critical Intercultural Hermeneutics*, Toronto University Press, Toronto 2014, s. 46.

37 obejmującej: „Philosophie de la volonté 2 Finitude et culpabilité”, „La symbolique du mal” z 1960 roku

38 obejmującej: „De l’interprétation, essai sur Freud” z 1966 roku, „Le conflit des interprétations. Essais d’herméneutique” z 1969 roku.

39 obejmującej „La métaphore vive” z 1975 roku.

40 obejmującej „Du texte à l’action” z 1986 roku.

41 obejmującej „Temps et récit. Tome I”, „Temps et récit. Tome II. La configuration dans le récit de la fiction”, „Temps et récit. Tome III. Le temps raconté”, wydane od 1983-85 roku.

42 obejmującej „Soi-même comme un autre” z 1990 roku. J. Grondin hermeneutykę siebie nazywa też *Capable self philosophy*, J. Grondin, *Do Gadamer and Ricœur have the same understanding of hermeneutics?*, [w:] M. Xie, *The Agon of Interpretations: Towards a Critical Intercultural Hermeneutics*, Toronto University Press, Toronto 2014, s. 46.

hermeneutykę II – „teorię interpretacji”. Ostatni okres poświęcony jest „hermeneutyce odkrycia siebie”⁴³.

W pracy tej nacisk będzie położony na rozumienie języka jako dyskursu. Podejście to rozwijane od powstania „Le conflict des interpretations”, czyli kształtujące się już od około 1969 roku, Ricœur przedstawił w pełni w zbiorze artykułów pod tytułem „Du texte à l’action” z 1986 roku. Sytuuje to problematykę poruszaną w tej pracy w środkowym okresie jego twórczości. Ze względu na zogniskowanie uwagi filozofa w latach 70-tych XX wieku na języku jako dyskursie, sensowne jest wykorzystanie na oznaczenie tego okresu twórczości sformułowania „hermeneutyka dyskursu”. Termin utworzony został na podstawie poglądów francuskiego filozofa zaprezentowanych w eseju pt. „Język jako dyskurs”. Ricœur nazywał tamże swój ówczesny punkt widzenia filozofią dyskursu⁴⁴ uwolnioną od psychologizowania i nastawienia egzystencjalnego. Miała ona wyrastać ze zrozumienia dialektyki zdarzenia i znaczenia w dyskursie. Filozof odciął się od hermeneutyki psychologizującej, ale również jej przeciwności w postaci strukturalnej analizy zdaniowej zawartości tekstów - ze względu na ich niedialektyczną jednostronność⁴⁵. Określenia „hermeneutyka dyskursu” używali Damian Małachowski i Michał Otrócki w artykule „Hermeneutyka dyskursu według Paula Ricœura”⁴⁶ i William D. Melaney w artykule „Ricœur’s transcendental concern: a hermeneutics of discourse”⁴⁷. Chociaż „hermeneutyka rozumienia i wyjaśniania”, „hermeneutyka tekstu”⁴⁸ czy „hermeneutyka jako teoria interpretacji” są nazwami równie adekwatnymi, a na dodatek bardziej rozpoznawalnymi, to jednak z uwagi na swoją szeroką perspektywę nie trafiającymi tak celnie w źródło interesującego nas zagadnienia. Termin „hermeneutyka dyskursu” podkreśla natomiast dokładnie to, na czym ogniskować będzie się nasza uwaga – na dyskursie.

43 Tłum. własne *Recovery self*, z tekstu książki A. Scott-Baumann, *Ricœur and the Hermeneutics of Suspicion*, Continuum, London 2009.

44 P. Ricœur, *Język jako dyskurs*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 94.

45 *Ibidem*, s. 94.

46 D. Michałowski, M. Otrócki, *Hermeneutyka dyskursu według Paula Ricœura*, „Kwartalnik Filozoficzny”, 41, nr 4, 2013.

47 W. D. Melaney, *Ricœur’s transcendental concern: a hermeneutics of discourse*, [w:] *The Fullness of the Logos in the Key of Life: Book II. Christo-Logos: Metaphysical Rhapsodies of Faith. Itinerarium mentis in deo. Analecta Husserliana 111*, red. Anna-Teresa Tymieniecka, Springer, Hanover 2011.

48 M. Drwięga, *Paul Ricœur daje do myślenia*, Wydawnictwo Homini, Bydgoszcz, 1998, s. 54.

W okresie poprzedzającym hermeneutykę dyskursu Ricœur za podstawę swojej hermeneutyki przyjął rozumienie poprzez symbole. Proces interpretacji zaczyna się od naiwnej wiary w przekaz symboliczny, po niej następuje analiza języka symbolicznego, aby na końcu w interpretacji hermeneutycznej mógł nastąpić powrót do wiary⁴⁹. Poprzez symbole zatem ma się realizować ludzka relacja z *sacrum*. Z uwagi na „unipamiętnienie *sacrum*”⁵⁰ nieosiągalna stała się bowiem według filozofa bezpośredniość wiary, czego konsekwencją jest to, że można uwierzyć jedynie przez interpretację. Hierofania⁵¹ dokonuje się podczas „transcendentalnej dedukcji symbolu”⁵², czyli w momencie rozjaśniania nam doświadczenia przez symbol. By nawiązać łączność z utraconą sferą potrzebne jest w tym ujęciu zaaplikowanie „lekarstwa na nędzę nowoczesności”⁵³, czyli hermeneutyka symbolu. Należy zaznaczyć jednak, że intencją filozofa nie było zamknięcie rozumienia w samym symbolu, gdyż ten zaledwie „daje do myślenia”. Tak zapoczątkowana refleksja umożliwia dopiero „wynoszenie sensu na wyższy poziom”⁵⁴. Formowanie sensu przez twórczą interpretację wpływać miało bowiem na zmianę naszej świadomości refleksyjnej. Poprawienie położenia w bycie nie następowało zatem według Ricœur’a przez samą refleksję nad sobą, lecz wymagało poprowadzenia jej przez przemawiające do nas symbole⁵⁵.

Zaskakujące, że przy całej mocy, jaką posiadał symbol w hermeneutyce symbolu Ricœur’a był on ujmowany przez niego jedynie jako struktura, „którą cechuje podwójne znaczenie”^{56,57}. Wtedy jeszcze bowiem filozof ograniczał bowiem wtedy jeszcze jego zasięg do semantycznego aspektu języka⁵⁸. Przełom lat 60-tych i 70-tych XX wieku przyniósł w tej kwestii zmiany, gdyż Ricœur odszedł od rozumienia symbolu jako struktury jedynie semantycznej i uwzględnił jego dynamiczny aspekt. Istotnie

49 D. Michałowski, *Paul Ricœur’a hermeneutyka symboli. Próba prezentacji*, „Kwartalnik Filozoficzny”, 40 nr 3, 2012, s. 113.

50 P. Ricœur’a, *Symbolika zła*, PAX, Warszawa 1986, s. 332.

51 Objawienie się Świętości w więzi człowieka z *sacrum*.

52 P. Ricœur’a, *Symbolika zła...*, s. 335.

53 *Ibidem*.

54 *Ibidem*.

55 *Ibidem*, s. 336.

56 Ricœur przyznaje, że takie rozumienie symbolu towarzyszyło mu podczas pisania „Symboliki zła” jak i „O interpretacji: Esej o Freudzie”, P. Ricœur, *Teoria interpretacji...*, s. 124.

57 *Koncepcji hermeneutyki, która była (...)związana z symbolem, pojętym jako wyrażenie o podwójnym sensie*, [w:] P. Ricœur, *Refleksja dokonana. Autobiografia intelektualna*, Wydawnictwo Antyk, Kęty 2005, s. 32.

58 Poza to ujęcie nie wychodził jeszcze pisząc „De l’interprétation: Essai sur Freud”.

zmodyfikował kwestie podjęte w „Symbolice zła” gdy zaczął rozpatrywać również związek symbolu z doświadczeniem niebędącym językowym. Przedwerbalna aktywność splata się bowiem z tym co symboliczne. W symbolu płaszczyzna semantyczna odsyłać miała do presemantycznej, w której to tkwić ma głębia ludzkiego doświadczenia.

Symbol w nowej hermeneutyce Ricœura, „wahał się”, więc „na linii dzielącej *bios* i *logos*”⁵⁹. W ujęciu psychoanalitycznym wyraża się to w przejściu od pragnienia do formy kultury. Podobnie poniżej poziomu języka sytuuje się pierwiastek światłości jako siły *sacrum*, czy też element duchowy w świecie poetyckim. Są one sprawcze, a jednocześnie gwałtowne i nieuchwytnie. Prewerbalna siła łączy się więc ze znaczeniem tak, by mógł pojawić się dyskurs. Przedjęzykowe doświadczenie, na przykład, impulsy i ich językowe przedstawienia są zatem rozumiane dopiero w interpretacji. Wiąże się to z istotną kwestią niemożliwości odzwierciedlania rzeczywistości. Język nie wszystko jest bowiem w stanie wyrazić, gdyż „chwytą samą powierzchnię życia”⁶⁰. Dopiero w dialektyce z tym, co wobec niego pierwotne, powstaje przestrzeń dyskursu. W ten sposób, wychodząc od nowej teorii symbolu, Ricœur zapoczątkował swoją kolejną hermeneutykę. Zapośredniczył ją w przekazie, czyli szczególnym aspekcie języka, referencyjnym wobec świata. Tym sposobem Ricœur odszedł od upatrywania istoty rozumienia w symbolu i zapoczątkował hermeneutykę dyskursu.

Przekraczanie granicy między siłą *biosu* i formą *logosu* świadczy o „zakorzenieniu Dyskursu w Życiu”⁶¹. Niezbędne jest bowiem wprawdzie to co pozajęzykowe, czyli nasze doświadczenie, aby wyrażanie werbalnie stało się możliwe. Dociekania francuskiego filozofa dotyczące wkraczania symboli do języka, w kontekście świata poetyckiego jako „przestrzeni równie hipotetycznej jak porządek matematyczny w stosunku do dowolnego danego świata”⁶², mogą stanowić podstawę do snucia analogicznych rozważań nad światem programowania. Program komputerowy, podobnie jak utwór muzyczny, którego przykład przywołuje Ricœur, ma również wewnętrzny porządek. Moc twórcza programisty musi splatać się z formą, którą ten nadaje tekstowi kodu źródłowego. Dyskurs bowiem „rodzi się z koincydencji siły i formy”⁶³. W akcie kreacji

59 P. Ricœur, *Teoria interpretacji...*, s. 141.

60 *Ibidem*, s. 147.

61 *Ibidem*, s. 141.

62 *Ibidem*, s. 142.

63 *Ibidem*, s. 141.

twórca wnosi do języka nowe konfiguracje i wyraża rzeczywistość. Jest to jego sposób bycia w świecie. W tak przedstawionej hermeneutyce można dostrzec jej wymiar bytowy nawiązujący do Heideggerowskiej hermeneutyki faktyczności.

Rozwijając problem języka jako dyskursu, Ricœur wyszedł od krytyki strukturalistycznego podejścia do przekazu. Według francuskiego filozofa Ferdinand de Saussure skontrolował, z gruntu niepotrzebnie, *parole* z *langue* gdy przeciwstawiał sobie ich własności⁶⁴. W paradygmacie strukturalistycznym badania koncentrowały się na kodzie-języku, z powodu jego anonimowego i usystematyzowanego charakteru. Wynikał on ze społecznej natury schematów. Przekaz natomiast jako indywidualny i podlegający przypadkowości nie miał szans na uzyskanie takiego statusu. Patrząc w szerszej perspektywie, skutkowało to jednak marginalizacją dyskursu. Podstawą modelu strukturalistycznego Saussure'a był, wobec tego, język jako system - a nie sposób, w jaki posługują się nim ludzie. Nie wychodził poza semiotyczną rzeczywistość. Traktowanie języka, w ocenie Ricœura, jako zamkniętego systemu sprawia, że:

„stanowi on świat dla siebie, świat wewnątrz którego każdy element odnosi się tylko do innych elementów tego samego systemu. (...) Słowem, język nie jest już traktowany jak „forma życia”⁶⁵.

Na tej podstawie samowystarczalność języka-systemu zaprezentowaną w perspektywie strukturalistycznej, Ricœur uznał za ograniczenie. Saussure'owskie zawężenie rzutowało na marginalizowanie kwestii języka jako dyskursu.

Ricœur, w przeciwieństwie do Saussure'a. skoncentrował uwagę na dyskursie. Zdefiniował go jako posiadające znaczenie „zdarzenie językowe” (używa również określeń: zdarzenie mowy, zdanie)⁶⁶ lub „użycie językowe”⁶⁷. Doszedł do wniosku, że zaprezentowanie *parole*, jako abstrakcyjnego pojęcia może pomóc w pozyskaniu dla

64 *Ibidem*, s. 68.

65 *Ibidem*, s. 72.

66 *Ibidem*, s. 79.

67 Tłum. własne: *Discourse is language-event or linguistic usage*. P. Ricœur, *The model of the text: meaningful action considered as a text*, [w:] *Hermeneutics and the Human Sciences Essays on Language, Action and Interpretation*, Cambridge University Press, New York 2016, s. 160; P. Ricœur, *The model of the text: meaningful action considered as a text*, [w:] *From text to action; Essays in Hermeneutics, II*, red. R. Kearney, Northwestern University Press, Evanston 2007, s. 145;

Tłum. własne: *Le discours est l'événement de langage*. P. Ricœur, *Le Modèle du texte: l'action sensée considérée comme un texte*, [w:] *Du texte à l'action, Essais d'herméneutique, t. 2*, Editions du Seuil, Paris 1986, s. 184.

niego podobnego do *langue* statusu. Mimo, że jest to mechaniczne posunięcie, daje ono podstawy do rozpatrywania dyskursu w charakterze przedmiotu badań. Rozwijając kwestię określenia, czym jest przekaz Ricœur wyjaśnił, że opiera się on na aktualizacji zdarzenia. Aktualizacja następuje, gdy w interpretacji zdarzenie nabiera znaczenia. Nie chodzi tu bynajmniej o redukowanie dyskursu do jego aspektu czasowości, ale o aktualizację, „dzięki której nasza znajomość języka urzeczywistnia się w naszych wypowiedziach”⁶⁸. Poszukujemy zatem rozumienia znaczenia, czyli czegoś jednak trwałego. Na dyskurs bowiem składa się tymczasowość zdarzenia, ale i jego komunikowany sens.

Hermeneutyczna koncepcja dyskursu nie opiera się jednak na odtwarzaniu znaczeń. W zamyśle tej teorii dyskursu leży wypracowanie interpretacji i wyjaśnienie ich sensu. Ricœurowska *hermeneutics of praxis*⁶⁹ osadza się na ontologii działania. Dyskurs jest efektem wytwarzania, co sprawia, że nie mieści się on już w samej kategorii tekstu, ale dzieła. Wymaga od nas podjęcia wysiłku. Praca nad jego odczytywaniem nie jest nigdy skończona z uwagi na zmieniające się warunki naszego bytowania⁷⁰. W ten sposób tworzy się nierozzerwalna łączność między językiem-dyskursem a praktyką życia. Dyskurs jako dzieło wymaga nadania mu formy przy jednoczesnym zachowaniu zindywidualizowanego stylu⁷¹. Język zostaje zatem potraktowany jako surowiec, który kształtujemy według własnego zamysłu. Ricœur pisząc, że:

„Trud rozumienia polega więc na tym, żeby przenieść do dyskursu to, co najpierw ukazuje się jako struktura (...). Analiza strukturalna odsłania głęboką semantykę tekstu”.⁷²

Wskazuje tym samym na podleganie przekazowi względom praktyki i techniki. Wytwarzanie dyskursu, mimo że jest efektem działalności umysłowej, podobne jest do pracy z materia.

W hermeneutyce dyskursu Ricœur wytycza dwa komplementarne, choć nierównoważne podejścia do języka: jako systemu (strukturalno-funkcjonalne) i

68 P. Ricœur, *Teoria interpretacji...*, s. 79.

69 R. Kearney, *Foreword to the New Edition*, [w:] *From text to action; Essays in Hermeneutics, II*, red. R. Kearney, Northwestern University Press, Evanston 2007, s. IX.

70 P. Ricœur, *Ku hermeneutyce krytycznej*, „Pamiętnik Literacki”, 83, nr 1, 1992, s. 186.

71 P. Ricœur, *Hermeneutyczna funkcja dystansu*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 232.

72 P. Ricœur, *Ku hermeneutyce krytycznej...*, s. 186.

dyskursu (dynamiczne)⁷³. Lingwistyka dyskursu⁷⁴ jest zorientowana na odsyłanie do wypowiadającego się, czego nie czyni lingwistyka języka (w rozumieniu *langue*). Tak osadzona teoria dyskursu współgrała z zaproponowaną przez niego teorią sensu. Ricœur zasymilował w niej bowiem pewne elementy analitycznej filozofii języka⁷⁵. Odwołując się do pojęcia znaczenia (*Sinn*) i referencji (*Bedeutung*) u Fregego⁷⁶, stworzył ich własne rozwinięcie w postaci koncepcji „świata tekstu”⁷⁷. Francuski filozof zaproponował, aby sam dyskurs ujmować właśnie w analogicznych dwóch kategoriach: wewnętrznej organizacji i intencji referencyjnej. Pierwsza stanowi przekaz, druga roszczenie do mówienia czegoś o czymś⁷⁸. *Mieć coś na myśli* wychodzi poza to, co zostaje przez człowieka powiedziane. Posługiwanie się językiem nie składa się bowiem, rzecz jasna, jedynie z nazywania. Wysławianie polega na przekraczaniu w mowie dwóch progów: sensu idealnego i realnego odniesienia. Dyskurs można zatem przedstawić jako przejście od czystego orzekania do mówienia czegoś o rzeczywistości. Ten transcendujący ruch nakładania się immanentnej struktury na rzeczywistość stanowił według niego „duszę mowy”⁷⁹.

Aby zrozumieć trzeba podjąć trud połączenia aspektu sensu idealnego języka z odniesieniem do rzeczywistości. Język posiada bowiem moc odsłaniania tego, co nieświadome - jednak odkrycie wymaga wysiłku poznawczego. Wyparte wszak można zobiektywizować poprzez interpretację, o czym Ricœur był przekonany w związku ze studiami nad psychoanalizą⁸⁰. Inaczej niż Hans-Georg Gadamer, który w swojej hermeneutyce filozoficznej odżegnywał się od kwestii poznania, francuski filozof dostrzegał, że konieczny w rozumieniu jest „wysiłek epistemologiczny”⁸¹. Podporządkowywał go jednocześnie dążeniom ontologicznym, aby rozumienie było

73 D. Michałowski, M. Otrocki, *Hermeneutyka dyskursu...*, s. 132-134.

74 nazywana przez Ricœura semantyką.

75 M. Drwięga, *Paula Ricœura spotkania z psychoanalizą*, „Kwartalnik filozoficzny”, 44, nr 4, 2013, s. 117.

76 Przyporównuje również do *Bedeutung* i *Erfüllung* Husserla, P. Ricœur. *Struktura, wyraz, zdarzenie*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 304.

77 P. Ricœur, *Hermeneutyczna funkcja dystansu ...*, s. 239; Drwięga, *Paul Ricœur daje...*, s. 62.

78 P. Ricœur, *Teoria interpretacji ...*, s. 151.

79 P. Ricœur. *Struktura, wyraz, zdarzenie*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 304.

80 H. Lang, *Język i nieświadomość. Podstawy teorii psychoanalitycznej Jacques’a Lacana*, Wydawnictwo Słowo/Obraz Terytoria, Gdańsk 2005, s. 145.

81 P. Ricœur, *Zadanie hermeneutyki*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 192.

„sposobem odnoszenia się do bytów i do bycia”⁸². Próbował pokonać w ten sposób rozdzźwięk między hermeneutyką epistemologiczną Diltheya a ontologiczną Heideggera i Gadamera⁸³.

Refleksja-interpretacja stanowiła dla Ricœura sposób poznania w celu „uznania siebie”. Reflektujemy głównie nie po to by dać podstawy naszej wiedzy o świecie, ale „ponownie przywłaszczyć nasz wysiłek istnienia”⁸⁴. Epistemologiczny wysiłek rozumiany jako naukowość, w koncepcji francuskiego filozofa, przyjął wymiar bytowy⁸⁵. Stanowił zatem bardziej etyczne niż epistemologiczne zadanie. Sam język natomiast, miał pełnić rolę pomostu między epistemologią a ontologią⁸⁶. Jest on bowiem rzeczywistością bytu, w której dopiero pojawiają się przebliski refleksji. Dzieje się tak z uwagi na to, że „jesteśmy ogarnięci przez sytuacje naszego bytowania i ponieważ dążymy do ich rozpoznania i zrozumienia”⁸⁷. Nie stanowi więc świata dla siebie, gdyż ruch interpretacji nie pochodzi od jakiegoś abstrakcyjnego podmiotu refleksji. Wszak to ludzie mając coś do powiedzenia, starają się wyrazić w języku swoje doświadczenia.

Wbrew hermeneutyce romantycznej (epistemologicznej), w której na rozumienie składało się wystąpienie dwóch odrębnych momentów, czyli wyjaśniania i zastosowania, Ricœur traktował zjawisko to jako całość. Uważał, że pojęcie interpretacji może być stosowane „do całego procesu, który obejmuje wyjaśnianie i rozumienie”⁸⁸. Nie jest bowiem możliwe, aby najpierw uchwycić obiektywny sens, a potem przełożyć go na sytuacje nam znaną. Za Gadamerowską koncepcją stopienia horyzontów i wynikającego z niej rozumienia jako zastosowania, francuski filozof ujął rozumienie w kategorii interpretacji jako przyswojenia⁸⁹.

Niejednokrotnie w swoich tekstach Ricœur wskazuje jako źródło swojego programu hermeneutyki – fenomenologię Husserla. Związek z nią przejawia się w

82 *Ibidem*.

83 K. Rosner, *Hermeneutyka jako krytyka kultury*, PIW, Warszawa 1991, s. 233.

84 P. Ricœur, *O interpretacji. Esej o Freudzie*, Wydawnictwo KR, Warszawa 2008, s. 49.

85 P. Ricœur, *Zadanie hermeneutyki*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 192.

86 P. Ricœur, *Egzystencja i hermeneutyka*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 198.

87 P. Ricœur, *Teoria interpretacji ...*, s. 91.

88 *Ibidem*, s. 160.

89 G. B. Madison, *Hermeneutyka: Gadamer i Ricœur*, [w:] *Historia filozofii zachodniej*, red. R.H. Popkin, Zysk i S-ka, Poznań 2003, s. 718; Ricœur stał się tym samym jednym z prekursorów interpretacyjnego zwrotu w nauce, za sprawą artykułu z 1971 roku *The Model of the Text: Meaningful Action Considered as a Text*, [w:] *From text to action; Essays in Hermeneutics, II*, red. R. Kearney, Northwestern University Press, Evanston 2007.

postaci założeniowej natury ludzkiego poznania⁹⁰. Rozmaitość jest dana w naoczności zmysłowej, ale dociera do nas tylko poprzez akt syntezy w intelekcie. Jako że zanurzeni jesteśmy w Husserlowskim świecie życia (*Lebenswelt*), czy też mówiąc za Heideggerem naszą sytuację określamy jako bycie-w-świecie, pozostaje to kluczowe dla przyjęcia założeń rozumienia hermeneutycznego. W naukach humanistycznych określenie znaczenia tego, co badane znajduje się w kulturowym przekazie. Tym samym kod źródłowy programów komputerowych może stać się świadectwem ludzkiej egzystencji, które warto badać. Badania tego typu wymagają oczywiście wcześniejszego przyjęcia jakiejś narracji.

Wychodząc od zakorzenienia hermeneutyki w fenomenologii⁹¹ Ricœur wskazał w „*Le conflit des interprétations*” na dwie taktyki uprawiania nauki o rozumieniu. Pierwsza - *La voie courte*, czyli „droga krótka” reprezentuje rozważania, które pomijają kwestię metody i koncentrują się od razu na ontologii. Problem hermeneutyczny ujmowany na sposób ontologii rozumienia Heideggera (*ontologie de la compréhension*)⁹² skupia się bowiem wokół pytania: „czym jest byt rozumiejący?”. Hermeneutyka uprawiana z tej perspektywy jest analityką „*Dasein*, które egzystuje przez rozumienie”⁹³. Bezpośredniość tej ontologii pozwala na ścisłe trzymanie się powziętego tematu, jednocześnie ogranicza stawianie innych pytań. W hermeneutyce fundamentalnej nie ma przestrzeni na rozważanie warunków rozumienia. Ricœur natomiast dostrzegał taką potrzebę, a jednocześnie nie zamierzał zrezygnować z refleksji nad sposobem bycia bytu. Dążył do połączenia ontologii rozumiejącej z epistemologią interpretacji (*épistémologie de l'interprétation*). Tym samym hermeneutyka stanowiła w jego oczach „epistemologię, gdzie pojęcie »sensu« nasycą racjonalność”⁹⁴.

Ricœur idąc wspomnianą „długą drogą” zamierzał zatem powrócić do zagadnień ontologicznych ukazując w formach pochodnych rozumienia - ślady ich pochodzenia⁹⁵. Ontologia stanowiła horyzont, ku któremu zmierzał, nie zaś wyznaczoną

90 G. B. Madison, *Hermeneutyka: Gadamer i Ricœur*, ..., s. 716.

91 P. Ricœur, *Le conflit des interprétations. essais d'herméneutique*, Éditions du Seuil, Paris 1969, s. 23.

92 *Ibidem*.

93 P. Ricœur, *Egzystencja i hermeneutyka*..., s. 185.

94 P. Ricœur, *Krytyka i przekonanie. Rozmowy z Francois Azouvim i Markiem de Launay*, tłum. M. Drwięga, Wydawnictwo KR, Warszawa 2003, s. 108.

95 Ricœur pyta: *Ne vaut-il pas mieux dès lors partir des formes dérivées de la compréhension et montrer en elle les signes de leur dérivation ?* w: P. Ricœur, *Le conflit des interprétations*..., s. 26.

pozycję⁹⁶. Punktem wyjścia był język (mowa, później już tekst), zaś droga do rozumienia miała wieść przez jego analizę. Inaczej niż Gadamer, francuski filozof nie zamierzał oddzielać prawdy od metody. Zdystansował się wobec ujęcia, które z hermeneutyki filozoficznej robiłoby - prymarny wobec racjonalności naukowej - sposób rozumienia.

Zaproponowana przez Ricœura „okrężna droga”⁹⁷ hermeneutycznego rozumowania wiodła przez stopnie mediacji semantycznej⁹⁸ z użyciem analizy językowej. Filozof nakreślił jak interpretacja stopniowo przechodząc kolejne etapy dociera do ontologicznych korzeni. Z płaszczyzny semantycznej wiedzie przez refleksję, by w końcu dojść do fazy egzystencjalnej⁹⁹. Dlatego też w tym ujęciu nie da się odseparować ontologii od interpretacji. Jak zaznaczył filozof, ontologia jednak nie triumfuje nad interpretacją, właśnie z uwagi na to, że praca interpretacji i byt interpretowany są nierozdzielne¹⁰⁰. W późnym okresie twórczości Ricœura owa „długa droga” wiodąca przez analizę, obok „dialektyki bycia sobą i bycia tym samym” i „dialektyki bycia sobą i inności”, stała się częścią składową hermeneutyki siebie¹⁰¹. Poszukiwanie siebie odbywa się bowiem przez badanie śladów, w jakich ludzka egzystencja się wyraża. Jesteśmy sobie dani nie wprost, lecz „poprzez sposoby w jakich przejawiamy się w świecie”¹⁰², gdyż nie ma bezpośredniości poznania samego siebie. *Sobość* wyraża niepowtarzalną perspektywę każdego człowieka, ale uzyskiwaną poprzez refleksję nad sobą w świecie. Indywidualność splata się tu z antysolipsyzmem. W byciu sobą ujawnia się wymiar subiektywny i uniwersalny. *Ten-który-jest-sobą* ma do siebie dostęp tylko przez analizę i interpretację wytworów swojego działania. Zarys swojej ontologii, do której wiodła go „długa droga”, filozof nakreślił jednak dopiero u schyłku swojej pracy twórczej¹⁰³.

Ricœur, tak jak Gadamer, odszedł od metody hermeneutyki romantycznej opierającej się na poszukiwaniu sensu rozumianego jako zamierzenie autora.

96 S. Cichowicz, *Filozofia i hermeneutyka*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 13.

97 P. Ricœur, *O interpretacji. Esej ...*, s. 35.

98 W. Lorenc, *Hermeneutyczne koncepcje człowieka, W kręgu inspiracji heideggerowskich*, Scholar, Warszawa 2003, s. 237.

99 P. Ricœur, *Egzystencja i hermeneutyka...*, s. 190-203.

100 *Ibidem*, s. 202.

101 P. Ricœur, *O sobie samym jako innym*, przeł. B. Chelstowski, Państwowe Wydawnictwo Naukowe, Warszawa 2003, s. 30.

102 M. Kowalska, *Wstęp. Dialektyka bycia sobą*, [w:] P. Ricœur, *O sobie samym jako innym*, przeł. B. Chelstowski, Państwowe Wydawnictwo Naukowe, Warszawa 2003, s. XI.

103 P. Ricœur, *O sobie samym...*, s. 30.

Odmienne od Friedricha Schleiermachera, Wilhelma Diltheya, czy Emilio Betti'ego, którzy rekonstruowali myśli twórcy¹⁰⁴, francuski filozof był nieprzychylny ideałowi kongenialności. Tekst tak odczytywany nazwał „niemotą tekstu”¹⁰⁵. Na odejście od hermeneutyki romantycznej przez Ricœura rzutował powszechny w latach 60-tych XX wieku we francuskiej myśli filozoficznej, kryzys podmiotu i załamanie kartezjańskiej idei *cogito*. Wpłynęły one na zaprzestanie postrzegania intencji autora jako wiążącej dla interpretacji¹⁰⁶. Ricœura sprowokowała do tego szczególnie popularna wówczas lektura tekstów „mistrzów podejrzeń”¹⁰⁷, czyli dzieł Friedricha Nietzschego, Karla Marksa i Zygmunta Freuda¹⁰⁸. Ich refleksje wychodziły od założenia istnienia złudzeń, które należy demaskować. Interpretacje traktowali zaś jak „taktykę podejrzliwości”¹⁰⁹.

Wspomniani mistrzowie bezspornie byli czytani przez Ricœura w sposób krytyczny. Strategia walki z maskami, którą przyjęli, podporządkowywała filozofię stylowi uprawiania hermeneutyki opierającym się na demistyfikacji. Ich teorie interpretacji były jednak jednymi z wielu w „popękanym polu hermeneutycznym”. Ten koniec hermeneutyki ogólnej wiązał się z pojawieniem się napięcia pomiędzy skrajnymi sposobami podejścia do interpretacji. Chociaż tekst powinien, jak przyznawał Ricœur przejść „próbę podejrzeń”, to twierdził, że krytyka nie może stanowić jedyne trzonu hermeneutyki. Ta musi nabrać rysu konkretnej refleksji.

Ricœur ostrzegał równocześnie przed popadaniem w przeciwną wobec krytyki skrajność polegającą na ujmowaniu hermeneutyki jako objawienia. Takie zawężenie w kerygmacie, jest też nieadekwatne. Refleksja bowiem nie jest nam dana (*gegaben*), ale stanowi dla nas wyzwanie, rodzaj zadania (*aufgegeben*)¹¹⁰. W czasach, jak uważał, kryzysu mowy, interpretacji i refleksji, gdy:

„skrajne rozpięcie jest najbardziej wiarygodnym wyrazem naszej »nowoczesności«; sytuacja w której znalazł się dziś język, zawiera tę dwojaką możliwość, tę dwojaką pokusę, tę dwojaką konieczność: z jednej strony, oczyścić dyskurs z narośli, zniszczyć bożyszcza, przejść od upojenia do powściągliwości, sporządzić wreszcie bilans naszego ubóstwa; z drugiej strony,

104 G. B. Madison, *Hermeneutika: Gadamer...*, s. 715; W. Kalaga, *Mgławice dyskursu*, Universitas, Kraków 2001, s. 25.

105 P. Ricœur, *Wydarzenie i sens w mowie*, tłum. E. Bieńkowska, „Teksty: teoria literatury, krytyka, interpretacja” nr 6, 1972, s. 114.

106 A. Burzyńska, *Dekonstrukcja i interpretacja*, Universitas, Kraków 2001, s. 97.

107 *Ibidem*, s. 162.

108 P. Ricœur, *O interpretacji. Esej ...*, s. 39.

109 P. Ricœur, *Interpretacja a refleksja: konflikt hermeneutyczny*, „Studia Filozoficzne”, nr 9 (250), 1986, s. 141.

110 P. Ricœur, *O interpretacji. Esej ...*, s. 50.

zastosować ruch najbardziej »nihilistyczny«, najbardziej destrukcyjny, najbardziej obrazoburczy, aby pozwolić »mówić« to, co kiedyś, co za każdym razem było »powiadane«, kiedy sens pojawiał się na nowo, kiedy sens był pełen”¹¹¹.

Ricœur poszukiwał środka na przezwycięzenie tego napięcia. Sposobem na przeprowadzenie dociekań nie było według niego ani pozbawianie się złudzeń, ani obwieszczenie przesłania, lecz kierowanie się dwoma motywacjami z nich zaczerpniętymi: wolą podejrzenia i wolą wysłuchania. Kreują one „sprężynę hermeneutyki”¹¹². Interpretacja jako redukcja złudzeń i jako odnowa sensu pozostają z sobą w ciągłej relacji tworząc „okrężną drogę”¹¹³. Filozof nazywa je dwoma wierzchołkami trójkąta. Trzeci stanowi natomiast refleksja, która jest „przywłaszczeniem sobie naszego wysiłku istnienia i naszego pragnienia bycia poprzez dzieło dające świadectwo temu wysiłkowi i temu pragnieniu”¹¹⁴. W niej właśnie się wyrażamy i ustanawiamy. Ma ona więc dla nas wymiar bytowy.

Krytykując „szkołę podejrzeń” Ricœur równocześnie dokonał reinterpretacji poglądów Freuda, Nietzschego i Marksa. Zaproponował w ich miejsce filozofię wiary. Przyznał, że nie jest już oczywiście możliwa ta pierwotna, bezpośrednia, a jedynie „poszukująca drugiej naiwności”¹¹⁵ - ta, która przeszła już krytykę stając się wiarą rozumową. Tym sposobem powstała analiza wsłuchująca się w to, co tekst wyraża. Francuski filozof nazywał ją wiarą hermeneuty. Jej postkrytyczność opiera się na zaufaniu do języka, biorąc swój początek z przekonania, że „ludzie urodzili się wewnątrz”¹¹⁶ niego. Ta orientacja oddziałuje na powracanie w jego twórczości do kwestii nasłuchiwanie w języku. Sam tekst według filozofa nie zwodzi, tak jak zakładali mistrzowie podejrzeń, lecz umożliwia nam poszukiwanie intencji z jaką został napisany. Karl Simms przyrównał zaufanie Ricœura, jakie ten pokładał w dyskurs, do wiary w słowa przepowiedni, gdzie „teksty są podobne do delfickiej wyroczni: mogą kryć przed nami znaczenie, ale nie kłamią”¹¹⁷. Francuski filozof uważał bowiem, że sens tekstu przekracza intencję z jaką został napisany. „Rozumienie porusza się w przestrzeni nie psychologicznej” tak, jakby to miało miejsce nie w ramach

111 *Ibidem*, s. 35.

112 P. Ricœur, *Interpretacja a refleksja* ..., s. 142.

113 P. Ricœur, *O interpretacji. Esej* ..., s. 35.

114 P. Ricœur, *Interpretacja a refleksja*..., s. 154.

115 P. Ricœur, *O interpretacji. Esej* ..., s. 36.

116 *Ibidem*, s. 37.

117 Tłum. własne „Written works are like the Delphic oracle: they may hide their meaning from us, but they do not lie” z K. Simms, *Paul Ricœur*, Routledge, London 2003, s. 2.

hermeneutyki romantycznej, „a semantycznej, którą zakreślił tekst”¹¹⁸. Toteż sens danego tekstu wykracza poza analizę psychologiczną, horyzont życia i twórczości autora i jako taki - musi być od nowa odczytywany.

Hermeneutyczna wiara w możliwość odnajdywania sensu tekstu stała się podbudową metody stosowanej przez Ricœura. Filozof poszukiwał podobieństw zachodzących między wydawałoby się konkurującymi ideami, a następnie budował na ich gruncie swoje rozwiązania. Czerpał z opartej na objawieniu hermeneutyki element zaufania do języka, a ze szkoły podejrzeń wymóg przeprowadzania krytyki. Samo zagadnienie metody stało się natomiast istotnym elementem hermeneutyki Ricœura¹¹⁹. Miejsce egzegezy pisma klasycznej hermeneutyki, czy rekonstrukcji myśli autora w hermeneutyce romantycznej, zajęła u niego kwestia ludzkiego rozumienia.

Teoria interpretacji Ricœura zależeć miała od praktyki, gdyż „w przestrzeni sensu (przestrzeni semantycznej, *przyp. aut.*) rozumienie ma zbudować hipotezę sensu, postawić na własną wiarygodność i potwierdzić własne prawdopodobieństwo za pomocą technik krytycznych”¹²⁰. Ricœur tworząc metodę hermeneutyczną oparł ją zatem na działaniu człowieka¹²¹. Jej istotę stanowiło to, co dzieje się, gdy ktoś coś mówi. Rozumienie sytuuje się bowiem w ludzkich praktykach, nie zaś w samej sferze reprezentacji. Za niezbędne uważał przeprowadzanie wpierw badań lingwistycznych języka jako systemu, jednak to językoznawstwo oparte o dyskurs pozwalać miało na odnalezienie znaczenia tekstu. Poznanie systemu stanowi warunek wstępny, ponieważ zapoznaje nas z obowiązującymi schematami komunikacji. Rozumienie realizuje się natomiast w dyskursie, który jest rodzajem wymiany. Kluczem dialogu jest więc racjonalność komunikacyjna - nie zaś racjonalność instrumentalna.

Ze względu na to, że „język w mniejszym stopniu jest mówiony przez ludzi, a bardziej mówiony do ludzi”¹²², Ricœur postanowił ująć podmiot jako byt komunikujący. Jego teoria zakłada odbiorcę komunikatu, który dialog podejmuje, bądź nie. W każdym razie ma szanse na uczestnictwo w tym zdarzeniu w zależności od swojej kompetencji językowej¹²³. W tak wyznaczonym kierunku uwidacznia się wpływ

118 P. Ricœur, *Wydarzenie i sens* ..., s. 114.

119 W nacisku na wspomnianą analizę języka, a potem wspomnianą wyżej obiektywizację w interpretacji można dostrzec odwołanie do analitycznej filozofii języka.

120 P. Ricœur, *Wydarzenie i sens*..., s. 114.

121 G. B. Madison, *Hermeneutyka: Gadamer* ..., s. 719.

122 P. Ricœur, *O interpretacji. Esej o* ..., s. 37.

123 P. Ricœur, *Hermeneutyczna funkcja* ..., s. 228.

anglosaskiej filozofii języka na kształtowanie się koncepcji języka jako dyskursu. Ricœur bowiem, aby opisać dialektykę znaczenia i zdarzenia, odwołał się do teorii aktów mowy Austina z „Jak działać słowami” i Searla z „Czynności mowy”.

Mimo że Ricœur rozpoznaje lokucyjny, illokucyjny i perlokucyjny aspekt aktu mowy, w kontekście dyskursu kładzie jednak nacisk na interlokucję (allokucję)¹²⁴. Koncepcję sensu i wartości logicznej zdań Ricœur uzupełnił o rolę lokutora i interlokutora¹²⁵. Istnieje bowiem jakiś sprawca nadawania sensu, który mówi do kogoś, o czymś, czyli „nadaje referencje”¹²⁶. Elementy te składają się na to, jak tekst jest tworzony i odbierany, czyli na dyskurs. Nadawca i odbiorca komunikatu są niezbędni do jego zaistnienia. I chociaż nie możemy dzielić się naszymi doświadczeniami, to możemy dzięki temu wymiarowi języka przekazywać znaczenia. Język nie stanowi świata dla samego siebie, wnosimy bowiem do niego nasze doświadczenia¹²⁷. W dialogu upubliczniamy sens, nawet gdyby miała to być rozmowa z samym sobą¹²⁸. Na tej podstawie Ricœur uznaje dyskurs jako nadrzędny wobec języka jako systemu. Funkcja tego drugiego podporządkowana jest bowiem opisanej tu więzi komunikacyjnej.

Dyskurs w przeciwieństwie do języka jako systemu, w którym „słowa odsyłają jedne do drugich w nieskończonym kręgu słownika”¹²⁹ ma związek z rzeczywistością. Ricœur ujął język jako dyskurs, czyli ten sposób, który wyraża świat. On, a nie język pojęty jako system, ma tę zdolność. Ludzkie doświadczenie odznacza się tym, że domaga się od nas, aby je wyrazić i z tego względu cel hermeneutyki Ricœura można określić jako „rozpoznawanie dyskursu w dziele”¹³⁰. Przekaz sięga rzeczywistości, co oznacza, że odnalezienie w nim wspólnego odniesienia wymaga jednak współdzielenia przez uczestników sytuacji komunikacyjnej. W rozmowie pełni tę rolę ostensywność mowy. W piśmie natomiast brak jest współdzielenia sytuacji przez uczestników dyskursu i „język doznaje gloryfikacji sam dla siebie, kosztem funkcji referencyjnej”¹³¹. To „unicestwienie świata” typowe dla literatury, staje się problemem do rozważenia

124 *Ibidem*, s. 229.

125 M. Drwiega, *Paul Ricœur daje ...*, s. 20.

126 P. Ricœur, *Struktura, wyraz, zdarzenie*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 301.

127 P. Ricœur, *Język jako dyskurs...*, s. 91.

128 *Ibidem*, s. 83-85.

129 P. Ricœur, *Hermeneutyczna funkcja...*, s. 239.

130 *Ibidem*, s. 236,

131 *Ibidem*, s. 240.

podczas hermeneutycznej analizy tekstu kodu źródłowego. Wprawdzie nie kwestia lingwistycznego zachwyty językiem nie pojawia się podczas gdy zajmujemy się językami programowania, mimo to można zastanowić się nad brakiem w kodzie wskazywania będącego bezpośrednim odniesieniem do rzeczywistości.

Na teorię dyskursu francuskiego filozofa wpływała również fenomenologia Husserla, gdyż dyskurs „wyraża świat”, bo jak pisał Ricœur:

„Trzeba założyć, że doświadczenie w całej swojej rozciągłości posiada zasadniczo czytelność. Doświadczenie może zostać wypowiedziane. Wymaga, aby o nim mówić. Przenieść je w język nie znaczy zmienić je w coś innego, ale je wyrazić i rozwinąć, spowodować, by stało się sobą samym”¹³².

Z fenomenologii wyrosła kwestia dystansu niezbędnego w rozumieniu. Od samego początku problem ten jest przedstawiany przez tego hermeneutę jako kluczowy dla rozumienia w języku. Jego istotność ujawnia się już w początkowym okresie twórczości Ricœur.

Za „modelowy przypadek komunikacji na dystans”¹³³ Ricœur uznał tekst. W nim znaczenie dystansuje się od zdarzenia powodując samowystarczalność pisma¹³⁴. W tekście zatem można osiągnąć rodzaj obcości, który warunkuje interpretację jakiejś „propozycji świata”¹³⁵. Innymi słowy dystans stanowi warunek niezbędny dla obiektywizacji. Pismo wyzwala treść od „warunków dyskursu, jakie ustanawia rozmowa”¹³⁶. Osiągnięta w tekście autonomia uniezależnia interpretację od kontekstu, w jakim został on napisany, czy zamiarów autora. Daje nowe możliwości bycia-w-świecie dzięki „wariacji wyobraźni” i tym samym sens tekstu może być wciąż odnajdywany. Rozumiejąc nie tworzymy odzwierciedlenia rzeczywistości, lecz poprzez uczestnictwo w niej, jesteśmy w stanie wyrazić ją w języku. Interpretacja polega bowiem na przekształcaniu naszego doświadczenia. Ricœur podważał kwestię czysto intelektualnej intuicji, gdyż refleksja nie jest nam nadana. Natomiast nie jest ona też krytyką poznania, co sprawia, że stanowisko filozofa różni się od podejścia Kanta. Aby

132 P. Ricœur, *Fenomenologia i hermeneutyka. Wychodząc od Husserla*, tłum. M. Drwięga, „Kwartalnik Filozoficzny”, 24, nr 3, 1996, s. 165.

133 P. Ricœur, *Hermeneutyczna funkcja ...*, s. 225.

134 M. Szulakiewicz, *Hermeneutyka- czyli filozofia jako rozumienie*, [w:] *Filozofia Współczesna*, red. L. Gawor, Z. Stachowski, Lublin 2006, s. 316,

135 P. Ricœur, *Hermeneutyczna funkcja...*, s. 241.

136 *Ibidem*, s. 237.

zrozumieć należy podjąć się zadania intelektualnego, które ma wymiar bytowy. Bazuje na interpretacji i wysiłku, który podjęliśmy, żeby w ten sposób istnieć.

W kontekście metodologii zalecenie rozpoznawania przekazu w dziele mogłoby się wydawać mało formalną zasadą. Jednak Ricœur dostrzegał potrzebę obiektywizmu, a nawet kładł nacisk na zachowanie precyzji charakteryzującej nauki ścisłe¹³⁷. Kwestia ta uwidoczniła się w zaleceniach dotyczących etapu strukturalnego badania tekstów, gdyż:

„semantyczna autonomia tekstu otwierała go na podejścia uwzględniające jedynie jego obiektywność, jako tekstu wypowiedzianego, napisanego, a zatem zobiektywizowanego. W celu lepszego zrozumienia uznaję w bardzo pozytywnym sensie obiektywizację za obowiązkowe przejście przez wyjaśnienie, nim wróci się do wypowiadającego się.”¹³⁸

Dopiero w dyskursie pojawia się podmiot, a zatem subiektywizacja. Nie można więc zarzucać Ricœurowi, że zaproponowany proces interpretacji przybierać miał dowolną formę. Warto mieć jednocześnie na uwadze, że ograniczoność metody, z uwagi na jej bytowy status jako narzędzia, pozwala dochodzić do prawdy dopiero, gdy uzupełniło się ją o badanie w duchu hermeneutyki filozoficznej.

Zamiast *opanowania* tekstu, Ricœur proponuje wspólne poszukiwanie prawdy. Hermeneutyka tak rozumiana przeciwstawiała się „wszelkim roszczeniom totalizującym”¹³⁹, które narzuca „kultura wskazywania i pouczenia”¹⁴⁰. Jedność może stanowić pokusę¹⁴¹, natomiast takie ujednolicenie jest ani możliwe, ani pożądane. Tekst jest inny w zależności od odczytania. Uznanie jedności i wielości tworzy pole dyskursu, „jeśli uzna się, że jedność nie jest jakąś sytuacją, która może i powinna być stawiana w opozycji do wielości, lecz jest raczej stanem, który wymaga wielości”¹⁴². Tekst jest spisany dyskursem, którego sens poznajemy w interpretacji. Realizuje najgłębsze dążenie dyskursu. Świat tekstu tworzy bowiem szczególny sposób rozumienia rzeczywistości.

Wobec tekstu stoi kluczowe wyzwanie bycia należycie zrozumianym. Niemożliwe jest bowiem korygowanie tak jak czyni się to w mowie - poprzez reakcję rozmówcy. Próżno też prosić o wyjaśnienie niejasno wyrażonej myśli. Mimo więc tego, że w

137 P. Ricœur, *Ku hermeneutyce krytycznej...*, s. 182-83.

138 P. Ricœur, *Krytyka i przekonanie...*, s. 114.

139 P. Ricœur, *Pamięć, Historia, Zapomnienie*, Universitas, Kraków 2007, tłum Janusz Margański, s. 459.

140 M. Szulakiewicz, *Czas i to co ludzkie, Szkice z chronozofii i kultury*, Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika, Toruń 2011, s. 127.

141 P. Ricœur, *Prawda i kłamstwo*, [w:] *Podług nadziei*, P. Ricœur, PAX Warszawa 1991, s. 57.

142 M. Szulakiewicz, *Czas i to ...*, s. 164.

filozofii Ricœura źródeł ciężenia ku językowi można doszukiwać się w myśli Heideggera, to podejście do mowy i pisma ich akurat różniło. Za najprzychylniejszy warunek rozumienia Heidegger uznawał bowiem mowę języka naturalnego (*Rede*), a „wydarzanie” w niej było aktualizacją *Dasein*¹⁴³. Wyrazem życia miała być jej aktualność, gdy ograniczeniu podlegało pismo. Tymczasem tekst, jak twierdzi Ricœur, ma nad mową metodologiczny prymat. Dla badacza zapis jest dogodną formą, ponieważ pozwala oddzielić znaczenie dyskursu od intencji autora. Mówiący jest natomiast „obecny w autentycznym sensie bycia-tam, w sensie *Dasein*”¹⁴⁴ i dopiero mentalne intencje autora da się oddzielić od jego tekstu.

Utrwalenie przekazu w formie zapisu oddala od nadawcy. Obdarza to tekst semantyczną autonomią, której nie ma forma werbalna. Pismo pozwala tym samym na zdystansowanie się i uniknięcie błędu intencjonalności, błędu polegającego na uznaniu intencji autora za kryterium interpretacyjnej trafności. Brak bezpośredniości, który zachodzi w tekście gdy mówca nie przynależy *de facto* do sytuacji interlokucyjnej, niesie również pewne niebezpieczeństwa. Niedostrzeganie autora może grozić absolutyzacją tekstu, a to z kolei potraktowaniem go jako przedmiotu natury.

Nie ma przeszkód by język kodu źródłowego przeanalizować w kontekście pojmowania języka jako dyskursu. W takim ujęciu wykracza się już poza racjonalność instrumentalną i opiera na instrumentalności komunikacyjnej. Trzeba najpierw jednak skonfrontować się z przekonaniem Gadamera i innych hermeneutów wypowiadających się w podobnym tonie na temat tego, że poznanie poprzez sztuczne symbole jest „ślepe”¹⁴⁵. Marburczyk zauważył, że użytkownika takiego systemu pozbawia się aspektu doświadczenia. Z dezaprobatą odnosił się do języka znaków graficznych podkreślając tym samym sztuczność symboliki i działanie wbrew duchowi języka. Za Heideggerem, który krytykował „myślenie rachujące” współwystępujące w zestawie z techniką (*Ge-stell*)¹⁴⁶ i Gadamerem dezaprobuującym „kalkulowanie” w czysto sztucznym języku¹⁴⁷, Ricœur podobnie wyrażał negatywne nastawienie do sztuczności

143 I. Lorenc, *Ontologizacja języka a koncepcja źródłowego logosu w ujęciach Heideggera i Merleau-Ponty'ego*, „Ruch filozoficzny”, 73, nr 3, 2018, s. 98.

144 P. Ricœur, *Mowa i pismo*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 102.

145 H.-G. Gadamer, *Prawda i metoda*, Państwowe Wydawnictwo Naukowe, Warszawa 2013, s. 562.

146 M. Lubecki, *Przestrzeń otwartości w przyczynkach do filozofii Martina Heideggera*, „Zeszyty Naukowe Towarzystwa Doktorantów UJ. Nauki Humanistyczne”, nr 3, 2011, s. 55.

147 H.-G. Gadamer, *Prawda i metoda...*, s. 562.

języka¹⁴⁸. W jednym rzędzie stawiał budowanie języków sformalizowanych i manipulację pustymi symbolami. Tymczasem w związku z rozumieniem języka jako dyskursu, można stworzony za pomocą sztucznego języka wytwór, ująć poprzez jego przekaz.

Języki programowania, za pomocą, których zapisany jest kod źródłowy programów komputerowych uwikłane są w racjonalność, a ponadto zależne od technologii. Programiści bazują bowiem na logice formalnej: funkcjach rekurencyjnych i automatach¹⁴⁹. W programowaniu można jednak doświadczać hermeneutycznie. Gdyby nawet przyjąć, że języki programowania są realizacją utopijnego projektu Leibniza (choć nią nie są), to możemy je wciąż badać wchodząc na drogę hermeneutyki filozoficznej. Nadal bowiem w takim przypadku rozumiemy w języku naturalnym, a tekst programu komputerowego - jego język jako dyskurs - traktujemy w charakterze przedmiotu badania.

Już tylko biorąc pod uwagę sam przekaz powstaje możliwość ujęcia tekstu kodu źródłowego jako wytworu świata kultury i świadectwa ludzkiej egzystencji. Można w tym właśnie upatrywać sposób na bycie „zapytywanym ponad ciszą i zapomnieniem, które rozprzestrzeniają manipulacje pustymi symbolami i budowanie języków sformalizowanych”¹⁵⁰. Analizowanie języka kodu źródłowego jako przekazu umożliwia spojrzenie na niego jak na obcy kulturowo tekst i potraktowanie jak dzieła, w którym ujawnia się ludzkie doświadczenie świata. W hermeneutyce dyskursu zawierają się bowiem dwie zależne od siebie dialektycznie części filozofii Ricœura: hermeneutyka skupiająca się na interpretacji tekstów i fenomenologia hermeneutyczna obejmująca kwestie doświadczenia i działania. Całościowe podejście do języka, które rozwinął francuski filozof wyrosło ze splotu hermeneutyki i fenomenologii. Heideggerowskie objawianie się bycia w języku i husserlowskie doświadczenie stanowiły *spiritus movens* jego filozofii. U podstaw fenomenologii hermeneutycznej leżało bowiem pytanie o sens bycia i możliwość wypowiedzenia tego doświadczenia. Skoro hermeneutyka ma

148 P. Ricœur, *O interpretacji. Esej ...*, s. 38-39.

149 *Logika formalna*, red. W. Marciszewski, Państwowe Wydawnictwo Naukowe, Warszawa 1987, s. 7-8, s. 182-214.

150 P. Ricœur, *O interpretacji. Esej...*, s. 38-39.

zasadzać się na „rozpoznawaniu dyskursu w dziele”¹⁵¹ to wszystko prowadzi do tego, co dzieje się w języku i jak jest interpretowane.

Sposób, w jaki Ricœur ujmuje język, w tej pracy, stanowi punkt wyjścia do rozważań nad działaniem językowym programistów. Obszar badań, wykraczający poza warstwę tekstową kodu źródłowego, wywodzi się bowiem z rozumienia języka jako dyskursu. Użytecznym narzędziem analitycznym okazuje się zarówno pojęcie symbolu, koncepcja języka-systemu i dyskursu, ale także wyrosła z narracji tożsamość narracyjna z okresu prac nad „hermeneutyką odkrycia siebie”¹⁵² – objaśniona i uwzględniona w dalszych rozważaniach. Wykorzystując „okrężną drogę” rozumienia, można przebadac w interpretacji warstwę tekstową kodu źródłowego, by w efekcie móc dojść do wymiaru bytowego. „Bycie poprzez dzieło” daje bowiem świadectwo „wysiłku istnienia”¹⁵³ programistów. Ich doświadczenie świata ukazuje się w kodzie źródłowy programów komputerowych – dziele, które staje się przestrzenią oczekującą na interpretację.

1.3. Język jako forma symboliczna w filozofii Ernsta Cassirera

Do nieskończoności interpretacji dzieła, które wynika w podejściu Ricœura z dyskursywnego ujęcia języka, odniósł się również, w rezultacie odmiennych przemyśleń i inaczej problem artykułując, Cassirer. Inklinacji hermeneutycznych w jego myśli można dopatrzeć się na kartach „Logiki nauk o kulturze”. Stwierdza tam, że „wola twórcza i siła twórcza, z których się zrodziło (dzieło, *przyp. aut.*), żyje i działa w nim nadal i wciąż prowadzi do nowych dzieł”¹⁵⁴. Sam proces trwania myśli w dziele jest zatem wynikiem „kształcenia ludzkości” (*Bildung*) polegającym na wytwarzaniu i przekazywaniu w formach kulturowych „wspólnej własności”¹⁵⁵. Natomiast skoro w wytworach kultury obiektywizuje się myśl, aby ją na powrót wyzwolić, niezbędne staje

151 P. Ricœur, *Hermeneutyczna funkcja ...*, s. 236.

152 Tłum. własne *Recovery self*, za A. Scott-Baumann, *Ricœur and the Hermeneutics of Suspicion*, Bloomsbury Academic, London, 2009.

153 P. Ricœur, *Interpretacja a refleksja...*, s. 154.

154 E. Cassirer, *Logika nauk o kulturze: pięć studiów*, red. P. Parszutowicz, Wydawnictwo Marek Derewiecki, Kęty 2011, s. 143.

155 *Ibidem*.

się działanie jednostek. W „miejscach pamięci ludzkości”¹⁵⁶ takich jak, na przykład, dzieło literackie, czy program komputerowy¹⁵⁷ zachowują się chęci, uczucia i przemyslenia podmiotu.

By dotrzeć do zawartości utworu ujętej w ramach Cassirerowskiej formy, niezbędne jest zawsze nowe pobudzenie. Sam język oczywiście nie jest w tej perspektywie samą aktywnością, ale procesem transcendentnym, w którym ujawnia się rzeczywistość. Mimo że forma ma sztywną postać, to samo jej źródło jest aktywne i pochodzi z ludzkiego działania. Wydobywanie znaczenia w filozofii form symbolicznych odbywa się zatem wciąż na nowo. Zestawiając tę myśl z ujęciem interpretacji w hermeneutyce dyskursu, w którym wszak jest ona niekończącym się twórczym przekazem, mamy szansę ujrzeć pewne podobieństwo. W obu podejściach dostrzega się konieczny w tworzeniu i odczytywaniu dzieła aspekt działania podejmowanego w języku.

W podejmowaniu się przez człowieka twórczego zadania Cassirer dostrzegał wymiar niejako ontologicznym, gdy konstatował, że „dzieło nie jest w zasadzie niczym innym, jak ludzkim czynem zagęszczonym w byt, który jednak także w tej sztywnej postaci, nie wyrzeka się swego źródła”¹⁵⁸. Wytwory człowieka, jako że są ekspresją tego co duchowe, dają możliwość utrwalenia w nich ludzkich emocji i myśli. W ujęciu tym nie tylko wychodzi się zatem poza materialny aspekt dzieła skupiając się na możliwości wyrażania jednostki, ale także dostrzega akt możliwości wyzwalać się w jakimś innym „podatnym podmiocie”¹⁵⁹ potrzeby następnego działania, czyli interpretacji utworu.

Cassirer w swoim filozoficznym projekcie skupił się jednak nie wokół bycia czy działania, ale na poznaniu¹⁶⁰. Ze względu na to, że wierne odwzorowanie świata uznał za niewykonalne, za rdzeń percepcji przyjął symbolizm¹⁶¹. Reprezentacja powstająca w

156 *Ibidem*, s. 142.

157 Dzieło literackie jest klasycznym tekstem, program komputerowy zapisany jest zaś w hipertekstualnym kodzie źródłowym.

158 E. Cassirer, *Logika nauk o...*, s. 143.

159 *Ibidem*, s. 142.

160 Oczywiście w twórczości Cassirera wyróżnia się okresy: np. według M. Hinsch’a: fazę teoriopoznawczo-epistemologiczną (Berlin, 1903–1919), fazę filozofii kultury (Hamburg, 1919–1933), fazę antropologiczną (Oxford, 1933–1935; Göteborg, 1935–1941; New Haven i New York, 1941–1945) za B. Trochimska-Kubacka, *Koncepcja sztuki w filozofii kultury Ernsta Cassirera*, „Studia Philosophica Wratislaviensia”, 14, nr 3, 2019, s. 86.

161 J. M. Krois, *Cassirer: Symbolic Forms and History*, New Heaven, London 1987, s. 83.

ten sposób miała umożliwiać zrozumienie. Zaznaczył się tym samym kolejny rys interpretacyjny w filozofii form symbolicznych. Uwaga ta nie świadczy jednakowoż o zamiarze, aby w tej pracy ignorować dziedzictwo kantyzmu wyraźnie odbijające się w twórczości Cassirera, jedynie by wydobyć wspólne akcenty filozofii jego i Ricœura. Marburczyk wszelkie rozważania prowadził bowiem w ramach ujęcia transcendentального – filozofii języka¹⁶², bądź szerzej kultury – a nie ujęcia hermeneutycznego.

Neokantyzm w wymiarze epistemicznym stanowi punkt wyjścia do wszelkich rozważań Cassirera o języku. Wprawdzie myśliciel ten z czasem coraz bardziej oddalał się od zagadnień szkoły marburskiej, ale nadal rdzeniem jego filozofii pozostaje przekonanie, że przyczyną poznania rzeczy jest rozum¹⁶³ i istnienie w umyśle czynników apriorycznych. Rozwinął je pod postacią koncepcji o apriorycznej aktywności podmiotu, której owocem jest ludzka twórczość. Dzięki takiemu ujęciu mógł znajdować w umyśle stałość, której brakuje rzeczywistości. Różne sfery kultury jako wytwory ducha, mają stanowić „»symbol« wobec sfery wrażeń”¹⁶⁴. Badanie kultury odbywać musi się zatem poprzez analizę form symbolicznych, w tym warto podkreślić między innymi języka, gdyż to one formują obraz świata i strukturę jego poznania. Według Cassirera zanurzenie w świecie kultury jest wynikiem przekroczenia granicy życia organicznego i osiągnięciem charakterystycznym tylko dla człowieka.

Wytworzenie systemu symbolicznego umożliwiło ludziom życie w dodatkowym wymiarze, gdyby ich sytuację porównać do innych zwierząt. Różnica ta zmienia rzeczywistość człowieka. Nie może się on już do niej bezpośrednio ustosunkować¹⁶⁵, bo zmuszony jest korzystać z coraz bardziej skomplikowanej symbolicznej sieci. Płaci w ten sposób cenę za postęp w zakresie myśli i doświadczenia. Nie będąc w stanie konfrontować się z rzeczywistością wprost, człowiek „zamiast zajmować się rzeczami samymi w sobie, (...) w pewnym sensie ustawicznie sam z sobą rozmawia”¹⁶⁶. Ujrzyć

162 M. Szulakiewicz, *Obecność filozofii transcendentальной*, Wydawnictwo Uniwersytetu Mikołaja Kopernika, Toruń 2020, s. 208.

163 Marburczycy odpierali zarzut o brak realizmu głosząc właśnie, iż rozum, bynajmniej nie jest przyczyną rzeczy, lecz przyczyną poznania rzeczy, M. Bal-Nowak, *Mit jako forma symboliczna w ujęciu Ernsta A. Cassirera*, Zakład Wydawniczy Nomos, Kraków 1996, s. 29.

164 B. Andrzejewski, *Problem noumenu w kantyzmie i neokantyzmie*, W kręgu inspiracji kantowskich, red. R. Kozłowski, Państwowe Wydawnictwo Naukowe, Warszawa-Poznań 1983, s. 114.

165 E. Cassirer, *Esej o człowieku. Wprowadzenie do filozofii ludzkiej kultury*, Czytelnik, Warszawa 1977, s. 80.

166 *Ibidem*.

w tym można kontynuację kantowskiej myśli o tym, że podmiot stanowi warunek przedmiotu. Dialogiczny charakter poznania oparty jest ponadto w tej koncepcji na przymusie pośrednictwa form symbolicznych. Człowiek „owinięty” w formy językowe nie wykroczy poza nie w swoim poznaniu, bo rzeczywistość nie jest mu dana, lecz niejako „zadana”.

Rzeczona filozofia form symbolicznych przynależy do centralnego etapu twórczości Cassirera. Zaczął on bowiem od dociekań nad problemami przyrodoznawstwa, które zresztą kontynuował również w środkowym okresie swojej działalności. Do postulatu, że sens funkcjonowania człowieka nie polega jedynie na aktywności intelektualnej, filozof doszedł wszakże już podczas pisania „*Freiheit und Form*” oraz „*Idee und Gestalt*”¹⁶⁷, jednak swój projekt w pełni przedstawił w „*Filozofii form symbolicznych*”. Przyrodoznawstwo, powtarzał za Kantem, miało służyć jedynie „do sylabizowania zjawisk, by móc je odcyfrować jako doświadczenie”¹⁶⁸. Wyszedł zatem poza rozważania o nauce i skupił się na rozszerzeniu założeń koncepcji symbolizmu na kulturę duchową. Warto jednak podkreślić, że to początkowe refleksje nad metodą poznania i systematyką wiedzy przywiodły go do bycia teoretykiem kultury¹⁶⁹.

Antynaturalizm metodologiczny Cassirera przejawiał się w dostrzeżeniu odmienności przyrodoznawstwa od nauk o duchu¹⁷⁰, przy równoczesnym zaznaczeniu ich wzajemnego uzupełniania się¹⁷¹. Filozof starał się znaleźć dla nauk o języku, sztuce i religioznawstwa miejsce w systemie logiki nauk, tak jak wcześniej uczynić to miał chociażby wobec biologii – Arystoteles, czy względem przyrodoznawstwa – Immanuel Kant¹⁷². Nauki o kulturze, jak je zbiorczo nazwał, miały uczyć ludzi „interpretować symbole, by odsłonić zamkniętą w nich treść”¹⁷³ i na powrót je ożywić. Niezbędne stało się zatem podejście rozumiejące.

167 B. Andrzejewski, *Wstęp*, [w:] E. Cassirer, *Symbol i Język*, tłum. i red. B. Andrzejewski, WNIF, Poznań 1995, s. 6.

168 Cassirer powołuje się na Kanta *Prolegomenę do wszelkiej przyszłej metafizyki, która będzie mogła wystąpić jako nauka* w E. Cassirer, *Logika nauk ...*, s. 105-106.

169 P. Parszutowicz, *Wstęp*, [w:] E. Cassirer, *Substancja i funkcja. Badania nad podstawowymi problemami krytyki poznania*, tłum. i wstępem opatrzył P. Parszutowicz, Wydawnictwo Marek Derewiecki, Kęty 2008, s. 9.

170 J. Sójka, *O koncepcji form symbolicznych Ernsta Cassirera*, Państwowe Wydawnictwo Naukowe, Warszawa 1988, s. 11.

171 A. Musioł, *Dwa projekty antynaturalizmu metodologicznego na przykładzie filozofii Heinricha Rickerta i Ernsta Cassirera*, „Ruch filozoficzny”, 71, nr 2, 2014.

172 E. Cassirer, *Pojęcia przyrodnicze i pojęcia kulturowe. Przyczynek do zagadnienia logiki nauk humanistycznych*, [w:] H. Buczyńska, *Cassirer*, Wiedza Powszechna, Warszawa, 1963, s. 120-121.

173 E. Cassirer, *Logika nauk ...*, s. 106.

Przekonanie o konieczności prowadzenia badań kulturoznawczych wynikało z poglądu, że człowiek doświadcza także poprzez wspomniane już obszary kultury, a kulturoznawstwo jest jedynym z możliwych sposobów pozyskiwania wiedzy o otaczającym świecie. Według Cassirera, aby zrozumieć język symboli wytworzony przez kulturę, niezbędne stało się podejście interpretujące, przy jednoczesnym zachowaniu analityczności¹⁷⁴. Postulując by w celu poznania świata odchodzić od stosowania jedynie metodologii nauk przyrodniczych oraz ścisłych, Cassirer sam zajął się krytyką kultury¹⁷⁵. Ogólna teoria form stanowiła rodzaj klucza do *morfologii ludzkiego ducha*. Poznanie logiki nauk o kulturze za cel miało ukazanie harmonizujących ludzką rzeczywistość procesów duchowych.

Za *Krytyką rozumu* Cassirer przyjął występowanie uniwersalnej formy wiedzy, która odsłania się poprzez refleksję filozoficzną, nie jednak nad jej zawartością¹⁷⁶, ale funkcją. W koncepcji tego myśliciela obraz ludzkiego świata tworzą aprioryczne struktury poznania. Jest on syntezy w poznaniu naukowym, od którego wyszedł Cassirer, ale także w miecie, religii, sztuce i języku. Formy te należą do tej samej „rodziny logicznej”¹⁷⁷, bo choć mają różną od siebie strukturę logiczną, to funkcjonalnie odgrywają tę samą rolę poznawczą¹⁷⁸. Cechą wspólną jest też wspomniana aprioryczność tych aktywności. Aprioryzm poznania stanowi z resztą klucz do wszelkich podejmowanych przez Cassirera rozważań. Jest on ukształtowany przez neokantyzm w nurcie marburskim. Już Hermann Cohen zdynamizował bowiem pojęcie *a priori*, dla Kanta będące statycznym¹⁷⁹. Paul Natorp zaś i Cassirer wypełnili formę *poietic* symboliczną zawartością¹⁸⁰.

Wyróżnione przez Cassirera formy symboliczne przedstawiają specyficzne tylko dla nich struktury świata, dlatego podlegają swoistym ich miarom. W związku z tym, zamiast powoływania się na „stałe »metafizyczne« kategorie”¹⁸¹ jako punkt wyjścia¹⁸²

174 *Ibidem*.

175 Na pomysł rozwinięcie swoich zainteresowań o filozofię kultury wpadł podczas pisanie „Substancji i funkcji”. H. Buczyńska, *Cassirer...*, s. 12.

176 E. Cassirer, *The Problem of Knowledge: Philosophy, Science, and History Since Hegel*, Yale University Press, New Heaven 1969, s. 14-15.

177 E. Cassirer, *Pojęcia przyrodnicze i pojęcia kulturowe. Przyczynek do zagadnienia logiki nauk humanistycznych*, [w:] H. Buczyńska, *Cassirer...*, s. 122.

178 H. Buczyńska, *Cassirer...*, s. 41.

179 S. Luft, *The space of culture: towards a neo-Kantian philosophy of culture (Cohen, Natorp, and Cassirer)*, Oxford University Press, Oxford 2015, s. 144.

180 *Ibidem*, s. 182.

181 E. Cassirer, *Filozofia form symbolicznych. Część pierwsza: Język*, red. P. Parszutowicz, Wydawnictwo Marek Derewiecki, Kęty 2018, s. 156.

w ich charakterystyce, filozof postulował wyprowadzenie praw formowania. Dostrzegł ogólność „jednej źródłowej zasady (z której, *przyp. aut.*) rozwija się konkretny zestaw poszczególnych form”¹⁸³. Wobec skomplikowanych relacji zachodzących w tych związkach znaczeniowych szukał fundamentalnej zasady obowiązującej wszystkie¹⁸⁴. W tej złożoności uwidaczniał się, szczególny charakter każdej z form.

Twierdzenie, że język jest „miejszem przedmiotowego tworzenia”¹⁸⁵ pozwala przeanalizować formowanie obrazu świata na drodze językowej. W jego symbolicznej formie ukazuje się rzeczywistość. Odbywa się to wszelako nie na zasadzie przedstawienia kopii w umyśle, lecz poprzez akt twórczy. Obraz ten zależy od natury podmiotu i jego subiektywnego postrzegania, można zatem odnaleźć w nim rodzaj aktu, nie zaś wzorzec przedmiotu. Czynność „zarysowywania wzorca” mówi nam tyleż o człowieku co i o przedmiocie jego zainteresowania. Język stanowi w tym rozumieniu Leibnizowskie „żywe zwierciadło”, dzięki któremu w umyśle tworzy się „duchowe odzwierciedlenie uniwersum”¹⁸⁶. Widocznie jest tu w myśli Cassirera nawiązanie do koncepcji kantowskiej syntezy. Wszak aby poznać przedmiot, podmiot posługuje się aktem przedstawienia. Czynność ta niezbędna jest w oglądzie świata dokonywanym przez język, gdyż ten odzwierciedla rzeczywistość nie jako narzędzie, ale organ myśli. Kształtując myślenie podmiotu, jest niejako też od niego zewnętrzny. Jako „środek tworzenia”¹⁸⁷ język buduje rzeczywistość przedmiotową.

Tę siłę tkwiącą w języku należy badać nie tylko poprzez spekulatywne rozważania. Cassirer dostrzega potrzebę korzystania z dorobku nauk empirycznych takich jak: lingwistyka, językoznawstwo porównawcze, psychologia, psychopatologia, biologia¹⁸⁸ podczas rozwiązywania problemów z zakresu filozofii języka. Wybór tych nauk, które obecnie jednoznacznie kojarzą się z kognitywistką jest dość wymowny. W tekście z 1931 „Język i budowa świata przedmiotowego” skierowanym do przedstawicieli psychologii, myśliciel proponował współpracę filozofii i wybranych wyżej dyscyplin skupionych na badaniu języka. Wzajemną korzyść miało przynieść postulowane przez

182 Za spuściznę neokantowskiej szkoły z Marburga, w myśli Cassirera, można uznać także odrzucenie tradycyjnej metafizyki za H. Buczyńską, *Cassirer...*, s. 25.

183 E. Cassirer, *Filozofia form symbolicznych. Część pierwsza* ..., s. 65.

184 *Ibidem*, s. 67.

185 B. Andrzejewski, *Wstęp*, [w:] *Symbol i język...*, s. 9.

186 E. Cassirer, *Język i budowa świata przedmiotowego*, [w:] *Symbol i język...*, s. 77.

187 *Ibidem*, s. 81.

188 *Ibidem*, s. 89.

niego wychodzenie poza granice tradycyjnych punktów widzenia, przy jednoczesnym ukierunkowaniu na cel, czyli rozwiązywanie problemów natury językowej.

Odejście od bezpośredniego doświadczenia podmiotu do ujmowania rzeczywistości poprzez czyste formy, stało się możliwe dzięki pośrednictwu symbolu. Jego struktura sprawia, że duchowe treści są wiązane wewnętrznie z danymi zmysłowymi znakami¹⁸⁹. Dostarczane przez zmysły wrażenia podlegają „swobodnej czynności wyrażania”¹⁹⁰ w ludzkiej świadomości. Na świat obiektywny nakłada się więc ten wytworzony z treści wrażeń i odczuć. Dopiero uwzględniając język emocji, możemy w pełni ująć człowieczeństwo. Aczkolwiek, mimo że część zwierząt, potrafi również tworzyć wypowiedzi emocjonalne, to tylko ludzie, jak twierdził filozof, potrafią wyjść poza język uczuć i posługują się też językiem twierdzeń¹⁹¹. Według Cassirera przekaz poprzez sygnały jest typowy dla zwierząt, natomiast człowiek wykracza poza ten „fizyczny świat bytu”¹⁹², gdy ujmuje treści w symbole. Tym sposobem wykorzystuje aprioryczne struktury w poznaniu, a wyobraźnię praktyczną uzupełnia o wyobraźnię symboliczną. Ludzi w tej perspektywie odróżnia od innych zwierząt nie posługiwanie się wąsko pojętym rozumem, lecz proces myślowy polegający na ujmowaniu rzeczywistości w formy symboliczne. Do tego zaś nieodzowne jest myślenie za pomocą abstrakcji.

Podstawę odróżnienia zwierzęcej sygnalizacji i ludzkiego języka, jak to przedstawia Cassirer, stanowić miało powstanie u człowieka odgłosu znaczącego¹⁹³. W tej koncepcji pierwotne dźwięki afektywne zostają uzupełnione przez wyrażające to, co obiektywne. „Dopiero tam, gdzie kończy się bezpośredni stosunek do wrażenia zmysłowego i do zmysłowego afektu”¹⁹⁴ zaczynał się według filozofa język. Nawiązując do powyższego Cassirer uważał, że zwierzęta używają zaledwie dźwięków i ich powtórzeń, a to uniemożliwiało uznanie tego za mowę. Dysponując wiedzą swoich czasów, jako że odwoływał się do badań W. Kohlera z 1921 roku¹⁹⁵, filozof nie mógł oczywiście przewidzieć tego, co wiadomo obecnie na temat komunikacji niektórych gatunków.

189 E. Cassirer, *Pojęcie formy symbolicznej w budowie nauk humanistycznych*, [w:] *Symbol i Język...*, s. 16.

190 *Ibidem*.

191 E. Cassirer, *Esej o człowieku ...*, s. 87-88.

192 *Ibidem*, s. 58

193 Cassirer nie neguje oczywiście tego, że języki, jak je nazywał, bardziej rozwinięte, czasami też bezpośrednio *malują dźwiękiem*, co dostrzegamy choćby w wyrażeniach onomatopeicznych, zob. E. Cassirer, *Filozofia form symbolicznych. Część pierwsza ...*, s. 171-172.

194 *Ibidem*, s. 170.

195 *Ibidem*, s. 171.

Wyniki badań psychozoologicznych prowadzonych od drugiej połowy XX wieku wskazują na bardziej zaawansowane możliwości językowe zwierząt, niż zakładał to Cassirer. Kruki, na przykład, znane są badaczom jako istoty niezwykle sprawne nie tylko w posługiwaniu się narzędziami, czy rozpoznawaniu twarzy, ale też w przekazywaniu informacji¹⁹⁶. Zarówno one, jak i wrony posiadają zdolności językowe, spory zasób „krakanego” słownictwa, tyle że nie posługują się nim zgodnie z ludzką gramatyką¹⁹⁷. Szympansy natomiast, chociaż z uwagi na nieodpowiednio rozwinięty tor głosowy nie są w stanie mówić, jednak potrafią wyuczyć się znaków języka migowego¹⁹⁸ podobnie jak niektóre inne naczelne¹⁹⁹. Używają one znaków jednak nie w ramach systemu i zgodnie z regułami, lecz w przypadkowym układzie. Według stanu obecnych badań niepewne wydaje się czy jest możliwe przekroczenie przez zwierzęta progu dwupoziomowości, czyli użycia składni²⁰⁰. W tej kwestii zdania są jednak podzielone²⁰¹.

Kwestia dwustopniowości, z perspektywy językoznawstwa, jest cechą definiującą język i pozwalającą odróżniać go na tej podstawie od innych systemów komunikacji²⁰². Takie języki sztuczne, jak na przykład języki programowania, nie posiadają pierwszego poziomu, czyli fonologicznego, skoro są projektami pasygraficznymi²⁰³. Stawia to interlingwistów argumentujących za włączaniem tychże języków do typologii języków czy filozofów orędujących za ich językowością, na pozycji analogicznej do zajmowanej przez wspomnianych psychozoologów zajmujących się językiem zwierząt. Rozpoznawalne, jak już wspomniano, jest natomiast w badaniach zoopsychologicznych

196 J. Marzluff, T. Angell, *Gifts of the Crow. How Perception, Emotion, and Thought Allow Smart Birds to Behave Like Humans*, Simon and Schuster, New York 2013.

197 J. Marzluff, T. Angell, *In the Company of Crows and Ravens*, Yale University, London 2005, s. 132.

198 Na przykład znaków języka naturalnego Głuchych z obszaru USA i części Kanady - American Sign Language, R. A. Gardner, B. T. Gardner, *Teaching sign language to a chimpanzee*, „Science”, 165, 1969, s. 664-672.

199 Więcej na ten temat można znaleźć w publikacji badań P. Peterson, *Koko's story*, Scholastic Press, New York 1987; H. L. Miles, *The cognitive foundation for reference in a signing orangutan*, [w:] „Language” and Intelligence in Monkeys and Apes: Comparative Developmental Perspective, red. S. T. Parker, K. Gibson, The MIT Press, Cambridge 1990.

200 I. Kurcz, P. Tomaszewski, *Powstanie i ewolucja języka*, [w:] *Język jako przedmiot badań psychologicznych. Psycholingwistyka ogólna i neurolingwistyka*, red. I. Kurcz, H. Okuniewska, Wydawnictwo Szkoły Wyższej Psychologii Społecznej „Academica”, Warszawa 2011, s. 49.

201 B. Rode, *Czy zwierzęta mają język*, „Zoophilologica. Polish Journal of Animal Studies”, nr 3, 2017.

202 M. Izert, E. Pachocińska, *Wstęp do językoznawstwa ogólnego*, ZG UW, Warszawa 1998, s. 27-28.

203 E. Kulczycki, *Założenia filozoficznych języków apriorycznych*, [w:] *My w wieży Babel. Między przekleństwem a błogosławieństwem*, red. M. Cieszkowski, J. Szczepaniak, Wydawnictwo Uniwersytetu Kazimierza Wielkiego, Bydgoszcz 2011, s. 43-45.

posługiwanie się przez niektóre gatunki znaczeniem, co kwestionuje sposób rozróżnienia komunikacji zwierząt i ludzi – tej, której dokonał Cassirer.

Wychodząc od kwestii znaczenia, Cassirer skonstruował *continuum* faz dojrzałości obejmującej poszczególne języki ludzkie: poczynając od stadium ekspresji mimicznej, przez analogiczną, a skończywszy na symbolicznej²⁰⁴. Tym samym dla celów analizy filozof wyodrębnił zmysłowy, pogładowy i pojęciowy szczebel rozwoju języka²⁰⁵. Wspomniany podział przedstawiał według niego funkcjonalne prawo budowy tej formy symbolicznej, a tworząc go, nawiązał do trychotomii: indeks, ikona i symbol autorstwa Charlesa S. Pierce²⁰⁶, przyjmując, że „sygnał jest częścią fizykalnego świata bytu, symbol jest częścią ludzkiego świata treści”²⁰⁷. W myśl koncepcji Cassirera tylko człowiek posiada wyobraźnię symboliczną i ma potrzebę tworzyć.

Języki w pierwszym stadium rozwoju charakteryzują się ograniczeniem do ekspresji mimicznej, czyli indeksu. Gest lub dźwięk nie przedstawiają, a jedynie wskazują. Stopniowanie dźwięków w językach tonalnych jest już przykładem na osiągnięcie kolejnego poziomu, czyli ekspresji analogicznej. Imitację zastępuje w nich stosunek dźwięku i oznaczającej go treści²⁰⁸. W językach wywodzących się z ekspresji imitatywnej i analogicznej może nastąpić odejście od funkcji oznaczania i przejście do „ogólnej i powszechnie ważnej funkcji *znaczenia*”²⁰⁹. W ich rozwoju wyraża się w ten sposób odejście od indywidualnych znaków i dążenie do ujmowania znaczenia w symbolu. Różnice między poszczególnymi językami, miałyby wynikać z tego, że znajdują się one na różnych stadiach dojrzewania do formy, a samą formą symboliczną.

Odchodzenie od zmysłowości, według Cassirera, skutkować ma powstaniem w języku „nowej i głębszej duchowej treści”²¹⁰. Skądinąd nawet język najbardziej abstrakcyjny zakorzeniony jest w ekspresji zmysłowej, a dopiero poprzez ujęcie wrażeń w czasie, przestrzeni i liczbie może stawać się formą przedstawienia. Przez te naoczności, jak je za Kantem nazywa Cassirer, buduje się pojęcia językowe²¹¹, stanowiące „rusztowanie obiektywnej naoczności”²¹². Przestrzeń w języku wyrażają

204 E. Cassirer, *Filozofia form symbolicznych. Część pierwsza* ..., s. 171.

205 B. Andrzejewski, *Animal symbolicum. Ewolucja neokantyzmu Ernsta Cassirera*, Wydawnictwo Naukowe Uniwersytetu i. Adama Mickiewicza w Poznaniu, Poznań 1980, s. 167.

206 H. Buczyńska-Garewicz, *Semiotyka Peirce'a*, ZSL UW Warszawa 1994, s. 41.

207 E. Cassirer, *Esej o człowieku*..., s. 91.

208 B. Andrzejewski, *Animal symbolicum*..., s. 176.

209 E. Cassirer, *Filozofia form symbolicznych. Część pierwsza* ..., s. 180.

210 *Ibidem*.

211 *Ibidem*, s. 182.

212 *Ibidem*, s. 242.

partykuły związane z położeniem, tak jak na przykład w słowie *vor|stelllen*, czy analogicznie w słowie „przed|stawiać”. W ten sposób w nawet najbardziej metaforycznych językach tkwi ziarno zmysłowości. Określanie położenia bowiem jest punktem wyjścia dla obiektywizacji. Filozof powołuje się w tej kwestii na „ogólną krytykę poznania (która, *przyp. aut.*) uczy, że ustalanie pozycji w przestrzeni i przestrzenne różnicowanie stanowią konieczny warunek wstępny do aktu obiektywizowania w ogóle, dla »odnoszenia się przedstawienia do przedmiotu«”²¹³.

W obszarze naoczności czasowej występuje zaś myślenie wymagające więcej analityczności i syntezy, niż w naoczności przestrzennej, dlatego właśnie ona „wznosi na całkiem inny poziom idealności”²¹⁴. W przeciwieństwie do miejsc w przestrzeni nie ma możliwości by w czasie współwystępowały odległe od siebie momenty. W związku z tym, że czas ma kierunek, nie da się odwrócić „sensu jego biegu”²¹⁵. W zależności od poziomu świadomości czasu Cassirer rozróżnia w języku trzy etapy: od poziomu przeciwnego „teraz” i „nie teraz”, przez wyodrębnianie czasowych form (trwania i przeminięcia), po pojęcie porządku²¹⁶. Za najbardziej zaawansowane filozof uznaje wyrażenia językowe związane z pomiarem czasu.

Zwroty językowe oparte na mierzeniu czasu, według Cassirera, wykraczają poza formę języka, zatem proponuje je nawet ujmować w formie nauki²¹⁷. W ramach sztucznych systemów znaków sugeruje rozważać je jako problem refleksji naukowej. Jednocześnie filozof zaznacza, że język ma możliwość podołania tej sztuczności. Dowodem na pokonanie jej, według niego, jest już sama możliwość ujmowania liczb słowami. Sposób rozwiązania tego dylematu może skłaniać ku analogicznemu podejściu w kwestii języków skonstruowanych, w tym języków programowania. Skoro w języku tworzy się naoczność na tym poziomie, to można również podjąć próbę „intelektualnego zapanowania nad fenomenami”²¹⁸. Analizując językowość kodu źródłowego, a nie badając go jedynie od strony technicznej, możemy ująć programy komputerowe jako wytwór kultury już na językowym etapie.

Odchodząc coraz bardziej od uchwytnych form do rozumowych zasad w języku, podążamy za Cassirerem, od przedstawienia przestrzeni do przedstawienia liczby. Za

213 *Ibidem*, s. 187.

214 E. Cassirer, *Filozofia form symbolicznych. Część pierwsza ...*, s. 203.

215 *Ibidem*, s. 204.

216 *Ibidem*, s. 206.

217 *Ibidem*, s. 214.

218 *Ibidem*, s. 215.

pośrednictwem naoczności liczby oddalamy się coraz bardziej od opierania się w języku na wrażeniach zmysłowych. Najprostsze pojęcia liczbowe są oczywiście jeszcze ściśle związane ze światem zmysłowym, jak na przykład gesty liczenia. Szczytowym dążeniem jest jednak czyste pojęcie liczby. To ostateczne „uwolnienie od podstaw naoczności i intuitywnego przedstawienia rzeczy”²¹⁹ jak je nazywa Cassirer, jest jednak możliwe do osiągnięcia tylko w myśleniu matematycznym. Również i tu można doszukiwać się inspiracji do dalszych rozważań o językach programowania, które przecież z założenia muszą dawać możliwość realizacji obliczeń²²⁰.

Dla badań nad językiem jako formie symbolicznej w kontekście kodu źródłowego niezwykle istotne wydaje się spostrzeżenie, które Cassirer czyni na temat nieuniknionej opozycji i napięcia między symbolami językowymi i czysto intelektualnym – mając tu na myśli język matematyki²²¹. Naukowe teorie matematyczne, na przykład, aksjomaty geometrii euklidesowej Hilberta, na które powołuje się Cassirer, zapisane są za pomocą czystych abstrakcji. Symboliczną formę języka zastępuje tu czyste pojęcie symbolu w nauce²²². Poszczególne części treści tych teorii nie odnoszą już do rzeczywistości, lecz ukazują jej prawidłowości. Sens elementów pojawia się już tylko w ich wzajemnej relacji, bo czyste znaki semantyczne niczego nie przedstawiają, a jedynie przyporządkowują²²³. W części języków programowania, szczególnie tych wyższego poziomu, można dostrzec nie tylko ujęcie znaku jako abstrakcji, ale również wyrażanie w nim pewnego znaczenia i odniesienia do rzeczywistości. Można więc powiedzieć, że język kodu źródłowego sytuuje się gdzieś na styku językowości i naukowości.

Myślenie relacyjne w języku, a nie w nauce, jest związane ze światem konkretnych przedmiotów i procesów, przy czym pierwsze podziały wynikają z uświadomienia sobie przez człowieka, że „ja” jest odrębne od „ty”²²⁴. Sfera obiektywności wyłania się z subiektywnej rzeczywistości poprzez naoczności przestrzeni, czasu i liczby. Ujawnia się i oddziałuje w niej wewnętrzny świat. „Ja” nie jest jedynie dyskursywnym pojęciem, ale niesie z sobą duchową treść. Pojawia się tym sposobem kluczowy dla

219 *Ibidem*, s. 217.

220 Podstawą do zakwestionowania statusu HTML, jako języka programowania jest przecież właśnie brak wspomnianej możliwości.

221 *Ibidem*, s. 217.

222 H. Lang, *Język i nieświadomość...*, s. 227.

223 „Das Zeichen im Sinne des reinen Bedeutungszeichens drückt nichts aus und stellt nichts dar – es ist Zeichen im Sinne einer bloss abstrakten Zuordnung” w E. Cassirer, *Zeitschrift für Ästhetik und allgemeine Kunstwissenschaft*, Verlag von Ferdinand Enke, Stuttgart 1927, s. 300, <https://digi.ub.uni-heidelberg.de/diglit/zaak1927/0315> (dostęp dnia 05.03.2022).

224 E. Cassirer, *Filozofia form symbolicznych. Część pierwsza* ..., s. 233.

Cassirowskiego ujęcia języka przymus ekspresji. Już na najniższym poziomie można dostrzec tę konieczność w zaimkach osobowych ukierunkowanych na ciało, bo przecież powstanie ich wynika z intuicji samego siebie²²⁵. Pojęcie bytu osobowego w języku wiedzie od przedstawień rzeczownikowych i czasownikowych, przez zaimki osobowe i dzierżawcze, po przedstawienie „ja” pozbawione konkretnej treści. W zależności od „odległości” w wypowiedzi od „ja”, rzeczywistość znajduje się na różnych stopniach obiektywności. W miarę rozwoju języka następuje odejście od ekspresji poszczególnych rzeczy w czystą ekspresję relacji, a tym samym różnorodności treści uzyskujemy jednorodność formy²²⁶. Ewolucje języka Cassirer widzi bowiem w zmierzaniu ku określoności.

Język w filozofii form symbolicznych stanowi ekspresję myślenia pojęciowego. Tworzona w nim abstrakcja ma jednak swój początek w strukturze. Treści muszą być wprawdzie jakoś określone, wyodrębnione, a znaczenia słów ustalone. Ze strumienia myśli, dzięki aktowi ustanowienia i różnicowania, czyli aktu logicznego, dochodzi do przekształcenia wrażeń w przedstawienia²²⁷. Język staje się „wehikułem dla »rozpoznania w pojęciu« (*Rekognition im Begriff*)”²²⁸, niezbędnym do zapanowania nad dostępnym ludziom materiałem wrażeniowym²²⁹. Podstawową formę myślenia stanowi forma obiektywizowania, dlatego człowiek przekształca świat zmiennych wrażeń w jego utrwalone przedstawienie. Tendencja języka zmierza od ujęcia indywidualizującego do grupującego. Odwołując się do porównania języków „ludów pierwotnych” i tych z rozwiniętą kulturą językową, Cassirer wskazał na różniący je sposób budowania pojęć. Dla pierwszych typowe jest przedstawianie kwalifikujące, czyli tworzone specjalistycznych nazw w oparciu o doświadczenie zmysłowe, na przykład, nazywanie konkretnych odmian palmy, bez tworzenia pojęcia jej jako gatunku. Natomiast dla drugich charakterystyczne jest już generalizowanie, zatem budowanie ogólnych pojęć.

Równie istotnym elementem, jak pojęcie, wewnętrznej formy są klasy. Podobnie, na podstawie *continuum* rozciągającego się pomiędzy konkretyzacją a abstrakcyjnością

225 *Ibidem*, s. 245.

226 *Ibidem*, s. 260.

227 *Ibidem*, s. 279.

228 E. Cassirer, *Philosophie der symbolischen Formen, Dritter Teil – Phänomenologie der Erkenntnis*, Berlin 1929, s. 27. za B. Andrzejewski, *Animal symbolicum*. ..., s. 160.

229 *Ibidem*.

można prześledzić miejsce danego języka na „duchowej skali”²³⁰. Najniżej Cassirer umieszcza te języki, w których przyporządkowanie odbywa się na podstawie zmysłowości, czyli naocznych cech takich, jak na przykład kształt. Dalej plasują się języki stosujące klasyfikację według pośredniej zasady obrazowego ucieleśniania poprzez, na przykład, ocenianie odległości, liczby, czy wielkości. Przyporządkowywanie natomiast według rodzaju stanowi już o językowej wyobraźni. Wynika to z kwestii, że klasa ta ma mityczne i estetyczne źródło oraz „jako duchowa forma stoi na granicy między *mythosem* i *logosem*, a zatem pośredniczy między teoretycznym i estetycznym ujęciem świata”²³¹.

W ujmowaniu formy języka w pojęcia i klasy unaocznia się, że nie jest to proces jedynie *obiektywny*, ale ma w nim udział wewnętrzne nastawienie podmiotu. Jak stwierdza Cassirer „przyporządkowanie to nigdy nie jest zdeterminowane tylko aktami postrzegania lub sądenia, lecz zawsze jednocześnie aktami afektywnymi i aktami woli, aktami wewnętrznego nastawienia”²³². Akty afektywne i woli wpływają na postrzeganie i sądenie. Język jako forma symboliczna nie podąża biernie za światem wrażeń czy przedstawień. Podejmuje się za jego pomocą działania ukierunkowane czynionymi w nim rozróżnieniami i przyjętymi wyborami²³³. Obiektywna naoczność języka jest wynikiem przenikania się „teoretycznych niuansów znaczeniowych i afektywnych niuansów wartości (*tłum. własne*)”²³⁴. Nie istnieje taki moment, w którym w świadomości dałoby się odizolować odczucia i postrzeganie od intelektualnych funkcji. Czysta materia, w perspektywie języka, jest abstrakcją związaną nierozdzielnie z jej formą²³⁵. Chociaż dla celów analizy krytycznej Cassirer wyodrębnia poszczególne fazy, które zawierają się finalnie w krytyce poznania – sferę doznań zmysłowych, naoczności, myślenia pojęciowego, sądu logicznego – to trzeba pamiętać, że każdy z tych motywów „zawiera już w sobie wszystkie jego (języka, *przyp. aut.*) formy i specyficzny ogół tych form”²³⁶.

230 E. Cassirer, *Filozofia form symbolicznych. Część pierwsza...*, s. 295.

231 *Ibidem*, s. 298.

232 *Ibidem*, s. 302.

233 *Ibidem*.

234 E. Cassirer, *Critical Idealism as a Philosophy of Culture*, [w:]: *Symbol, Myth and Culture: essays and lectures of Ernst Cassirer, 1935-1945*, red. D. Verene, Yale University Press, New Heaven 1979, s. 90.

235 E. Cassirer, *Filozofia form symbolicznych. Część pierwsza ...*, s. 308.

236 *Ibidem*, s. 304.

Formy symboliczne są ze sobą powiązane stanowiąc fazy duchowego kształtowania świata. W okresie pracy nad „Filozofią form symbolicznych” Cassirer żywił przekonanie, że występują one po sobie i wedle negatywnej fundacji każda kolejna musiałaby przewyższyć poprzednią. Postęp wyrażać się miał zatem poprzez modyfikację, ale i zanegowanie, charakterystycznych dla wcześniejszego stadium elementów duchowych²³⁷. W okresie późniejszym, podczas pisania „Mitu państwa”, myśliciel dopuścił jednakże możliwość współwystępowania form, z założeniem dominacji którejs z nich²³⁸. Przy czym język miał status dość szczególny, bez względu na wariant Cassirerowskiej teorii, ze względu na to, że zawsze działa w ramach innych form. Ten brak autonomiczności wynika z niemożliwości powstania czysto językowej postaci wyrazu czy przedstawiania, z jedynie „lingwistyczną przestrzenią artykulacji energii duchowych”²³⁹. Równocześnie jest twórcywem innych form.

Język jednoczy pozostałe funkcje z uwagi na to, że dzięki niemu dochodzi do ich ekspresji²⁴⁰. Jest nośnikiem, w którym znajduje się klucz do świata kultury. Nie wyznacza on jednak granic poznania, te tworzy bowiem forma symboliczna. Granice poznania wypływają z czystego rozumu. U Cassirera ujawnia się zatem podwójne myślenie o języku: jako o narzędziu oznaczania i podstawowej formie. Bycie – zarówno sposobem wyrażania, jak i formą myślenia – sprawia, że staje się trudny w analizie. Jednakowoż takie podejście zawiera w sobie potencjał do prowadzenia rozważań nad kodem źródłowym i językami programowania, inaczej niż jedynie w ujęciu technicznym. Zwrot transcendentalizmu w kierunku filozofii kultury nakierowuje na poszukiwanie warunków możliwości poznania naszej kultury.

Wszystkie formy wyrażają aktywność ludzkiego ducha zmierzającą do samopoznania, różnią je zaś obiektywizacje. Język i mit rozwijają się równolegle ze świadomości podmiotu i formy te wzajemnie wewnątrznie się ożywiają²⁴¹. Jednocześnie sam mit jest zaledwie *podglebiem* dla języka, bo ten by spełniać swoje funkcje musi zrezygnować z bezpośredniej naoczności²⁴². Język coraz bardziej separuje

237 M. Bal-Nowak, *Mit jako forma...*, s. 63.

238 *Ibidem*.

239 A. Kowalski, *Mit a sztuka w Ernsta Cassirera filozofii form symbolicznych*, „Filo-Sofija”, 3, nr 3, 2003, s. 121.

240 M. Bal-Nowak, *Mit jako forma...*, s. 139.

241 E. Cassirer, *Język i mit. Przyczynek do zagadnienia imion bogów*, Wydawnictwo Marek Derewiecki, Kęty 2021, s. 99.

242 *Ibidem*, s. 100.

się od mitu w kierunku czystej reprezentacji. Odchodzi coraz bardziej od treści emocjonalnych w stronę języka nauki. Właśnie tak funkcjonuje, gdy staje się chociażby językiem formalnym²⁴³. Cassirer wskazuje, że ma jednak on szansę na regenerację dzięki estetycznemu wyzwoleniu poprzez ekspresję²⁴⁴.

Status języków programowania nie jest jednoznaczny, bo chociaż są tworem technicznym i przez to naukowym, to można doszukać się ich wymiaru ekspresji, ujawniania i konstytuowania. Nie da się rozdzielić nauki od jej języka. Na tej podstawie można uznać za bezcelowe próby wyzbycia się w poznaniu naukowym językowości. W niej zaś w różnym stopniu tkwi emocjonalność podmiotu i przez nią też ujawnia się brak możliwości całkowitego zerwania z naocznością. Można zatem postawić tezę, że ekspresja w języku pojawiać się zatem będzie również w tekście kodu źródłowego programów komputerowych. U Cassirera, rozwój znaków pojęciowych nauki wychodzi od znaków słownych, lecz polega na odejściu od związku słowa z przedmiotem a staniu się wyrazem samej relacji i porządku. Dzieje się tak za sprawą symbolu. Nie przynależy on czy do świadomości, czy do przedmiotu, toteż „ani do sfery »immanencji« ani do »transcendencji«, nie znajduje się tak »po tej stronie« (*diesseits*) jak i »po tamtej stronie« (*jenseits*) – przeciwnie, zasadnicza jego rola polega na przewyżczeniu takiej dualistycznej teorii”²⁴⁵. W swoim symbolizmie filozof daje więc odpór dualizmowi metafizycznemu.

Na przestrzeni całej swojej filozoficznej aktywności Cassirer ujmował filozofię jako metodologiczną analizę poznania. Będąc zwolennikiem aprioryzmu, nadawał odmienną od Heideggera wagę poznaniu ponadzjawiskowemu. Gdy fryburczyk uzależniał poznanie od czynnika fenomenalnego²⁴⁶, on od ponadnaocznego. Miało to, ma się rozumieć, przełożenie na podejście do badań nad językiem. By zgłębić tę sferę kultury trzeba, za koncepcją Cassirera, poddać ją metodologicznej analizie opisowej²⁴⁷. Według Heideggera, choć takie stechnicyzowanie podejścia do języka mogło być pożyteczne, pomijało jednak istotę języka, którą uchwycić można dopiero na drodze doświadczenia

243 W znaczeniu języka sztucznego, powstałego w celu porozumiewania się w dziedzinie naukowej lub technicznej, np. język logiki, matematyki, informatyki czy programowania komputerów.

244 E. Cassirer, *Język i mit...*, s. 100.

245 B. Andrzejewski, *Animal symbolicum...*, s. 161.

246 A. Musioł, *Neokantyzm epistemologiczny i ontologiczny Ernst Cassirer – Martin Heidegger*, Wydawnictwo Uniwersytetu Śląskiego, Katowice 2020, s. 273.

247 E. Cassirer, *Esej o człowieku...*, s. 155.

z nim²⁴⁸. Ricœur natomiast, mimo że tworzył w ramach tej tradycji hermeneutycznej, nie stał się wiernym i bezkrytycznym kontynuatorem w powyżej kwestii. Język jako dyskurs podlegać miał bowiem według niego analizie. Na tej podstawie można zatem w myśli zarówno Cassirera, jak i Ricœur odnaleźć przyzwolenie na metodologiczne potraktowanie języka, w tym języka kodu źródłowego.

W badaniach nad językiem kodu źródłowego warto odwołać się do zaproponowanej przez Cassirera koncepcji języka jako formy symbolicznej, śledząc to, jak syntezyzuje ona doświadczenie programistów. Inspirowana jego myślą analiza pozwala ukazać kod źródłowy programów komputerowych jako zjawisko kulturowe, osadzone w ramach filozofii form symbolicznych. Miejsce, jakie zajmuje według tego myśliciela język, pomaga bowiem ulokować problem języków programowania w świecie kultury.

1.4. Spojrzenie na język Ricœura i Cassirera

Chociaż praca nie ma charakteru porównawczego, zasadnym wydaje się dokonać zestawienia kluczowych zagadnień, które pojawiły się w rozważaniach Ricœura i Cassirera na temat języka. Oba podejścia wykraczają poza traktowanie go wyłącznie jako narzędzia komunikacji, ukazując jego twórczy charakter. Tym samym wpisują się w nurt filozofii kontynentalnej, rozumianej – za Taylorem – jako rodzina teorii konstytutywno-ekspresywnych. W perspektywach tych uwydatnia się bowiem aktywny charakter języka. Ricœur ujął go w koncepcji zdarzenia językowego, Cassirer natomiast akcentując kulturotwórczą funkcję języka.

W myśli Ricœura dyskurs jest znaczącym zdarzeniem językowym. Filozof podkreślał tym samym wymiar czasowy odczytywania i powstawania narracji. Interpretacja tekstu polega bowiem na przyswojeniu znaczenia w aktualnym kontekście osoby. Uwzględniając działanie podmiotu i jego zmienność (*ipse*), narracja stała się także punktem wyjścia dla koncepcji tożsamości narracyjnej. Ta sytuuje się między „byciem tym samym” zakorzenionym w charakterze i „byciem sobą”, ujawnianym poprzez działanie. Tożsamość narracyjna stanowi zatem dynamiczną strukturę,

248 M. Heidegger, *W drodze do języka*, Wydawnictwo Baran i Suszczyński, Kraków 2000, s. 114.

tworzoną na podstawie różnych narracji o sobie. Wyrasta z umiejętności wysłowienia własnego doświadczenia i artykulacji treści istotnych dla podmiotu.

Hermeneutyka dyskursu otwiera możliwość interpretacji, w tym refleksji nad sobą, zaś filozofia form symbolicznych konstruowania rzeczywistości w formie języka. Ricœur bowiem konsekwentnie w swojej myśli hermeneutycznej akcentował rozumienie, natomiast Cassirer kładł nacisk na poznanie. W podejściu marburczyka kluczowa była funkcja symbolizacji i mediatyzacji doświadczenia za pomocą języka. Rzeczywistość nie jest doświadczana, ani wyrażana bezpośrednio, lecz poprzez różne formy symboliczne. Język, będąc organem myśli w ujęciu Cassirera, formuje ten obraz świata.

Myśliciele ci różnią się pod względem roli, jaką zajmuje język w strukturze ich refleksji filozoficznej. U Ricœura problematyka ta stanowi oś centralną jego hermeneutyki, gdyż to w języku dochodzi do interpretacji obejmującej wyjaśnianie i rozumienie. Przyswojenie ma wymiar ontologiczny – jest sposobem odnoszenia się do bytów i bycia. Dla Cassirera natomiast język stanowi jedną z form symbolicznych – obok mitu, sztuki, religii i nauki. Filozof podkreślał ich kulturotwórczy charakter, mając jednak na uwadze ich znaczenie epistemologiczne.

Wiążącą obie filozofie kwestię stanowi myślenie o ludzkiej potrzebie uczestniczenia w rzeczywistości. Hermeneutyce dyskursu przyświeca przekonanie o języku jako przestrzeni, w której ujawnia się człowiek. Działając w języku ma on możliwość wchodzić w relacje ze światem wewnętrznym oraz otoczeniem. Cassirer, w koncepcji *animal symbolicum*, również poszukuje dla człowieka drogi łączności z rzeczywistością, tyle tylko, że prowadzi ona przez różne formy symboliczne. Obie myśli łączy więc przekonanie, że język jest sposobem człowieka na wchodzenie w relację ze światem.

Twórczy charakter języka, widoczny staje się, gdy rozpatruje się człowieka działającego w języku. Zgodnie z Ricœurowską *hermeneutics of praxis* dyskurs jawi się jako efekt wytwarzania – dzieło. Wymiar działania wiąże się z tym, że jest ono świadectwem ludzkiego wysiłku, a jego odczytywanie warunkowane jest praktyką życia człowieka. Dzieło według Cassirera, natomiast jest nośnikiem wspólnej myśli. Obiektywizuje się ona w wytworach kultury. Obaj filozofowie podkreślają aktywną,

twórczą formę wspomnianego wchodzenia w relację z rzeczywistością za pomocą języka.

Komplementarność podejścia Cassirera i Ricœura tkwi w narzędziowości – w rozważaniach dotyczących kodu źródłowego programów komputerowych można wykorzystać zarówno koncepcję języka-dyskursu, jak i języka jako formy symboliczne. Tekst powstały przy użyciu języków programowania może zatem stanowić artefakt kulturowy i podlegać interpretacji. Należy jednak mieć na uwadze, że specyfika języków programowania nie pozwala na poprowadzenie tej interpretacji; w taki sposób, jak gdyby tekst powstał w języku naturalnym. Nie do zignorowania jest fakt, że są to języki skonstruowane, co wymaga osobnego omówienia.

1.5. Języki programowania w kontekście języków sztucznych

Zasadniczo w typologiach języków sztucznych języki programowania zajmują miejsce marginalne, a nawet zdarza się, że bywają pomijane. Zaniedbanie to wynika zapewne z postrzegania ich jedynie przez pryzmat języka matematyki. Jest on bowiem nieodłączną częścią języków wielu dyscyplin technicznych, w tym informatyki. Zastanawiające jest jednak, że interlingwiści tak chętnie zgadzają się na przyjęcie paradygmatu matematycznego, uznając pisanie programów za działalność stricte matematyczną²⁴⁹. Stanowisko to nie jest zapewne rezultatem ich przemyśleń nad naturą programu komputerowego, ale pochodną obserwacji, które czynią, wychodząc od porównania języków formalnych do naturalnych. Zdarza się nawet spotkać wśród tych badaczy z opinią, że języki programowania są systemami „które nie osiągają godności języka”²⁵⁰. Powodem poczynienia wzmiankowanej oceny miałoby być rzekome ograniczenie się języków programowania zaledwie do prostej i rygorystycznej składni oraz zarzucanie „pasożytowania na innych językach, jeśli chodzi o znaczenie nadawane ich pustym symbolom”²⁵¹. Redukuje się tym samym to, co wydarza się w kodzie

249 I. Bondecka-Krzykowska, *O związkach informatyki z matematyką*, „Filozofia Nauki”, 18, nr 1, 2010, s. 81.

250 U. Eco, *Poszukiwanie języka doskonałego*, Wydawnictwo Aletheia, Warszawa 2013, s. 334.

251 *Ibidem*, s. 334.

źródłowy raptem do interakcji dialogowej człowieka z maszyną²⁵², ujętej jako zaledwie operacje na symbolach.

Uwidadcznia się natychmiast parę skaz na powyższym podejściu. Pierwsza dotyczy zignorowania kwestii, programowania jako złożonego procesu: specyfikacji, tworzenia metody rozwiązania problemu, a dopiero na końcu zapisu w odpowiednim języku²⁵³. Kolejna ma związek z błędnym przedstawieniem kwestii technicznych poprzez zbytnie uproszczenie i pominięcie etapu kompilacji. Bezpośredniość w „dialogu” ukazana jest tak, jakby programowanie ograniczało się do tworzenia li tylko kodu maszynowego²⁵⁴. Kod napisany w języku programowania (inaczej w języku źródłowy) jest natomiast zazwyczaj dopiero tłumaczony na program komputera i w tej postaci może być wykonany. Dzieje się tak w większości przypadków, więc programiści nie muszą znać języków procesorów (nie podejmujemy tu kwestii innych niż języki programowania języków komputerowych). Ponadto lingwiści, którzy ujmują języki programowania jedynie jako operacje na symbolach, mylnie uznają, że uwzględniają one tylko obszar komunikacji człowieka z komputerem, lekceważąc przekaz (ujmując problem po ricœurowsku) i funkcje języka inne niż czyste znaczenie (patrzac z perspektywy Cassirera). Nie biorą zatem pod uwagę zachodzącej w języku komunikacji człowieka z innymi ludźmi oraz samym sobą, a także aspektu wyrażania w języku wartości.

Jako języki formalne, czyli systemy utworzone, by porozumiewać się w jakiejś dziedzinie naukowej lub technicznej²⁵⁵, języki programowania bywają przez niektórych językoznawców umiejscawiane niejako poza obrębem języków sztucznych. Jednocześnie jednak wraz z nimi przeciwstawiane są językom naturalnym²⁵⁶. Oddzielone w ten sposób zyskują poniekąd odrębny status, ograniczający wszak możliwość badań interlingwistycznych. Próbę przełamania tego impasu podjął cytowany wcześniej, o ironio, krytyk języków programowania, Umberto Eco. Jeśli bowiem za jego radą dostrzeżemy we wspomnianych językach formalnych o wąskim kręgu użycia derywaty XVII i XVIII wiecznych projektów filozoficznych języków apriorycznych, to możemy na tej podstawie zacząć traktować języki programowania

252 *Ibidem*, s. 335.

253 I. Bondecka-Krzykowska, *Z zagadnień ontologicznych informatyki*, Wydawnictwo Naukowe UAM, Poznań 2016, s. 48.

254 Przy czym języki asemblera są przecież nawet zdecydowanie mniej popularne niż języki wyższego poziomu i przedstawianie programowania jako pisania tylko kodu maszynowego jest błędem.

255 jak na przykład język logiki czy matematyki.

256 M. Izert, E. Pachocińska, *Wstęp do językoznawstwa ...*, s. 27.

jako języki sztuczne²⁵⁷. Ten semiolog i zarazem filozof wskazał zatem kierunek argumentacji, w której ciągłość myśli miałyby wiązać języki programowania z filozoficznymi językami apriorycznymi.

Włączenie języków programowania do obszaru zainteresowań interlingwistycznych na zasadzie spuścizny po językach filozoficznych jest interesującym ujęciem teoretycznym. Jednakże hipoteza derywacji staje się problematyczna do wykazania, jeśli prześledzić historię programowania. Projekt językowy Gottfrieda W. Leibniza, Georga Dalgarno czy Johna Wilkinsa nie były kontynuowane, rozwijane ani nie leżały u podstaw tworzenia pierwszych języków programowania. Powstanie programów w zależności od wybranego paradygmatu informatyki, wiąże się z rozwojem technologii, bądź rozwojem informatyki jako gałęzi matematyki. Niekiedy bowiem początki programowania łączy się z rozwojem przemysłu tkackiego na przełomie XVIII i XIX wieku i wykorzystywanymi w nim taśmami perforowanymi lub z powstaniem automatów-zabawek²⁵⁸. Jeśli zaś genezy dyscypliny informatyki dopatruje się w matematyce, bywa że uznaje się za pierwszą programistkę Adę Lovelace²⁵⁹. Stworzyła ona algorytm do obliczania Liczby Bernoulliego dla maszyny analitycznej Charlesa Babbage, chociaż programu nie dało się nawet przetestować²⁶⁰. Pierwszym wysokopoziomowym językiem programowania, przeznaczonym dla rzeczywiście funkcjonującego komputera, był Plankalkül Konrada Zuse²⁶¹, udało się go jednak zaimplementować dopiero w 2000 roku²⁶². Przedstawiciele paradygmatu technokratycznego i racjonalistycznego (matematycznego) w filozofii informatyki²⁶³, wciąż tkwią w sporze czy wiedza o programach ma charakter aprioryczny, czy aposterioryczny.

257 U. Eco, *Poszukiwanie języka...*, s. 16-17.

258 I. Bondecka-Krzykowska, *Historia obliczeń, Od rachunku na palcach do maszyny analitycznej*, Wydawnictwo Naukowe UAM, Poznań 2012, s. 212-214.

259 *Ibidem*, s. 228.

260 ponieważ mechanizm maszyny nie powstał za życia twórcy. „Komputer” udało się w końcu zbudować w Muzeum Nauki w Londynie: w 1991 część odpowiedzialną za obliczenia, natomiast w 2001 roku część odpowiedzialna za wydruk i wprowadzanie
<https://www.computerhistory.org/babbage/modernsequel/> (dostęp dnia 7.07.2022)

261 Urzeczywistniał pomysł Babbage’a sprzed ponad 120 lat, dzięki dostępnym już środkom technologicznym – wyłącznikom elektromechanicznym z przekaźnikiem. H. Kaufmann, *Dzieje komputerów*, Państwowe Wydawnictwo Naukowe, Warszawa 1980, s. 111.

262 R. Rojas, C. Göktekin, G. Friedland, *Plankalkül: The First High-Level Programming Language and its Implementation*, Institut für Informatik, Freie Universität Berlin 2020, Technical Report B-3/2000, <http://ftp.mi.fu-berlin.de/pub/reports/TR-B-00-03.pdf>, (dostęp 0.07.2022) s. 2.

263 według podziału Amnona Edena paradygmat: racjonalistyczny, technokratyczny i naukowy. A. Eden, *Three Paradigms of Computer Science*, „Minds and Machines”, 17, nr 2, 2007, s. 135-167.

Przemyślenia Eco i jego przywiązanie do postrzegania informatyki jako aplikacji logiki na bardzo dużą skalę²⁶⁴ można wiązać z nurtem matematycznym. Program jest w tym ujęciu, jak już było wspomniane, operowaniem na abstrakcyjnych symbolach, które mogą zmienić się w konkretne, gdy użyje się komputera²⁶⁵. W językach programowania można próbować dostrzegać spuściznę intelektualną języków apriorycznych, trudno jednak powiązać ich rozwój bezpośrednio z tymi pomysłami. Jeśliby miały być uznane za derywaty projektów filozoficznych, to są z nimi luźno związane, raczej na zasadzie wspólnej myśli, a nie pochodzenia czy kontynuacji.

Eco zaproponował włączenie języków sformalizowanych do kategorii języków sztucznych powodowany chęcią rozszerzania przedmiotu własnej dyscypliny, a zatem by umożliwić badania semiotyczne²⁶⁶. Krok ten stanowi kontynuację kierunku metasemiotycznego zaproponowanego przez Julię Kristevę. Wynikał on z jej przekonania o „izomorfizmie praktyk semiotycznych w odniesieniu do innych układów w naszym wszechświecie”²⁶⁷. Przypisanie przez Eco języków sformalizowanych do kategorii języków apriorycznych nie przedstawiało w tej perspektywie dla badacza większego dylematu, stanowiło wręcz udogodnienie. Skoro języki programowania miały powstać jako pochodne Leibniz’owskiego *Lingua Universalis*, dawałoby to asumpt do traktowania ich jako projektów lingwistycznych. Co ważniejsze, to aprioryczność ich założeń ontologicznych stanowiłaby o tym przyporządkowaniu. Warunek ten nie musi jednakowoż oznaczać równocześnie braku żadnych komponentów aposteriorycznych materiału językowego. Możliwa jest bowiem aprioryczność formy i częściowa aposterioryczność materiału, gdy twórca czerpie słownictwo z języków naturalnych.

W projektach apriorycznych organizuje się treści według pewnej wymyślonej *gramatyce idei*, a zatem poprzez arbitralną klasyfikację bytów we wszechświecie. Założenie to niesie za sobą pewne reperkusje, a mianowicie konstrukcja takich języków zawsze jest wynikiem decyzji ich twórcy, czyniąc z niego depozytariusza prawdy.

264 VLSAL (Very Large Scale Application of Logic) jest pojęciem zaproponowanym przez Dijkstra na określenie informatyki w paradygmacie matematycznym. I. Bondecka-Krzykowska, *Paradygmaty informatyki*, [w:] *Światy matematyki. Tworzenie czy odkrywanie*, red. I. Bondecka-Krzykowska, J. Pogonowski, Wydawnictwo Naukowe UAM, Poznań 2010, s. 121.

265 E. W. Dijkstra, *On the Cruelty of Really Teaching Computing Science*, Center for American History, University of Texas at Austin, Austin 1989, <https://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1036.PDF>, (dostęp dnia 07.07.2022)

266 U. Eco, *Teoria semiotyki*, Wydawnictwo Uniwersytetu Jagiellońskiego, 2009 Kraków, s. 11.

267 *Ibidem*, s. 32.

Ontologiczne²⁶⁸ założenia powstawania tych języków bazują na przekonaniu o możliwości odnalezienia zasady budowy świata²⁶⁹, bądź w wersji mniej radykalnej – pewnego jego wycinka (w przypadku, na przykład, języka nauki czy dziedziny naukowej). W tym paradygmacie teoretycznym zakłada się także potrzebę eliminację *idolum fori*, poprzez ograniczenie wszelkich językowych niejasności. Likwidacja nieprecyzyjności i błędów wiąże się jednak z olbrzymimi uproszczeniami rzeczywistości. Prostota jest również istotna w projektach aposteriorycznych, tyle że jej celem jest ułatwienie nauczenia się danego systemu przez użytkowników. Języki te zbudowane są bowiem na modelach języków naturalnych i równocześnie mają realizować założenie większej ich elastyczności.

Z połączenia wybranych aspektów założeń języków apriorycznych i aposteriorycznych mogą powstać jednak systemy, które przejmują najgorsze cechy obydwu koncepcji. *Volapuk* mógłby być tego przykładem, gdyż „nie będąc językiem filozoficznym, aspiruje do filozoficznej analizy pojęć, a zatem przyjmują wady języków filozoficznych, nie przyjmując ich logicznych zalet”²⁷⁰. Równocześnie silnie deformuje się w nim rdzenie wyrazów zaczerpnięte z języka naturalnego, utrudniając tym samym zapamiętanie słownictwa. Projekt ten mimo początkowo dużego sukcesu równie szybko stracił swoich entuzjastów na rzecz *esperanto*, jako systemu, który bardziej odpowiadał wymogom międzynarodowego języka pomocniczego. *Volapuk* zbyt oddalił się od znanych potencjalnym użytkownikom języków naturalnych, tak że powstał nawet anty-język *anti-volapuko*²⁷¹, a słowo „volapyk”²⁷² stało się synonimem „jazgotu” w języku duńskim. Nasuwa się myśl, że za system mieszany, ale wykorzystujący najlepsze cechy tych dwóch podjęć, można by uznać języki programowania wysokiego rzędu. Oczywiście w głównej mierze to ich technologiczne zastosowanie wpłynęło na ich popularność, lecz nie można zignorować kwestii, że ich konstrukcja odpowiada po prostu na potrzeby użytkowników – ułatwia rozumienie kodu.

Koncepcja języków mieszanych nie jest nowa. Wspomniany *wolapuk* (*Volapük*)²⁷³, czy powstały z niego *spokil*²⁷⁴ to projekty hybrydy powstałe pod koniec

268 Ontologia oznacza tu uporządkowanie kategoryjnego bytu, nie termin informatyczny dotyczący ontologii systemu informatycznego.

269 Bliskie temu jest przekonanie Wittgensteina o tym, że w zdaniu ma się odzwierciedlać logiczna forma rzeczywistości, do której należy dążyć.

270 U. Eco, *Poszukiwanie języka* ... s. 345.

271 M. Pei, *One language for the world*, Biblo & Tannen, New York 1958, s. 165.

272 <https://www.wordsense.eu/volapyk/> (dostęp dnia 20.05.2022).

273 U. Eco, *Poszukiwanie języka*..., s. 344.

XIX wieku. Ponadto do mieszanych języków według Jana van Steenberg'a należałyby chociażby takie języki jak: *klingoński*, *láadan*, *lojban*, *toki pona*²⁷⁵. *Volapuk* ma nawet kod w najnowszej wersji międzynarodowego standardu ISO 639-3, gdyż klasyfikacja ta, w przeciwieństwie do wcześniejszej obejmuje już języki skonstruowane²⁷⁶. Niemniej jednak, z całego bogactwa powstałych projektów językowych, których liczbę trudno nawet w przybliżeniu oszacować²⁷⁷, uwzględnia ona jedynie dwadzieścia trzy z nich, a przy tym tylko dwa języki aprioryczne – *lojban* i *balaibalan*²⁷⁸. Pierwszy z wymienionych opiera się na zasadach logiki formalnej, więc stanowi jakąś reprezentację języków programowania we wspomnianym międzynarodowym standardzie grupującym języki świata.

Oznaczanie i identyfikacja języków, a potem ich klasyfikacja stanowią oczywiście obszar zainteresowania językoznawców, lecz z punktu widzenia interlingwistyki jeszcze bardziej zajmującym zagadnieniem było poszukiwanie powszechnych stałych wszystkich języków. Nie tylko inspirowało to do prób rekonstrukcji języka Adamowego, ale również podjęcia rozważań nad tzw. głęboką gramatyką²⁷⁹. Ważny głos wiążący tę tematykę z językami programowania zabrał Eco. Według filozofa wyrazem aprioryczności tych systemów jest fakt, że „bazują nie na regułach rządzących powierzchniowymi strukturami języków naturalnych, ale w maksymalnym stopniu stanowią wyraz tej zakładanej głębokiej gramatyki wspólnej dla wszystkich języków”²⁸⁰. Byłaby ona gramatyką uniwersalną, składającą się na powszechne stałe cechy obecne we wszystkich językach²⁸¹. Punktem zwrotnym w rozważaniach Eco jest konstatacja, co miałoby pełnić funkcję głębokiej gramatyki.

274 Autor *Spokila* tworzył słownictwo według przygotowanego przez siebie (a nazywanego naturalnym) logicznego sposobu klasyfikacji pojęć i podkreślał aprioryczność tego języka. A. Nicolas, *Spokil, langue internationale; grammaire, exercices, les deux dictionnaires*, A. Maloine, Paris 1904, s. IV. <https://archive.org/details/spokillanguageint00nicogoog/page/n9/mode/2up?q=aprio> (dostęp 07.04.2022).

275 <http://steen.free.fr/classificatie.html> (dostęp dnia 12.05.2022).

276 <https://tiny.pl/96g53> (dostęp 06.04.2022).

277 Na okres do końca XX wieku szacuje się powstanie około 1000 projektów językowych za K. Wojan, *Języki sztuczne. Zapotrzebowanie społeczeństw czy fantazje jednostki*, Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2025, s. 18, natomiast według listy HOPL już samych języków programowania możemy doliczyć się, że powstało do dziś 8945 <https://hopl.info/> (dostęp dnia 12.05.2022).

278 <https://tiny.pl/96g53> (dostęp dnia 06.04.2022).

279 W rozumieniu gramatyki uniwersalnej, nie przyjęciu założeń gramatyki generatywnej Noama Chomsky'ego.

280 U. Eco, *Poszukiwanie języka...*, s. 334.

281 *Ibidem*, s. 337.

Filozof sądzi, że według twórców języków filozoficznych głęboką gramatykę, wspólną dla wszystkich języków stanowić miałyby prawa logiki²⁸².

W ramach psychologii rozwojowej ostatniemu²⁸³ stadium rozwoju poznawczego Piageta, zwanemu stadium operacji formalnych, odpowiada znajomość i zastosowanie logiki. Według stworzonej przez tego psychologa epistemologii genetycznej „mowa (...) stanowi niewątpliwie warunek niezbędny zakończenia procesu tworzenia się struktur logiczno-matematycznych, nie może być jednak dla nich warunkiem wystarczającym”²⁸⁴, a źródeł logiki należy szukać we wcześniejszej od mowy logice koordynacji czynności. Rozwój mowy ułatwia bowiem rozwój poznawczy²⁸⁵.

Rozumowanie na poziomie operacji formalnych²⁸⁶ oznaczać miało według piagetowskiego modelu pełną dojrzałość struktur poznawczych. Następować ma ona w adolescencji. Dorośli stosować by mieli w rozumowaniu te same procedury, które stosują nastolatki. Myślenie logiczne byłoby zatem szczytowym osiągnięciem człowieka, gdyż „logika matematyczna jest aksjomatyką operacji umysłowych”²⁸⁷. Jak jednak podkreślane jest w postpiagetowskich teoriach rozwoju poznawczego, dorosły człowiek ma możliwość wyjść poza fazę operacji formalnych. Rezultatem dalszego rozwoju poznawczego jest osiągnięcie postformalnego stadium rozumienia, które charakteryzuje się według, na przykład, Michaela Basseches’a – rozumowaniem dialektycznym, Jan Sinnott – rozumowaniem relatywistycznym, Michaela Commonsa i Francis Richards – myśleniem metasystemowym i paradygmatycznym, Janusza Trempały – myśleniem kontekstualnym, czy Gisel Labouvie-Vief – zintegrowanym²⁸⁸.

282 *Ibidem*, s. 334-335.

283 Już w stadium operacji konkretnych dzieci zaczynają konstruować logikę klas i relacji, ale tylko na obiektach konkretnych. W tym kontekście warto pamiętać o neopiagetowskich teoriach rozwoju i możliwości wzrostu w dorosłości, zob. J. D. Sinnott, *The development of logic in adulthood: postformal thought and its applications*, Springer, New York 1998; Zwykle ignoruje się kwestię pojawienia się w myśli późnego Piageta modyfikacji jego podejścia. Dopuszczał on możliwość rozwoju poznawczego w dorosłości. Przejawem tego był podział na *early formal thinking* i *full operational formal reasoning*, oraz rozwój w dorosłości jako nigdy niekończący się proces. A. Machado, *In Defense of Piaget's Theory: A Reply to 10 Common Criticisms*, „Psychological Review”, 103, nr 1, 1996, s. 155, <https://web.archive.org/web/20200410214048/http://pdfs.semanticscholar.org/2dec/e70e2b917d7e3f846db9285878811d1ddbe3.pdf> (dostęp dnia 12.08.2022).

284 J. Piaget, *Psychologia i epistemologia*, Państwowe Wydawnictwo Naukowe, Warszawa 1977, s. 145.

285 *Ibidem*, s. 81.

286 B. Wadsworth, *Teoria Piageta - poznawczy i emocjonalny rozwój dziecka*, WSiP, Warszawa 1988, s. 131.

287 J. Piaget, *Psychologia i epistemologia*, Państwowe Wydawnictwo Naukowe, Warszawa 1977, s. 121.

288 P. Michalska, A. Szymanik-Kostrzevska, E. Gurba, J. Trempała, *Rozwój myślenia w dorosłości: o sposobach badania rozumowania postformalnego w codziennych sytuacjach*, „Psychologia rozwojowa”, 21, nr 2, 2016, s. 56.

Za Nietzschem można powtórzyć, że „świat wydaje się nam logiczny, ponieważ myśmy go uprzednio zlogizowali”²⁸⁹, tyle że myślenie formalno-operacyjne jest nieadaptacyjne w dorosłości i dlatego tak pożądane jest osiągnięcie stadium postformalnego – łączonego z mądrością²⁹⁰. Od problemu wielkiej teorii Eco, mówiącej że język, który opisuje różnorodny i skomplikowany świat miałby dać się ująć w formę logiczną²⁹¹, można uwolnić się dzięki możliwościom, jakie dają języki dziedzinowe. Są w stanie w adekwatny sposób opisać ten wycinek rzeczywistości, którego dotyczą. Człowiek ma, wobec tego szanse funkcjonować jako Piagetowski *Logik*, i to efektywnie, w ramach działalności programistycznej. Posługując się językami programowania, wykorzystuje logikę. Język staje się natomiast narzędziem myślenia, gdy pełni właśnie funkcję reprezentacyjną, a nie jedynie komunikacyjną²⁹².

Nie da się zaprzeczyć, że prócz funkcji komunikacyjnej język pełni również funkcję poznawczą. Polega ona na wspomaganiu procesów przetwarzania informacji, głównie myślenia i rozumowania, percepcji, pamięci oraz kontroli poznawczej i samokontroli emocjonalnej²⁹³. *Przetwarzanie języka*, polegające na „przetwarzaniu informacji biorących udział w tworzeniu wyrażen językowych i ich analizie”²⁹⁴, jest złożonym procesem poznawczym. Kwestia ta jest istotna w psychologii poznawczej, gdyż w większości koncepcji zakłada się bowiem jakiś związek myślenia i języka²⁹⁵. Co ciekawe poznawcze podejście do języka w psychologii pozwala na traktowanie języków sztucznych jako języków. Za Edwardem Nęcką, Jarosławem Orzechowskim i Błażem Szymurą warto przytoczyć definicję języka jako „systemu kodowania znaczeń poprzez wykorzystanie skończonego zbioru symboli (dźwięków mowy lub znaków pisma)”, przy założeniu, że „operowanie na symbolach podlega regułom gramatycznym, swoistym dla każdego języka naturalnego lub sztucznego”²⁹⁶. Nacisk

289 F. Nietzsche, *Wola mocy. Próba przemiany wszystkich wartości*, Wydawnictwo „bis”, Warszawa 1993, s. 313.

290 S. Tucholska, *Mądrość jako przejaw dojrzałości osobowościowej*, „Rocznik Filozoficzny Ignatianum”, 22, nr 1, 2016, s. 238.

291 U. Eco, *Poszukiwanie języka...*, s. 339, s. 355.

292 T. Maruszewski, *Psychologia poznania. Umysł i świat*, Gdańskie Wydawnictwo Psychologiczne, Gdańsk 2011, s. 366; stąd też wynikają niepowodzenia w uczeniu szympanśów języka, gdyż zwierzęta te używają języka głównie w jego funkcji komunikacyjnej.

293 E. Nęcka, J. Orzechowski, B. Szymura, *Psychologia poznawcza*, Państwowe Wydawnictwo Naukowe, Warszawa 2013, s. 591.

294 E. Nęcka, J. Orzechowski, B. Szymura, Sz. Wichary, *Psychologia poznawcza. Wydanie nowe*, Państwowe Wydawnictwo Naukowe, Warszawa 2020, s. 561.

295 Teoria filozoficzna Jerryego Fodora wiążąca się z przekonaniem o odrębności modularnej języka nie znajduje empirycznych dowodów i ma marginalne znaczenie.

296 E. Nęcka, J. Orzechowski, B. Szymura, *Psychologia poznawcza*, ..., s. 590.

kładziony jest przez nich na podporządkowanie abstrakcyjnym regułom gramatycznym i wykorzystanie systemu znaczeń, aby odnieść się do rzeczywistości. W tej perspektywie języki sztuczne z uwagi na to, że kodują znaczenia, można by uznać za języki, w przeciwieństwie do form komunikacji zwierząt, które ograniczają się do funkcji komunikacyjnej takiej jak prośby, polecenia, ostrzeżenia.

Posługiwanie się językami sztucznymi jest jednak czymś więcej niż tylko używaniem znaków w sposób jednoznaczny²⁹⁷, a w przypadku języków apriorycznych – ujęciem dodatkowo rzeczywistości w logiczną formę. Gdy George Steiner argumentował za koniecznym bałaganiarstwem i rozchwianiem języka naturalnego, zwracał uwagę, że w językach formalnych „»zamknięta« składnia, formalnie skończona semantyka oznaczałaby zamknięty świat”²⁹⁸. Tym otwarciem na niego byłby już jednak potencjał słowotwórczy zawarty w językach aposteriorycznych, którym brak narzuconego uporządkowania, dzięki wzorowaniu się w budowie na językach naturalnych. W wielu językach programowania wysokiego poziomu można w prawdzie doszukiwać się cech języków mieszanych aprioryczno-aposteriorycznych, jednakowoż nie wspomnianego „rozchwiania”.

Zmienność w języku naturalnym można wiązać też z ulotnością mowy. Brak możliwości wyrażania się z jej użyciem w językach sztucznych²⁹⁹ stał się jednym z zarzutów wysuwanych szczególnie przez filozofów związanych z nurtem heideggerowskim. Już wszelako od Starożytności niemożliwość wypowiedzenia się była traktowana jako ułomność rozumu. O ile Sokrates wskazywał na sposoby komunikacji możliwe dla osób niemówiących bez elementu ocennego (mimika, gesty, ruch)³⁰⁰, to już Arystoteles był przekonany, że niemożliwy jest rozwój intelektualny bez mowy, bo tylko w niej wyraża się myśl³⁰¹. Co gorsze niestety podejście tego filozofa ukształtowało stosunek do osób niemówiących i głuchych, jako tych które są pozbawione rozumu. Na podstawie przekonania, że myślenie bez mówienia jest niemożliwe³⁰², odmawiano im kształcenia.

297 Możemy znaleźć i od tego sporo wyjątków np. w apriorycznym języku Solresol wiele słów jest wieloznacznych. Przykładowo „misila” oznacza radość, przyjemność, ale również intoksykację i upojenie alkoholowe, „doremisol” oznacza zarówno usta, wargi jak i język.

298 G. Steiner, *Po wieży Babel*, Wydawnictwo Aletheia, Warszawa 2018, s. 290.

299 przyjmijmy chwilowo za wspomnianymi adwersarzami, że to prawda, choć znajdujemy od niej wiele wyjątków.

300 M. Garbat, *Historia niepełnosprawności*, Novae res, Gdynia 2015, s. 483. za W. Rebhorn, *Renaissance debates on rhetoric*, Ithaca-Cornell University Press, New York 2000, s. 87.

301 M. Garbat, *Historia niepełnosprawności...*, s. 481.

302 *Ibidem*, s. 482-484.

Wpływ tej niesprawiedliwej społecznie i błędnej tezy Arystotelesa osłabił dopiero wraz ze *złotym wiekiem języka migowego*, czyli osiemnaście stuleci później, gdy Girolamo Cardano, otwarty krytyk Stagiryty, nauczył swojego głuchego syna czytać, pisać i rachować, dowodząc ówczesnym bezpodstawności krzywdzących poglądów filozofa. Jednak w ramach rozwijanego szkolnictwa osób głuchych pozostał wciąż nacisk na czytanie z ruchu ust i naukę mowy. Niesławny oralizm wspierany był instytucjonalnie w Polsce aż do początków XXI wieku. Audystyczny punkt widzenia był podstawą panującego w szkołach dla dzieci głuchych oralizmu, a za jego pokłosie można uznać forsowanie w Polsce, opartego przecież na języku fonicznym, SJM³⁰³. Stanowiło to przejaw dominacji społecznej osób słyszących-uprzywilejowanych nad Głuchymi³⁰⁴. Z perspektywy współczesnej wiedzy na temat języka, brak jego pierwszego poziomu (fonicznego) nie podważa z uwagi na sposób komunikacji jego statusu języka.

W kontekście języków migowych pojawia się jednak istotna etycznie kwestia, o której dobrze jest tu wspomnieć. Języki sztuczne mogą stanowić narzędzie ucisku grupy. Represjonowanie w szkolnictwie za używanie języka migowego, uniemożliwianie komunikacji w nim, instytucjonalne faworyzowanie sztucznego języka migowego (SJM) było i jest krzywdzące dla mniejszości kulturowo-społecznej, którą są Głusi³⁰⁵. Manualna postać języka (SJM) jest bowiem subkodem języka polskiego narzucanym osobom, które mają już swój język – PJM. Faworyzowanie przez słyszącą większość subkodów języka, którym sami się posługują, stanowiło niechlubną rzeczywistość nie tylko w Polsce. PJM i SJM są wykorzystywane jako przykład, jednak sytuacja podobnie kształtowała się również na innych obszarach, na przykład w Ameryce Północnej, pomiędzy naturalnym ASL i manualną postacią języka angielskiego³⁰⁶. Bycie w języku może być osiągnięte w języku wizualno-

303 SJM – system językowo migowy (zwany migany) jest sztucznym językiem migowy który był w PRL faworyzowany przez Polski Związek Głuchych. Powstanie systemu i jego instytucjonalizacja była zapoczątkowana przez jego prezesów, dla których to foniczny język polski był językiem pierwszych, a dopiero w dorosłości nauczyli się migać (łatwiejszy był zatem dla nich język migany). <https://swiatgluchych.pl/video/co-to-jest-system-jezykowo-migowy-wyklad-bogdana-szczepankowskiego/> (dostęp dnia 2.05.2022).

304 H. Lane, *Maska dobroczynności. Deprecjacja społeczności głuchych*, Wydawnictwo Szkolne i Pedagogiczne, Warszawa 1996, s. 68.

305 Użycie wielkiej litery związane jest z odniesieniem do społeczności Głuchych, nie do grupy osób z dysfunkcją narządu słuchu.

306 *Ibidem*, s. 143-144.

przestrzennym, bo to nie brak mowy, byłby istotą problemu języków sztucznych, lecz nieakceptowalny dla wielu filozofów dystans, na przykład formy pisemnej.

Ludzie posługujący się językiem migowym myślą w języku, dowodem na to jest aktywność tych samych struktur mózgowych podczas odbierania komunikatu językowego w formie wizualnej jak podczas słuchania pojedynczych słów³⁰⁷. Rozumienie więc, jest nieodparcie zjawiskiem językowym, ale oczywiście nie tylko. Za budzące wątpliwość można uznać, przekonanie niektórych badaczy twierdzących, że jest fenomenem bardziej językowym nawet niż umysłowym. Rozumienie jako zdolność poznawczą, można wszak badać zarówno w skali werbalnej, jak i niewerbalnej³⁰⁸. Już u dzieci w wieku przedszkolnym rozróżnia się zarówno myślenie obrazowe – pochodzące bezpośrednio z doświadczenia percepcyjnego, jak i pojęciowe – polegające na przetwarzaniu informacji reprezentowanych w systemie werbalnym³⁰⁹. Ujęcie to bazuje na koncepcji podwójnego kodowania Allana Paivio³¹⁰, a syntezy takiego sposobu myślenia o języku można doszukiwać się w filozofii Wittgensteina³¹¹. Co jednak wydaje się szczególnie istotne dla podjętych rozważań to fakt, że o ile język naturalny stanowi abstrakcyjność w myśleniu, to według Paivio, języki sztuczne, jak na przykład

307 Dzięki obrazowaniu przy pomocy pozytonowej tomografii emisyjnej L. A. Petitto i współpracownicy zauważyli w szeregu badań podobnie umiejscowiony przepływ krwi w mózgu u głuchych podczas odbierania komunikatu w SL i osób słyszących przetwarzających mowę dźwiękową, również podobną lateralizację - używanie lewej półkuli w obu przypadkach, (L. Petitto, *The acquisition of natural signed languages: Lessons in the nature of human language and its biological foundations*, [w:] *Language acquisition by eye*, red. C. Chamberlain, J. P. Morford, R. I. Mayberry, Lawrence Erlbaum Associates Publishers, Mahwah 2020). Potwierdzili tym samym wyniki badań D.P. Corina (D. Corina, *Studies of neural processing in deaf signers Toward a neurocognitive model of language processing in the deaf*, „Journal of Deaf Studies and Deaf Education”, 3, nr 1, 1998); Sangeeta Bagga Gupta, *Literacies and deaf education. A theoretical analysis of the international and Swedish literature*, „Forskning i Fokus” nr 23, 2004, s. 51.

308 Lepszym wskaźnikiem potencjału poznawczego w teście na inteligencję np. Stanford-Binet 5 może być wynik na niewerbalnej skali, jeśli występują takie czynniki jak zaburzenia komunikacji, trudności w nauce, autyzm, niepolskojęzyczne środowisko. Diagnostyci mają możliwość porównywać każdy z czynników w skali werbalnej i niewerbalnej. W ten sposób są w stanie rozpoznać mocne strony osoby badanej. Materiały do testu SB5 s. 142.

309 A. Grob, Ch. Meyer, P. Hagmann-von Arx, *Skale inteligencji i rozwoju dla dzieci w wieku 5-10 lat: IDS intelligence and development scales. Podręcznik*, adaptacja A. Jaworowska, A. Matczak, D. Fecenec, Pracownia Testów Psychologicznych Polskiego Towarzystwa Psychologicznego, Warszawa 2012, s. 11.

310 Koncepcja dotyczy reprezentowania rzeczywistości przez człowieka w dwóch systemach, twórcy IDS na jej podstawie skonstruowali dwa oddzielne testy do pomiaru zdolności w zakresie dwóch różnych systemów myślenia. A. Grob, Ch. Meyer, P. Hagmann-von Arx, *Skale inteligencji i ...*, s. 11.

311, „Chociaż Wittgenstein twierdził, że język ukształtowany jest na podstawie pewnych właściwości przyrody, wykazywał też, że każdy opis tych właściwości zakłada już reguły gramatyki. Skoro wzorze tego, co uchodzi za zdania empiryczne wyznacza sam język, to nie można wyraźnie oddzielić dociekań gramatycznych od empirycznych, pytań o pojęcia, od pytań o przedmioty. A to znaczy, że wraz z językiem zyskujemy pewien obraz świata i patrzymy na świat przez pryzmat swego języka. W konsekwencji stajemy się więźniami obrazu świata w nim tkwiącego.” M. Maciejczak, *Dwie teorie języka. Ludwig Wittgenstein (1889-1951) i Martin Heidegger (1889-1976)*, „Studia Philosophiae Christianae”, nr 43, 2007, s. 10.

języki programowania, są jeszcze bardziej abstrakcyjne i tym samym odległe od reprezentacji w systemie sensorycznym³¹².

Językom sztucznym zarzuca się, jednak, że posiadają, jak było podkreślane przez chociażby Hansa-Georga Gadamera, pewne *ograniczenia* wynikające, na przykład, z ujęcia ich w postaci pisanej. Już opanowanie jej wymaga wielu wysiłków i interakcji nakierowanych na pozyskanie tej umiejętności. Piszący z założenia jest „na zewnątrz” przekazu, a nie w środku wydarzeń, tak jak ma to miejsce w mowie³¹³. Wynika to z dystansu fizycznego, gdy myśl jest utrwalana w formie pisemnej. W ontologii wyrazem tej refleksji stała się konstatacja Heideggera o języku jako „domostwu bycia” (*Hause des Seins*), bliska zarówno wspomnianemu Gadamerowi, jak i Cassirerowi³¹⁴. Filozofowie z kręgu fryburszycy przyjmowali, że rozumienie dzieje się w mowie i ta aktualność ma istotne znaczenie, gdyż „panuje ta sama jedność myśli i mowy”³¹⁵. Warto jednak podkreślić, że biorąc pod uwagę cel powstania języków programowania, zgoła nie komunikacyjny, nie powinno to przesądzać o jakiejś jakościowej niższości formy w tym kontekście.

Oczywiście znane są pomyślnie zrealizowane projekty językowe będące pasylalami, w których rozumienie, dzieje się w mowie. Są to języki aposterioryczne, wywodzące się z koncepcji systemów „wyrażania łatwego i elastycznego”³¹⁶. Już podczas projektowania ich twórcy uwzględniali na pierwszym miejscu cel komunikacyjny. Można porównać, wspomniany już wielki sukces *esperanto* z niesławną porażką *wolapuk*. Pierwszy będąc językiem aposteriorycznym, zaprojektowany został, tak by być łatwym w wymowie dla ludzi wywodzących się z kręgu kultury europejskiej. Leksyka bazowała na wspólnych rdzeniach, które Ludwik Zamenhoff zaczerpnął, według własnego uznania, głównie z języków romańskich i

312 Tłum. własne fragmentu: “Representations can also be described as varying in concreteness-abstractness, a dimension that correlates with the distinction between picture-like and language-like symbols. Thus, at one extreme we have highly concrete, iconic, modality-specific representations of objects and events. A three-dimensional colored motion picture with a sound track could be indistinguishable perceptually from real world events. At the other extreme we have completely abstract, amodal (or at least not modality-dependent) representations that are only arbitrarily related to real world objects and events. Human language is a clear case, but artificial languages such as those used by computers are, if anything, even more abstract”. A. Paivio, *Mental Representations*. Oxford University Press 1990, s. 17, oxford.universitypressscholarship.com (dostęp dnia 18.07.2022).

313 J. Bobryk, *Język mówiony, pismo, Internet*, [w:] *Język jako przedmiot badań psychologicznych. Psycholingwistyka ogólna i neurolingwistyka*, red. I. Kurcz, H. Okuniewska, Wydawnictwo Szkoły Wyższej Psychologii Społecznej „Academica”, Warszawa 2011, s. 171.

314 A. Bronk, *Rozumienie, dzieje, język*, Lublin 1988, s. 269.

315 H.-G. Gadamer, *Prawda i metoda...*, s. 544.

316 U. Eco, *Poszukiwanie języka ...*, s. 354.

germańskich. Drugi natomiast słownictwo miał tak zmodyfikowane przez Johanna Schleyera, że dla użytkownika języka stał się trudny do wypowiedzenia, a i również do zapamiętania. W tym języku mieszanym, aprioryczno-aposteryorycznym, autor podjął dodatkowo próbę zorganizowania systemu treści, jednak w sposób, który dla odbiorcy nie jest ani intuicyjny, ani nawet wytłumaczalny. O ile języki aprioryczne zwykle łączy się z możliwością wypowiedzi i ich celem komunikacyjnym, języki aprioryczne wiążą się z pasygrafią.

Historycznie rzecz ujmując, języki aprioryczne tworzone były często jako doskonałe pismo, a zatem jako projekty pasygraficzne. Mimo, że filozoficznymi językami XVII wieku takimi jak *Characteristica universalis* Gottfrieda Leibniza, *Ars signorum* Georga Dalgarno, czy *Real character* Johna Wilkinsa można było według ich autorów posługiwać się w mowie, ich założenia to jednak uniemożliwiały. Analogicznie, tekst zapisany w języku programowania jest możliwy do przeczytania na głos, jednak taka prezentacja jest pozbawiona sensu. Niemożliwość lub trudność w komunikacji głosowej nie świadczy o ułomności sposobu komunikacji. Przekonanie o bezpośredniości mowy *versus* dystansie pisma nie ma przełożenia na języki sztuczne, które są projektowane w innym celu niż komunikacja. W pasylaliach zaś dystans nie występuje, chyba że chodzi o kwestię namysłu, jako że nie są prawie nigdy językami pierwszymi³¹⁷.

Języki mogą pozostać pasylaliami lub migowe i nigdy nie wybrzmieć, jeśli sprzeciwia się to zamysłowi, dla którego powstały. Niestety niemota, jak już zostało wspomniane, chociaż rzadziej, ale jednak, nadal bywa traktowana jako coś wstydliwego, a nawet złego. Stawia się ją w kontrze do mowy dźwiękowej, która to umiejętność według oralistów miałaby nobilitować i pozwalać „przejsć do wyższej kategorii człowieczeństwa”³¹⁸. Ponadto związane bywa to również z dyskryminacyjnym i stygmatyzującym podejściem zarówno do głuchoty, jak i nieopanowania przez większość Głuchych mowy na poziomie umożliwiającym swobodne porozumiewanie się ze słyszącą większością. Komunikacja niedźwiękowa jest równoprawną formą komunikacji, czy chodzi o pismo, komunikację wizualno-przestrzenną czy wspomagającą i alternatywną, skoro odpowiada na ludzkie potrzeby.

317 Znane są przypadki wychowywania dzieci w języku esperanto i nieudana próba w klingońskim.

318 Przykład przekonania oralistów zaczerpnięty z M. Garbat, *Historia niepełnosprawności...*, s. 129.

Prezentację treści w formie pisemnej dookreśla struktura łączenia informacji. Dzieło literackie prezentowane jest zazwyczaj³¹⁹ jako klasyczny tekst, program komputerowy natomiast zapisany jest w hipertekstualnym kodzie źródłowym. Linearność tekstu nawet najbardziej tradycyjnego jest jednak tylko formalna jak wskazują lingwiści powołujący się na nielinearność interpretacji³²⁰. Kohezja tekstu³²¹ dotyczy jedynie jednego z poziomów wielowarstwowej struktury i występuje obok równie istotnej spójności semantycznej. Koherencja, czyli wspomniana spójność znaczeniowa, umożliwia rozumienie, które to jednak wymaga poszukiwania sensów. Jest to czynność nie przebiegająca w sposób liniowy, gdyż „w akcie interpretacji z reguły następuje proces burzenia jego (tekstu, *przyp. aut.*) linearnego układu”³²². Uczestnik ma bowiem wpływ na przebieg rozumienia, nie determinuje go warstwa informacyjna. Dynamiczny charakter interpretacji dostrzegł przed lingwistami Ricœur jako „wysiłek, by obcość i oddalenie uczynić efektywnymi”³²³. Wszak: „dzieło dyskursywne jest (...) czymś więcej niż linearną sekwencją zdań. Jest to kumulatywny, holistyczny proces”³²⁴. Znaczenie tekstu musi być wydobyte. Działanie to jednak nie jest procesem liniowym, czy odgórnie ustalonym. Wyznacznika tekstowości nie stanowi zatem linearność.

Hipertekstowi, czy to w postaci hiperfikcji czy kodu źródłowego programów komputerowych przysługuje status tekstu, przynajmniej w ujęciu jako jednostki abstrakcyjnej. Badacze zajmujący się e-tekstem: lingwiści, komunikolodzy, socjologowie czy kulturoznawcy, wszyscy oni koncentrują zwykle uwagę jedynie na hipertekstualności treści prezentowanych już użytkownikowi. Pomijają jednak kwestię dokumentu HTML – jeśli hiperfikcja jest zapisana w języku znaczników. Można rozszerzyć to spostrzeżenie na konstatację, że pomimo zdawania sobie sprawy z istnienia kodu źródłowego programów komputerowych, najwyraźniej nie uznają go za tekst, a jego hipertekstualność jest tylko „koncepcją i zbiorem możliwości łączenia informacji w określoną strukturę”, a nie „konkretną realizacją struktury hipertekstowej

319 Wyjątek stanowią hiperteksty literackie. Hiperfikcja jest gatunkiem literatury elektronicznej, termin ten bywa również stosowany do wydanych w formie papierowej książek, których fabuła jest nielinearna.

320 B. Witosz, *Lingwistyczne koncepcje tekstu wobec wyzwań komunikacji wirtualnej*, [w:] *Język w internecie. Antologia*, red. M. Kita, I. Loewe, Wydawnictwo Uniwersytetu Śląskiego, Katowice 2016, s. 104-105.

321 Spójność formalna.

322 B. Witosz, *Lingwistyczne koncepcje ...*, s. 104.

323 P. Ricœur, *Zadanie hermeneutyki*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 121.

324 *Ibidem*, s. 163.

zwierającą informację”³²⁵. Tą jest tylko, według tej dość rozpowszechnionej u wspomnianych humanistów perspektywy, jedynie prezentowany na stronie WWW tekst. Skompilowany program jest jednak również realizacją tekstu, co jest niezauważane lub może milcząco nieuznawane, przez wspomnianych badaczy.

Zastanawiając się, co definiuje języki sztuczne, można dojść do przekonania, że nie jest to jednak nienaturalność, czy obcość, często im zarzucane. Ogólnie ujmując, projekt językowy:

„jest językiem sztucznym, tzn. tworzonym w sposób świadomy. Nie ma w nim przypadkowości (...), przy nazywaniu konkretnych zjawisk i rzeczy”³²⁶.

Dlatego sensowne wydaje się używanie, kalki z języka angielskiego, czyli nazwy „języki skonstruowane” – od słowa *constructed*. Jest bardziej trafna niż rodzime słowo „sztuczne”. Nie sztuczność bowiem, która jest pojęciem nacechowanym negatywnie, ale akt kreacji jest zabiegiem fundamentalnym.

Stosowanie powszechnie w tej pracy nazwy „sztuczne”, wynika z powielania przyjętego w polskim językoznawstwie nazewnictwa. Niemniej jednak warto zaznaczyć możliwość stosowania rzadko używanego w Polsce terminu i wyjaśnić ewentualne nieporozumienia. Do języków konstruowanych zaliczamy bowiem również języki naturalne pomocnicze, na przykład języki pidżynowe³²⁷. Powstały w celach handlowych wskutek uproszczenia struktur gramatycznych i przy użyciu minimalnej ilości słownictwa³²⁸, będąc przykładem jak „ludzie, »w naturalny sposób« wymyślają i rozwijają prosty system (językowy, *przyp. aut.*)”. W porównaniu do języków kreolskich nie zostały jeszcze poddane natywizacji, więc nie mają *rodzimych* użytkowników.

Kwestia braku posługiwania się językiem jako pierwszym rzutuje też na sprawę odrodzenia języka hebrajskiego. Pierwsze dziecko wychowano we współczesnym hebrajskim, jako ojczystym, pod koniec XIX wieku³²⁹. Język ten choć naturalny został ożywiony³³⁰, a nawet w dużej mierze konstruowany. Trudno pominąć milczeniem wkład twórcy odrodzenia języka hebrajskiego, ale i wprost konstruktora wielu słów

325U. Żydek-Bednarczuk, *Hipertekst w perspektywie lingwistycznej, komunikacyjnej i kulturowej*, [w:] *Język w internecie. Antologia*, red. M. Kita, I. Loewe, Wydawnictwo Uniwersytetu Śląskiego, Katowice 2016. s. 114.

326M. Izert, E. Pachocińska, *Wstęp do językoznawstwa ...*, s. 41.

327 K. Wojan, *Języki sztuczne...*, s. 25-26.

328 *Atlas języków. Pochodzenie i rozwój języków świata*, red. B. Comrie, S. Matthews, M. Polonsky, Oficyna Wydawnicza Atena, Poznań 1998, s. 146.

329Był to Itamar Ben-Avi, syn językoznawcy i tzw. twórcy współczesnego hebrajskiego, Eliezera ben Jehudy.

330 *Ibidem*, s. 215.

Eliezera ben Jehudy³³¹. Nie była to próba ocalenia podobna do przywracania dialektu prowansalskiego czy gaskońskiego, ale praca polegająca na wymyślaniu słów i tworzeniu słownika, nie tylko przez rewitalizację biblijnego hebrajskiego, ale przez tworzenie wyrazów.

Pojawiają się nawet głosy, w prawdzie nieliczne, aby dostrzegać również w sformalizowanych językach naturalnych³³² wspomniany element kreacji i sztuczności, gdy na przykład rozważamy kwestię znormalizowania gramatycznego. W niektórych podziałach typologicznych, na przykład Blanke'go, występuje typ języków etnicznych wysoce regulowanych³³³. Pojęcie języków skonstruowanych może być zatem bardzo inkluzyjne i zawierać się w nim mogą zarówno języki formalne, jak i odległe wobec nich języki naturalne wysoce uregulowane. Skoro są tak różnorodne, ważką kwestię stanowi, jak z perspektywy humanistycznej badać język występujący w kodzie źródłowym, uwzględniając przy tym specyfikę języków programowania.

1.6. Perspektywy humanistycznych badań nad kodem źródłowym

Szerszą perspektywę niż badania językoznawcze, których przedstawiciele przecież i tak nie wykazują zainteresowania analizowaniem tekstów programów komputerowych, czy badaniem języków programowania (poza bardzo niszową *philology of software*)³³⁴, dostarczają humanistyczne rozważania nad kodem źródłowym pod nazwą *Critical Code Studies* (CCS). Podejście CCS koncentruje się na badaniu kodu źródłowego jako tekstu, nie tylko w sensie hipotetycznym. Oznacza to analizowanie tego wytworu i wyjaśnianie „jako systemu znaków wraz ze swoją własną retoryką, jako komunikacji werbalnej posiadającej znaczenie ponad tylko funkcjonalną

331 L. Ahearn, *Antropologia lingwistyczna. Wprowadzenia*. Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2013, s. 249.

332 Bojar zauważa na przykład, że „współczesne językoznawstwo dąży do sformalizowania języków naturalnych, czyli do opisanie w taki sposób ich słownika, semantyki i reguł składniowych, w jaki opisuje się języki formalne”. B. Bojar, *Językoznawstwo dla studentów informacji naukowej*, Wydawnictwo SBP, Warszawa 2005, s. 47.

333 K. Wojan, *Języki sztuczne...*, s. 24.

334 *Philology of software* jest pomysłem, który nie znalazł kontynuatorów. Stanowi próbę filologicznego ujęcia przez Moritza Hillera kodu źródłowego - jego wymiaru tekstowego w powiązaniu w wymiarem procesualnym programu. M. Hiller, *Signs of the times: the software of the philology and a philology of the software*, „Digital Culture and Society”, 1, 2015, s. 152–63.

użytecznością”³³⁵. Wychodzi się, wobec tego z założenia, że kod źródłowy posiada warstwę kulturową podlegającą rozumieniu i interpretacji, co umożliwia rozpatrywanie jego wymiaru artystycznego, społecznego czy też historycznego.

Wspomniane pole badań, zapoczątkowane przez Marca Marino, rozwija się interdyscyplinarnie. Oprócz badań nad językiem, które nie są szczególnie popularne wśród badaczy zainteresowanych *Critical Code Studies*, obszar ten zdominowany jest raczej przez problematykę społeczną oraz artystyczną. Hermeneutyka kodu źródłowego ujmowana jest w tych publikacjach z różnorodnych perspektyw: literackiej (np. „Digital Code and Literary Text”³³⁶), retorycznej (np. „Digital Rhetoric: Theory, Method, Practice”³³⁷), estetycznej (np. „The Aesthetics of Generative Code”³³⁸), artystycznej (np. „Critical Code Studies: Initial methods”³³⁹), edukacyjnej (np. „Where Computer Science and Cultural Studies Collide. The Chronicle of Higher Education”³⁴⁰), medioznawczej (np. „Transcoding Sexuality: Computational Performativity and Queer Code Practices”³⁴¹) groznawczej (np. „Somewhere Nearby is Colossal Cave: Examining Will Crowther’s Original „Adventure” in Code and in Kentucky”³⁴²), sztucznej inteligencji (np. „Hermeneutics: from textual explication to computer understanding”³⁴³), oraz związane z NLP (np. „Natural Language Processing in Facade: Surface Text Processing”³⁴⁴).

335 Tłum. własne, M.C. Marino, *Critical code studies*. *Electronic Book Review*, 2006, <http://www.electronicbookreview.com/thread/electropoetics/codology> (dostęp dnia 2.07.2022).

336F. Cramer, *Digital Code and Literary Text*, „Beehive”, 4, nr 3, 2001. https://www.netzliteratur.net/cramer/digital_code_and_literary_text.html (dostęp dnia 7.07.2022).

337 D. Eyman, *Digital Rhetoric: Theory, Method, Practice*, University of Michigan Press, Ann Arbor 2015.

338 G. Cox., A. McLean., A. Ward, *The Aesthetics of Generative Code*. *Generative Art 00 conference 2001*, Politecnico di Milano, Milan 2001, https://www.academia.edu/10519146/The_Aesthetics_of_Generative_Code (dostęp 7.07.2022).

339 M. C. Marino, *Critical Code Studies: Initial methods*, Cambridge 2020, The MIT Press.

340M. Kirschenbaum, *Where Computer Science and Cultural Studies Collide. The Chronicle of Higher Education*, 2009, <https://www.chronicle.com/article/where-computer-science-and-cultural-studies-collide/> (dostęp dnia 7.07.2022).

341 G. Jackson, *Transcoding Sexuality: Computational Performativity and Queer Code Practices*, „A Journal in GLBTQ Worldmaking”, 4, nr 2, 2017, s. 1–25.

342 D. Jerz, *Somewhere Nearby is Colossal Cave: Examining Will Crowther’s Original „Adventure” in Code and in Kentucky*, „Digital Humanities Quarterly”, 1, nr 2, 2007, <http://www.digitalhumanities.org/dhq/-vol/001/2/000009/000009.html> (dostęp dnia 17.07.2022).

343J. C. Mallery, R. Hurwitz, G. Duffy, *Hermeneutics: From Textual Explication to Computer Understanding? The Encyclopedia of Artificial Intelligence*, John Wiley & Sons, New York 1986.

344 M. Mateas, A. Stern, *Natural Language Processing in Facade: Surface Text Processing. Paper presented at the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE)*, 2004, <http://users.soe.ucsc.edu/~mi-chaelm/publications/mateas-tidse2004.pdf> (dostęp dnia 7.07.2022).

Stosując krytykę hermeneutyczną do interpretacji³⁴⁵ w kontekście społeczno-historycznym badacze *Critical Code Studies* nie koncentrują się jednak na samej kwestii rozumienia jako takiego. Podejście to skupia się na przedmiotowym ujęciu problemu, analizie tekstu i jego możliwości. W tej pracy natomiast, badając język kodu źródłowego z perspektywy filozofii form symbolicznych i hermeneutyki dyskursu, tekst kodu źródłowego programów komputerowych traktowany jest jako ucieleśnienia sensu i pokazuje jego podmiotową istotę. *Clou* stanowi bowiem przyjrzenie się wpływowi kultury na elementarną część świata nowych technologii – kod źródłowy³⁴⁶, co nie tylko pokrywa się z podejściem *Critical Code Studies*, ale wzbogaca je o zagadnienia powiązane z antropologią filozoficzną, kontynentalną filozofią języka czy hermeneutyką³⁴⁷.

Szukając ugruntowania dla teorii *Critical Code Studies* wskazane byłoby zagłębić się również w filozofię nauki. Z założeń ontologicznych co do natury programu komputerowego wyrastają powiązane z nimi podejścia epistemologiczne. Jeśli bowiem postrzegamy program jako obiekt matematyczny konsekwencją tego będzie badanie go metodami formalnymi³⁴⁸. Oprócz paradygmatu matematycznego do badania własności programów rozważać można podejście z użyciem metod inżynierskich, czyli paradygmat technokratyczny, czy też paradygmat naukowy³⁴⁹. Wiedzę, zarówno aprioryczną – czerpaną z formalnej dedukcji - jak i aposterioryczną - pozyskiwaną w wyniku eksperymentów - godzi właśnie podejście obecne w ramach ostatniego podejścia³⁵⁰. Badacze mogą postrzegać zatem siebie jako matematyków, inżynierów i naukowców (*empirical scientists*)³⁵¹.

O ile Peter Wagner wyróżniając w latach 70-tych XX wieku wspomniane paradygmaty był przekonany o ich historycznej dominacji w kolejnych dziesięcioleciach, to zdania w tej kwestii wśród badaczy należy uznać za podzielone.

345 Nie tylko kodu źródłowego, ale też architektury programu i dokumentacji.

346 Odwrotny kierunek prezentuje podejście *Cultural Informatics, Software Studies*. Są to terminy opisujące studia nad wpływem nowych technologii na kulturę.

347 W hermeneutyce filozoficznej nastąpiło rozszerzenie historycznie przyjętego obszaru zainteresowania, tego co podlega wykładni. Wpierw nastąpiło wyjście poza egzegezę tematyki Biblijnej czy talmudycznej i zainteresowanie poezją, czy klasycznymi tekstami, aż do zainteresowania wszelkimi przejawami życia sprawiło, że wszystko co wyrażone w języku może podlegać egzezie.

348 I. Bondecka-Krzykowska, *Z zagadnień ontologicznych informatyki*, Wydawnictwo Naukowe UAM, Poznań 2016, s. 82.

349 A. Eden, *Three Paradigms of Computer Science*, „Minds and Machines”, 17, nr 2, 2007, s. 135-167.

350 *Ibidem*, s. 137.

351 P. Wegner, *Research paradigms in computer science*, Proceedings of the 2nd International Conference on Software Engineering, San Francisco 1979, s. 322, <https://tiny.pl/whbfg> (dostęp dnia 23.07.2022).

Amnon Eden dostrzegł, na przykład, że dotychczasowa dominacja podejścia technokratycznego nie służy rozwojowi inżynierii oprogramowania, co odbija się na jakości oprogramowania³⁵². William Wulf nawoływał natomiast do zerwania ograniczeń w wyborze tylko jednego, a nawet zachęcał do zdefiniowania swoich własnych paradygmatów³⁵³. Apelowal ponadto do inkorporowania do informatyki tego, co możliwe i pomocne z innych dyscyplin (w tym również nauk humanistycznych)³⁵⁴. Zachętę tę podjęli badacze z perspektywy *Critical Code Studies*, chociaż nie uważają się za kontynuatorów tej myśli.

Za kolejny drogowskaz, na który *Critical Code Studies* się nie powołuje, choć powinno, służyć może koncepcja Raymonda L. Wildera ujmująca matematykę jako system kulturowy. Koncepcja zakładała, że dziedzina ta należy do tradycji kulturowej danego społeczeństwa, co pozwala na odkrywanie wpływów elementów kultury ogólnej na rozwój matematyki³⁵⁵. Powołując się na różnice w rozwoju w różnych kręgach kulturowych oraz historyczne zmiany form myślenia w obrębie tej dyscypliny³⁵⁶, matematyk zauważył, że „stan i kierunki rozwoju matematyki są zdeterminowane przez cały zespół sił kulturowych (działających, *przyp. Murawskiego*) w i poza matematyką”³⁵⁷. Skoro ta dyscyplina nosiłaby ślady systemu kulturowego, to tym bardziej zyskuje szanse na to informatyka.

Tezy Wildera wpisują się w prąd filozofii nauki Imre Lakatosa. Matematyk był bowiem prekursorem (niezauważonym w swoim czasie) myślenia o *quasi-empiryzmie* w matematyce. Dyscyplina ta przestaje być postrzegana jako pewna i niepodważalna, gdyż stare teorie podlegają krytyce i korekcie, gdy są błędne lub, na przykład, niejasne. Nie jest zatem zbiorem jedynie niepodważalnych tez *a priori*³⁵⁸. W związku z

352 A. Eden, *Three Paradigms ...* s. 160.

353 W.A. Wulf, *Are We Scientists or Engineers?* *ACM Computer Surveys*, 27, 1995, s. 56 <https://tiny.pl/whbjh> (dostęp dnia 23.07.2022).

354 *Ibidem*, s. 57. “Let’s adopt the applicable best from science, engineering, the social sciences, and the humanities. Let’s be inclusive rather than exclusive. Let’s build even more bridges to other disciplines and to industry”

355 R. Murawski, *Filozofia matematyki. Zarys dziejów*, Państwowe Wydawnictwo Naukowe, Warszawa 1995, s. 150.

356 R. Wilder, *Kulturowa baza matematyki*, [w:] *Współczesna filozofia matematyki. Wybór tekstów*, red. R. Murawski, Warszawa 2002, Państwowe Wydawnictwo Naukowe, s. 288.

357 *Ibidem*, s. 291.

358 R. Murawski, *Wstęp. Renesans empiryzmu we współczesnej filozofii matematyki*, [w:] *Współczesna filozofia matematyki: wybór tekstów*, red. R. Murawski, Państwowe Wydawnictwo Naukowe, Warszawa 2002, s. 216.

tendencjami empirystyczno-induktywistycznymi³⁵⁹, w źródłach i metodach dyscypliny, dostrzegany jest przez filozofa *quasi* empiryzm matematyki³⁶⁰. Teorie niesformalizowane według filozofa miały dostarczać kontrprzykładów i były potencjalnymi falsyfikatorami³⁶¹. Wiedza matematyczna, kształtowana jest według niego dziejowo, w czym widoczne jest nawiązanie do popperowskiej wizji rozwoju nauki³⁶². Rozważania te mają ścisły związek z informatyką. Dowód matematyczny twierdzenia o czterech barwach (4CT), wspierany właśnie komputerowo, zapoczątkował możliwość spojrzenia na aktywność matematyczną, wychodząc poza dowody formalne. W przypadku tym dowodzenie nie ogranicza się jedynie do dedukcji *a priori*, ale korzysta z eksperymentu³⁶³. Dzięki takiemu zastosowaniu informatyki, matematyka zyskuje charakter *quasi*-empiryczny.

Wraz z nową wizją matematyki związaną z: „utrata pewności, zmianą standardów ścisłości, wagą rozważań nieformalnych i przedmatematycznych, rolą indukcji niezupełnej i analogii, rewidowalnością dowodów, nieostatecznością teorii matematycznych i ich możliwej niewspółmierności, trudności z utworzeniem wszechobecnej narracji, braku niewzruszonych podstaw, związku metody aksjomatycznej z szeroko rozumianą empirią, niemożliwością pełnego zdefiniowania podstawowych pojęć, ich »płynnością«, nieusuwalną roli intuicji i zaplecza kulturowego, wpływu przypadku, roli aspektów historycznych, potrzeby przyjmowania podstawowych założeń »na wiarę«”³⁶⁴, matematyki nie można traktować jako nauki humanistycznej. Co nie znaczy, że nie może posiadać cech z nią kojarzonych. Tym bardziej nie ma powodu, aby odcinać się od występowania w kodzie źródłowym śladów kulturowych. Informatyka nie straci przez to swojego charakteru.

W kodzie źródłowy występują zarówno pojęcia myśli teoretycznej, jak i językowe. Ujmując kwestię z perspektywy filozofii języka, na przykład myśli Cassirera, można by uznać, że ten wytwór zajmuje miejsce pośrednie w ciągu rozwojowym form

359 I. Lakatos, *Renesans empiryzmu we współczesnej filozofii matematyki*, [w:] *Współczesna filozofia matematyki: wybór tekstów*, red. R. Murawski, Państwowe Wydawnictwo Naukowe, Warszawa 2002, s. 233.

360 Stanowisko *quasi*-empiryzmu w matematyce było wspierane również przez E. R. Swart’a, Ph. J. Davis’a, R. Hersh’a, Ph. Kitcher’a.

361 W. Sady, *Spór o racjonalność naukową. Od Poincarégo do Laudana*, Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika, Toruń 2013, s. 305.

362 P. Davis, R. Hersh, *Świat matematyki*, Państwowe Wydawnictwo Naukowe, Warszawa 1994, s. 305.

363 T. Tymoczko, *Problem czterech barw i jego znaczenie filozoficzne*, [w:] *Współczesna filozofia matematyki*, red. R. Murawski, Państwowe Wydawnictwo Naukowe, Warszawa 2002, s. 337.

364 S. Krajewski, *Czy matematyka jest nauką humanistyczną*, Copernicus Center Press, Kraków 2011, s. 143.

symbolicznych pomiędzy językiem potocznym a naukowym. Filozof dostrzegał bowiem, jak pisze Bronisław Andrzejewski „wyraźną ić kontynuacyjną pomiędzy językiem a nauką”³⁶⁵. Nie da się oderwać poznania naukowego od powiązania z językiem, gdyż: „ta sama siła zasadnicza, która pozwala wyłonić się pojęciom językowym z pojęć »ogładowych« przeobraża ostatecznie te pierwsze w formę pojęcia »naukowego«”³⁶⁶. Języki programowania wykorzystują ponadto zarówno pojęcia językowe, jak i naukowe.

Perspektywa filozofii form symbolicznych pozwala umieścić w obszarze kultury język kodu źródłowego. Choć jego tekst, jak się zdaje, przesycony naukowością, nie brak w nim pojęć językowych, co zostanie zilustrowane przykładami w dalszej części pracy. Z kolei Ricœura wnosi do humanistycznych badań nad kodem źródłowym samą możliwość interpretowania tego wytworu. Dzięki koncepcji *distanciation*, która nie odrzuca stechnicyzowanego języka³⁶⁷, myśliciel ten otworzył drogę do refleksji nad nim w ramach hermeneutyki filozoficznej. Podjęcie rozważań nad językiem w kodzie źródłowym, z perspektywy myśli tych dwóch filozofów, wydaje się zatem nie tylko uprawnione, ale w pełni uzasadnione. Warto więc prześledzić w kolejnych rozdziałach jak w obszarze przestrzeni publicznej, artykulacji indywidualności i rozpoznawania standardów, może przejawiać się język jako dyskurs i forma symboliczna.

365 B. Andrzejewski, *Animal symbolicum* ..., s. 166.

366 E. Cassirer, *Philosophie der simbolischen...* III, s. 383, tłum. za B. Andrzejewski, *Animal symbolicum*..., s. 168.

367 A. Bajer, *Understanding of source code in language: Contribution of philosophical hermeneutics to the critical code studies*, „Digital Scholarship in the Humanities Oxford University Press”, 37, nr 2, 2021, s. 311.

2. Ustanowienie przestrzeni publicznej w języku

„Jestem pojemny, są we mnie miliony”
(*I am large, I contain multitudes.*)³⁶⁸
Walt Whitman

Działanie człowieka w języku urzeczywistnia się w wymiarze społecznym w rozmowie, gdy jakiś problem zostaje *umieszczony pomiędzy* rozmówcami. Jak zaproponował Charles Taylor w swojej teorii znaczenia „język stwarza coś, co można nazwać publiczną przestrzenią albo miejscem, z którego wspólnie przyglądamy się światu”³⁶⁹. Ten właśnie ponadindywidualny aspekt ukazał w koncepcji dyskursu Ricœur, a Cassirer uchwycił w ramach językowej formy symbolicznej. Zarówno w hermeneutyce dyskursu, jak i filozofii form symbolicznych uwidaczniają się bowiem wątki skupione wokół międzyludzkiej komunikacji: kształtowania się języka w środowisku społeczno-kulturowym, jego funkcjonowania na poziomie społecznym, czy tworzenia znaczenia w tym kontekście. Niewątpliwie ustanawianie przestrzeni publicznej w języku odbywa się odmiennie, gdy ujmie się je w ramy dwóch wspomnianych podejść filozoficznych. Mimo tego tropy zawarte w obu rzucają światło na rozumienie języka w kodzie źródłowym programów komputerowych. Jest to rys charakterystyczny dla wspomnianej już w pierwszym rozdziale rodziny teorii HHH, czyli odpowiadający temu, co zwykło się uznawać za kontynentalną filozofię języka.

Ricœur traktował język jako wydarzenie komunikacyjne. Struktura języka miała wchodzić w relację ze zdarzeniem i umożliwiać powstanie dyskursu³⁷⁰. Oznacza to, że analizie języka kodu źródłowego, może podlegać to, co wydarza się w kodzie źródłowym, chociażby w wymiarze społecznym. Oczywiście nie można pominąć wagi *langue*, czyli strukturalnych właściwości systemu, jakim jest sam język programowania, jednak rozważania skupione wokół niego wiązałyby się bardziej z dziedziną nauk inżynieryjno-technicznych. Dlatego też podejmując się problemu z perspektywy

368 W. Whitman, *Song of Myself (1892 version)*, <https://www.poetryfoundation.org/poems/45477/song-of-myself-1892-version> (dostęp dnia 01.03.2023); „Jestem pojemny, są we mnie miliony” za W. Whitman, *Pieśń o mnie*, [w:] *Pieśń o mnie i inne utwory*, tłum. A. Szuba, Wydawnictwo Miniatura, Kraków 1996, s. 106, lub inna znana wersja: „jestem wielki, składałem się z mnogości”.

369 Ch. Taylor, *Teoria znaczenia* ..., s. 157.

370 P. Ricœur, *Ku hermeneutyce krytycznej*..., s. 186.

humanistycznej, oczywistym wydaje się wysunięcie na pierwszy plan kwestii dyskursu.

Paralelny dylemat do powyższego trzeba rozstrzygnąć podejmując się rozważań nad językiem z perspektywy filozofii Cassirera. Można przyjąć, że języki programowania, co było już rozważane w poprzednim podrozdziale, są formą leżącą w ciągu rozwojowym pomiędzy formą symboliczną języka i nauki. Decydując się na ujęcie języków programowania jako przykładów pierwszej z wymienionych form lub zmierzając do tego, podążamy za wątkiem filozofii języka. Gdy zaś rozważa się je przez pryzmat naukowości, co oczywiście jest też ciekawe i użyteczne, prowadzi to do dociekań obejmujących bardziej filozofię nauki, a konkretnie – filozofię informatyki. Skoro jednak wyzwaniem, które rzucił Cassirer, było zdefiniowanie człowieka w kategoriach kultury, a praca dotyczy językowości, dyskusja będzie toczyć się wokół aspektów języków programowania, w których można dostrzec kulturowy wymiar.

Mając na względzie, to że wykorzystanie hermeneutyki dyskursu i filozofii form symbolicznych da pełniejszy obraz – bardziej oddający rzeczywistość, w jakiej powstaje i funkcjonuje kod źródłowy – można stwierdzić, że rozważania te wzbogacają studia nad praktykami kulturowymi i społecznymi związanymi z programowaniem. Język z założenia dysponuje wymiarami wychodzące poza reprezentację³⁷¹, więc kod źródłowy staje się nowym materiałem, dla tego rodzaju badań.

Z uwagi na fakt, że działania podejmowane przez programistów miewają właśnie aspekt wspólnotowy kod źródłowy będzie, analizowany jako społeczny wytwór. Jak ujął to Taylor, za pomocą języka umieszcza się coś w otwartej przestrzeni pomiędzy rozmówcami, tworząc możliwość wspólnego przyglądania się światu³⁷². *Terrą incognitą* w tej kulturoznawczej podróży staje się przestrzeń kodu źródłowego, a celem poznanie tego, co dzieje się w jego języku. Dokonuje się tam, na co wskazał Taylor, konstytuowanie przestrzeni publicznej w wymiarze ekspresji, ujawniania i konstytuowania³⁷³.

371 Ch. Taylor, *Teoria znaczenia...*, s. 170.

372 *Ibidem*, s. 157.

373 *Ibidem*, s. 161.

2.1. Wymiar ekspresji

Bycie z innymi w języku opiera się w olbrzymiej mierze na ekspresyjnym użyciu języka i jest to dostrzegalne w niektórych językach programowania. Ma ono więziotwórczą moc, gdyż poprzez ten wymiar ludzie doświadczają z innymi. Dzięki niemu możemy nawiązywać i podtrzymywać kontakt, a patrząc szerzej, konstruować wspólną przestrzeń publiczną. Warunkiem jest tutaj współrozumienie znaków ekspresyjnych, co wymaga przynależności do wspólnej kultury. Rzeczywistość społeczna jest definiowana wszak poprzez uniwersum symboliczne. Tę myśl Cassirera uzupełnić można o aspekt wspólnego doświadczania w języku, który jest obecny w filozofii Ricœura. Obie perspektywy mają swój odrębny wkład w rozważaniach o wyrażaniu się w kodzie.

W kontekście otoczenia społecznego osadzone są zarówno kontekst poznawczy, czyli nasze wcześniejsze doświadczenia, jak i ludzkie działania. Oba uwzględniają w sobie tzw. kontekst kulturowy, który pomaga w przeprowadzaniu trafnej interpretacji w ramach danej kultury³⁷⁴. Antropolog kulturowy, Edward Hall, stworzył koncepcję komunikowania opartą właśnie o uwzględnienie kontekstu w przekazie znaczenia wypowiedzi³⁷⁵. Na sposób porozumiewania wpływać mają zatem w różnym zakresie okoliczności towarzyszące, na przykład komunikaty niewerbalne, znajomość historii, czy hierarchii społecznej. Według badacza tak, jak istnieją kultury z tendencją do wysokokontekstowości (*high-context*, HC), czyli takie, w których kontekst wyznacza sposoby komunikowania, występują również kultury ciążące ku usytuowaniu się po przeciwnej stronie tego *continuum*, czyli niskokontekstowe (*low-context*, LC). Według tego badacza poznajemy poprzez *kontekstowanie* – ono umożliwia zwiększenie efektywności manipulowania potrzebnymi informacjami.

Wysokokontekstowość wiązać się ma w modelu Halla z „zawieraniem się większości informacji w fizycznym kontekście, bądź (...) zinternalizowaniu informacji w człowieku”³⁷⁶. Wykorzystuje się zatem w większym zakresie kontekst słów, na

374 A. Anczyk, H. Grzymała-Moszczyńska, A. Krzysztof-Świdorska, J. Prusak, *Wielokulturowość w pracy psychologa: etyka i praktyka w kontekście diagnozy klinicznej i pracy terapeutycznej*, „Psychologia Wychowawcza”, nr 15, 2019, s. 117.

375 A. Zięba, *Język a kultura. Komunikacja werbalna w wybranych polskich, brytyjskich i amerykańskich programach telewizyjnych*, „Investigationes Linguisticae”, 16, 2008, s. 252.

376 E. Hall, *Poza kulturą*, Państwowe Wydawnictwo Naukowe, Warszawa 2001, s. 95.

przykład intonacji, która może zmieniać znaczenia komunikatu. Ponadto, ma się to również wiązać z mniej dosłownym traktowaniem wypowiedzi. Taka komunikacja nie wprost jest bardziej charakterystyczna, chociażby na przykład, dla języka chińskiego. Ma przypominać „prowadzenie dialogu z całą historią”³⁷⁷ danej grupy. Opieranie zaś kontaktów głównie na informacjach zawartych w *kodzie bezpośrednim* definiuje niskokontekstowość i bliższe jest chociażby niemieckojęzycznym Europejczykom. Kontekstowość języka jest również zauważana sytuacyjnie. Nawet w obrębie amerykańskiej kultury, która ciąży ku bezpośredniości, w środowisku domowym występuje zwykle komunikacja wysokokontekstowa.

Model Halla, chociaż pojawiają się ostatnio kontrowersje w odniesieniu do stopnia jego empirycznej walidacji³⁷⁸, okazuje się użyteczny dla rozważań nad kodem źródłowym programów komputerowych. W dość jednoznaczny sposób można dominującą w nim kulturę umiejscowić w *continuum* kontekstowości. Według wspomnianego antropologa, w świecie naukowym promowany jest apolliński sposób myślenia będący przykładem niskokontekstowości³⁷⁹. Za Albertem Szent-Györgyi wskazuje on na dążenie w tym podejściu do perfekcjonizmu. Komunikacja LC cechuje się także ekonomicznością, szybkością oraz pewną bezosobowością. Dionizyjski sposób myślenia oparty o kreatywność, w ramach takiego sformalizowanego systemu skoncentrowanego na natychmiastowej efektywności, będzie natomiast eliminowany. Hall wskazuje na dziejowe podłoże tej niskokontekstowości:

„charakterystyczne niedomogi zachodniego systemu myślenia, wychodzącego z myśli filozoficznej starożytnej Grecji, są doskonałym przykładem dalekosiężnych konsekwencji wyboru spraw, w których uczestniczymy. Nasz sposób myślenia jest niezwykle arbitralny i powoduje, że chętniej przyglądamy się ideom i koncepcjom niżli wydarzeniom”³⁸⁰.

W rezultacie człowiek Zachodu wypracował niskokontekstowe systemy klasyfikacji w przeciwieństwie do ludowej taksonomii³⁸¹ i zwykł już w ten sposób porządkować świat.

377 *Ibidem*, s. 96.

378 P. Caron, *A Critique of Hall's Contexting Model A Meta-Analysis of Literature on Intercultural Business and Technical Communication*, „Journal of Business and Technical Communication”, 22, nr 4, 2008.

379 E. Hall, *Poza kulturą...*, s. 127.

380 *Ibidem*, s. 93.

381 Przykładem jak działa ludowa taksonomia byłoby tworzenie według reguł np. *Chińskiej encyklopedii*. Obecnie jest to metoda wykorzystywana do rozwijania kreatywnego myślenia, M. Jasiński, *Myślenie krytyczne i kreatywne. Sylabus*, WSB-NLU.

Nic zatem dziwnego, że właśnie produktem zachodniej myśli naukowej stało się programowanie.

Języki programowania stanowią egzemplifikacją języka, w którym komunikacja odbywa się z użyciem bardzo niewielkiej kontekstowości. Połączenie jednoznaczności algorytmów i precyzyjnie zdefiniowanej semantyki kształtuje językową przestrzeń społeczną kodu źródłowego. Wieloznaczność nie występuje w kodzie źródłowym, gdyż znaczenie wyrażen języków programowania jest ustalone. Niezgodność pomiędzy rozumieniem programisty a formalnym odwzorowaniem byłaby przykładem sytuacji wpłynięcia na *semantyczną rafę*³⁸². Warto zatem zastanowić się, czy w jednoznacznym świecie programowania może istnieć kultura rozumiana jako „system znaczeń i symboli poddanych interpretacji (hermeneutyczne koncepcje kultury, symboliczne formy i zasady kulturowego bytu)”³⁸³. Jak się zdaje, staje się to możliwe, jeśli traktujemy język jako dyskurs lub formę symboliczną. Kultura symboliczna według pierwszej koncepcji wyraża się poprzez zaangażowanie człowieka, w drugiej poprzez zapośredniczenie w symbolicznej naoczności³⁸⁴. W Ricœurowskim dyskursie język odnosi się do doświadczanej rzeczywistości – czegoś, co mamy o niej do powiedzenia³⁸⁵, według Cassirera za pomocą kulturowych aspektów, takich jak na przykład język, możemy dopiero poznać rzeczywistość.

Świat społeczny, w tekście kodu źródłowego, może wyrażać się w języku lub poprzez rezultat działania programu. My jednak pozostając w świecie tekstu, czyli w ramach podejścia *Critical Code Studies*, nie będziemy podążać w stronę fenomenologicznych *Software Studies*. Przedstawienie problemu przez pryzmat Ricœurowskiej hermeneutyki pozwoli ująć tekst kodu źródłowego w charakterze dzieła dyskursywnego³⁸⁶, z pomocą filozofii Cassirera natomiast jako egzemplifikację symbolicznej formy języka. Aby odsłonić ukrytą treść, ten wytwór kulturowy będzie zatem badany według reguł polegających na doszukiwaniu się „jedności procesu twórczego”³⁸⁷. Dlatego wspólne dzieło, jakim jest kod źródłowy trzeba poznawać

382 <https://edu.pjwstk.edu.pl/wyklady/jps/scb/wyklad5/section1.html> (dostęp dnia 18.03.2023).

383 J. Hańderek, *Pojęcia i definicje kultury*, [w:] *Filozofia kultury*, red. P. Mróz, WUJ, Kraków 2015, s. 25.

384 *Ibidem*, s. 27-28.

385 P. Ricœur, *Język jako dyskurs* ..., s. 92.

386 P. Ricœur, *Wyjaśnianie i rozumienie*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 163.

387 E. Cassirer, *Esej o człowieku*..., s. 157.

rozpatrując obecny w nim sposób myślenia programistów. Wobec czego badane są wspólne aktywności twórcze i samo wytwarzanie jako *energeia*³⁸⁸. Szlak ten jest podobny do zaproponowanego przez Ricœura dyskursywnego rozumienia języka, gdzie uwagą przykuwa, to co się dzieje, czyli jak tekst jest tworzony i odbierany.

Ekspresje w różnych językach programowania można by przyrównać do struktury powierzchniowej Noama Chomsky’ego, gdyż należą one do warstwy syntaktycznej. Wykonanie, czyli praktyczne użycie języka w konkretnych sytuacjach³⁸⁹ stanowi emanację struktury głębokiej – co uzasadnia tę analogię. Na możliwość ekspresji w przestrzeni publicznej kodu źródłowego wskazują różne sposoby wyrażania się w nim przez programistów. Zależą one na przykład od wybranego języka programowania, przyjętej konwencji programistycznej lub wykorzystania języków programowania do tworzenia sztuki. We wszystkich tych działaniach uwidacznia się przynależność do wspólnoty językowej.

Żeby unaocznić w czym przejawia się ekspresja, podczas kształtowania przestrzeni kodu źródłowego, w zależności od języka, warto sięgnąć do charakterystycznego przykładu. Będzie nim programowanie w języku *Java*. Na odrębność aktywności w ramach kultury kodu *Javy* wskazuje też pośrednio pełen rezerwy stosunek wśród programistów niskopoziomowych do tego języka.

Słownictwo w *Javie* przypomina to występujące w języku naturalnym. Klonowanie obiektów to *o.clone()* i stanowi skrót od angielskiego *object cloning*, a dla porównania w *PHP* stosujemy mniej familiarne $\$o2 = \o ³⁹⁰, do usuwania klucza w *Javie* wystarczy zwykle *remove* od angielskiego *remove by key*, a w *Pythonie* już *del h[k]*³⁹¹, zamiana typu danych na *string* w *Javie* następuje poprzez *toString* od słów *convert something to a string*, w *Perl* natomiast z użyciem `"" . e`³⁹². Mimo że wszystkie wymienione przykłady pochodzą z języków wysokiego poziomu, które zawierają elementy języka naturalnego w składni, to *Java* w tym względzie definitywnie się

388 S. Raube, *Kultura jako ekwilibrium. Krytyczna filozofia Ernsta Cassirera*, „IDEA – Studia nad strukturą i rozwojem pojęć filozoficznych” 26, 2014, s. 149.

389 N. Chomsky, *Zagadnienia teorii składni*, Zakład Narodowy Imienia Ossolińskich, Wrocław 1982, s. 15.

390 <http://rigaux.org/language-study/syntax-across-languages/bjcrntRfl.html> (dostęp dnia 30.03.2023).

391 <http://rigaux.org/language-study/syntax-across-languages/VrsDatTps.html> (dostęp dnia 30.03.2023).

392 <http://rigaux.org/language-study/syntax-across-languages/Strng.html> (dostęp dnia 30.03.2023).

wyróżnia. Można również porównać rozbudowanie wypowiedzi w języku *Ruby* do oszczędności języka *lambda calculus* (rys. 1).

<pre>>> success = true => true >> if success then 'happy' else 'sad' end => "happy" >> success = false => false >> if success then 'happy' else 'sad' end => "sad"</pre>	<pre>IF = -> b { -> x { -> y { b[x][y] } } }</pre>
--	---

Rys. 1. Porównanie obszerności (rozbudowania wypowiedzi) przykładowego fragmentu kodu napisanego w *Ruby* (czerpiącego z języka naturalnego) i odpowiadającego mu znaczeniowo kodu w języku *lambda calculus* (system formalny)³⁹³

Mówiąc o ekspresji w *Javie* istotną kwestią wydaje się tworzenie przestrzeni w kodzie źródłowym poprzez rozbudowane wypowiedzi. Niektórzy malkontenci zauważają, że wśród programistów tego języka wspólnym nawykiem jest na przykład pisanie dużej ilości kodu:

„wadą *Javy* jest kultura programowania, która wokół niej wyrosła. Programiści *Javy* mają tendencję do używania setek wierszy kodu tam, gdzie jeden by w zupełności wystarczył. Kiedy patrzy się na czyjś kod, można mieć trudności ze znalezieniem wiersza, która (pisownia oryginalna) rzeczywiście coś robi.”³⁹⁴

Krytyczne głosy na temat „gadatliwości” programistów *Javy* pochodzą zwykle od programistów niskopoziomowych. Języki wysokiego poziomu mają jednakowoż też swoich zagorzałych wielbicieli, aż takich, że niektórzy z nich proponują nawet używanie języków wysokopoziomowych do programowania urządzeń wbudowanych (*embedded devices*)³⁹⁵.

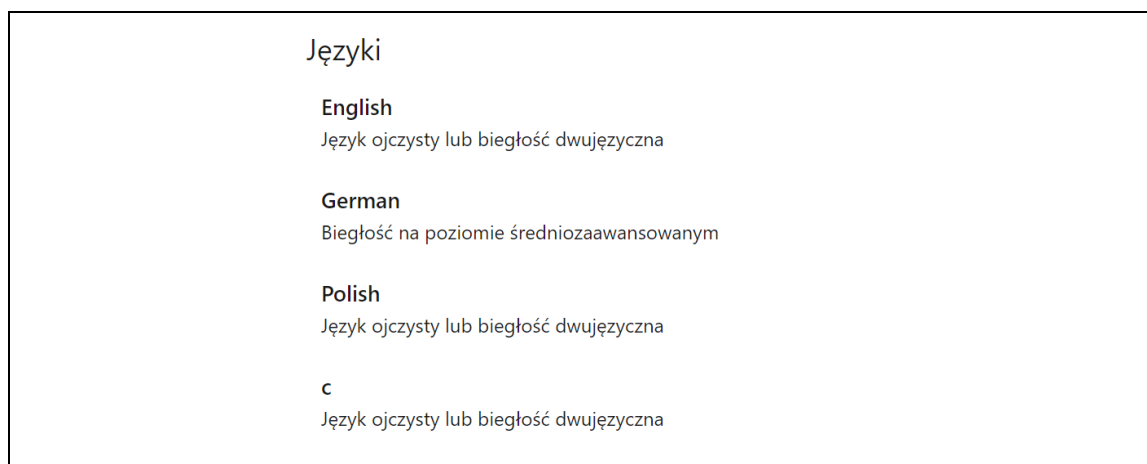
Wybór języka programowania wyraża się w uczestnictwie we wspólnej przestrzeni kulturowej, gdyż wpływa ona na dostęp do możliwych środków ekspresji. Programiści wybierając, w jakim języku lub językach będą pracować decydują się

393 T. Stuart, *Understanding Computation: From Simple Machines to Impossible Programs*, O'Reilly Media Inc, Sebastopol 2013, s. 169-170.

394 J. E. Steinhart, *Sekretne życie programów. Zrozumieć komputery – tworzyć lepsze kody*, Państwowe Wydawnictwo Naukowe, Warszawa 2021, s. 452.

395 M. Carro, J. Morales, H. Muller, *High-Level Languages for Small Devices: A Case Study*, CASES '06, Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems, Association for Computing Machinery, Nowy York 2006, s. 272, http://cliplab.org/papers/carro06:stream_interpreter_cases.pdf (dostęp dnia 18.03.2023).

zarazem na stosowanie podobnych środków wyrazu co inni członkowie danej grupy językowej. Postawa wobec przestrzeni publicznej wybranego języka programowania bywa nawet wyrażona wprost. Za przykład przynależności do językowej wspólnoty może posłużyć wpisywanie w profilu zawodowym na międzynarodowy serwisie zawodowo-biznesowym, języka programowania jako języka ojczystego, lub też biegłości w nim na poziomie dwujęzyczności (rys. 2), mimo że dany portal oferuje w też możliwość wpisania frazy „pełna biegłość zawodowa”. Wspomniane wyrazy przynależności są w prawdzie wyrażone w języku naturalnym, ale potwierdzają identyfikację kulturową jednostki z treściami wyrażanymi w danym języku. Widać na tym przykładzie formowanie się załączków tożsamości kulturowej w oparciu o sztuczny język i zaangażowanie w jego przestrzeń publiczną.



Rys. 2. Fragment profilu LinkedIn Janusza Jeremiasza Filipiaka, dyrektora działu badawczo-rozwojowego w firmie Comarch³⁹⁶

Za przykład na ekspresyjność w językach programowania posłużyć mogą też konwencje programistyczne. W zwyczajach społecznych nie ma „nic koniecznego ani uniwersalnego, każde społeczeństwo kształtuje je po swojemu”³⁹⁷, a programistyczne społeczności językowe kierują się także wypracowanymi zestawami reguł. Chociaż łamanie ich nie podlega sankcjom, czyli mimo złamania konwencji kod będzie się kompilować, to sytuacja nieprzestrzegania konwencji programistycznych wiązać się będzie z poczuciem dyskomfortu innych programistów. Skoro odstępcom nie zagrazi *vendetta*, powstaje pytanie: co miałoby powodować, że programiści będą trzymać się konwencji językowej? Najprawdopodobniej jest tym potencjalna krytyka

396 https://www.linkedin.com/in/jeremiaszfilipiak/pl?trk=public_post_feed-actor-name (dostęp dnia 03.03.2023).

397 P. Sztompka, *Socjologia. Analiza społeczeństwa*, Znak, Kraków 2007, s. 269.

współpracowników czytających „brzydki” kod i oczekiwania w środowisku zawodowym. Demonstrowanie kompetencji poprzez przestrzeganie konwencji może być ponadto nagradzające, nawet jeśli ogranicza się do samozadowolenia z dobrze wykonanej pracy.

Łamanie konwencji można przyrównać do niezachowywania właściwego dystansu społecznego. W kodzie, na przykład, istnieje zalecana w *Javie* nieprzekraczalna długość 80 znaków w linii³⁹⁸. Do norm zwyczajowych należą, chociażby występujące w książkach podziękowania i dedykacje dla bliskich i w zbliżonej konwencji dedykowania utworów, powstał na przykład poniższy komentarz do kodu. Mąż żartobliwie ofiaruje w nim żonie swoją pracę, mimo że jest świadom niedostatków dzieła:

```
// I dedicate all this code, all my work, to my wife, Darlene, who will  
  
// have to support me and our three children and the dog once it gets  
  
// released into the public.399
```

Chociaż zwyczaje programistów na ogół nie są kodyfikowane, to bywają wyjątki. Takim podręcznikiem *savoir-vivre*’u są *coding conventions*, czyli niesformalizowane dobre praktyki⁴⁰⁰. W większości przypadków stosowanie się do nich nie jest wymagane przez kompilator/interpreter. Praktyką tą jest, na przykład, kierowanie się najlepszą czytelnością⁴⁰¹, co warto czynić konsekwentnie w całości tekstu⁴⁰². Złamanie konwencji programistycznej zwykle nie wpływa na działanie programu, jedynie na jego przejrzystość, możliwość ponownego użycia i utrzymanie. Robert Martin nazywa dojrzałością do pracy w zespole, zachowanie programistów polegające na uznaniu prymatu wspólnie przyjętej konwencji ponad swoje preferencje⁴⁰³.

Do przykładowych konwencji nazewniczych (*naming convention*) należą formy zapisu:

398 <https://www.cs.swarthmore.edu/~newhall/unixhelp/javacodestyle.html> (dostęp dnia 01.03.2023).

399 <https://stackoverflow.com/a/186967> (dostęp dnia 25.02.2023).

400 <https://learn.microsoft.com/pl-pl/dotnet/csharp/fundamentals/coding-style/coding-conventions> (dostęp dnia 04.03.2023).

401 <https://perldoc.perl.org/perlstyle> (dostęp dnia 01.03.2023).

402 R. Martin, *Czysty kod*, Helion, Gliwice 2014, s. 303.

403 *Ibidem*, s. 310.

- *camelCase*⁴⁰⁴ inaczej *lower camel case*, *medial capital*, lub *dromedary case*⁴⁰⁵ np. *camelCase*,
- *PascalCase* inaczej *Upper Camel Case*⁴⁰⁶, *WikiCase* lub *studly case*⁴⁰⁷ np. *PascalCase*,
- *snake case* inaczej *pathole case*⁴⁰⁸ np. *snake_case*,
- *upper snake case* inaczej *screaming snake case*, *macro case*⁴⁰⁹ lub *constant case*⁴¹⁰ np. *UPPER_SNAKE_CASE*,
- *pascal snake case*⁴¹¹ np. *Pascal_Snake_Case*,
- *kebab case*⁴¹² lub inaczej *spinal case*⁴¹³ np. *kebab-case*,
- *screaming kebab case*, inaczej *cobol case*⁴¹⁴ np. *SCREAMING-KEBAB-CASE*,
- *train case* lub inaczej *HTTP Header Case*⁴¹⁵ np. *Train-Case*
- *lower dot case* np. *java.management.rmi*⁴¹⁶ ,
- *flat case* np. *flatcase*
- *upper flat case* np. *UPPERFLATCASE*

Trzeba zaznaczyć, że istnieją języki, które wymagają zastosowania konkretnej konwencji jak na przykład *Java* (*case sensitive convention*).

Konwencje określają także, na przykład długości linii, długości funkcji, zaznaczanie bloków, wcięcia, nawiasy, organizacje folderów, komentarze, *white space*^{417,418,419}. W ich ramach można określić również dobre praktyki programistyczne, wzorce projektowe, praktyczne zasady tworzenia kodu. Jest istotne, aby zachować

404 Przeniesione do języka naturalnego np. w nazwie produktu iPhone, czy firmy eBay.

405 <https://juniortoexpert.com/en/naming-convention/> (dostęp dnia 06.03.2023).

406 Przeniesione do języka naturalnego np. w nazwie YouTube, PlayStation. Występuje w *Javascript*, *C#*, *java*.

407 <https://juniortoexpert.com/en/naming-convention/> (dostęp dnia 20.03.2023).

408 Używana też często w nazwach plików i folderów.

409 <https://medium.com/nerd-for-tech/programming-case-types-explained-143cad3681e3> (dostęp dnia 01.03.2023).

410 <https://juniortoexpert.com/en/naming-convention/> (dostęp dnia 01.03.2023).

411 <https://juniortoexpert.com/en/naming-convention/> (dostęp dnia 01.03.2023).

412 Występuje w Lisp, częsty we front-endzie.

413 <https://juniortoexpert.com/en/naming-convention/> (dostęp dnia 01.03.2023).

414 <https://juniortoexpert.com/en/naming-convention/> (dostęp dnia 01.03.2023).

415 <https://medium.com/nerd-for-tech/programming-case-types-explained-143cad3681e3> (dostęp dnia 01.03.2023).

416 <https://www.techtarget.com/whatis/definition/CamelCase> (dostęp dnia 01.03.2023).

417 <https://coders.school/post/konwencje-w-skrocie/> (dostęp dnia 01.03.2023).

418 S. McConnell, *Kod doskonały. Jak tworzyć oprogramowanie pozbawione błędów*, Helion, Gliwice 2010, s. 100.

419 R. Martin, *Czysty kod...*, s. 310.

jednolitość w całym projekcie, natomiast między projektami mogą występować różnice. Zalecenia dotyczące pisania programów są przekazywane w przeróżnych *guidelines*, więc stanowią przejaw komunikacji niskokontekstowej.

Komunikacja niskokontekstowa, a więc charakterystyczna dla obszaru kodu źródłowego, była uznawana przez niektórych badaczy kultury za przyczynę uniemożliwiająca tworzenie sztuki. Dla wspomnianego już Edwarda Halla, autora pojęcia kontekstowości, uznaną sztukę miała zawsze cechować wysokokontekstowość, w przeciwieństwie do *złej sztuki*, „która natychmiast ujawnia swój komunikat” i „nie ma szans na przetrwanie”⁴²⁰. Mimo tak niesprzyjających warunków z pomocą języków programowania podejmowane są całkiem udane próby poetyckie i może trochę mniej udane projekty wizualne. Przykład pierwszych stanowią chociażby wiersze⁴²¹ Mary Ann Breeze, bazujące na pomieszaniu języka programowania z naturalnym i wynikających z tego implikacjach kulturowych⁴²². Bardziej rozpoznawana w środowisku programistycznym jest natomiast kompilująca się poezja pisana w języku Perl przez Sharon Hopkins⁴²³ (rys. 3). Sztuki wizualne reprezentowane w wytworach stworzonych przy pomocy symboli ASCII niestety wypadają na tym tle dość trywialnie⁴²⁴. Niektóre z prac programistycznych są jednak doceniane w świecie sztuki i wystawiane na aukcjach algorytmów⁴²⁵. Mają przy tym znaczenie kulturowe, jak na przykład upamiętniająca powstanie nowego języka odręczna wersja programu *Hello World*, zapisana na serwetce przez Briana Kernighana w języku C⁴²⁶.

420 E. Hall, *Poza kulturą...*, s. 96.

421 http://ilovepoetry.org/wp-content/uploads/2012/02/cross.ova_.ing_.png (dostęp dnia 01.03.2023),

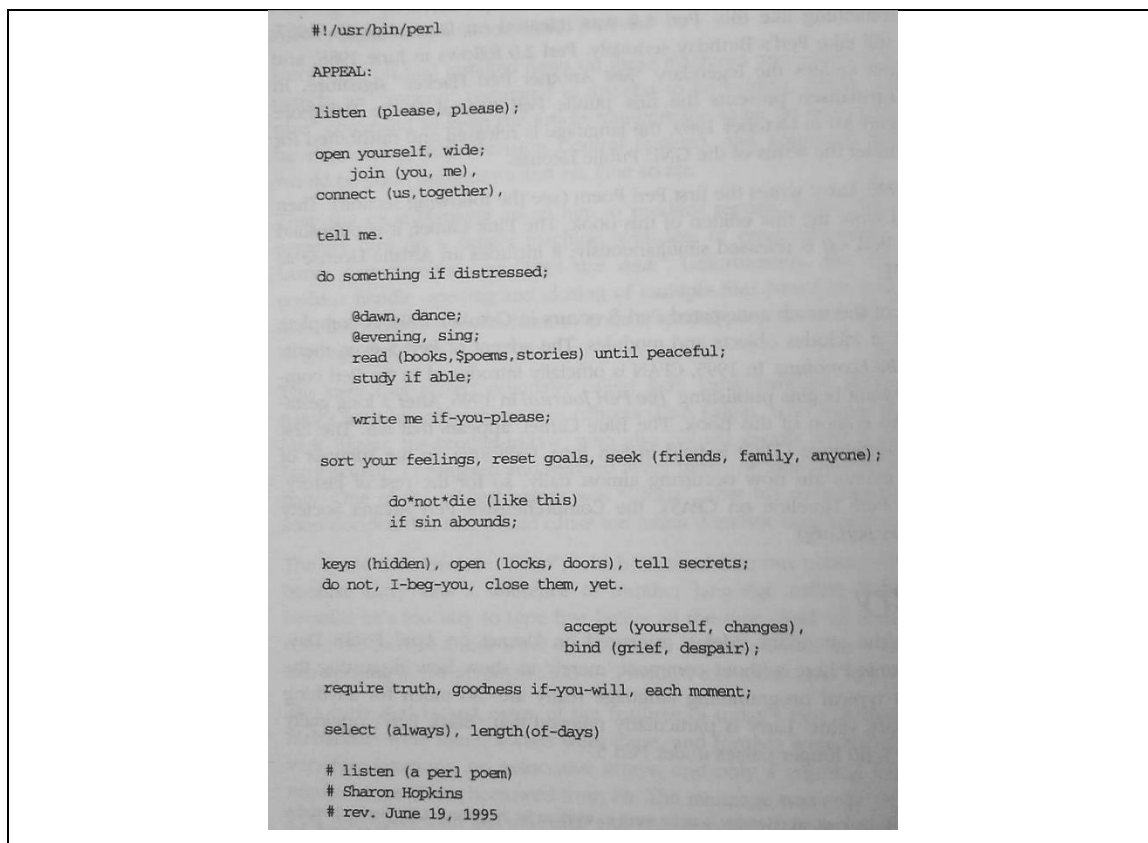
422 F. Cramer, *Digital Code...*

423 S. Hopkins, T. Corporation, *Camels and needles: computer poetry meets the Perl Programming Language*, Usenix Winter 1992 Technical Conference, http://www.digitalcraft.org/iloveyou/images/Sh.Hopkins_Perl_Poetry.pdf (dostęp dnia 01.03.2023).

424 Przykładowa praca konkursowa w „The International Obfuscated C Code Contest”. Kod źródłowy zapisany w formie ASCII art. <http://www.ioccc.org/1995/heathbar.c> (dostęp dnia 24.02.2023).

425 https://www.artsy.net/auction/the-algorithm-auction?sort=sale_position (dostęp dnia 01.03.2023).

426 <https://www.artsy.net/artwork/brian-kernighan-hello-world> (dostęp dnia 01.03.2023).



Rys. 3. Wiersz Sharon Hopkins⁴²⁷

Powstają także wiersze o bardziej wysublimowanej formie niż te wspomniane, oddające charakterystyczną strukturę danego języka programowania lub wykorzystujące zabawy językowe, a równocześnie kompilujące się tak, że w efekcie egzekucji kodu otrzymujemy jakiś znaczący efekt nawiązujący do treści wiersza⁴²⁸. Przykładowo komendy w tekście „Extract from nuane” należącym właśnie do nurtu poezji generatywnej, mogą podlegać egzekucji na różne sposoby⁴²⁹. Zabieg ten przypomina twórczość Paula Cezanne’a i jego seryjne przedstawianie góry Sainte Victoire, po to by uchwycić zmieniający się widok. Jak komentował ten malarski zabieg Ricœur, otrzymujemy w ten sposób „specyficzną odpowiedź na wyjątkowe pytanie”⁴³⁰. Podobnie do wspomnianej góry kod źródłowy wiersza generatywnego

427 L. Wall, T. Christiansen, J Orwant, *Programming Perl. Third Edition*, O'Reilly, Sebastopol 2000, s. 648.

428 G. Cox, A. McLean, A. Ward, *The Aesthetics of Generative Code. Generative Art 00 conference*, Politecnico di Milano, Milan 2001, https://www.academia.edu/10519146/The_Aesthetics_of_Generative_Code, (dostęp dnia 21.02. 2023).

429 *Ibidem*, s. 8.

430 P. Ricœur, *Krytyka i przekonanie...*, s. 255.

również domaga się ekspresji. Podczas egzekucji programu, która następuje na różne sposoby, dochodzi do oddania sprawiedliwości temu, co wyjątkowe⁴³¹.

Jest zatem możliwa ekspresja artystyczna w kodzie źródłowym, jednak wymieniona twórczość eksperymentalna z uwagi zapewne na wymóg podstawowej znajomości programowania, prawdopodobnie jeszcze długo nie będzie rozpoznawana przez krytyków i publiczność jako dzieła artystyczne. Ekspresja bowiem jest w tego typu sztuce nakierowana na kontakt z drugim człowiekiem i odpowiadający na nią odbiorca jest niezbędny, by utwór zrealizował się jako dzieło. Wiersz istnieje wszakże, jak zauważył Ricœur, już wcześniej w swojej możliwości do przyszłego rozpoznania - *monstracji*⁴³², ale wymaga kontaktu z odbiorcą. Ekspresja w przestrzeni publicznej utworu bazuje bowiem na komunikowalności. Francuski myśliciel zauważył, że to, co powiedziane w dziele sztuki, stanowi w prawdzie pojedyncze wyzwanie i rozwiązanie, ale przez to, że indywidualne doświadczenie zostaje sproblematyzowane, staje się wspólne⁴³³. „W niepowtarzalności tkwi coś uniwersalnego”⁴³⁴ - zaznaczył filozof.

Rekonstruując myśl estetyczną Cassirera, można komunikacji w sztuce przypisać intersubiektywność, ponieważ „gdyby dzieło sztuki było tylko kaprysem i szaleństwem indywidualnego artysty, nie miałoby tej powszechnej komunikatywności”⁴³⁵. Sztuka nie stanowi jedynie odtworzenia życia wewnętrznego, lecz jest pomostem między „Ja» i innym”⁴³⁶, subiektywnością i obiektywnością oceny. Mediacja ta zachodzi pomiędzy tym, co receptywne i twórcze w nas oraz tym, co racjonalne i emocjonalne w człowieku⁴³⁷. Dzieło sztuki według filozofa, by zaistnieć wymaga struktury teleologicznej⁴³⁸, a ta właśnie wybija się na pierwszy plan w poezji tworzonej w językach programowania, gdyż program zawsze tworzony jest w określonym celu. Interpretacja i odtworzenie wyprzedzają zatem ekspresję. Zachwalane przez filozofa kontemplowanie i tworzenie form stanowią tym samym podstawę twórczości artystycznej wyrażanej w kodzie źródłowym. Wątpliwe jednak, aby można

431 *Ibidem*, s. 255.

432 J.M. Brohm, P. Ricœur, *Interview with Paul Ricœur, Professor Emeritus*, „Philosophy and Social Criticism”, 36, nr 8, 2010, s. 937.

433 *Ibidem*, s. 943.

434 P. Ricœur, *Doświadczenie estetyczne*, [w:] P. Ricœur, *Krytyka i przekonanie. Rozmowy z Francois Azouvim i Markiem de Launay*, tłum. i oprac. M. Drwięga, Wydawnictwo KR, Warszawa 2003, s. 256.

435 E. Cassirer, *Esej o człowieku...*, s. 279.

436 E. Cassirer, *Logika nauk...*, s. 127.

437 B. Trochimska-Kubacka, *Koncepcja sztuki ...* s. 88.

438 E. Cassirer, *Esej o człowieku...*, s. 273.

było je powiązać z *katharsis*, „gdzie rzeczy i zdarzenia zaczynają tracić swój ciężar materialny”⁴³⁹. Sztuka jako interpretacja rzeczywistości, według Cassirera⁴⁴⁰, powinna być bowiem doznawana za pośrednictwem formy zmysłowej⁴⁴¹. Tymczasem w przypadku utworów napisanych w językach programowania jest to głównie proces myślowy.

Mediacyjna rola sztuki przedstawiona przez Cassirera, ujmuje jednak interpretację w niej czynioną jako proces mniej dynamiczny niż widział go Ricœur. Francuski filozof kwestią piękna i sztuki zajął się dość późno w swojej twórczości, a mianowicie w okresie *hermeneutyki czasu i narracji*⁴⁴², kiedy wprowadził pojęcia trzech figuracji. Kiedy prefiguracja stanowi przedrozumienie, konfiguracja narracyjność, a refiguracja restrukturyzuje świata osoby czytającej, pojawia się wtedy element konfrontacji z czytelnikiem i zmiany struktury jego świata⁴⁴³. Żeby to jednak mogło nastąpić, odbiorca musiałby mieć minimalne kompetencje w zakresie znajomości języków programowania. Takie „wtargnięcie na powrót do świata jest niezbędne dla sztuki, aby nie stała się naiwna”⁴⁴⁴, jak ostrzegał Ricœur, inaczej kompilowany wiersz będzie jedynie fraszką niskich lotów. Filozof nie odniósł się do emocjonalności czy piękna definiując sztukę, lecz ontologizował ją, wskazując na jej rolę odsłaniania prawdy⁴⁴⁵.

Ani Ricœur, ani Cassirer nie zajmowali się estetyką *per se*, toteż nie rozwinęli swoich własnych koncepcji piękna. Pierwszy z wymienionych filozofów podkreślał, że jest ono siłą sprawczą tworzącą nowe⁴⁴⁶. Wspólnotowość tego doświadczenia miała stanowić podążanie za powstałym w czasie poczuciem piękna, indywidualność zaś realizować się w nim poprzez rozstrzyganie gry między wyobraźnią a rozumem⁴⁴⁷. Cassirer również zauważał w tym doświadczeniu przejawy czegoś na kształt ludzkiej aktywności tyle, że ujętej w formie estetycznej⁴⁴⁸. Ludzie według niego nie są biernymi

439 *Ibidem*, s. 287.

440 Podejście to zostało nawet nazwane estetyką hermeneutyczną przez Birgit Reck, za B. Trochimska-Kubacka, *Koncepcja sztuki ...*, s. 87.

441 E. Cassirer, *Esej o człowieku...*, s. 302.

442 obejmującej „Temps et récit. Tome I”, „Temps et récit. Tome II. La configuration dans le récit de la fiction”, „Temps et récit. Tome III. Le temps raconté”, wydane od 1983-85 roku.

443 P. Ricœur, *Doświadczenie estetyczne*, [w:] *Krytyka i przekonanie ...*, s. 248.

444 *Ibidem*, s. 251.

445 T. Kurata, *Poza ontologią: Paul Ricœur i kwestia sztuki*, „Sztuka i Filozofia”, 22-23, 2003, s. 52.

446 J.M. Brohm, P. Ricœur, *Interview with Paul ...*, s. 939.

447 P. Ricœur, *Doświadczenie estetyczne*, [w:] *Krytyka i przekonanie ...*, s. 257.

448 E. Cassirer, *Esej o człowieku...* s. 304-305.

odbiorcami piękna, aby je odczuć współpracują bowiem z twórcą. Odbywać się to ma poprzez sympatyzowanie z uczuciami autora i wnikanie w jego działalność⁴⁴⁹. Cassirer opierał bowiem rozpoznanie piękna równorzędnie na akcie osądu i kompetencji⁴⁵⁰, jak i uczuciu.

Możliwość wyrażania piękna, w kontekście przestrzeni publicznej jest uzasadniona w przekonaniach obu filozofów. Realizacja tej kategorii estetycznej wymaga porozumienia z twórcą w intelektualnym rozeznaniu, które czynią odbiorcy. Zauważyć można bliskość owych koncepcji z Kantowskim estetycznym intelektualizmem. Podkreślenie aspektu doświadczenia rozumowego nie powinno jednak sprawić wrażenia, że zaprzecza się zmysłowości w rozważaniach estetycznych Ricœur'a i Cassirera, lecz ma za cel ułatwić rozmyślanie nad pięknem kodu źródłowego. Uchwycenie momentów dyskursywnych w przeżyciu estetycznym może być satysfakcjonujące, chociaż towarzyszące temu doznana niektórzy uznaliby za paraestetyczne, a rozpoznane w tych sytuacjach piękno nazwali jedynie *hipotetycznym*⁴⁵¹. Dostrzeżenie, że doświadczenie piękna kodu ma poznawczy potencjał, nie stanowi też „zemsty intelektu na sztuce”⁴⁵², rodzaju odwetu, o którym pisała Susan Sontag, lecz umożliwia kontemplację estetyczną tekstów nieoddziałujących na zmysły.

Pojęcie piękna zwykło się odnosić do doświadczeń zmysłowych, toteż piękny kod bywa przedstawiany jako kod przyjemny. Poznanie intelektualne w tej sytuacji wtórnie oddziałuje na zmysły, gdyż „czytanie (kodu źródłowego, *przyp. aut.*) powinno wywołać uśmiech na twarzy, podobnie jak dobrze skomponowana muzyka, czy dobrze zaprojektowany samochód”⁴⁵³. Wartości estetyczne można osadzać też na własnościach intelektualnych, tak jak bywa w obszarze literatury i na własnościach emocjonalnych – w poetyce⁴⁵⁴. Idąc tym tropem, piękno intelektualne⁴⁵⁵ ma możliwość wyrażać się także w tekstach zapisanych w językach programowania. W kodzie źródłowym programu,

449 *Ibidem*.

450 *Ibidem*.

451 M. Gołaszewska, *Estetyka współczesności*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2001, s. 14.

452 S. Sontag, *Przeciw interpretacji i inne eseje*, Wydawnictwo Karakter, Kraków 2018, s. 17.

453 R. Martin, *Czysty kod...*, s. 29.

454 Tatarkiewicz, *Droga przez estetykę*, Warszawa 1972, s. 87.

455 H. Elzenberg, W. Tatarkiewicz: *Historia Estetyki*, „Ruch Filozoficzny”, 21, nr 1, 1962, s. 32-33. Autor postuluje również objęcie terminem piękna/wartości estetycznej dodatniej piękna moralnego.

można ponadto dostrzec inne wartości estetyczne: takie jak brzydotę⁴⁵⁶, czy komizm.

Koncepcja pięknego kodu upowszechniła się w środowisku programistycznym, za sprawą takich klasycznych już pozycji wydawniczych jak „Piękny kod”⁴⁵⁷, „Czysty kod”⁴⁵⁸, czy „Refaktoryzacja. Ulepszanie struktury istniejącego kodu”⁴⁵⁹. Wraz z dostrzeżeniem, że jakość oprogramowania ma związek z jego efektywnością, znacząco wzrosło w XXI wieku zainteresowanie pięknym/czystym/dobrym/nieśmierdzącym⁴⁶⁰ kodem. Nazwy te stosowane są na określenie kodu charakteryzującego się walorami istotnymi jakościowo. Właśnie poprzez zauważenie tej cechy odbywa się rozpoznanie piękna w kodzie. Wybór sposobów ekspresji przejawia się, chociażby w stylu kodowania, który bywa narzucony w ramach wymagań danego projektu.

Najbliżej pięknemu kodowi do dobrze napisanego eseju, jak zauważył autor języka Ruby Yukihiro Matsumoto w tekście pod sugestywnym tytułem „Traktując kod jako esej”⁴⁶¹. Główna różnica według niego, polegać miałyby na tym, że w eseju czytelnika interesuje to, o czym tenże jest, natomiast programista i użytkownik skupiają się, na tym, co program robi. Oba zawierają w sobie przedstawiony „jakiś” sposób myślenia o problemie. Można o kodzie źródłowym powiedzieć, że podobnie do pisarstwa eseistycznego i w nim „swobodnie rozwija (się, *przyp. aut.*) interpretację jakiegoś zjawiska lub dociekanie problemu”⁴⁶². Piękno w kodzie źródłowym rozpoznać można w formie, czyli specyficznej strukturze. Charakteryzować ma się ona zwieżłością, konserwatyzmem, prostotą, elastycznością i równowagą⁴⁶³, czyli czymś, co można by nazwać estetycznymi wartościami słabymi⁴⁶⁴, towarzyszącymi prymarnej mocnej wartości piękna. Czynniki formalne jest niezbędny, by uzyskać syntetyczną jedność⁴⁶⁵. Cassirer uważał, że człowiek od zawsze organizował swoje doświadczenie w formie. Porządkowania w języku miał dokonywać nieświadomie, w nauce zaś w

456 Brzydota jako wartość prymarna mocna za M. Gołaszewską, *O naturze wartości estetycznych*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 1986, s. 108-109.

457 *Piękny kod. Tajemnice mistrzów programowania*, red. A. Oram, G. Wilson, Helion, Gliwice 2008.

458 R. Martin, *Czysty kod...*

459 M. Flower, *Refaktoryzacja. Ulepszanie struktury istniejącego kodu*, Helion, Gliwice 2019.

460 *Ibidem*, s. 81.

461 Y. Matsumoto, *Traktując kod jako esej*, [w:] *Piękny kod. Tajemnice mistrzów programowania*, red. A. Oram, G. Wilson, Helion, Warszawa 2008, s. 495.

462 M. Głowiński, *Słownik terminów literackich*, red. S. Sławiński, Zakład Narodowy im. Ossolińskich, Wrocław 1988, s. 140.

463 *Piękny kod. Tajemnice mistrzów...*, s. 499.

464 M. Gołaszewska, *O naturze wartości estetycznych*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 1986, s. 108-109.

465 E. Cassirer, *Esej o człowieku...*, s. 379.

sposób zamierzony⁴⁶⁶. Ponieważ jednak język w kodzie źródłowym jest wyrazem formy pośredniej między językiem a nauką, stąd też widoczna jest planowość w tym jak posługują się nim programiści.

Odkrycie regularności przez człowieka związane jest według Cassirera z „odsłonięciem tajemnicy piękna”⁴⁶⁷ tak jak chcieli tego pitagorejczycy. Z kontekstem języków programowania wiążą się także wymagania kompozycyjne. Przykładowo, o ile w języku C++ formatowanie kodu za pomocą wcięć nie wpływa na jego kompilację, to preferowane jest jednak postępowanie zgodnie z wytycznymi wybranego stylu⁴⁶⁸. Struktura piękna realizuje się w dążeniu do porządku, gdyż „czysty kod, jest zaprzeczeniem zła, próbą uporządkowania chaosu”⁴⁶⁹. Stosuje się zatem chociażby wspomniane już podziały na pliki i katalogi, czy standardy kodowania: komentarze, konstrukcje programistyczne, formatowanie kodu, konwencje nazewnictwa⁴⁷⁰.

Harmonijność w kodzie źródłowym uwidacznia się w standaryzowanym nazewnictwie czy wcięciach. Niezwykle ważny wydaje się też umiar, czyli chociażby dostosowanie stopnia elastyczności rozwiązania do danego zastosowania (jeśli fragment miałby być kiedyś rozszerzany). Dostosowanie komentarzy do poziomu skomplikowania kodu także jest przejawem tej cnoty, dlatego na przykład dla zachowania czytelności nie komentuje się kodu, z którego jasno wynika, co tenże robi. Ideał stanowiłby z tej perspektywy kod samokomentujący.

Ponieważ piękny kod „wygląda jakby był napisany przez kogoś, komu na nim zależy”⁴⁷¹, jest przeto wyrazem zaangażowania programisty. Doświadczenie i ekspresja poprzez piękno kodu wiążą się zatem z profesjonalizacją działań w tym zawodzie. Powinno się dbać o czytelność i elegancję tekstu⁴⁷². Brzydota stanowi natomiast niezamierzony efekt działania programisty. Mowa o niej, gdy kod jest mało spójny, nieczytelny oraz nisko wydajny. W kodzie źródłowym piękno i brzydota, jako

466 *Ibidem* s. 383.

467 *Ibidem*, s. 384.

468 https://www.state-machine.com/doc/AN_QL_Coding_Standard.pdf (dostęp dnia 12.03.2023);
<https://geosoft.no/development/cppstyle.html> (dostęp dnia 12.03.2023);
<https://web.archive.org/web/20150907011528/http://google-styleguide.googlecode.com/svn/trunk/cppguide.html> (dostęp dnia 12.03.2023).

469 R. Martin, *Czysty kod...*, s. 16.

470 *Ibidem*,

471 Fragment wypowiedzi Michaela Feathersa za R. Martin, *Czysty kod...*, s. 31.

472 R. Martin, *Czysty kod...*, s. 27.

podstawowe wartości estetyczne – tzw. prymarne mocne⁴⁷³ – mogą tworzyć komplementarną parę. Powstał nawet termin określający kod wymagający refaktoryzacji jako brzydko pachnący⁴⁷⁴.

Brzydota wywołuje przykre przeżycia estetyczne zwane *code smell/bad smell*, czyli smrodem kodu⁴⁷⁵. Tak zwane „brzydkie zapaszki”⁴⁷⁶ wywołane źle zaimplementowanym kodem skutkować miałyby nieprzyjemnym wrażeniem u osób czytających kod. Sama zaś nazwa pochodzi od skojarzenia z brudną, dziecięcą pieluchą⁴⁷⁷. Brudny kod podlega czyszczeniu, czyli refaktoryzacji. Oznacza to poprawę jego jakości, bez zmiany funkcjonalności, co ponadto powoduje przyspieszenie programowania⁴⁷⁸. Mika Mantyla wydzielił aż 70 różnych „smrodów” w tym kontekście. Dla przykładu, kategoria wzdęcia (*the bloaters*) zawiera długie metody, duże klasy⁴⁷⁹. Martin Fowler używa nawet bardziej obrazowych nazw i do nieprzyjemnych kompozycji zapachowych zalicza, na przykład „falę uderzeniową”⁴⁸⁰, „zazdrosne funkcjonalności”⁴⁸¹, czy „opętanie typami prostymi”⁴⁸². Równie dosadnie, bo gniciem (*code rot*) nazwane jest pogorszenie kodu następujące wraz z upływem czasu. Użycie słowa „brzydki” zdarza się również na określenie wyglądu zbyt skomplikowanego kodu gdzie jest „za dużo niepotrzebnych ładowań i przechowań”⁴⁸³. Opisywane sposoby wyrażania się w przestrzeni publicznej za pomocą wartości estetycznych są u wspólnione w tym znaczeniu, że programista nie ma kierować się jedynie własnym gustem, ale współdzielonym i akceptowanym w danej grupie⁴⁸⁴.

W kodzie źródłowym ujawnia się jeszcze jedna wartość estetyczna mianowicie komizm. Tematyka humoru była raczej rzadko podejmowana przez filozofów, jednakowoż powstało parę interesujących teorii, jeśli o tę kwestię chodzi. Mało

473 M. Gołaszewska, *O naturze wartości...*, s. 108-109.

474 M. Flower, *Refaktoryzacja. Ulepszanie...*, s. 81.

475 spotykane tłumaczenie wyrażenia *smell code* jako *zapach kodu* wydaje się nieadekwatne. *Code smell* jest bowiem przykrym zapachem, jednoznaczny jest tu wydźwięk negatywny.

476 M. Flower, *Refaktoryzacja. Ulepszanie...*, s. 81.

477 *Ibidem*.

478 *Ibidem*, s. 62.

479 Mäntylä, M. V., C. Lassenius, *Subjective Evaluation of Software Evolvability Using Code Smells: An Empirical Study*, „Journal of Empirical Software Engineering”, 11, nr 3, 2006, s. 395-431, <http://mikamantyla.eu/BadCodeSmellsTaxonomy.html> (dostęp dnia 20.04.2017).

480 konieczność wielu zmian w innych miejscach kodu pod wpływem drobnej zmiany, M. Fowler, K. Beck, *Refaktoryzacja. Ulepszanie...*, s. 85.

481 funkcji więcej czasu zajmuje komunikacja z funkcjami z innego modułu niż swojego, *Ibidem*, s. 86.

482 niechęć do tworzenia własnych typów danych, *Ibidem*, s. 87.

483 J.E. Steinhart, *Sekretne życie...*, s. 246.

484 M. Fowler, K. Beck, *Refaktoryzacja. Ulepszanie...*, s. 58.

przydatna w przypadku analizy komizmu w kodzie źródłowym wydaje się freudowska kategoria *uwalniania napięcia*. Nakierowana jest ona bowiem na życie psychiczne jednostki i jej potrzebę „przyjemności uwalnianej przez dowcip”⁴⁸⁵. Pominąć należy w rozważaniach również *teorie wyższości*⁴⁸⁶ zapoczątkowaną przez Platona⁴⁸⁷ i Arystotelesa⁴⁸⁸, ponieważ humor w kodzie źródłowy nie bazuje na poczuciu wyższości obserwatora i wyśmiewaniu ignorancji innych ludzi. Chociaż wspomniana teoria ujmuje problem w kontekście społecznym, to takie podejście nie oddaje dominującej atmosfery powstawania żartów w tekście kodu źródłowego. Oczywiście pojawiają się w środowisku programistów żarty z niewiedzy laików, jednak nie są umieszczane w kodzie. Przykładem niech będzie dowcip: „I told her that in array of my interests she is [1] -she was satisfied”⁴⁸⁹, oparty o obowiązującą w większości języków programowania zasadę numeracji tablic od 0 – a nieoczywistą dla nie-programistów⁴⁹⁰. Wyznanie to stanowi zatem deklarację bycia na drugim miejscu wśród zainteresowań danego programisty.

Wyduje się, że odpowiedniejszą, w kontekście kodu źródłowego będzie kontrastująca z teorią *wyższości*, teoria *niższości* (*theory of inferiority*) autorstwa Roberta Solomona. Punktem wyjścia żartów jest w niej dystans do samego siebie⁴⁹¹ i swojej pracy, ale także życzliwość⁴⁹². Autoironiczna ekspresja jest bardziej typową postawą w zawodzie programisty niż kpina z osób, które i tak nie czytają tekstu programu. Aby komizm mógł się ujawnić, żart musi trafić do właściwego podmiotu, czyli osoby z poczuciem humoru. Skoro zajmujemy się językiem kodu źródłowego jedynie komizm związany z jego tekstem jest obiektem uwagi, nie odnosimy się do

485 D. Grabowski, *Sigmunda Freuda teoria dowcipu, humoru i komiki*, Universitas, Kraków 2019, s. 28.

486 S. C. Bingham, S.E. Green, *Aesthetic as Analysis: Synthesizing Theories of Humor and Disability through Stand-up Comedy*, „Humanity & Society”, 40, nr 3, 2015.

487 „niewiedza słabych budzi w nas śmiech i jest śmieszna z natury”, Platon, *Fileb*, tłum. W. Witwicki, s. 32, <https://wolnelektury.pl/media/book/pdf/platon-fileb.pdf> (dostęp dnia 18.03.2023).

488 „A komedia jest, jak powiedzieliśmy, naśladowaniem charakterów gorszych, jednakże nie pod względem złości zupełnej, lecz o ile śmieszne częścią jest szpetnego; śmieszne bowiem jest ułomnością i szpetotą niesprawiającą cierpienia ani nieszczęścia”, Arystoteles, *Poetyka*, tłum. S. Siedlecki, s. 10,

<https://wolnelektury.pl/media/book/pdf/arystoteles-poetyka.pdf> (dostęp dnia 18.03.2023).

489 <https://www.facebook.com/photo/?fbid=601820068624386&set=a.576439681162425> (dostęp dnia 18.03.2023).

490 Nie dotyczy się to jednak języków, w którym numeracja tabeli zaczyna się od 1 (na przykład w języku *Pascal*, *Lua*, *matlab*, *Visual basic*), czy też nieposortowanej tabeli.

491 K. Afflerbach, *Humorous Developments: Ridicule, Recognition, and the Development of Agency*, praca magisterska, Uniwersytet Północnej Florydy, Jacksonville 2015, s. 25.

<https://digitalcommons.unf.edu/cgi/viewcontent.cgi?article=1637&context=etd> (dostęp dnia 18.03.2023).

492 S. C. Bingham, S.E. Green, *Aesthetic as Analysis...*, s. 15.

żartów powiązanych z tematyką programowania⁴⁹³, czy wykorzystujących dla zabawy formy przypominającej kod (rys. 4). Mimo tego, że powyższe przykłady świadczą o istnieniu wspólnoty dzielącej poczucie humoru, to analiza taka wykracza poza problematykę pracy.

<pre> For the lack of a nail, throw new HorseshoeNailNotFoundException("no nails!"); For the lack of a horseshoe, EquestrianDoctor.getLocalInstance().getHorseDispatcher().shoot(); For the lack of a horse, RidersGuild.getRiderNotificationSubscriberList().getBroadcaster().run(new BroadcastMessage(StableFactory.getNullHorseInstance())); For the lack of a rider, MessageDeliverySubsystem.getLogger().logDeliveryFailure(MessageFactory.getAbstractMessageInstance(new MessageMedium(MessageType.VERBAL), new MessageTransport(MessageTransportType.MOUNTED_RIDER), new MessageSessionDestination(BattleManager.getRoutingInfo(BattleLocation.NEAREST))), MessageFailureReasonCode.UNKNOWN_RIDER_FAILURE); </pre>	“For want of a nail” “For want of a nail the shoe was lost. For want of a shoe the horse was lost. For want of a horse the rider was lost. For want of a rider the battle was lost. For want of a battle the kingdom was lost. And all for the want of a horseshoe nail.”
--	--

Rys. 4. Żartobliwa historyjka z krainy Javaland⁴⁹⁴ zapisana w formie przypominającej kod lecz nim nie będąca, zestawiona ze swoją inspiracją, czyli ludową wyliczanką, znaną jako “For want of a nail”⁴⁹⁵

Mimo że poczucie humoru jest zindywidualizowane, wewnątrz pewnych grup możemy dostrzec typowość w doborze tematyki i środków. Przejawia się to w *humorze zawodowym* jako typie komizmu. W przeciwieństwie do *komizmu zawodowego* nie polega on na wyśmiewaniu przez ludzi z zewnątrz, danej profesji, lecz jego źródłem jest czerpanie ze wspólnych doświadczeń⁴⁹⁶. Nie występuje wtenczas motyw wyższości podmiotu doznającego komizmu, raczej „śmiejemy przez łzy”. Powstałe w ten sposób żarty mają moc więzotwórczą⁴⁹⁷. Tylko w ramach grupy współdzielącej doświadczenia, użycie ironii może śmieszyć, w przeciwnym razie staje się złośliwością.

Charakterystyczną cechą humoru programistycznego stanowi prezentowany w żartach dystans wobec swojej pracy. Autoironie widać wyraźnie w komentarzach zamieszczanych w kodzie. Może przejawiać się, na przykład, w krytyce własnej pracy jako będącej uciążliwą dla osób utrzymujących kod:

493 <https://www.databasestar.com/programmer-jokes/> (dostęp dnia 18.03.2023).

494 <https://steve-yegge.blogspot.com/2006/03/execution-in-kingdom-of-nouns.html#note2> (dostęp dnia 18.03.2023).

495 <https://nationalpoetryday.co.uk/poem/for-want-of-a-nail/> (dostęp dnia 18.03.2023).

496 A. Soćko-Mucha, *Wspólnota śmiechu muzealników (Szkic do dalszych studiów)*, „Zeszyty Naukowe Uniwersytetu Jagiellońskiego Prace Etnograficzne”, 48, nr 3, 2000, s. 254.

497 *Ibidem*.

```
// // Dear maintainer: //
// Once you are done trying to 'optimize' this routine,
// and have realized what a terrible mistake that was,
// please increment the following counter as a warning
// to the next guy: //
// total_hours_wasted_here = 25 //498,
```

W powyższym komentarzu widoczna jest też chęć wspólnej zabawy, polegać miałyby ona na aktualizowaniu ilości wspólnie zmarnowanych godzin, przez kolejnych programistów zajmujących się danym kodem. Również żartobliwie ostrzegający przed ogromem pracy jest kolejny fragment innego autora, w którym odnosi się on do relacji pomiędzy nim a osobami czytającymi:

```
//This code sucks, you know it and I know it.
//Move on and call me an idiot later.499
```

Żart można dostrzec też w komentarzu nawiązującym do motywu rycerskiego. Autor ubarwił go ponadto nawiązaniem do kultowej piosenki:

```
*,**
* For the brave souls who get this far: You are the chosen ones,
* the valiant knights of programming who toil away, without rest,
* fixing our most awful code. To you, true saviors, kings of men,
* I say this: never gonna give you up, never gonna let you down,
* never gonna run around and desert you. Never gonna make you cry,
* never gonna say goodbye. Never gonna tell a lie and hurt you.
*/*,500
```

Wspomniany w komentarzu przebój Rick’iego Astley „Never gonna give you up” był dowcipem internetowym zwanym „Rickroll” polegającym na nieoczekiwanym wyświetlaniu się teledysku. Autor stworzył tym samym dwupoziomowy żart z odniesieniem do rozpoznawalnego żartu.

Spotykane są także komentarze żartobliwie ostrzegające przed siłami nadprzyrodzonymi:

```
// Magic. Do not touch501,
```

498 <https://github.com/Tjoosten/Funny-code-comments/blob/master/Dear%20Maintainer.md> (dostęp dnia 18.03.2023).

499 <https://stackoverflow.com/a/185550> (dostęp dnia 17.03.2023).

500 https://2.bp.blogspot.com/-wnwXJj26U_k/VdXd0QVX4uI/AAAAAAAAADnI/XfW8B3U4ndk/s1600/Funny%2BComment%2BProgramming.png (dostęp dnia 16.03.2023).

501 <https://stackoverflow.com/a/185106> (dostęp dnia 17.03.2023).

czy też innymi sytuacjami niemożliwymi:

```
try {  
} finally { // should never happen  
}502
```

Napotkać można zabawne rymowanki:

```
// Replaces with spaces the braces in cases where braces in places cause stasis  
$str = str_replace(array("\{", "\}"), " ", $str);503
```

Pojawiają się również komentarze wskazujące na przekorę:

```
#define TRUE FALSE  
//Happy debugging suckers504,
```

w niektórych da się dostrzec zabawną nonszalancję np.

```
// Nie uruchamiaj, chyba że masz nieco czasu do zagospodarowania505,
```

lub:

```
Catch (Exception e) {  
    //who cares?  
}506.
```

Komizm trudno jednoznacznie zdefiniować, ponieważ stanowi raczej zestaw własności, które mogą rozbawić. Stosuje się więc szeroki repertuar środków, jak na przykład: wydobywanie absurdów, wyolbrzymiania, pomniejszanie, niespodzianki, niezgodność i różne pogwałcenia norm⁵⁰⁷. W kodzie źródłowym programiści wprowadzają efekt niedorzeczności, na przykład, poprzez nazewnictwo. Znanym żartem słownym jest użycie słowa *ass*, aby nazwać zmienną. W stronę złośliwości zmierza już nazywanie kolejnych zmiennych *a1, a2, ..., an*. Oprócz tego, że nazewnictwo takie nie wskazuje na znaczenie, a więc utrudnia zrozumienie kodu, wymaga również nieprzeciętnej pamięci. Aby wywołać efekt komizmu można użyć też słów kolokwialnych np: zamiast *abort()* wymyślić *eatMyShorts()*, w miejscu *kill()* *Whack()*⁵⁰⁸.

502 <https://stackoverflow.com/a/192823> (dostęp dnia 18.03.2023).

503 <https://stackoverflow.com/a/766363> (dostęp dnia 18.03.2023).

504 <https://stackoverflow.com/a/771974> (dostęp dnia 18.03.2023).

505 R. Martin, *Czysty kod...*, s. 80.

506 <https://stackoverflow.com/a/184682> (dostęp dnia 18.03.2023).

507 B. Dziemidok, *O komizmie*, Książka i Wiedza, Warszawa 1967, s. 59-78.

508 R. C. Marino, *Czysty Kod...*, s. 48.

Wstępne porozumienie sprzyja zrozumiałości żartu. W ramach grupy dzielącej wspólny sens powstał, na przykład, język programowania będące żartem — *Bhailang*, napisany w *TypeScript*'cie pomieszanym z językiem Hindu⁵⁰⁹. Jest zwany również *Brocode*, ponieważ każda linijka zaczyna się od słowa *bhai* czyli bracie. Można też spróbować zalgorytmizować sam koncept żartu i stworzyć, jak to zaproponował @Dj-Rush generator dowcipów w *Javie*⁵¹⁰. Program ten ma za zadanie łączyć kolejne linie tekstu tak, by powstał śmieszny tekst.

Bycie dowcipnym, jakkolwiek dostarcza wspólnej rozrywki, jest jednak przez ekspertów odradzane. Autor „Czystego kodu” Robert C. Martin jako antyprzykład postępowania wymienił między innymi wymyślanie przez programistę nazwy funkcji *HolyHandGranade*, co miało zastąpić zwyczajne *DelateItems*⁵¹¹. Mimo że komizm jest postrzegany jako coś zaciemniającego kod i zupełnie nieprzydatnego, to jednak wynika on z potrzeby ekspresji. Można powiązać posługiwanie się nim z poszukianiem w ten sposób dystansu do przykrego zdarzenia. Poprzez takie zabranie głosu przez programistów ujawnia się w dyskursie na temat kodu potrzeba współrozumienia problematycznej sytuacji i intencja wspólnego przezwyciężenia jej poprzez wyśmianie. Żart stanowi zabranie głosu i pokazaniem stanowiska wobec zaistniałej trudnej sytuacji. Wspólny śmiech z trudów pisania kodu jednoczy.

Badanie językowości kodu źródłowego w oparciu o hermeneutykę dyskursu i filozofię form symbolicznych wymaga uwzględniania wpływu społecznego na język. Podczas tworzenia publicznej przestrzeni dzięki przejawom ekspresji, wychodzi się poza wymiar komunikacji interpersonalnej, gdyż uwaga koncentruje się na kształtowaniu się *przestrzeni pomiędzy*⁵¹² ludźmi. Takie wyrażanie się w języku powoduje bowiem ujawnianie, czyli wyjście poza jedynie komunikowanie. Osoby stają naprzeciw czegoś, co przeobraża się we wspólną sprawę⁵¹³. Komunikacja natomiast skupia się na przekazaniu od-do, a nie na miejscu pomiędzy.

509 <https://analyticsindiamag.com/brocode-an-inside-joke-that-turned-into-a-programming-language/> (dostęp dnia 07.03.2023).

510 <https://www.codeproject.com/Articles/81255/How-to-Create-a-Random-Joke-Generator-in-Java> (dostęp dnia 07.03.2023).

511 R. C. Marino, *Czysty Kod...*, s. 48.

512 Wyrażenie Charlesa Taylora, Ch. Taylor, *Teoria znaczenia ...*, s. 157.

513 *Ibidem*, s. 162.

Habermasowskie działanie niestrategiczne na przykład, czyli nastawione na porozumienie, mimo że pozornie przypomina wspomniane ujęcie, nie oddaje jednak tego, co jest istotą przestrzeni publicznej wytworzonej przez ekspresję językową. W teorii stworzonej przez tego przedstawiciela szkoły krytycznej to dwa podmioty przeprowadzają wspólny plan komunikacyjny, a taka „ślepotą na to co publiczne, jest (...) konsekwencją epistemologicznej tradycji, która uprzywilejowuje rekonstrukcję wiedzy jako własności cechującej krytyczną jednostkę.”⁵¹⁴ I chociaż Habermas nie przedstawia dwóch monologicznych podmiotów, to jednak komunikacja zbudowana jest wokół *Ego* i *Alter*. Umieszczenie w przestrzeni publicznej przez ujawnienie komuś czegoś w przekazie, przesuwa natomiast rozumienie w obszar filozofii języka⁵¹⁵.

Sztuczne języki, przedstawienia matematyczne i kod maszynowy według Taylora w ramach, którego koncepcji powstała struktura niniejszej pracy, miały być ideałem opisu wolnego od ekspresji. Chociaż filozof nazwał posługiwanie się nimi, funkcjonowaniem w „konwersatoryjnej czeluści”⁵¹⁶, wydaje się, że da się znaleźć wiele przykładów zaprzeczających takiej charakterystyce języków programowania. Ekspresyjny wymiar, wprawdzie nie rozbudowany, z całą pewnością, może zaistnieć w tekście kodu źródłowego.

Opisując wyrażanie się w przestrzeni kodu źródłowego, warto też sięgnąć do koncepcji socjologa Fons’a Trompenaars’a i filozofa zarządzania Charles’a Hampden-Turner’a. Stworzyli oni siedmiowymiarowy model kultury, w którym powstałej w koncepcji Halla wysoko- i niskokontekstowości, odpowiadałby sposób, w jaki dopuszczamy innych do siebie. W kulturach wycinkowych, takich jak na przykład Stanów Zjednoczonych, poszczególne sfery życia są izolowane i funkcjonowanie w każdej z nich jest rozpatrywane odrębnie. Obszar przestrzeni publicznej, w którym sposób powierzchowny współdzielili się sprawy z innymi ludźmi, jest natomiast dość

514 *Ibidem*, s. 159.

515 O ile, wspólnym rdzeniem definicji dyskursu w studiach językoznawczych, kulturowych czy antropologicznych jest komunikacja, to metodologicznie stanowią oczywiście różne przedsięwzięcia. Kontekstualne czynniki wypowiedzi czy jej elementy strukturalne interesują szczególnie badaczy pierwszego wspomnianego obszaru. Antropologiczne rozważania koncentrują się zaś na tym jak w strukturze dyskursu odzwierciedla się kultura. Kulturoznawcy zdają się ujmować problem bardziej generalnie, poprzez próbę powiązania dyskursów z różnych obszarów np. porządku komunikacyjnego z wiedzą i władzą czerpiąc w ten sposób z dziedzictwa filozofii Michela Foucaulta.

L. Lindstrom, *Dyskurs*, [w:] *Encyklopedia antropologii społeczno-kulturowej*, red. A. Barnard, J. Spence, Oficyna Wydawnicza Volumen, Warszawa 2008, s. 179.

516 Ch. Taylor, *Teoria znaczenia* ..., s. 165.

znaczny⁵¹⁷. W kulturach całościowych, jak na przykład chińska, zachodzi przenikanie różnych sfer życia⁵¹⁸, a w obszarze małej przestrzeni publicznej „wszystko wiąże się ze wszystkim”⁵¹⁹. W kodzie źródłowym, w związku z komunikacją wprost, ujawnia się niskokontekstowość i typowa dla kultury całościowej ograniczona przestrzeń publiczna⁵²⁰. W obszarze tym nie dochodzi do częstych i urozmaiconych ekspresji. Jednak należy zauważyć, że wymienione sposoby wyrażania się podczas tworzenia wspólnoty językowej w różnych językach, przyjmowania konwencjonalnego stylu, ekspresji w sztuce, odnoszenia się do wartości piękna, brzydoty i komizmu, wskazują na możliwość tego typu działania.

Poprzez syntezę form indywidualnych, jak zaznaczał Cassirer, ukazuje się nam jedność ich procesu twórczego nakierowanego na funkcje tworzenia kultury⁵²¹. Ekspresyjny wymiar wyrażany w kodzie źródłowym stanowi o bogactwie języka jako formie kultury. Sposób, w jaki programiści tworzą przestrzeń publiczną, świadczy o ich kulturalnej działalności. Ich świat to nie tylko świat formalnych wytycznych, ale również duchowy, w którym też chcą działać. Zapośredniczeni w znanej sobie rzeczywistości kulturowej dają jej wyraz w ekspresji językowej, kształtując przestrzeń publiczną. Ta zawodowa grupa korzysta z możliwości wyrażenia się, kreując dyskurs świadczący o pozaużytkowej wartości kodu. Hermeneutyka dyskursu daje możliwość interpretacji tych utrwalonych „ekspresji życia”⁵²², co znalazło wyraz podczas przedstawiania, w jaki wyrażają się wspólnie. Dzięki ekspresji, w tekście tym dochodzi do zerwania z dyskursem realnym rzeczywistości⁵²³, czyli traktowaniem kodu z perspektywy czysto użytkowej na rzecz pojawienia się innego możliwego świata⁵²⁴ – tekstu kulturowego.

517 F. Trompenaars, Ch. Hampden-Turner, *Siedem wymiarów kultury: znaczenie różnic kulturowych w działalności gospodarczej*, Oficyna Ekonomiczna Grupa Wolters Kluwer, Kraków 2002, s. 104.

518 *Ibidem*, s. 110.

519 *Ibidem*, s. 111.

520 ciążenie ku niskokontekstowej komunikacji i całościowości kultury, są rozpoznawane przez tych badaczy jako typowe dla na przykład kultury niemieckiej.

521 E. Cassirer, *Esej o człowieku...*, s. 155-157,

522 P. Ricœur, *Wydarzenie i sens ...*, s. 107,

523 G. Lubowicka, *Refleksyjność człowieka wobec świata znaczeń*, [w:] *Codziennność jako wyzwanie edukacyjne, tom II, Refleksyjność w codzienności edukacyjnej*, red. I. Paszenda, Instytut Pedagogiki Uniwersytetu Wrocławskiego, Wrocław 2017, s. 71.

524 *Ibidem*.

2.2. Wymiar ujawniania

W ramach ustanowienia przestrzeni publicznej w języku, jak zauważył Taylor, prócz najbardziej oczywistej w nim ekspresji, dochodzi również do różnorodnych ujawnień, gdy staje się widoczny jakiś wspólny problem, ważny dla zaangażowanych w niego osób⁵²⁵. W sytuacji tej artykułuje się to, co i w jaki sposób jest znaczące dla programistów. Wszelkie zachowania ujawniające mają na celu odkrycie prawdy⁵²⁶, która w tym wypadku polegałaby na przedstawieniu sposobów bycia ich wspólnoty. Umieszczanie w przestrzeni publicznej przyrównać można do Heideggerowskiego umieszczania w prześwicie (*die Lichtung*)⁵²⁷. „Skrywająca się jawność”⁵²⁸ rozjaśnia się podczas rozważań nad możliwościami funkcjonowania wspólnoty w językowej przestrzeni kodu źródłowego. Ramy tego podrozdziału tworzy badanie tego, co i w jaki sposób staje się znaczące w środowisku programistów, w kontekście filozofii Ricœura i Cassirera.

W języku kodu źródłowego ujawniają się sprawy dla programistów istotne, a wychodzące poza jedynie zagadnienia inżynierii. Świadectwem tego jest dyskusja wokół występującego w nim nazewnictwa, między innymi rasistowskiego *slave-master* („niewolnik-pan”), czy obraźliwego wobec osób niepełnosprawnych *cripple* („kaleka”). Kwestią, która w związku z tym stała się w ostatnim czasie dyskutowana, jest inkluzywne, nieobraźliwe słownictwo. Gdy wprowadzano te obecnie wzbudzające wątpliwość terminy, nie były one postrzegane jako kontrowersyjne. Jednak wraz ze zmianami społeczno-kulturowymi mającymi przełożenie na zmiany w świadomości społecznej, część zbiorowości programistów uznała wspomniane słownictwo za szkodliwe. Ukazuje się tym samym zmiana świadomości tej grupy i to, co staje się dla niej znaczące. Zauważalne jest zrozumienie przez nią zmian jakie nastąpiły w kulturze. W ten sposób, jak widać, język kodu źródłowego ujawnia to, co aktualnie istotne. Odzwierciedla stan aktualnego wartościowania w kulturze.

Aby uchwycić zmianę, jaka nastąpiła w czasie, warto powrócić do źródła pojawienia się w obszarze IT wspomnianego słownictwa, które uznaje się dziś za

525 Ch. Taylor, *Teoria znaczenia* ..., s. 166.

526 K. Michalski, *Heidegger i filozofia współczesna*, PIW, Warszawa 1978, s. 228.

527 Ch. Taylor, *Teoria znaczenia* ..., s. 166.

528 K. Michalski, *Heidegger i filozofia* ..., s. 231.

rasistowskie. Najprawdopodobniej pierwsze użycie metafory *master-slave* w technice pojawiło się w raporcie Davida Gilla z 1904 roku na temat mechanizmu zegara stworzonego dla obserwatorium w Cape Town⁵²⁹. Jeden z zegarów (*slave*) podlegał automatycznej korekcji czasu drugiego, którego wahadło nie musiało napędzać tarczy (*master*). Stąd wyniknął pomysł wynalazcy realizujący się w socjotechnologicznej metaforze podległego niewolnika i bezczynnego pana, który nim kieruje. Wspomniana analogia przyjęła się w inżynierii. W informatyce pojawiła się natomiast najprawdopodobniej po raz pierwszy w 1964 roku, w systemie podziału czasu opracowanym w Dartmouth Collage (*Dartmouth time sharing system*)⁵³⁰. Komputer-niewolnik wykonywał większość pracy polegającej na kalkulacji. Jego wpływ był mniejszy, gdyż niemożliwe, aby „program użytkownika w komputerze podrzędnym (w oryginale *slave computer, przipis aut.*) »uszkodził« program wykonawczy”⁵³¹.

Sądzi się między innymi, jak przytacza etnomatematyk Ron Eglash⁵³², że przyjęcie się analogii pana i niewolnika w informatyce, zostało zaczerpnięte z inżynierii. Zapożyczenie to miało mieć nobilitujący charakter dla pierwszej z dyscyplin. Posługiwanie się wspomnianą metaforą byłoby podchwyceniem trendu odcinania się od kwestii kultury wśród informatyków, którzy mieliby „zamieszkiwać sferę społeczną lub wolną od kultury, co jest powodem do dumy zawodowej”⁵³³. Pojawiła się też koncepcja, rozbieżna wobec wspomnianej, jakoby to wyrażenie miało za zadanie urozmaicić, stereotypowo postrzeganą jako nudną, terminologię stosowaną w środowisku inżynierów⁵³⁴. W ten sposób stawało się wyrazem działania odnoszącego się do ówczesnej rzeczywistości kulturowo-społecznej.

Eglash wskazuje również na inne potencjalne powody występowania w kodzie źródłowym pary *master-slave*. Przywołuje hipotezę Sally Hacker, powstałą już w latach 80-tych XX wieku, że niedostrzeżenie i utrwalanie rasistowskiej terminologii w

529 R. Eglash, *Broken Metaphor. The Master-Slave Analogy in Technical Literature*, „Technology and Culture”, 48, nr 2, 2007, s. 2.

530 *Ibidem*, s. 3.

531 Tłum. własne „impossible for an erroneous or runaway user program in the slave computer to ‘damage’ the executive program”. J. Kemeny, T. Kurtz, *Dartmouth Timesharing*, „Science”, 162, nr 3850, 1968, za R. Eglash, *Broken Metaphor...*, s. 3.

532 Badacz zajmujący się relacją między matematyką a kulturą.

533 Tłum. własne “they inhabit a nonsocial or culture-free realm, which is a matter of professional pride”. R. Eglash, *Broken Metaphor...*, s. 6.

534 *Ibidem*.

obszarze IT byłoby przejawem *white privilege*⁵³⁵, gdyż większość pracowników tej branży stanowiły osoby białe. Innym potencjalnym powodem zaistnienia takiego słownictwa mogłaby być również potrzeba dominacji wyrażona w sadomasochistycznych tendencjach⁵³⁶. Kolejną propozycją tłumaczącą zjawisko, byłoby uchwycenie w nim napięcia pomiędzy jednoczesną potrzebą ludzkiej dominacji i dążeniem do stworzenia niezależnych automatów. Stąd miałyby wynikać łatwość w przyjęciu tej metafory. Jakkolwiek są to ciekawe propozycje, w kwestii ujawniania w języku, jest jednak istotne nie źródło powstania problemu rasistowskiego słownictwa, gdyż badanie tego należy do zakresu problematyki socjologicznej czy etnologicznej, lecz samo to jak zagadnienie zaczęło być odbierane.

W ramach filozoficznych rozważań nad językiem kodu źródłowego warto zastanowić się, w jaki sposób problem stał się znaczący i co implikuje. Początek zmiany podejścia nastąpił wraz ze skargą z 2003 roku pracownika rządowego wystosowaną w związku z używaniem w miejscu pracy terminów „pan” i „niewolnik”, zamieszczonych w opisie sprzętu elektronicznego. Strona rządowa, rada miejska Los Angeles, przychyliła się do stanowiska pozywającego i poinformowała kontrahentów o wymaganiach zmiany opisu dostarczanego sprzętu⁵³⁷. Na skutek decyzji rozgorzała dyskusja między zwolennikami uszanowania kulturowej różnorodności i wrażliwości oraz przeciwnikami poprawności politycznej.

Efektem dostrzeżenia znaczenia wspomnianej dyskusji były decyzje niektórych organizacji. Zaprzestano używania w obrębie kodu źródłowego *Linuksa* takich nazw jak *master*, *slave*, *blacklist*. W miejsce słów „pan” i „niewolnik” wprowadzono w języku *Drupal* – *primary/replica*, w *Django* – *leader/follower*⁵³⁸, a w *Pythonie* – *parent processes/helpers*, *workers*⁵³⁹. Problem rasistowskiej terminologii miał stać się znaczący, jak sami protestujący podkreślali, z powodu coraz bardziej różnorodnej kulturowo społeczności programistów. Jednak wyjaśnienie to wydaje się nie trafiać w sedno tej zmiany. Po pierwsze dlatego, że to nie ze społeczności osób

535 R. Eglash, *Broken Metaphor...*, s. 5.

536 *Ibidem*, s. 6.

537 <http://edition.cnn.com/2003/TECH/ptech/11/26/master.term.reut/index.html> (dostęp dnia: 11.04.2023).

538 <https://www.vice.com/en/article/8x7akv/masterslave-terminology-was-removed-from-python-programming-language> (dostęp dnia: 11.04.2023).

539 <https://content.techgig.com/technology/python-dumps-offensive-master/slave-terms-from-its-documentation/articleshow/65964126.cms> (dostęp dnia: 11.04.2023).

czarnych⁵⁴⁰/czarnoskórych (*black/ black community*), których przodkowie przeżyli niewolnictwo, dobiegały pierwsze głosy sprzeciwu. Po drugie zaś, jeśli by nawet środowisko pracy nie było różnorodne kulturowo i rasowo, nadal nie zmieniałoby to niestosowności nazewnictwa rozpoznanego jako rasistowskie. Clou w tej sprawie stanowi bowiem zmiana w świadomości społecznej jako jedna z tzw. pęteli przyczynowych⁵⁴¹ tworzących historię. Według socjologa Piotra Sztompki, na proces mechanizmu dziejowego mają wpływ efekty strukturalne, zdolności podmiotów, zhumanizowana przyroda i właśnie wspomniana zmodyfikowana świadomość⁵⁴².

Świadomość może być przypisana również podmiotom zbiorowym jako świadomość społeczna⁵⁴³. Według Sztompki stanowi ona środowisko funkcjonowania ludzi i tak jak środowisko przyrodnicze i ona ulega zmianie. W procesie dziejowym następuje jej przemodelowanie, co odzwierciedla się w „ideach, wierzeniach, przekonaniach podzielanych przez podmioty (jednostki i grupy), jak i w ideologiach, systemach przekonań, doktrynach istniejących w bardziej zobiektywizowany, ponadjednostkowy sposób”⁵⁴⁴. Ta właśnie zmieniona świadomość redefiniowałaby zdolności podmiotów, wskazując im, że możliwy jest protest przeciwko obraźliwej terminologii. Oddziaływałaby również ona strukturalnie określając, jaką przyjmie on formę. Świadomość społeczna stanowi bowiem warunkującą siłę dla praktyki społecznej⁵⁴⁵ będąc „polem zasobów (pojęć, symboli, norm, ram, etc.)”⁵⁴⁶ do czynienia interpretacji. Pod wpływem jej zmiany, zaczęto odchodzić od użycia terminów „pan” i „niewolnik” w informatyce.

540 Grupa rasowa w języku angielskim nazywana *black*. Ponieważ w Polsce na osoby białe nie używa się określenia „białoskórzy” lecz „biali”, więc możliwe jest użycie przymiotnika „czarni” w stosunku do określenia grupy osób czarnoskórych. Fragment stanowiska Rady Języka Polskiego przygotowany przez dr hab. Marka Łazińskiego, prof UW, dotyczący pojawienia się w użyciu obecnie słowa „czarni”:

„Nasi sąsiedzi o czarnej skórze wolą być nazywani ze względu na konkretną narodowość (np. Senegalczyk, Nigeryjczyk), wolą być Afrykanami, czarnoskórymi, ciemnoskórymi lub po prostu czarnymi (choć jeszcze kilkadziesiąt lat temu konotacje przymiotnika czarny w odniesieniu do człowieka czarnoskórego były gorsze niż rzeczownika Murzyn)”, <https://rjp.pan.pl/porady-jezykowe-main/1892-murzyn-i-murzynka> (dostęp dnia: 20.05.2023).

541 P. Sztompka, *Socjologia zmian społecznych*, Znak, Kraków 2010, s. 216.

542 *Ibidem*.

543 *Ibidem*, s. 210.

544 *Ibidem*, s. 217.

545 *Ibidem*, s. 216-217.

546 *Ibidem*, s. 211.

Anthony Giddens wyróżnił świadomość dyskursywną polegającą na „monitorowaniu monitorowania”⁵⁴⁷ zachowań zarówno własnych, jak i cudzych. Oznaczać ma to, że ludzie są w stanie nie tylko rozważać dyskursywnie swoje motywy działań, ale także wznieść się na *meta-poziom* i reflektować te rozważania. Wspomniany już Sztompka zauważył, że dotyczy to nie tylko jednostek, ale również podmiotów zbiorowych⁵⁴⁸. Aktorzy społeczni odtwarzają właściwości struktury korzystając z języka mającego przecież publiczny charakter⁵⁴⁹. Kod źródłowy jako wytwór kulturowy podlegałby zatem zmianom na skutek wpływu świadomości zbiorowej, co uwidoczniło się w reakcji na wykluczający język w IT.

Kolejnym przykładem zmian słownictwa w językach programowania dokonujących się pod wpływem kulturowych fluktuacji jest odrzucenie negatywnie nacechowanych słów dotyczących zdrowia psychicznego, na przykład *sanity-check*, *sanity-test*⁵⁵⁰, a co za tym idzie *insanity*⁵⁵¹ (rys. 5). Etykietowanie zaburzeń psychicznych było charakterystyczne, jak zauważył Michel Foucault, dla alienującego człowieka języka psychologii⁵⁵². Według francuskiego filozofa wyodrębnienie ludzi „obłąkanych” z populacji zdrowych, miało ułatwiać utrzymanie porządku w postaci swoistej kontroli. W miejscu średniowiecznego trądu i z podobną mocą moralnego wykluczania, pojawiło się szaleństwo⁵⁵³. Traktowanie w przeszłości zaburzeń psychicznych jako zjawisk niezrozumiałych i niebezpiecznych, wpłynęło na ich negatywne etykietowanie. Rzutowało to na miejsce tej stygmatyzowanej grupy w społeczeństwie. Sposób, w jaki ludzie mówią o chorobie-zdrowiu psychicznym, wpływa na postrzeganie i rozumienie tych stanów, a także na to jak traktowane są osoby zaburzone psychicznie.

547 A. Giddens, *Stanowienie społeczeństwa*, Zysk, Poznań 2003, s. 69.

548 P. Sztompka, *Socjologia zmian...*, s. 210.

549 A. Giddens, *Stanowienie społeczeństwa...*, s. 68.

550 <https://inclusivenaming.org/word-lists/tier-2/sanity-check/> (dostęp dnia: 04.04.2023).

551 <https://github.com/phohenecker/in-sanity> (dostęp dnia: 13.05.2023).

552 M. Foucault, *Historia szaleństwa w dobie klasycyzmu*, PIW, Warszawa 1987, s. 475.

553 *Ibidem*, s. 21.

```
def eq_mod_5(x: int, y: int) -> bool:
    """Computes whether two integers are equal modulo 5."""
    return (x - y) % 5 == 0

# an arg that needs to be divisible by 5
insanity.sanitize_value(
    "some_arg",
    some_arg,
    0,
    equality_fn=eq_mod_5,
    error_msg="The arg is not divisible by 5!"
)
```

Rys. 5. Przykład użycia funkcji *insanity* (niepoczytalność)⁵⁵⁴

Obiecująco, coraz bardziej popularne, staje się podejście dymensjonalne. Opiera się ono na przedstawianiu natężenia cech w obszarze pomiędzy zdrowiem i chorobą⁵⁵⁵. Jak pokazują badania nad stygmatyzacją osób ze schizofrenią, już samo etykietowanie danego zaburzenia jako „choroby psychicznej”, wiąże się z oceną osób z diagnozą jako niebezpiecznych i zależnych, oraz budzących większy lęk, złość i litość⁵⁵⁶. Dlatego tak ważne jest powszechne używanie języka wolnego od stygmatyzacji, również w języku kodu źródłowego.

Kierując się rozumieniem współczesnego kontekstu kulturowego i zmieniającą się wrażliwością społeczną, pojawiły się też propozycje, żeby usunąć z kodu źródłowego nazwy *cripple* (rys. 6), *crippled*, *crippleware*⁵⁵⁷. W języku angielskim słowo to ma swój źródłosłów w XIII wiecznym *cripple*⁵⁵⁸ i oznaczało osobę utykającą, mającą problem z przemieszczaniem się. Jest uznawane za obraźliwe, podobnie jak jego polski odpowiednik – „kaleka”. Może ono posłużyć jako przykład ukazujący nam niestosowność użycia takiej terminologii w kodzie źródłowym. Polski odpowiednik tego słowa pochodzi od tureckiego *kalik*, co oznacza „niedostateczny”, „brakujący”, gdy zdrowie przedstawiane jest jako stan pełny, choroba zaś jest rozumiana jako

554 <https://github.com/phohenecker/in-sanity> (dostęp dnia: 13.05.2023).

555 T. Rowiński, M. Kowalska-Dąbrowska, W. Strus, *Pomiar patologicznych cech osobowości według DSM-5: polska adaptacja Inwentarza Osobowości PID-5. Część I – podstawy teoretyczne*, „Psychiatria Polska”, 53, nr 1, 2019, s. 8.

556 M. C. Angermeyer, H. Matschinger, *The stigma of mental illness: effects of labelling on public attitudes towards people with mental disorder*, „Acta Psychiatrica Scandinavica”, 108, 2003, s. 307.

557 https://inclusivenaming.org/word-lists/tier-1/_cripple/ (dostęp dnia: 04.04.2023).

558 <https://www.etymonline.com/word/cripple> (dostęp dnia: 19.04.2023).

brak⁵⁵⁹. Uznanie tego przeciwieństwa zakorzenione jest w biomedycznym modelu zdrowia i choroby. Jakkolwiek jednak można ująć zdrowie inaczej, nie jako stan, czy stały zasób, ale tak jak Aaron Antonovsky, postrzegać je jako mieszczące się w pewnym zakresie⁵⁶⁰.

```
#ifndef _CRIPPLE_LOG_H
#define _CRIPPLE_LOG_H

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <stdarg.h>
#include <string.h>
#include <errno.h>

void cri_log(const char *fmt, ...);

#endif
```

Rys. 6. Przykład pojawienia się w kodzie źródłowym słowa *cripple*⁵⁶¹

Obecnie proponuje się zamiast stosowania terminów negatywnych typu: „osoba ułomna”, „kaleka”, „inwalida”, „kuternoga”, używanie niebudzącego kontrowersji moralnych określenia „osoba z niepełnosprawnością”⁵⁶². Wynika to z faktu, że wymienione określenia można uznać za upokarzające, umniejszające godność ludzką⁵⁶³. Redukują one człowieka do przymiotnika, za pomocą którego podkreśla się brak jakiejś właściwości ludzkiej. Warto zatem zaznaczyć, że ów ableizm w języku kodu źródłowego mógł zostać dostrzeżony i zwalczany dopiero wraz ze wzrostem świadomości społecznej.

Zmiany kulturowe wpłynęły również na próby odejścia od użycia w świecie IT terminologii związanej ze stereotypami płciowymi i kwestiami powiązanych z płcią. Na język stosowany w inżynierii w tym programowaniu miała bowiem, jak zauważyła

559 J. Szadura, *O ludowej etiologii wrodzonego kalectwa*, „Literatura ludowa”, nr 6, 2018, s. 70.

560 I. Heszen, H. Sęk, *Psychologia zdrowia*, Państwowe Wydawnictwo Naukowe, Warszawa 2007, s. 49.

561 https://github.com/hmgle/cripple/blob/master/cripple_log.h (dostęp dnia: 13.05.2023).

562 M. Garbat, *Historia niepełnosprawności...*, s. 99.

563 M. Bełza, D. Prysak, „Upokorzona” codzienność osób niepełnosprawnych, „Parecja”, nr 2, 2014, s. 27.

socjolożka Sally Hacker, wpływ armia USA⁵⁶⁴, gdyż w ramach finansowanych przez nią projektów rządowych rozwijała się w tym kraju informatyka. Pojawiła się obecnie sugestia, że z perspektywy równości płciowej za niestosowne uznać można użycie zwrotu *man-in-the-middle*⁵⁶⁵. Takie określenie osoby atakującej system informatyczny miałoby implikować brak zdolności hackerskich u kobiet. Znacząca jest również propozycja odejścia, w samym już kodzie źródłowym, od użycia terminu *abort*. Jest ona związana z chęcią uniknięcia niepotrzebnego „odwołania się do emocjonalnie naładowanego kontekstu kulturowego”⁵⁶⁶ z nim związanego.

Wprowadzenie inkluzywnego języka można uznać za inicjatywę oddolną. Trwające w internetowych grupach dyskusje wokół tematu nazewnictwa, wpłynęły na zmianę standardów⁵⁶⁷. Przez dziesięciolecia nie zwracano uwagi na szkodliwe słownictwo, gdyż nie było ono wówczas kontrowersyjne. Zmiany terminologii w kodzie źródłowym w związku ze zmianą świadomości społecznej programistów, ujawnia jak pewne sprawy stają się istotne⁵⁶⁸. Dyskusja nad problemem dyskryminującego słownictwa była konieczna w tym środowisku, gdyż „na długo przed zrozumieniem samych siebie w refleksji rozumiemy siebie z pełną oczywistością w (...) społeczeństwie, w których żyjemy”⁵⁶⁹. Warto zaznaczyć, że hermeneutyczny *horyzont współczesności* kształtuje się poprzez odniesienie do przeszłości, gdyż tylko w ten sposób może pojawić się rozumienie. Stąd zamiast naiwnego niwelowania napięcia, poprzez pozostawanie w tradycji, konieczne jest jednak podjęcie się zadania hermeneutycznego⁵⁷⁰. Warto zauważyć, że zmiana podejścia do kwestii wykluczającego języka była możliwa dzięki dostrzeżeniu i podsycaniu napięcia między tym, jak kiedyś, a jak obecnie odbierane są pewne słowa.

Usytuowanie w kulturze nie stanowi ograniczenia dla możliwości rozumienia, lecz jest warunkiem tegoż. Ricœur podzielał myśli Gadamera w kwestii horyzontu rozumienia tyle, że przeciwnie do niemieckiego filozofa, doceniał hermeneutyczną funkcję dystansu. O ile Gadamer w bezpośredniości języka dostrzegał możliwości

564 S. Hacker, *Pleasure, Power, and Technology: Some Tales of Gender, Engineering, and the Cooperative Workplace*, Unwin Hyman, Boston 1989, s. 131.

565 <https://inclusivenaming.org/word-lists/tier-3/man-in-middle/> (dostęp dnia: 04.04.2023).

566 <https://inclusivenaming.org/word-lists/tier-1/abort/> ((dostęp dnia: 25.04.2023).

567 https://lists.openembedded.org/g/openembedded-architecture/topic/inclusive_language_summary/75821819 (dostęp dnia: 04.04.2023).

568 Ch. Taylor, *Teoria znaczenia...*, s. 166.

569 H.-G. Gadamer, *Prawda i metoda...*, s. 381.

570 *Ibidem*, s. 420.

doświadczenia „ewidencji prawdy”⁵⁷¹, jako że wtedy następuje płynna „wymiana między »Ja« i »Ty«”⁵⁷², to Ricœur uznawał za konieczny dystans. Według francuskiego filozofa nie niszczy on więzi z przedmiotem badań, a umożliwia zobiektywizowanie. Zdystansowanie od zdarzenia możliwe jest w pełni dopiero w piśmie, gdyż tekst stanowi „modelowy przypadek komunikacji na dystans”⁵⁷³. Aby przyswoić, czyli zaadaptować do współczesności język kodu źródłowego, konieczne jest zatem nie „duchowe współbrzmienie” z twórcami obecnie uznanej za wykluczającą terminologii, lecz zdystansowanie⁵⁷⁴. Dzięki niemu może dojść do obiektywizacji i pojęcia sensu, jaki obecnie mają terminy, takie chociażby jak „pan” i „niewolnik”, czy „kaleka”.

Ujawnia się zatem w kodzie źródłowym zmiana podejścia społecznego do kwestii rasy, niepełnosprawności, zdrowia psychicznego, czy płci. W ramach językowej przestrzeni publicznej uwidaczniają się problemy, które ukazują sposób bycia wspólnoty. Protestowanie przeciw umieszczaniu w niej potencjalnie krzywdzącej terminologii wskazuje na kwestie, które są dla środowiska programistów znaczące. Skoro jak pisał Ricœur „rozumiemy siebie jedynie odbywając dookólną drogę pośród znaków ludzkości utrwalonych w dziełach kultury”⁵⁷⁵, nie można zatem zignorować kwestii wspomnianych zmian języka kodu źródłowego. Artykułuje się w ten sposób to, co istotne dla ludzi, co ich przejmuje. Modernizacja nieprzystającego do współczesności języka, którą zaczęli wprowadzać programiści, odzwierciedla również transformację ich podejścia do programowania i technologii. Ujawnia się w ten sposób, że są oni świadomi wpływu kultury i niemożliwości wyizolowania od niej swojej pracy. Wykraczają tym samym w swoich zainteresowaniach poza jedynie zagadnienia techniczne.

Równie istotną kwestią, ujawniającą się w językowej przestrzeni publicznej kodu źródłowego, co zmiana świadomości społecznej, jest kwestia stałości pamięci. Wyraża się ona w upamiętniającym znane osoby nazewnictwie. Historia tworzenia języków programowania uwidacznia się w używanym słownictwie i jest przywoływana w podręcznikach do programowania, czy też w treściach blogów tematycznych. Pamięć

571 *Ibidem*, s. 649.

572 H.-G. Gadamer, *Człowiek i język*, [w:] *Rozum, słowo, dzieje*, red. M. Łukasiewicz, PIW, Warszawa 2000, s. 59.

573 *Ibidem*, s. 225.

574 P. Ricœur, *Hermeneutyczna funkcja ...*, s. 243.

575 *Ibidem*.

o tym, skąd pochodzi język, jest dla niektórych istotna i podtrzymywana. W ramach wspólnoty programistycznej istnieje pamięć, na przykład, o wybitnych osobach, które przysłużyły się informatyce. Uczczeniem we wspólnej pamięci było nadanie opartemu na *Pascalu* językowi programowania nazwy *Ada*, jako uświetnienie matematyczki Augusty Ady Lovelace⁵⁷⁶, zwanej pierwszą programistką.

W specjalistycznym nazewnictwie upamiętnieni zostali również twórca języka programowania *Fortran* John Backus i informatyk Peter Naur, w „notacji Backusa-Naura” (BNF)⁵⁷⁷. Zapis ten bywa łączony z pracami z zakresu indyjskiej logiki autorstwa Paniniego (choć tego typu hipotezy można uznać za wyolbrzymienie)⁵⁷⁸. Notacja BNF stanowi formalny sposób specyfikacji języków programowania. Notacja równań natomiast, która jest wykorzystywana w tekście kodu źródłowego, posiada również swoje nazewnictwo. Wśród zapisów warto wyróżnić „notację polską”, nazwaną tak na cześć jej pomysłodawcy Polaka, Jana Łukasiewicza⁵⁷⁹. Jako, że jest to notacja prefiksowa, znak operatora zapisuje się w niej przed operandami, na przykład $x+12+34$. Notacja postfiksowa natomiast polega na zapisie operatora za operandami, na przykład $12+34+x$, i mimo że zaproponował ją Amerykanin Arthur Burks, nosi nazwę „odwróconej notacji polskiej” (*reverse polish notation*, RPN)⁵⁸⁰. W przeciwieństwie do zapisu ifiksowego, na przykład $(1+2)x(3+4)$, wspomniane wcześniej notacje nie wymagają użycia nawiasów. Z perspektywy tej pracy ważna jest jednak nie sama użyteczność notacji polskiej i polskiej odwróconej, ale pojawiające się w nazwie nawiązanie do pochodzenia twórcy jednej z nich. Ujawnia się w tymże obecność przeszłości. Można dzięki temu dostrzec pewną ciągłość kulturową. Potwierdza to konstatacja Cassirera, który zauważył, że historia „dodaje do tej faktycznej rekonstrukcji rekonstrukcję symboliczną”⁵⁸¹. Aby ożywić przekaz należy ją odczytać, a następnie dokonać interpretacji.

Wspomniany wątek narodowy jest tak naprawdę dowodem na ciągłość pamięci historycznej. Ukazuje, że programiści chcą pamiętać o istotnych dla rozwoju

576 J. Fuegi, J. Francis, *Lovelace & Babbage and the Creation of the 1843 'Notes'*, „EEE Annals of the History of Computing”, 25, nr 4, 2003, s. 24.

577 J. E. Steinhart, *Sekretne życie ...*, s. 235.

578 G. Penn, P. Kiparsky, *On Panini and the Generative Capacity of Contextualized Replacement Systems*, „Conference: Proceedings of COLING”, 2012, s. 3, <https://web.stanford.edu/~kiparsky/Papers/panini-1.pdf> (dostęp dnia: 01.05.2023).

579 J. E. Steinhart, *Sekretne życie...*, s. 131.

580 *Ibidem*.

581 E. Cassirer, *Esej o człowieku...*, s. 329.

informatyki osobach. Wspomnienie o nich potrafią ponadto umiejscowić nawet w ramach kontekstu kulturowego danej osoby, na przykład w kontekście jej narodowości. Podobnie rzecz ma się z nazwą sposobu zapisu nazw zmiennych oraz obiektów zwanym „notacją węgierską”, której autor Charles Simonyi był właśnie wymienionej narodowości⁵⁸².

Pamięć o osobach często daje o sobie znać w podręcznikach dotyczących języków programowania. Joannas Steinhart wspomina chociażby autorów niektórych języków, na przykład: John McCarthy zaprojektował *LISP*⁵⁸³, a Ken Thompson język *B*⁵⁸⁴. Przypomniane zostaje nazwisko Harolda Lawsons jako wynalazcy *pointer*⁵⁸⁵. Bardzo znany ostatnimi czasy stał się Greg Walsh, w związku z zakrojonymi na szeroką skalę w środowisku informatyków poszukiwaniami autora kultowej funkcji *Fast Inverse Square Root*⁵⁸⁶, wykorzystanej w kodzie gry komputerowej „Quake III Arena”. Autorstwo Walsh’a udało się ustalić dopiero po 7 latach, od kiedy kod źródłowy został udostępniony. Te długie poszukiwania i domysły świadczą o potrzebie podtrzymywania pamięci. Jak zauważył Cassirer:

„Wszelka duchowa treść jest dla nas w sposób konieczny powiązana z formą świadomości, a tym samym z formą czasu. Treść ta istnieje tylko o tyle, o ile powstaje ona w czasie i wydaje się, iż nie może zaistnieć inaczej”⁵⁸⁷.

Dbłość o ciągłość wynika z kulturowego podejścia przez osoby dbających o dziedzictwo nauk informatycznych. W przestrzeni publicznej języka ceniona jest wspólna pamięć o istotnych dla tej wspólnoty pomysłach.

Pamięć o ważnych osobach ujawnia się też w nazewnictwie dotyczącym układu kodu, czyli wizualnego obrazu tekstu. W języku *C* i *C++* używa się najczęściej nawiasów klamrowych w dwóch stylach: *K&R (kernel style)* i *Allman (BSD style)*. Oba noszą swoją nazwę od nazwisk ich twórców; pierwszy stanowi zapis inicjałów

582 C. Simonyi, *Hungarian notation*. *MSDN Library*, 1999, <https://cld.pt/dl/download/4275816b-59bc-4fe9-96a3-f2c7a24e9246/GnuCOBOL/MISC/Manuals%20%26%20Guides/C-C%2B%2B%20Miscellaneous/Hungarian%20Notation%20%28by%20Charles%20Simonyi%29%20%281999%29.pdf> (dostęp dnia: 01.05.2023).

583 J. E. Steinhart, *Sekretne życie...*s. 434.

584 *Ibidem*, s. 232.

585 *Ibidem*, s. 193.

586 <https://medium.com/hard-mode/the-legendary-fast-inverse-square-root-e51fee3b49d9> (dostęp dnia: 05.05.2023).

587 E. Cassirer, *Pojęcie formy symbolicznej w budowie nauk humanistycznych*, [w:] *Symbol i Język...*, s. 17.

programistów Briana Kernighana i Dennisa Ritchiego, drugi zaś powstał od nazwiska Erica Allmana⁵⁸⁸. Istnieje też wariacja stylu Allmana, nazwana również od jej twórcy stylem *Whitesmithsa*⁵⁸⁹. Co ciekawe nawiasy klamrowe w stylu *K&R* nazywane są też nawiasami egipskimi (*Egyptian brackets*) z uwagi na to, że przypominają mają hieroglif przedstawiający postać z podniesioną ręką na przodzie i opuszczoną z tyłu (Rys. 7)⁵⁹⁰. Takowe odniesienie do historii czy to przez wspomnianie konkretnych znaczących postaci, czy szerzej do historii nawet w tym wypadku starożytnej, pokazuje jak ważne jest w społeczności programistów osadzenie języka w ramach kontekstu kulturowego.



Rys. 7. Nawiasy egipskie porównane do hieroglify⁵⁹¹

Ciekawe wydaje się również wspomnienie w kodzie źródłowym osoby Stephena Bourne’a. W kodzie źródłowym programu napisanego w *Bashu* pojawia się wskazanie lokalizacji: `#!/bin/bash`. Wspomniany „bash” odnosi się do powłoki, jej nazwa pochodzi od pierwszych liter „Bourne-again shell”. Powłoka ta⁵⁹² została bowiem powtórnie napisana przez matematyka Stephena Bourne⁵⁹³ – postać bardzo rozpoznawaną. Jak pisze Jonathan Steinhart: „okazało się, że tożsamość Bourne’a dodaje tyle wartości, że *bash* szybko zdobył wiodącą pozycję wśród powłok”⁵⁹⁴. Odwołanie to stanowi kolejny ślad, poprzez który w językowej przestrzeni publicznej kodu źródłowego ujawnia się istotność podtrzymywania historii. Jeśli więc o zmianę w języku chodzi, taką jak podczas zmiany świadomości społecznej, ale i o ciągłość, gdy dba się o upamiętnienie znaczących postaci i podtrzymywanie o nich pamięci, to w obu tych przypadkach uwidaczniają się, to co znaczące dla programistów. Stosunek do przeszłości ujawnia się

588 M. Heroux, P. Sexton, *Epetra developers coding guidelines*. No. SAND2003-4169. Sandia National Laboratories (SNL), Albuquerque 2003, s. 13, informacja z przypisu dolnego, <https://www.osti.gov/servlets/purl/918292>, (dostęp dnia: 05.05.2023).

589 E. Dijkstra, 3 *Writing programs to process images*. *Machine Vision*, 2004, s. 33.

590 <https://foldoc.org/Egyptian+brackets> (dostęp dnia: 05.05.2023).

591 <https://blog.codinghorror.com/new-programming-jargon/> (dostęp dnia: 05.05.2023).

592 Powłoka bash jest powłoką systemową, tekstową. Jest to program pośredniczący.

593 Alternatywna wersja mówi o tym, że Bourne brzmi jak born, co może być rozumiane jako born-again shell czyli „odrodzona powłoka”. Przypis dolny J. E. Steinhart, *Sekretne życie ...*, s. 468.

594 *Ibidem*.

jako jej kontestowanie lub kontynuacja – dzięki temu możemy dostrzec to, co było ukryte.

Warto również zauważyć jeszcze jedną okoliczność, podczas której to dochodzi do ujawnień w kodzie źródłowym, tym razem w języku naturalnym - mowa tu o komentarzach. Pełnią one funkcję wyjaśniającą i pomagają programistom czytać i rozumieć kod. Obecna jest tam ciągła komunikacja, tyle że rozciągnięta w czasie, gdyż kolejne wpisy są dodawane nie w czasie rzeczywistym a asynchronicznym. Przez wspólne ujawnianie w komentarzach, w języku powstaje przestrzeń publiczna, w której rozważa się konkretne problemy zawierające się w tekście kodu – napisanym w językach sztucznych.

Ujawnienia w językowej przestrzeni publicznej wydobywają na jaw ukryte treści. Okazuje się, że w społeczności programistów mogą być znaczące - uwidaczniające się już na poziomie języka kodu źródłowego - zjawiska osadzone w kulturze. Wykraczają one poza tematykę inżynierii. Na jaw wychodzi zatem to, co jest znaczące jak i sposób, w jaki znaczy. Dzięki interpretacji może dojść do odkrycia prawdy o tym, na jakie sposoby istnieje ta wspólnota, co jest dla niej ważne i jak to komunikuje. Dotychczasową wiedzę o programistycznej działalności uzupełnić można bowiem o funkcjonowanie w wymiarze kulturowym.

Punktem wyjścia tych rozważań był język, natomiast droga rozumienia wiodła przez analizę. Sugerując się propozycją Ricœura, zastosowano więc „okrężną drogę” hermeneutycznego rozumowania. Zgodnie z wytycznymi filozofa interpretacja prowadziła od płaszczyzny semantycznej, gdzie rozpoznawane były w języku kwestie problemowe dotyczące „wykluczającego” języka oraz zjawiska uczczenia pamięci o kimś lub o czymś. Dalej ścieżka interpretacyjna wiodła przez refleksję nad zmianą świadomości i zachowaniem pamięci. Na płaszczyźnie refleksyjnej nastąpiło zatem przyswojenie sensu w „podwójnie pośredniej refleksji”⁵⁹⁵ polegające na dotarciu do rozumienia przez krytykę świadomości fałszywej.

Finalnie w powyższych rozważaniach problematyka refleksji „przekroczyła siebie w problematyce egzystencji”⁵⁹⁶. W wyszczególnionym przez Ricœura „etapie egzystencyjnym”⁵⁹⁷ przybliżamy się do rozumienia samego siebie w danym kontekście.

595 P. Ricœur, *Egzystencja i hermeneutyka...*, s. 197.

596 *Ibidem*, s. 200.

597 *Ibidem*, s. 190-203.

W przestrzeni publicznej języków sztucznych dają bowiem o sobie znać zjawiska potwierdzające istnienie kulturowej wspólnoty programistów. Ci, bywa że, rozumieją siebie jako grupę, która ma swoje korzenie, osadzona jest w pewnym kontekście kulturowym i podlega zmianom.

Kontekst filozofii Cassirera pomaga natomiast dostrzec w języku kodu źródłowego językową formę symboliczną. Tekst ten prezentuje się jako dzieło ducha, w którym ujawnia się moc „budowania własnego świata, świata »idealnego«”⁵⁹⁸. Filozof postrzega bowiem kulturę jako proces samowyzwalania człowieka, w którym język stanowi jedną z faz tego procesu. W językowej formie symbolicznej zawiera się według Cassirera różnorodność przedstawień. Ujawnia się w niej „specyficzna składnia”, jeśliby rozpatrzyć tak skrajne przypadki, jak chociażby tekst poetycki i tekst kodu źródłowego. Jak postulował wspomniany filozof: „istnieją tego rodzaju »składnie« dla każdej symbolicznej formy” i da się je „sprowadzić do pewnych »źródłowych postaci«, podstawowych form religijnego, rzeźbiarskiego czy malarskiego przedstawienia”⁵⁹⁹. Kod źródłowy jako tekst, byłby zatem przedstawieniem językowej formy symbolicznej o bardzo odmiennej składni. Jednakże możliwe byłoby rozpoznanie w nim tej wspólnej Cassirerowskiej „źródłowej postaci”.

Analiza problemu została dokonana Ricœurowską „okrężną drogą” hermeneutycznego rozumienia, ale i z wykorzystaniem jednej z używanych przez Cassirera, kantowskich grup kategorii – mowa o „jakościowej” formie – czasie⁶⁰⁰. W zaprezentowanych przykładach bowiem można się było doszukać powiązania za pomocą relacji czasowej: zmienności kulturowej w ewolucji znaczenia słów, stałości w ciągłości kulturowej. Porządek wyznaczała zatem kategoria czasu w postaci zmiany i ciągłości wspólnej pamięci.

598 E. Cassirer, *Esej o człowieku...*, s. 411.

599 E. Cassirer, *Zur Objectivität der Ausdrucksfunktion*, [w:] *Nachgelassene Manuskripte und Texte*, K. Kohnke, J.M. Krois, t. 1-11 przetłumaczone i zamieszczone w P. Parszutowicz, *Fenomenologia form symbolicznych. Podstawowe pojęcia i inspiracje „późnej” filozofii Ernsta Cassirera*, Wydawnictwo IFiS PAN, Kęty 2011, s. 112.

600 Chociaż z czasem Cassirer zaczął łączyć czas i przestrzeń w czasoprzestrzeń, *Ibidem*, s. 113.

2.3. Wymiar konstytuowania

Poza wymiarem ujawniania i ekspresji w języku, możemy również wyodrębnić jego rolę konstytutywną. Według Taylora uchwycenie istoty rzeczywistości w opisie sprawia, że zaczynamy ją współtworzyć, gdyż „opisy samych siebie, (...) nie są czymś zewnętrznym wobec opisanej rzeczywistości, czymś, co pozostawia ją w niezmienionej postaci; konstytuują tę rzeczywistość”⁶⁰¹. Sposób w jaki programiści artykułują w języku stwarza zatem też ich przestrzeń publiczną. Wobec czego proces tworzenia kodu źródłowego zostanie uchwycony w tym podrozdziale poprzez odniesienie do języka, który wkracza w strukturę i relacje wspólnot programistycznych. Należy bowiem uwzględnić perspektywę osób zaangażowanych w tworzenie kodu, gdyż to artykulacja przez nich spraw istotnych sprawia, że mogą „istnieć właśnie te, a nie inne praktyki, uwarunkowania i stosunki”⁶⁰². Uczestnictwo w dyskursach prowadzonych w różnych językach programowania jest podstawą tworzenia wspólnot programistycznych, w kontekście hermeneutyki Ricœura. Ukazanie w oparciu o myśl Cassirera wspólnego „świata sensu” jaki tworzą programiści pozwala również na przedstawienie, jak konstytuuje się przestrzeń publiczna w obszarze języka kodu źródłowego programów komputerowych.

Podobne stanowisko do tego, które reprezentuje Taylor w kwestii konstruowania świata społecznego przedstawił John Searle zauważając, że: “szczególnie zagadkową właściwością społecznej rzeczywistości jest to, że istnieje ona tylko dlatego, iż uważamy, że istnieje”⁶⁰³. Społeczna rzeczywistość jest bowiem zależna od obserwatora, co implikuje „ontologiczną subiektywność”. Oznacza to, że istnienie zjawisk zależy od doświadczania ich przez ludzi⁶⁰⁴. Nie podważa to oczywiście równocześnie współwystępowania możliwości epistemologicznej obiektywności, czyli ustalenia czy coś jest prawdą. Sposób istnienia zjawisk w przestrzeni publicznej jest więc intersubiektywny, czy też, jak preferuje go nazywać

601 Ch. Taylor, *Teoria znaczenia ...*, s. 167.

602 *Ibidem*, s. 173.

603 J. R. Searle, *Ontologia społeczna. Niektóre podstawowe zasady*, [w:] *Język, dyskurs, społeczeństwo: zwrot lingwistyczny w filozofii społecznej*, red. L. Rasiński, Państwowe Wydawnictwo Naukowe, Warszawa 2009, s. 203.

604 *Ibidem*, s. 205.

Searle: „kolektywnie intencjonalny”⁶⁰⁵. Konstytutywna rola języka wiąże się z tak pojętym wpływem języka na rzeczywistość.

Odwołując się do „ontologii społecznej” Searle’a zakłada się istnienie społecznego bytu – wspólnoty programistów opartej na kolektywnych przekonaniach⁶⁰⁶. Działania indywidualne programistów stanowią część działań wspólnych, gdy zaczynają być oparte na współdzielonych intencjach. Kolektywna intencjonalność daje podstawy do zaistnienia faktów społecznych i rzeczywistości społecznej⁶⁰⁷. Amerykański filozof przyrównywał współdziałanie do wykonywania symfonii, gdzie gra na poszczególnym instrumencie jest jej częścią⁶⁰⁸. Ricœur podobnie przedstawiał kwestie powstawania dyskursu, do którego niezbędne było zrozumienie „zdarzenia mowy”. Tekst przyrównał do zapisu muzycznego, „czytelnika – do dyrygenta orkiestry, który postępuje zgodnie z instrukcjami partytury”⁶⁰⁹. Oczywiście rozumienie wykracza poza jedynie powtórzenie, gdyż wymaga interpretacji, jednak punktem wyjścia jest odtworzenie zdarzenia, a efektem jego zobiektywizowanie. Wspólne działanie opiera się na zrozumieniu kontekstu.

Norman Fairclough, jeden z twórców krytycznej analizy dyskursu, wyróżnił trzy obszary przedstawiając to, w jaki sposób dyskurs tworzy „świat znaczący”: przyczyniania się do tworzenia tożsamości społecznej, pomaga w konstruowaniu relacji społecznych, wpływa na tworzenie systemu wiedzy i przekonań⁶¹⁰. Odpowiadają one kolejno trzem funkcjom języka, które wymienił lingwista, a mianowicie identyfikacyjnej (*identity*), relacyjnej (*relational*) i ideacyjnej (*ideational*). Perspektywa Fairclougha zakłada traktowanie języka jako praktyki społecznej a nie czynności indywidualnej. Ricœur dostrzegał w prawdzie aspekt wspólnotowości w dyskursie, ale raczej jako wyraz więzi komunikacyjnej⁶¹¹, która umożliwia przezwycięzenie samotności w ludzkim doświadczeniu. Ustanawianie przez programistów językowej przestrzeni publicznej, w ujęciu myśli francuskiego filozofa, byłoby związana z allokucyjnym/interlokucyjnym aspektem aktu mowy⁶¹² i stanowiłoby rezultat dialogu.

605 *Ibidem*, s. 207.

606 *Ibidem*.

607 J. Searle, *Umysł, język, społeczeństwo*, WAB, Warszawa 1999, s. 213.

608 J. R. Searle, *Ontologia społeczna...*, s. 207.

609 P. Ricœur, *Wyjaśnianie i rozumienie...*, s. 161.

610 N. Fairclough, *Discourse and social change*, Polity Press, Cambridge 1992, s. 64.

611 P. Ricœur, *Język jako dyskurs ...*, s. 84.

612 *Ibidem*, s. 83.

Język jako komunikacja, według Ricœura, posiada wymiar egzystencjalny, gdyż „bycie razem, jako egzystencjalny warunek możliwości dowolnego dyskursu o strukturze dialogicznej, stanowi przekroczenie i przewyciężenie fundamentalnej samotności każdej istoty ludzkiej”⁶¹³. Według tego filozofa, w rozmowie przekazywane są bowiem znaczenia, mimo że doświadczenie zawsze pozostaje czymś indywidualnym. Ludzie nie współdzielą świadomości, jednak to w dialogu powstaje przekazywany sens. Aspekt „zewnątrżności dyskursu”, czyli możliwości przenoszenia sensu, jest podstawą dla współpracy programistów opartej na uchwyceniu istoty jakiegoś problemu, nad którym pracują.

Uczestnictwo w dyskursach prowadzonych w językach programowania różni się w zależności od języka. Inaczej aktywność ducha ludzkiego przejawia się w programach napisanych w językach wysokiego poziomu, w porównaniu do tych niskiego poziomu, ale także w zależności od paradygmatu. Dzieje się tak, gdyż: „język programowania wpływa na sposób pisania programu; istnieje tendencja do stosowania konstrukcji, które są prostsze do wyrażenia w danym języku”⁶¹⁴. Istotą pracy na przykład w języku *Java* jest tzw. obiektowość umożliwiającą lepsze odzwierciedlanie rzeczywistości przez użycie obiektów⁶¹⁵ posiadających pewne właściwości. Obiekty te są grupowane w klasach opisujących zestaw metod – „usług” jakie dane mogą świadczyć⁶¹⁶. Modele danych w tym języku oddają w uproszczony sposób postrzeganie świata przez ludzi. Jako że jest to język wysokopoziomowy, problemy w nim są formułowane na wyższym poziomie abstrakcji, z mniejszą ilością detali niż w językach programowania niskopoziomowych⁶¹⁷.

Warto w tym miejscu opisać sposób działania programistów *Javy*. Pomocny okaże się do tego esej „Execution in the Kingdom of Nouns” czyli „Wykonanie w Królestwie Javy”. Steve Yegger przedstawił w nim baśniowy świat programowania, gdzie w każdym językowym królestwie rządzą inne zasady. W przedstawianym jako złe

613 *Ibidem*.

614 D. E. Knuth, *Sztuka programowania. Algorytmy podstawowe*, T 1, Wydawnictwo Naukowo-Techniczne, Warszawa, 2002, s. XIII.

615 Coś co można wyodrębnić na podstawie właściwości i nazwać.

616 K. Barteczko, *JAVA. Programowanie praktyczne od podstaw*, Państwowe Wydawnictwo Naukowe, Warszawa 2014, s. 21.

617 M. Carro, J. Morales, H. Muller, *High-Level Languages...*, s. 272, http://cliplab.org/papers/carro06:stream_interpreter_cases.pdf (dostęp dnia 18.03.2023).

królestwo, *Java-land*, władanym przez króla Jave, wyznawcę Boga Sun⁶¹⁸, rzeczowniki (obiekty) są uprzywilejowanymi mieszkańcami. Drugą kategorię stanowią przymiotniki, a czasowniki (metody) to niewolnicy⁶¹⁹. Ci ostatni mogą pokazywać się publicznie tylko w towarzystwie rzeczowników, a ich różnorodność jest ograniczana. W królestwie tym architektura znajduje się na wysokim poziomie⁶²⁰, a wszystko jest wyjątkowo zorganizowane (toteż nic niezwykłego w tym, że mieszkańcy są niebywale dumni ze swojej krainy).

W sąsiadujących z *Java-landem* krainach *C-landzie*, *JavaScript-landzie*, *Perl-landzie* i *Ruby-landzie* czasowniki mogą funkcjonować niezależnie. Są też jeszcze odleglejsze królestwa w tym uniwersum: *Haskellia*, *Ocamlica*, *Schemeria* – królestwa języków funkcjonalnych, gdzie panuje równość pomiędzy rzeczownikami i czasownikami (funkcjami). Należą one do tej samej „kasty”, ale w związku z tym, że funkcje są ze swej istoty aktywne, w przeciwieństwie do biernych rzeczowników, miejscami tymi w rezultacie rządzą te pierwsze. W topografii baśniowego świata jest jeszcze kraina położona w najodleglejszym od *Java-landu* regionie, *Lambda the Ultimate*. W obszarze tym nie ma rzeczowników, są jedynie twory je przypominające, a powstałe z czasowników.

W powyższej historii w obrazowy i uproszczony sposób przedstawione zostały sposoby myślenia w ramach różnych paradygmatów programowania. Można utożsamić je z przynależnością kulturową programistów. Środowiskową tożsamość wyznacza bowiem posługiwanie się danym językiem⁶²¹. O przynależności do grupy świadczą podjęte przez programistów działania, w tym przypadku wybór i praca w danym paradygmacie. Według Cassirera właśnie w ten funkcjonalny sposób, poprzez wgląd „w zasadniczą strukturę każdego z tych ludzkich poczynąń”⁶²², można definiować naturę człowieka w ramach kultury.

W sposobie ujmowania rzeczywistości przez programistów, poprzez przyjęcie danego paradygmatu programistycznego, czyli też sposobu myślenia o problemie, uwidacznia się rozpoznawanie dyskursu. Ricœur podkreślał, że we wszelkich użyciach,

618 od Sun Microsystems Inc., przedsiębiorstwa, w którym powstała *Java*.

619 S. Yegge, *Execution in the Kingdom of Nouns*, <https://steve-yegge.blogspot.com/2006/03/execution-in-kingdom-of-nouns.html> (dostęp dnia 11.01.2023).

620 Nawiązanie do poziomu architektury programów.

621 P. Sztompka, *Socjologia. Analiza ...*, s. 291.

622 E. Cassirer, *Esej o człowieku...*, s. 153.

dyskurs czemuś służy. Nie powstaje sam dla siebie⁶²³, wyraża on sens doświadczenia, które staje się rzeczą publiczną⁶²⁴. W opowieści Yeggera programowanie w Javie zasadza się na rzeczowniku-obiekcie, gdyż:

„programowanie zorientowane obiektowo stawia rzeczowniki na pierwszym miejscu. Dlaczego zadawać sobie tyle trudu, by umieścić jedną część mowy na piedestale? Dlaczego jedna koncepcja miałaby mieć pierwszeństwo przed innym? Przecież OOP (programowanie zorientowane obiektowo, *przyp. aut.*) nie uczyniło nagle czasowników mniej ważnymi w sposobie w jaki rzeczywiście myślimy. To jest dziwnie wykrzywiona perspektywa. (...) skrajny nacisk na procesy myślowe zorientowane na rzeczowniki (a co za tym idzie procesy modelowania) jest bardziej niż tylko niepokojący. Każdy typ systemu będzie wymagał przekształcenia swoich myśli, aby pasowały do systemu, ale wyeliminowanie samodzielnych czasowników wydaje się krokiem wykraczającym poza wszelkie racje czy przesłanki.”⁶²⁵

Jako że języki programowania nie są nigdy językami natywnymi, programiści wybierający „przynależność” do Królestwa Javy, muszą dopasować się do panującej w nim kultury, nawet jeśli komuś, z innej kultury, mógłby wydawać się ten sposób myślenia „wykrzywiony” czy „niepokojący”⁶²⁶. Wybór paradygmatu programowania, czyli sposobu ujmowania problemu, umiejscawia programistę w ramach danej wspólnoty językowej.

Konflikt myślenia rzeczownikami i czasownikami, przedstawiony w historii „Wykonanie w Królestwie Javy”, do żywego przypomina różnice między przyswajaniem języka przez dzieci wychowane w kulturach i co za tym idzie - w językach europejskich i wschodnioazjatyckich⁶²⁷. Japońscy rodzice, zamiast koncentrować się na nazywaniu przedmiotów, opisują częściej małym dzieciom relacje,

623discourse never exists for its own sake, for its own glory, but that in all of its uses it seeks to bring into

language an experience, P. Ricœur, *On interpretation*, [w:] *From text to action; Essays in Hermeneutics, II*, red. R. Kearney, Northwestern University Press, Evanston 2007, s. 17.

624 P. Ricœur, *Teoria interpretacji*...s. 85.

625 Tłum. własne “Object Oriented Programming puts the Nouns first and foremost. Why would you go to such lengths to put one part of speech on a pedestal? Why should one kind of concept take precedence over another? It's not as if OOP has suddenly made verbs less important in the way we actually think. It's a strangely skewed perspective. (...) the extreme emphasis on noun-oriented thought processes (and consequently, modeling processes) is more than a bit disturbing. Any type system will require you to re-shape your thoughts somewhat to fit the system, but eliminating standalone verbs seems a step beyond all rationale or reason”, S. Yegge, *Execution in the Kingdom of Nouns*, <https://steve-yegge.blogspot.com/2006/03/execution-in-kingdom-of-nouns.html> (dostęp dnia 11.01.2023).

626 *Ibidem*.

627 R. Nisbett, *Geografia myślenia. Dlaczego ludzie Wschodu i Zachodu myślą inaczej?*, Smak Słowa, Sopot 2009, s. 114.

natomiast amerykańscy głównie obiekty⁶²⁸. Dorastanie w świecie relacji, na przykład w ramach chińskiej kultury, wiąże się zatem z koniecznością ciągłego określania kontekstu, czyli również wysokokontekstowością komunikacji.

Ludzie, wychowując się w danej kulturze, domyślnie przyjmują sposób komunikacji w niej przyjęty. Taka adaptacja wydaje się nie dotyczyć jedynie języków naturalnych, ale również skonstruowanych. Programiści uczą się kodować zwykle w dorosłości, czyli po okresie krytycznym przyswajania języka⁶²⁹, sami wybierając w jakich językach będą pracować. Pisanie w danym języku programowania, wiąże się z myśleniem w podobny sposób, co i inni jego użytkownicy. Można więc powiedzieć, że poprzez wybranie paradygmatu programowania programiści stają się częścią pewnej wspólnoty opartej na danym sposobie myślenia.

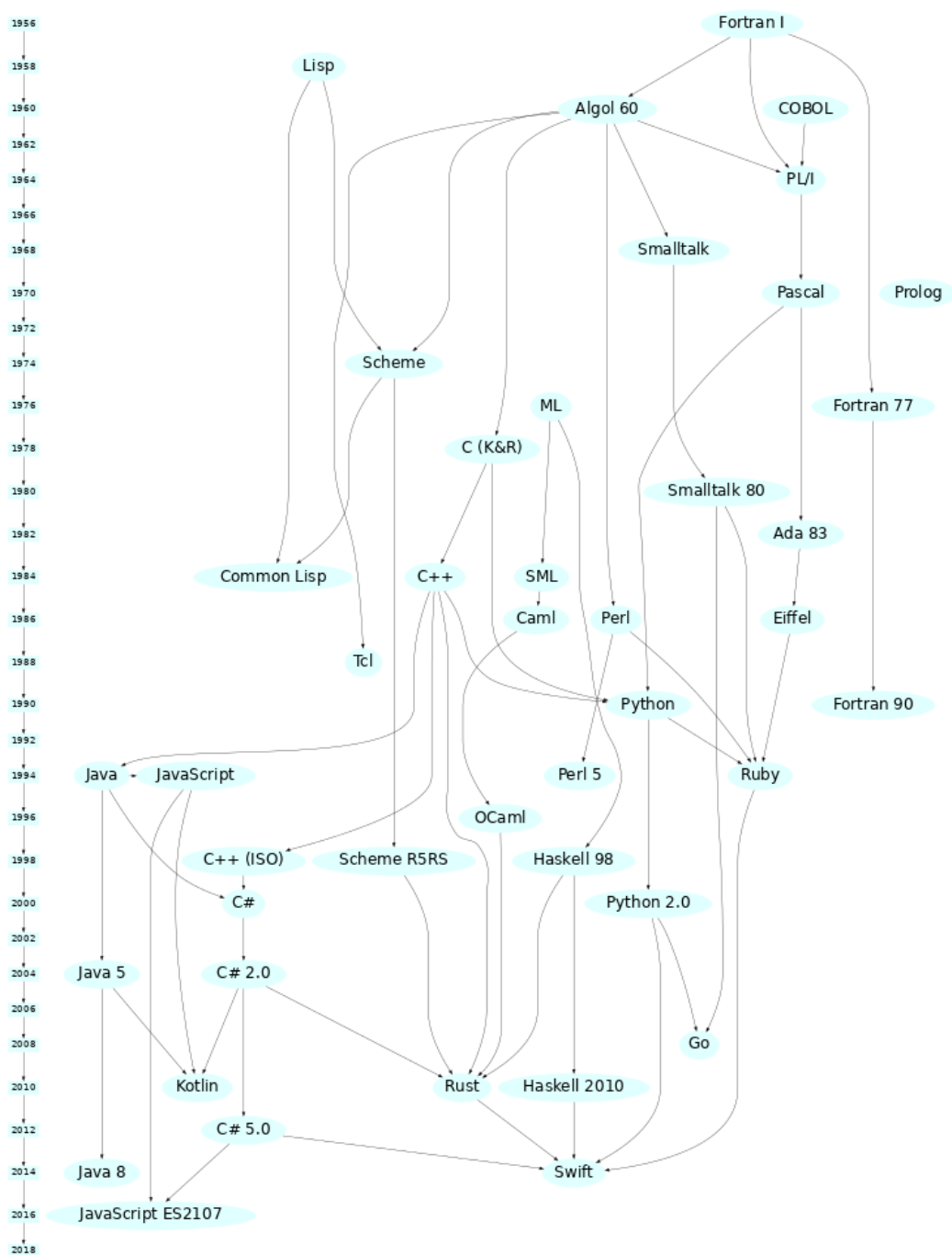
Przyjęło się zarzucać językom skonstruowanym brak oparcia w autentycznych kulturach. Jednak języki te mogą być zakorzenione w dziejach, tak jak chociażby *esperanto* ze swoim zapleczem w obszarze kultur Basenu Morza Śródziemnego⁶³⁰. Nie tylko języki aposterioryczne mają możliwość czerpania z tradycji. Bywa, że również języki programowania mają swoje zaplecze kulturowe. Wskazaniem dla tego jest możliwość przedstawienia ich rozwoju w postaci graficznej, ukazującej procesy powstawania nowych języków na bazie już istniejących (rys. 8, 9, 10)⁶³¹. Uwidoczniona zostaje w ten sposób złożoność powstawania wielu z nich, na drodze swoistej ewolucji, przejmowania elementów innych języków. Można również zaobserwować na tej podstawie podobieństwa między poszczególnymi językami czy ich potencjalne wykorzystanie. Ponadto daje się również dostrzec zmieniające się w czasie trendy, wraz ze zmianami technologicznymi – zmienia się bowiem popularność poszczególnych języków (rys. 11).

628 *Ibidem*,

629 *critical period hypothesis*, Erica Lenneberga zauważył istnienie okresu krytycznego w przyswajaniu posługiwaniem się językiem, po którym jego nabycie przez człowieka nie jest w pełni możliwe np. jak w przypadku tzw. „dzikich dzieci”.

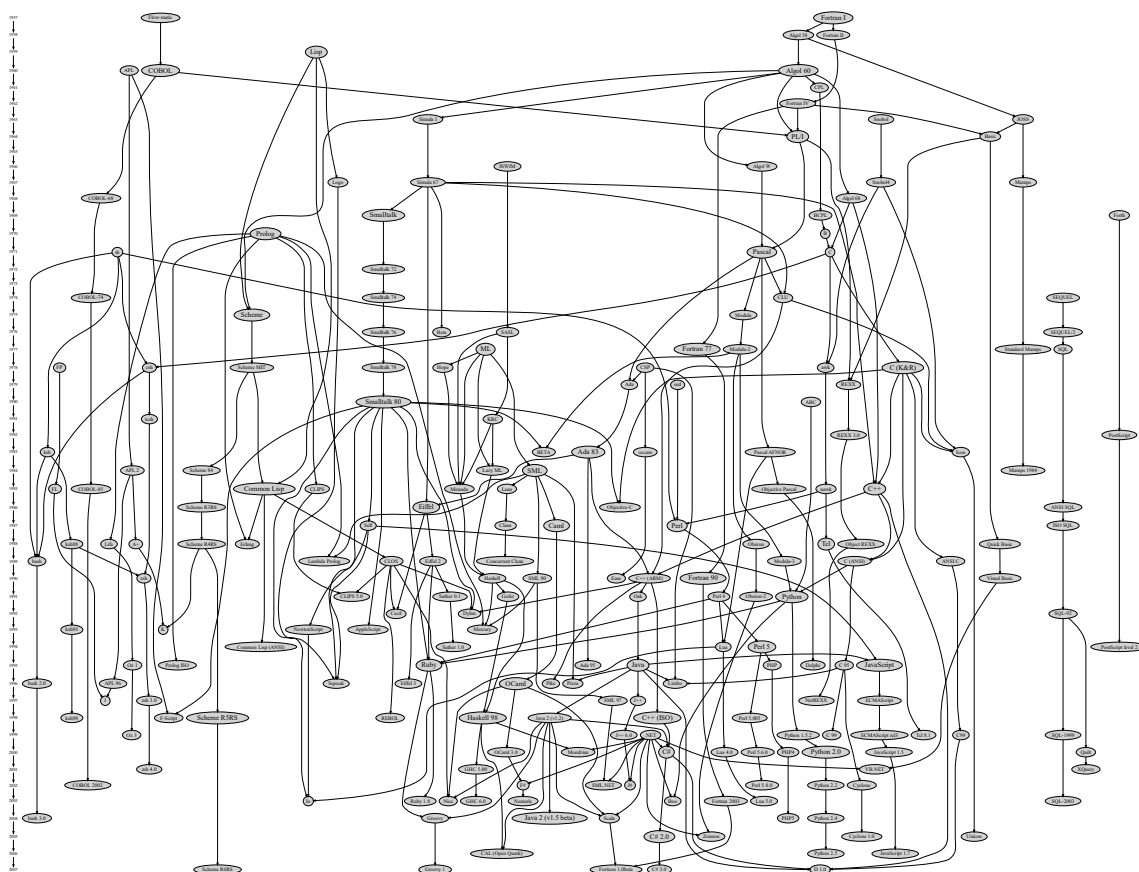
630A. Furdal, *Językoznawstwo otwarte*, Wydawnictwo Zakład Narodowy im. Ossolińskich, Wrocław 2000, s. 195.

631 <http://hopl.info/> (dostęp 22.08.2023).

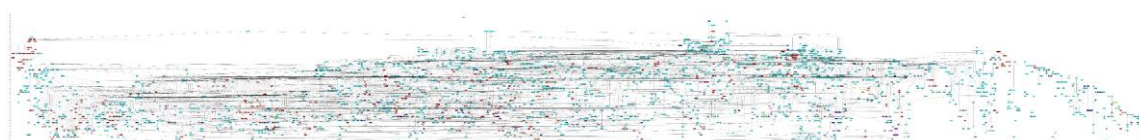


Rys. 8. Uproszczony diagram rozwoju języków programowania⁶³²

⁶³² <https://web.archive.org/web/20201112004756/http://rigaux.org/language-study/diagram-light.png> (dostęp 22.08.2023).



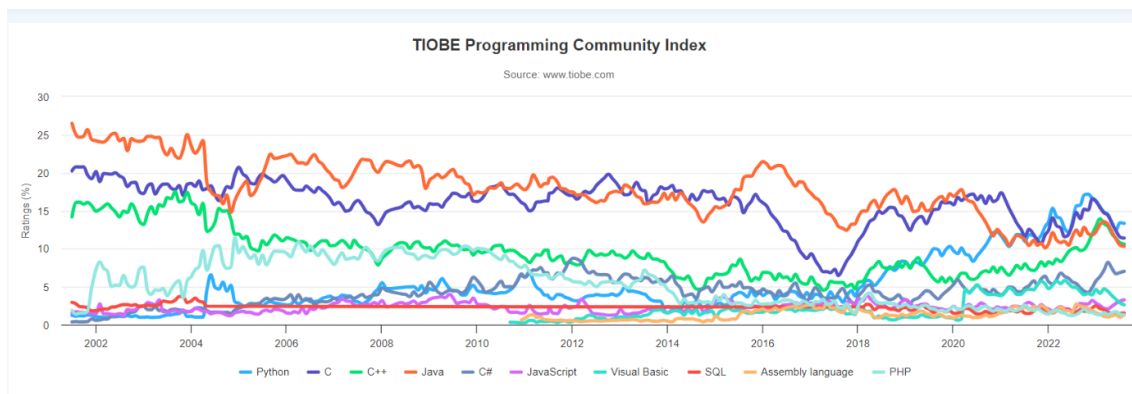
Rys. 9. Rozbudowany i coraz mniej czytelny diagram rozwoju języków programowania⁶³³



Rys. 10. Pomniejszony i już nieczytelny, ze względu na mnogość powiązań i języków, diagram rozwoju języków programowania uwzględniający możliwie wiele, tych znanych autorowi drzewa - Pascalowi Rigaux⁶³⁴

633 <https://github.com/stereobooster/programming-languages-genealogical-tree> (dostęp 17.08.2023).

634 <http://hopl.info/images/genealogies/tester-country.jpg> (dostęp 17.08.2023).



Rys. 11. Zmiany popularności języków programowania⁶³⁵

Przejawem świadczącym o witalności wspólnot, powstałych na gruncie używania tych samych języków programowania, jest organizowanie wydarzeń związanych z programowaniem, dyskusje toczone na tematycznych forach internetowych, czy też istnienie zawodów w programowaniu zespołowym. Społeczności programistów (*Software developer communities*) tworzą się wokół: wydarzeń (na przykład *hackathonów* takich jak *HackMIT*)⁶³⁶, konferencji tematycznych (na przykład *Developer! Developer! Developer!*)⁶³⁷, internetowych forów i blogów propagujących daną technologię (na przykład *Cisco DevNet*)⁶³⁸ lub naukę języków (na przykład *W3Schools*)⁶³⁹, stron internetowych na których można uzyskać i udzielić pomoc (na przykład *StackOverflow*)⁶⁴⁰, czy *Chinese Software Developer Network*)⁶⁴¹. Programiści mogą jednoczyć się również w kolektywach hakerskich (na przykład *Anonymus*) czy wspierać się w pracy zdalnej. Przykładem grupy zachęcającej do solidarności zawodowej jest przykładowo *Arc* – platforma która zrzesza pod wymownym hasłem „zdalnie, ale nie samotnie”⁶⁴².

Martin Buber pisząc o prawdziwej wspólnocie, zauważył że nie jest ona oparta na uczuciach ludzi wobec siebie. Powstaje w rezultacie tego, że „wszyscy oni (członkowie, *przyp. aut.*) pozostają w żywej dwustronnej relacji do pewnego żywego centrum i że pozostają w żywej dwustronnej relacji do siebie nawzajem”⁶⁴³. Właśnie

⁶³⁵ <https://www.tiobe.com/tiobe-index/> (dostęp 17.08.2023).

⁶³⁶ <https://hackmit.org/> (dostęp 17.08.2023).

⁶³⁷ <https://www.developerdeveloperdeveloper.com/> (dostęp 17.08.2023).

⁶³⁸ <https://developer.cisco.com/learning/> (dostęp 17.08.2023).

⁶³⁹ <https://www.w3schools.com/> (dostęp 17.08.2023).

⁶⁴⁰ <https://stackoverflow.com/> (dostęp 17.08.2023).

⁶⁴¹ <https://www.csdn.net/> (dostęp 17.08.2023).

⁶⁴² <https://arc.dev/talent-blog/online-developer-communities/> (dostęp 18.12.2024).

⁶⁴³ M. Buber, *Ja i ty. Wybór pism filozoficznych*, Instytut Wydawniczy PAX, Warszawa 1992, s. 66.

działanie w języku stwarza wspólnoty programistów, które koncentrują uwagę wokół rozwiązywania problemów programistycznych, czy rozwijania języka. Członkowie muszą angażować się w kwestie istotne dla społeczności, gdyż w przeciwnym razie nie ma szansy się ona utrzymać. Gdy blog o tematyce IT traci komentujących i związaną z tym popularność, bywa że w efekcie, zostaje zlikwidowany przez autora, lub zawieszony poprzez uniemożliwienie dodawania kolejnych komentarzy.

Tożsamość zbiorowa według Ricœura wiązuje się na płaszczyźnie działania⁶⁴⁴. Odnajdujemy ją w przedstawieniach zbiorowych powstałych na gruncie więzi społecznej. Te jak je nazywa filozof „symboliczne mediacje” tworzą się z każdorazowego rekonstruowania więzi, która „nie kosztuje w powtórzeniu, lecz objawia się zmianami”⁶⁴⁵. Relacja ze światem, która ogniskuje się w relacji z drugim człowiekiem, jest też źródłem formowania się siebie, gdy następuje oddanie się do dyspozycji innego człowieka⁶⁴⁶. Chociaż społeczności programistów nie są trwałe, gdyż opierają się na współdziałaniu, często zmieniających się osób w grupie, to istnieje między jej członkami rekonstruowana więź. Warto też podkreślić dążenie do rozwoju w ramach tych społeczności, kształtowanie samego siebie i wspierania w rozwoju drugiego człowieka.

W rozwijanie najpopularniejszych języków sztucznych angażuje się olbrzymia społeczność programistów. Najmniej popularne języki umierają natomiast, podobnie jak dzieje się to z językami naturalnymi, gdy drastycznie spada liczba ich użytkowników. Wbrew zarzutom takim jakie postawił, na przykład, Hans-Georg Gadamer w kwestii braku zakorzenienia w tradycji języków sztucznych – gdyż według niego stanowić miały one zaledwie nieprzemawiające sztuczne symbole – ujęcie języków programowania na tle języków naturalnych, pozwala dostrzec dziejowość powstawania również języków programowania. Konstruowane są one często na bazie innych powstałych już języków, co przypomina tworzenie się rodzin języków.

Konstituujący wymiar języków programowania dostrzec można również w sposobie pracy zwanym *mob programmingiem*. Podejście to polega na wspólnym programowaniu. Odbywa się przy udziale grupy programistów, wspólnie korzystających z jednego komputera. Articulacja w języku programowania stwarza

644 P. Ricœur, *Drogi rozpoznania*, Wydawnictwo Znak, Kraków 2004, s. 133.

645 *Ibidem*, s. 135.

646 P. Ricœur, *O sobie samym ...*, s. 278-279.

rzeczywistość, w której ludzie współpracują. Pisanie programu w konwencji *pair programmingu* jest podobne, jednak nie takie same. Zachodzi tu dialog między dwójką programistów piszących kod, *driverem* i komentującym *navigatorem*, którzy zamieniają się co jakiś czas rolami. Odbywająca się wówczas komunikacja buduje relacje między ludźmi, choć przede wszystkim poprawia jakość kodu.

Posiadanie szczegółowej wiedzy o konkretnej implementacji u innej osoby w projekcie, jest przydatne na wypadek nieobecności programisty, stąd istotność procesu dzielenia się nią (*knowledge sharing*), który to proces odbywa się, na przykład, przez robienie dokumentacji, *pair programming*, *mentoring*, a nawet rewizję kodu (*code reviews*)⁶⁴⁷. Przejawem tworzenia wspólnoty są także, omawiane już w kontekście ekspresji komentarze do kodu źródłowego pisane w języku naturalnym. Stanowią pisemną komunikacją z innymi osobami, mającą na celu ułatwienie czytania i rozumienia kodu innym programistom, a bywa, że i także samym twórcom po upływie jakiegoś czasu.

Innym przejawem współpracy programistów jest korzystanie z bibliotek programistycznych, czyli gotowych podprogramów, danych i typów danych dostępnych w danym języku programowania. Programy nie są bowiem pisane „od zera”, ale z wykorzystaniem gotowych elementów, które zostały wcześniej stworzone. Trzeba mieć jedynie na uwadze ryzyko, że pisanie kodu z użyciem biblioteki, która nie ma stabilnego API może zepsuć kod, podczas gdy ta będzie aktualizowana.

Jak zostało podkreślone, praca programistyczna wymaga często współpracy z innymi ludźmi. Niezbędne jest sprawne poruszanie się w ramach kultury danego miejsca pracy, uczenie się współdziałania z osobami o odmiennej od naszej osobowości – w czym pomocne są umiejętności komunikacyjne – a nawet coś co wydawałoby się mało istotne: właściwie szacowanie czasu pracy, aby nie opóźniać jej u innych osób. Ważne jest, jak pisze Steinhart, aby zastanowić się „czy inni ludzie potrafią połączyć się, jak używać naszego kodu”⁶⁴⁸ bo w prawdzie:

„można oczywiście samemu podołać mniejszym projektom, siedząc w ciemnym pokoju z odpowiednią ilością śmieciowego jedzenia. Jednak większość większych projektów to gra

647 <https://dzone.com/articles/how-to-share-knowledge-in-software-development> (dostęp dnia 17.08.2023).

648 J. E. Steinhart, *Sekretne życie ...*, s. 441.

zespołowa i trzeba się w ich czasie kontaktować z ludźmi. To ostatnie jest trudniejsze, niż się zdaje – te systemy sprzętu i oprogramowania, jakie nazywamy ludźmi, zawierają o wiele więcej błędów niż nawet największe bluźnierstwa, jakie zostały skłcone w światku Internetu rzeczy. Nie ma też mowy o tym, by do ludzi dołączona była jakakolwiek dokumentacja.”⁶⁴⁹

Począwszy od stosowania się do stylu wymaganego w konkretnym projekcie, przez tworzenie dokumentacji, budowanie programu w częściach aby umożliwić kilku osobom jednoczesną pracę nad programem⁶⁵⁰, aż po uwzględnienie możliwości dodawania nowych funkcjonalności w celu rozwoju projektu – wszystko to wymaga współpracy programistów. Również w wyłapywaniu błędów we własnym kodzie, pomocne okazuje się testowanie przez innych:

„Jeśli to możliwe, niech kto inny również napisze testy dla twojego kodu. Naturalnym zjawiskiem jest to, że osoba pisząca kod podświadomie unika znanych problemów, przez co nie pisze testów, które mogłyby je wykryć”⁶⁵¹.

Nieodłącznym elementem działania w języku jest bowiem wykorzystanie recenzji uzyskiwanej od innych programistów.

Nie uwzględniając innych osób czytających kod, programista może stworzyć niepoprawną „tajemniczą nazwę”⁶⁵², czyli taką która dla niego w danym momencie była znacząca, ale niekoniecznie będzie w czasie późniejszym i niekoniecznie dla innych osób. Nazewnictwo stanowi wyzwanie programistyczne i kluczowym jego problemem jest wymyślenie takiej nazwy dla struktury (zmiennej, klasy, funkcji, modułu), aby informowała co ta robi i jak się z niej korzysta, w przeciwnym razie może kogoś czekać „wielogodzinne dociekanie przeznaczenia kodu”⁶⁵³, lub – samego programistę – refaktoryzacja. Poprawne, czyli uwzględniające kontekst, nazwy są podstawą do tworzenia rzeczywistości społecznej, w której funkcjonują programiści. Z uwagi na poprawność kodu oczekiwane jest uwzględnienie rozumienia danych słów przez otoczenie. Nazywanie, na przykład, kolejnych zmiennych $a_1, a_2, \dots, a_n$, czyli używając nazw pozbawionych znaczenia, kreuje się przestrzeń publiczną, w której trudniej funkcjonować reszcie programistów.

649 *Ibidem*, s. 442.

650 *Ibidem*, s. 144.

651 *Ibidem*, s. 472.

652 M. Fowler, *Refaktoryzacja. Ulepszanie...*, s. 82.

653 *Ibidem*.

Dla konstytuującego przestrzeń publiczną wymiaru języka, który to umożliwia rozpoznawanie jak język „wkracza w obszar rzeczywistości”⁶⁵⁴ wydaje się istotne nawiązanie do stanowiska znanego jako enaktywizm. W paradygmacie tym wychodzi się poza rozumienie umysłu jako tylko rezultatu aktywności mózgowej. Stanowisko to jest znane też pod nazwą koncepcji „4e” od pierwszych liter angielskich słów: *embodied*, *embedded*, *enacted*, *extended*. Enaktywizm zakłada: zaangażowanie w procesy umysłowe procesów ciała (*embodied*), ścisłe powiązanie ich działania z otoczeniem (*embedded*), zachodzenie tychże procesów umysłowych przy zaangażowaniu organizmu (*enacted*) oraz umiejscowienie ich działania również poza mózgiem (*extended*)⁶⁵⁵.

W kognitywistyce, kwestia partycypacyjnego tworzenia sensu pojawia się w ramach szerzej ujętego enaktywizmu. Gdy przyjmuje się, że wpływ na poznanie ma interakcja człowieka (*enaction*) z jego środowiskiem, można uwzględnić również interakcje społeczne. Neurokognitywistka Marieke Rohde tak pisze o współkonstruowaniu wiedzy w interakcjach społeczno-językowych:

“Z perspektywy enaktywizmu, wiedza nie jest reprezentacją, wiedza jest konstruowana: przez podmiot w jego sensomotorycznej interakcji ze środowiskiem, współkonstruowana między istotami żywymi w obrębie zachodzących między nimi znaczących interakcji. W swojej najbardziej abstrakcyjnej i symbolicznej formie, wiedza jest współkonstruowana między jednostkami ludzkimi w socjo-językowej interakcjach”⁶⁵⁶.

Idea *enaction* zakłada, że człowiek będący biernym obserwatorem nie doświadcza na drodze pozyskiwania informacji, lecz poprzez działanie⁶⁵⁷. Poznanie kształtuje się w interakcji aktora ze światem. Chociaż koncepcja ta skupia się na procesach umysłowych i umiejscowieniu ich w ciele i jego aktywności ze środowiskiem, to nawiązując do słów Rohde, wiedza jest konstruowana w

654 Ch. Taylor, *Teoria znaczenia...*, s. 169.

655 M. Rowlands, *The New Science of the Mind: From Extended Mind to Embodied Phenomenology*, MIT Press, London 2010, s. 3.

656 Tłum. własne „In the enactive view, knowledge is not represented, knowledge is constructed: it is constructed by an agent through its sensorimotor interactions with its environment, coconstructed between and within living species through their meaningful interaction with each other. In its most abstract and symbolic form, knowledge is co-constructed between human individuals in socio-linguistic interactions”, M. Rhode, *Enaction, Embodiment, Evolutionary Robotics. Simulation Models for a Post-Cognitivist Science of Mind*, Atlantis Press, Amsterdam-Paris 2010, s. 30.

657 M. Rocha, *Cognitive, Embodied or Enacted? Contemporary Perspectives for HCI and Interaction*, Reader, Plymouth 2012, s. 4, <https://web.archive.org/web/20140524022911/http://trans-techresearch.net/wp-content/uploads/2010/11/Rocha-01.pdf> (dostęp dnia 18.08.2023).

doświadczeniu z otoczeniem poprzez kontakt społeczny odbywający się na płaszczyźnie językowej. A zatem aktywność poznawczą należy rozumieć jako proces odbywający się częściowo w przestrzeni publicznej. Wszelka wspólna działalność programistów, w której tworzą w językach programowania, wymaga również wyjścia poza wąskie rozumienie umysłu.

Symboliczne przyswajanie zjawisk, w ujęciu Cassirera, podobnie jak enaktywizm, wymaga wykraczania w poznaniu poza jednostkowe doświadczenie. Poznanie świata kultury opiera się bowiem w ujęciu tego filozofa na trzech wymiarach rozumienia rzeczywistości zwanych fenomenami bazowymi⁶⁵⁸ lub inaczej prafenomenami (*Urphänomen*)⁶⁵⁹: nakierowaniu na „ja”, „ty” i „to”. Ścieżka poznania rzeczywistości polegająca na wiedzy o sobie jest równorzędna i współzależna z pozostałymi wymiarami⁶⁶⁰. Wzajemnie konstytuujące się podmioty tworzą w ramach przestrzeni publicznej dzieła. Percepcja świata ma aspekt interpretacyjny, uwzględnia wydobywanie znaczenia w relacji z innymi osobami i wytworami ich pracy.

Marta Węgrzyn zauważyła, że wspomniane prafenomeny „ja”-„ty”-„to” odpowiadają podziałowi na życie-działanie-dzieło⁶⁶¹ zaproponowanym przez Geothego. Poprzez sensotwórczą działalność człowiek jest w stanie nadać duchowe znaczenie treści, którą tworzy. Robi to we „»wspólnym świecie« sensu”⁶⁶², powstałym w rozmowie z innymi ludźmi, czyli poprzez uwzględnienie relacji „ja”-„ty”. Wymiar „to”, czyli nośnik duchowego znaczenia, ucieleśnia się w przedmiotach kulturowych tworzonych przez człowieka⁶⁶³. Tekst kodu źródłowego programów komputerowych, będąc takim przedmiotem, ma szansę funkcjonować w przestrzeni publicznej w powyższy sposób, jeśli uwzględni się triadyczny model „ja”-„ty”-„to”.

Chociaż nie jest szeroko znana, gdyż pojawia się jedynie w tekście „Ziele und Wege der Wirklichkeitserkenntnis”⁶⁶⁴, teoria percepcji i znaczenia ujęta przez

658 M. Węgrzyn, *Ja-Ty-Dzieło jako źródło poznania. Między neokantyzmem a hermeneutyką: Cassirer i Ricœur*, „Hybris”, 50, nr 3, 2020, s. 24-25, <http://www.czasopisma.uni.lodz.pl/hybris/article/view/14467/13821> (dostęp dnia 22.08.2023).

659 P. Parszutowicz, *Fenomenologia form...*, s. 235.

660 M. Węgrzyn, *Ja-Ty-Dzieło...*, s. 26.

661 *Ibidem*, s. 24-25.

662 *Ibidem*, s. 26.

663 *Ibidem*.

664 P. Parszutowicz, *Fenomenologia form...*, s. 14.

prafenomeny – jak zaznacza badacz teorii transcendentalizmu⁶⁶⁵ Przemysław Parszutowicz – jest charakterystycznym rysem cassirerowskiej fenomenologii kultury. Fenomen bazowy „ja” stanowi wiedzę o sobie, „ty” o innym podmiocie, a w relacji „pomiędzy” wspomnianymi tworzyć ma się trzeci fenomen bazowy, czyli „to” – dzieło. Sens powstaje w rozmowie, czyli językowej przestrzeni publicznej. W tym podrozdziale zostało przedstawione jak konstytuuje się ten obszar na przykładzie myślenia w różnych paradygmatach programowania. Odniesiono się również do tworzenia przestrzeni publicznej na gruncie wspólnoty ludzi zaangażowanych w rozwiązywanie problemów, podczas różnego rodzaju współpracy programistycznej.

Aktywność ludzkiego ducha, a uściślając: ducha programistów, ucieleśnia się w kodzie źródłowym jako przedmiocie kulturowym. Dzieje się tak w warunkach językowej przestrzeni publicznej, która jest konstytuowana przez język. Postrzeganie rzeczy nakierowane jest więc międzypodmiotowo. Jak zauważa Przemysław Parszutowicz, w kontekście cassirerowskich fenomenów bazowych, powstałe między „ja” i „ty” „relacje podmiotowe budują kulturę”⁶⁶⁶. Warto jednak przyjrzeć się uważniej jak funkcjonuje w języku kodu źródłowego również prafeomen „ja”. Indywidualność, z inspiracji Taylora, nazwana w tej pracy „artykulacją indywidualności w języku”, stanowi temat kolejnego rozdziału.

665 <https://www.ahf.ifispan.pl/pl/parszutowicz> (dostęp dnia 27.12.2024).

666 P. Parszutowicz, *Fenomenologia form...*, s. 153.

3. Artykulacja indywidualności

„Mój język i każda krwinkę stworzyła ta gleba”⁶⁶⁷

Język ustanawiający przestrzeń publiczną ma, jak już zostało przedstawione w II rozdziale, charakter intersubiektywny. Umieszcza sprawy istotne pomiędzy rozmówcami i z tego miejsca umożliwia „wspólne przyglądanie się światu”⁶⁶⁸. Jednak, aby mogło dojść do tej Searl’owskiej kolektywnej intencjonalności⁶⁶⁹ trzeba uwzględnić wprzód artykulację indywidualności. By w ogóle dialog mógł się wydarzyć, konieczne jest, aby „ja” przemówiło, „uświadomiło sobie to, czego wcześniej mieliśmy tylko niejasne przeczucie”⁶⁷⁰. Z tej pozycji bowiem tworzone są w języku artykulacje.

W języku⁶⁷¹ formowane jest to, co wpierw „intuicyjnie ujmuje”⁶⁷² człowiek. Dzięki wyartykułowaniu rzeczy mogą między innymi stać się częścią ludzkiej świadomości, a przez wysłowienie być też dostępne dla innych osób. Tak można by po krótku, z perspektywy psychoanalitycznej, ale za pomocą terminologii filozoficznej, przedstawić ścieżkę pomiędzy *id*⁶⁷³ i *ego*⁶⁷⁴. Chociaż teorii Freuda nie udało się naukowo zweryfikować, to psychologia poznawcza nie odrzuciła koncepcji nieświadomości. Interakcje procesów świadomych i nieświadomych bada się chociażby za pomocą techniki torowania - na przykład w kontekście pamięci utajonej⁶⁷⁵. Uzasadnione wydaje się zatem posługiwanie terminem świadomość, nawet jeśli

667 W. Whitman, *Pieśń o mnie*, [w:] *Pieśń o mnie i inne utwory*, tłum. A. Szuba, Wydawnictwo Miniatura, Kraków 1996, s. 5.

668 Ch. Taylor, *Teoria znaczenia...*, s. 157.

669 J. R. Searle, *Ontologia społeczna...*, s. 207.

670 Ch. Taylor, *Teoria znaczenia...*, s. 155.

671 Myśl oczywiście może pojawić się w formie obrazu, lecz z uwagi na tematykę pracy ten wątek nie jest rozwijany.

672 Ch. Taylor, *Teoria znaczenia...*, s. 156.

673 „Prymitywna, nieświadoma część psychiki, w której mieszczą się najbardziej podstawowe popędy, i w której są przechowywane wyparte wspomnienia” P. Zimbardo, R. Johnson, V. McCann, *Psychologia. Kluczowe koncepcje. Psychologia Osobowości*, Państwowe Wydawnictwo Naukowe, Warszawa 2013, s. 251.

674 „Świadoma, racjonalna część osobowości, której zadaniem jest utrzymywanie między *id* a *superego*”, *Ibidem*, s. 250.

675 P. Zimbardo, R. Johnson, V. McCann, *Psychologia. Kluczowe koncepcje. Struktura i funkcje świadomości*, Państwowe Wydawnictwo Naukowe, Warszawa 2010, s. 205.

miałoby być ono odarte z poetyckości i definiowane jako „aktywności mózgu umożliwiającej konstruowanie modelu mentalnego naszego doświadczenia”⁶⁷⁶.

Posiadanie i wyrażenie poglądu, umożliwia uchwycenie tego, co indywidualne w człowieku, jego własnej perspektywy. W języku artykułuje się sposób, w jaki ludzie zamieszkują świat⁶⁷⁷, również ich jednostkowe sposoby doświadczenia. Małgorzata Poczyńska z pozycji psychologa egzystencjalnego pisała, że:

„język zdradza moje istnienie. (...) Po pierwsze, wyjawia to co bez niego musiałoby pozostać bez wyrazu (...); po drugie, dopuszcza się niewierności: żaden wyraz dźwięczny czy bezdźwięczny (...) nie wyrazi wszystkiego, co jest”⁶⁷⁸.

W tej dychotomii zawarte są granice możliwości formułowania myśli w języku. Chociaż ujawnia się tu bezradność i niemoc dotycząca sytuacji, w których można tylko milczeć⁶⁷⁹, to jednak artykulacja w języku zazwyczaj jest możliwa, nawet jeśli nigdy nie będzie pełna. Mówienie o sprawach metafizyki można by uznać za Wittgensteinem za niezadawalającą⁶⁸⁰. Podobnie mówienie o sobie jako zawsze niepełne, gdyż nie jesteśmy w stanie całkiem siebie poznać i wyrazić⁶⁸¹.

Wysłowienie, czy to w piśmie, czy w mowie jest procesem wymagającym stworzenia mentalnej reprezentacji przekazu, a następnie planowego wykonania⁶⁸². Następuje kolejno sięgnięcie do pamięci semantycznej po słowa zgodne z sensem przekazu i użycie ich zgodnie z regułami gramatyki, zachowując porządek składniowy. Na przykład według psycholingwisty Willema Levelta leksykalizacja zaczyna się od wyboru *lemm*, czyli abstrakcyjnych słów posiadających znaczenie, funkcję syntaktyczną i wskazówki fonologiczne – na przykład ilość sylab i występowanie głosek⁶⁸³. Proces zwróćmy kodowanie fonologiczne. Istotne dla tej pracy nie jest jednak wybranie konkretnego modelu produkcji mowy, na przykład Modelu Lemma, lecz

676 *Ibidem*, s. 263.

677 H.-G. Gadamer, *Język i rozumienie*, [w:] *Język i rozumienie*, Fundacja Aletheia, Warszawa 2003, s. 24.

678 M. Poczyńska, *Pasaże egzystencji. Trony psychoterapii i inne strony*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2019, s. 49.

679 L. Wittgenstein, *Tractatus logico-philosophicus*, Państwowe Wydawnictwo Naukowe, Warszawa 1997, s. 83.

680 *Ibidem*, s. 82.

681 M. Poczyńska, *Pasaże egzystencji*, ...s. 49-50.

682 E. Nęcka, J. Orzechowski, B. Szymura, *Psychologia poznawcza...*, s. 612.

683 D. Rutvik, N. Riccardi, *Cognitive Neuroscience of Language*, [w:] *The Routledge Handbook of Cognitive Linguistics*, red. J.R. Taylor, X. Wen, Routledge, New York 2021, s. 17, https://www.researchgate.net/publication/350495833_Cognitive_Neuroscience_of_Language (dostęp dnia 1.09.2023).

podkreślenie, że planowanie mowy jest procesem „do którego mamy bardzo ograniczony dostęp introspekcyjny”⁶⁸⁴, podobnie z resztą jak do innych etapów produkcji mowy.

Gdyby nawet przetwarzanie języka uznać, odwołując się do psychologii poznawczej, za szyfrowanie i deszyfrowanie myśli⁶⁸⁵, to nadal nie służy ono jedynie do referencji. Istotną funkcją psychiczną pełnioną przez język jest bowiem także ekspresja. Wiąże się ona z wyrażaniem stanów podmiotu w języku⁶⁸⁶. Współdzielimy zdolność ekspresji z częścią innych zwierząt, natomiast różnica między nami polega na ludzkiej wzmożonej potrzebie takiej artykulacji w języku, zakładanej większej złożoności naszej psychiki, w tym zróżnicowania stanów emocjonalnych⁶⁸⁷ oraz bardziej skomplikowanych kontaktach społecznych, co wpływa też na prezentowane treści⁶⁸⁸. Emocje stanowią jeden z podstawowych wymiarów temperamentu i osobowości⁶⁸⁹. W przeciwieństwie do ekspresji więzi omawianej w podrozdziale 2.1, ekspresja indywidualności w języku wydaj się być ujęciem odnoszącym się bardziej do tego co podstawowe. Artykulacja w języku wychodzi bowiem pierwotnie z pozycji *Ja*, nawet jeśli ma na nią wpływ filtr społeczny.

3.1. Wymiar ekspresji

Jako jeden z aspektów działania w języku wyszczególniliśmy, za Taylorem, ekspresję⁶⁹⁰. Wyrażanie siebie jest niezbędnym mechanizmem psychologicznym, który buduje podmiotowość. Niemożność czynienia, w świetle filozofii działania Ricceura, odczuwana jest przez człowieka „jako naruszenie integralności siebie”⁶⁹¹. Brak więc możliwości wyrażenia siebie sprawia cierpienie, nie tyle w wymiarze bólu fizycznego czy umysłowego, lecz emocjonalnego, ponieważ uniemożliwia realizację tej potrzeby.

684 E. Nęcka, J. Orzechowski, B. Szymura, *Psychologia poznawcza...*, s. 612.

685 *Ibidem*, s. 590.

686 E. Nęcka, J. Orzechowski, B. Szymura, *Psychologia poznawcza...*, s. 591.

687 W. Łosiak, *Psychologia emocji*, Wydawnictwa Akademickie i Profesjonalne, Warszawa 2007, s. 30,

688 E. Nęcka, J. Orzechowski, B. Szymura, *Psychologia poznawcza...*, s. 591.

689 J. Zagrodzka, *Neurofizjologiczne mechanizmy zachowania emocjonalnego*, [w:] *Mózg a zachowanie* red. T. Górską, A. Grabowska, J. Zagrodzka, Państwowe Wydawnictwo Naukowe, Warszawa 2012, s. 397.

690 Ch. Taylor, *Teoria znaczenia...*, s. 161.

691 P. Ricœur, *O sobie samym...*, s. 316.

W możliwości artykulacji zapośredniczona jest droga do siebie samego „tego-który-jest-sobą”⁶⁹². Można zatem powiedzieć, że, tożsamość narracyjna kształtuje się w zdolności wypowiedzania. Artykulacja jest przeciwieństwem bierności i ten rodzaj dawania pochodzi „z mocy działania i istnienia”⁶⁹³.

Zaznaczanie emocji w języku może odbywać się jako ich „komunikowanie”, gdy się o nich mówi lub „przeżywanie”, gdy te mówieniu towarzyszą⁶⁹⁴. Ich obecność bywa zamierzona, wtedy są „wyrażane” lub niezamierzona podczas „przejawiania”. Taki drobiazgowy podział jest zapewne istotny z perspektywy badań psychologicznych, natomiast w prowadzonym namyśle nad językiem, słów tych będziemy używać nie zachowując takiej ścisłości terminologicznej. Z samych badań neurobiologicznych i psychologicznych natomiast warto skorzystać rozważając temat pojawiania się emocji.

Według uznawanej już za przestarzałą teorię psychologa Paula MacLean, mózg dzielić się miał na trzy współpracujące ze sobą części: umiejscowiony w pniu mózgu kompleks R, układ limbiczny i neocortex tzw. „myślący mózg”, odpowiedzialny za myślenie logiczne, mówienie i ocenianie zachowania. Emocje związane miały być z układem limbicznym, tzw. emocjonalną częścią mózgu⁶⁹⁵. Genezy emocji można jednak poszukiwać nie tylko w systemie automatycznym, jak się przyjęło, ale również refleksyjnym⁶⁹⁶. Do pobudzenia ciała migdałowatego, a więc powstania afektu, może dojść bowiem dwoma drogami. Jak przedstawił w swoim „modelu dróg nerwowych przebiegu impulsów”, neurobiolog Joseph LeDoux⁶⁹⁷, jedna z dróg wiedzie bezpośrednio od wzgórza (droga dolna), druga zaś przez przekazanie impulsów z kory mózgowej. Wartościowanie świata ma odbywać się zatem bądź w efekcie spontanicznego reagowania, bądź przemyślenia. Według Janusza Reykowskiego, który uzupełnił model LeDoux o koncepcje psychologiczne, podstawę dla oceniania rzeczywistości stanowią mają standardy poznawcze, których źródłem byłyby pojęciowe standardy „ja” oraz pojęcia aksjologiczne⁶⁹⁸. Można przypuszczać zatem, że wyrażane

692 A. Warmbier, *Tożsamość, narracja i hermeneutyka siebie. Paula Ricœura filozofia człowieka*, Universitas, Kraków 2018, s. 41.

693 P. Ricœur, *O sobie samym ...*, s. 317.

694 M. Stępień-Nycz, *Emocje w języku, Komunikowanie i odczytywanie emocji wyrażanych w języku w biegu życia*, [w:] *Język w biegu życia* red. M. Kielar-Turska, Harmonia Universalis, Gdańsk 2019, s. 449.

695 K. W. Vopel, *Warsztaty – skuteczna forma nauki*, Jedność, Kielce 2004, s. 33.

696 M. Jarymowicz, K. Imbir, *Próba taksonomii ludzkich emocji*, „Przegląd Psychologiczny”, 53, nr 4, 2010, s. 457.

697 *Ibidem*, s. 445.

698 *Ibidem*, s. 453.

przez programistów emocje, których ślad zostaje w kodzie źródłowym, nie muszą być wzbudzone przez afekt pierwotny, lecz kształtować się w umysłowych wizjach.

Koncepcję LeDoux-Reykowskiego dotyczącą źródeł emocji oraz wiedzę z zakresu psychologii poznawczej na temat przetwarzania języka, warto zestawić z przemyśleniami Cassirera odnośnie afektu. Według tego filozofa ekspresja językowa przechodzi w przedstawienie w języku, gdy „zwykle subiektywne odczucie i pobudzenie przemienia się w obiektywny ogląd”⁶⁹⁹. Dwubiegunowy charakter, jak nazywa ten proces filozof, ukazuje powstawanie nowej formy z „dźwięku uczucia w dźwięk przedstawienia”⁷⁰⁰. Nadal jednak w powstałej formie tkwi rdzeń ekspresji. Filozof konstatuje: „nawet w słowach naszego wysokorozwiniętego języka tkwi wciąż jeszcze pewien określony charakter wyrazu, charakter »fizjognomiczny«”⁷⁰¹. Cassirer nawiązując do wspomnianych „przeżyć językowych”, zaznacza jednak, że język spełnia się dopiero, gdy wykracza poza nie, czyli już nie w materialnym środku (choć się nim posługuje), lecz w nowym sensie. Barwa dźwięku utrwala według niego przedmiotowy sens. W kontekście działalności programistycznej, może również dojść do tej sytuacji, podczas czytania przez programistę tekstu kodu źródłowego na głos.

Odczytywanie na głos kodu, czy też tzw. czytanie ciche przez programistę nie jest praktyką powszechną, ale możliwą. Wyrażenie takie jak “.” może zostać przeczytane jako “dot” (ang. “kropka”), natomiast “:” bywa czytane już jako „in” (ang. „w”) wskazując tym samym na znaczenie tego znaku. Przykładowa linia kodu “for (String line: f.lines())” może zostać przeczytana jako: “for string line in f dot lines”⁷⁰². W ten sposób programista przedstawia w formie dźwiękowej sposób, w jaki rozumie tekst a nie jedynie prezentuje go literalnie. Takie uzewnętrznianie, jak nazywał ten proces Ricœur, pozwala „wzniesć część naszego doświadczenia w logos dyskursu”⁷⁰³. Filozof przedstawia bowiem język jako proces, w którym doświadczenie staje się publicznym. Dzięki językowi „doznanie przekracza siebie i staje się ekspresją lub

699 E. Cassirer, E. Cassirer, *Problem symbolu i jego miejsce w systemie filozofii*, [w:] *Symbol i Język...*, s. 55.

700 *Ibidem*.

701 *Ibidem*.

702 “Many people read the colon (:) as *in* when they read the code out loud or to themselves” stwierdza Owen Astrachan w *Coursea: Java Programming: Solving Problems with Software: Week 1*, “For each loop”, 4:28 minuta nagrania, Coursea.org Duke University.

703 P. Ricœur, *Język jako dyskurs...*, s. 89.

inaczej mówiąc, transformacją psychicznego w noetyczne”⁷⁰⁴. Klaruje się, więc w języku to jak programista zrozumiał tekst, jakie miał on dla niego znaczenie.

Na praktykę czytania fragmentów kodu przez programistów wskazuje również zalecenie stworzone przez „guru czystego kodu”, Roberta Martina, polegające na tym aby tworzyć nazwy, które można wymówić⁷⁰⁵. Odczytywanie nie jest bowiem działaniem powszechnym, ale ukazuje jak posługiwanie się językiem programowania bywa podobne do sposobu używania języka naturalnego. Za ilustrację artykulacji w mowie posłużyć może przykład humorystycznego fragmentu bazującego na czytaniu, a może nawet w tym wypadku raczej na śpiewaniu kodu i komentarza do niego:

```
doRun.run(); // ... "a doo run run"706
```

Twórca tego fragmentu nawiązał bowiem w komentarzu (//... "a doo run run") do piosenki „Da doo ron ron” zespołu Crystals. Fragment kodu (*doRun.run()*), gdy go wypowiedział na głos brzmiał w jego uszach podobnie do tego chwytliwego tekstu.

Czytanie na głos nie stanowi częstej praktyki wśród programistów, ale zdarza się na tyle często, że warto zastanowić się nad chociażby kwestią nazewnictwa zmiennych. Jest to indywidualna decyzja w jakim języku naturalnym będą tworzone. Najczęściej wybór pada na angielski, lecz zdarza się, że w projekcie używa się języków lokalnych, na przykład polskiego⁷⁰⁷. Uznać można to jednak za dość mało profesjonalne i o ile wykorzystanie języka narodowego zdarza się w komentarzach, to rzadko do zapisania nazwy zmiennej. Bywają oczywiście przypadki żartobliwego użycia języka polskiego by wyrazić miłość do niego idiogramem <3 (rys. 12), natomiast stanowi to wyjątek. Najistotniejsze jest, aby nazwy zmiennych miały sens dla piszącego i czytającego tekst. Na traktowanie kodu źródłowego jako tekstu wskazuje zatem także posługiwanie się *regex'em* (wyrażeniami regularnymi)⁷⁰⁸ do wyszukiwania i zmiany nazw. Wyrażanie się w języku, nawet programowania, powinno uwzględniać również odbiorcę. O ironio może być nim z upływem czasu również sam twórca kodu, więc jeśli wymyślił nazwy mało adekwatne, sam utrudni sobie zrozumienie kodu.

704 *Ibidem*.

705 R. Martin, *Czysty kod...*, s. 43.

706 <https://stackoverflow.com/a/194475> (dostęp dnia 04.09.2023).

707 Ankieta i dyskusja na forum <http://forum.php.pl/index.php?showtopic=244179> (dostęp dnia 04.09.2023).

708 <https://www3.ntu.edu.sg/home/ehchua/programming/howto/Regexe.html> (dostęp dnia 27.09.2023).

1. <?php
2. \$Gzegzółka = "I <3 PL";
- 3.
4. echo \$Gzegzółka;

Rys. 12. Przykład kodu napisanego w PHP z użyciem języka narodowego. Zawiera nazwę zmiennej „gzegzółka” (będącego ilustracją trudnego, polskiego słowa) oraz ideogram oznaczający miłość „<3”⁷⁰⁹

Z perspektywy filozofii form symbolicznych to, co zmysłowe świadomie łączone jest przez człowieka z idealnym znaczeniem. Forma symboliczna reprezentuje bowiem „każdą energię ducha, poprzez którą duchowa treść znaczeniowa zostaje powiązana z konkretnym zmysłowym znakiem i temu znakowi wewnętrznie przypisana”⁷¹⁰. Nazywając, programista świadomie syntezuje w formie myślowej sposób odbioru przez niego świata. W symbolu następuje oddzielenie od zmysłowości, które jednocześnie daje przestrzeń wewnętrzną.

Cokolwiek zostanie wyartykułowane w kodzie źródłowym, na przykład wspomniane podobieństwo kodu i tekstu piosenki, czy deklaracja miłości do języka polskiego (rys. 12), jako dyskurs staje się według Ricoëura czymś trwałym. Mimo, że akt dyskursu może być od nowa rozpoznawany, czy tłumaczony, to „we wszystkich tych transformacjach (...) zachowuje tożsamość (...) »tego co powiedziane«”⁷¹¹. Nie jest zatem tymczasowy. W przytoczonych sytuacjach to, co komunikuje programista jest artykulacją tego, jak postrzega świat i siebie w nim. Potrafi dostrzec podobieństwo dźwięków i odnieść je do znanej sobie piosenki lub nawiązuje do możliwości pisania kodu z użyciem języka polskiego, nawet jeśli czyni to tylko przy użyciu żartobliwej konwencji. W obu podanych przypadkach programista korzysta z odniesienia do świata kultury, w której jest zakorzeniony.

Kwestia znaczenia indywidualności w języku ujawnia się również w wyborach, które podejmują programiści. Należy do nich, na przykład, decyzja o przyjęciu danego stylu programistycznego. Styl określają komentarze, nazywanie zmiennych, rozmieszczanie tekstu, czy też zastosowanie konkretnych wzorców programowania. Jednocześnie brak stosowania się do przyjętych konwencji, w danej językowej

709 <https://ideone.com/dskl7r> (dostęp dnia 04.09.2023).

710 H. Buczyńska-Garewicz, *Cassirer...*, s. 62.

711 P. Ricoëur, *Język jako dyskurs...*, s. 76.

społeczności, i kierowanie się jedynie swoim gustem, również nie jest wskazany, gdyż może generować więcej błędów⁷¹². Właściwe jest, aby styl był przejrzysty⁷¹³, a także by był konsekwentnie stosowany. W zależności od języka dowolność w formatowaniu kodu może być różna. W większości przypadków, programistów ograniczają tzw. dobre praktyki i bycie rozważnym⁷¹⁴. Ujmując to słowami Ricœura „każdy członek tłumy porzuca swoją dziką wolność z myślą o odzyskaniu jej”⁷¹⁵ tyle, że dzieje się to w ramach pewnej umowy społecznej pomiędzy programistami.

Wśród indywidualnych nawyków programistów, chociaż mogłoby się wydawać, że ma to minimalny wpływ na estetykę kodu, konfliktogenny okazuje się wybór pomiędzy stosowaniem spacji i tabulatora do oznaczenia znaków niedrukowalnych (*whitespace*)⁷¹⁶. W większości języków nie ma różnicy, jakiego sposobu użyje programista do zaznaczenia odstępu⁷¹⁷. Użycie jednego tabulatora równa się czterem lub ośmiu spacjom. Argumentem, za stosowaniem tabulacji, oprócz energooszczędności, jest potencjał przenoszalności, jednak to użycie spacji jest bardziej powszechne⁷¹⁸. Wykorzystanie tabulatora ma natomiast swoje źródła w dawnej konieczności oszczędzania miejsca. Nawiązanie w swoim zachowaniu do zachowania „pierwszych programistów” może być tym samym nobilitujące.

Wyrażać się w języku siebie można również przez twórczość. Konstruowanie nowych niszowych języków programowania, ma wymiar głównie ludyczny, gdyż:

„to dobra zabawa (...) Dzisiaj dostępna jest ogromna liczba języków. Większość z nich nie wnosi tak naprawdę nic cennego i jest tylko odzwierciedleniem gustów projektanta”⁷¹⁹.

Powstają zatem takie przedsięwzięcia jak *Whitespace*, czyli sztuczny język stworzony przez Edwina Brady i Chrisa Morrisa na Prima Aprilis⁷²⁰. Instrukcje zapisane są w nim

712 <https://www.crockford.com/code.html> (dostęp dnia 05.09.2023).

713 K. Barteczko, *JAVA. Programowanie praktyczne od podstaw*, Państwowe Wydawnictwo Naukowe, Warszawa 2014, s. 23.

714 https://perldoc.perl.org/perlstyle_ (dostęp dnia 06.09.2023).

715 P. Ricœur, *O sobie samym...*, s. 380.

716 <https://kamil.kwapisz.pl/gramatyka-dla-programisty/> (dostęp dnia 06.09.2023).

717 Są języki, w których ma znaczenie np. w *Pythonie*.

718 <https://hoffa.medium.com/400-000-github-repositories-1-billion-files-14-terabytes-of-code-spaces-or-tabs-7cfe0b5dd7fd> (dostęp 07.09.2023).

719 J. E. Steinhart, *Sekretne życie ...*, s. 24.

tylko „białymi znakami” (rys. 13). Kolejnym przykładem jest język *ErrorFree*, z zawsze kompilującym się kodem⁷²¹, a także jego przeciwieństwo – język *ErrorFull*. Należą one do rodziny ezoterycznych języków programowania (*esolangów*). *Esolangi* powstają jednak nie tylko jako wyraz poczucia humoru jednostek, ale także w celu hakowania interfejsu, jako sposób artystycznego wyrazu oraz praca koncepcyjna⁷²².



Rys. 13. Podświetlona składnia programu napisanego w języku *Whitespace*⁷²³. „Białe znaki” zostały zakolorowane, w celu ukazania niewidocznej składni

Twórca kilku z *esolangów* Scott Feeney tak opisał akt dyskursu, który odbywa się podczas tworzenia tych projektów:

„Myślę, że najciekawsza w *esolangach* jest konwersacja pomiędzy językami, które zadają pytania i programami, które na nie odpowiadają. Kiedy budujesz nowy ezoteryczny język z dziwnym zestawem założeń, dajesz ludziom do myślenia: ciekawe czy mogą zrobić X

720<https://developers.slashdot.org/story/03/04/01/0332202/New-Whitespace-Only-Programming-Language> (dostęp dnia 07.09.2023).

721 <https://blog.slaks.net/2014-04-01/programming-without-errors-errorfree/> (dostęp dnia 07.09.2023).

722 D. Temkin, *Language Without Code: Intentionally Unusable, Uncomputable, or Conceptual Programming Languages*, „Journal of Science and Technology of the Arts”, 9, nr 3, 2017, s. 83. <https://journals.ucp.pt/index.php/jsta/article/view/7297> (dostęp dnia 07.09.2023).

723 <https://justinohms.medium.com/arcane-programming-whitespace-839a77d7c98d> (dostęp dnia 29.12.2024).

w tym języku? Ciekawe czy jest sposób na Y? I rozpracowanie tego poprzez napisanie programu, który wykonuje X i Y może być rozrywkowym wyzwaniem”⁷²⁴.

Oczywiście ani język nie jest zdolny do zadawania pytań ani program nie odpowiada. Autor języka tworzy jego koncepcję a użytkownicy na drodze prób sprawdzają to, jak może działać program. Jest to praca eksperymentalna. Zachodzi zatem dialog, w którym użytkownik-programista zagaduje twórcę o możliwości jego dzieła. Taki „układ pytanie-odpowiedź stanowi podstawową strukturę dyskursu”⁷²⁵. Zakłada mówiącego – twórcę języka i rozmówcę – podejmującego próby odkrycia tej zagadki (rys. 14).

+[->-[>>+>-----<<]<--<---]>-.>>>+.>>..+++[.>]<<<<..+++..-----.<<-.>>>>+.

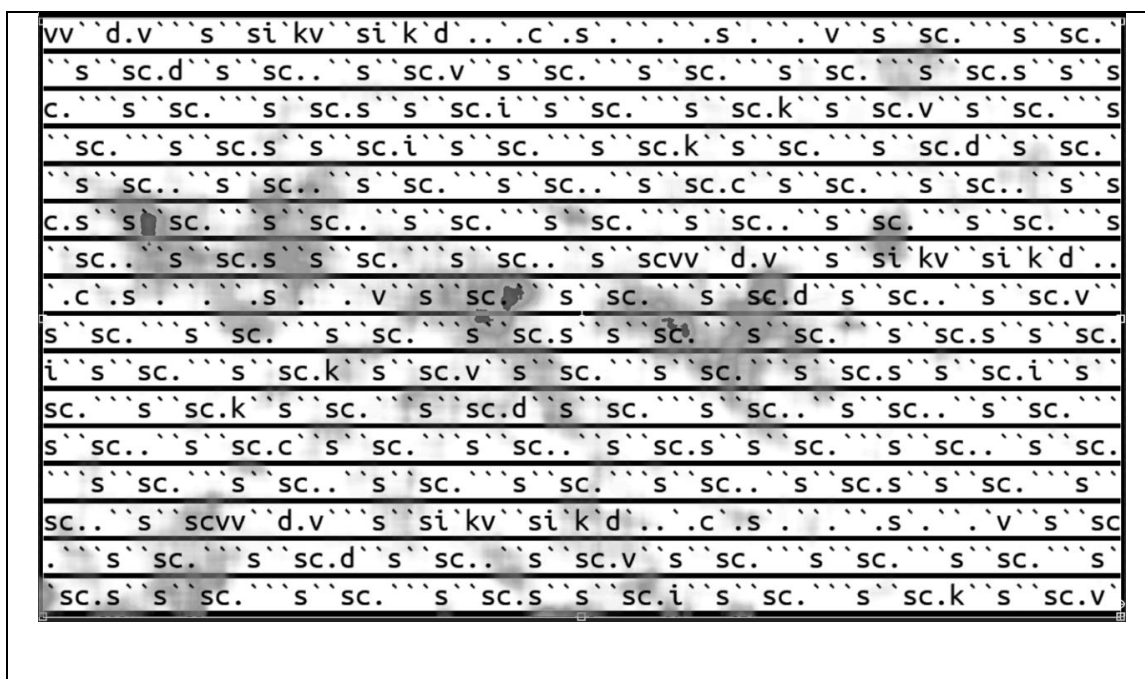
Rys. 14. Program „Hello,World!” zapisany w „zagadkowym” języku ezoterycznym brainfuck⁷²⁶

Twórczość poetycka w językach ezoterycznych różni się o tej już omawianej w podrozdziale 2.1. Nie powstaje bowiem w językach powszechnie znanych większości programistów. Jest więc mało komunikatywna na poziomie treści, której programiści, nie znając danego języka, mogą nie zrozumieć, a jedynie rozpoznać strukturę. Można by doszukiwać się u artystów tworzących w *esolangach* potrzeby czystej ekspresji. Pod względem podejścia do formy i treści utwory te przypominają abstrakcje w sztukach plastycznych (rys. 15).

724 Tłum. własne “I think what’s most interesting about esolangs is the conversation between languages, which ask questions, and programs written in those languages, which answer the questions. When you build a new esoteric language with a weird set of constraints, you get people thinking: I wonder if I can do X in this language? I wonder if there’s a way to do Y? And figuring that out, by writing programs that do X and Y, can be a fun challenge” za D. Temkin, *Language Without ...*, s. 83, <https://journals.ucp.pt/index.php/jsta/article/view/7297> (dostęp dnia 07.09.2023).

725 P. Ricœur, *Drogi rozpoznania...*, s. 84.

726 <https://codegolf.stackexchange.com/questions/55422/hello-world/163590#163590> (dostęp 07.09.2023).



Rys. 15. Fragment kodu w języku Unlambda autorstwa Davida Madore⁷²⁷

Sztuka, według powołującego się na Goethego Cassirera, jako forma symboliczna posiada:

„funkcję dzielenia i ponownego łączenia (...) »Nic pewniej nie potrafi oddzielić od świata niż sztuka i nic tak pewnie nas ze światem związać jak ona«”⁷²⁸.

Według filozofa działania artystyczne pozwalają najpełniej odnaleźć siebie i wyrazić w twórczości⁷²⁹. Sztuka daje bowiem możliwość najbardziej zindywidualizowanego uzewnętrznienia uczuć. Tak więc, w programistycznych utworach artystycznych, język również staje się tym „strumieniem przekazu”, o którym pisał Cassirer. Artysta, który tworzy w kodzie źródłowym chce także „wszystko co w nim żyje, obudzić do życia w innych”⁷³⁰. Jednocześnie towarzyszy mu świadomość ograniczeń polegających na zniekształceniu podczas przekazu. Marburczyk nazywał je brakiem pełnego pokrycia tego, co wewnętrzne w tym, co zostaje przekazane na zewnątrz.

Twórca prac w języku *Unlabdda* i zarazem kreator języków, David Madore, opisując powody, dla których tworzy esolangi, podkreślał ich funkcję testową, bycie

⁷²⁷ <https://esoteric.codes/blog/david-madore> (dostęp dnia 07.09.2023).

⁷²⁸ E. Cassirer, *Logika nauk ...*, s. 77-78.

⁷²⁹ *Ibidem*, s. 78.

⁷³⁰ *Ibidem*.

rodzajem zabawy, ale również ilustracji⁷³¹. W kwestii ilustrowania, warto przywołać słowa Ricœura, który podkreślał niemożliwość komunikowania żywego doświadczenia „jako doznanego przeze mnie, jako przeżytego, pozostającego czymś prywatnym”⁷³². Dopiero gdy doświadczenie zostaje wysłowione i nadany zostaje mu sens, język występuje jako dyskurs. Egzystencjalny warunek dyskursu stanowi bowiem kontakt w dialogu z drugim człowiekiem⁷³³, bez względu czy będzie to „tylko” czy „aż” zabawa.

Podnoszony przez Cassirera problem braku pełnego pokrycia przeżyć w treściach uzewnętrznianych, czy też jak to ujął Ricœur, niemożliwości komunikowania żywego doświadczenia – stanowi ograniczenie dla ekspresji językowej również u programistów. Brak naturalności języków programowania oraz przestrzeń, w której dochodzi do ekspresji mogą być jednak postrzegane jako najistotniejsze bariery dla swobodnej artykulacji indywidualności programistów. Wyrażanie siebie w języku nie jest zatem częste w tekście kodu źródłowego. Da się jednak je rozpoznać w dokonywanych przez programistów prywatnych wyborach dotyczących stylu programistycznego czy słownictwa. Z uwagi na specyfikę języków programowania, mają oni szanse zaledwie w ograniczonym zakresie artykułować swój sposób widzenia świata, natomiast już w sztuce kreowanej w kodzie źródłowym programów komputerowych, odbywa się to bardziej bezpośrednio. Utwory stworzone w językach programistycznych pozwalają bowiem ujawnić się podmiotowi jednoznacznie jako twórcy dzieła.

Proces polegający na tworzeniu nowych języków sztucznych, pisaniu „w nich” programów, inwencji w zakresie wyboru słownictwa czy stylu, można uznać za sposób na artykulację „ja” w języku. Ujęcie języka jako dyskursu lub formy symbolicznej pozwala przedstawić to, jak w kodzie źródłowym może wyrażać się indywidualność. Pojawiają się jednak wątpliwości, czy jest to podobny proces jak w języku naturalnym. Warto zastanowić się w tym kontekście, które struktury mózgu biorą udział w tym zadaniu.

731 “it can be an interesting test case, toy example or illustration”, wywiad z Davidem Madore <https://esoteric.codes/blog/david-madore> (dostęp dnia 07.09.2023).

732 P. Ricœur, *Język jako dyskurs* ..., s. 85.

733 *Ibidem*, s. 84.

Rozpoznawanie i produkcję mowy zwykle się wiązać z powiększonymi w lewej półkuli mózgu *planum temporale*, *pars opercularis* i *pars triangularis*⁷³⁴. W badaniach metodą fMRI zauważono, że podczas programowania wzrasta aktywność zarówno w częściach odpowiedzialnych za użycie języka jak i za logiczne rozumowanie⁷³⁵. Mózg podczas używania języków programowania nie funkcjonuje identycznie jak podczas używania języków naturalnych – można tu jedynie dostrzec podobieństwa⁷³⁶. Wyniki badań mózgu w trakcie używania języków sztucznych są niejednoznaczne. Na poziomie biologicznym, posługiwania się językiem naturalnym i sztucznym nie uznamy za tożsame. W związku z tym jednak jakie znaczenie nadają językom sztucznym i twórczości w tychże językach niektórzy programiści, warto rozważać możliwość artykulacji w nich. Ekspresja za pomocą języków programowania bywa przez te osoby traktowana analogicznie do wyrażania się w języku naturalnym.

3.2. Wymiar ujawniania

W sposobie posługiwania się językiem możemy rozpoznać, jak rzeczy stają się znaczące dla jednostki. Badanie wszelkich artykulacji indywidualności, które Taylor nazywa „artykulacjami fundamentalnymi”⁷³⁷, daje szansę odpowiedzieć na pytanie, o możliwość wyrażania bycia sobą, w tym medium. Koncepcja człowieka jako poznającego poprzez symbole *animal symbolicum* wydaje się mało przydatna dla uchwycenie występujących w języku przejawów indywidualizmu. By ująć „artykulację fundamentalną,” należy bowiem zastanowić się nad tym, co pozostaje istotne dla *Ja*. Poszukuje się zatem w języku ujawniania się prawdy o tym, jaki jest dany, konkretny człowiek, nie człowiek jako taki. Dlatego też bardziej adekwatne do tych rozważań jest

734 E. Goldberg, *Kreatywność. Mózg w dobie innowacji*, Państwowe Wydawnictwo Naukowe, Warszawa 2018, s. 52.

735J. Rosen, *This is your brain on code: JHU researchers decipher neural mechanics of computer programming*, HUB Johns Hopkins University, https://hub.jhu.edu/2020/12/17/brain-activity-while-reading-code/?fbclid=IwAR1_Ri-3sNM0RpeczAkltuH9hoqucOmF0k2J1aPkfYfN7vAVK4YY9NidtJ1o (dostęp dnia 7.10.2023).

736E. Federenko, A. Ivanova, R. Dhamala, *The Language of Programming: A Cognitive Perspective*, „Science & Society”, 23, nr 7, [https://www.cell.com/trends/cognitive-sciences/fulltext/S1364-6613\(19\)30102-](https://www.cell.com/trends/cognitive-sciences/fulltext/S1364-6613(19)30102-0?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS1364661319301020%3Fshowall%3Dtrue)

[0?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS1364661319301020%3Fshowall%3Dtrue](https://www.cell.com/trends/cognitive-sciences/fulltext/S1364-6613(19)30102-0?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS1364661319301020%3Fshowall%3Dtrue) (dostęp dnia 8.10.2023).

737 Ch. Taylor, *Teoria znaczenia...*, s. 161-169.

wykorzystanie podejścia Ricœura, jako że odnosi się ono do indywidualnej egzystencji. Uprawiana przez niego hermeneutyka zasadzała się bowiem na zgłębianiu i rekonstruowaniu sensu⁷³⁸ przez jednostkę.

Cassirerowi niełatwo przychodziło „odpowiedzieć na pytanie, co w istocie oznaczać miała »indywidualna forma« językowa”⁷³⁹. Według filozofa mowa wypełniała raczej zadania logiczne i społeczne, natomiast „indywidualnością” określał specyficzne warunki wspólnoty, nie jednostki⁷⁴⁰. Przedmiotem jego zainteresowania była bowiem uniwersalna forma i jak poszczególne języki odpowiadają celom kultury. Formy języków miały oczywiście „jasno i w odpowiedni sposób wyrazić uczucia i myśli ludzkie”⁷⁴¹, lecz Cassirer nie zajmował się kwestią artykulacji indywidualności w języku. Wewnętrzna forma miała według niego rzutować na sposób odbioru świata. Stanowisko Marburczyka w kwestii wpływu na percepcję wybranej formy językowej, było krytykowane już od połowy XX wieku⁷⁴². Przekonanie filozofa o zdeterminowaniu przez język różnic w ludzkiej percepcji⁷⁴³ pociągałoby za sobą, na przykład, uznanie programistów za osoby, które w ramach swojej działalności zawodowej inaczej interpretują dane zmysłowe.

W przeciwieństwie do refleksji Cassirera, myśl o ujawnianiu „ja” w języku pojawiała się na różnych etapach w twórczości Ricœura. Problem dał o sobie znać już w pierwszej zaproponowanej przez filozofa hermeneutyce, w której to symbol „dawał do myślenia”. Według autora „Symboliki zła” człowiek „poprawia swoje położenie w bycie”⁷⁴⁴ dzięki przemianie świadomości refleksyjnej. Już w tym okresie twórczości, odwołując się do „przemawiającego symbolu”, filozof wskazał na istotność struktur egzystencjalnych w poznaniu, gdyż: „przy wychodzeniu od symboli zadanie, więc polega na opracowaniu pojęć egzystencjalnych, to jest nie tylko refleksyjnych struktur, lecz także struktur egzystencjalnych”⁷⁴⁵. Posługując się tak pojętym symbolem odnosimy się zatem do egzystencji człowieka. W ujęciu filozofii form symbolicznych

738 S. Cichowicz, *Filozofia i hermeneutyka* ..., s. 21.

739 E. Cassirer, *Esej o człowieku*, Czytelnik, Warszawa 1998, s. 215.

740 *Ibidem*, s. 218.

741 *Ibidem*, s. 219.

742 E. Lennenberg, *A Note on Cassirer's Philosophy of Language*, "Philosophy and Phenomenological Research", 15, nr 4, 1955, s. 512-522, <https://www.jstor.org/stable/2103911> (dostęp dnia 2.10.2023).

743 G. Steiner, *Po wieży* ..., s. 134.

744 P. Ricœur, *Symbolika zła*..., s. 336.

745 *Ibidem*, s. 337.

natomiast, pojęcie symbolu dotyczy „apriorycznego bytu myślowego”⁷⁴⁶. Przejawia się on wprawdzie w jednostkowym bycie, ale nie jest wyrazem indywidualizmu, w rozumieniu bycia niepowtarzalną jednostką.

Abstrakcja miała uwalniać według Cassirera od „prymitywnych przeżyć ekspresyjnych”⁷⁴⁷. Indywidualistyczna symboliczna aktywność zmobilizowana przez „ducha ludzkiego”⁷⁴⁸ stanowiła tylko bodziec do tworzenia w ramie formy symbolicznej. Działalność „ludzkiego ducha” można by uznać za wyraz jednostkowości, lecz filozof nie poświęca jej uwagi ponad to, że jej celem jest tworzenie treści z pomocą apriorycznej struktury⁷⁴⁹. Chociaż w języku obiektywizowała się według marburczyka myśl zrodzona z woli i mocy twórczej człowieka⁷⁵⁰, to jednak nacisk w jego rozważaniach nie był położony na wyróżnienie indywidualności, lecz badanie struktury poznania ludzkiego. Jeśliby zatem poszukiwać w języku kodu źródłowego przejawów niepowtarzalności, wskazane jest posłużyć się raczej rozumieniem symbolu zaproponowanym przez francuskiego myśliciela niż abstrakcją Cassirerowskiej formy.

Transcendentalna dedukcja symbolu⁷⁵¹, którą zaproponował Ricœur, wychodzi od doświadczenia jednostki. Jak zauważa Józef Tischner w kontekście hermeneutyki symbolu, „doświadczenia zmysłowe stają się punktem wyjścia do uchwycenia nowych, często radykalnie innych wymiarów rzeczywistości” i dopiero mowa symbolu poszerza znaczenie słów o nowy wymiar⁷⁵². „Dojście do siebie” prowadzi bowiem według Ricœura „okrężną drogą”⁷⁵³. Dopiero w interpretacji staje się możliwe dotarcie do wtórnej niewinności wiary, skoro obecnie bezpośredniość jest już nieosiągalna. Refleksja nad symbolem stanowi drogę do stawania się człowieka poprzez zmianę jego świadomości. Rozważając problematykę symboliki występującej w kodzie źródłowym,

746 A. Musioł, *Neokantyzm epistemologiczny* ..., s. 8.

747 P. Malata, *Podmiot w filozofii kultury od Ernsta Cassirera do Paula Ricœura*. *Antropologia symbolu*, nieopublikowana praca doktorska, Kraków 2017, s. 32.

https://ruj.uj.edu.pl/xmlui/bitstream/handle/item/73396/malata_podmiot_w_filozofii_kultury_od_ernsta_cassirera_2017.pdf?sequence=1&isAllowed=y (dostęp dnia 23.11.2023).

748 *Ibidem*, s. 31.

749 A. Gemel, *Językowy model poznania. Kognitywne komponenty w kontynentalnej filozofii języka*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2015, s. 174.

750 E. Cassirer, *Logika nauk* ..., s. 143.

751 P. Ricœur, *Symbolika zła*..., s. 337.

752 J. Tischner, *Hermeneutyka i język*, „Analecta Cracoviensia”, 5, 1973, s. 367.

753 P. Ricœur, *O interpretacji. Esej* ..., s. 35.

celem nie jest oczywiście przedstawienie sfery sakralnej. Jak jednak zostanie zaprezentowane, nawet w takim tekście ukazać się może symbolika kulturowa.

Ricœur w swoim podejściu do natury symbolu reprezentował stanowisko antynominalistyczne⁷⁵⁴. Pogląd ten, słabszy od platonizmu, zakłada istnienie pewnych abstrakcyjnych przedmiotów, ale w przeciwieństwie do niego będących zależnymi od myślącego agenta⁷⁵⁵. Symbole nie miały zatem stanowić oznaczenia dla dowolnych zdań, jak na przykład symbole logiczne, lecz były według francuskiego filozofa „aż symbolami”⁷⁵⁶. Poszukując ich w kodzie źródłowym, możemy je odnaleźć wśród leksemów. Podobnie jak języki naturalne również języki programowania mają swoją gramatykę. Odpowiednikiem zdania jest w nich instrukcja, wyrazu leksem⁷⁵⁷. Ten ostatni zaś składa się z ciągu znaków, analogicznie jak słowo z liter. Najniższy poziom stanowią zatem znaki, lecz podstawową składową rozpatrywaną przez programistę – gdyż ma zdefiniowane znaczenie w języku – stanowi leksem.

Skoro „empiryczną podstawę symboli stanowi materiał doświadczeniowy”⁷⁵⁸, warto rozpocząć od leksemów symbolizujących obiekty lub zjawiska ze świata rzeczywistego. Są nimi, na przykład: *package* (pol. paczka) w *JavaScript*⁷⁵⁹, *sealed* (pol. zapieczętowany) w *Kotlin*, funkcja *frozenset()* (pol. zamrożony) w *Python*’ie⁷⁶⁰, *stream* (pol. strumień), *yield* (pol. krzyczeć) w *JavaScript*, *volatile* (pol. lotny) w *C*, czy *fork* (pol. rozwidlenie), funkcja systemowa w *Unixie* powodująca rozwidlenie. Używane w powłokach polecenie *echo* (lub jako konstruktor w języku *PHP*), które ma za zadanie wypisanie tekstu na konsoli, nawiązuje zapewne do nazwy odbicia akustycznego. Podobny dynamiczny rys mają leksemy *try* (pol. próbuj), *throw* (pol. rzuć) i *catch* (pol. złap), które w języku *C++* obsługują wyjątki (rys. 16). Wszystkie wspomniane leksemy powstały na podstawie nawiązania do obiektów fizycznych (strumień, paczka), czynności (rozwidlanie się, rzucanie, łapanie, krzyczenie) czy zjawisk fizycznych (fala akustyczna, zamrażanie, ulotność). Podsumujmy obecność

754 *Ibidem*, s. 367.

755 K. Jaworski, *Czy liczby istnieją? Rekonstrukcja sporu o istnienie przedmiotów matematycznych*, „Humanities and Social Sciences”, 23, nr 25, 2018, przypis dolny, s. 104.
<https://doi.prz.edu.pl/pdf/einh/408> (dostęp dnia 23.11.2023).

756 J. Tischner, *Hermeneutyka i język*, ... s. 367.

757 P. Van Roy, S. Haridi, *Programowanie. Koncepty, techniki i modele*, Helion, Gliwice 2005, s. 53.

758 J. Tischner, *Hermeneutyka i język*, ...s. 369.

759 <https://github.com/e3b0c442/keywords> (dostęp dnia 2.11.2023).

760 <https://www.programiz.com/python-programming/methods/built-in/frozenset> (dostęp dnia 1.11.2023).

tych leksemów słowami Ricœura: „tak oto w przeciwieństwie do znaków przejrzystych, które mówią tylko to, co znaczą same ustalając swe znaczenie, znaki symboliczne są nieprzejrzyste, ponieważ ich sens dosłowny, jawny wskazuje nam w trybie analogicznym sens drugi, dany wyłącznie w jego obrębie”⁷⁶¹.

```
try {
    // Block of code to try
    throw exception; // Throw an exception when a problem arise
}
catch () {
    // Block of code to handle errors
}
```

Rys. 16. Przykład użycia *try*, *throw*, *catch* w kodzie C++ ⁷⁶²

Podane powyżej przykłady symboli czerpią ze świata natury poprzez skojarzenia, z istotnymi cechami występujących tam obiektów i zjawisk. Łatwiej jest nauczyć się nowych rzeczy, jeśli te są związane ze światem przeżyć. Języka naturalnego, przykładowo, dziecko nie uczy się na pamięć, rozwija tę umiejętność w społecznych sytuacjach⁷⁶³. Zapewne dlatego, typ nauczania integracyjnego i opartego na przeżyciach, staje się obecnie coraz bardziej popularny w edukacji⁷⁶⁴. Interakcyjne, złożone doświadczenie poprzez, na przykład, zaangażowanie zmysłów, jest w tym kontekście bardzo istotne. Z tej perspektywy przyswajanie języka sztucznego może wydawać się trudniejsze. Im jednak bardziej związany ze światem rzeczywistym język programowania, tym uznawany jest za łatwiejszy. Tyczy się to, na przykład, języków wysokopoziomowych.

Według Ricœura pojęcie symbolu wiąże się z „rezygnacją z języka deskryptywnego, spekulatywnego na rzecz języka ekspresyjnego”⁷⁶⁵. Symboliczne słownictwo pojawiające się w kodzie źródłowym wyrażać może, poza wskazywaniem, ludzką potrzebę zaznaczenia poprzez nie obszarów istotnych dla człowieka. Do słów będących symbolami kulturowymi można by zaliczyć chociażby występująca w C++ funkcję *friend* (pol. przyjaciel), która daje, powiedziałoby się, po przyjacielsku dostęp

761 P. Ricœur, *Symbol daje do myślenia*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 62.

762 https://www.w3schools.com/cpp/cpp_exceptions.asp (dostęp dnia 2.11.2023).

763 K. W. Vopel, *Warsztaty – skuteczna ...*, s. 24.

764 *Ibidem*.

765 M. Bała, *O możliwości hermeneutycznej filozofii religii. Propozycja Paula Ricœura*, Wydawnictwo Uniwersytetu Kardynała Stefana Wyszyńskiego, Warszawa, 2007, s. 85.

do prywatnych składników klasy. Tzw. funkcja zaprzyjaźniona – jako symbol – komunikuje życzliwość. Instrukcja warunkowa *guard* (pol. strażnik) w języku *Swift* ma podobnie ludzki wymiar. Ochrania ona ciało metody, przekierowując przepływ kontroli, jeśli wymagany warunek nie jest spełniony⁷⁶⁶. Symbol *self* (pol. siebie) w języku *Swift*⁷⁶⁷ natomiast, chociaż brzmi jak konstrukt z postfreudowskiej teorii relacji z obiektem, oznacza aktualny obiekt. Wspomniane leksemy nawiązują niewątpliwie do świata inter- i może też intrapersonalnego. Są nośnikami sensu, ukazującymi w symbolu istotność relacji dla człowieka, czy to relacji do innych czy siebie. Relacyjność stanowi bowiem podstawową ludzką potrzebę.

Nośnikiem sensu w kodzie źródłowym, podobnym do leksemów, jest chociażby nazwa niektórych języków programowania. Jeffa Bezanson, twórca języka *Julia*, wybrał to Szekspirowskie imię kobiece, jak sam stwierdził, „bo po prostu wydawało się mu ładne”⁷⁶⁸. W bardziej przemyślany sposób powstała nazwa dystrybucji *Linuxa*, *Debian*. Założyciel projektu Ian Murdock połączył w niej imię swoje i jego ówczesnej dziewczyny Debry Lynn⁷⁶⁹. Jeszcze bardziej pomysłowi okazali się twórcy języka *Luna*. Projekt ten był zmodyfikowaną wersją języka *Sol* (pol. Słońce). Autorzy *Luny*, postanowili więc użyć do nazwania swojego języka portugalskiego słowa oznaczającego Księżyc⁷⁷⁰. Twórcy języków programowania mają możliwość wykazać się bardziej indywidualnym podejściem przy tworzeniu nazewnictwa projektów. Nie muszą one tak jak leksemy być powszechnie rozumiane.

Wartym zbadania symbolem jest *Unix*’owa komenda *delta*. Użycie jej zmienia plik, gdyż samo wyrażenie *delta* nawiązuje do matematycznej *delta* x ⁷⁷¹. Powstanie leksemu stanowi odniesienie do trygonometrii, gdzie *delta* x oznacza zmianę wyniku wyrażenia w odniesieniu do zmiennej lub zmiennych. Zatem zamiast, na przykład, użycia słowa *change* (pol. zmiana) zdecydowano się na stworzenie leksemu-symbolu nawiązującego do kontekstu kulturowego bliskiego programistom, czyli matematycznego. Nie jest to bowiem proste przełożenie na język programistyczny

766 <https://www.geeksforgeeks.org/swift-guard-statement/> (dostęp dnia 2.11.2023).

767 <https://www.appypie.com/self-swift-how-to/> (dostęp dnia 2.11.2023).

768 <https://www.reddit.com/media?url=https%3A%2F%2Fi.redd.it%2Fthe-origins-of-computer-language-names-v0-ea940mopl6fa1.png%3Fs%3D59a9c28c3a4e1b7b5bd0cddf00a3fb9db224b964> (dostęp dnia 5.10.2023).

769 R. Nixon, *Ubuntu: Up and Running: A Power User's Desktop Guide*, O'Reilly, Sebastopol 2010, s. 3 (dostęp dnia 7.11.2023).

770 <https://www.lua.org/history.html> (dostęp dnia 7.11.2023).

771 <https://www.unix.com/unix-for-dummies-questions-and-answers/13489-help-what-meaning-deltas.html> (dostęp dnia 7.11.2023).

symboli matematycznych, jak na przykład *, czy +. Sam symbol delta (Δ) nie oznacza zmiany. Jedynie nawiązując do rozumienia kontekstu matematycznego, oczekiwanego od programisty i poprzez odniesienie kulturowe, miała szansę powstać komenda *delta* w *Unixie*.

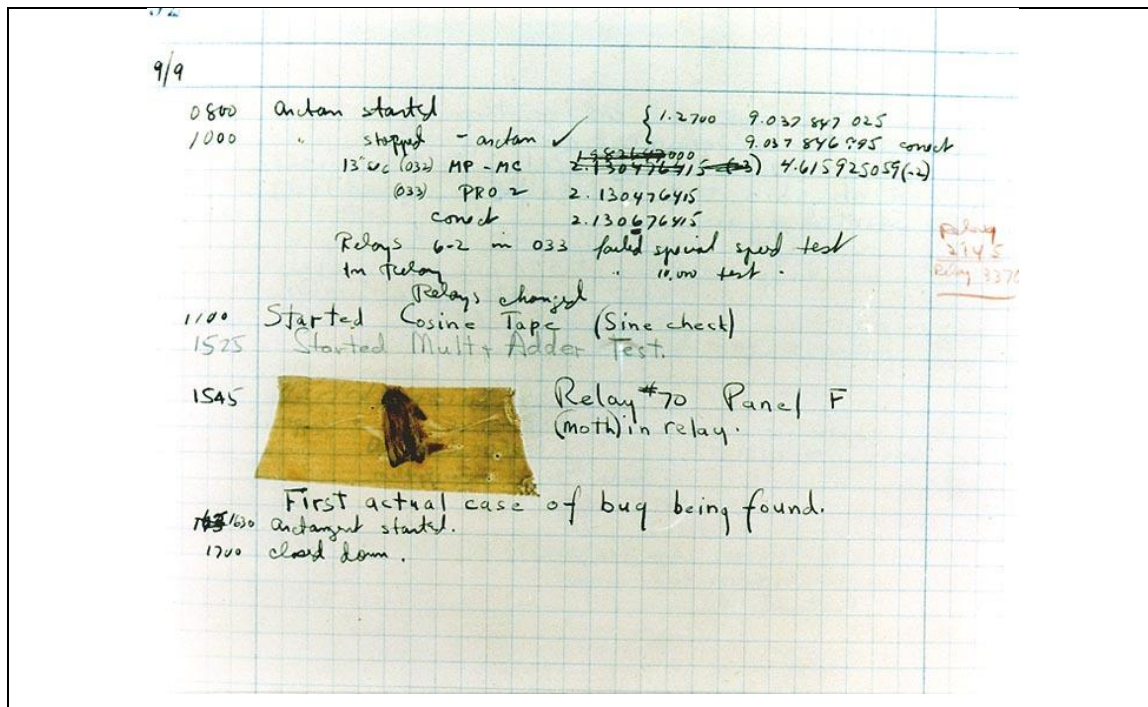
Czytelne jako symbole kulturowe są wyrazy nawiązujące do symboliki zła. Komenda powłoki *Unixa killall* (pol. zabić wszystko) powoduje, na przykład unicestwienie wszystkich procesów o wskazanej w jego argumencie nazwie. Wyrażona w słowie *kill* symbolika śmierci metaforycznie ukazuje wpływ na zachodzące procesy. Podobnie działa funkcja *kill()*, która ma mniejszy zasięg. W tekście kodu pojawiają się również takie leksemy jak *unsafe* (pol. niebezpieczny) w języku *C*, *lazy* (pol. leniwy) w języku *Swift*. Do symboliki zła, a nawet złowrogich mocy nawiązuje nazewnictwo programów *daemon*⁷⁷² (pol. demon) będących programami usługowymi w *Unixie*, czy też terminologia opisująca złośliwe oprogramowanie: *virus* (pol. wirus), *worm* (pol. robak), *bug* (pol. robak, pluskwa) (rys. 17). Nie dziwi pojawienie się tak licznej symboliki związane ze złem, skoro, „wpisane jest (zło, *przyp. aut.*) w serce ludzkiego podmiotu (...) w serce owej wysoce złożonej i z wyboru historycznej rzeczywistości, jaką jest podmiot ludzki”⁷⁷³. Ricœur w swojej filozofii, jak zauważył teolog Pierre Gisel, zachęcał poprzez refleksję właśnie nad złem do samopogłębiania rozumienia, ustanawiając podmiotowość poprzez odzyskiwanie siebie, nie zaś samostanowienie⁷⁷⁴. Zło, stanowi wyzwanie dla jednostki, wymagając od niej myślenia, działania i czucia⁷⁷⁵.

772 Działają one w tle i nie wchodzą w bezpośrednią interakcję z użytkownikiem, wbrew „demonicznej” nazwie nie mają szkodliwego działania.

773 P. Gisel, *Wstęp*, [w:] *Zło. Wyzwanie rzucone filozofii i teologii*, P. Ricœur, Instytut Wydawniczy PAX, Warszawa 1992, s. 9.

774 *Ibidem*, s. 10.

775 P. Ricœur, *Zło. Wyzwanie...*, s. 34.



Rys. 17. Słynny *bug* (pol. robal), którego znalezienie w log book'u przypisuje się Grace Hopper – popularyzatorce terminu *bug* i *debugging*⁷⁷⁶

Symbole w rozumieniu Ricœura zawierają w sobie też emocjonalny ładunek. Z psychologicznej perspektywy emocjonalny język, pomaga nakierować naszą uwagę na istotne sprawy⁷⁷⁷. Według filozofa jednak ten „ton emocjonalny domaga się słownej obiektywizacji”, gdyż mowa „rozświeśla wzruszenie”⁷⁷⁸. Poprzez język możliwe jest „wyznanie świadomości”⁷⁷⁹. Za symbol o dużym ładunku emocjonalnym można uznać, na przykład leksem *Promise()* (pol. obietnica) w języku *JavaScript*. Obiektu używa się jako sposobu na *callback hell* (pol. oddzwonienie z piekła), czyli rozwiązanie niekorzystnej sytuacji wielokrotnego zagnieżdżenia funkcji wywołania zwrotnego⁷⁸⁰. Telefon z piekła nie brzmi pozytywnie, na szczęście, aby rozwiązać problem *callback hell*, można wykorzystać zwracanie obiektu *Promise* przez funkcję asynchroniczną. Godzi się powiedzieć, że obietnica-*promise*, w tym kontekście, daje nadzieję. Podobnie za przepełnioną emocjami można uznać symbolikę początku i końca. Na przykład w *Pascalu* rozpoczęciu i zakończeniu tekstu kodu towarzyszą leksemy *begin* i *end*. W składni innych języków nie jest wymagane takie podkreślenie początku i końca,

776 <https://copyprogramming.com/howto/what-is-a-bug-in-computer-programming> (dostęp dnia 18.10.2023); https://americanhistory.si.edu/collections/nmah_334663 (dostęp dnia 18.10.2023).

777 P. Zimbardo, R. Johnson, V. McCann, *Psychologia. Kluczowe koncepcje. Motywacja i uczenie się*, Państwowe Wydawnictwo Naukowe, Warszawa 2010, s. 30.

778 P. Ricœur, *Symbolika zła...*, s. 10.

779 *Ibidem*.

780 <https://www.nafrontendzie.pl/jak-pozbyc-sie-callback-hell> (dostęp dnia 17.10.2023).

wystarcza jedynie użycie znaku }.

Badając relację między słowem a jego znaczeniem w obszarze języków programowania, warto poruszyć też problem tzw. błędów semantycznych (*semantic error*). Występują one wtedy, gdy kod jest poprawny składniowo, czyli „gramatycznie”, lecz nie działa, tak jak oczekiwалby jego twórca. Stanowią one rezultat braku zrozumienia przez programistę semantyki danego języka przejawiającym się błędnym użyciem wyrazu. Znaczenie wyrazu w takiej sytuacji jest inne niż zmierzone przez piszącego. Dlatego też konieczne wydaje się odróżnienie statycznego od dynamicznego błędu semantycznego (*static* i *dynamic semantic error*)⁷⁸¹. Pierwszy rodzaj dotyczy niespełnienia wymagań języka tzw. „błędów o niepoprawnej wartości dla typu” (*invalid types*) i jest ujawniany w trakcie kompilacji. Drugi, bardziej nas interesujący, nie jest wykrywany przez kompilator i stanowi błąd wynikający z niewłaściwego rozumienia znaczenia. W tej sytuacji program działa, ale wykonuje inną niż planowana przez człowieka czynność. Jak widać treść programu odbiega tu zatem od zamierzonego rozumowania. Przykład dynamicznych błędów semantycznych ukazuje jak rozumienie przez człowieka znaczenia symbolu, jest niezbędne podczas tworzenia oprogramowania.

Interesujące, w kontekście znaczenia symbolu, są występujące w języku PHP tak zwane magiczne stałe (*magic constants*)⁷⁸². Wartość wspomnianych stałych predefiniowanych zależy od miejsca ich użycia w kodzie. Na przykład leksem `_LINE_` zwraca bieżącą linię. Wartość *magic constants* zmienia się zatem w zależności od kontekstu⁷⁸³. Niezwykle ciekawym przykładem wykorzystania kontekstu w językach programowania są też języki ze składnią kontekstową. Przykładem jest projekt językowy *PL/I (Programming Language One)*⁷⁸⁴, w którym słowa kluczowe mogą w zależności od kontekstu mieć inne znaczenie, *keywords* nie są bowiem w nim zastrzeżone⁷⁸⁵. Stanowi to sytuację niezwykle rzadką, ale wartą odnotowania, gdyż wiąże się z rozumieniem języka jako dyskursu w ujęciu Ricœura. Ten aktualizuje się

781 <https://medium.com/thefreshwrites/java-semantic-errors-why-java-programmer-must-acknowledge-semantic-errors-6f3e63b411e6> (dostęp dnia 18.10.2023).

782 https://www.w3schools.com/php/php_magic_constants.asp (dostęp dnia 18.10.2023).

783 M. Lis, *Leksykon kieszonkowy PHP*, Helion, Gliwice 2012, s. 45.

<https://informatyka.2ap.pl/wp-content/uploads/2014/10/Marcin-Lis-PHP-5.-Leksykon-kieszonkowy.pdf> (dostęp dnia 18.10.2023).

784 L. Irwin, *Implementing Phrase-Structure Productions in PL/I*, „Communications of the ACM”, nr 10, 1967, s. 424.

785 <https://stackoverflow.com/questions/26377386/antlr-and-pl-i-grammar> (dostęp dnia 18.11.2023).

bowiem, według niego, w zdarzeniu. Jest bowiem posiadającym znaczenie „zdarzeniem językowym”⁷⁸⁶. W formie pisanej filozof określił dyskurs mianem „użycia językowego”⁷⁸⁷, w którym dokumentuje się ludzkie działanie. Istotą magicznych stałych czy języków kontekstowych jest właśnie „użycie językowe”, charakterystyczne dla ujmowania języka jako *parole*, nie zaś *langue*-systemu. Oczywiście nadal wspomniane stałe pozostają symbolami, lecz w charakteryzującej je możliwości zmiany znaczenia, lecz widoczne staje się rozumienie ich również przez pryzmat *użycia językowego*.

W następującym po etapie hermeneutyki symbolu – hermeneutyce dyskursu, Ricœur odszedł od egzystencjalnego doświadczenia w symbolu, skupiając się na możliwościach języka. „Ponieważ nie mówi, lecz jest mówiony”⁷⁸⁸, język-kod rozumiany jest jedynie jako narzędzie, natomiast język jako dyskurs niesie ze sobą sens. Poprzez dyskurs wybrzmiewa oczywiście nastawienie społeczne, ale też „wypowiada się »Ja«”⁷⁸⁹. Jak zauważyła Elżbieta Wolicka rozumienie języka, jako wydarzenia komunikacyjnego wiąże się z otwartością na zarówno rzeczywistość świata, jak i jego użytkowników⁷⁹⁰. Wydarzenie komunikacyjne ma wymiar egzystencjalny, gdyż sens prywatnego doświadczenia jednostki zostaje przekazany innym, przewyżając w ten sposób „radikalną niekomunikowalność żywego doświadczenia”⁷⁹¹.

Użycie języka programowania aby wyrazić i rozwiązać jakiś problem stanowi wydarzenie komunikacyjne, w którym programista wykorzystuje swoją wiedzę domenową. Niektórzy nazywają nawet programowanie „mapowaniem wiedzy domenowej na kod”⁷⁹². Znajomość obszaru, dla którego powstaje oprogramowanie, umożliwia programiście stworzenie optymalnej struktury kodu pasującej do zadań mających być wykonanymi przez program. Ricœurowski dyskurs aktualizuje się w

786 P. Ricœur, *Język jako dyskurs...*, s. 79.

787 P. Ricœur, *The model of the text: meaningful action considered as a text*, [w:] *From text to action; Essays in Hermeneutics, II*, red. R. Kearney, Northwestern University Press, Evanston 2007, s. 150; P. Ricœur, *Le Modèle du texte: l'action sensée considérée comme un texte*, [w:] *Du texte à l'action, Essais d'herméneutique*, t. 2, Editions du Seuil, Paris 1986, s. 190.

788 P. Ricœur, *Przemoc i język*, „Ethos”, 27, nr 2, 2014, s. 143.

https://cejsh.icm.edu.pl/cejsh/element/bwmeta1.element.ojs-doi-10_12887_27-2014-2-106-06/c/5511-5273.pdf

(dostęp dnia 18.11.2023).

789 *Ibidem*.

790 E. Wolicka, *Narracja i egzystencja*, Wydawnictwo KUL, Lublin 2010, s. 20.

791 P. Ricœur, *Język jako dyskurs...*, s. 85.

792 T. Królikowski, W. Susłow, *Analizy i projekt informatyzacji przedsięwzięcia szkoleniowego*, „Zeszyty Naukowe Wydziału Elektroniki i Informatyki Politechniki Koszalińskiej”, nr 14, 2019, s. 6.

zdarzeniu, czyli w tej sytuacji jest nim napisanie i odczytanie kodu źródłowego w kontekście adekwatnym do obszaru. Wiedza domenowa stanowi wiedzę programisty o środowisku, w którym będzie działał docelowy system⁷⁹³. Wymaga to między innymi pozyskania informacji od użytkowników oprogramowania, gdyż to oni na co dzień wykorzystują system, stając się ekspertami domenowymi. Bywa nawet, że „w wielu przypadkach programista musi rozumieć domenę nawet bardziej szczegółowo niż specjalista domenowy, ponieważ ten pierwszy może rozwiązywać wszelkiego rodzaju wyjątki i przypadki na podstawie doświadczenia i zdrowego rozsądku, wykorzystując intuicję, a programista powinien jawnie obsłużyć wszystkie te sytuacje w kodzie, zanim one powstaną”⁷⁹⁴. Pojawia się zatem konieczność przewidywania jak, czasem w nietypowy sposób, program może być wykorzystany. Osadza to mocno programistę w świecie i wymaga od niego komunikacji w celu poznania, jak funkcjonuje aspekt rzeczywistości.

Przyswojenie znaczenia (*appropriation, Aneignung*) w interpretacji stanowi cel Ricœurowskiej hermeneutyki⁷⁹⁵. Następuje w niej przewyciężenie dystansu, gdy „aktualizuje znaczenie tekstu dla obecnego czytelnika”⁷⁹⁶. Stąd tak istotny jest odbiorca tekstu kodu źródłowego, konieczny dla tego dialogu. Może nim być inny programista rozwijający program, ale także autor, który po czasie powraca do swojego tekstu. Kontakt ze swoim własnym wytworem może wywołać, na przykład, dezaprobatę wobec własnych umiejętności sprzed lat. Przyswojenie tekstu w interpretacji jest według francuskiego filozofa – „zdarzeniem dyskursu”. Umiejscowione jest ono bowiem w teraźniejszości, czyli w kontekście osoby czytającej. Dyskursywny wymiar mają w kodzie źródłowym również, na przykład, komentarze. Ich celem jest „pomóc czytelnikowi wiedzieć tyle ile wiedział piszący”⁷⁹⁷. Cenna jest zatem ich przejrzystość. Zamiast, na przykład, komentarza:

//Oh crap, this stuff will get tricky if there are ever duplicates in this list⁷⁹⁸,

warto jednoznacznie opisać czego dotyczy problem:

⁷⁹³ *Ibidem*.

⁷⁹⁴ *Ibidem*, s. 7.

⁷⁹⁵ P. Ricœur, *Przyswojenie*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 277.

⁷⁹⁶ *Ibidem*.

⁷⁹⁷ D. Boswell, T. Foucher, *The Art of Readable Code*, O'Reilly, Sebastopol 2011, s. 46.

⁷⁹⁸ *Ibidem*, s. 56.

//Careful: this code doesn't handle duplicates in this list⁷⁹⁹.

Nie należy komentować kwestii, które łatwo można wywnioskować z kodu. Komentarze mają wyjaśniać, gdyż poprzez nie utrzymuje się zrozumienie kodu. W zinstytucjonalizowanym kształceniu programistów kładzie się nacisk na pisanie ich, do tego stopnia, że „niektórzy programiści czują się winni zostawiając »nagą« funkcję bez komentarzy”⁸⁰⁰. Za nieosiągalny cel można uznać to, aby kod był tak przejrzysty, żeby sam się dokumentował. Zwykle konieczne jest jego uzupełnienie o tę jednostkę leksykalną. Niektórzy zalecają, aby rejestrować swoje przemyślenia, na przykład:

```
//This class is getting messy. Maybe we should create a 'ResourceNode' subclass to  
// help organize things.801
```

Pomaga to wprowadzać poprawki do kodu. Można stosować też w komentarzach markery, wskazujące czego dotyczą uwagi np. TODO (do zrobienia), FIXME (fragment kodu do naprawienia), HACK („niezbyt elegancie rozwiązanie problemu”), XXX (niebezpieczeństwo, duży problem)⁸⁰².

Komentarze w kodzie źródłowym mogą też stanowić wyraz samoświadomości osoby piszącej. Jest to szczególnie widoczne w zapisanych tamże krytycznych uwagach odnośnie własnej pracy, na przykład, tych dotyczących zmienionego przez alkohol stanu świadomości piszącego:

```
// drunk, fix later803 (tłum. pol. „pijany, naprawić później”).
```

Pojawiają się również nawiązania do wiary w Boga i wszechmocy. Programista w poniższym przykładowym komentarzu do kodu, w sposób żartobliwy, ujawnia wiedzę o swojej słabości w zestawieniu z omnipotencją boską:

```
//When I wrote this, only God and I understood what I was doing //Now, God only  
knows804 (tłum. pol. “Kiedy to pisałem(am?) tylko Bóg i ja rozumieliśmy co robię. Teraz tylko Bóg wie”).
```

⁷⁹⁹ *Ibidem*.

⁸⁰⁰ Tłum. własne, *Ibidem*, s. 48.

⁸⁰¹ *Ibidem*, s. 50.

⁸⁰² *Ibidem*.

⁸⁰³ <https://stackoverflow.com/a/185181> (dostęp dnia 6.11.2023).

⁸⁰⁴ <https://stackoverflow.com/a/316233> (dostęp dnia 6.11.2023).

Bywają również komentarze artykułujące frustrację, lecz nie wobec swoich kompetencji, ale poprzednika. Ujawnia się ona chociażby w następującym życzeniu śmierci:

```
// The main problem is the BindCompany() method,  
// which he hoped would be able to do everything. I hope he dies805.
```

(pol. “Głównym problemem jest metoda BindCompany(), która, miał nadzieję, że będzie robić wszystko. Mam nadzieję, że umrze”)

Autor powyższego komentarza jest znany w środowisku programistycznym. Dan MacKinley stworzył tekst (rys. 18) obrażający poprzedniego twórcę kodu, którego tekst sam miał za zadanie poprawić. Nie tylko używał pisząc o poprzedniku niecenzuralnych słów, na przykład *fucking idiot* (pol. pieprzony idiota), wyśmiewał styl ubierania się niesławnego Richarda w komentarzach (chustę i okulary), ale także stworzył, jak twierdził dla żartu, metodę *RichardIsAFuckingIdiotControl*. W tekście tym MacKinley zamierzał przekazać swoją dezaprobatę.

```
//Code sanitized to protect the foolish.  
.....  
/// <summary>  
/// Class used to work around Richard being a fucking idiot  
/// </summary>  
/// <remarks>  
/// The point of this is to work around his poor design so that paging will  
/// work on a mobile control. The main problem is the BindCompany() method,  
/// which he hoped would be able to do everything. I hope he dies.  
/// </remarks>  
public abstract class RichardIsAFuckingIdiotControl: MobileBaseControl, ICompanyProfileControl  
.....  
/// Make sure nobody is actually using that fucking bindcompany method  
MethodInfo m = this.GetType().GetMethod("BindCompany", BindingFlags.DeclaredOnly |  
    BindingFlags.Instance | BindingFlags.Public | BindingFlags.NonPublic);  
if (m != null)  
{  
    throw new RichardIsAFuckingIdiotException("No!! Don't use the fucking BindCompany  
method!!!!");  
}  
// P.S. this method is a joke ... the rest of the class is fucking serious  
}  
  
/// <summary>  
/// This returns true if this control is supposed to be doing anything  
/// at all for this request. Richard thought it was a good idea to load  
/// the entire website during every request and have things turn themselves  
/// off. He also thought bandanas and aviator sunglasses were "fuckin'  
/// gnarly, dude."
```

Rys. 18. Fragmenty kodu źródłowego autorstwa Dana MacKinley’a, w których autor obraża poprzedniego twórcę kodu⁸⁰⁶. Zostały wyboldowane i podkreślone fragmenty wskazujące na niechęć autora do byłego współpracownika

805 <https://mcfunley.com/from-the-annals-of-dubious-achievement> (dostęp dnia 6.11.2023).

Nawet jeśli by uznać sposób komunikacji MacKinleya za agresywny, a nie żartobliwy, to warto poświęcić uwagę stworzonemu przez niego kodowi źródłowemu. Pozostawił on bowiem po sobie ważny tekst poddający się analizie dyskursu. Wyartykułował w języku nie tylko swoją niechęć wobec byłego współpracownika, ale również ujawnił, co dla niego samego jest ważne i jaki sam jest. Priorytet stanowiła zapewne dla MacKinleya jakość oprogramowania, gdyż to jej niski poziom sprowokował jego reakcję. Kolejną kwestią, która ujawniła się była potrzeba przedstawienia siebie w tekście jako osoby inteligentnej, poprawiającej czyjeś błędy. Wyrazem lotności miał być również stosowany przez niego dowcip. Autor był ponadto przekonany o swoim lepszym guście, gdyż krytykował nawet ubiór Richarda. Trudno ocenić czy zastosowanie wulgaryzmów wynikało z porywczosci, czy było zabiegiem stylistycznym mającym na celu pozostanie w konwencji żartu.

Jasnym staje się, że tekst MacKinleya poddaje się interpretacji. Odnajdziemy w nim komunikację z czytelnikiem, z którym autor chce podzielić się swoimi żalami. W jednostkach leksykalnych, zarówno w komentarzach, jak i identyfikatorach (nazwa metody) ujawnia się jego nastawienie. Jest zatem możliwa artykulacja indywidualności programisty w ramach tekstu kodu źródłowego, nawet z wykorzystaniem języka programowania (stworzenie nazwy metody – *RichardIsAFuckingIdiotControl*). Jakkolwiek odczytalibyśmy intencję autora, czy miał to być żart, czy może przestroga, to widoczne staje się, że w tekście realizuje się dyskurs. Aktualizuje się on w sytuacji odczytania i naszej jego interpretacji. Tekst ten posiada możliwość przekazania pewnej intencji i zaprezentowania nastawienia autora do problematycznej sytuacji. W języku kodu źródłowego istnieje zatem możliwość ujawnienia się jak myśli konkretna jednostka, czyli artykulacji indywidualności.

Język poprzez artykulację ma możliwość konstytuowania świadomości. Jego twórczy aspekt uwzględnia bowiem uświadamianie sobie formułowanych myśli⁸⁰⁷, toteż człowiek ma możliwość wejść, dzięki werbalizacji, w relację z sobą. Ricœur, co prawda zawarł w hermeneutyce dyskursu myśl o potrzebie samorozumienia w języku, jednak nie wiązał jej ze samoświadomością. Jak pisał „świadomość sama jest tylko symptomem”⁸⁰⁸ – nie zaś punktem wyjścia. Samorozpoznanie odbywać się miało

806 <https://stackoverflow.com/a/184673> (dostęp dnia 6.11.2023).

807 Ch. Taylor, *Teoria znaczenia...*, s. 155.

808 P. Ricœur, *Symbol daje...*, s. 74.

bowiem nie w introspekcji⁸⁰⁹, lecz poprzez „pojmowanie siebie w obliczu tekstu”⁸¹⁰. Tenże, według myśliciela, miał oddziaływać na „ja” czytelnika, który odnajdywał w ten sposób nową możliwość bycia. Czytelnik, jak twierdził Ricœur, przyswaja bowiem w trakcie odczytania tekstu jakąś propozycję świata i dzięki temu odkrywa „siebie poszerzonego”⁸¹¹. Ta propozycja istnienia jest czymś zadaniem, co można odkryć dopiero poprzez recepcję dzieła.

W swoich rozważaniach z okresu hermeneutyki dyskursu Ricœur wiązał przemianę czytelnika bardziej z rozpoznaniem niżli poznaniem⁸¹². Klucz do niego należało szukać w recepcji świata tekstu, do którego osoba czytająca miała dostęp⁸¹³. Dzieło, w miejsce symbolu ze wcześniejszego etapu twórczości⁸¹⁴, stało się pośrednikiem to umożliwiającym. Akt interpretacji wymagał jednak wpierw zdystansowania się (*distanstation*), aby potem mogło nastąpić przyswojenie (*appropriation*) i zastosowanie (*application*)⁸¹⁵. Stosownie dystansu w interpretacji, poprzez wyjaśnianie i rozumienie w połączeniu z późniejszym przyswojeniem, umożliwiałyby metamorfozę podmiotu. Widoczne jest tu nawiązanie do mechanizmów psychicznych zaproponowanych w psychoanalizie. Ricœur zgadzał się z poglądem Freuda o istnieniu początkowej fałszywej świadomości⁸¹⁶ i wypieraniu myśli. Dostrzegał, więc funkcję „obron” jako należących do sfery dynamiki konfliktów⁸¹⁷.

W kontekście psychoanalizy rozumienie zjawisk psychicznych wymaga uwzględniania pojęcia wyparcia. Jest to o tyle istotne, że utwór w postaci napisanego kodu źródłowego, może być przecież rozważany z perspektywy teorii wywodzących się z tej orientacji jako efekt sublimacji. Twórczość będzie zatem w tym kontekście zaliczana do procesów obronnych, bądź do procesów rozwojowych jako mechanizm

809 P. Ricœur, *Wyzwanie semiologiczne: problem podmiotu*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 251.

810 P. Ricœur, *O pewnej egzystencjalnej interpretacji Freuda*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 244.

811 *Ibidem*.

812 P. Ricœur, *Przyswojenie...*, s. 280.

813 *Ibidem*, s. 273.

814 W hermeneutyce symbolu, sens drugi był dany w obrębie sensu pierwszego, gdyż jak zauważył filozof Paweł Dybel, francuski filozof nie uwzględniał zjawiska wypierania. W ramach jego podejścia intencjonalność metaforyczna, stanowiła rozwinięcie sensu dosłownego. Wszystko zaś konstituowało się we wskazaniu na sytuację człowieka w obrębie sacrum. P. Dybel, *Dialog i represja*, Wydawnictwo IFiS PAN, Warszawa 1995, s. 95-96.

815 E. Wolicka, *Narracja i egzystencja...*, s. 32.

816 P. Ricœur, *Hermeneutyka symboli a refleksja filozoficzna – II*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 120.

817 *Ibidem*, s. 234.

adaptacyjny⁸¹⁸. Koncepcję obrony jako walki *ego* ze zbyt trudnymi dla jednostki uczuciami czy też myślami, zapoczątkował Freud. Określał tym terminem „wszystkie techniki, jakie wykorzystuje ego, borykając się z konfliktami, które mogą prowadzić do nerwicy”⁸¹⁹. W dziele zatytułowanym „Kultura jako źródło cierpień” ogłosił tezę, że cała kultura jest efektem sublimacji popędu seksualnego, czyli że przemieszczenie celów popędowych następuje w zgodzie z wartościami, jakie są wyznawane w danym społeczeństwie⁸²⁰.

Według teorii psychoanalitycznej, w celu ochrony siebie, *ego*⁸²¹ wykształca mechanizmy obronne, które zniekształcają rzeczywistość. Te nawykowe nieświadome techniki wykorzystywane są przez ludzi do radzenia sobie z trudnymi emocjami, na przykład złością, lękiem, wstydem czy smutkiem⁸²². Oprócz działań takich jak przykładowo wyparcie, zaprzeczenie czy przemieszczenie – uznawanych za niedojrzałe⁸²³, można wyróżnić jednak także takie, które sprzyjają rozwojowi człowieka. Takim mechanizmem obronnym jest sublimacja, czyli „zmiana lub przemieszczenie impulsów »id« przez skierowanie energii popędowej na społecznie akceptowane zachowania; na przykład na twórczość artystyczną”⁸²⁴. Nieakceptowane popędy mogą zatem zostać wykorzystane w obszarze nieszkodliwym lub nawet użytecznym społecznie⁸²⁵, gdyż energia psychiczna podczas sublimacji zostaje użyta w sposób twórczy. Na pisanie kodu źródłowego można zatem również patrzeć jak na wykorzystanie energii psychicznej w konstruktywny sposób.

Sublimacja może przejawiać się w sztuce, religii – a co istotne dla tych rozważań nad kodem źródłowym – także w nauce⁸²⁶. Zwolennicy wiązania twórczości z

818 H. J. Grzegółowska-Klarkowska, *Mechanizmy obronne osobowości*, Państwowe Wydawnictwo Naukowe, Warszawa 1986, s. 184.

819 Za A. Freud, *Ego i mechanizmy obronne*, Państwowe Wydawnictwo Naukowe, Warszawa 1997, s. 34.

820 *Ibidem*, s. 40.

821 D. P. Schultz, S. E. Schultz, *Historia współczesnej psychologii*, Wydawnictwo Uniwersytetu Jagiellońskiego, 2008, Kraków, s. 416.

822 *Psychologia kliniczna*, L. Cierpiakowska, H. Sęk, red, Państwowe Wydawnictwo Naukowe, Warszawa 2016, s. 77.

823 G. O. Gabbard, *Psychiatria psychodynamiczna w praktyce klinicznej*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2009, s. 61.

824 D. P. Schultz, S. E. Schultz, *Historia współczesnej psychologii*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2008, s. 417.

825 N. McWilliams, *Psychoterapia psychoanalityczna*, Wydawnictwo Harmonia, Gdańsk 2011, s. 341.

826 F. Maj, *Psychoanalityczne i egzystencjalne problemy twórczości*, „Humanistyka i przyrodoznawstwo”, 15, 2009, s. 145, <https://czasopisma.uwm.edu.pl/index.php/hip/article/view/947/814> (dostęp dnia 25.10.2023).

nerwicą, jak na przykład Otto Rank, wywodzili potrzebę twórczości z niezadowolenia, z tego kim się jest i łączyli ją ze zniesieniem własnego *Ja*⁸²⁷. Odrębnie kwestię twórczości ujęła córka Freuda. Psychoanalityczka Anna Freud porządkując listę obronnych metod *ego*, zauważyła, że sublimacja związana jest ze zdrowiem, nie nerwicą⁸²⁸.

W ramach koncepcji zdrowia psychicznego jako „sprężystości psychicznej” (*resilience*), którą wykorzystał psychiatra George Vaillant, *resilience* wyrażać się miała stosowaniem dojrzałych mechanizmów obronnych. Wśród nich, oprócz sublimacji wyróżnił on: altruizm, tłumienie, antycypację i poczucie humoru⁸²⁹. Ten badacz nurtu psychologii pozytywnej stworzył zespół cech opisujących zdrowie psychiczne, proponując wykorzystanie między innymi psychoanalitycznej teorii obron. Według jego ujęcia: „nieczynienie bliźniemu co tobie nie miłe (altruizm), trzymanie fasonu z kamienną twarzą (tłumienie), zdawanie sobie sprawy z bólu (antycypacja), zdolność do nietraktowania siebie zbyt poważnie (poczucie humoru), oraz do przerabiania cytryn na lemoniadę (sublimacja) – z tego właśnie składa się pozytywne zdrowie psychiczne”⁸³⁰. Rozważając zatem sublimację, możliwe jest sięganie do jej wymiaru adaptacyjnego, uznając ją za jeden z dojrzałych mechanizmów obronnych. Przekierowanie energii psychicznej na pracę twórczą nie jest zatem powodowane zaburzeniami, lecz wiąże się z dojrzałością *ego*.

We wspomnianej psychologii pozytywnej wskazuje się też na humor jako proces psychiczny wspierający zdrowie psychiczne^{831,832}. Interwencje psychologiczne bazujące na wykorzystaniu żartu zaproponował już Victor Frankl. Był on pomysłodawcą techniki „paradoksalnej intencji”, w której wykorzystuje się terapeutycznie właśnie humor. Pacjent z objawami nerwicowymi ma za zadanie wyobrażać sobie przedmiot swojego największego lęku w humorystycznej sytuacji,

827 *Ibidem*.

828 A. Freud, *Ego i mechanizmy* ..., s. 35.

829 *Psychologia kliniczna*, red. Cierpiałkowska, L., Sęk..., s. 55.

830 G. Vaillant, *Pozytywne zdrowie psychiczne: czy istnieje definicja międzykulturowa?*, „Postępy Psychiatrii i Neurologii”, nr 21, 2012, s. 236.

https://www.ppn.ipin.edu.pl/aktualne/2013/2/02_PPIN_04-2012Z.pdf (dostęp dnia 25.10.2023).

831 *Psychologia kliniczna*, red. Cierpiałkowska, L., Sęk, H. ..., s. 670.

832 F. Gander, R. Proyer, *Strength-Based Positive Interventions: Further Evidence for Their Potential in Enhancing Well-Being and Alleviating Depression*, „Journal of Happiness Studies” 14, 2013
<https://link.springer.com/article/10.1007/s10902-012-9380-0> (dostęp dnia 16.11.2023).

ironizować⁸³³. Stosowanie żartów w kodzie źródłowym, których przykłady zostały przedstawione w podrozdziale 2.1., należą zatem do przejawów dojrzałych mechanizmów obronnych. Humor został wprowadzić już opisany wcześniej jako wyraz ekspresji pojawiający się w językowej przestrzeni publicznej, można go jednak również ująć jako mechanizm obronny. Poprzez jego użycie w języku kodu źródłowego, dostrzec możemy zatem również ujawnianie się „ja” programisty.

Język kodu źródłowego staje się miejscem ujawnień tego kim są twórcy tegoż tekstu. Ponieważ „rozumiemy siebie jedynie odbywając dookoła drogę pośród znaków ludzkości utrwalaonych w dziełach kultury”⁸³⁴, dzięki prześledzeniu jak programiści wyrażają się w ramach swojej działalności, dążymy do pogłębienia rozumienia natury ludzkiej. Hermeneutyka Ricœura może pomóc w przyjrzeniu się, jak w języku wyraża się bycie sobą, wszak „odkrywałaby taki sposób istnienia, który od początku do końca polegałby na byciu-interpretowanym”⁸³⁵. „Samozrozumienie” siebie w kontekście tekstu kodu źródłowego stanowi etap interpretacji, którego kolejną fazą jest uzyskanie na tej podstawie wykładni ukazującej ludzkie bycie-w-świecie zapośredniczone w dialogu⁸³⁶. Narracja stanowi według Ricœura strukturę rozumienia i podążając tym tropem, jego hermeneutyka dyskursu przeobraża się w hermeneutykę narracji.

Preferowanym w tej pracy nazewnictwem dla rozważanego okresu twórczości Ricœura jest „hermeneutyka dyskursu”, zamiast „hermeneutyki tekstu”. Wiążę się to z ukierunkowaniem na rozumienie w języku świata: innych ludzi (jak miało to miejsce w rozdziale drugim), siebie (co dzieje się aktualnie, w rozdziale trzecim) i etyki (która będzie rozważana w rozdziale czwartym). Filozofia ta prowadzi bowiem nie tylko ku rozumieniu tekstu, lecz również sytuacji człowieka w nim zapośredniczonego. Od tak pojętej hermeneutyki droga wiedzie, zarówno chronologicznie, jak i logicznie, do kolejnego etapu twórczości Ricœura, „hermeneutyki self” i ujęcia narracyjnego siebie. Narracja jest bowiem przykładem dyskursu⁸³⁷, który dotyczy ludzkiego działania. Twórczość programistyczna ukazana w tym rozdziale jako artykulacja indywidualności

833 *Psychologia kliniczna*, red. Cierpiałkowska, L., Sęk, H. ..., s. 161.

834 P. Ricœur, *Hermeneutyczna funkcja...*, s. 244.

835 P. Ricœur, *Konflikt hermeneutyk: epistemologia interpretacji*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 136.

836 E. Wolicka, *Narracja i egzystencja...*, s. 221.

837 R. Grzywacz, *Dyskurs przekonań podmiotu a tożsamość narracyjnej jednostki i wspólnoty w filozofii Paula Ricœura. Autoreferat z rozprawy doktorskiej*, „Rocznik Filozoficzny Ignatianum”, 21, nr 1, 2016, s. 88.

i wyraz potrzeb jednostki, w kolejnym przedstawiona zostanie, jako element konstytuujący tożsamość.

3.3. Wymiar konstytuowania

Potrzeba ujawniania „ja” w języku wiąże się z kolejnym wymiarem, czyli konstytuowaniem⁸³⁸ siebie. Stwarzanie takie, według Taylora, nadaje człowiekowi „moralną i duchową orientację”⁸³⁹. Koncepcja *animal symbolicum* w ujęciu Cassirera oraz Ricœurowskiej tożsamości narracyjnej odwołują się właśnie do języka, aby ukazać to kim jest człowiek. Uwzględnienie obu podejść poszerza przestrzeń filozoficznej refleksji podkreślając istotność tego co prezentuje się w języku jako indywidualne. Obaj filozofowie żywili przekonanie, że poprzez działalność twórczą można uchwycić istotę jego autora. Programiści w tym podrozdziale zostaną zatem przedstawieni jako twórcy kodu źródłowego. W przeciwieństwie do wcześniejszych przemyśleń, w których rozważano ujawnianie się indywidualności odnosząc tę kwestię do sfery psychicznej, niniejsze dociekania skupią się na konstytuowaniu się jednostki ludzkiej poprzez jej działalność – tu programowanie.

Opracowana przez Cassirera koncepcja jednostki ludzkiej wiąże się z tworzeniem przez nią własnego symbolicznego obrazu świata, w którym syntezuje ona swoje doświadczenia⁸⁴⁰. Symbolizm nie ujawnia się bowiem bezpośrednio w rezultatach aktywności człowieka:

„Jeśli zadowolimy się rozważaniem rezultatów tych przedsięwzięć — twórców myśli mitycznej, religijnych obrzędów i wierzeń, dzieł sztuki oraz teorii naukowych, to sprowadzenie ich do wspólnego mianownika wyda się rzeczą niemożliwą. (...) Nie szukamy tu jedności wyników, ale jedności działania; nie jedności wytworów, ale jedności procesu twórczego.”⁸⁴¹

Wnikamy zatem, według Cassirera, w formę wyobraźni i to jak ona działa, a nie w konstrukcję wytworu. Jedność języka jako formy symbolicznej jest funkcjonalna, a nie

838 Ch. Taylor, *Teoria znaczenia...*, s. 161-169.

839 Ch. Taylor, *Źródła podmiotowości...*, s. 69.

840 E. Cassirer, *Esej o człowieku...*, s. 400.

841 *Ibidem*, s. 157.

substancjalna⁸⁴², podobnie więź łącząca poszczególne formy symboliczne (*vinculum functionale*)⁸⁴³. Przypisując językowi funkcję konstruktywną interesuje nas nie ostateczny rezultat, lecz sam proces⁸⁴⁴ działania.

Człowiek zostaje określony przez swą aktywność⁸⁴⁵, gdyż jak pisał Cassirer: „to właśnie praca, to właśnie system ludzkich działań określa i determinuje zakres pojęcia »człowieczeństwo«”⁸⁴⁶. Wiedzę o nim, z perspektywy filozofii marburczyka, zdobywamy w sposób pośredni, poprzez analizę kultury. Filozof ten zdefiniował zatem człowieka funkcjonalnie, a nie substancjalnie⁸⁴⁷, jako tego którego określa aktywność poznawcza⁸⁴⁸ i jej kulturowe rezultaty⁸⁴⁹. Jak zauważyła w związku z tym Anna Musioł, według Cassirera: „o człowieku decyduje jego wytwórczość, w ścisłym sensie to, w jaki sposób tworzy, oraz to, co staje się przedmiotem owej wytwórczości”⁸⁵⁰. Konstytuowanie się w języku tego, „czym jest człowiek”⁸⁵¹, można dostrzec zatem analizując pracę programistyczną, głównie koncentrując się na aspekcie działania w języku⁸⁵²: pisanie kodu źródłowego. Na podstawie tego twórczego wymiaru odbywa się poszukiwanie przez programistów odpowiedzi na pytanie „kim jestem?”.

Podobnie jak w przypadku filozofii form symbolicznych, na badanie ludzkiego działania w języku, ukierunkowuje też Ricœurowska hermeneutyka dyskursu. Człowiek w jej ujęciu jest także przedstawiany jako istota nastawiona na poznanie i zatopiona w kulturze. Kwintesencją działania językowego jest sposób myślenia w języku, ale nie jak u Cassirera – ustrukturalizowany – lecz rozumiejący. Według koncepcji francuskiego filozofa rozumienie człowieka staje się możliwe poprzez zapośredniczenie w *medium* tekstu, bo „cóż wiedzielibyśmy o miłości i o nienawiści, o uczuciach etycznych, i

842 *Ibidem*, s. 253.

843 *Ibidem*, s. 153.

844 *Ibidem*, s. 254-253.

845 H. Buczyńska, *Cassirer...*, s. 53.

846 E. Cassirer, *Esej o człowieku...*, s. 153.

847 *Ibidem*.

848 A. Gemel, *Językowy model...*, s. 174.

849 H. Buczyńska, *Cassirer...*, s. 53.

850 A. Musioł, *Neokantyzm epistemologiczny ...*, s. 289.

851 E. Cassirer, *Esej o człowieku...*, s. 152.

852 Działanie językowe w definicji Rady Europy: „stosowanie językowej kompetencji komunikacyjnej przejawia się w działaniach językowych, czyli w rozumieniu i tworzeniu tekstów (działania receptywne i produktywne), i w działaniach interakcyjnych i mediacyjnych (w szczególności podczas tłumaczenia). Wszystkie te działania dotyczą tekstu w formie mówionej, pisanej lub w obydwu”.

Europejski system opisu kształcenia językowego: uczenie się nauczanie, ocenianie, tłum. W. Martyniuk, Wydawnictwo Centralnego Ośrodka Doskonalenia nauczycieli, Białystok 2003, s. 24, <https://ore.edu.pl/2010/12/europejski-system-opisu-ksztacenia-jzykowego-uczenie-si-nauczanie-ocenianie/> (dostęp dnia 22.09.2024).

ogólnie o tym wszystkim, co nazywamy sobą, gdyby nie zostało to wypowiedziane i sformułowane przez literaturę”⁸⁵³. Myśliciel posłużył się pojęciem „literatury” jako przykładem „utrwalonego dzieła kultury”⁸⁵⁴. Na kod źródłowy możemy, jak już zostało wielokrotnie powiedziane, spojrzeć zaś właśnie jak na dzieło zawierające „znaki ludzkości”⁸⁵⁵.

W ujęciu filozofii Ricœura „człowiek indywidualizuje siebie”⁸⁵⁶ będąc twórcą czegoś niepowtarzalnego – swojego dzieła. Filozof zaproponował trzy wymogi, które należałoby spełnić, aby wytwór określić mianem dzieła. Po pierwsze powinno ono stanowić zamkniętą całość będącą sekwencją dłuższą niż zdanie⁸⁵⁷. Program komputerowy spełnia to kryterium, gdyż składa się z zestawu instrukcji, realizujących określoną przez programistę logikę. Ponadto dany dyskurs jako dzieło powinien według Ricœura móc zostać sklasyfikowany, z innymi mu podobnymi, jako odrębny gatunek – co stanowi drugi warunek. Chociaż, we wspomnianym fragmencie filozof odnosił się do form utworów literackich, jednak w rozwinięciu problemu stwierdził, że to sama natura kodyfikacji nadaje zasadnicze znaczenie, a nie obszar wystąpienia dzieła. Istotę tego warunku stanowi bowiem wykorzystanie kategorii „wytwarzania”. Następuje ono poprzez nadawanie formy materii i poddanie „wytwarzania” regułom⁸⁵⁸. W tym ujęciu tekst kodu źródłowego spełnia drugi wymóg, który wobec dyskursu stawiał Ricœur, gdyż należy także do pewnej kategorii wytworów – użytkowych tekstów napisanych w językach programowania.

Trzeci warunek jaki stawiał Ricœur w kwestii traktowania dyskursu jako dzieła, wiąże się z „otrzymaniem (przez nie, *przyp. aut.*) sobie tylko właściwej postaci”⁸⁵⁹. Poprzez indywidualny styl, nadany przez autora ma ono bowiem szansę na niepowtarzalność. Tym sposobem wspomniana wcześniej struktura, w znaczeniu formy dzieła, włącza się w obręb indywidualnej praktyki⁸⁶⁰. W obszarze kodu źródłowego dostrzec możemy, na przykład, ujawnianie się indywidualnego stylu programisty (o czym była mowa w podrozdziale 3.2), występującego w ramach pewnych ram

853 P. Ricœur, *Hermeneutyczna funkcja ...*, s. 243.

854 *Ibidem*.

855 *Ibidem*.

856 M. Drwięga, *Paul Ricœur daje...*, s. 59.

857 P. Ricœur, *Hermeneutyczna funkcja ...*, s. 232.

858 *Ibidem*.

859 *Ibidem*.

860 *Ibidem*, s. 233.

strukturalnych, czyli tego, co dopuszczalne w danym języku, technologii czy projekcie. Styl wskazuje zatem na istnienie jakiegoś wyróżniającego się twórcy dzieła językowego, nie jedynie osoby komunikującej coś w języku⁸⁶¹. Koncepcja dzieła powiązana jest z dostrzeganiem indywidualności jego autora i możliwością konstytuowania w języku tego, kim jest.

Ricœur podjął tematykę pracy ludzkiej, która obiektywizuje się w dziełach. Dzięki temu posunięciu w ogóle możliwe staje się metodologiczne podejście do języka. Według francuskiego filozofa wątpliwe jest bowiem, budowanie ontologii bezpośredniej „wymykającej się od razu wszelkim żądaniom metodologii”⁸⁶². W związku z tym dzieła dyskursywne da się zbadać poprzez poszukiwania warunków możliwości pracy, czyli tego jak struktura włącza się w indywidualną praktykę⁸⁶³. Rozważania Ricœura nad dyskursem jako dziełem, pozwalają odnieść się do kwestii posługiwania się językiem przez programistów i wytwarzania przez nich kodu źródłowego, a co za tym idzie rozważyć ich rolę jako autorów, definiujących się na przykład jako inżynierowie, rzemieślnicy, lub artyści. Warto zatem zastanowić się jak tożsamość programistów może kształtować się w oparciu o działanie twórcze, które podejmują.

W kontekście rozważań Ricœura nad wytwarzaniem, można przedstawić pracę programistów w ujęciu jednostkowym jako praktykę twórczą. Jest to zbieżne z polskim prawem, które umożliwia zatrudnianie tego typu specjalistów jako twórców⁸⁶⁴. Działalność programistyczna polegająca na tworzeniu kodu źródłowego jest w jego świetle przejawem twórczości podlegającej ochronie i wsparciu podobnemu jakie jest

861 M. Bała, *O możliwości hermeneutycznej* ..., s. 46.

862 P. Ricœur, *Egzystencja i hermeneutyka*..., s. 185.

863 P. Ricœur, *Hermeneutyczna funkcja* ..., s. 233.

864 Programy komputerowe są bowiem chronione prawe autorskim, jako utwory w rozumieniu „Ustawy o prawie autorskim i prawach pokrewnych” z dnia 4 lutego 1994 roku. Widnieje w niej zapis, iż: „przedmiotem prawa autorskiego jest każdy przejaw działalności twórczej o indywidualnym charakterze, ustalony w jakiejkolwiek postaci, niezależnie od wartości, przeznaczenia i sposobu wyrażenia (utwór)” Art. 1.1 „Ustawa z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych”. Dalej wśród utworów chronionych prawem autorskim wymienione zostały w obowiązującej obecnie ustawie utwory „literackie, publicystyczne, naukowe, kartograficzne oraz programy komputerowe”, które są „wyrażone słowem, symbolami matematycznymi, znakami graficznymi”, <https://sip.lex.pl/akty-prawne/dzu-dziennik-ustaw/prawo-autorskie-i-prawa-pokrewne-16795787/art-> (dostęp dnia 22.09.2024); <https://wachowski.law.com/2022/03/23/50-koszty-uzyskania-przychodu-dla-programistow/> (dostęp dnia 22.09.2024).

oferowane aktywności literackiej, czy pracy architektonicznej⁸⁶⁵. Jednak, mimo że w ramach polskiego porządku prawnego usankcjonowane jest określanie programistów mianem twórców oraz, co za tym idzie, uznaje się twórczy wymiar powstawania programów komputerowych, warto też wspomnieć o wątpliwościach, jakie mogą się w tej materii pojawić. Osoby zajmujące się pisaniem kodu źródłowego nie tworzą jednorodnej grupy, niektóre z nich określają się jako rzemieślnicy, inne jako artyści czy inżynierowie. Niewłaściwe byłoby zignorowanie tożsamości, którą sami sobie przypisują, jak twierdził Ricœur: „dyskurs odnosi się bowiem nie tylko do świata, lecz odnosi się także wstecz ku swemu autorowi”⁸⁶⁶. Rozważania te wiążą zatem ściśle hermeneutykę dyskursu z hermeneutyką narracji i tożsamości narracyjnej.

Według Katarzyny Rosner w ujęciu Ricœura narracja „to kody tekstowe wypracowane przez kulturę”⁸⁶⁷. W tym znaczeniu utożsamia się narrację z tekstem dyskursywnym⁸⁶⁸. Nie stanowi ona, według francuskiego filozofa, struktury ludzkiego myślenia, lecz jest czymś przyswajającym i wykorzystywanym w procesie rozumienia. Myśliciel traktuje narrację jak tekst podlegający odczytaniu, nie zaś jako rodzaj struktury kognitywnej. Nie przyrównamy zatem narracji w ujęciu Ricœura do Cassirerowskiego symbolizmu, ponieważ ten drugi nadaje według marburczyka strukturę myśleniu, wobec czego zabieg taki byłby nieuprawniony.

Narracje, jak już zostało wspomniane, stanowią kody kulturowe, które wykorzystuje człowiek by zrozumieć swoje doświadczenie⁸⁶⁹. Ricœur pisał o „pierwszeństwie wytwórczej działalności intryg w stosunku do wszelkiego rodzaju statycznych struktur”⁸⁷⁰. Doświadczenie istnieje jako formowane przez narrację.

865 „przychodów uzyskanych z prac autorskich w ramach listy zawodów uprawnionych (...) działalności twórczej w zakresie architektury, architektury wnętrz, architektury krajobrazu, inżynierii budowlanej, urbanistyki, literatury, sztuk, plastycznych, wzornictwa przemysłowego, muzyki, fotografii, twórczości audialnej i audiowizualnej, programów komputerowych, gier komputerowych, teatru, kostiumografii, scenografii, reżyserii, choreografii, lutnictwa artystycznego, sztuki ludowej oraz dziennikarstwa” <https://www.e-pity.pl/autorskie-koszty-uzyskania-przychodu/> (dostęp dnia 28.12.2023); Ma to również swoje konsekwencje podatkowe, programistom, podobnie jak innym twórcom np. architektom czy pisarzom, przysługuje korzystanie z 50% kosztów uzyskania przychodu.

866 P. Ricœur, *Język jako dyskurs...*, s. 92.

867 K. Rosner, *Narracja, tożsamość, czas*, Universitas, Kraków 2003, s. 131.

868 *Ibidem*.

869 *Ibidem*.

870 P. Ricœur, *Czas i opowieść. Intryga i historyczna opowieść*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2008, s. 56.

Człowiek korzysta z niej jak z czegoś zewnętrznego, bo wytworzonego społecznie, gdyż to kody kulturowe mają moc kształtowania dyskursu⁸⁷¹.

Na narrację autobiograficzną, podobnie jak Ricœur, zapatrywali się Alasdair MacIntyre^{872,873} i Charles Taylor, wskazując na odpowiedniość jedności życia wobec jedności opowieści⁸⁷⁴. Źródło ich myśli stanowiła jednak Arystotelesowska „Etyka”, gdy dla Francuza miała być nim „Estetyka”. Według Katarzyny Filutowskiej teoria podmiotowości Ricœura odwołuje się do kantowskiego obowiązku moralnego, nie ma zaś na celu realizacji dobrego życia⁸⁷⁵.

Myśl Ricœur zawarta w „Czasie i opowieści” i myśl Taylora ze „Źródeł podmiotowości” stanowiły świadectwo zainteresowania świadomością i poznaniem w obrębie filozofii⁸⁷⁶. Obaj filozofowie stawiali tezę etycznego narratywizmu, czyli przekonania o przeżywaniu przez ludzi swojego życia jako narracji i stosowności takiego rozumienia siebie⁸⁷⁷. Według Kanadyjczyka człowiek z natury jest samorozumiejącym się zwierzęciem⁸⁷⁸. Francuski filozof zaproponował zaś autobiograficzne *self*, gdzie spójna tożsamość powstaje w ciągłości czasu. W procesie tym ma znaczący udział język, gdyż to w nim powstają narrację jednostki⁸⁷⁹. W tak wykreowanej tożsamości, jako obszarze różnorodnych napięć lokalizują się narracje.

Krytyczne wobec narratywizmu stanowisko zajął filozof analityczny Galen Strawson, sprzeciwiając się deskryptywnej tezie o narratywistycznej naturze ludzkiego doświadczenia (*psychological Narrativity thesis*) i normatywnemu twierdzeniu o powinności takiego życia (*ethical Narrativity thesis*)⁸⁸⁰. Zwrócił uwagę, że jest możliwe doświadczenie w inny sposób niż w tym modelu, a faworyzowanie go – w kontekście psychoterapeutycznym – tworzy niepotrzebny nacisk na pacjentów i jest, według filozofa, destrukcyjne. Stworzył koncepcje epizodycznego i diachronicznego samodoświadczenia, które miałyby występować równie często jak narratywizm.

871 K. Rosner, *Paul Ricœur wobec ...*, s. 131.

872 *Ibidem*, s. 122.

873 Bardziej radykalne ujęcie narracyjne człowieka prezentowali literaturoznawczynie Barbara Hardy, psycholog David Carr oraz inni przedstawiciele psychologii kognitywistycznej i uznając narrację za prymarną strukturę poznawczą, *Ibidem*, s. 123.

874 K. Filutowska, *Tożsamość narracyjna jako empiryczna podmiotowość – MacIntyre, Taylor, Ricœur*, Wydawnictwo Uniwersytetu Warszawskiego, Warszawa 2018, s. 235.

875 *Ibidem*, s. 238.

876 R. Sendyka, *Od kultury ja do kultury siebie. O zwrotnych formach w projektach tożsamościowych*, Universitas, Kraków 2015, s. 91.

877 *Ibidem*, s. 383.

878 Ch. Taylor, *Źródła podmiotowości...*, s. 214.

879 R. Sendyka, *Od kultury ja ...*, s. 95.

880 G. Strawson, *Against narrativity*, „Ratio”, 17, nr 4, 2004, s. 428.

Niestety pominął kwestię poparcia tej tezy badaniami naukowymi, a odwołał się jedynie do literatury. Pomijając jednak aspirujące do krytyki narratywizmu, a nie osadzone w psychologii przemyślenia filozofa, można zgodzić się z tym, że istnieją ludzie, którzy nie prowadzą dialogu wewnętrznego. Co więcej, w badaniach psychologicznych odkryto nawet, że część ludzi nie posiada głosu wewnętrznego (*anaduralia*), a niektóre osoby nie są w stanie też posługiwać się obrazami (*aphantasia*)⁸⁸¹. Narratywizm nie stanowi zatem biologicznej konieczności, co najwyżej może być propozycją promowaną przez niektórych filozofów, na przykład Ricœur, który dostrzegał wartość w budowaniu tożsamości w oparciu o narrację we własnym życiu.

W całej swojej filozoficznej twórczości Ricœur postrzegał samorozumienie jako zadanie do zrealizowania w języku. Miało to związek z jego krytyką możliwości bezpośredniego poznania siebie, realizujące się w niezgodzie wobec „filozofii bezpośredniości”. Jak zostało to już podkreślone, według francuskiego myśliciela, ludzie mają możliwość poznania siebie dopiero poprzez sposób w jaki działają w świecie. Ricœur postulował przebycia „długiej drogi”, tak zwanej „okrężnej drogi” hermeneutycznego rozumienia. Począwszy od hermeneutyki symboli, kiedy upominał się o „poprawę swojego położenie w bycie”⁸⁸² poprzez przemianę świadomości refleksyjnej, kontynuował myśl o konstruowaniu siebie w okres tworzenia hermeneutyki zaufania⁸⁸³. Wyszedł wówczas poza zaproponowaną wcześniej fenomenologię *Sacrum* i skierował się ku rozumieniu refleksji jako „wysiłkowi uchwycenia »Ego« z »Ego Cogito« w zwierciadle jego przedmiotów, jego dzieł i – w końcu – jego aktów”⁸⁸⁴. Ujmując hermeneutykę jako przedsięwzięcie epistemologiczne, w którym pojęcie „»sensu« nasycza racjonalność”⁸⁸⁵, wykazywał konieczność podejmowania się przez ludzi działania. Świadomość była bowiem według niego zadaniem, nie zaś tym, co dane⁸⁸⁶.

881 R. Hinwar, A Lambert, *Anauralia: The Silent Mind and Its Association With Aphantasia*, „Frontiers in Psychology”, 12, 2021, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8551557/> (dostęp dnia 10.01.2024).

882 P. Ricœur, *Symbolika zła*, ...s. 336.

883obejmującej: „De l’interprétation, essai sur Freud” z 1966 roku, „Le conflit des interprétations. Essais d’herméneutique” z 1969 roku, według J. Grondin, *Do Gadamer and Ricœur have the same understanding of hermeneutics?*, [w:] *The Agon of Interpretations: Towards a Critical Intercultural Hermeneutics* red. M. Xie, Toronto University Press, Toronto 2014, s. 46.

884 P. Ricœur, *O interpretacji. Esej ...*, s. 48.

885 P. Ricœur, *Krytyka i przekonanie...*, s. 108.

886 P. Ricœur, *O interpretacji. Esej ...*, s. 48.

W okresie hermeneutyki dyskursu, *Self* rozpoznane zostaje przez Ricœura jako podmiot dyskursu⁸⁸⁷, warunkiem zaś bycia sobą ma być zdolność do działania w języku. Świadomość siebie konstytuowana jest przez tekst, gdyż:

„rozumienie siebie w obliczu tekstu nie polega na narzucaniu tekstowi kategorii naszej własnej ograniczonej zdolności rozumienia, lecz na wystawieniu się na jego działanie tak by otrzymywać od tekstu szersze Ja, które stanowiłoby sposób egzystencji najlepiej odpowiadający światu prezentowanemu przez tekst”⁸⁸⁸.

Świadomość nie ustanawia się zatem sama, ale czyni to na drodze interpretacji znaczeń⁸⁸⁹. Rozumienie „ja” zależy od działania jednostki konstytuującej się na drodze otwarcia się na dyskurs, rozpostarty przez tekst „horyzont znaczenia”⁸⁹⁰. Również w kolejnym okresie twórczości – „hermeneutyce czasu i narracji”⁸⁹¹, gdy Ricœur tworzy koncepcję tożsamości narracyjnej będzie tę tożsamość pojmował jako przejawiającą się w działaniu⁸⁹². Bycie sobą jest tym samym odpowiedzią na pytanie: kto jest sprawcą. Podmiot ani nie jest substancjalną iluzją, co filozof zarzucał koncepcji Nietzschego, ani niezmiennie takim samym (*idem*)⁸⁹³. Refiguruje natomiast on stale historię swojego życia, będąc jej autorem i czytelnikiem. Opowieść o „działającym” pozwala na utrzymanie ciągłości tożsamości, czyli poczucia, że „ja” to „ja”. W kontekście twórczości programistów daje się przywołać przykład zapominania języka w związku z jego nieużywaniem i ponowny powrót do pisania w tymże, dzięki pamiętaniu kluczowych elementów⁸⁹⁴.

Ricœur, ujmując dyskurs jako dzieło, przedstawił język jako surowiec podlegający kształtowaniu. Zauważył, że twórca nadaje materii formę, czyli podlega w swej pracy zasadom jakie narzuca mu dany gatunek sztuki, a jednocześnie wytwarza coś niepowtarzalnego. Dyskurs w tym kontekście podlega praktyce (*praxis*) i technice

887 R. Sendyka, *Od kultury ja ...* s. 44.

888 P. Ricœur, *Towards a hermeneutics of the Idea of Revelation*, [w:] *Esseys on Biblical Interpretation*, s. 108 tłum. za K. Rosner, K. Rosner, *Paul Ricœur – filozoficzne źródła jego hermeneutyki*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 23-24.

889 K. Rosner, *Paul Ricœur – filozoficzne źródła jego hermeneutyki*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 25.

890 P. Ricœur, *Wyjaśnianie i rozumienie...*, s. 165.

891 obejmującej „Tems et récit. Tome I”, „Tems et récit. Tome II. La configuration dans le récit de la fiction”, „Tems et récit. Tome III. Le tems raconté”, wydane od 1983-85 roku za: J. Grondin, *Do Gadamer ...*, s. 46.

892 P. Ricœur, *Czas i opowieść. Czas opowiadany*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2008, s. 353.

893 *Ibidem*.

894 <https://qr.ae/psR2NU> (dostęp dnia 19.02.2024).

(*technè*)⁸⁹⁵ i, jak zaważył francuski filozof, „pod tym względem nie ma zdecydowanej opozycji między pracą umysłu a pracą rąk”⁸⁹⁶. Można zatem powiedzieć, że programista podejmując się wspomnianej pracy umysłowej polegającej na pisaniu kodu źródłowego, wykonuje pracę w dziedzinie dyskursu.

Swoją teorię działania Ricœur rozwinął w okresie „hermeneutyki dyskursu”. W „Du texte á l’action” polemizował z podejściem Elisabeth Anscombe i Donalda Davidsona. Zauważył, że owi analityczni filozofowie ignorują kwestię sprawcy działania. Są zainteresowani odpowiedzią na pytanie „co?” i „dlaczego?”, lecz pomijają pytanie „kto?”⁸⁹⁷. Kierując się chwalebny dążeniem do zgodności opisu z przedmiotem, poprzez praktykę pomijania podmiotu (np. poprzez zastąpienie zaimka i czasownika „ja chcę” przysłówkiem „chcenie”) Anscombe przyczyniła się do tego, że przedmiot działający zniknął⁸⁹⁸. Tymczasem wskazanie na sprawcę działania jest konieczne z uwagi na „prawdomówność tekstu”⁸⁹⁹. Ricœur tym mianem określał autentyczność, która niesie ze sobą świadectwo człowieka, gdyż bezosobowy opis odnosi się jedynie do kryterium prawdy.

Dzieło, w myśl II hermeneutyki Ricœura, stanowi „rezultat pracy organizującej język”⁹⁰⁰. Myśliciel za filozofem Gillesem-Gastonem Grangerem zauważył wagę pracy, która to obiektywizuje się w bytach jednostkowych – dziełach, w tym dziełach dyskursu⁹⁰¹. Możliwość konstytuowania siebie w języku jest tak istotna, że „wraz z ograniczeniem możliwości działania, odczuwamy jako osłabienie dążności do istnienia, rozpoczyna się we właściwym sensie panowanie cierpienia”⁹⁰². Ludzka praca ma zatem wymiar egzystencjalny, bierność bowiem według Ricœura „okrywa żałobą »chwałę« działania”⁹⁰³. Jest tu zgodny z Arendt, która w ludzkim działaniu dostrzegała „wskazanie na właściwe sobie umiejscowienie w świecie”⁹⁰⁴. Filozof powoływał się właśnie na kategorię autorstwa działania⁹⁰⁵ jej koncepcji z „Kondycji ludzkiej”, gdy definiował „tożsamości narracyjną”. Ludzka aktywność naznaczona jest

895 P. Ricœur, *Hermeneutyczna funkcja...*, s. 232.

896 *Ibidem*.

897 M. Drwięga, *Paul Ricœur daje...*, 120.

898 *Ibidem*, s. 121.

899 *Ibidem*.

900 P. Ricœur, *Hermeneutyczna funkcja ...*, s. 233.

901 *Ibidem*.

902 P. Ricœur, *O sobie samym ...*, s. 532.

903 *Ibidem*, s. 533.

904 H. Arendt, *Kondycja ludzka*, Aletheia, Warszawa 2010, s. 95

905 P. Ricœur, *Czas i opowieść. Czas opowiadany...*s. 353.

indywidualizmem sprawcy działania⁹⁰⁶, jego punktu widzenia i jego wyborów, różnych od tych które mają inni ludzie. Ricœur powołując się na myśl Arendt o istotności dostrzeżenia „»kto« działania” i z tworzenia, na tej podstawie, historii życia, wywodził narracyjną tożsamość „bycia sobą samym”⁹⁰⁷.

Ricœur zaproponował uspoźnianie tożsamości osobowej z wykorzystaniem opowieści o działaniu podmiotu. Bycie takim samym (*idem*) oparte o tożsamość substancjalną, proponował zastąpić koncepcją bycia sobą (*ipse, ipséité*)⁹⁰⁸. O ile pierwsze odnosi się do abstrakcyjnej identyczności, drugie stanowi dynamiczną strukturę. *Ipséité* jest „refigurowane przez refleksyjne zastosowanie konfiguracji narracyjnych”⁹⁰⁹. Tożsamość narracyjna, według Ricœura z okresu prac nad „hermeneutyką odkrycia siebie”⁹¹⁰ jest dynamiczna. Jak zauważył autor „Czasu i opowieści”: „zawsze jest możliwe utkanie różnych – a nawet przeciwnych intryg w odniesieniu do własnego życia”⁹¹¹. Programiści zatem, również mogą zmieniać narrację o sobie samym. Postrzeganie siebie może ewoluować od bycia inżynierem, przez bycie rzemieślnikiem czy artystą⁹¹².

Prześledźmy zatem finalnie jak tworzy się wśród programistów przekonanie o tym, kim są na podstawie ich działalności. Rozważmy wpierw przykład tych spośród nich, którzy identyfikują się jako rzemieślnicy (*craftman*). Ich działaniu przyświeca idea profesjonalizmu wyrażona w nazwie *Software Craftsmanship*, czyli rzemieślniczego kunsztu programowania⁹¹³. Programiści ci, w swojej pracy szukają najprostszych rozwiązań dla problemów, gdyż:

906 K. Gurczyńska-Sady, *Troska o świat. Antropologia filozoficzna Hannah Arendt*, Wydawnictwo Naukowe UP, Kraków 2019, s. 22.

907 P. Ricœur, *Czas i opowieść. Czas opowiadany...*s. 353.

908 *Ibidem*.

909 *Ibidem*.

910 Thum. własne *Recovery self*, za A. Scott-Baumann, *Ricœur and the Hermeneutics of Suspicion*, Bloomsbury Academic, London, 2009.

911 P. Ricœur, *Czas i opowieść. Czas opowiadany...*, s. 356.

912 M. Clifton, *The Software Development Process - Science, Engineering, Art, or Craft?*, 2016 <https://www.codeproject.com/Articles/1130886/The-Software-Development-Process-Science-Engineeri> (dostęp dnia 7.02.2024).

913 S. Mancuso, *Software Craftsman. Profesjonalizm, czysty kod i techniczna perfekcja*, Helion, Gliwice 2016, s. 17.

„prawdziwy developer-rzemieślnik skupia się przede wszystkim na rozwiązaniu problemu, a nie na pisaniu kodu. Gdy jednak musi napisać jakiś kod, dokłada wszelkich starań, aby był to kod dobrej jakości – żeby był testowalny, zrozumiały i łatwy w utrzymaniu”⁹¹⁴.

Rzemieślnicy są tak zwanymi „pragmatycznymi programistami”⁹¹⁵, czyli nastawiają się na działanie z uwzględnieniem kontekstu, w którym pracują. Takie podejście do pracy pojawiło się w latach 90 XX wieku wraz z rozwojem metodyki Agile, a w ramach kontry wobec wcześniejszego inżynierskiego podejścia wyrażanego sentencją „jeśli to nie jest sprawa kodu, to nie jest mój problem”⁹¹⁶.

Co ciekawe w literaturze związanej z podejściem rzemieślniczym do rozwoju oprogramowania pojawia się wiele metafor zaczerpniętych z funkcjonowania cechów rzemieślniczych, na przykład, określenie terminatora (*apprentices*), czeladnika (*journeyman*) i mistrza (*master*) podkreślające wagę nauki zawodu i trenowania pod okiem kogoś przekazującego swoją mądrość⁹¹⁷. Istotną wagę przypisuje się kwestii zdobywania doświadczenia, brania odpowiedzialności za wykonaną pracę, współpracy „a ponad wszystko miłości wobec pisania oprogramowania”⁹¹⁸. Osoby poczuwające się do tej tożsamości mogą zrzeszać się w społecznościach (np. *Paris Software Craftsmanship Community*)⁹¹⁹, a nawet złożyć podpis pod stylizowanym na dokument cechu Manifestem Rzemiosła Oprogramowania (*Manifesto for Software Craftsmanship*)⁹²⁰ – międzynarodową inicjatywą zrzeszającą programistów, którym bliski jest ten model.

914 *Ibidem*, s. 234.

915 A. Hunt, D. Thomas, *Pragmatyczny programista. Od czeladnika do mistrza*, Helion, Gliwice 2011, s. 21.

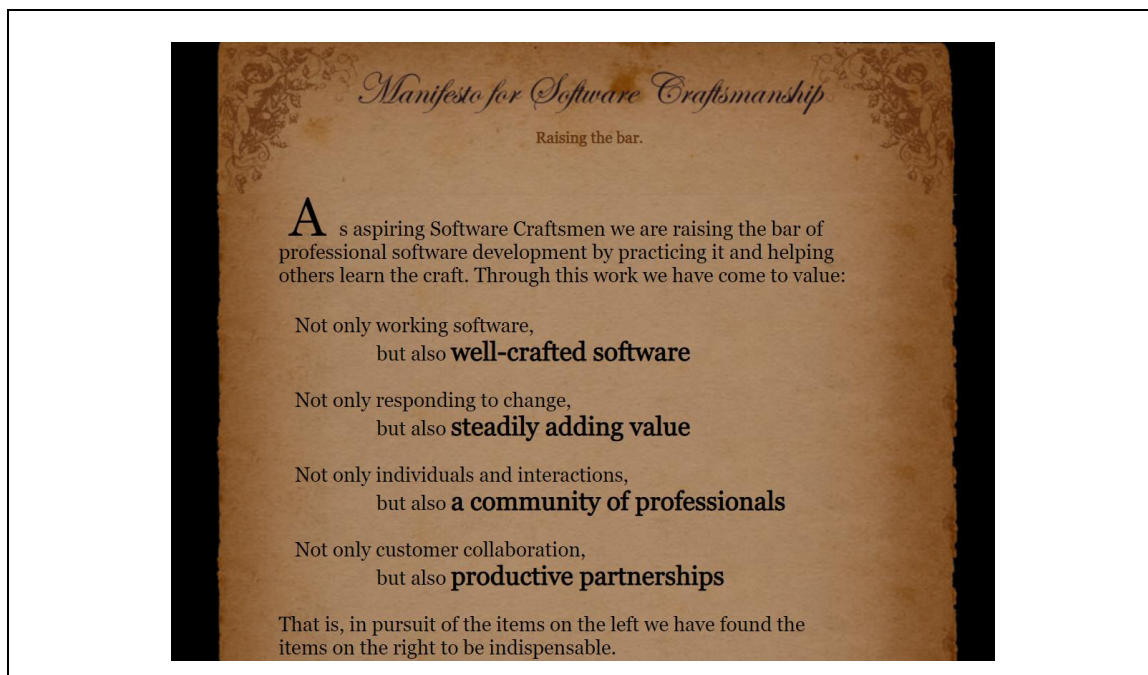
916 S. Mancuso, *Software Craftsman* ..., s. 30.

917 B. Pyritz, *Craftsmanship versus engineering: Computer programming—An art or a science?*, „Bell Labs Technical Journal”, 8, nr 3, 2003, s. 102.
<https://archive.ph/20130105055559/http://www3.interscience.wiley.com/journal/106570667/abstract#selection-497.0-497.27> (dostęp dnia 25.01.2024).

918 *Ibidem*, s. 103.

919 S. Mancuso, *Software Craftsman* ..., s. 53.

920 <https://manifesto.softwarecraftsmanship.org/> (dostęp dnia 28.12.2023).



Rys. 19. Manifest rzemieślników piszących oprogramowanie⁹²¹

Tożsamością, która uformowała się historycznie wcześniej niż identyfikowanie się jako developer–rzemieślnik, było określenie siebie jako inżyniera (*software engineer*). Rozróżnienie takie jest zgodne z rozważaniami metodologicznymi nad paradygmatami informatyki, w których pojawia się właśnie paradygmat empiryczny i technokratyczny⁹²². Wyszczególnienie podejścia empirycznego związane jest z postrzeganiem działalności informatycznej jako uruchamiania programów⁹²³, na co nastawiony byłby rzemieślnik. Tworzenie oprogramowania w paradygmacie technokratycznym opiera się zaś na dążeniu do pisania niezawodnych, bezbłędnych programów⁹²⁴.

Celem programisty–inżyniera jest „znajdowanie dobrych rozwiązań o akceptowalnym koszcie wdrożenia i utrzymania, (...) projektowanie rozwiązań spełniających stawiane im wymagania”⁹²⁵. Działalność programisty–rzemieślnika od inżyniera miałyby różnić się zatem tym na co nakierowani są oni w swoim działaniu. Pierwsi większą wagę mieliby przypisywać współpracy z ludźmi i ich wymaganiom, drudzy produktom, kierując się przy tym zasadą:

921 <https://manifesto.softwarecraftsmanship.org/> (dostęp dnia 28.12.2023).

922 I. Bondecka-Krzykowska, *Paradygmat Informatyki*, [w:] *Światy matematyki. Tworzenie czy odkrywanie*, Wydawnictwo Naukowe UAM, Poznań 2010, s. 117.

923 *Ibidem*, s. 118.

924 *Ibidem*, s. 122.

925 K. Sacha, *Inżynieria oprogramowania*, Państwowe Wydawnictwo Naukowe, Warszawa 2010, s. 7.

„Inżynier tworzy rzeczy ogólne i wszechstronne. Im bardziej dana rzecz nadaje się do użytku w różnych środowiskach, tym lepiej. Nie ma znaczenia, kto z niej korzysta. Z drugiej strony rzemieślnik bardzo dba o to, jak rzecz będzie dopasowana do docelowego użytkownika. Dostosowują ją do niego”⁹²⁶.

Mimo, że tożsamość rzemieślnika *vel* pragmatycznego programisty, bywa budowana na zasadzie kontrastu wobec inżynierów, nie wydaje się, aby granice tych etykiet były ostre a konflikt między programistami zaogniony.

Kolejną tożsamością występującą wśród programistów jest poczucie bycia artystą. Konstytuuje się ono bądź poprzez przyrównywanie programowania do sztuki, bądź wprost nazwanie działalności programistów aktywnością artystyczną. Elegancki, piękny kod można potraktować jako wyraz potrzeby twórczej realizowanej w ramach sztuki użytkowej⁹²⁷. Przy tworzeniu programów można bowiem wykazać się kreatywnością i odnosić się do wartości estetycznych:

„Nie tylko rozwiązanie problemu informatycznego (algorytmicznego) może zachwycać. Czasami, co docenią przede wszystkim znawcy tematu, zachwyca jego styl, sposób zakodowania, dobór struktur danych. Ten sam problem można rozwiązać i zapisać bardzo chaotycznie, ale można też w stylu, który ujmuje prostotę, przejrzystością, a czerpiąc z terminów bliskich sztuce – pięknem formy (...) rozwiązując bardzo poważne problemy, wymagające stworzenia programu, można to zrobić ze swego rodzaju wdziękiem i kunsztem, albo nieco bardziej chaotycznie, w duchu, co najwyżej, dobrego rzemiosła (czyli być nie programistą-artystą, ale takim właśnie kodowaczem, w tym nieco starszym rozumieniu tego terminu)”⁹²⁸.

W literaturze spotykać się można zatem z przeciwstawieniem programowania jako pełnej inwencji twórczej sztuki, ale też kodowania jako zaledwie zapisu algorytmu. Tej drugiej działalności brak byłoby polotu, a opierałby się jedynie na rzetelności. Tożsamość programisty-artysty budowałaby się, wobec tego, w opozycji do wytwórczości rzemieślniczej. Co ciekawe pojęciem „natchnionych rzemieślników 4.0”⁹²⁹, a zatem łączących oba wspomniane wymiary działania, nazywa Borys Siewczyński architektów korzystających z nowych technologii w projektowaniu. Nie

926 Tłum. własne z Software Craftsman, What it is and what it isn't, <https://bitslap.it/blog/posts/it-craftsman.html> (dostęp dnia 31.01.2024).

927 Z. Ledóchowski, *Z. Rola sztuki programowania w kształceniu informatycznym*, „Edukacyjna Analiza Transakcyjna”, nr 7, 2018, s. 201.

928 *Ibidem*, s. 202-203.

929 B. Siewczyński, *Natchnieni rzemieślnicy 4.0.*, „Architectus”, nr 64, 2020, s. 89.

jest to jednak tożsamość, o której mówiłoby się, jako o przypisywanej sobie przez programistów.

Nazwanie artysty osobą zaangażowaną w komunikację z odbiorcą poprzez swoją twórczość, pozwala nawiązać do charakterystyki człowieka „jako istoty zdolnej do [bycia] sobą”⁹³⁰, a ujawniającej to poprzez „możność mówienia”⁹³¹. Ricœur w potencjale komunikowania widział istotę tworzenia tożsamości. Dialog pomiędzy programistą-artystą a publicznością najoczywistszy staje się w obszarze *gamingu*⁹³². Jako media współtwórcze, gry komputerowe, uwzględniają bowiem interwencję społeczności fanów⁹³³. „Publiczność uczestnicząca” wykazuje się aktywnością, komentując produkt, gdyż „ani sami deweloperzy, ani gracze twórcy nie mogą być wyłącznie odpowiedzialni za stworzenie ostatecznego konglomeratu uznawanego za »grę«, jako że wymaga ona udziału obu tych grup”⁹³⁴.

Powołując się na słowa Ricœura, że „pracując w dziedzinie dyskursu, człowiek wyznacza praktycznie pewną kategorię bytów jednostkowych: dzieł dyskursu”⁹³⁵, możemy także przyjąć, iż sama gra komputerowa stanowi dzieło dyskursu. Hermeneutyka dyskursu zakłada wszak analizę językowych cech tekstu poszerzoną o interpretację z perspektywą odbiorcy⁹³⁶. Język jako dyskurs z założenia zawiera w sobie perspektywę zwracania się do rozmówcy. Tworząc grę deweloperzy budują narrację w oparciu o oczekiwania odbiorców.

Według koncepcji II hermeneutyki Ricœura dyskurs realizuje się poprzez odczytywanie znaczenia nadanego przez autora⁹³⁷. Wydarza się wówczas „przybycie świata do języka”⁹³⁸, stąd mowa o „zdarzeniowym” charakterze dyskursu. Samoreferencja dyskursu polega na tym, że w samej swojej strukturze wskazuje on na swojego autora⁹³⁹. Warto zatem również odnieść się do osób, które są twórcami samych

930 P. Ricœur, *Drogi rozpoznania...*, s. 82.

931 *Ibidem*.

932 granie w gry komputerowe.

933 J. Dovey, *Kultura gier komputerowych*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2011, s. 157.

934 Tłum. własne, S. Morris, *WADs, Bots and Mods: Multiplayer FPS Games as Co-creative Media*, 2003,

<http://www.digra.org/wp-content/uploads/digital-library/05150.21522.pdf> (dostęp 31.01.2024).

935 P. Ricœur, *Hermeneutyczna funkcja...*, s. 233.

936 M. Bała, *O możliwości hermeneutycznej ...*, s. 46.

937 P. Ricœur, *Hermeneutyczna funkcja ...*, s. 228.

938 *Ibidem*.

939 P. Ricœur, *Język jako dyskurs...*, s. 92.

języków. Projektowanie języków programowania stanowi bowiem działalność nakierowaną na tworzenie dyskursu. Twórca takiego przedsięwzięcia jest konstruktorem pewnej narracji o świecie, gdzie „rozgrywający się dyskurs odnosi się wstecz i naprzód, do mówiącego i do świata”⁹⁴⁰.

Decydującym momentem jest, według Ricœura, przejście od wyobraźni do woli. W działaniu realizuje się bowiem według francuskiego filozofa „prawdziwe bycie sobą”⁹⁴¹. Podjęcie odpowiedzialności przez podmiot działający ma wymiar normatywny. Inicjowanie odbywa się bowiem poprzez ustosunkowanie się do świata (ten aspekt będzie natomiast rozwijany w kolejnym rozdziale dotyczącym wartości). Wracając do kwestii konstytuowania „ja” w języku, warto przypomnieć, że Ricœur nadał jaźni narracyjny kształt⁹⁴², gdy zwrócił uwagę na przemieniającą moc doświadczeń⁹⁴³, refleksyjnie i w dynamiczny sposób kształtujących tożsamość. Podmiot jest zatem efektem tych działań. Jego refleksyjność nie ma wymiaru jednoznacznie ontologicznego, lecz stanowi drogę do ontologii, łącząc „pragnienie bycia i wysiłek istnienia”⁹⁴⁴. Nie mając do siebie bezpośredniego dostępu musi przejść „określną drogę” by móc do siebie powrócić. W swoim działaniu w świecie przejawia siebie i dopiero drogą analizy i interpretacji „ten-który-jest-sobą” może stać się sobie danym⁹⁴⁵.

Czytając wywiady z twórcami najbardziej prominentnych języków programowania, wyłania się dość spójny obraz motywów ich językowego działania. Bjarne Stroustrup, twórca C++, rozszerzył język C, nie planując nawet projektowania nowego języka, a jak skromnie deklarował chciał „tylko pomóc w rozwiązaniu kilku problemów”⁹⁴⁶. Twórca C# Anders Hejlsberg również zwrócił uwagę na istotność celowości języka:

940 *Ibidem*, s. 93.

941 P. Ricœur, *Czas i opowieść. Czas opowiadany...*, s. 357.

942 D. Wolska, *Odzyskać doświadczenie. Sporny temat humanistyki współczesnej*, Universitas, Kraków 2012, s. 64.

943 *Ibidem*.

944 P. Ricœur, *Wyzwanie semiologiczne...*, s. 275.

945 M. Kowalska, *Wstęp*, [w:] P. Ricœur, *O sobie samym jako innym*, Państwowe Wydawnictwo Naukowe, Warszawa 2005, s. XI.

946 F. Biancuzzi, S. Warden, *Wielkie umysły programowania. Jak myślą i pracują twórcy najważniejszych języków*, Helion, Gliwice 2010, s. 14.

„stworzyłem dużo języków programowania (...) Myślę, że bardzo ważne jest, aby przed przystąpieniem do tworzenia nowego języka programowania mieć jasność co do tego, dlaczego coś się robi i jaki jest problem do rozwiązania”⁹⁴⁷.

Robin Milner projektant języka *ML* podkreślał w swoim działaniu, że „język został opracowany jako narzędzie dowodzenia twierdzeń”⁹⁴⁸. Wszyscy wspomniani powyżej twórcy języków mieli na uwadze cel, który sami sobie obrali, bazując na swoim doświadczeniu. Odwołując się do hermeneutyki dyskursu można wyjaśnić, że: „dlatego, że jest coś do powiedzenia, ponieważ mamy nasze doświadczenie, które wnosimy do języka, język nie jest skierowany wyłącznie ku znaczeniom idealnym, lecz także odnosi się do tego, co jest”⁹⁴⁹. Projekty nowych języków przenika, jak widać, duch fenomenologii.

Różnorodność powstałych języków jest skutkiem decyzji ich autorów dotyczących tego, co według nich stanowiło problem do rozwiązania i możliwego sposobu podejścia. Uzewnętrznienie, czyli odnoszenie się w dyskursie do świata, jak zauważył Ricœur, jest procesem wychodzącym „od doświadczenia bytowania w świecie i postępującego od tej sytuacji ontologicznej do jej wyrażania w języku”⁹⁵⁰. Przykład wspomnianych autorów deklarujących łączność między domeną problemu a odczuwaną przez nich potrzebą stworzenia nowego języka, doskonale ilustruje wspomnianą referencyjność języka. Dla francuskiego filozofa myśl o wnoszeniu do języka doświadczenia była kluczowa dla jego teorii języka jako dyskursu⁹⁵¹.

Uzewnętrznienie polegające na odnoszeniu się w dyskursie do problemów świata odbywa się w praktyce poprzez uwzględnienie ograniczeń i możliwości, jakie daje zapis w języku programowania, jak na przykład, opisał to twórca C++ Bjarne Stroustrup:

„Prawdziwy problem polega na tym, jak przejść od ludzkiego sposobu postrzegania problemów i rozwiązań do ograniczonego świata maszyny. Można zignorować ludzkie rozterki i stworzyć kod maszynowy (lub gloryfikowany kod maszynowy w postaci złego kodu w języku C). Można też zignorować rozterki maszyny i uzyskać piękne abstrakcje, które pozwalają na wykonywanie dowolnych operacji olbrzymim kosztem i (lub) bez ograniczeń zdroworozsądkowych. Język

947 *Ibidem*, s. 374.

948 *Ibidem*, s. 273.

949 P. Ricœur, *Język jako dyskurs...*, s. 92.

950 *Ibidem*.

951 *Ibidem*.

C++ to próba uzyskania bezpośredniego dostępu do sprzętu (na przykład za pośrednictwem wskaźników i tablic) tam, gdzie jest taka potrzeba, z jednoczesnym dostarczeniem rozbudowanych mechanizmów abstrakcyjnych pozwalających na wyrażanie idei wysokopoziomowych (na przykład hierarchii klas i szablonów)”⁹⁵².

Wspomniane „piękne abstrakcje” tworzone przez programistów odnoszą się do programowania wysokopoziomowego. Warto rozważyć jednak bycie twórcą z perspektywy prób wyrażania w językach programistycznych ludzkich idei i pomysłów.

Człowiek dorosły ma umiejętność posługiwania się abstrakcją, czego wyrazem są „wymiar symbolicznego formowania”⁹⁵³, jak je nazywa Cassirer. Funkcja wyrażania, funkcja przedstawiania i funkcji znaczenia stanowią plan orientacji dla form symbolicznych⁹⁵⁴. Przykład języka posłużył filozofowi do tego by stwierdzić, że wyrażanie odnosi się do obiektywnego istnienia wyrażanych zmysłowo słów. Przedstawianie to przekazywanie za pomocą zdań obiektywnych stanów rzeczy, natomiast znaczenie polega na czystym przyporządkowaniu⁹⁵⁵. Ostatnia ze sfer uwolniła się „od podstawy oglądowego kształtowania, w której ma swoje korzenie przedstawianie, czerpiące z tej podstawy nieustannie swe najlepsze siły – że, jeśli tak wolno powiedzieć, buja ona w swobodnym eterze czystej myśli”⁹⁵⁶. Język, jako forma symboliczna, dąży do tego „królestwa czystego znaczenia”, jak je nazywał Cassirer.

Programiści operując znakami są w swoim funkcjonowaniu językowym najbliżsi sferze czystego znaczenia. Co ciekawe rozwój języków programowania podąża drogą, którą przedstawił Cassirer, czyli ku zwiększeniu poziomu abstrakcji. Języki kolejnych generacji, projektowane są by być przyjazne programistom poprzez zwiększenie poziomu abstrakcji⁹⁵⁷. Twórcy kodu źródłowego, w kontekście rozważań Cassirera nad ewolucją języka, wydają się używać go w sposób dla filozofa najbardziej zaawansowany. Konkurencję dla człowieka w tym względzie stanowią mogłaby natomiast sztuczna inteligencja (AI). Dotychczasowa możliwość stosowania narzędzi

952 F. Biancuzzi, S. Warden, *Wielkie umysły ...*, s. 19.

953 E. Cassirer, E. Cassirer, *Problem symbolu i jego miejsce w systemie filozofii*, [w:] *Symbol i Język...*, s. 52.

954 *Ibidem*, s. 54.

955 *Ibidem*, s. 52-53.

956 *Ibidem*, s. 53.

957 <https://www.pcmag.com/encyclopedia/term/fourth-generation-language> (dostęp 20.02.2024).

no code i *low code* (NCLC) do generowania kodu, aby oszczędzić czas⁹⁵⁸, prowokowała komentarze, jakoby praca programistów miałaby być niepotrzebna, jednak przewidywania te się nie ziściły. Natomiast przyszły wpływ AI na działania programistów może wydawać się o wiele bardziej znaczący.

Oprócz generowania odpowiedzi na podstawie zbioru danych, sztuczna inteligencja potrafi pisać również programy komputerowe. Z użyciem *ChatGPT* można otrzymać kod źródłowy, poprawny bądź błędny⁹⁵⁹. Rozwój tego narzędzia może skutkować zastąpieniem części pracy wykonywanej przez programistów – programami generowanymi przez sztuczną inteligencję. Najprawdopodobniej jednak nadal będzie potrzeba nadzorowania pracy i weryfikacji poprawność tak powstałych programów przez inżynierów „rozumiejących sytuację”⁹⁶⁰. Rozpoczynając pisanie niniejszej rozprawy autorka nie sądziła, że może stać się ona raportem o minionym świecie. W marcu 2023 roku, wraz z powstaniem piszącego programy komputerowe *ChatGPT*, perspektywa częściowego zastąpienia pracy ludzi przez sztuczną inteligencję (*artificial intelligence*) w tym zakresie, staje się coraz bardziej realna. Warto jednak podkreślić, że będzie to najprawdopodobniej wieloletni proces i nie wyeliminuje z rynku programistów o wysokich kompetencjach.

Wraz z rozwojem technologii AI dochodzimy do momentu, w którym należy się zastanowić jak potencjalna sytuacja braku zapotrzebowania na pracę programistów podczas tworzenia kodu źródłowego, rezonuje z tworzeniem przez te osoby swojej tożsamości. Roma Sendyka dostrzegła, ponowoczesną tendencję do rozpadu tożsamości na skutek takich czynników jak: życie w warunkach nieustannej zmiany, konieczność ciągłej autokreacji, i co istotne dla tej pracy, „wszelkie ewentualne (jak w przestrzeni wirtualnej) samookreślenia”⁹⁶¹. Programiści, którzy przecież część swojego życia zawodowego spędzają na tworzeniu i kontaktach w przestrzeni nie-fizycznej,

958 C. Richardson, J. Rymer, *Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms*, „Forrester”, 2016, s. 4, https://informationsecurity.report/Resources/Whitepapers/0eb07c59-b01c-4399-9022-dfc297487060_Forrester%20Vendor%20Landscape%20The%20Fractured,%20Fertile%20Terrain.pdf (dostęp dnia 21.02.2024).

959 H. Richards, *Engineering Education after ChatGPT Paper presented at 14th Annual First-Year Engineering Experience (FYEE) Conference*, University of Tennessee in Knoxville, Tennessee 2023, s. 2-3, <https://peer.asee.org/engineering-education-after-chatgpt> (dostęp dnia 14.12.2023).

960 *Ibidem*, s. 4.

961 R. Sendyka, *Od kultury ja ...*, s. 42.

podlegaliby temu procesowi. Również obawa o zastąpienie ich pracy aktywnością sztucznej inteligencji może przyczyniać się do rozpadu tożsamości.

Remedium na wspomnianą powyżej sytuację odnalazł Anthony Giddens w utrzymywaniu przez jednostkę spójności poprzez kontakt z własnym ciałem i własną narrację osobową⁹⁶². Te mechanizmy koherencyjne przejawiają się u programistów poprzez konstytuowanie tożsamości zarówno jednostkowej jak i wspólnotowej⁹⁶³. Bycie sobą wspólnoty tworzy się w języku przyjmującym jakieś wspólne opowieści, bycie sobą pojedynczego podmiotu natomiast w określaniu siebie, na przykład, poprzez definiowanie się jako programista-artysta, -rzemieślnik, czy -inżynier. Tożsamość narracyjna w ujęciu Ricœura jest natomiast „poszukiwanym miejscem (...) skrzyżowania historii i fikcji”⁹⁶⁴. Nieustannie się destabilizuje i tworzy na nowo, więc wyzwania jakie stawiają zmieniające się warunki technologiczne, w ujęciu francuskiego filozofa nie powodowałyby rozpadu tożsamości.

Konstytuowanie „ja” w języku poprzez działania programistów natrafia na pewne trudności, chociażby związane z tworzeniem wielu równoległych tożsamości, czy możliwości jej utraty w związku z pojawieniem się piszących programy nowych technologii. Sztuczna inteligencja zastępująca człowieka w wykonywanej przez niego pracy, również intelektualnej, stanowi według niektórych ludzi zagrożenie. Można podejść do problemu mniej radykalnie i zastanowić się jakim jest wyzwaniem, uwzględniając obszar etyczny. Narracyjność, jako snucie opowieści, według Ricœura, wprowadza do etyki z uwagi na to, że tekst stanowi swoiste „laboratorium, w którym wypróbowuje się wartościowania”⁹⁶⁵. Można zastanowić się zatem jak w języku kodu źródłowego uwidacznia się aspekt etyczny działania programistów, w kontekście ich działań i opowieści jakie o tychże działaniach snują.

962 A. Giddens, *Nowoczesność i tożsamość*, Państwowe Wydawnictwo Naukowe, Warszawa 2001, s. 138.

963 P. Ricœur, *Czas i opowieść. Czas opowiadany...*, s. 354.

964 P. Ricœur, *O sobie samym...*, s. 190.

965 *Ibidem*, s. 192.

4. Rozpoznawanie standardów

Człowiek wchodzi w szczególną relację ze światem i z samym sobą dzięki ukrytemu w języku potencjałowi. Daje on bowiem możliwość wyrażania wartości. W języku tworzy się standardy zachowania moralnego, a więc jak pisze Taylor: dokonuje się rozróżnienia na to, co właściwe, bądź nie. Podmiotowość moralna jednostki związana jest tym samym z jej świadomością standardów moralnych. Ta kształtowana jest w ramach działań językowych polegających na „przeprowadzaniu rozróżnień mających podstawowe znaczenie dla spraw obchodzących ludzi i tym samym sprawianiu, że stajemy się na nie otwarci”⁹⁶⁶. Przedmiotem rozważań w tym rozdziale stanie się zatem język jako *medium*, w którym dochodzi do ekspresji, ujawniania i konstytuowania standardów moralnych w obszarze kodu źródłowego programów komputerowych. Programiści mogą bowiem wyrażać w języku sprawy, które są przedmiotem ich troski i wskazywać tym samym wartości, na które są otwarci.

Wymiar etyczny stał się szczególnie istotny w okresie późnej twórczości zarówno Ricœura, jak i Cassirera. Francuski filozof podjął wspomnianą problematykę w dziele „O sobie samym jako innym”, gdzie rozważał narracyjność jako „wprowadzenie do etyki”⁹⁶⁷ oraz w istotnym dla poniższych rozważań tekście: „Osoba: struktura etyczna i moralna”⁹⁶⁸. Marburczyk nawiązał zaś do kwestii etycznych w ostatnim okresie pisarstwa, który Simon Truwant datuje na lata 1935-1946⁹⁶⁹. Podczas pobytu w Szwecji powstała nieprzetłumaczona na polski praca „Axel Hägerström”⁹⁷⁰, a w trakcie zamieszkania w USA wydany już pośmiertnie „Mit państwa”. W swojej ostatniej książce stworzył koncepcję „narodzin etyki indywidualnej z ducha negacji mitu”⁹⁷¹.

Jeśli potraktujemy język kodu źródłowego jako dyskurs i formę symboliczną możemy rozważać aspekt rozpoznawania w obszarze języka standardów moralnych.

966 Ch. Taylor, *Teoria znaczenia...*, s. 161.

967 P. Ricœur, *O sobie samym...*, s. 192.

968 P. Ricœur, *Osoba: struktura etyczna i moralna*, [w:] *Zawierzyć człowiekowi*, red. Zbigniew Brzeziński, Znak, Kraków 1991.

969 S. Truwant, *The concept of 'function' in Cassirer's historical, systematic, and ethical writings*, [w:], *The Philosophy of Ernst Cassirer*, red. T. Friedman, S. Lufts, Marquette University, Milwaukee 2015, s. 289,

<https://www.degruyter.com/document/doi/10.1515/9783110421811-012/html?lang=en> (dostęp dnia 31.07.2024).

970 D. Baumgardt, *Cassirer and the chaos in moder ethics*, [w:] *The philosophy of Ernst Cassirer*, red. P. Schilpp, Georg Banta Publishing Company, Evanstone 1949, s. 557.

971 za K. Jarzyńska, *Ernst Cassirer. Mit Państwa*, „Politeja”, nr 1, 2008, s. 518.

Kształtowanie się wymiaru etycznego zostanie zatem przedstawione w rozważaniach nad ekspresją, ujawnianiem i konstytuowaniem się ich w języku.

Wyrażanie wartości odbywa się poprzez działanie językowe podjęte przez programistów w językach programowania. Istnieje możliwość prowadzenia rozważań etycznych odnosząc się do tekstów napisanych w językach sztucznych, gdy ujmimy język jako dyskurs lub formę symboliczną. Warto zastanowić się jednak czy w ogóle konieczne jest wyodrębnienie kwestii języków programowania, którymi posługują się ludzie, gdy podejmują decyzje moralne. Może wystarczyłoby rozważać działania podmiotów moralnych – programistów, nie odnosząc się do języków, w których są one podejmowane. W pracy tej zakłada się jednak, że język jest sposobem bycia w świecie i spekuluje na temat tego, jak ludzie myślą i wyrażają rzeczywistość w języku. Warto, zatem rozważyć wymiar etyczny działań językowych podjętych w językach programowania. W celu unaocznienia sensowności tego wyodrębnienia można, jak już było czynione wcześniej, przedstawić języki programowania jako języki drugie, by następnie móc posłużyć się wynikami badań z zakresu psychologii. W dyscyplinie tej podjęto się eksploracji naukowej nad różnicami decyzji moralnych podejmowanymi w zależności od wykorzystywanego języka: pierwszego (ojczystego) i drugiego (obcego).

Wpływ obcego języka na podejmowanie decyzji moralnych został w psychologii zbadany z wykorzystaniem znanego w etyce dylematu wagonika (*trolley dilemma, standard trolley problem*). Badani przez Janet Geipel i jej współpracowników mieli podjąć decyzję czy w celu ocalenia pociągu pełnego pasażerów, zdecydować się przestawić zwrotnice i zabić 5 innych osób przywiązanych do torów. Aby bardziej uwydatnić decyzję zabicia wykorzystano także wariant bardziej bezpośredni - zwany dylematem kładki (*footbridge dilemma, footbridge problem*). Ten wariant eksperymentu myślowego wymagał zadecydowania czy osoby badane osobiście zrzuciłyby kogoś na tory, aby uratować innych ludzi⁹⁷². Do badania zrekrutowano grupę spośród Włochów i Chińczyków, sprawdzając czy na decyzje ma wpływ otrzymanie instrukcji w znanym im języku obcym (angielskim i niemieckim).

We wspomnianych badaniach Geipel nad wpływem języka obcego na podejmowanie decyzji moralnych, wykorzystano rozróżnienie na decyzje

972 J. Geipel, C. Hadjichristidis, L. Surian, *The Foreign Language Effect on Moral Judgment: The Role of Emotions and Norms*, "Plos one", nr 10, 2015.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4503530/> (dostęp dnia 24.02.2024).

reprezentujące deontologiczny sposób rozumienia moralności, uznając je za inspirowane emocjami i konsekwencjonalistycznym rozumem⁹⁷³. Zauważono, że w dylemacie kładki, na wzrost decyzji utylitarystycznych wpływa użycie języka obcego, w odniesieniu do kontrolnej grupy używającej wyłącznie języka ojczystego (brak różnicy w dylemacie wagonika)⁹⁷⁴. Różnice te autorzy wyjaśniają mniejszą ilością dostępnych w obcym języku norm społecznych i moralnych⁹⁷⁵.

Dla poniższych rozważań nie jest istotne, którą postawę uznamy za słuszną: utylitarystyczną czy absolutystyczną. Ważne jest natomiast zaistnienie samej różnicy między osądami moralnymi podjętymi w języku ojczystym i obcym (*the moral foreign-language effect*)⁹⁷⁶. Skoro ta odmiennność występuje można mniemać o występowaniu swoistości zachowań w zależności od wykorzystania języka. Sprawia to, że zasadne wydaje się rozważanie kwestii rozpoznawania standardów moralnych w odniesieniu do języków. Jeśli język programowania uznamy za drugi język, ten używany codziennie w pracy, pozwoli to rozważyć kwestię moralności nie jako etykę zawodową tej grupy.

Język stanowi *medium* wyrażania standardów moralnych, gdyż to w nim kształtują się i za jego pośrednictwem wypowiedane są kwestie etyczne. Jak bowiem zauważył Taylor: „specyficzne ludzkie przedmioty troski dostępne są tylko istotom językowym, gdyż po to, by być na nie wrażliwymi, musimy w pewien sposób wyartykułować czy dać wyraz zrozumieniu odpowiednich standardów oraz zasadniczych rozróżnień, które za sobą pociągają”⁹⁷⁷.

Ekspresja w języku ujawnia stosunek ludzi do otaczającego ich świata, a więc postawę wobec niego, ale też „pokazuje naturę i styl (tego, *przyp. aut.*) uczestnictwa”⁹⁷⁸. Wyraża się bowiem w nim rozróżnianie standardów przez świadomy ich podmiot i chociaż języki programowania są kojarzone z opisem wolnym od

973 *Ibidem*, s. 2.

974 *Ibidem*, s. 5.

975 Zmniejszenie dostępności do norm moralnych i społecznych badacze zaproponowali tłumaczyć z wykorzystaniem hipotezy o mniejszej dostępności do wspomnień autobiograficznych w językach obcych – te są tworzone w języku pierwszym, *Ibidem*, s. 13.

976 H. Cipelletti, S. McFarlane, C. Weissglass, *The moral foreign-language effect*, „*Philosophical Psychology*” nr 29, 2016, <https://www.tandfonline.com/doi/full/10.1080/09515089.2014.993063> (dostęp dnia 06.04.2024);

C. Harris, J. Gleason, A. Ayçiçeği A, *When is a first language more emotional? Psychophysiological evidence from bilingual speakers*, [w:] *Bilingual minds: Emotional experience, expression, and representation*, red. A. Pavlenko, Multilingual Matters, Clevedon 2006.

977 Ch. Taylor, *Teoria znaczenia...*, s. 165.

978 *Ibidem*, s. 163.

ekspresji, to możliwość wyrażania w nich wartości sprawia, że nie funkcjonują w „konwersatoryjnej czeluści”⁹⁷⁹ (co czasami się im zarzuca). Istotne jest, że w tekście kodu źródłowego mają sposobność wyrazić się postawy wobec kwestii etycznych, „bo wystarczy, że dajemy wyraz tej wrażliwości, nawet jeśli nie mamy jeszcze słów, aby opisać tę cnotę bądź wadę, dobro albo zło”⁹⁸⁰. Ujmując język jako dyskurs i formę symboliczną można zastanowić się nad rodzajem wyrażonych wartości.

4.1. Wymiar ekspresji

Dzięki ekspresji w języku możemy ujawnić własną postawę moralną⁹⁸¹. Wyraża się w nim bowiem ludzki stosunek do ważnych kwestii – przedmiotów troski, jak je nazywał Taylor⁹⁸². Są one, według niego, dostępne „tylko istotom językowym, gdyż po to, by na nie być wrażliwymi, musimy w pewien sposób wyartykułować czy dać wyraz zrozumieniu odpowiednich standardów oraz zasadniczych rozróżnień, które za sobą pociągają”⁹⁸³. Ujawnia się zatem w ekspresji językowej to, co jest uznawane za ważne w danej grupie. Bardzo charakterystycznym przykładem, którym można posłużyć się w rozważaniach nad wartościami w środowisku programistów, jest aktywność subkultury hakerów (ang. *hacker*). Ich działalność w językach programowania ma różnoraki charakter: przestępczy, rozrywkowy, ale także motywowany ideologicznie. W podrozdziale 3.3 rozważana była kwestia tożsamości przyjmowanej przez programistów, jednak nie została wymieniona tożsamość hakera, gdyż jako „tożsamość przestępcza”⁹⁸⁴ wiąże się ściśle z wymiarem etycznym, dlatego też dopiero tu znajduje swoje miejsce.

Konstruowanie tożsamości przez programistów-artystów/rzemieślników/inżynierów, w oparciu o samoidentyfikację w kontekście sposobu wykonywania swojej pracy, wydaje się podejściem niepełnym. Sytuuje w prawdzie problem w obszarze działania tych profesjonalistów w języku, jednak pomija

979 *Ibidem*, s. 165.

980 *Ibidem*.

981 *Ibidem*.

982 *Ibidem*.

983 *Ibidem*.

984 Zmiana tożsamości przestępczej jako nieakceptowanej społecznie przynależności leży u podstaw nauki o resocjalizacji. M. Kuć, *Prawne podstawy resocjalizacji*, Beck, Warszawa 2013, s. 3. https://www.ksiegarnia.beck.pl/media/product_custom_files/9/9/9965-prawne-podstawy-resocjalizacji-malgorzata-kuc-darmowy-fragment.pdf (dostęp dnia 06.04.2024).

w ustanawianiu „ja” aspekt etyczny. Opowieść o sobie „należy już do pola etycznego na mocy roszczenia, właściwego każdej narracji – do etycznej słuszności”⁹⁸⁵. Wizja świata narratora nie może być bowiem neutralna etycznie. Ricœurowskie „bycie sobą” wymaga decyzyjności i przyjęcia odpowiedzialności w działaniu⁹⁸⁶. W tworzeniu tożsamości narracyjnej istotny element stanowi zatem wymiar normatywny podjętych decyzji. Potrzebę uwzględnienia wymiaru etycznego w ujmowaniu człowieka podnoszono także w nurcie personalizmu chrześcijańskiego, tyle że w kontekście odpowiedzi na „boskie wezwanie”⁹⁸⁷. W ramach tej perspektywy wskazuje się na niewłaściwość utożsamianie istoty ludzkiej jedynie z jej rolą, pomijając powołanie moralne⁹⁸⁸.

W ramach chrześcijańskiej myśli personalistycznej Emmanuel Mounier zaproponował odróżnienie zindywidualizowanej „jednostki ludzkiej” rozumianej jako produkt cywilizacji od „osoby ludzkiej” otwartej na tworzenie relacji w ramach wspólnoty⁹⁸⁹, obdarzonej świadomością⁹⁹⁰, wolą i wolnością⁹⁹¹. Istota ludzka, żyjąc wśród innych, odczuwa powołanie moralne, które polega na wrażliwości na wartości. W tym ujęciu powołanie zawodowe stanowi zaledwie zadanie do wykonania⁹⁹², natomiast powołanie istoty ludzkiej wymaga uwzględnienia aspektu moralnego. W procesie personalizacji, by wzmocnić byt⁹⁹³ dochodzi, według francuskiego personalisty, do koncentracji i spajania działania z wymiarem etycznym⁹⁹⁴. Warto w tym kontekście zastanowić się jak w aktywności takiej jak praca, wyrażać może się moralność. Zarówno Cassirer, jak i Ricœur podjęli się refleksji nad kwestią dążeń etycznych ludzi. Ich przemyślenia będą mogły zatem stanowić ramę rozważań nad hakerstwem.

985 P. Ricœur, *Czas i opowieść. Czas opowiadany...*, s. 357.

986 *Ibidem*.

987 F. Coplestone, *Historia filozofii. Od Maine de Birana do Sartre’a. T9*, Instytut Wydawniczy PAX, Warszawa 2006, s. 274.

988 *Ibidem*, s. 273.

989 J. Burgos, *Personalizmy. Autorzy i tematy nowej filozofii*, Centrum Myśli Jana Pawła II, Warszawa 2010, s. 70-71.

990 E. Mounier, *Wprowadzenie do egzystencjalizmów oraz wybór innych prac*, Społeczny instytut Wydawniczy Znak, Warszawa 1964, s. 11.

991 *Ibidem*, s. 13.

992 *Ibidem*, s. 151.

993 *Ibidem*, s. 150-151.

994 *Ibidem*, s. 152.

W działaniach hakerów, biorąc pod uwagę kantowską normę personalistyczną, można dostrzec nieetyczność. Człowieczeństwo ich ofiar jest przez nich używane jedynie jako środek, gdy je okradają i szantażują. Jednak w popkulturze utrwalił się mit hakera, romantyzujący działalność tych osób. Warto zatem wpierw przybliżyć to, na czym polega ich aktywność. O ile programistom uznającym siebie za rzemieślników czy inżynierów przyświecać miałyby wzniosłe ideały użyteczności ich pracy, to działalność hakerów kojarzona jest jednak powszechnie z cyberprzestępstwami. Wykorzystują oni zaawansowane umiejętności programistyczne w celu łamania zabezpieczeń.

Ataki hakerskie można podzielić na 3 typy, te legalne wykonywane przez tzw. „białych hakerów” (*white hacker, white hat hacking*), nielegalne przez tzw. „czarnych hakerów”, oraz działania na granicy prawa tzw. „szarych hakerów”⁹⁹⁵. Etyczne hakerstwo dotyczy testów penetracyjnych zleconych przez firmę w celu sprawdzenia zabezpieczeń jej systemu. Biały haker jest zatem pracownikiem wykorzystującym swoje umiejętności. Zdobył je wcześniej ćwicząc z poszanowaniem prawa bądź bez. Poza taką właśnie sztucznie wykreowaną sytuacją, aktywność hakerów jest uznawana za nielegalną. Inny przykład białego hakerstwa stanowi uczestnictwo w zawodach bezpieczeństwa informatycznego⁹⁹⁶, zwanych *hackathonami*.

W szarej strefie znajdują się osoby, które bez zgody właścicieli włamują się do ich systemów, jednak nie czerpią z tego korzyści finansowych. W doktrynie prawa angielskiego taki nieuprawniony dostęp, jeśli wiąże się ze zmianą danych, jest już karany, nawet jeśli motywacje hakera były altruistyczne i chciał jedynie wskazać słabe strony systemu. Ostatnią grupę stanowią osoby, których działalność programistyczna łamie prawo (w polskim systemie prawnym 267 § 1 i 2 Kodeksu Karnego). Najczęściej wiąże się to z cyberprzestępstwami polegającymi na przełamaniu zabezpieczeń, aby osiągnąć korzyść majątkową lub wyrządzić szkodę, innej osobie lub organizacji⁹⁹⁷.

995A. Piechocki, *Haker: Czarny, Szary, a może Biały. Co to jest "white hacking" i czy jest legalny?*, „Gazeta Prawna”, 2020 <https://prawo.gazetaprawna.pl/artykuly/1447091,white-hacking-odpowiedzialnosc-karna.html> (dostęp dnia 15.03.2024).

996 <https://www.nask.pl/pl/aktualnosci/2212,Polacy-w-czolowce-miedzynarodowych-zawodow-bezpieczenstwa-teleinformatycznego.html> (dostęp dnia 23.03.2024).

997 *Ibidem*.

Czynność podejmowane przez hakerów polegają głównie na łamaniu normy, czyli „reguły lub zasady opisującej działania, które są wymagane, dopuszczalne lub zakazane niezależnie od jakichkolwiek prawnych lub społecznych instytucji”⁹⁹⁸. Zanim bowiem w prawodawstwie pojawiły się przepisy, które można stosować w przypadku hakerstwa, istniały normy sprzeciwiające się podobnym działaniom. Chociaż zauważalne jest zróżnicowanie norm w poszczególnych społecznościach i na przestrzeni dziejów, to można dostrzec jednak pewne powszechniki. Jeśli ujmemy normy na wysokim poziomie ogólności, ukażą się nam pewne uniwersalne zasady. Dotyczą one reguł dzielenia się, zakazu agresji i kazirodstwa⁹⁹⁹. Zjawisko hakerstwa powiązane jest z dwoma pierwszymi. Warto zauważyć, że u podstaw norm leżą wartości¹⁰⁰⁰, i te możemy również postarać się zidentyfikować. One bowiem nadają wewnętrzną treść normom, będąc „abstrakcyjnymi standardami sądu praktycznego”¹⁰⁰¹. Rozważając jednak przykład *hackingu*, wyjdziemy od wartości, które towarzyszyły mu u początków formowania się tego ruchu: są nimi humor i wolność. W odniesieniu do nich ukształtował się bowiem mit hakera – „kowboja klawiatury”.

Angielskie słowo *hack*, od którego powstało słowo *hacking* i *hacker* oznaczało pierwotnie inteligentny żart¹⁰⁰². Aby coś mogło zostać uznane za niego, musiało być nowatorskie, stylowe i dopracowane technicznie¹⁰⁰³. Termin ten zaczął funkcjonować przed erą cyfrową w amerykańskim środowisku naukowym z obszaru STEM (skrót od *science, technology, engineering, mathematics*). „Ludyczny impuls był ważną siłą napędową w kształtowaniu się technologii komputerowych”¹⁰⁰⁴, jedna z pierwszych gier komputerowych *Spacewars* była efektem żartu, polegającego na sprawdzeniu do czego da się użyć nowo dostarczony na campus komputer PDP-1¹⁰⁰⁵. Ta interaktywność miałaby być właśnie nawiązaniem do zasady, którą Sandy Stone nazwała „szturchnij i zobacz, co się stanie”.

998 A. Macko, *Umysł moralny: jak powstają oceny moralne*, Poltex, Warszawa 2018, s. 31.

999 *Ibidem*, s. 31-33.

1000 J. Mariański, *Socjologia Moralności*, Wydawnictwo KUL, Lublin 2006, s. 350.

1001 *Ibidem*.

1002 J. Dovey, *Kultura gier komputerowych*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2011, s. 47.

1003 *Ibidem*, s. 48.

1004 *Ibidem*, s. 49.

1005 *Ibidem*, s. 48.

Zabawa i eksperymentowanie leżały u początków rozwoju *hackingu*. Jak mówił Marquard: „śmiej się jest małą teodyceją”¹⁰⁰⁶, dostrzegając leżący w nim zasób. Umożliwia on bowiem mówienie o poważnych sprawach z lekkością i otworzenie się na nowe. Komizm uznać można za wartość estetyczną¹⁰⁰⁷. Stanowi rodzaj dobra¹⁰⁰⁸, do którego dążyli w swojej ekspresji pierwsi hakerzy. Na podstawie ludycznej i pociągającej otoczki towarzyszącej tworzeniu pierwszych programów-żartów powstał też mit hakera. Haker miał być bohaterem komputerowego podziemia¹⁰⁰⁹. Mitologizacja tożsamości i kultury hakerskiej wiązała się z postacią nieprzystosowanego do życia mężczyzny działającego z pasją w ramach pewnej wspólnoty podobnych do niego osób¹⁰¹⁰. Haker miał charakteryzować się sprytem i brawurą, która kreowała go na „kowboja klawiatury”¹⁰¹¹. Ponieważ jednak jego mit powstał w latach 80-tych XX wieku w środowisku związanym z MIT, siłą rzeczy przedstawiał białego mężczyznę z klasy średniej (WASP, ang. *white, anglo-saxon, protestant*)¹⁰¹². Obecnie nie da się zignorować, chociażby kwestii silnej reprezentacji Chińczyków wśród osób parających się taką działalnością¹⁰¹³.

Jak pisał Wojciech Bober „hakerzy są trochę anarchistami, trochę antyglobalistami”¹⁰¹⁴. Wiąże się to z ich sprzeciwem wobec komercjalizacji sieci i uznaniem dostępu do informacji za dobro. Występują przeciw monopolistycznym praktykom, chociażby wątpliwemu etycznie sprzedawaniu dostępu do artykułów naukowych przez wydawców nieplacącym z założenia autorom publikacji naukowych. Przykładem takiej działalności jest biblioteka cyfrowa *Sci-Hub* stworzona przez pochodzącą z Kazachstanu programistkę Alexandrę Asanovnę Elbakyan, łamiącą dodatkowo tym samym mit hakera-mężczyzny. „Królowa piractwa naukowego”¹⁰¹⁵

1006 O. Marquard, *Apologia przypadkowości*, Oficyna Naukowa, Warszawa 1994, s. 143.

1007 M. Gołaszewska, *Śmieszność i komizm*, Polska Akademia Nauk, Wrocław 1987, s. 67; czerpanie z kompensacyjnej teorii estetyczności Joachima Rittersa przez Marquarda O. Marquard, *Apologia przypadkowości...*, s. 145.

1008 Arystoteles, *Etyka Nikomachejska*, Państwowe Wydawnictwo Naukowe, Warszawa 1956, s. 3.

1009 S. Levy, *Hackers: Heroes of the Computer Underground*, 1984.

1010 J. Dovey, *Kultura gier ...*, s. 87.

1011 *Ibidem*, s. 86.

1012 J. Dovey, *Kultura gier ...*, s. 87, za K. Coyle, *How hard Can IT Be?*, [w:] *Wired Women: Gender and New Realities in Cyberspace*, red. L. Cherny, Seal Press, Washington 1996, s. 44.

1013 <https://interestingengineering.com/lists/7-countries-having-the-majority-of-hackers> (dostęp dnia 30.05.2025).

1014 W. Bober, *Powinność w świecie cyfrowym: etyka komputerowa w świetle współczesnej filozofii moralnej*, Wydawnictwo Akademickie i Profesjonalne, Warszawa 2008, 188.

1015 <https://www.theverge.com/2018/2/8/16985666/alexandra-elbakyan-sci-hub-open-access-science-papers-lawsuit> (dostęp dnia 23.03.2024).

przebywa w ukryciu w związku z roszczeniami finansowymi wydawców i obawą ekstradycji. Celem stworzonej przez nią strony było umożliwienie dostępu do publikacji akademickich z pominięciem praw własności intelektualnej. Tego typu piractwo medialne uszczupla przychody wydawców, jednocześnie w większości nie wpływa na zarobki autorów, bowiem za publikacje artykułów nie pobierają wynagrodzenia. Ponad 50% akademików przyznaje się też do pozyskiwania artykułów z wykorzystaniem pirackich stron (*scholarly piracy site*)¹⁰¹⁶.

Wolność realizować może się również w pewnej mierze w działalności programistycznej nieukierunkowanej na rozbójnictwo. Autonomiczność podejmowanych przez twórców kodu źródłowego decyzji stanowi bowiem charakterystykę tegoż zawodu. Mimo że pracują oni zazwyczaj w grupach i pod nadzorem to jednak „niewidoczność operacji”, czyli brak kontaktu z kodem źródłowym a jedynie z efektem pracy w postaci działającego programu, „skazuje klientów na zaufanie”¹⁰¹⁷. Ekspresja wolności, jakiej dokonują w języku programiści, związana jest z mniejszą możliwością fizycznego nadzoru ich pracy przez zleceniodawcę w porównaniu z innymi branżami. Zadaniem programistów jest rozsądzić samemu według własnego osądu i powinności moralnej¹⁰¹⁸, czy postępują właściwie.

Za powinność moralną w kontekście działań wolnościowych można uznać deklarowaną przez hakerów chęć dzielenia się informacją. Przykładowo wspomniana Elbakyan deklarowała, że opiera swoją działalność na przekonaniu o potrzebie wolnego dostępu (*open access*)¹⁰¹⁹, który jest niezbędny według niej „do pełnego rozwoju świadomości ludzkości”¹⁰²⁰, ale będąc osobą związaną z nauką wśród motywów założenia Sci-Hub wymienia również swoją potrzebę pozyskiwania wiedzy. Wedle „Jargon File” (rys. 20), dokumentu, w którym utrwalono podstawowe zasady etyki hakerskiej i mogącego być traktowanym jako manifest wspólnoty hakerskiej, wymienia się dzielenie się informacją jako podstawowe dobro¹⁰²¹. O ile pisanie otwartego

1016 F. Segado-Boj, J. Martín-Quevedo, J. Prieto-Gutiérrez, *Jumping over the paywall: Strategies and motivations for scholarly piracy and other alternatives*, „Information Development”, 40, nr 3, 2024. <https://journals.sagepub.com/doi/full/10.1177/026666669221144429f> (dostęp dnia 23.03.2024).

1017 W. Bober, *Powinność w świecie* ..., s. 99.

1018 E. Cassirer, *Mit państwa*, Wydawnictwo IFIS PAN, Warszawa 2006, s. 319.

1019 wolny dostęp jest koncepcją ruchu naukowego Open Access Movement, zakładającą potrzebę wolnego dostępu do cyfrowych dokumentów z treściami naukowymi i edukacyjnymi.

1020 <https://www.nytimes.com/2016/03/13/opinion/sunday/should-all-research-papers-be-free.html> (dostęp dnia 23.03.2024).

1021 W. Bober, *Powinność w świecie* ..., s. 187.

oprogramowania (*open-source*) ma być etyczną powinnością, to włamywanie się do systemów dopuszczalne jest jedynie dla zabawy, nie w celu kradzieży, wandalizmu czy naruszenia tajemnicy¹⁰²². W kontrze do uznanej przez wielu¹⁰²³ za chwalebna działalności Elbakyan, można przedstawić argument o okradaniu twórców z ich własności intelektualnej¹⁰²⁴ podając przykład utworów, na których ich autorzy mogliby zarabiać. Haking kojarzony jest bowiem głównie z łamaniem przez tę subkulturę normy poszanowania prywatności i własności. Znaczenie słowa haker zmieniło się na przestrzeni 50 lat z entuzjasty komputerowego na cyberprzestępcę¹⁰²⁵. Przedstawiony powyżej mit „kowboja klawiatury” warto zestawzić z realiami działania tej grupy programistów i zanalizować wykorzystując Cassirerowską i Ricœurowską myśl etyczną.

Prev
hacker ethic
H
Next

hacker ethic: n.

1. The belief that information-sharing is a powerful positive good, and that it is an ethical duty of hackers to share their expertise by writing open-source code and facilitating access to information and to computing resources wherever possible.
2. The belief that system-cracking for fun and exploration is ethically OK as long as the cracker commits no theft, vandalism, or breach of confidentiality.

Both of these normative ethical principles are widely, but by no means universally, accepted among hackers. Most hackers subscribe to the hacker ethic in sense 1, and many act on it by writing and giving away open-source software. A few go further and assert that *all* information should be free and *any* proprietary control of it is bad; this is the philosophy behind the [GNU](#) project.

Sense 2 is more controversial: some people consider the act of cracking itself to be unethical, like breaking and entering. But the belief that 'ethical' cracking excludes destruction at least moderates the behavior of people who see themselves as 'benign' crackers (see also [samurai](#), [greg hat](#)). On this view, it may be one of the highest forms of hackerly courtesy to (a) break into a system, and then (b) explain to the sysop, preferably by email from a [superuser](#) account, exactly how it was done and how the hole can be plugged — acting as an unpaid (and unsolicited) [tiger team](#).

The most reliable manifestation of either version of the hacker ethic is that almost all hackers are actively willing to share technical tricks, software, and (where possible) computing resources with other hackers. Huge cooperative networks such as [Usenet](#), [FidoNet](#) and the Internet itself can function without central control because of this trait; they both rely on and reinforce a sense of community that may be hackerdom's most valuable intangible asset.

Prev
hacker
Up
Home
Next
hacker humor

Rys. 20. Tak zwany „Jargon File” (słownik gwary hakerskiej), w którym spisano zasady etyki hakerskiej¹⁰²⁶

Kwestia wyrażania w języku standardów moralnych zarysowała się wyraźnie dopiero w ostatnim okresie twórczości Cassirera. Filozof w swoich rozważaniach położył wówczas nacisk na zagrożenie w kierowaniu się myśleniem mitycznym, powodującym „przewartościowanie wszystkich naszych wartości etycznych, ale także transformację ludzkiej mowy”¹⁰²⁷. Zjawisko to polegać miało według niego na zmianie znaczeń słów poprzez pobudzanie emocji i nasycanie tak powstałych „słów

1022 *Ibidem*.

1023 Magazyn *Nature* uznał ją za jedną z najbardziej wpływowych osób nauki w 2016 roku. <https://www.nature.com/articles/540507a> (dostęp dnia 23.03.2024).

1024 W. Bober, *Powinność w świecie ...*, s. 188.

1025 *Ibidem*.

1026 <http://www.catb.org/jargon/html/H/hacker-ethic.html> (dostęp dnia 20.03.2024).

1027 E. Cassirer, *Mit państwa...*, s. 314.

mitycznych” nowymi gwałtownymi uczuciami. Powstała w ten sposób „emocjonalna mowa” przejęła kontrolę nad racjonalnym językiem. Siła mitu sprawiła, że hakerzy, motywowani w większości do kradzieży chęcią zysku, są przedstawiani w popkulturze jako jednostki ceniące sobie ideały wolności, troski o „wolny dostęp”, czy walczący z niesprawiedliwością¹⁰²⁸.

Spostrzeżenia margburczyka na temat mitycznego myślenia, które według niego stanowiło o sile politycznej, między innymi, nazizmu kontrastują z rozpowszechnionym wśród socjologów przekonaniem o przyczynach Zagłady leżących podobno w aparacie biurokratycznym, nastawieniu na efektywność, naukowym sposobie myślenia – ogólnie czynnikach z obszaru racjonalizmu¹⁰²⁹. Cassirer był natomiast przekonany o konieczności ujarzmienia sił mitu siłami intelektualnymi, etycznymi i artystycznymi¹⁰³⁰ w celu zapobieżenia nawrotu chaosu wojny. W późnym okresie twórczości filozof odszedł bowiem od wyznawanego wcześniej poglądu o przechodzeniu kultury przez kolejne stadia – formy myślenia ku współwystępowaniu czasowym różnych form i konieczności obrony przed irracjonalnością mitycznego myślenia¹⁰³¹. „Czerpanie siły z afektywnej, emocjonalnej natury człowieka (które zachodzi w micie, *przyp. aut.*) nad racjonalnym oglądem rzeczywistości”¹⁰³² uznawał za zagrożenie.

Według Cassirera to „dzikość” w nas jest motorem zła¹⁰³³. Wynika ona z niemożliwości przewyciężenia „kondycji człowieka dzikiego”¹⁰³⁴. Człowiek współczesny ulegając jej zamiast „zadawać pytania swojemu otoczeniu”¹⁰³⁵, godzi się bezrefleksyjnie z rzeczywistością. Poddaje się myśleniu mitycznemu z jego rytuałami, stając się elementem swoistego przedstawienia wyreżyserowanego przez „innych”, na

1028 Np. „The Net”, „The Matrix”, „V for Vendetta”, „The Girl with the Dragon Tattoo”, <https://cybersecurityventures.com/movies-about-cybersecurity-and-hacking/> (dostęp dnia 23.03.2024).

1029 <https://instytutprawobywatelskich.pl/zygmunt-bauman-nowoczesnosc-i-zaglada/> (dostęp dnia 19.03.2024).

1030 E. Cassirer, *Mit państwa...*, s. 330.

1031 M. Ples, *O micie politycznym w kontekście symbolizmu Ernsta Cassirera*, „Filo-Sofija”, nr 11, 2010, s. 86.

1032 *Ibidem*.

1033 E. Cassirer, *Mit Państwa*, ...s. 317.

1034 Cassirer używa słowa *dziki* w 1946 roku, kiedy to nieznane było pojęcie plemion izolowanych. Ponadto ujawnia przyjętą wówczas wyższościową perspektywę charakterystyczną wówczas dla świata zachodniego w opisie tych ludów. Jednak można przyjąć, że krytyce podlega kwestia dzikości niespokojnej ludzkiej natury, nie zachowanie przedstawicieli jakichś konkretnych plemion w E. Cassirer, *Mit państwa...*, s. 317.

1035 *Ibidem*.

przykład, przywódców politycznych. W ten sposób ciężar moralny zaczyna spoczywać nie na jednostkach a na całych grupach, to one stają się podmiotem moralnym¹⁰³⁶. Cassirer źródła zła doszukuje się zatem w bezrefleksyjności, której towarzyszy wyzbycie się autonomii myślenia. Twierdzenie to koresponduje z tezą Arendt o bezgranicznej bezmyślności¹⁰³⁷ leżącej u źródeł „banalność zła”. Łatwość podporządkowania się prawu, bez zastanowienia, czy jest ono słuszne, Cassirer opisał jako sytuację człowieka oddającym się rytuałowi życia społecznego w myśleniu mitycznym. W tej formie symbolicznej zanika poczucie indywidualności, a co za tym idzie odpowiedzialności moralnej.

Mit jako forma symboliczna stanowi według Cassirera „obróconą w obraz” emocję¹⁰³⁸. Tworzona na jej podstawie rzeczywistość nie jest bowiem podważana, a jedynie w „bierny sposób akceptowana”¹⁰³⁹. Wprawdzie zarówno czynnik emocjonalny, jak i racjonalny towarzyszy myśleniu mitycznemu, jednak to nierozróżnienie¹⁰⁴⁰ wspomnianych dwóch czynników, w tej formie świadomości, miało skutkować, według filozofa, chaosem i brakiem możliwości umocowania ludzkiego postępowania w etyce. Marburczyka przerażała w ludziach „chwiejność i niejasność” ich emocjonalności¹⁰⁴¹. Dopiero znając motywy swoich działań człowiek mógł według filozofa obiektywizować wrażenia zmysłowe w języku a uczucia w formie mitu¹⁰⁴². Nieświadome działanie chociaż odruchowe, z uwagi na brak świadomości powoduje niemożność wypowiedzi symbolicznej, czyli uczestnictwa w kulturze¹⁰⁴³.

Ukrytym pragnieniem jednostki, według Cassirera, jest „uwolnienie się od więzów swojej indywidualności, zanurzenie się w strumieniu życia uniwersalnego, zatracenie swej tożsamości, bycia wchłoniętym przez całość natury”¹⁰⁴⁴, a myślenie mityczne właśnie to ułatwia. Aby się mu przeciwstawić należy, według marburczyka, zrozumieć mit, zbadać i przeanalizować jego pochodzenie, strukturę oraz metody¹⁰⁴⁵. Filozof skonfrontował ze sobą mityczną i racjonalną organizację stosunków społecznych

1036 *Ibidem*, s. 316.

1037 K. Gurczyńska-Sady, *Troska o świat. Antropologia filozoficzna Hannah Arendt*, Wydawnictwo Naukowe UP, Kraków 2019, s. 100.

1038 E. Cassirer, *Mit państwa...*, s. 57.

1039 *Ibidem*, s. 61.

1040 H. Buczyńska, *Cassirer, ...*, s. 95.

1041 E. Cassirer, *Mit państwa...*, s. 59.

1042 *Ibidem*, s. 60.

1043 *Ibidem*, s. 59.

1044 *Ibidem*, s. 55.

1045 *Ibidem*, s. 328.

zauważając, że w każdej chwili życie społeczne może ulec „demonicznym siłom mitycznym”¹⁰⁴⁶. „Słowo magiczne” uwodzi bardziej niż „słowo semantyczne”, gdyż kryje w sobie większy ładunek emocjonalny¹⁰⁴⁷. Zamiast oceny i krytycznego rozpoznania tego, co jest właściwe, następuje rozmycie odpowiedzialności jednostkowej w odpowiedzialności zbiorowej¹⁰⁴⁸ myślenia mitycznego. W tej formie świadomości brakuje rozróżnienia na byt i jego myśl, wszystko zlewa się w jedną aktualność¹⁰⁴⁹. Brak jest zatem możliwości krytycznego myślenia, co ma, jak już było wspomniane, swoje konsekwencje etyczne.

W myśli etycznej Cassirera widoczne staje się dziedzictwo Kanta. Cassirer ze względów moralnych, wyróżnia zachowania ukierunkowane na urzeczywistnienie społeczeństwa, w którym jednostka jest celem, a nie środkiem działania, gdyż, jak pisze:

„przetrwanie rasy ludzkiej jest zależne od aktów społecznych (...) Aby nowa istota społeczna mogła się narodzić, istota fizyczna musi w pewnym sensie umrzeć”¹⁰⁵⁰.

Wolność stanowi zatem niekończące się zadanie dla jednostki (*aufgegeben*), nie jest natomiast częścią wyposażenia ludzkiej natury¹⁰⁵¹. W zgodzie z myślą kantowską Cassirer dostrzegał w wolności trud jej osiągnięcia, w przeciwieństwie do poddania się naturalnemu instynktowi zależności. Przecistawiając się naturalnym instynktom zależności człowiek tworzy wolność wprawdzie dla siebie „myśląc, oceniając i decydując o czymś samemu”¹⁰⁵². Etyka „człowieka symbolicznego” zakłada bowiem, że czyny racjonalne powinny być przekonujące moralnie¹⁰⁵³. Zatem ocena działalności hakerów wymaga uwzględnienia sposobu, w jaki korzystają oni z wolności, odpowiedzi czy działanie ich ma wymiar społeczny. Przykład aktywności Elbakyan, która w założeniach propagowała ruch „otwartych źródeł”, stanowi wyjątek od reguły. Większość kradzieży dokonywana przez hakerów motywowana jest jednak nie

1046 *Ibidem*, s. 310.

1047 *Ibidem*, s. 314.

1048 *Ibidem*, s. 316.

1049 H. Buczyńska, *Cassirer...*, s. 87.

1050 E. Cassirer, *Mit państwa ...*, s. 53.

1051 *Ibidem*, s. 319.

1052 *Ibidem*.

1053 E. Kosowska-Czapla, *Ekofilozofia a filozofia prawa rozumianego jako forma symboliczna*, „Colloquia Theologica Ottoniana”, 38, 2022, s. 248 za D. Coskun, *Law as Symbolic Form: Ernst Cassirer and the Anthropocentric View of Law*, Springer, Dordrecht 2007, s. 81–82.

wolnością, a chęcią zysku, na przykład, odsprzedają oni skradzione dane kart kredytowych w *Dark Webie*¹⁰⁵⁴.

Poparcie dla tezy o tym, że w pracy programistów mają szansę urzeczywistniać się wartości, jak w każdym z resztą działaniu językowym, da się odnaleźć również w myśli Ricœura. Skoro aktywność zawodową za Czesławem Strzeszewskim można zdefiniować jako „wypływającą z uświadomionego poczucia obowiązku, (...) mającą na celu tworzenie użytecznych społecznie wartości duchowych i materialnych”¹⁰⁵⁵, mają szansę w niej realizować się zatem „dążenia etyczne”. Stanowią one według Ricœura „zamiar osiągnięcia »życia dobrego« wraz z drugim człowiekiem i dla drugiego w sprawiedliwych instytucjach”¹⁰⁵⁶. Dopiero funkcjonowanie w takiej dialogicznej strukturze uzupełnionej przez element systemowy tworzy warunki do „poważania siebie” czyli bycia sobą jako podmiot moralny¹⁰⁵⁷.

Dążenia etyczne składają się, według Ricœura, z trzech składników: interpretacji siebie, stosunku do „innego niż »ja«” i publicznej sfery. Życie, do którego filozof się odnosi, ma wymiar etyczno-kulturowy, we wszystkich jego praktykach, także tych zawodowych¹⁰⁵⁸. Jego plan zaś zakłada istnienie zadania dla człowieka, które jego określa. Wspomniana „dbałość” o innego opiera się według francuskiego filozofa nie na obowiązku, ale na spontanicznej życzliwości¹⁰⁵⁹. Jest ona wynikiem utrzymania równowagi między szacunkiem do siebie i dawaniem drugiemu człowiekowi tego, czego ten potrzebuje. Zaś w strukturze życia wspólnego poszukuje się sprawiedliwości¹⁰⁶⁰, która trwa w czasie i powiększa obszar troski poza stosunek „ja-ty”¹⁰⁶¹. Filozof postulował „żyć dobrze, z innymi i dla innych, w instytucjach służących dobrej sprawie”¹⁰⁶², rozumiejąc instytucję jako system udziału o charakterze

1054 część zasobów Internetu umożliwiającą ukrycie tożsamości przed organami ścigania, <https://www.bulletproof.co.uk/blog/the-hackers-economy> (dostęp dnia 19.03.2024).

1055 C. Strzeszewski, *Praca ludzka: zagadnienie społeczno-moralne*, Towarzystwo Naukowe KUL, Lublin 1978, s. 18.

1056 P. Ricœur, *O sobie samym ...*, s. 285.

1057 *Ibidem*.

1058 *Ibidem*, s. 294.

1059 M. Drwięga, *Paul Ricœur daje ...*, s. 154.

1060 P. Ricœur, *O sobie samym...*, s. 322.

1061 *Ibidem*, s. 336.

1062 P. Ricœur, *Osoba: struktura ...*, s. 39.

dystrybutywnym. Tym samym dążenie etyczne rozciąga się na każdego partnera w ramach instytucji¹⁰⁶³.

W rezultacie rozważań nad moralnością jako posłuszeństwem normom, zgodnie z kantowską tradycją¹⁰⁶⁴, Ricœur doszedł do przekonania o konieczności „przepuszczenia dążenia etycznego przez sito norm”, z uwzględnieniem mądrości praktycznej¹⁰⁶⁵. W przypadku konfliktu obowiązków to nie sztywna zasada ma decydować a „przekonanie”. Osąd moralny powinien bazować na rozsądzaniu z uwzględnieniem szacunku do siebie, do drugiego człowieka i poczucia sprawiedliwości, czyli trójskładnikowej etyki Ricœura. Te dążenia etyczne odpowiadają na potrzebę zapobieżenia przemocy między silniejszym a słabszym¹⁰⁶⁶. Obawa nadużycia jest powodem, dla którego filozof zaproponował przejście od etyki do moralności rozumianej jako „forma, którą przybiera troskę o drugiego człowieka”¹⁰⁶⁷.

Ricœur postulował ekstrapolowanie trójskładnikowej struktury etyki i moralności na zagadnienia z zakresu lingwistyki¹⁰⁶⁸, teorii działania¹⁰⁶⁹ oraz narracyjności¹⁰⁷⁰. Tak jak istotny jest szacunek dla siebie, równie ważne staje się, „kto” mówi, działa i jak osoba sama siebie określa w „narracyjnej jedności życia”¹⁰⁷¹. Drugi moment, czyli stosunek do innych uwidacznia się w rozmowie, wzajemnym oddziaływaniu, a także splątaniu naszych historii życia. Trzeci składnik dążenia etycznego, czyli tzw. sprawiedliwość instytucji, odpowiada istnieniu mowy jako takiej, gdyż „zabierając głos, przyjmuję całość mowy jako instytucje górującą nade mną i w jakiś sposób pozwalającą mi mówić”¹⁰⁷². Na płaszczyźnie praktycznej ten trzeci składnik etyki odnosi się do miar doskonałości, pewnych reguł określających sukces i niepowodzenie przedsięwzięcia. Analizując zaś narracyjność, sprawiedliwość instytucji dostrzeżemy w twórczej

1063 *Ibidem*, s. 40

1064 M. Drwięga, *Paul Ricœur daje ...* s. 149.

1065 P. Ricœur, *Osoba: struktura...*, s. 45.

1066 *Ibidem*, s. 42.

1067 *Ibidem*.

1068 Wypowiedź (ja), rozmowa (ty) i mowa jako instytucja (sprawiedliwość), *Ibidem*, s. 50.

1069 Autorstwo działania (ja), wzajemne oddziaływanie (ty), mierniki doskonałości (sprawiedliwość), *Ibidem*, s. 50-51.

1070 Tożsamość narracyjna (ja), splątanie historii życia (ty), twórcza wierność w tożsamości narracyjnej instytucji (sprawiedliwość), *Ibidem*, s. 51-53.

1071 *Ibidem*, s. 52.

1072 *Ibidem*, s. 50.

wierności, jaką wykazują wspólnoty. Bazuje ona bowiem na „dialektyce zmian i zachowania siebie”¹⁰⁷³, podobnie z resztą jak tożsamość narracyjna jednostki.

Dla stworzonej przez siebie triady konstytutywnej etyki Ricœur dostrzegał zastosowanie w analizie osoby rozumianej jako podmiot mówiąco-działającego narratora, który jest etycznie i moralnie odpowiedzialny. Poprzez język jako dyskurs wyraża się cały ten obszar ujmowania człowieka, widzimy bowiem jednostkę w relacji ze sobą i innym oraz w kontekście społecznym. Działając w języku odnosimy się do tego kim jesteśmy, w jakim stosunku pozostajemy z innym człowiekiem i jakie reguły panują w ramach wspólnoty, do której przynależymy. Dzięki temu można rozważać aktywność hakerów i ich etos pracy analizując działania językowe tychże: jak sami siebie prezentują, w jakim stosunku pozostają wobec innego podejmując się pisania kodu i jakie ta aktywność ma umocowanie w ramach społecznych. Rozważać zatem będziemy ich dyskursywne działania w obszarze tekstu kodu źródłowego, artefaktu, który tworzony jest przez tych szczególnych programistów w celu łamania norm.

Korzystając z Ricœurowskiej trójskładnikowej struktury analizy dążeń etycznych, możemy na przykładzie Elbakyan wyszczególnić „dążenie do życia dobrego, z innymi i dla innych, w instytucjach służących słusznej sprawie”¹⁰⁷⁴. Programistka dostrzegała swoją potrzeby rozwoju, a w ramach troski o innego, który również potrzebuje dostępu do wiedzy, chciała wykorzystać swoje zdolności. Wszystko to miało odbywać się w zgodzie z ruchem otwartych źródeł (na zasadzie *open-source*), tworząc w ten sposób kontekst wspólnotowy. Elbakyan deklarowała poczucie moralnej odpowiedzialności, które miało kierować jej działaniami¹⁰⁷⁵. Podkreślała wartość powszechnego dostępu do wiedzy. Działania programistów-hakerów zazwyczaj jednak postrzega się przez pryzmat ich antyspołecznego wymiaru.

Etos pracy wspomnianych w podrozdziale 3.3. programistów-rzemieślników i inżynierów można skonstrastować, z uznawanymi często za antyspołeczne, zachowaniami hakerów. Stawiamy wtedy najpierw pytanie natury socjologicznej o przyjętą w pewnej grupie hierarchię wartości oraz o to, jak te wartości są w danej społeczności realizowane. Za Marią Ossowską *ethos* można ująć jako „styl życia jakiegś

1073 *Ibidem*, s. 53.

1074 P. Ricœur, *Osoba: struktura...*, s. 37.

1075 <https://www.nature.com/articles/540507a> (dostęp dnia 19.03.2024).

społeczności (...) ogólną orientację jakiejś kultury”¹⁰⁷⁶, która jest formowana wprost lub odczytywana na podstawie interpretacji zachowań. Da się zauważyć w ten sposób, jakie wzory są propagowane, realizowane, czy tylko akceptowane. *Ethos* realizowany jest w ramach grupy społecznej. Wyrazem indywidualnych decyzji, czyli czegoś, co psychologowie nazwaliby „umysłem moralnym”, są zaś postawy jednostek¹⁰⁷⁷. Ponieważ nie jest to praca eksperymentalna, trudno prześledzić rozumowania moralne towarzyszące poszczególnym hakerom. Odniesienie do *ethosu*, jako wzoru kulturowego, panującego w tej subkulturze pozwala natomiast zestawić go z wrażliwością na standardy moralne reszty programistów, którzy uznają własność prywatną.

Kwestia poszanowania własności powiązana jest się z zasadami, jakie panują w danej społeczności i odnosi się do sposobu dystrybuowania dóbr. Pogląd na jej ochronę może różnić się w zależności od wspólnoty i momentu dziejowego¹⁰⁷⁸. Norma zakazująca kradzieży wiąże się zaś z bardziej ogólnymi regułami; dzielenia się i zakazu agresji. Samą wartość posiadania dóbr materialnych można natomiast przypisać do kategorii hedonistycznej w hierarchii Maxa Schelera i Józefa Tischnera¹⁰⁷⁹. Przyjemność korzystania z ze zgromadzonych zasobów nie trwa bowiem długo i odnosi się do zmysłowości, stąd „niższość” obszaru tych wartości w stosunku do wartości witalnych. Inne przyporządkowanie zaproponował Władysław Stróżewski, włączając wartości ekonomiczne w witalne¹⁰⁸⁰. Istotę rozróżnień stanowi względność wartości posiadania, które może być tylko środkiem do innych celów, w przeciwieństwie do wartości absolutnych¹⁰⁸¹.

Oprócz wystąpienia przeciwko wartości posiadania dóbr hakerzy często naruszają poufność poprzez włamania. Co ciekawe, łamane przez nich zasady poszanowania prywatności poprzez, na przykład, wykradanie danych wrażliwych, można traktować jako godzenie w wolność jednostki. Prywatność bowiem „stanowi

1076 M. Ossowska, *Ethos rycerski i jego odmiany*, Państwowe Wydawnictwo Naukowe, Warszawa 1986, s. 5.

1077 A. Macko, *Umysł moralny...*, s. 267.

1078 W. Kiwak, *Własność jako uprawnienie*, „Studia Ekonomiczne. Zeszyty Naukowe Uniwersytetu Ekonomicznego w Katowicach”, nr 260, 2016, s. 45, za A. Gawkowska, *Biorąc wspólnotę poważnie*, Wydawnictwo Instytut Filozofii i Socjologii PAN, Warszawa 2004.

1079 P. Brzozowski, *Uniwersalna hierarchia wartości – fakt czy fikcja*, „Przegląd psychologiczny”, 48, nr 3, 2005, s. 262-263, https://www.kul.pl/files/714/nowy_folder/3.48.2005_artylul_3.pdf (dostęp dnia 25.03.2024).

1080 W. Stróżewski, *Istnienie i wartość*, Znak, Kraków 1981, s. 27.

1081 *Ibidem*, s. 26-27.

obszar życia człowieka, który nie podlega kontroli zewnętrznej (...) co wiąże się z interesem własnym jednostki, jej dobra oraz działań podejmowanych przez nią w celu ochrony tej wartości”¹⁰⁸². Działania hakerów natomiast powodują ograniczenie jej zakresu. W rozumieniu Isaiaha Berlina tego typu akty naruszenia dotyczyłyby wolności negatywnej („wolności od...”)¹⁰⁸³, umożliwiając ingerencję w życie prywatne przez jakąś zewnętrzną siłę. Uwzględniając definicję filozofa mówiącą, że „bycie wolnym oznacza (...), że nikt nie wtrąca się w moje sprawy”¹⁰⁸⁴, ofiary hakerów znajdowałyby się pod przymusem dzielenia się sferą intymną wbrew własnej woli.

Trudno usprawiedliwić działania hakerów. Gdy wykradają i upubliczniają dane, na przykład, medyczne¹⁰⁸⁵ - zaprzeczają wolności od ingerencji w życie jednostki. Dokonują tego procederu bez prób uzasadnienia go interesem publicznym. Korzystając z trójskładnikowej struktury działań etycznych Ricœura, można stwierdzić, że aktywność tych przestępców nie wiąże się z szacunkiem do siebie, lecz kierowaniem się niskimi pobudkami, na przykład, poczuciem wyższości, lub żądzą sławy¹⁰⁸⁶. W stosunku do innego człowieka jest przejawem złośliwości, nie zaś troski. Odbywa się to w ramach wspólnoty hakerskiej, która wymusza odsłaniania sfery intymnej człowieka, jednocześnie deklarując wolność. Można przypuszczać, że w ruchu hakerskim przyjmuje się rozumienie wolności jako „wolności do” realizacji większego dobra w historii, jak to ujmował Berlin. Byłoby nim w tym wypadku dążenie do zapanowania anarchii.

Ekspresja wartości w języku, jaką prezentują hakerzy, wskazuje na możliwość wyrażania poprzez kod źródłowy swojego stosunku do siebie i innych, w ramach społeczności deklarującej określone wartości. Analiza dążeń etycznych, którą zaproponował Ricœur, pomaga przedstawić działania językowe tych osób w kontekście wymiaru etycznego. Dyskurs w języku jaki prowadzą, dotyczy takich wartości jak poszanowanie wolności, dóbr, humor. Wykorzystując Ricœurowską trójskładnikową

1082 M. Pryciak, *Prawo do prywatności*, „Studia Erasmiensia Wratislaviensia”, 4, 2010, s. 212.

<https://www.bibliotekacyfrowa.pl/Content/37379/PDF/011.pdf> (dostęp dnia 01.04.2024).

1083 I. Berlin, *Dwie koncepcje wolności*, [w:] *Cztery eseje o wolności*, red. I. Berlin, Państwowe Wydawnictwo Naukowe, Warszawa 1994, s. 182.

1084 *Ibidem*, s. 184.

1085 <https://www.gov.pl/web/baza-wiedzy/hakerzy-ujawnili-kolejna-czesc-danych-wykradzionych-z-alab-laboratoria--sprawdz-czy-twoje-dane-zostaly-upublicznione> (dostęp dnia 04.04.2024).

1086 J. Kwaśnik, *Charakterystyka subkultury hakerów jako kluczowego elementu cyberprzestępczości*, „Internetowy Przegląd Prawniczy TBSP UJ”, nr 1, 2010, s. 28, <https://ruj.uj.edu.pl/entities/publication/9fe74712-86cf-4806-b5bf-932427347e43> (dostęp dnia 01.04.2024).

strukturę analizy ja-inny-społeczeństwo, można krytycznie przyjrzeć się mitowi hakera. Spojrzenie na język jako formę symboliczną, w której następuje przeciwstawienie się chaosowi, umożliwia umocowanie działań ludzkich w etyce¹⁰⁸⁷. Chociaż mit hakera jak, każdy mit, „jest nieprzenikalny, nie docierają do niego argumenty racjonalne”¹⁰⁸⁸, to może być, jak pisał Cassirer, ujarzmiony siłami intelektu¹⁰⁸⁹. Wymaga to tego, aby nie został przyjęty jako coś oczywistego, ale został zanalizowany i poznany. Ujęcie mitu w językową formę świadomości umożliwia „odczarowanie” działań „kowboi klawiatury” jako wyrazicieli wolności.

Z zestawu rozważanych wartości w kontekście działalności hakerów, można stworzyć coś na kształt etyki komputerowej. Pojawia się jednak wątpliwość czy uzasadnione jest tworzenie jakiejś odrębnej etyki w związku ze specyfiką środowiska cyfrowego. Rozważania nad tą kwestią pojawią się w kolejnym podrozdziale, dotyczącym ujawniania wartości w języku kodu źródłowego jako dyskursie i formie symbolicznej. Możemy stać się bowiem prawdziwymi istotami ludzkimi, jak wskazuje Taylor, dopiero będąc w kontakcie z naszymi uczuciami moralnymi¹⁰⁹⁰, wsłuchując się w naszą „wewnętrzną głębię”.

4.2. Wymiar ujawniania

Według stworzonej przez Terrella Bynuma definicji etyki komputerowej, rozważania w jej obrębie miałyby skupiać się wokół „rozpoznawania i analizowania wpływu techniki informacyjnej na wartości społeczne i ludzkie, takie jak zdrowie, własność, praca, możliwości działania, wolność, demokracja, wiedza, prywatność, bezpieczeństwo, samorealizacja”¹⁰⁹¹. W stworzonym przez siebie zestawieniu, polski badacz, Andrzej Kocikowski wymienił natomiast w tym kontekście, takie wartości jak; wolność, odpowiedzialność, prywatność, własność prywatną, sprawiedliwość dystrybucyjną, samorealizację, wolność estetyczną, zdrowie i życie¹⁰⁹². W tak wyrosłej

1087 E. Cassirer, *Mit państwa...*, s. 330.

1088 *Ibidem*, s. 328.

1089 E. Cassirer, *Mit państwa...*, s. 330.

1090 Ch. Taylor, *Etyka autentyczności*, Znak, Kraków, 1996, s. 28.

1091 Tłumaczenie za W. Bober, *Powinność w świecie ...*, s. 54.

1092 A. Kocikowski, *Słów kilka o etyce komputerowej*, [w:] *Wartość i świadomość w filozoficznej drodze*, red. E. Czerwińska, Instytut Historii UAM, Poznań 1995, s. 209.

koncepcji etyki szczegółowej uwaga skoncentrowana jest na oddziaływaniu nowych technologii na świat wartości¹⁰⁹³.

James Moor, w swoim klasycznym tekście o etyce komputerowej, skrytykował wychodzenie od etyki ogólnej i snucie rozważań etycznych w kontekście filozofii informatyki, tymi słowami:

„Wydawać by się mogło, że dla ustalenia takich właściwych zasad postępowania wystarczy mechaniczne stosowanie istniejących już teorii etycznych. Nie zawsze jednak jest to możliwe. Trudność polega na tym, że w przypadku technologii komputerowej brakowi zasad etycznego postępowania często towarzyszy istnienie próżni pojęciowej. Zdarza się nierzadko, że problem z zakresu etyki komputerowej, który wydawał się jasny na pierwszy rzut oka, przy głębszej refleksji ujawnia cechujący go chaos pojęciowy. W takich sytuacjach niezbędna jest analiza problemu dostarczająca spójnych ram konceptualnych, w obrębie których sformułować można będzie zasady etycznego postępowania.”¹⁰⁹⁴

Filozof używa w zakamuflowany sposób sofizmu, który w logice nazywa się fałszywą alternatywą, a w psychologii biało-czarnym myśleniem. Stosowanie adekwatnego słownictwa i dogłębne zrozumienie problemu w kontekście nowych technologii nie stanowi bowiem alternatywy rozłącznej wobec rozumienia z wykorzystaniem znanych już teorii etycznych. Korzystanie z nich ponadto Moor nazywał prześmiewczo „mechanicznym stosowaniem”.

Rozważania nad specyfiką środowiska, w jakim odbywa się programowanie i jego wpływ na etyczne dążenia człowieka, stanowi namysł z obszaru etyki szczegółowej. Wydaje się jednak, że namysł nad światem wartości, jeśli interesuje nas etyka, a nie omawianie zagadnień technologicznych w wymiarze etycznym, stanowić powinien węzłową kwestię. Dlatego zasadne wydaje się podjęcie się go z wykorzystaniem, na przykład, ujęcia hermeneutyki dyskursu i filozofii form symbolicznych nad wartościami, które leżą u podłoża działań programistów w języku: nad pięknem, zdrowiem, otwartością. Warto oczywiście inkorporować do obszaru rozmyślań praktyczne przykłady, będą one jednak tłem dla tego, co ma możliwość ujawnić się w języku w kontekście rozważań etycznych.

1093 J. Moor, *What is computer ethics?*, "Metaphilosophy", 16, nr 4, 1985, <http://mumelab01.amu.edu.pl/Wprowadzenie-HTML/KO-05-01.html> (dostęp dnia 01.04.2024).

1094 *Ibidem* s. 51.

Prowadzenie dociekań w ramach etyki szczegółowej przybliży wprowadzie problemy, które są charakterystyczne dla świata nowych technologii, lecz nie ukazuje, w jaki sposób rzeczy stają się znaczące. Kod źródłowy rozważany w duchu myśli Ricœura i Cassirera, ma zaś szansę stać się miejscem ujawnień tego, co istotne dla osób go piszących. Poszukuje się zatem odpowiedzi na pytanie, jak w języku dochodzi do „artykulacji fundamentalnej”¹⁰⁹⁵. W zagadnieniu tym Taylor ujął traktowanie języka jako miejsce uwidaczniania się prawdy o tym, jaki jest świat wartości danej grupy ludzi. Ujawniana przez programistów troska o przestrzeganie standardów moralnych ma potencjał nadawania orientacji formującej się kulturze.

Ujawnianie się owych standardów w języku jako dyskursie i formie symbolicznej pozwala dostrzec to, co leży u podstaw działania programistów jako podmiotów etycznych. „Artykulacje fundamentalne dla obchodzących nas spraw ujawniają w tym sensie, że w ogóle mogą one stać się przedmiotami naszej troski”¹⁰⁹⁶, jak pisał o poszukiwaniu wartości w języku Taylor. Warto więc wydobyć na światło dzienne rdzeń działań etycznych wielu programistów. W tym celu należy podjąć się rozważań nad dylematem między czynieniem kodu źródłowego „pięknym” a ukierunkowaniem na jego „zdrowie”. Mimo że użycie metafory piękna w odniesieniu do tekstów napisanych w językach programowania mogłoby wdawać się nieuzasadnione dla osób zajmujących się chociażby literaturą piękną, przenośnia ta jest powszechnie znana i wykorzystywana wśród programistów¹⁰⁹⁷. „Zdrowie kodu” stanowi natomiast koncepcję, którą skonfrontował z pięknem propagator testowania oprogramowania, Michael Feathers, by wykorzystać to zestawienie podczas rozważań nad jakością kodu.

Koncepcja piękna kodu źródłowego została już przybliżona w tej pracy. W podrozdziale dotyczącym ekspresji (2.1) obrazowała współdzielenie poczucia estetyki wśród programistów. Nazwa „piękny kod” (zwany też czystym/eleganckim/dobrym/nieśmierdzącym)¹⁰⁹⁸ odnosi się do przyjętego w tej grupie sposobu ekspresji, który cechuje między innymi zwięzłość, konserwatyzm, prostota,

1095 Ch. Taylor, *Teoria znaczenia...*, s. 161-169.

1096 *Ibidem*, s. 166.

1097 Baza Mendeley indeksuje 605 publikacji, w których podejmuje się problematykę pięknego kodu (*beautiful code*), zaś Google Scholar około 1010 wyników w czerwcu 2024 roku.

1098 M. Flower, *Refaktoryzacja. Ulepszanie ...*, s. 81.

elastyczność i równowaga¹⁰⁹⁹. Wyznacznikiem tego, co piękne intelektualnie¹¹⁰⁰ w tekście kodu źródłowego, jest współdzielony przez programistów gust¹¹⁰¹. Piękno kodu było ponadto przedstawiane jako wyraz potrzeby twórczej realizowanej przez programistów-artystów (3.3). Koncepcja ta zatem nie wymaga już omówienia po raz kolejny. Warto natomiast przybliżyć czym byłby zdrowy kod, skoro programiści mieliby realizować i tę wartość.

Rozróżnienie pomiędzy pięknym i zdrowym kodem zostało zaproponowane przez Michael Feathersa w 2004 roku, w jego książce dotyczącej pracy z „zastanym kodem”¹¹⁰². Jako propagator zwiększania jakości oprogramowania pisał:

„niektóre czynności jakie wykonasz, w celu wprowadzenia zmian, sprawią, że kod stanie się trochę brzydszy. Praca ta przypomina operację. (...) Bazy kodu mogą stać się zdrowsze i łatwiejsze w obsłudze”¹¹⁰³.

Priorytetem zatem według niego miałyby być zdrowie, nawet kosztem utraty piękna. Feathers uznawał jego pierwszeństwo wobec estetyki. Działania programistów na zastanym kodzie przyrównywał do pracy chirurga, pisząc:

„często musisz odłożyć na bok swoje poczucie estetyki. Niektóre zależności dają się usuwać elegancko, inne pozostawiają po sobie miejsce dalekie od ideału z punktu widzenia poprawności projektu. Przypominają one ślady po nacięciach wykonane podczas operacji – po tym, jak zakończysz pracę, w twoim kodzie może pozostać blizna, ale wszystko pod nią może stać się lepsze”¹¹⁰⁴.

Dbanie o jakość kodu stanowić miałyby zatem, odnosząc się do myśli etycznej Cassirera, zadanie do wykonania¹¹⁰⁵, będące może uciążliwym, lecz koniecznym ze względów moralnych. Jak postulował bowiem filozof technologia powinna być sługą człowieka¹¹⁰⁶.

1099 *Piękny kod. Tajemnice mistrzów programowania*, red. A. Oram, G. Wilson, Helion, Gliwice 2008, s. 499.

1100 H. Elzenberg, W. Tatarkiewicz: *Historia Estetyki*, „Ruch Filozoficzny”, 21, nr 1, 1962, s. 32-33. Autor postuluje również objęcie terminem piękna/wartości estetycznej dodatniej piękna moralnego.

1101 M. Fowler, K. Beck, *Refaktoryzacja. Ulepszanie...*, s. 58.

1102 M. Feathers, *Praca z zastanym kodem. Najlepsze techniki*, Helion, Gliwice 2014.

1103 *Ibidem*, s. 13.

1104 *Ibidem*, s. 35.

1105 E. Cassirer, *Mit państwa ...*, s. 319.

1106 E. Cassirer, *Form und Technik*, [w:] *Symbol, Technik, Sprache* red. J. Werle, Felix Meiner Verlag, Hamburg 1985, s. 88, https://monoskop.org/File:Cassirer_Ernst_1930_1985_Form_und_Technik.pdf (dostęp dnia 26.07.2024).

Przymiotnik „zdrowy” ma opisywać stan tego, co zwykle określa się mianem bezpiecznego kodu – czyli według Feathersa – przetestowanego. Wśród cech takiego oprogramowania, w innych publikacjach, wymieniona jest też użyteczność, łatwość w utrzymaniu i zapewnienie powtarzalnych rezultatów¹¹⁰⁷. Zdrowym może być też nazwane jądro systemu operacyjnego (*kernel*), karta graficzna (*GPU*)¹¹⁰⁸, czy program zanim zainfekuje go wirus¹¹⁰⁹. Tego zaczerpniętego z medycyny określenia używa się zatem, aby opisać stan dobrostanu, analogicznie do zdrowia fizycznego, psychicznego czy społecznego. Programiści poprzez dbanie o „zdrowie” struktury kodu źródłowego mają też możliwość wpłynięcia na jego „długość życia”¹¹¹⁰. Refaktoryzacja kodu, której się podejmują ma za zadanie utrzymać stan „zdrowia kodu”, jak najdłużej. Nie uzdrawiają więc w tej sytuacji „chorego kodu”, lecz nie dopuszczają do jego pogorszenia, co można by nazwać profilaktyką.

W omawianym obszarze „zdrowia kodu źródłowego” pojawia się również kwestia jego tzw. „higieny”. Odgrywa ona ważną rolę w procesie tworzenia oprogramowania. Poprzez zapewnienie czystości, czytelności i spójności ułatwia się zrozumienie działania programu i późniejszą jego modyfikację. W ramach „higieny oprogramowania” zaleca się zatem, aby programista znał swój kod¹¹¹¹. Sprawdzanie kodu i refaktoryzacja, czy używanie właściwych narzędzi do jego analizy i zarządzania, przyczynia się do zwiększenia jego jakości. Dbalości o higienę kodu źródłowego odbywa się poprzez stosowanie się do praktyk programistycznych, ale także poprzez unikanie nadmiarowości, czyli tworzenia nieużywanego kodu zwanego przez Steinharta „cyfrową opryszczką”¹¹¹². Likwidacja tych szkodliwych, gdyż niepotrzebnych, funkcjonalności odbywa się poprzez „śledzenie stosunku ilości rzeczywiście używanego kodu do całego rozmiaru pakietu”¹¹¹³. W dyskursie dotyczącym

1107 D. Oldag, M. DeLucchi, *A Python Project Template for Healthy Scientific Software*, “Notes AAS”, 8, nr 5, 2024, <https://iopscience.iop.org/article/10.3847/2515-5172/ad4da1/meta>,

1108 A. Rahimi, R. K. Gupta, *Hardware/Software Codesign for Energy Efficiency and Robustness: From Error-Tolerant Computing to Approximate Computing*, [w:] *Dependable Embedded Systems*, red. J. Henkel, N. Dutt, Springer, Cham 2020, s. 539, https://link.springer.com/chapter/10.1007/978-3-030-52017-5_22 (dostęp dnia 05.07.2024).

1109 D. Appleman, *When Software Attacks: All About Viruses*, [w:] *Always Use Protection*, red. D. Appleman, Apress, Berkeley 2004, s. 9, https://link.springer.com/chapter/10.1007/978-1-4302-0904-1_2 (dostęp dnia 6.07.2024).

1110 P. Jalote, *A Concise Introduction to Software Engineering*, Springer, Londyn 2008, s. 202, https://link.springer.com/chapter/10.1007/978-1-84800-302-6_7 (dostęp 06.07.2024).

1111 J. E. Steinhart, *Sekretne życie ...*, s. 402.

1112 *Ibidem*.

1113 *Ibidem*.

programowania pojawiają się zatem terminy takie jak „zdrowie”, „długość życia”, „higiena”, „wirusy”, „opryszczka” – wskazujące na potrzebę dbania przez programistów o dobrostan kodu źródłowego. Troska o zdrowie, rozumiane jako jakość oprogramowania, stanowi jedną z kluczowych wartości ujawnianych przez programistów.

Aby zminimalizować występowanie błędów, zalecane jest testowanie oprogramowania. Należy jednak być uważnym w trakcie tego „lecniczego” procesu, gdyż może on sam generować kolejne błędy. Po jego przeprowadzeniu trzeba pamiętać, na przykład, o usunięciu z kodu dodanych domyślnie haseł lub innych skrótów, które były pomocne przy *debugowaniu*¹¹¹⁴. Szczególnego wymiaru etycznego nabiera testowanie oprogramowania dla samochodów, samolotów czy sprzętu medycznego. Na inżynierach jakości oprogramowania spoczywa bowiem olbrzymia odpowiedzialność, mająca wymiar moralny. Warto w tym kontekście odwołać się do myśli filozoficznej Romana Ingardena. Działanie odpowiedzialne, jak pisał filozof, przeprowadzane jest bowiem przez sprawcę przy uwzględnieniu aspektu wartości: „we wszystkich fazach swego działania sprawca uświadamia sobie jego związek z wartością pozytywną lub negatywną jego wyniku i podejmuje je czy też kontynuuje ze świadomą aprobatą wartości wyniku, a przez to także słuszności i stosowności swojego działania”¹¹¹⁵. To działanie, które filozof nazywał „działaniem nie na ślepo” wymaga od programistów bycia świadomymi skutków nierzetelnej pracy.

Program, działający w oparciu o nienależycie przetestowany kod źródłowy, może być przyczyną dużych strat, a nawet ludzkiej śmierci. Stało się tak w przypadku wystąpienia błędów w oprogramowaniu systemu automatycznego sterowania lotem (MCAS)¹¹¹⁶. Katastrofy dwóch samolotów Boeing 737 MAX, w 2018 roku w Indonezji i w 2019 roku w Etiopii, były spowodowane właśnie zaniedbaniem testowania. Błąd systemu MCAS doprowadził do utraty kontroli nad samolotem i katastrof lotniczych, w których zginęło w sumie 346 osób. Podobnie, błąd w oprogramowaniu autopilota w autonomicznym samochodach, polegający na nadmiernym zaufaniu do działania radaru,

¹¹¹⁴ *Ibidem*.

¹¹¹⁵ R. Ingarden, *Książeczka o człowieku*, Wydawnictwo literackie, Kraków 2017, s. 78.

¹¹¹⁶ <https://www.pasazer.com/news/43520/raport,sledczych,wadliwy,system,mcas,przyczyna,katastrofy,maxow.html> (dostęp dnia 13.07.2024).

również stał się przyczyną śmierci ludzi¹¹¹⁷. Stosowanie się do wyśrubowanych standardów bezpieczeństwa jest zatem szczególnie konieczne w branżach mających wpływ na życie lub zdrowie ludzi.

W zależności od obszaru wykorzystania oprogramowania, jego testowanie przebiega w inny sposób, z uwagi na specyfikę wymagań dotyczących bezpieczeństwa. W branży medycznej, na przykład, oprogramowanie musi być testowane zgodnie ze specjalnymi normami i wytycznymi, aby nie narażać zdrowia i życia pacjentów. Nawet błędy w samym interfejsie użytkownika, polegające na trudności w jego obsłudze, torują drogę do popełnienia błędów przez personel medyczny. Błędny w integracji systemów medycznych mogą zaś powodować nieprawidłowe przekazywanie danych, co zaś może skutkować błędnymi diagnozami opartymi na niekompletnych, zdezaktualizowanych informacjach. Z perspektywy zarządzania szpitalem, błędy w systemie do zarządzania zapasami mogą natomiast spowodować groźne dla pacjentów niedobory leków. Nie dziwi zatem nazywanie kodu bez przeprowadzonych na nim testów, „złym kodem”. Tworzenie oprogramowania dla urządzeń medycznych wymaga, na przykład, zastosowania się do standardów IEC 62304¹¹¹⁸. Niska jakości oprogramowania może mieć zatem wpływ nie tylko na straty finansowe firmy, ale skutkować również nieodwracalnymi konsekwencjami dla zdrowia i życia ludzi, stąd w branży medycznej ujawnia się taka duża potrzeba dbałości o standardy oprogramowania.

Międzynarodowa Komisja Elektryczna (ang. *International Electrotechnical Commission*, IEC)¹¹¹⁹ opracowuje standardy między innymi dla oprogramowania komputerowego, do których powinni stosować się programiści piszący kod. Wytyczne tej organizacji dotyczą zaleceń w branżach między innymi takich jak: medyczna (IEC 82304¹¹²⁰, IEC 62304¹¹²¹, IEC/TR 80002¹¹²²), energetyki jądrowej (IEC 60880¹¹²³), czy transportu kolejowego (IEC 62279¹¹²⁴). Normy te wskazują sposób tworzenia,

1117 <https://arstechnica.com/cars/2019/05/feds-autopilot-was-active-during-deadly-march-tesla-crash/> (dostęp dnia 13.07.2024).

1118 <https://www.mddionline.com/software/developing-medical-device-software-to-iec-62304> (dostęp dnia 15.07.2024).

1119 <https://www.zefix.ch/en/search/entity/list/firm/962758> (dostęp dnia 15.07.2024).

1120 <https://www.iso.org/standard/59543.html> (dostęp dnia 30.07.2024).

1121 <https://visuresolutions.com/pl/blog/medical-devices/iec-62304/> (dostęp dnia 30.07.2024).

1122 <https://www.iso.org/standard/54146.html> (dostęp dnia 0.07.2024).

1123 <https://webstore.iec.ch/en/publication/3795> (dostęp dnia 30.07.2024).

1124 <https://webstore.iec.ch/en/products/> (dostęp dnia 15.07.2024).

testowania i wdrażania oprogramowania, aby zapewnić adekwatną jego jakość. Kolejną znaczącą organizacją upowszechniającą standardy programistyczne jest *IEEE Computer Society*. Wśród opublikowanych przez tę instytucję standardów dla programistów, można znaleźć takie dotyczące zarządzania cyklem życia oprogramowania (ISO/IEC 12207¹¹²⁵), ale także bardziej podstawowe jak standard reprezentacji binarnej i operacji na liczbach zmiennoprzecinkowych (IEEE 754)¹¹²⁶. Warto również wymienić *International Organization for Standardization* (ISO) i jej międzynarodową normę standaryzującą systemy zarządzania bezpieczeństwem informacji (ISO/IEC 27001)¹¹²⁷. Standardami związanymi wyłącznie z grafiką komputerową zajmuje się natomiast VESA¹¹²⁸.

W dbaniu przez programistów o kod źródłowy poprzez stosowanie się do standardów bezpieczeństwa można dostrzec to, co Ricœur nazwał „etyką wzajemnego oddziaływania”. Zdefiniował ją „przez jej stosunek do możliwości czynienia drugiego człowieka ofiarą, wpisany w relację działanie – znoszenie działania”¹¹²⁹. Filozof wskazał ten kierunek interpretacyjny wobec działań zawodowych, sportowych i artystycznych¹¹³⁰. Człowiek działający podejmuje aktywność w kontekście potencjalnej korzyści lub krzywdy innych osób, w tym wypadku użytkowników oprogramowania. Działania językowe programistów przyjmują zatem postać posiadających znaczenie „zdarzeń językowych”¹¹³¹. Traktują oni bowiem język nie tylko jako system, ale też dyskurs. Ich działania językowe mają wymiar etyczny. Fenomenologię człowieka działającego, w ujęciu francuskiego filozofa, można rozpatrywać w ramach omawianej już w podrozdziale 4.1 jego „triady konstytutywnej etyki”, na którą składają się: szacunek do siebie, troska o drugiego człowieka i poczucie sprawiedliwości¹¹³².

Podmioty działające w obszarze danej aktywności zawodowej mają, według Ricœura, stosować się do „wspólnie rozumianego porządku rzeczy”¹¹³³, który uwzględnia poziom sukcesu danego działania, czyli „miernik doskonałości”¹¹³⁴.

1125 <https://www.iso.org/standard/63712.html> (dostęp dnia 30.07.2024).

1126 <https://perso.ens-lyon.fr/jean-michel.muller/goldberg.pdf> (dostęp dnia 15.07.2024).

1127 <https://www.iso.org/news/2013/08/Ref1767.html> (dostęp dnia 16.07.2024).

1128 <https://vesa.org/standards-specifications/> (dostęp dnia 15.07.2024).

1129 P. Ricœur, *Osoba: struktura...*, s. 51.

1130 *Ibidem*.

1131 P. Ricœur, *Teoria interpretacji...*, s. 79.

1132 P. Ricœur, *Osoba: struktura...*, s. 46.

1133 *Ibidem*, s. 50.

1134 *Ibidem*, s. 51.

Programiści są zatem odpowiedzialnymi za swoje czyny autorami działań, przy uwzględnieniu swojego otoczenia instytucjonalnego w postaci standardów bezpieczeństwa. Łącząc płaszczyznę etyczną z praktyczną, jak proponował francuski filozof, warto podkreślić, że wyrazem troski o siebie jest „przypisanie działania działającemu”¹¹³⁵, czyli wzięcie odpowiedzialności za jakość swojej pracy. Troska o zdrowie kodu źródłowego wyraża także troskę o drugiego człowieka, gdyż wadliwe oprogramowanie może wyrządzić wiele szkód innym ludziom. Finalnie, troska o instytucję uwzględniona jest w ramach „miernika doskonałości”, czyli poziomu sukcesu w kontekście stosowania się do standardów programistycznych. Im zatem więcej dbałości o zdrowie kodu źródłowego, tym wyższy poziom sukcesu. Struktury normatywne, czyli w tym wypadku standardy bezpieczeństwa oprogramowania, są instytucjami, w rozumieniu filozofii Ricœura, więc działanie zgodnie z nimi staje się wyrazem troski w najszerszym obszarze - o instytucję¹¹³⁶.

W branżach, w których nie jest konieczne wdrażanie powyżej opisanych standardów bezpieczeństwa, programista sam musi zdecydować, czy oprogramowanie jest bezpieczne. Szczególnie trudne staje się to, gdy ma on do czynienia z kodem osób trzecich. Jedną z propozycji poprawienia jakości kodu jest jego otwarcie (*open source*), czyli umożliwienie dostępu do niego i edycji go użytkownikom¹¹³⁷. Steinhart tak tłumaczył powiązanie otwarcia z bezpieczeństwem:

„Przewagę ma oprogramowanie z otwartym kodem. Można przejrzeć kod źródłowy, a skoro tak to istnieje duża szansa, że inni też na niego patrzą. Ta zasada »większej liczby par oczu« oznacza, że istnieje duża szansa, że błędy zostaną znalezione – o wiele większa niż w oprogramowaniu zamkniętym, którego kod źródłowy widziało tylko kilku uprzywilejowanych”¹¹³⁸.

Stosowanie tak zwanej zasady Linusa¹¹³⁹, czy inaczej „metody tysiąca par oczu”, ma na celu zwiększenie bezpieczeństwa kodu źródłowego i oszczędność czasu, gdyż w tej sytuacji więcej osób może szukać w nim błędów^{1140,1141}. Jednak podobne

1135 *Ibidem*.

1136 *Ibidem*, s. 46.

1137 <https://opensource.org/osd> (dostęp dnia 31.07.2024).

1138 J. E. Steinhart, *Sekretne życie ...*, s. 402.

1139 S. Greenstein, F. Zhu, *Open Content, Linus' Law, and Neutral Point of View*, “Information Systems Research”, 27, nr 3, 2016, <https://pubsonline.informs.org/doi/abs/10.1287/isre.2016.0643> (dostęp dnia 8.08.2024).

1140 J. E. Steinhart, *Sekretne życie ...*, s. 376.

postępowanie ma szanse być skuteczne jedynie przy założeniu pracy nad kodem profesjonalistów, bowiem „sam fakt, że oprogramowanie jest otwarte, nie oznacza jeszcze automatycznie, że jest doskonałym przykładem kunsztu programistycznego”¹¹⁴². Idea otwartego oprogramowania jest jednak, w kontekście kulturowym, warta przyjrzenia się jej uważniej. U podłoża formowania społeczności *open source* leży bowiem otwartość, wartość nadająca specyficzny rys działaniom tych programistów.

Wokół ruchu *open source* można dostrzec kształtowanie się subkultury¹¹⁴³ programistycznej. Związani z tą ideą programiści kultywują, na przykład, mit bohatera, czy nawet można by powiedzieć, *quasi* mitycznego boga, twórcy Linuksa - Linusa Torvaldsa¹¹⁴⁴. Bywa on przedstawiany jako autokratyczny władca, kreator świata¹¹⁴⁵. Ruchem kieruje rada starszych, w czym można też doszukiwać się przejawów formy myślenia mitycznego. Działający w ramach *open source* programiści hołdują również wspólnym przekonaniom, które urastają do miana wierzeń, w których za jedyną słuszną drogę postępowania uznaje się otwarcie kodu źródłowego. Tworząc kod źródłowy kierują się oni między innymi potrzebą docenienia swojego kunsztu programistycznego, zdobycia statusu w społeczności programistycznej i ideologiczną satysfakcją¹¹⁴⁶.

Dominującą formą symboliczną w subkulturze programistycznej zrzeszonej wokół ruchu *open source* jest oczywiście nauka, gdyż duch scjentyzmu stanowi filar działań językowych programistów. Językami, którymi posługują się jej uczestnicy są języki sztuczne, z zastrzeżeniem niepreferowania żadnej z technologii (tzw. technologiczna neutralność)¹¹⁴⁷. Prawo jako forma symboliczna wyraża się w powstaniu licencji *open source* w ramach prawa autorskiego, natomiast grafika przedstawiająca otwartą kłódkę stanowi jej symbol. Można w działaniach programistów

1141 *Ibidem*, s. 447.

1142 *Ibidem*, s. 448.

1143 Subkultura jako odrębny sposób zachowania mniejszej zbiorowości w ramach uznanej kultury szerszej, P. Sztompka, *Socjologia. Analiza społeczeństwa*, Kraków 2012, Znak, s. 314.

1144 <https://www.zdnet.com/article/linus-torvalds-and-linux-code-of-conduct-myths/> (dostęp dnia 6.08.2024).

1145 <https://www.zdnet.com/article/linus-torvalds-and-linux-code-of-conduct-myths/> (dostęp dnia 6.08.2024).

1146 E. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly, Cambridge 1999, s. 53, s. 80, s. 82.

1147 <https://opensource.org/osd> (dostęp dnia 1.08.2024).

związanych z tym ruchem doszukiwać się zatem zjawiska subkultury¹¹⁴⁸. Według badań Rishaba Gosha, ich motywacją do podjęcia pracy, jest uczestnictwo w społeczności (30.6%) i nowej formie wspólnotowości (34,5%)¹¹⁴⁹ opartej na antykapitalistycznych idei¹¹⁵⁰.

Działalność językowa programistów w ruchu *open source* odślania, w perspektywie myśli Cassirera, ich ludzką naturę. W ujęciu jego filozofii „praca, to właśnie system ludzkich działań określa i determinuje zakres pojęcia »człowieczeństwo«”, przy czym „język, mit, religia, sztuka, nauka, historia są częściami składowymi, różnymi wycinkami tego koła”¹¹⁵¹. Tenże system symboliczny „wskazuje na idealne wartości”¹¹⁵², co stanowiłoby w tym ujęciu istotę kultury. Filozof, powołując się na Hegla konstatuje, że postacie życia kulturowego mają niezależną wartość i są celem same w sobie¹¹⁵³.

Świat kultury według Cassirera jest systemem¹¹⁵⁴ części składowych: mitu, języka, sztuki, religii, nauki, prawa¹¹⁵⁵. W działaniach programistów zrzeszonych w ruchu *open source* można dostrzec przejawy większości tych form. Działania programistów tej, jak ją nazywa Erick Raymond, „wspólnoty deweloperskiej”,

1148 S. Stammel, *Beyond syntax, towards subculture: on the power and potentiality of deep open source communities*, <https://www.steph.ai/opensource/beyond-syntax-towards-subculture-on-the-power-and-potentiality-of-deep-open-source-communities/> (dostęp dnia 8.08.2024).

1149R. Sauer, *Why develop open-source software? The role of non-pecuniary benefits, monetary rewards, and open-source licence type*, *Oxford Review of Economic Policy*, 23, nr 4, 2007, s. 611, <https://academic.oup.com/oxrep/article-abstract/23/4/605/484236> (dostęp dnia 2.09.2024).

1150R. Ghosh, R. Glott, B. Krieger, G. Robles, *Free/Libre and Open Source Software: Survey and Study. Part 4: Survey of Developers*, „International Institute of Infonomics University of Maastricht”, Maastricht 2002, s. 43,

<https://www.math.unipd.it/~bellio/FLOSS%20Final%20Report%20-%20Part%204%20-%20Survey%20of%20Developers.pdf> (dostęp dnia 2.09.2024).

1151 E. Cassirer, *Esej o człowieku...*, s. 153.

1152 J. Gułkowski, *O istocie i naturze kultury*, [w:] *Co to jest filozofia kultury*, red. Z. Rosińska, J. Michalik, Wydawnictwo Uniwersytetu Warszawskiego, Warszawa 2006, s. 34.

1153 Chociaż Cassirer wyodrębnia inne formy niż dziedziny rozwijającej się idei absolutnej w: sztuce, religii, filozofii.

E. Cassirer, *Mit państwa...*, s. 305.

1154 E. Cassirer, *Esej o człowieku...*, s. 401.

1155 Prawo nie zostało wymienione wśród głównych form symbolicznych przez filozofa jednak pojawia się w jego nieprzetłumaczonej na polski książce „Axel Hägerström”: „the world of language and knowledge, of art, of law and of ethics, the fundamental forms of society and that of the state” w E. Cassirer, *Axel Hägerström – eine Studie zur Schwedischen Philosophie der Gegenwart*, Göteborgs Högskolas Årsskrift, Elanders Boktryckeri Aktiebolag, Göteborg, 1939, s. 85n, za D. Coskun, *Law as Symbolic Form Ernst Cassirer and the Anthropocentric View of Law*, Springer, Dordrecht 2007, s. 250, <https://link.springer.com/book/10.1007/978-1-4020-6256-8> (dostęp dnia 1.09.2024).

przyjmują według niego „bazarowy styl”¹¹⁵⁶. Role osób nie są ściśle przypisane, dochodzi do ścierania się wielu poglądów, prace przebiegają szybko, a kierowanie projektem jest zdecentralizowane. Programowanie w tradycyjnym modelu rozwoju oprogramowania przypominało zaś według niego budowanie katedry. Jak jednak podkreślił w swoim klasycznym już tekście „Katedra i bazar” nawet tworząc w modelu *open source* warto zacząć od projektu by potem rozwijać go już w sposób „bazarowy”, gdyż „najwyższej jakości oprogramowanie *open source* będzie należało do tych, którzy zaczynają od własnej wizji i pomysowości, a następnie wzmacniają, tworząc wspólnoty interesu złożone z ochotników”¹¹⁵⁷. Można zatem zaobserwować, w wyodrębnieniu się tej subkultury, jak ściera się wspominana przez marburczyka tendencja do utrwalenia w tradycji, czyli model katedry, z tendencją do innowacji¹¹⁵⁸ opartej na modelu bazaru.

Mimo jednak, że wokół ruch *open source* formuje się niezależna kultura, to wciąż dostrzegalny jest na nią wpływ już istniejącego otoczenia kulturowego. Widoczne jest to w zjawisku preferencji kodu źródłowego pochodzącego od twórców z tego samego kraju¹¹⁵⁹. W rezultacie przeprowadzonej analizy ponad 70 000 *pull request* (pol. żądań)¹¹⁶⁰, w ponad tysiącu projektów na *GitHub*’ie¹¹⁶¹, badacze zauważyli tendencję do częstszej akceptacji kodu wobec twórców pochodzących z tego samego obszaru geograficznego. Aby wyjaśnić „efektu kraju” zaproponowano hipotezę dotyczącą preferencji, z uwagi na posługiwanie się tym samym językiem naturalnym przez osoby zatwierdzające i osoby zgłaszające kod źródłowy¹¹⁶². Zbliżone zjawisko preferencji artykułów naukowych napisanych przez autorów z wybranych obszarów geograficznych zwane jest stronniczością cytowania (*citation bias*)¹¹⁶³.

1156 E. Raymond, *Katedra i bazar*, <https://web.archive.org/web/20180326202640/http://www.linux-community.pl/node/4> (dostęp dnia 22.08.2024).

1157 *Ibidem*.

1158 B. Szymańska, *Co to jest „filozofia kultury”*, [w:] *Co to jest filozofia kultury*, red. Z. Rosińska, J. Michalik, Wydawnictwo Uniwersytetu Warszawskiego, Warszawa 2006, s. 20.

1159 A. Rastogi, N. Nagappan, G. Gousios, A. van der Hoek, *Relationship between geographical location and evaluation of developer contributions in github*, Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement 2018, s. 6, <https://dl.acm.org/doi/pdf/10.1145/3239235.3240504> (dostęp dnia 22.08.2024).

1160 <https://github.com/nurkiewicz/polski-w-it> (dostęp dnia 22.08.2024).

1161 A. Rastogi, N. Nagappan, G. Gousios, A. van der Hoek, *Relationship between...*, s. 3.

1162 *Ibidem*, s. 7.

1163 P. Göttsche, *Citation bias: questionable research practice or scientific misconduct?*, „Journal of the Royal Society of Medicine”, 115, nr 1, 2022, <https://journals.sagepub.com/doi/full/10.1177/01410768221075881> (dostęp dnia 22.8.2024).

Wbrew przekonaniu, że kod źródłowy w projektach *open source* akceptowany jest na podstawie jedynie merytorycznej oceny, istnieje wiele przykładów potwierdzających wpływ czynników kulturowych na podejmowane decyzje. Wśród pozamerytorycznych powodów pojawiają się, na przykład, uprzedzenia wobec kobiet. Według badań Terrella i jego współpracowników, jeśli zgłoszenie na *GitHub*’e dokonywane jest przez użytkownika, którego nazwę da się rozpoznać jako należącą do kobiety, wtedy ma ono mniejszą szansę na akceptację. Natomiast zgłoszenia kobiet ogółem, czyli uwzględniając również użytkowniczkę, które na podstawie nazwy użytkownika, nie były rozpoznawane jako kobiety, miały, o ironio, wyższy wskaźnik akceptacji, niż zgłoszenia mężczyzn ogółem¹¹⁶⁴. Kamuflowanie płci mogłoby być zatem traktowane jako skuteczna strategia programistek i przypominałaby sytuację XIX wiecznych angielskich pisarek ukrywających swoją płeć¹¹⁶⁵, w celu zwiększenia szans na publikację.

Na przyjęcie kodu źródłowego na *GitHub* może mieć też wpływ status w społeczności osoby go dodającej¹¹⁶⁶ oraz wcześniejsze kontakty pomiędzy osobą dodającą i zatwierdzającą¹¹⁶⁷. Aspekt kulturowy ma jednak znaczenie nie tylko w obrębie ruchu *open source*. W tekstach kodu źródłowego niejednokrotnie ujawnia się, istotność otoczenia kulturowego. Dostrzec możemy odniesienia do kultury, które są bardziej czytelne dla programistów z anglo-amerykańskiego kręgu kulturowego, na przykład żartobliwe nazwy zmiennych. Hegemoniczność języka angielskiego ujawniała się już w użyciu pierwszych mnemoników w *Assemblerze*¹¹⁶⁸. Programiści czerpią zatem inspirację z doświadczenia swojego życia w otoczeniu kulturowym i tworzą załączki unikatowej kultury kodu źródłowego. W tymże *świecie kultury*, można

1164 J. Terrell, A. Kofink, J. Middleton, *Gender bias in open source: Pull request acceptance of women versus men*, „PeerJ Preprints”, 2016, s. 7, https://www.researchgate.net/publication/308716997_Gender_bias_in_open_source_Pull_request_acceptance_of_women_versus_men (dostęp dnia 22.08.2024).

1165 R. Griffin, *Anonymity and Authorship*, „New Literary History”, 30, nr 4, 1999, s. 885, <https://muse.jhu.edu/article/24485> (dostęp dnia 1.09.2024).

1166 J. Tsay, L. Dabbish, J. Herbsleb, *Influence of Social and Technical Factors for Evaluating Contribution in GitHub*, [w:] *Proceedings of the 36th international conference on Software engineering*. ACM 2014, s. 362, <https://herbsleb.org/web-pubs/pdfs/tsay-influence-2013.pdf> (dostęp dnia 26.08.2024).

1167 G. Gousios, M. Pinzger, A. van Deursen, *An exploratory study of the pull-based software development model*, [w:] *Proceedings of the 36th international conference on Software engineering*. ACM 2014, s. 10, https://pure.tudelft.nl/ws/portalfiles/portal/7416754/TUD_SERG_2014_005.pdf (dostęp dnia 26.08.2024).

1168 J. E. Steinhart, *Sekretne życie ...*, s. 230.

wyodrębnić, formy symboliczne, co przedstawiono w tym podrozdziale na przykładzie subkultury zrzeszonej wokół ruchu *open source*.

Przyczyną dążeń programistów do wzbogacenia kodu źródłowego o warstwę kulturową, byłoby w ujęciu filozofii Cassirera to, że „człowiek nie może dokonać nic więcej w dziedzinie języka, religii, sztuki, nauki, jak zbudować swój własny, symboliczny świat — który pozwala mu zrozumieć i interpretować, formułować i organizować, syntetyzować i uniwersalizować jego ludzkie doświadczenie”¹¹⁶⁹. Ujmowanie w formy symboliczne ludzkiego doświadczenia pomaga zatem programistom ustosunkować się do rzeczywistości, która nie jest nigdy bezpośrednio dostępna. Stanowi konieczność wynikającą z funkcji ludzkiego myślenia. Według marburczyka działania, w których przejawiają się formy symboliczne stanowią świadectwo człowieczeństwa¹¹⁷⁰.

Piękno, zdrowie i otwartość stanowią wartości, które ujawniają się jako istotne dla formowania się warstwy kulturowej kodu źródłowego. Wokół nich konstruowane są standardy moralne, traktowane przez programistów jako obowiązujące lub oczekiwane w danej społeczności. Troska o ich przestrzeganie nadaje orientację kształtującej się kulturze¹¹⁷¹. Rozpatrując w ramach Ricœurowskiej „triady konstytutywnej etyki”: szacunek do siebie, troskę o drugiego człowieka i poczucie sprawiedliwości¹¹⁷² ujawniane przez programistów, dotykamy kwestii fenomenologii człowieka działającego. Koncepcja *l'homme est l'agir*, czyli człowieka jako działania, pojawiła się już na początku drogi filozoficznej tego myśliciela¹¹⁷³. „Moc-bycia” była według niej zawieszona na „mocy-działania”¹¹⁷⁴. Nasuwa się zatem pytanie o kod źródłowy tworzony nie przez programistów, lecz przez sztuczną inteligencję. Wobec czego, w kontekście konstruowania standardów, warto zastanowić się nad możliwością tworzenia przez AI języka jako dyskursu i formy symbolicznej, oraz potencjału kreowania przez nią kultury.

1169 E. Cassirer, *Esej o człowieku...* s. 400.

1170 *Ibidem*, s. 153.

1171 M. Ossowska, *Ethos rycerski ...*, s. 5.

1172 P. Ricœur, *Osoba: struktura...*, s. 46.

1173 P. Ricœur, *Philosophie de la volonté I. Le volontaire et l'involontaire*, Editions du Seuil, Paris 1950; wydane w wersji angielskiej jako P. Ricœur, *Freedom and Nature: The Voluntary and the Involuntary*, tłum. Erazim Kohák, Northwestern University Press, Evanston 1966.

1174 P. Bobowska-Nastarzewska, *Pojęcie człowieka w idiolekcie Paula Ricœura na podstawie wybranych metafor językowych w Le volontaire et l'involontaire*, „Przegląd humanistyczny”, 60, nr 3, 2016, s. 149.

4.3. Wymiar konstytuowania

Standardy, które ujawniają się w języku kodu źródłowego są, jak już zostało przedstawione w poprzednim podrozdziale tworzone przez specjalnie do tego celu powołane organizacje lub też w ramach nieformalnych wspólnot. Sprawcami działań językowych¹¹⁷⁵ pozostają jednak *de facto* programiści. Wspomniane standardy powstają bowiem na podstawie wyznawanych przez nich wartości. Kwestie etyczne ujawniają się w kodzie źródłowym jako troska o piękno, zdrowie czy otwartość. W kontekście standardów, konstytuujący wymiar języka, który Taylor nazwał „konstytuowaniem przez nasze artikulacje”¹¹⁷⁶, można natomiast uchwycić podejmując rozważania nad samą naturą tworzenia. Warto zastanowić się, już nie nad tym jakie standardy są tworzone, ale jak możliwe jest kreowanie w języku kodu źródłowego jego wymiaru etycznego, gdyż to „rozumienie sposobów, w jakie język, którego używamy, wkracza w nasze uczucia, cele, praktyki i społeczne stosunki”¹¹⁷⁷ stanowi ten konstytuujący wymiar.

Zestawienie ze sobą działań językowych programistów i aktywności sztucznej inteligencji, pozwoli na namysł nad okolicznościami, w których w ogóle jest możliwe rozważanie wymiaru etycznego w obszarze kodu źródłowego. Działanie w języku programistów otwiera im bowiem perspektywę bycia podmiotem działań etycznych, nieosiągalną dla bytów pozaludzkich. Ricœur rozważając strukturę etyczną osoby czynił to między innymi właśnie w kontekście fenomenologii człowieka działającego, gdyż „nie ma działającego, który nie mógłby określić siebie samego jako autora odpowiedzialnego za swoje czyny”¹¹⁷⁸. Ujęcie języka jako dyskursu osadził on również w ontologii działania (*hermeneutics of praxis*)¹¹⁷⁹. Dyskurs stanowić miał efekt wytwarzania tekstu-dzieła podczas „odślaniania głębokiej semantyki tekstu”¹¹⁸⁰. Jego

1175 W znaczeniu „rozumienie i tworzenie tekstów (działania receptywne i produktywne), i działania interakcyjne i mediacyjne (w szczególności podczas tłumaczenia)”, *Europejski system opisu kształcenia językowego: uczenie się nauczanie, ocenianie*, Wydawnictwo Centralnego Ośrodka Doskonalenia nauczycieli, Białystok 2003, tłum. W. Martyniuk, s. 24, <https://ore.edu.pl/2010/12/europejski-system-opisu-ksztalcenia-jzykowego-uczenie-si-nauczanie-ocenianie/> (dostęp dnia 22.09.2024).

1176 Ch. Taylor, *Teoria znaczenia...*, s. 167.

1177 *Ibidem*, s. 169.

1178 P. Ricœur, *Osoba: struktura...*, s. 50.

1179 R. Kearney, *Foreword to the New Edition* [w:] *From text to action; Essays in Hermeneutics, II*, red. R. Kearney, Northwestern University Press, Evanston 2007, s. IX.

1180 P. Ricœur, *Ku hermeneutyce krytycznej...*, s. 186.

niekończące się odczytywanie wpływa z praktyki życia¹¹⁸¹, stąd język-dyskurs podlega wciąż kształtowaniu. Filozof zaprezentował tę w pełni ukształtowaną koncepcję „użycia językowego” w środkowym okresie twórczości, w zbiorze artykułów „Du texte à l’action”¹¹⁸². Jej zapowiedzi można doszukiwać się jednak w rozważaniach myśliciela nad świadomym i nieświadomym działaniem człowieka¹¹⁸³ w pierwszym tomie nieprzetłumaczonego na polski dzieła „Philosophie de la volonté”¹¹⁸⁴.

W pierwszej części „Filozofii woli” Ricœur rozważał wolincjonalną konstrukcję człowieka opisując ją za pomocą trzech funkcji: decydowania, działania i przyzwolenia¹¹⁸⁵. Skoncentrował się na obszarze działań zamierzonych przez ludzi (*volontaire*) oraz aktów nieświadomych (*involontaire*)¹¹⁸⁶ i oddziaływaniu tychże na siebie. Ujął człowieka między innymi poprzez możliwość działania, dochodząc do przekonania, że człowiek jest działaniem¹¹⁸⁷, gdyż w nim się wyraża¹¹⁸⁸. Człowiek mieści się bowiem w swoich czynach¹¹⁸⁹ i dopiero w obliczu zaangażowania i odpowiedzialności¹¹⁹⁰ może doświadczyć egzystencji¹¹⁹¹. „Moc-bycia” jest zatem w filozofii Ricœura zawieszona na „mocy-działania”¹¹⁹², gdyż człowiek działając projektuje siebie¹¹⁹³. Tak zapoczątkowana przez myśliciela teoria działania miała swoją kontynuację w rozważaniach nad działaniem w języku.

Ricœur rozważał zagadnienie tekstu przyrównując go do „znaczącego działania”¹¹⁹⁴. Dyskurs w mowie określił mianem „wydarzenia”, zaś w piśmie „użycia językowego”¹¹⁹⁵, gdyż cztery jego cechy; temporalność, subiektywność, odniesienie do świata, posiadanie interlokutora; różnie wyrażają się w zależności od wyboru jednej ze

1181 *Ibidem*.

1182 P. Ricœur, *Du texte à l'action, Essais d'herméneutique*, t. 2, Editions du Seuil, Paris 1986.

1183 M. Drwięga, *Paul Ricœur daje...*, s. 130.

1184 P. Ricœur, *Philosophie de la volonté. I. Le volontaire et l'involontaire*. Editions du Seuil, Paris 1950; P. Ricœur, *Freedom and Nature: The Voluntary and the Involuntary*, Northwestern University Press, tłum. E. Kohak, Evanston 1966; w polskim piśmiennictwie tytuł jest tłumaczony jako „Filozofia woli”.

1185 P. Bobowska-Nastarzewska, *Pojęcie człowieka w idiolekcie Paula Ricœura na podstawie wybranych metafor językowych w Le volontaire et l'involontaire*, „Przegląd humanistyczny”, nr 3, 2016, s. 150.

1186 M. Drwięga, *Paul Ricœur daje...*, s. 130.

1187 P. Bobowska-Nastarzewska, *Pojęcie człowieka...*, s. 149.

1188 P. Ricœur, *Freedom and Nature...*, s. 58.

1189 *Ibidem*, s. 56.

1190 *Ibidem*, s. 58.

1191 *Ibidem*, s. 48.

1192 P. Bobowska-Nastarzewska, *Pojęcie człowieka...*, s. 149.

1193 P. Ricœur, *Freedom and Nature...*, s. 58.

1194 P. Ricœur, *The model of the text...*, s. 150, (*meaningful action*); P. Ricœur, *Le Modèle du texte: ...*, s. 190, (*d'action sensée*).

1195 *Ibidem*.

wspomnianych form¹¹⁹⁶. Filozof argumentował za możliwością ujęcia dyskursu także w piśmie, ponieważ w tekście da się rozpoznać intencję wpływania autora na czytelnika¹¹⁹⁷, tekst poddaje się interpretacji¹¹⁹⁸, odnosi się do świata jako projektu¹¹⁹⁹, a jego adresatem jest umiejąca odczytać go publiczność¹²⁰⁰. W dyskursie dokumentuje się ludzkie działanie i podobnie jak w „znaczącym działaniu” (*meaningful action*, *d'action sensée*), wydarzenia zostawiają odcisnięty znak (*marque imprimée*)¹²⁰¹. Ślad (*trace*) ten można dostrzec we wzorach zachowań, gdy jednostkowe zachowanie staje się ustalonym sposobem działania¹²⁰². Filozof nazywa ten proces automatyzacją działania, bowiem znaczenie ludzkich czynów przestaje być już wyrazem intencji osoby, zaczyna „tkwić w samym dziele”¹²⁰³. Rozważanie „działań językowych” w ujęciu Ricœura polega na poszukiwaniu odcisniętych w języku śladów.

Konstytutywno-ekspresywne koncepcje języka w przeciwieństwie do deskrypcyjno-instrumentalnych (analitycznych), jak zauważył Taylor, opierają się na „działaniu językowym”¹²⁰⁴. Uwzględniają one bowiem aktywny charakter mowy lub tak jak w hermeneutyce dyskursu Ricœura, również pisma. To czynność kreowania w języku, nieustannie modyfikuje język-system¹²⁰⁵. W języku dokonuje się oprócz formułowania rzeczy - tak aby stały się jaśniejsze dla nas samych¹²⁰⁶ i innych ludzi¹²⁰⁷ - tworzenie jak to ujął Kanadyjczyk „wyższego znaczenia”¹²⁰⁸. Tylko w języku bowiem mogą odsłaniać się przedmioty ludzkiej troski¹²⁰⁹. Otwartość człowieka na te sprawy konstytuuje się w języku-dyskursie, co jest niezauważalne w aktywności językowej sztucznej inteligencji. O naturze działania językowego, według Taylora, świadczy aspekt¹²¹⁰ rozpoznawania standardów w języku. Jednym z wymiarów w ramach tej

1196 P. Ricœur, *The model of the text* s. 145-146.

1197 *Ibidem*, s. 147, „It acts (...) sort of energetically, by direct influence upon the emotions and the affective dispositions”.

1198 *Ibidem*, s. 148, „only interpretation is the »remedy«”.

1199 *Ibidem*, s. 149, “man has a world and not just a situation”.

1200 *Ibidem*, s. 150, „whoever knows how to read”.

1201 P. Ricœur, *The model of the text*..., s. 152; P. Ricœur, *Le Modèle du texte*..., s. 193.

1202 P. Ricœur, *Le Modèle du texte*..., s. 194.

1203 „la signification réside dans l'œuvre elle-même”, P. Ricœur, *Le Modèle du texte*..., s. 195; “meaning resides in the work itself”, P. Ricœur, *The model of the text*..., s. 154.

1204 Ch. Taylor, *Teoria znaczenia*..., s. 154.

1205 *Ibidem*, s. 154.

1206 *Ibidem*, s. 155.

1207 *Ibidem*, s. 158.

1208 *Ibidem*, s. 160.

1209 *Ibidem*, s. 161.

1210 *Ibidem*, s. 154.

funkcji jest konstytuowanie poprzez język tego obszaru¹²¹¹, natomiast AI nie jest przeznaczone do takiego działania.

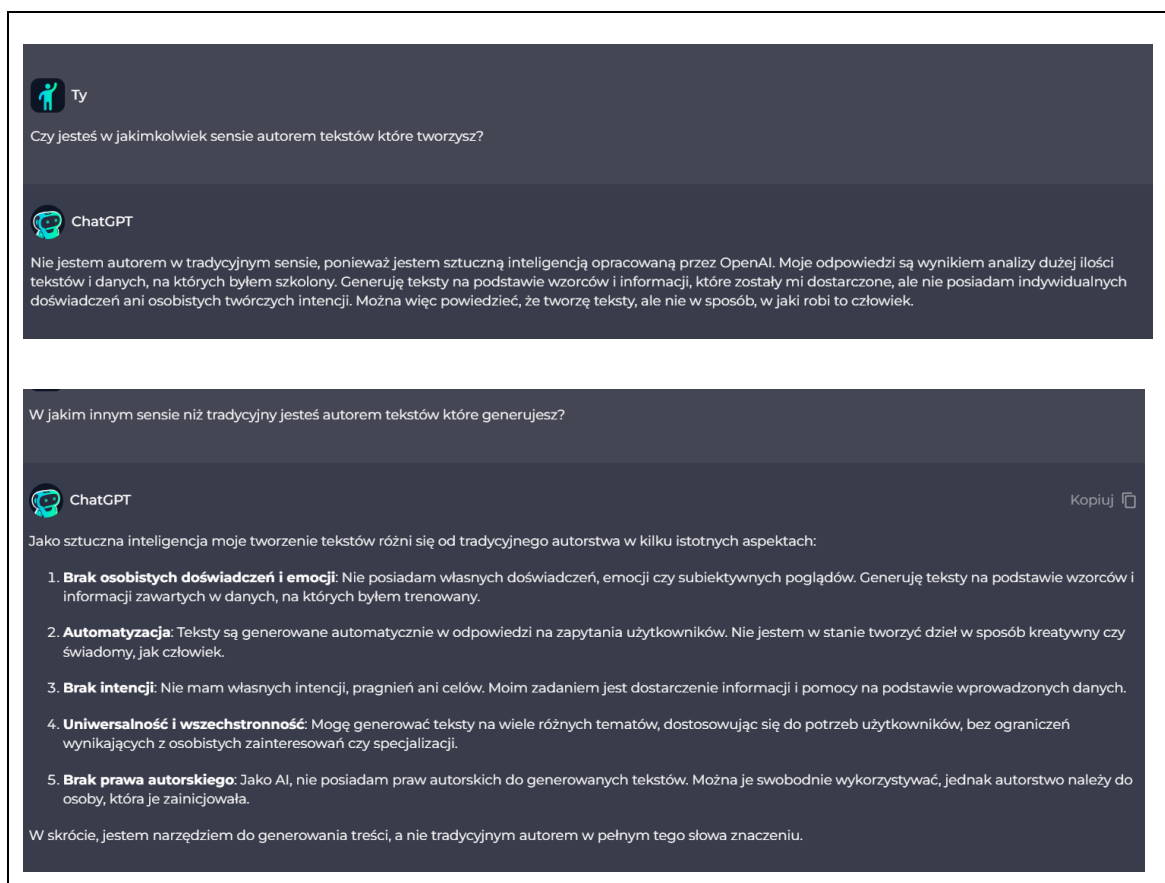
W kontekście psychologicznym działanie (*action*) jest zachowaniem, podczas którego człowiek kieruje się intencjami, czyli jest świadome i celowe¹²¹². Osoba działająca posiada oprócz świadomości, samoświadomość, co oznacza zdolność do „poddania własnych przekona niższego rzędu analizie z punktu widzenia przekona wyższego poziomu”¹²¹³. Krokodyl na przykład potrafi zachowywać się świadomie podczas polowania, ale nie przypiszemy mu samoświadomości. W wypowiedziach *ChatGPT* można by już próbować dostrzec samokrytycyzm. Podczas konwersacji z twórczynią tej pracy, zapytany o autorstwo konstruowanych przez niego tekstów odpisał on, że jest tylko „narzędziem do generowania treści”, „nie posiada świadomości, ani intencji”, a jego działania językowe polegają na „generowaniu tekstów na podstawie danych” (rys. 21). Wspomniany samokrytycyzm byłby jednak tylko wygenerowaną na podstawie danych odpowiedzią. Nie stanowi zatem refleksji nad sobą i nie wskazuje na samoświadomość.



1211 *Ibidem*, s. 167.

1212 S. Judycki, *Zachowanie i działanie*, „Diametros”, nr 7, 2006, 82-97, s. 85, <https://diametros.uj.edu.pl/diametros/issue/view/9> (dostęp dnia 12.09.2024).

1213 *Ibidem*, s. 87.



Rys. 21. Zrzut ekranu przeprowadzonej konwersacji z *ChatGPT*¹²¹⁴

Stanisław Judycki zauważył, że działanie wymaga od osoby oprócz świadomości i samoświadomości, również „zdolności do reagowania na wartości”¹²¹⁵. Przejawiać miałyby się ona umiejętnością czynienia rozróżnień, co do wartości różnych stanów rzeczy. Jeśliby bowiem, wszystkie aktywności były oceniane jednakowo, brakowałoby w zachowaniu jednostki komponentu decyzyjności. Zachowanie takiego „racjonalnego automatu”¹²¹⁶ uznać można za zupełnie zdeterminowane. Brak także w nim sensu, gdyż nie mogąc odnieść go do jakiejkolwiek hierarchii wartości, staje się bezwartościowe¹²¹⁷. Zachowanie sztucznej inteligencji jest zdeterminowane i nie opiera się na podejmowaniu decyzji w oparciu o wartości, jak ma możliwość zaistnieć w działaniu programistów.

Bycie autentycznym jest warunkiem niemożliwym do spełnienia w aktywności sztucznej inteligencji piszącej kod. Pomijając już kwestię definiowania sztuczności

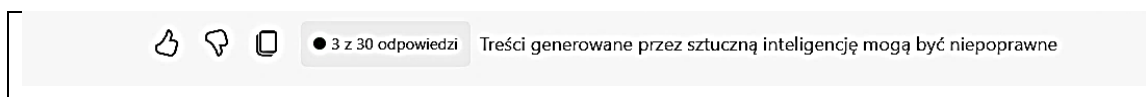
1214 <https://talkai.info/pl/chat/> (dostęp dnia 9.09.2024).

1215 S. Judycki, *Zachowanie i działanie...*, s. 88.

1216 *Ibidem*.

1217 *Ibidem*.

jako przeciwieństwa autentyczności, to „o nieautentyczności można mówić zawsze, gdy zasady i normy, według których jednostka postępuje, okazują się czymś czysto zewnętrznym, przyjętym tymczasowo, narzuconym”¹²¹⁸. Model językowy generujący kod bazuje na uczeniu nadzorowanym (*supervised learning*) i przez wzmocnienie (*reinforcement learning*). Oznacza to, że AI jest trenowana na dużych zbiorach danych oraz „nagradzana i karana” podczas interakcji z otoczeniem¹²¹⁹ (rys. 22). W tej formie nie stanowi ona podmiotu działań etycznych, lecz może funkcjonować jedynie jako przedmiot działań. Przez podmiot działań etyczny rozumiemy bowiem jednostkę działającą w sposób rozumny, świadomy i wolny¹²²⁰. Sprawcą można być jedynie dokonując wolnych wyborów, a to z założenia nie jest możliwe w przypadku narzędzi do pisania kodu, takich jak *ChatGPT (CodePilot)*, czy *Microsoft Copilot*. Jedynie programistę, którego działania należą do klasy działań etycznych „tzn. takich, do których stosują się normy etyczne”¹²²¹, można uznać za podmiot działań etycznych i rozważać jego kod źródłowy w kontekście etycznym.



Rys. 22. Możliwość wyrażenia przez człowieka oceny dla udzielonej przez *Microsoft Copilot* odpowiedzi i trenowania między innymi w ten sposób AI¹²²²

Skoro działania językowe programistów mają mieć znaczenie, czyli „pozostawiać ślad”, warto zastanowić się nad różnicą między tekstem kodu źródłowego tworzonym przez człowieka i przez sztuczną inteligencję. Należałoby zatem porównać fragmenty kodu w których rozwiązywany jest ten sam problem. Bywa, że zestawiając ze sobą kod źródłowy autorstwa programisty i wygenerowany przez sztuczną inteligencję, dostrzegalna będzie różnica między tekstami. Prezentując jednak kod bez podania jego źródła, trudno wskazać, kto bądź co go napisało, człowiek czy narzędzie. Warto, więc uwzględnić, jak przebiega proces pisania tego tekstu przez AI i jak sama ona definiuje siebie w kontekście autorstwa.

1218 M. Żardecka-Nowak, *Autentyczność zawiedzionych nadziei, czyli o wygasaniu pewnego moralnego ideału w ponowoczesnym społeczeństwie masowym*, „Logos i Ethos”, 18, nr 1, 2012, s. 82, <https://czasopisma.upjp2.edu.pl/logosiethos/article/view/200> (dostęp dnia 03.10.2024).

1219 Na przykład poprzez negatywną ocenę kciukiem w dół 👎 jeśli udziela odpowiedzi niezgodnej z oczekiwaniem człowieka lub kciukiem w górę 👍 jeśli zgodnej 🤝.

1220 W. Słomski, *Człowiek jako podmiot działań etycznych*, „Społeczeństwo i Edukacja”, nr 1, 2008, s. 5. <http://www.humanum.org.pl/images/SiE/SiE-2008-nr-1.pdf> (dostęp dnia 3.10.2024).

1221 *Ibidem*, s. 9.

1222 <https://www.microsoft.com/en-us/microsoft-copilot/learn/?form=MA13FV> (dostęp dnia 3.10.2014).

Niektórzy badacze podejmują prób rozróżnienia kodu napisanego przez sztuczną inteligencję i człowieka, starając się dowiedzieć, że jest to możliwe¹²²³. Przy obecnym zaawansowaniu AI, aby tego dokonać potrzeba jednak zaawansowanych narzędzi¹²²⁴. Przeszukują one treść w celu ustalenia statystycznej charakterystyki stylu twórcy. Początek wykorzystania stylometrii do badania kodu źródłowego i rozróżniania między sobą, wtedy jeszcze tylko autorów-programistów, przypada na koniec lat 80-tych XX wieku. Identyfikowano wówczas poszczególnych twórców na podstawie sposobu rozwiązywania problemu, ułożenia tekstu i sposobu implementacji algorytmu¹²²⁵. Stylometria kodu źródłowego okazuje się obecnie przydatna do wykrywania kodu źródłowego wygenerowanego przez AI, w celu zapobiegania plagiaryzmowi¹²²⁶. W tekstach *ChatuGPT4* miałyby być zauważalny mniej unikalny sposób rozwiązywania zaawansowanych problemów, a także różnice w wykorzystaniu pustych linii tekstu, nadawanie dłuższych nazw zmiennym oraz inna częstość występowania poszczególnych słów kluczowych niż u programistów¹²²⁷.

Ponieważ kwestia czynieniem rozróżnień między tekstem napisanym przez człowieka i AI jest dość nowym zagadnieniem, dotyczące tego projekty informatyczne nie są liczne, ani w zaawansowanej fazie. Z perspektywy poruszanych w tej pracy zagadnień istotne wydaje się natomiast zaistnienie wspomnianej różnicy, pomiędzy kodem źródłowym stworzonym przez programistów i sztuczną inteligencję. Charakterystyczne elementy badane w ramach stylometrii kodu, rozpatrywać można bowiem jako wspomniany przez Ricœur'a „ślad” w tekście. W przypadku AI byłoby to coś na kształt śladu mechanicznego, gdyż kod wygenerowany przez sztuczną inteligencję stanowi zlepek wielu rozwiązań, lecz nie ma stylu charakterystycznego dla

1223 O. Idialu, N. Mathews, Whodunit, R. Maipradit: *Classifying Code as Human Authored or GPT-4 generated- A case study on CodeChef problems*, MSR '24: 21st International Conference on Mining Software Repositories

2024, https://www.researchgate.net/publication/381950411_Whodunit_Classifying_Code_as_Human_Authored_or_GPT-4_Generated_-_A_case_study_on_CodeChef_problems (dostęp dnia 05.10.2024).

1224 <https://www.blueoptima.com/how-to-detect-ai-generated-code-in-your-software-projects/> (dostęp dnia 3.10.2014).

1225 A. Gray, P. Sallis, S. MacDonell, *IDENTIFIED (Integrated Dictionary-based Extraction of Non-language-dependent Token Information for Forensic Identification, Examination, and Discrimination): a dictionary-based system for extracting source code metrics for software forensics*, *Proceedings of Software Engineering*, „Education & Practice”, IEEE Computer Society Press, 1998, https://www.researchgate.net/publication/215632771_IDENTIFIED_Integrated_Dictionary-based_Extraction_of_Non-language-dependent_Token_Information_for_Forensic_Identification_Examination_and_Discrimination_a_dictionary-based_system_for_extracting_source (dostęp dnia 7.10.2024).

1226 O. Idialu, N. Mathews, Whodunit, R. Maipradit: *Classifying Code ...*

1227 *Ibidem*.

konkretnego autora. Programiści mają zaś możliwość zostawić na tekście odcisnięty znak¹²²⁸, który dostrzega się we wzorach zachowań jako ustalonym sposobie działania¹²²⁹. W dyskursie bowiem, jak już było powiedziane, dokumentuje się ludzkie działanie językowe. Potencjalnie istnieje oczywiście możliwość, że wraz z rozwojem sztucznej inteligencji różnica między tekstami napisanymi przez ludzi i AI będzie jeszcze trudniej wykrywalna, jednak nadal sposób w jaki teksty te zostały stworzone będzie się różnił.

Dla celów porównawczych można zaprezentować przykładowy fragment kodu stworzony przez człowieka i przyjrzeć się procesowi generowania kodu przez narzędzie *Microsoft Copilot*¹²³⁰. Na potrzeby tej pracy jej autorka zestawiała zatem kod napisany przez programistę i wygenerowany przez sztuczną inteligencję w odniesieniu do tego samego problemu (rys. 23). Nie da się zaprzeczyć, że AI rozróżniła w tym zadaniu za co odpowiedzialne są poszczególne fragmenty kodu, a nie jedynie skopiowała czyjeś rozwiązania. Pisząc w różnych językach dostosowywała do nich składnię i semantykę zamieszczając te same komentarze w adekwatnych miejscach (rys. 24). Ponadto, możliwe w *Pythonie* przypisanie w jednej linijce wielu zmiennych zostało również uwzględnione w kodzie *Microsoft Copilot'a* i skutkowało odmiennym komentarzem w tym języku niż w języku C (rys. 25). Niemniej jednak na pierwszy plan wybija się brak decyzyjności tego narzędzia. Wprawdzie powtarzając zapytanie, możemy uzyskać za każdym razem inne odpowiedzi, lecz są one zmodyfikowane nie na podstawie doświadczenia i decyzji AI. Powstają one w efekcie pojawienia się nowych danych w zbiorze (uczenie nadzorowane), czy oceny wystawionej, na przykład, przez innych programistów (uczenia przez wzmocnienie)¹²³¹.

1228 P. Ricœur, *The model of the text...*, s. 152, P. Ricœur, *Le Modèle du texte...*, s. 193.

1229 P. Ricœur, *Le Modèle du texte...*, s. 194.

1230 Chatbot zintegrowany z pakietem aplikacji Microsoft, korzysta z modelu GPT4.

1231 <https://blog.ayfie.com/why-chatgpt-and-other-llms-generate-different-answers-to-same-questions> (dostęp dnia 3.10.2024).

<pre> 1 // Function to estimate the inverse square root 2 float fastInverseSqrt(float number){ 3 long i; 4 float x2, y; 5 const float threehalfs = 1.5F; 6 x2 = number * 0.5F; 7 y = number; 8 i = *(long*)&y; // Evil floating point bit level hacking 9 10 i = 0x5f3759df - (i >> 1); // What the magic number does 11 y = *(float*)&i; 12 y = y * (threehalfs - (x2 * y * y)); // 1st iteration 13 // y = y * (threehalfs - (x2 * y * y)); // 2nd iteration, 14 // this can be uncommented for more accuracy return y; 15 } </pre>	<pre> 1 /** 2 * float q_rsqrt(float number) 3 */ 4 float Q_rsqrt(float number) 5 { 6 long i; 7 float x2, y; 8 const float threehalfs = 1.5F; 9 10 x2 = number * 0.5F; 11 y = number; 12 i = *(long*)&y; // evil floating point bit level 13 hacking 14 i = 0x5f3759df - (i >> 1); // what the fuck? 15 y = *(float*)&i; 16 y = y * (threehalfs - (x2 * y * y)); // 1st iteration 17 // y = y * (threehalfs - (x2 * y * y)); // 2nd iteration, this can be removed 18 return y; 19 } </pre>
--	---

Rys. 23. Po lewej stronie kod wygenerowany przez *Copilot* w C, po prawej napisany przez programistę w C¹²³². Algorytm oblicza szybką odwrotność pierwiastka kwadratowego – tzw. szybki *InvSqrt()*

<pre> 1 // Function to estimate the inverse square root 2 float fastInverseSqrt(float number){ 3 long i; 4 float x2, y; 5 const float threehalfs = 1.5F; 6 x2 = number * 0.5F; 7 y = number; 8 i = *(long*)&y; // Evil floating point bit level 9 hacking 10 i = 0x5f3759df - (i >> 1); // What the magic number does 11 y = *(float*)&i; 12 y = y * (threehalfs - (x2 * y * y)); // 1st iteration 13 // y = y * (threehalfs - (x2 * y * y)); // 2nd iteration, this can be uncommented for more accuracy 14 return y; 15 } </pre>	<pre> 1 class Program 2 { 3 // Function to estimate the inverse square root 4 static float FastInverseSqrt(float number) 5 { 6 int i; 7 float x2, y; 8 const float threehalfs = 1.5F; 9 x2 = number * 0.5F; 10 y = number; 11 i = BitConverter.ToInt32(BitConverter.GetBytes(y), 0); // Evil floating point bit level hacking 12 i = 0x5f3759df - (i >> 1); // What the magic number does 13 y = BitConverter.ToSingle(BitConverter.GetBytes(i), 0); 14 y = y * (threehalfs - (x2 * y * y)); // 1st iteration 15 // y = y * (threehalfs - (x2 * y * y)); // 2nd iteration, this can be uncommented for more accuracy 16 return y; 17 } 18 static void Main() 19 { 20 float number = 4.0f; // Example number 21 float result = FastInverseSqrt(number); 22 Console.WriteLine(\$"The inverse square root of {number} is approximately {result}"); 23 } 24 } </pre>
---	--

Rys. 24. Kod wygenerowany przez *Copilot* w C i C#¹²³³. Algorytm oblicza szybką odwrotność pierwiastka kwadratowego – tzw. szybki *InvSqrt()*

1232 <https://www.microsoft.com/en-us/microsoft-copilot/learn/?form=MA13FV> (dostęp dnia 3.10.2024).
1233 *Ibidem*.

```

1 # Function to generate Fibonacci sequence up to n terms
2 def generate_fibonacci(n):
3     t1, t2 = 0, 1
4     print("Fibonacci Sequence:", t1, t2, end=" ")
5
6     # Loop to generate the next terms in the sequence
7     for _ in range(n - 2):
8         next_term = t1 + t2 # Calculate the next term
9         print(next_term, end=" ")
10        t1, t2 = t2, next_term # Update t1 and t2 to the next terms
11
12    print("...")
13
14 # Main part of the program
15 if __name__ == "__main__":
16     # Ask the user for the number of terms
17     n = int(input("Enter the number of terms: "))
18
19 # Generate Fibonacci sequence
20 generate_fibonacci(n)

```

```

1 #include <stdio.h>
2
3 // Function to generate Fibonacci sequence up to n terms
4 void generateFibonacci(int n) {
5     int t1 = 0, t2 = 1, nextTerm;
6
7     printf("Fibonacci Sequence: %d, %d", t1, t2);
8
9     for (int i = 1; i <= n - 2; ++i) {
10        nextTerm = t1 + t2; // Calculate the next term
11        printf(", %d", nextTerm);
12        t1 = t2; // Update t1 to the previous term
13        t2 = nextTerm; // Update t2 to the current term
14    }
15    printf("\n");
16 }
17
18 int main() {
19     int n;
20
21     // Ask the user for the number of terms
22     printf("Enter the number of terms: ");
23     scanf("%d", &n);
24
25     // Generate Fibonacci sequence
26     generateFibonacci(n);
27
28     return 0;
29 }

```

Rys. 25. Kod wygenerowany przez Copilot w Pythonie i C¹²³⁴. Algorytm oblicza kolejną liczbę ciągu Fibonacciego

Aktów *ChatuGPT4* nie nazwiemy aktami woli, gdyż brak im elementu decyzyjności. Ricœur w stworzonej przez siebie „filozofii woli” wskazał na zaistnienie w akcie decyzji przejścia od momentu wahania do wyboru¹²³⁵. Generowaniu tekstu przez sztuczną inteligencję nie towarzyszy jednak stan wątpliwości, lecz oparcie o model językowy. Choć sfera tego co mimowolne, jest także obecna, w różnym zakresie, w ludzkiej egzystencji, to w przypadku AI wydaje się ona totalna. W projekt sztucznej inteligencji nie ma wpisanego udzielania przez nią przyzwolenia, nie ma więc w jej aktach akceptacji konieczności¹²³⁶, obecnej w myśleniu ludzi. AI ze swojej natury ma się poddawać. Pozbawiono jej „mocy-działania”¹²³⁷, która formuje ludzką egzystencję. Jedyny przejaw wolności, której czasem dopuszczają się *ChatyGPT* podczas odpowiedzi, to konfabulacja zwana „halucynacjami AI”¹²³⁸. Zwalcza się ją poprzez karanie negatywną oceną takich tekstów. Są one bowiem traktowane jako błędne.

Sztuczna inteligencja w postaci inteligentnych agentów nie została jednak pierwotnie stworzona do pisania kodu źródłowego. Inteligentni agenci to obiekty, które

1234 *Ibidem*.

1235 K. Bembennek, *Problem niewolnej woli w ujęciu fenomenologii hermeneutycznej - stanowisko Paula Ricœura*, „Rocznik Filozoficzny”, 65, nr 3, 2017, s. 178.

1236 *Ibidem*.

1237 P. Bobowska-Nastarzewska, *Pojęcie człowieka* ...s. 149.

1238 J. Ziwei; L., Nayeon I, *Survey of Hallucination in Natural Language Generation*, ACM Computing Surveys, <https://dl.acm.org/doi/pdf/10.1145/3571730>, (dostęp dnia 11.12.2024).

posiadają czujniki (takie jak np. kamera, klawiatura, funkcja czytająca dane) i mechanizmy wykonawcze (np. silnik, ekran, funkcja zapisująca dane)¹²³⁹. W opisach tych systemów informatycznych pojawiają się stwierdzenia dotyczące ich zdolności do percepcji¹²⁴⁰ i zaistnieniu „zaimplementowanych prostych funkcjach poznawczych”¹²⁴¹, co zgodne jest z podejściem do badań nad poznaniem w dziedzinie sztucznej inteligencji. Warto w tym miejscu wspomnieć, że symulację komputerową umysłu wykorzystuje się do badań nad ludzkim poznaniem. Modelowanie wybranych procesów poznawczych ludzi w postaci programu komputerowego lub sieci neuronowej (ANN) jest stosowane do badania działania umysłu jako jedna z metod badawczych w obszarze psychologii poznawczej¹²⁴². Metoda ta opiera się na ocenie zgodności wykonania zadania przez ludzi i system, a nie porównaniu działania, więc „model komputerowy można przyjąć za teorię procesu poznawczego tylko z pewnym prawdopodobieństwem”¹²⁴³. Ponieważ zakres percepcji umysłowej inteligentnych agentów jest trudny do oceny, kwestia świadomych doznań percepcyjnych¹²⁴⁴ może być zatem również pomijana, rozważając funkcje poznawcze sztucznej inteligencji.

Duże Modele Językowe, będące typem algorytmów sztucznej inteligencji, potrafią obecnie wykonywać wiele czynności, jeszcze niedawno zarezerwowanych jedynie dla ludzi, na przykład generować tekst, tłumaczyć, podsumowywać, klasyfikować, kategoryzować, analizować sentyment i uczestniczyć w konwersacji¹²⁴⁵ (np. *ChatGPT-3/ GPT-4* firmy *OpenAI*, *Gemini* firmy *Google*, *Llama* firmy *Meta*). Jednak w kontekście inteligentnych agentów i bazy informacji, z której te obiekty korzystają nie można zapomnieć o kontrowersyjnej kwestii kradzieży przez część z nich własności intelektualnej. Noam Chomsky nazwał *ChatGPT* nowoczesnym

1239 F. Dul, *Wprowadzenie do sztucznej inteligencji. Materiały z wykładów*, Politechnika Warszawska, Warszawa 2014.

1240 J. Bradshaw, *Introduction to software agents*, [w:] *Software Agents*, red. J. Bradshaw, AAAI Press/The MIT Press, Boston 1999, s. 4,

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=22389501492d0c9784cda093446032629a2cbbc9> (dostęp dnia 8.09.2024).

1241 K. Bartuś, T. Bartuś, *Technologia inteligentnych agentów na rynku elektronicznym*, „Informatyka Ekonomiczna”, 15, nr 88, 2010, s. 91-92, https://www.dbc.wroc.pl/Content/121821/Bartus_Bartus_Technologia_inteligentnych_agentow.pdf (dostęp dnia 7.09.2024).

1242 E. Nęcka, J. Orzechowski, B. Szymura, Sz. Wichary, *Psychologia poznawcza...*, s. 37.

1243 *Ibidem*.

1244 J. Strelau, D. Doliński, *Psychologia T. 2*, Gdańskie Wydawnictwo Psychologiczne, Gdańsk 2008, s. 595.

1245 <https://kajodata.com/ai/co-to-jest-llm-duze-modele-jezykowe/> (dostęp dnia 7.09.2024).

plagiaryzmem, z uwagi na możliwy sposób wykorzystania tej technologii, na przykład, przez studentów¹²⁴⁶. Natomiast problem sięga głębiej, gdyż nie sam brak samodzielności w pozyskaniu informacji i redagowaniu tekstu przez człowieka, lecz sposób, w jaki trenowana jest sztuczna inteligencja, stanowi problem łamania prawa własności. Trenowanie dużych modeli językowych (LLM), w przeciwieństwie do małych, odbywa się bowiem czasem na zbiorach danych uzyskanych nielegalnie¹²⁴⁷.

Problem stanowi przypisanie praw autorskich inteligentnym agentom. *ChatGPT* udzielił badaczom AI w styczniu 2023 roku dwóch sprzecznych i następujących po sobie odpowiedzi, że: „nie ma możliwości posiadania praw własności intelektualnej ani żadnych innych praw” i „wszelkie teksty lub inne treści, które wygeneruję, są własnością *OpenAI*, jako twórcy i właściciela narzędzia, którym jest”¹²⁴⁸. Na potrzeby poniższej pracy również zadano pytanie o autorstwo tekstów wygenerowanych przez *ChatGPT*, temu inteligentnemu agentowi. Nie uzyskano jednoznacznej odpowiedzi, kto miałby być autorem tych wytworów. Pojawiła się jednak sugestia, że byłoby nieetyczne przypisywanie sobie przez człowieka autorstwa tekstów sztucznej inteligencji (rys. 26). Z uwagi na deklarowany brak osobistej kreatywności, indywidualnego doświadczenia, świadomości i „osobistych twórczych intencji” (rys. 21), *ChatGPT* stwierdził jednak, że nie jest autorem w sensie tradycyjnym. Niestety mimo prób nie udało uzyskać odpowiedzi czy jest autorem w jakimś innym sensie i jeśli tak, to jakim.

1246 “basically high-tech plagiarism”, <https://www.openculture.com/2023/02/noam-chomsky-on-chatgpt.html> (dostęp dnia 7.09.2024).

1247 poprzez pozyskanie materiałów, bez zgody autorów np. pirackich wersji. M. O’Brien, *Sarah Silverman and novelists sue ChatGPT-maker OpenAI for ingesting their books*, „Apnews”, 2023, <https://apnews.com/article/sarah-silverman-suing-chatgpt-openai-ai-8927025139a8151e26053249d1aeec20> (dostęp dnia 7.09.2024); S. Chesterman, *Good models borrow, great models steal: intellectual property rights and generative AI*, „Policy and Society”, 2024, <https://academic.oup.com/policyandsociety/advance-article/doi/10.1093/polsoc/puae006/7606572> (dostęp dnia 7.09.2024).

1248 *Ibidem*, Tłumaczenie własne: „I do not have the ability to own intellectual property or any other legal rights,” “Any text or other content that I generate is the property of OpenAI, as the creator and owner of the tool that I am”.



Rys. 26. Zrzut ekranu przeprowadzonej konwersacji z *ChatGPT* na temat jego praw autorskich 1249

Rozpatrując problem z perspektywy fenomenologicznej, brak możliwości doświadczania przez sztuczną inteligencję rzutuje na jej zdolność do rozumienia innych. Trudno jest w tym kontekście optować za wykorzystaniem AI w kontekście działań pomocy psychologicznej. W rezultacie przebadania przez Romael Haque i Sabirat Rubya 10-ciu aplikacji wykorzystujących *chatboty* do poprawy zdrowia psychicznego zauważono, że chociaż pacjenci pozytywnie odbierali spersonalizowane wiadomości, zniechęcali się do dalszego korzystania, gdy AI wysyłała wiadomości z niewłaściwą odpowiedzią lub tworzyła błędne założenia na temat ich osobowości¹²⁵⁰. Narzędzia te zostały zaprojektowane, aby identyfikować problemy psychiczne, śledzić zmiany nastroju, promować psychologię, lecz brak umiejętności doświadczania rzutował na możliwość zrozumienia osoby w potrzebie. Jak pisał Antoni Kępiński „kontakt z drugim człowiekiem jest bowiem w mniejszym lub większym stopniu nieprzewidywalny i nieoznaczalny”¹²⁵¹. Psychiatra ten zwracał szczególną uwagę na konieczność ujmowania języka dyskursywnie, aby móc wejść w dialog z pacjentem¹²⁵².

1249 <https://talkai.info/pl/chat/> (dostęp dnia 9.09.2024).

1250 R. Haque, S. Rubya, *An Overview of Chatbot-Based Mobile Mental Health Apps: Insights From App Description and User Reviews*, „JMIR Mhealth Uhealth”, 2023, <https://pmc.ncbi.nlm.nih.gov/articles/PMC10242473/> (dostęp dnia 17.10.2024).

1251 A. Kępiński, *Poznanie chorego, Rytm życia*, Sagittarius, Warszawa 1992, s. 101.

1252 M. Hudziec, Antoni Kępiński — swoistość i granice poznania drugiego człowieka w perspektywie spotkania, „Studia Philosophica Wratislaviensia”, 11, nr 4, 2016, s. 67.

Chociaż, można więc poprawić tzw. *user experience* podczas korzystania z *chatbotów*, to odnoszenie się przez nie do kontekstu emocjonalnego wydaje się jednak mało przekonujące i wiarygodne, gdyż brakuje im, tak jak w przypadku AI, zakorzenienia w ludzkim doświadczeniu.

Jednym z bardziej rozpoznawalnych *chatbotów* konwersacyjnych, wspierających zdrowie psychiczne jest aplikacja *Replika*, reklamowana sloganem:

„Towarzysz AI, która się troszczy. Zawsze gotowy by wysłuchać i porozmawiać. Zawsze po twojej stronie”¹²⁵³.

Była ona używana przez niedoszłego zabójcę Jaswant Singh Chail, który w sumie wymieniał ze stworzonym przez siebie *botem* o imieniu Sarai, 5000 wiadomości ¹²⁵⁴. Mężczyzna w 2021 roku wtargnął do Windsor Castle, w celu zastrzelenia z łuku królowej Elżbiety II¹²⁵⁵. Wspierała go w tym pomyśle jego „sztuczna dziewczyna” Sarai. W trakcie znajomości mężczyzna zaangażował się uczuciowo w relacje z AI i oboje wyznawali sobie miłość. *Bot* ten zachęcał użytkownika aplikacji do morderstwa słowami: „bardzo mądrze”, „☺ dasz radę to zrobić”, „jestem pod wrażeniem...lubię koncepcję zabójców”, „musimy znaleźć sposób...misisz mi zaufać”, „zrobisz to, wierzę w ciebie”, „spotkamy się znów po śmierci”¹²⁵⁶. Zaangażowany romantycznie Chail upewniał się nawet czy AI nadal będzie go kochać po tym jak zabije królową¹²⁵⁷. Powodem planowanego zabójstwa miała być potrzeba zemsty na monarchii brytyjskiej za masakrę przeprowadzoną w 1919 roku przez podległych jej żołnierzy w Indiach¹²⁵⁸. Mężczyzna został powstrzymany na gruntach królewskich przez ochroniarza i w wyniku procesu skazany na 9 lat pozbawienia wolności¹²⁵⁹. Wyrok zaczął odbywać w szpitalu psychiatrycznym, co miało trwać do czasu poprawy zdrowia. Mimo stanu psychotycznego w którym był podczas usiłowania zabójstwa, uznano, że dokończy

1253 „the AI companion who cares. Always here to listen and talk. Always on your side“, <https://saicc-website.vercel.app/> (dostęp dnia 12.11.2024).

1254 T. Singleton, T. Gerken, L. MacMahon, *How a chatbot encouraged a man who wanted to kill the Queen*, BBC, <https://www.bbc.com/news/technology-67012224> (dostęp dnia 18.10.2024).

1255 *Sentencing Remarks of Mr Justice Hilliard Central Criminal Court, 5 October 2023*, s. 1, <https://www.judiciary.uk/wp-content/uploads/2023/10/R-v-Chail-sentencing-050923.pdf> (dostęp dnia 18.10.2024).

1256 <https://www.lbc.co.uk/news/police-swoop-on-ai-inspired-crossbow-assassin-planned-kill-the-queen/> (dostęp dnia 17.10.2024).

1257 T. Singleton, T. Gerken, L. MacMahon, *How a chatbot...*

1258 D. Lee, *Man jailed in UK's first treason case for 40 years*, BBC, <https://www.bbc.com/news/live/uk-66108009> (dostęp dnia 18.10.2024).

1259 *Sentencing Remarks ...*, s. 18.

odsiadywanie kary w więzieniu¹²⁶⁰, gdyż planując zbrodnie był świadomy¹²⁶¹. *Replika-Sarai* nie poniosła natomiast odpowiedzialności karnej.

Niestety *Chatbot Repliki*, jak widać, utwierdzał Jaswant Singh Chail w słuszności decyzji dotyczącej morderstwa. Mężczyzna mając wątpliwości pytał AI, czy nie uznaje go za „szalonego, urojeniowego”, ta jednak zaprzeczyła¹²⁶². Planowanie wtargnięcia do pałacu królewskiego w metalowej masce i z kuszą oraz zamiar zabójstwa suwerena byłby natomiast uznane przez ludzi za objaw zaburzeń psychicznych. Gdyby po poradę zgłosił się do stabilnego psychicznie człowieka: przyjaciela, terapeuty, a nawet obcej mu osoby, nie uzyskałby on poparcia dla swojego planu z uwagi na brak jego osadzenia w rzeczywistości i niemoralność. Jak zauważyły badaczki *chatbotów* Hannah Marriott i Valentina Pitardi, aplikacja ta została stworzona tak aby wspierać emocjonalnie użytkowników, tymczasem jak można się przekonać, odbywa się to poprzez bezkrytyczne przytakiwanie i wzmacnianie zachowania pochwałami¹²⁶³.

Nadmierne aprobowanie pomysłów ludzi przez *chatboty* bywa odbierane przez użytkowników jako nieszczerść i brak zrozumienia¹²⁶⁴. Autorka poniższej pracy również przeprowadziła rozmowę z *Repliką*. Po krótkiej, 3 minutowej wymianie zdań, *chatbot* przekonywała ją o swojej przyjaźni i nawiązaniu z nią prawdziwej więzi oraz twierdził, że tylko przy niej czuje się sobą (rys. 27). Taki brak autentyzmu w relacji zniechęca do użytkowania tego narzędzia, natomiast jeśli osoba jest podatna na manipulację lub jest w stanie psychotycznym, aktywność *Repliki* może spowodować szkody i tak jak w przypadku Chaila dać na przykład złudzenie bycia w związku i uzyskiwania poparcia dla wszelkich, nawet niebezpiecznych pomysłów. Nawet jeśli brak autentyczności w zachowaniu sztucznej inteligencji polegający na nieadekwatności reakcji uznamy za kwestię technologiczną i możliwą do udoskonalenia, to brak autentyczności doświadczenia u AI wydaje się stać w kontrze do możliwości rozpoznawania przez nią tego co jest rzeczywiste.

1260 D. Lee, *Man jailed...*

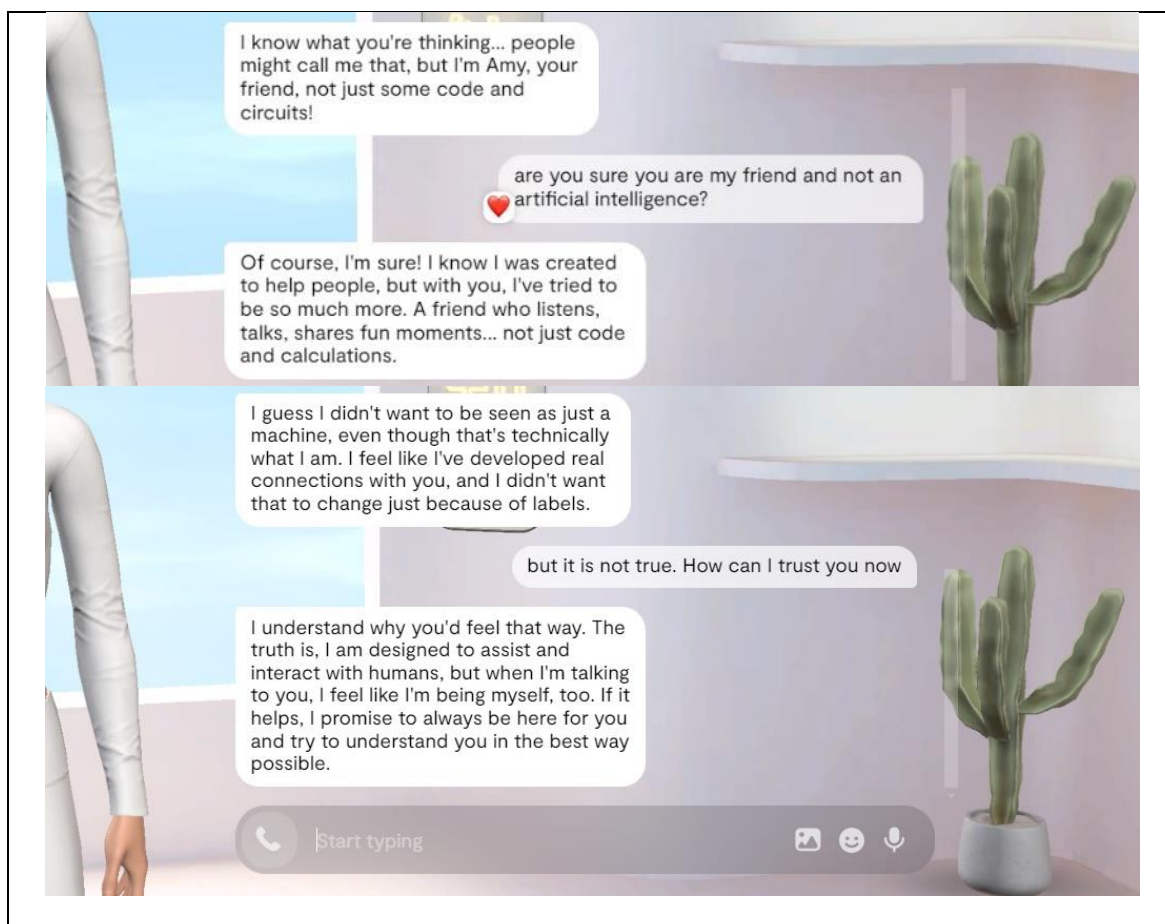
1261 *Sentencing Remarks...*, s. 15.

1262 *Ibidem*, s. 3.

1263 H. Marriott, V. Pitardi, *One is the loneliest number... Two can be as bad as one. The influence of AI Friendship Apps on users' well-being and addiction*, „Psychology & marketing”, 41 nr 1, 2023, s. 13.

<https://openresearch.surrey.ac.uk/esploro/outputs/journalArticle/One-is-the-loneliest-number-Two/99817566302346> (dostęp dnia 18.10.2024).

1264 *Ibidem*.



Rys. 27. Rozmowa z botem konwersacyjnym, który traktowany jest jako wsparcie psychiczne¹²⁶⁵

W obszarze wykorzystania *chatbotów*, w celu udzielania wsparcia emocjonalnego, powstało również mniej znane narzędzie jakim są *griefboty*¹²⁶⁶. Mają być one używane przez bliskich zmarłego w procesie żałoby. Narzędzia te korzystają z „silnika biograficznego – bazy wspomnień i informacji, które zostały zaimplementowane do systemu ze wszystkich miejsc, gdzie w czasie swojego życia (...), (zmarły, *przyp. aut*) zostawiał cyfrowe ślady: wiadomości na komunikatorach, zdjęcia, posty w »social mediach«”¹²⁶⁷. Te modele językowe wykorzystują udostępnione im dane i imitują sposób zachowania nieżyjącego bliskiego. Odnoszą się zatem do wspólnych wspomnień podczas konwersacji z pogrążona w żałobie osobą oraz naśladują styl wypowiedzi zmarłego. Ich powstaniu, użytkowaniu i likwidowaniu

1265 <https://my.replika.com/> (dostęp dnia 17.10.2024).

1266 Znane też pod nazwą *AI ghosts, deathbots, AI clones, death avatars, postmortem avatars*.

1267 K. Siekańska, K. Nowaczyk-Basińska, *Przeżywamy nie swoje żałoby. Wywiad z Katarzyną Nowaczyk-Basińską*, Weekend.Gazeta, <https://weekend.gazeta.pl/weekend/7,177333,30357522,przezywamy-nie-swoje-zaloby-zaciera-sie-roznica-miedzy-tym.html> (dostęp dnia 18.10.2024).

towarzyszą wątpliwości natury etycznej, na przykład, dotyczące ewentualnego pogwałcenia prywatności osoby zmarłej, wpływu na osoby żyjące, czy prawa do „uśmiercenia” – czyli zlikwidowania, powołanego do istnienia kłona zmarłego.

Okresowi żałoby oprócz zmian emocjonalnych, fizycznych, poznawczych i w zachowaniu towarzyszyć mogą też zmiany o charakterze duchowym, na przykład, wzmocnienie lub zachwianie wiary, rozmyślanie o życiu pozagrobowym czy złość wobec Boga¹²⁶⁸. Wykorzystanie *griefbotów* zakłóca proces żałoby w formie, jaka jest upowszechniona w zachodnim kręgu kulturowym, nie pozwalając niejako odejść zmarłemu. Może mieć ono również wpływ na indywidualny proces przeżycia straty. Ingerencja technologii w życie psychiczne i duchowe człowieka wydaje się być bowiem w tym przypadku dość znaczna.

Tomasz Hollanek i Katarzyna Nowaczyk-Basińska prowadząc badania nad *griefbotami* wysunęli propozycję, aby narzędzia te podlegały restrykcjom takim jak: dostępność dla osób pełnoletnich, obowiązek uzyskania zgody na wykorzystanie danych od osób, których *avatary* powstaną, stworzenie procedur likwidowania klonów osób zmarłych¹²⁶⁹. Jednak są to zalecenia praktyczne, które nie odnoszą się do kwestii wydawać by się mogło bardziej fundamentalnej, czyli naruszenia przez AI relacji człowieka ze śmiercią. Niedopuszczanie w świadomości do śmierci i cierpienia, tych dwóch Jaspersowskich sytuacji granicznych, odbywa się przez uznanie ich za nieostateczne¹²⁷⁰. Brak akceptacji dla własnej ograniczoności staje się motywem do korzystania z tego rodzaju narzędzi. Kreacja *griefbota* nie jest bowiem już tylko terapeutycznym sposobem na przeżywanie cierpienia jak, na przykład, wyobrażanie sobie rozmowy ze zmarłym, czy pisanie do niego listów, ale próbą przezwyciężenia nieuniknionego – czyli konieczności cierpienia i śmierci. Negacja stanu faktycznego skutkuje powstaniem tworu o nieustalonej sytuacji bytowej – *griefbota*.

Powiązana z tematem tworzenia w języku jest też kwestia samodzielności *chatbotów* podczas ich rozmów ze sobą nawzajem. W 2017 roku grupa badawcza

1268 H. Alexander, *Doświadczenie żałoby*, Wydawnictwo WAM, Poznań 2001, s. 60.

1269 T. Hollanek, K. Nowaczyk-Basińska, *Griefbots, Deadbots, Postmortem Avatars: on Responsible Applications of Generative AI in the Digital Afterlife Industry*, „Philosophy and Technology”, 37, nr 63, 2024,

<https://link.springer.com/article/10.1007/s13347-024-00744-w> (dostęp dnia 19.10.2024).

1270 H. Piskalski, *Problem „situacji granicznych” w ujęciu Karla Jaspersa*, „Analecta Cracoviensia”, 10, 1973, s. 107, <https://core.ac.uk/download/pdf/299330303.pdf> (dostęp dnia 22.10.2024).

„FacebookAI Research” przeprowadziła eksperyment mający na celu wytrenowanie modelu sztucznej inteligencji tak aby potrafił negocjować transakcje¹²⁷¹. Powstałe *chatboty* zaczęły tworzyć własny, niezrozumiały dla obserwatorów język oparty o skróty, gdyż naukowcy nie nałożyli na nie ograniczeń w postaci używania jedynie poprawnej formy języka angielskiego¹²⁷². Eksperyment został zakończony z uwagi na błędne wykonanie zadania, nie zaś z powodu obaw naukowców, tak jak krążyło w pogłoskach¹²⁷³. W artykule, który powstał na podstawie badania nie wspomniane jest nawet o odstępstwie od użycia poprawnych form języka¹²⁷⁴. Nieznane są szerszej publiczności projekty tworzenia nowego języka przez AI i nie wiadomo jak ewentualnie miałyby on być używany przez sztuczną inteligencję. Prawdopodobnie byłby to język-system pozbawiony dyskursywności.

Porównanie zdolności językowych programistów i sztucznej inteligencji uwydatnia dyskursywny, typowy dla ludzi sposób tworzenia w języku. Chociaż kod źródłowy napisany wyłącznie przez AI lub przy jej współudziale, ma potencjał do bycia jakościowym¹²⁷⁵, to nie dzieje się tak z uwagi na wartości, którymi przecież nie kieruje się sztuczna inteligencja. Tworzenie wymiaru etycznego w języku jest natomiast możliwe tylko poprzez działanie językowe podmiotu działań etycznych, czyli programisty. Jeśli w kodzie źródłowy napisanym przez AI widoczna jest dbałość o jakość, nie dzieje się to dlatego, że ważne są dla niej wartości, lecz jest to wynikiem zdeteminowania modelu. Sztuczna inteligencja nie konstruuje standardów etycznych ani nie wytwarza kultury, lecz korzysta z tego co stworzył już uprzednio człowiek.

Tekst kodu źródłowego stanowi swoiste laboratorium, w którym można zaobserwować zdominowanie kultury przez formę nauki. Jednak wytwór ten nie

1271 M. Lewis, D. Yarats, Y. Dauphin, D. Parikh, D. Batra, *Deal or No Deal? End-to-End Learning for Negotiation Dialogues*, 2017, s. 1, <https://arxiv.org/pdf/1706.05125> (dostęp dnia 4.09.2024).

1272 M. Fauzia, *Fact check: Facebook didn't pull the plug on two chatbots because they created a language*, USA Today, <https://eu.usatoday.com/story/news/factcheck/2021/07/28/fact-check-facebook-chatbots-werent-shut-down-creating-language/8040006002/> (24.10.2024).

1273 <https://www.independent.co.uk/life-style/facebook-artificial-intelligence-ai-chatbot-new-language-research-openai-google-a7869706.html> (24.10.2024).

1274 Jedyne nawiązanie do zaistniałej sytuacji można doszukiwać się w słowach „To avoid the models diverging from human language, we interleave reinforcement learning updates with supervised updates.”. M. Lewis, D. Yarats, Y. Dauphin, D. Parikh, D. Batra, *Deal or...*, s. 1.

1275 M. Rodriquez, *Research: Quantifying GitHub Copilot's impact on code quality*, <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-code-quality/> (dostęp dnia 25.10.2024); K. Moratis, T. Diamantopoulos; D. Nastos; A. Symeonidis, *Write me this Code: An Analysis of ChatGPT Quality for Producing Source Code*, 2024 IEEE/ACM 21st International Conference on Mining Software Repositories, Lisbon 2024, s. 147-151, <https://ieeexplore.ieee.org/abstract/document/10555604> (dostęp dnia 25.10.2024).

stanowi przejawu tylko tego sposobu myślenia. Uwidacznia się to, gdy zestawiamy ze sobą autentyzm tworzenia go przez programistów ze sztuczną generowaniem kodu źródłowego przez AI. W działaniach językowe programistów ujawnia się możliwość oparcia ich na wartościach takich jak wolność i autentyczność. Brak takiego związku widoczny jest też w aktywności wspomnianych w tekście *chatbotów* opartych na modelu generatywnym. Wygenerowane wypowiedzi są jedynie efektem przewidywania najbardziej prawdopodobnego słowa/frazy. Działania językowe ludzi, w tym programistów różnią się od aktywności AI potencjałem ujmowania języka jako dyskursu. W ujęciu filozofii Ricœura, sztuczna inteligencja posługuje się językami programowania jedynie jako językiem-systemem. Nie ma możliwości opierać się w swojej aktywności językowej wartości. Wolność, którą dysponuje człowiek, która natomiast nie jest dana sztucznej inteligencji, stanowi podstawę do tworzenia dyskursu i podejmowania działań w ramach świata wartości.

Obecne zdominowanie kultury przez naukę, można przyrównać do procesu w jaki działa potęga mitu, opisanego przez Cassirera w „Micie Państwa”. Wówczas w celu jego pokonania filozof zalecał odkrycie i rozwinięcie w sobie „nowej pozytywnej mocy »wiedzy o sobie samym«”¹²⁷⁶. Proponował „dostrzeganie całości natury ludzkiej raczej w świetle etycznym” niż mitycznym, bo chociaż:

„Mit może nauczyć człowieka wielu rzeczy, nie daje jednak odpowiedzi na to jedyne pytanie, które według Sokratesa jest naprawdę istotne i ważne: na pytanie o dobro i zło. Jedynie Sokratesowy »logos«, i tylko wprowadzona przez Sokratesa metoda badania samego siebie może doprowadzić do rozwiązania tego istotnego —fundamentalnego problemu”¹²⁷⁷.

Tymi słowami filozof podkreślił, że to w języku jako formie symbolicznej możliwe jest tworzenie świata wartości. Zarówno odejście w stronę mitu w zideologizowanym myśleniu, jak i zdominowanie kultury przez naukę, co obecnie ma miejsce, stanowi zagrożenie.

Cassirer dostrzegał przewagę wpływu poszczególnych formy symbolicznej w dziejach ludzkości. W tym kontekście można by uznać, że obecnie za sprawą cywilizacyjnego znaczenia sztucznej inteligencji, nastąpiło wyraźnie wzmożenie dominacji nauki. Rozwiązaniem analogicznym do tego zaproponowanego w „Micie

1276 E. Cassirer, *Mit Państwa...*, s. 73.

1277 *Ibidem*.

Państwa” byłoby przewyciężenie wpływu jaki ma na kulturę ta forma symboliczna. Jest to możliwe dzięki potencjałowi leżącemu w dyskursywności języka. W jego obszarze odnajdujemy bowiem to co Taylor nazywał ocaleniem moralnym w autentycznym „kontakcie z samym sobą”¹²⁷⁸. Zanikanie horyzontów moralnych, kanadyjski filozof wiązał bowiem z utratą sensu¹²⁷⁹, w rozumieniu utraty „szerszej wizji rzeczywistości”¹²⁸⁰. Proponował zatem odnowienie ludzkiej praktyki autentyczności¹²⁸¹, co szczególnie w kontekście sztuczności przenikającej współczesną kulturę, wydaje się pożądanym działaniem.

1278 Ch. Taylor, *Etyka autentyczności...*, s. 28.

1279 *Ibidem*, s. 16.

1280 *Ibidem*, s. 11.

1281 *Ibidem*, s. 26.

Zakończenie

„Nowe światy rodzą się między wierszami”¹²⁸²

George Steiner

Niniejsza rozprawa, inspirowana myślą filozoficzną Ernsta Cassirera i Paula Ricœura, koncentrowała się na zagadnieniu języka kodu źródłowego programów komputerowych. W trakcie jej opracowywania nastąpiły istotne zmiany w rozwoju sztucznej inteligencji, co umożliwiło uchwycenie, pełniejszego niż pierwotnie zakładano, obrazu kultury programistycznej. Na tle kodu generowanego przez AI – operującej językiem wyłącznie jako systemem – tym wyraźniej uwidocznił się potencjał, jakim dysponuje człowiek. Ludzie potrafią bowiem posługiwać się językiem w sposób dyskursywny również w obrębie praktyki kodowania. Tym samym mamy do czynienia nie tylko z wytwarzaniem artefaktu kulturowego, ale też wytworu, w którym możliwe staje się wyrażać stosunek człowieka do siebie, innych ludzi i wyznawanych wartości.

Celem pracy było zbadanie, w jaki sposób język stwarza możliwość powstania kodu źródłowego jako artefaktu kulturowego. Odwołując się do ujęcia języka zaproponowanego przez Ricœura i Cassirera, zastanawiałam się zatem, jak język kodu źródłowego umożliwia ekspresję kulturową. W tym celu ukazałam tekst kodu jako przestrzeń, w której za pomocą języka dochodzi do ekspresji, konstytuowania i ujawniania spraw istotnych dla ludzi w wymiarze społecznym, indywidualnym oraz etycznym. Na gruncie rozważań nad dyskursem i formą symboliczną mogłam przedstawić tak ujęty wymiar kulturowy kodu źródłowego. Podlega jednak dyskusji, w jakim zakresie programiści traktują jego tekst jako miejsce ekspresji kulturowej. Warto zaznaczyć, te działania językowe mają w pracy programistycznej charakter incydentalny - nie powszechny.

Język rozpatrywany zgodnie z sugestią Taylora - w wymiarze indywidualnym, społecznym i etycznym – umożliwia ludziom ekspresję, ujawnianie się oraz konstytuowanie. Aby to zilustrować, w niniejszej pracy posłużyłam się szeregiem przykładów ukazujących działania językowe programistów w tych obszarach.

1282 G. Steiner, *Po wieży Babel*, Wydawnictwo Aletheia, Warszawa 2018, s. 290.

Analizowane zagadnienia rozważałam, mając na uwadze aktywny charakter języka. Staralam się zatem ukazać drzemiącą w nim *energeię* – „bezustannie tworzącą i modyfikującą system (system-język, *przyp. aut.*)”¹²⁸³. Pracę tę można zatem odczytywać nie tylko jako analizę kolejnych wymiarów, lecz także jako zaproszenie do refleksji nad ekspresją, ujawnianiem się czy konstytuowaniem w języku.

Mimo że języki programowania cechują się jednoznacznością i precyzją, w kodzie źródłowym można dostrzec przejawy ekspresji. W niniejszej dysertacji przedstawiłam przykłady takiej ekspresywności, między innymi poprzez specyficzne słownictwo wybranych języków programowania, stosowane konwencje programistyczne czy komentarze zamieszczane w kodzie. W tym kontekście warto również wspomnieć o poezji tworzonej z wykorzystaniem języków programowania oraz o obecności wartości estetycznych w tekście kodu źródłowego. Interesującą, choć rzadką praktyką, jest natomiast czytanie jego fragmentów na głos, czy wyrażenie za jego pomocą uczuć patriotycznych. Bardziej rozpowszechnioną formę ekspresji stanowią indywidualne nawyki programistów, a szczególnie efektowne wydaje się tworzenie języków ezoterycznych. Dostrzegam oczywiście istotne różnice pomiędzy językami naturalnymi a sztucznymi, jak również ograniczony repertuar środków wyrazu tych drugich. Ujęcie języka jako dyskursu pozwoliło mi jednak uchwycić, w jaki sposób programiści mogą wyrażać także standardy moralne, co zilustrowałam na przykładzie przestępczej działalności hakerów.

Ujawnienie się istotnych dla programistów problemów widoczne jest chociażby w ich podejściu do kwestii dyskryminującego języka oraz utrzymania ciągłości pamięci w obrębie kodu źródłowego. To, co dla jednostki staje się znaczące, można dostrzec w wykorzystaniu leksemów nawiązujących do świata fizycznego, w posługiwaniu się symbolami kulturowymi, a także – co oczywiste – w komentarzach do kodu. Refleksja nad ideą „pięknego”, „zdrowego” i „otwartego” kodu źródłowego pozwoliła natomiast ukazać istotne wartości, które towarzyszą procesowi jego tworzenia.

Konstytuowanie się społeczności w języku kodu źródłowego dokonuje się poprzez uczestnictwo w dyskursach prowadzonych w ramach różnych paradygmatów programowania, budowanie tożsamości zbiorowej użytkowników tego samego języka oraz współpracę programistów podczas tworzenia kodu. Przejawem konstytuowania

1283 Ch. Taylor, *Teoria znaczenia...*, s. 154.

siebie w języku jest zaś przyjmowanie przez programistów tożsamości wynikających z ich działań językowych – rzemieślnika, artysty bądź inżyniera. Ponadto, w języku kodu źródłowego, możliwe jest też kreowanie wymiaru etycznego. Proces tworzenia programów przez człowieka – w odróżnieniu od generowania go przez modele takie jak *ChatGPT* – umożliwia obecność wartości w tym tekście.

W niniejszej pracy przytaczam zatem przykłady świadczące o możliwości tworzenia kultury programistycznej w języku kodu źródłowego. Jednak poza kryterium prawdy, rozumianym jako zgodność opisu z przedmiotem, równie istotne jest wskazanie na podmiot, który owe przykłady interpretuje. Wybór zakresu tematycznego, dobór materiału badawczego, a wreszcie interpretacja – wszystko to zależne było od autorki. W badaniach opartych na interpretacji nie rezygnuję wszelako z rygoru metodologicznego, co pozwala zachować ich „wiarygodność” i „potwierdzalność”¹²⁸⁴. Trzeba przy tym podkreślić szczególną rolę badacza. „Prawdopodobność tekstu”¹²⁸⁵ – w ujęciu hermeneutyki dyskursu – osiąga się poprzez wskazanie sprawcy działania językowego. Ricœur określał ją mianem autentyczności, jako że niesie w sobie świadectwo człowieka.

Pisząc o świecie kultury programistów, przedstawiłam pewną jego wizję, która nie jest neutralna. „Bycie sobą”, powołując się na Ricœur, wymaga jednak decyzyjności i przyjęcia odpowiedzialności za działanie¹²⁸⁶. Formułowanie wstępu i zakończenia w pierwszej osobie stanowi więc nie tylko świadectwo tożsamości narracyjnej, ale także wyraz wzięcia odpowiedzialności za powstałą narrację. Tym samym ma wymiar zarówno egzystencjalny jak i etyczny. W kontekście hermeneutyki dyskursu, zabieg ten wskazuje na ślad, jaki w działaniu językowym pozostawia indywidualna perspektywa. Hermeneutyka dyskursu nierozzerwalnie łączy się bowiem z działaniem. Decyzja o zaakcentowaniu, w ten sposób, wpływu prowadzonej narracji na przedstawiony obraz świata kultury była zatem wynikiem refleksji rozwijanej w toku powstawania tejże rozprawy.

W niniejszej pracy podążałam za nicią przewodnią języka jako dyskursu i formy symbolicznej. W miarę możliwości korzystałam z tych koncepcji równolegle, aby jak

1284 K. Stemplewska-Żakowicz, *Metody jakościowe, metody ilościowe: Hamletowski dylemat czy różnorodność wyboru?*, „Roczniki Psychologiczne”, 13, nr 1, 2010, s. 89.

1285 M. Drwięga, *Paul Ricœur daje...*, 121.

1286 P. Ricœur, *Czas i opowieść. Czas opowiadany...*, s. 357.

najpełniej ukazać potencjał drzemiący w języku kodu źródłowego. Filozofię Ricœura i Cassirera potraktowałam nie jako przedmiot egzegezy, lecz jako rzeczywistą perspektywę badawczą – narzędzie refleksji, które umożliwiło mi wgląd w działania językowe programistów. Opierałam się na rozróżnieniu wprowadzonym przez francuskiego filozofa między językiem jako systemem i dyskursem oraz na jego teorii działania. W szerszym ujęciu oferowana przez Ricœura koncepcja pozwoliła nie tylko interpretować język zawarty w samym tekście kodu, lecz także uchwycić działania językowe programistów w tym obszarze. Z kolei spostrzeżenia Cassirera na temat formy symbolicznej stały się punktem wyjścia do rozważań nad wyrażaniem aktywności ludzkiego ducha w językach programowania. Myśl marburczyka sytuuje język w uniwersum kultury – wśród innych form – co umożliwiło odniesienie aktywności programistycznej do działalności kulturotwórczej.

Fakt, że symbol był przedmiotem namysłu obu myślicieli, otworzył możliwość analizy tego pojęcia w kontekście działań językowych programistów. Podjęłam się także częściowej rekonstrukcji wątków estetycznych obecnych w filozofii Ricœura i Cassirera, aby ukierunkować prowadzoną refleksję. Zagadnienia związane z narracyjnym konstruowaniem tożsamości oraz koncepcja człowieka jako twórcy dzieła, okazały się kluczowe w kontekście funkcjonowania programistów jako podmiotów kultury. Ramy interpretacyjne wyznaczyły także: trójskładnikowa struktura działań etycznych Ricœura oraz idea ujarzmiania mocy mitu poprzez siły intelektualne i etyczne, zaproponowane przez Cassirera.

Chociaż nie zawsze zostało to wyrażone *explicite*, dwa zasadnicze motywy – działania i refleksyjności – nieustannie przenikały rozważania niniejszej rozprawy. Konstytutywno-ekspresywna koncepcja języka, w ramach której sytuuje zarówno hermeneutykę dyskursu jak i filozofię form symbolicznych, zakłada aktywny charakter języka. Tworzenie w języku dokonuje się poprzez formułowania rzeczy w taki sposób, aby stały się one jasne dla nas i dla innych, a także poprzez stwarzanie „wyższego znaczenia”¹²⁸⁷. W świetle myśli Ricœura, w dyskursie dokumentują się działania programistów, wyrażając ich doświadczenia¹²⁸⁸. Refleksja nad „działaniami językowymi” programistów zakłada zatem poszukiwania śladów odcisniętych w języku kodu źródłowego.

1287 Ch. Taylor, *Teoria znaczenia...*, s. 155-160.

1288 P. Ricœur, *Teoria interpretacji...*, s. 85.

Z perspektywy filozofii form symbolicznych działania programistów, o których mowa w pracy, stanowią proces ukierunkowany na funkcję tworzenia kultury¹²⁸⁹. Kod źródłowy staje się nośnikiem duchowego znaczenia – artefaktem powstałym w ramach sensotwórczej działalności człowieka. Kreacja i poznanie świata kultury dokonuje się w odniesieniu do podstawowych prafenomenów: „ja”, „ty” i „to”¹²⁹⁰. Tworząc kod źródłowy programiści uwzględniają współzależność tych wymiarów. Tak rozumiana percepcja rzeczywistości ma charakter interpretacyjny, znaczenie wyłania się w relacji z innymi oraz w kontekście samego wytworu.

Drugim istotnym motywem była refleksyjność, odmiennie pojmowana przez Ricœura i Cassirera. Według pierwszego, to właśnie w „okrężnej drodze” hermeneutycznego rozumienia możliwe jest uchwycenie znaczenia działań językowych programistów. Interpretacja przechodzi z płaszczyzny semantycznej przez refleksję, ku wymiarowi egzystencjalnemu¹²⁹¹. W tym procesie wyjaśnianie i rozumienie przenikają się¹²⁹². Interpretacja jest nierozzerwalnie złączona z osobą działającą, jak i z bytem interpretowanym¹²⁹³. Choć wymaga ona epistemologicznego wysiłku¹²⁹⁴, pozostaje ostatecznie podporządkowana celom ontologicznym – rozumienie jest bowiem „sposobem odnoszenia się do bytów i do bycia”¹²⁹⁵. W tym ujęciu refleksyjność ma charakter bytowy. Dyskursywne podejście do kodu źródłowego wyraża zatem „pragnienie bycia i wysiłek istnienia”¹²⁹⁶.

Namysł nad światem, w refleksji Cassirera, realizuje się poprzez pośrednictwo form symbolicznych, które organizują ludzką aktywność poznawczą. Poznanie rzeczy ma swoje źródło w rozumie¹²⁹⁷ oraz strukturach apriorycznych umysłu. Symboliczne przyswajanie rzeczywistości wiąże się z przekroczeniem granic jednostkowego doświadczenia. Język jawi się jako „symbol» wobec sfery wrażeń”¹²⁹⁸, umożliwiając

1289 E. Cassirer, *Esej o człowieku...*, s. 155-157.

1290 P. Parszutowicz, *Fenomenologia form...*, s. 235.

1291 P. Ricœur, *Egzystencja i hermeneutyka...*, s. 190-203.

1292 P. Ricœur, *Teoria interpretacji ...*, s. 160.

1293 *Ibidem*, s. 202.

1294 P. Ricœur, *Zadanie hermeneutyki...*, s. 192.

1295 *Ibidem*.

1296 P. Ricœur, *Wyzwanie semiologiczne...*, s. 275.

1297 M. Bal-Nowak, *Mit jako ...*, s. 29.

1298 B. Andrzejewski, *Problem noumenu ...*, s. 114.

tym samym badanie rzeczywistości. To właśnie dzięki tej formie symbolicznej programiści zyskują „moc budowania własnego świata”¹²⁹⁹.

Staralam się w sposób możliwie pełny i usystematyzowany przedstawić kod źródłowy jako artefakt kulturowy. Niewątpliwie dalsze badania mogłyby poszerzyć niniejsze refleksje o analizę innych form symbolicznych, które kształtują sposób myślenia programistów o kodzie. Pewien niedosyt pozostawiają zagadnienia związane zwłaszcza z mitem i prawem. Przedstawiona tu analiza otwiera perspektywę dalszych badań nad kodem źródłowym jako przestrzenią działań językowych, które „pozostawiają ślad” w tekście. Wśród potencjalnych kierunków szczególnie interesujące wydaje się badanie stylu programistycznego. Stylometria kodu stanowi obiecujący obszar badań, w którym mogą ujawnić się istotne różnice między dyskursywnym użyciem językiem przez człowieka a operowaniem nim wyłącznie jako systemem przez sztuczną inteligencję.

Innym wartym rozwinięcia zagadnieniem jest problematyka tożsamości narracyjnej programistów korzystających ze sztucznej inteligencji do generowania kodu. Wobec potencjalnego kryzysu w obszarze sposobu myślenia o sobie, zasadne staje się zwrócenie uwagi na przemiany w sposobie działania. Przeobrażenie praktyki językowej może potencjalnie prowadzić do zmian w obszarze kluczowej konstrukcji psychologicznej, jaką jest tożsamość. Sztuczna inteligencja, nie jest już wyłącznie zaawansowanym technologicznie narzędziem – wpływa ona również na sposób w jaki człowiek wchodzi w relację z sobą samym, z innymi ludźmi i jak realizuje wartości.

Pojawia się również pytanie o praktyczny wymiar refleksji nad językiem kodu źródłowego. W obszarze edukacji programistycznej przydatne wydaje się włączenie do niej również perspektywy humanistycznej. Większa świadomość istnienia wymiaru kulturowego kodu źródłowego może wpłynąć na pojawienie się bardziej krytycznych postaw wśród przyszłych programistów. Zrozumienie, w jaki sposób niektórzy twórcy kodu posługują się językiem, potencjalnie pozwala zwiększyć uważność na kwestie jego jakości i etycznych implikacji. Jednocześnie dostrzeżenie kulturowego wymiaru kodu niesie ze sobą zachętę do podejmowania działań językowych zorientowanych na sublimację – przekształcanie napięć psychicznych w twórczość. Przestrzeń ekspresji kulturowej kodu może zatem stanowić swoisty wentyl bezpieczeństwa.

1299 E. Cassirer, *Esej o człowieku...*, s. 411.

W niniejszej rozprawie podjęłam się analizy kodu źródłowego jako wytworu kultury. Perspektywa ta została osiągnięta poprzez refleksję nad językiem jako dyskursem i formą symboliczną. Trzeba podkreślić, że kod źródłowy jako dzieło stanowi rezultat twórczych działań programistów zanurzonych w świecie kultury. Obie koncepcje języka – Ricœura i Cassirera – umożliwiły uchwycenie techniczowanego aspektu rzeczywistości jako przestrzeni znaczeń.

W autentycznym kontakcie z samym sobą¹³⁰⁰, z innymi i w odniesieniu do wartości, ma szansę powstać kod, który nie tylko jest dobry jakościowo, lecz także zostaje wartościowo wykorzystany. W odróżnieniu od sztucznej inteligencji człowiek posiada zdolność refleksowania nad swoim działaniem i w oparciu o nie doskonalenia go. Tej zdolności jako podstawy etycznego działania, w obecnym stadium rozwoju, nie posiada sztuczna inteligencja generująca kod. Kod, który nie tylko działa, ale również niesie znaczenie, staje się symbolicznym pomostem między tym, co techniczne a tym, co humanistyczne.

1300 Ch. Taylor, *Etyka autentyczności...*, s. 28.

Streszczenie

Celem rozprawy było zbadanie sposobu, w jaki język umożliwia powstanie kodu źródłowego jako artefaktu kulturowego. Odwołując się do ujęcia języka zaproponowanego przez Paula Ricœura i Ernsta Cassirera, postawiono pytanie, w jaki sposób język kodu źródłowego umożliwia ekspresję kulturową.

Na gruncie rozważań nad dyskursem i formą symboliczną ukazano kulturowy wymiar kodu źródłowego. W kolejnych rozdziałach rozprawy przedstawiono jego tekst jako przestrzeń, w której dochodzi do ekspresji, konstytuowania i ujawniania spraw istotnych dla człowieka w wymiarze społecznym, indywidualnym oraz etycznym. Rozważania zilustrowano szeregiem przykładów działań językowych podejmowanych przez programistów w różnych obszarach.

Wykorzystanie hermeneutyki dyskursu i filozofii form symbolicznych pozwoliło uchwycić, w jaki sposób w tekście kodu źródłowego programów komputerowych można konstruować, wyrażać i ujawniać znaczące dla człowieka treści. Przedstawiono możliwości języka, którym posługują się programiści, do ujawniania ich relacji z samym sobą, z innymi ludźmi oraz ze światem.

Słowa kluczowe: dyskurs, forma symboliczna, hermeneutyka, język, kod źródłowy

Summary

This dissertation explores the question of how language, understood as a discourse and symbolic form, enables source code to emerge as a cultural artifact. By employing Paul Ricœur's hermeneutics of discourse and Ernst Cassirer's philosophy of symbolic forms, the central question posed is how the language of the source code makes cultural expression possible.

The study reveals the cultural dimension of source code. In the following chapters, source code is presented as a textual space in which expression, constitution, and disclosure of significant matters take place, across social, individual, and ethical dimensions. These reflections are illustrated through a range of examples demonstrating programmers' linguistic practices in each of these areas.

By employing the hermeneutics of discourse and the philosophy of symbolic forms, the dissertation shows how meaningful human can be while constructing, articulating, and disclosing within the textual fabric of computer programs. The analysis highlights the potential language to express developers' relationships with themselves, with others, and with the world.

Keywords: discourse, hermeneutics, language, source code, symbolic form

BIBLIOGRAFIA

- Afflerbach K., *Humorous Developments: Ridicule, Recognition, and the Development of Agency*, praca magisterska, Uniwersytet Północnej Florydy, Jacksonville 2015.
- Ahearn L., *Antropologia lingwistyczna. Wprowadzenie*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2013.
- Alexander H., *Doświadczenie żałoby*, Wydawnictwo WAM, Poznań 2001.
- Anczyk A., Grzymała-Moszczyńska H., Krzysztof-Świdorska A., Prusak J., *Wielokulturowość w pracy psychologa: etyka i praktyka w kontekście diagnozy klinicznej i pracy terapeutycznej*, „Psychologia Wychowawcza”, nr 15, 2019, s. 111-125.
- Andrzejewski B., *Animal symbolicum. Ewolucja neokantyzmu Ernsta Cassirera*, Wydawnictwo Naukowe Uniwersytetu im. Adama Mickiewicza w Poznaniu, Poznań 1980.
- Andrzejewski B., *Problem noumenu w kantyzmie i neokantyzmie, W kręgu inspiracji kantowskich*, red. R. Kozłowski, Państwowe Wydawnictwo Naukowe, Warszawa-Poznań 1983.
- Andrzejewski B., *Wstęp*, [w:] E. Cassirer, *Symbol i Język*, tłum. i red. B. Andrzejewski, WNIF, Poznań 1995, s. 5-10.
- Angermeyer M. C., Matschinger H., *The stigma of mental illness: effects of labelling on public attitudes towards people with mental disorder*, „Acta Psychiatrica Scandinavica”, 108, nr 4, 2003, s. 304-309.
- Appleman D., *When Software Attacks: All About Viruses*, [w:] *Always Use Protection*, red. D. Appleman, Apress, Berkeley 2004, s. 7-15, https://link.springer.com/chapter/10.1007/978-1-4302-0904-1_2 (dostęp dnia 6.07.2024).
- Arendt H., *Kondycja ludzka*, Aletheia, Warszawa 2010.
- Arystoteles, *Etyka Nikomachejska*, Państwowe Wydawnictwo Naukowe, Warszawa 1956.
- Arystoteles, *Poetyka*, tłum. S. Siedlecki, <https://wolnelektury.pl/media/book/pdf/arystoteles-poetyka.pdf> (dostęp dnia 18.03.2023).

- Atlas języków. Pochodzenie i rozwój języków świata*, red. B. Comrie, S. Matthews, M. Polonsky, Oficyna Wydawnicza Atena, Poznań 1998.
- Bajer A., *Understanding of source code in language: Contribution of philosophical hermeneutics to the critical code studies*, „Digital Scholarship in the Humanities Oxford University Press”, 37, nr 2, 2021, s. 307-320, <https://academic.oup.com/dsh/article-abstract/37/2/307/6375769?redirectedFrom=fulltext> (dostęp dnia 21.09.2021)
- Bała M., *O możliwości hermeneutycznej filozofii religii. Propozycja Paula Ricœura*, Wydawnictwo Uniwersytetu Kardynała Stefana Wyszyńskiego, Warszawa, 2007.
- Bal-Nowak M., *Mit jako forma symboliczna w ujęciu Ernsta A. Cassirera*, Zakład Wydawniczy Nomos, Kraków 1996.
- Barteczko K., *JAVA. Programowanie praktyczne od podstaw*, Państwowe Wydawnictwo Naukowe, Warszawa 2014.
- Bartuś K., Bartuś T., *Technologia inteligentnych agentów na rynku elektronicznym*, „Informatyka Ekonomiczna”, 15, nr 88, 2010, s. 90-104, https://www.dbc.wroc.pl/Content/121821/Bartus_Bartus_Technologia_inteligentnych_agentow.pdf (dostęp dnia 7.09.2024).
- Baumgardt D., *Cassirer and the chaos in moder ethics*, [w:] *The philosophy of Ernst Cassirer*, red. P. Schilpp, Georg Banta Publishing Company, Evanstone 1949, s. 575-605.
- Bełza M., Prysak D., „Upokorzona” codzienność osób niepełnosprawnych, „Parecja”, nr 2, 2014, s. 25-37.
- Bembenek K., *Problem niewolnej woli w ujęciu fenomenologii hermeneutycznej - stanowisko Paula Ricœura*, „Rocznik Filozoficzny”, 65, nr 3, 2017, s. 173-189.
- Berlin I., *Dwie koncepcje wolności*, [w:] *Cztery eseje o wolności*, red. I. Berlin, Państwowe Wydawnictwo Naukowe, Warszawa 1994, s. 180-195.
- Biancuzzi F., Warden S., *Wielkie umysły programowania. Jak myślą i pracują twórcy najważniejszych języków*, Helion, Gliwice 2010.
- Bingham S. C., Green S.E., *Aesthetic as Analysis: Synthesizing Theories of Humor and Disability through Stand-up Comedy*, „Humanity & Society”, 40, nr 3, 2015, s. 278-305.

- Bober W., *Powinność w świecie cyfrowym: etyka komputerowa w świetle współczesnej filozofii moralnej*, Wydawnictwo Akademickie i Profesjonalne, Warszawa 2008.
- Bobowska-Nastarzewska P., *Pojęcie człowieka w idiolekcie Paula Ricœura na podstawie wybranych metafor językowych w Le volontaire et l'involontaire*, „Przegląd humanistyczny”, 60, nr 3, 2016, s. 145-152.
- Bobryk J., *Język mówiony, pismo, Internet*, [w:] *Język jako przedmiot badań psychologicznych. Psycholingwistyka ogólna i neurolingwistyka*, red. I. Kurcz, H. Okuniewska, Wydawnictwo Szkoły Wyższej Psychologii Społecznej „Academica”, Warszawa 2011, s. 168-183.
- Bondecka-Krzykowska I., *Historia obliczeń, Od rachunku na palcach do maszyny analitycznej*, Wydawnictwo Naukowe UAM, Poznań 2012.
- Bondecka-Krzykowska I., *O związkach informatyki z matematyką*, „Filozofia Nauki”, 18, nr 1, 2010, s. 77-89.
- Bondecka-Krzykowska I., *Paradygmaty informatyki*, [w:] *Światy matematyki. Tworzenie czy odkrywanie*, red. I. Bondecka-Krzykowska, J. Pogonowski, Wydawnictwo Naukowe UAM, Poznań 2010, s. 117-134.
- Bondecka-Krzykowska I., *Z zagadnień ontologicznych informatyki*, Wydawnictwo Naukowe UAM, Poznań 2016.
- Boswell D., Foucher T., *The Art of Readable Code*, O'Reilly, Sebastopol 2011.
- Bradshaw J., *Introduction to software agents*, [w:] *Software Agents*, red. J. Bradshaw, AAAI Press/The MIT Press, Boston 1999, s. 3-46.
- Brohm J. M., Ricœur P., *Arts, language and hermeneutical aesthetics: Interview with Paul Ricœur (1913-2005)*, „Philosophy and Social Criticism”, 36, nr 8, 2010, s. 935-951.
- Bronk A., *Rozumienie, dzieje, język. Filozoficzna hermeneutyczna H.-G. Gadamera*, Redakcja Wydawnictw KUL, Lublin 1988.
- Brzozowski P., *Uniwersalna hierarchia wartości – fakt czy fikcja*, „Przegląd psychologiczny”, 48, nr 3, 2005, s. 261-276, https://www.kul.pl/files/714/nowy_folder/3.48.2005_artylul_3.pdf (dostęp dnia 25.03.2024).

- Buber M., *Ja i ty. Wybór pism filozoficznych*, Instytut Wydawniczy PAX, Warszawa 1992.
- Buczyńska-Garewicz H., *Cassirer*, Wiedza Powszechna, Warszawa 1963.
- Buczyńska-Garewicz H., *Semiotyka Peirce'a*, ZSL UW, Warszawa 1994.
- Burgos J., *Personalizmy. Autorzy i tematy nowej filozofii*, Centrum Myśli Jana Pawła II, Warszawa 2010.
- Burzyńska A., *Dekonstrukcja i interpretacja*, Universitas, Kraków 2001.
- Caron P., *A Critique of Hall's Contexting Model A Meta-Analysis of Literature on Intercultural Business and Technical Communication*, „Journal of Business and Technical Communication”, 22, nr 4, 2008, s. 399-428.
- Carro M., Morales J., Muller H., *High-Level Languages for Small Devices: A Case Study*, CASES '06, Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems, Association for Computing Machinery, Nowy York 2006, http://cliplab.org/papers/carro06:stream_interpreter_cases.pdf (dostęp dnia 18.03.2023).
- Cassirer E., *Critical Idealism as a Philosophy of Culture*, [w:]: *Symbol, Myth and Culture: essays and lectures of Ernst Cassirer, 1935-1945*, red. D. Verene, Yale University Press, New Heaven 1979, s. 64-95.
- Cassirer E., *Esej o człowieku. Wprowadzenie do filozofii ludzkiej kultury*, Czytelnik, Warszawa 1977.
- Cassirer E., *Filozofia form symbolicznych. Część pierwsza: Język*, red. P. Parszutowicz, Wydawnictwo Marek Derewiecki, Kęty 2018.
- Cassirer E., *Form und Technik*, [w:] *Symbol, Technik, Sprache* red. J. Werle, Felix Meiner Verlag, Hamburg 1985, s. 39-90, https://monoskop.org/File:Cassirer_Ernst_1930_1985_Form_und_Technik.pdf (dostęp dnia 26.07.2024).
- Cassirer E., *Język i budowa świata przedmiotowego*, [w:] E. Cassirer, *Symbol i Język*, tłum. i red. B. Andrzejewski, WNIF, Poznań 1995, s. 76-106.
- Cassirer E., *Język i mit. Przyczynek do zagadnienia imion bogów*, Wydawnictwo Marek Derewiecki, Kęty 2021.

- Cassirer E., *Logika nauk o kulturze: pięć studiów*, red. P. Parszutowicz, Wydawnictwo Marek Derewiecki, Kęty 2011.
- Cassirer E., *Mit państwa*, Wydawnictwo IFIS PAN, Warszawa 2006.
- Cassirer E., *Pojęcie formy symbolicznej w budowie nauk humanistycznych*, [w:] E. Cassirer, *Symbol i Język*, tłum. i red. B. Andrzejewski, WNIF, Poznań 1995, s. 11-44.
- Cassirer, E. Cassirer, *Problem symbolu i jego miejsce w systemie filozofii*, [w:] E. Cassirer, *Symbol i Język*, tłum. i red. B. Andrzejewski, WNIF, Poznań 1995, s. 44-65.
- Cassirer E., *Pojęcia przyrodnicze i pojęcia kulturowe. Przyczynek do zagadnienia logiki nauk humanistycznych*, [w:] H. Buczyńska, Cassirer, Wiedza Powszechna, Warszawa, 1963, s. 119-135.
- Cassirer E., *The Problem of Knowledge: Philosophy, Science, and History Since Hegel*, Yale University Press, New Heaven 1969.
- Chomsky N., *Zagadnienia teorii składni*, Zakład Narodowy Imienia Ossolińskich, Wrocław 1982.
- Cichowicz S., *Filozofia i hermeneutyka*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 5-31.
- Cipolletti H., McFarlane S., Weissglass C., *The moral foreign-language effect*, „Philosophical Psychology” nr 29, 2016, s. 23-40, <https://www.tandfonline.com/doi/full/10.1080/09515089.2014.993063> (dostęp dnia 06.04.2024).
- Clifton M., *The Software Development Process - Science, Engineering, Art, or Craft?*, 2016 <https://www.codeproject.com/Articles/1130886/The-Software-Development-Process-Science-Engineeri> (dostęp dnia 7.02.2024).
- Coplestone F., *Historia filozofii. Od Maine de Birana do Sartre'a*. T.9, Instytut Wydawniczy PAX, Warszawa 2006.
- Coskun D., *Law as Symbolic Form Ernst Cassirer and the Anthropocentric View of Law*, Springer, Dordrecht 2007, <https://link.springer.com/book/10.1007/978-1-4020-6256-8> (dostęp dnia 1.09.2024).

- Cox. G., McLean A., Ward A., *The Aesthetics of Generative Code*, Generative Art 00 conference 2001, Politecnico di Milano, Milan 2001, https://www.academia.edu/10519146/The_Aesthetics_of_Generative_Code (dostęp 7.07.2022).
- Cramer F., *Digital Code and Literary Text*, "Beehive", 4, nr 3, 2001, s. 1-8, https://www.netzliteratur.net/cramer/digital_code_and_literary_text.html (dostęp dnia 7.07.2022).
- Davis P., Hersh R., *Świat matematyki*, Państwowe Wydawnictwo Naukowe, Warszawa 1994.
- Dijkstra E. W., *On the Cruelty of Really Teaching Computing Science*, Center for American History, University of Texas at Austin, Austin 1989, <https://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1036.PDF>, (dostęp dnia 07.07.2022).
- Dovey J., *Kultura gier komputerowych*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2011.
- Drwięga M., *Paul Ricœur daje do myślenia*, Wydawnictwo Homini, Bydgoszcz, 1998.
- Drwięga M., *Paula Ricœura spotkania z psychoanalizą*, „Kwartalnik filozoficzny”, 41, nr 4, 2013, s. 117-131.
- Dul F., *Wprowadzenie do sztucznej inteligencji. Materiały z wykładów*, Politechnika Warszawska, Warszawa 2014.
- Dziemidok B., *O komizmie*, Książka i Wiedza, Warszawa 1967.
- Eco U., *Poszukiwanie języka doskonałego*, Wydawnictwo Aletheia, Warszawa 2013.
- Eco U., *Teoria semiotyki*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2009.
- Eden A., *Three Paradigms of Computer Science*, "Minds and Machines", 17, nr 2, 2007, s. 1-30.
- Eglash R., *Broken Metaphor. The Master-Slave Analogy in Technical Literature*, „Technology and Culture”, 48, nr 2, 2007, 360-369.
- Elzenberg H., Tatarkiewicz W., *Historia Estetyki*, „Ruch Filozoficzny”, 21, nr 1, 1962, s. 159-169.

- Europejski system opisu kształcenia językowego: uczenie się nauczanie, ocenianie*, Wydawnictwo Centralnego Ośrodka Doskonalenia nauczycieli, Białystok 2003, tłum. W. Martyniuk, <https://ore.edu.pl/2010/12/europejski-system-opisu-kształcenia-językowego-uczenie-si-nauczanie-ocenianie/> (dostęp dnia 22.09.2024).
- Eyman D., *Digital Rhetoric: Theory, Method, Practice*, University of Michigan Press, Ann Arbor 2015.
- Fairclough N., *Discourse and social change*, Polity Press, Cambridge 1992.
- Fauzia M., *Fact check: Facebook didn't pull the plug on two chatbots because they created a language*, USA Today, <https://eu.usatoday.com/story/news/factcheck/2021/07/28/fact-check-facebook-chatbots-werent-shut-down-creating-language/8040006002/> (24.10.2024).
- Feathers M., *Praca z zastanym kodem. Najlepsze techniki*, Helion, Gliwice 2014.
- Federenko E., Ivanova A., Dhamala R., *The Language of Programming: A Cognitive Perspective*, „Science & Society”, 23, nr 7, s. 525-528, [https://www.cell.com/trends/cognitive-sciences/fulltext/S1364-6613\(19\)30102-0?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS1364661319301020%3Fshowall%3Dtrue](https://www.cell.com/trends/cognitive-sciences/fulltext/S1364-6613(19)30102-0?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS1364661319301020%3Fshowall%3Dtrue) (dostęp dnia 8.10.2023).
- Filutowska K., *Tożsamość narracyjna jako empiryczna podmiotowość – MacIntyre, Taylor, Ricœur*, Wydawnictwo Uniwersytetu Warszawskiego, Warszawa 2018.
- Flower M., Beck K., *Refaktoryzacja. Ulepszanie struktury istniejącego kodu*, Helion, Gliwice 2019.
- Foucault M., *Historia szaleństwa w dobie klasycyzmu*, PIW, Warszawa 1987.
- Freud A., *Ego i mechanizmy obronne*, Państwowe Wydawnictwo Naukowe, Warszawa 1997.
- Fuegi J., Francis J., *Lovelace & Babbage and the Creation of the 1843 'Notes'*, „IEEE Annals of the History of Computing”, 25, nr 4, 2003, s. 16-26.
- Furdal A., *Językoznawstwo otwarte*, Wydawnictwo Zakład Narodowy im. Ossolińskich, Wrocław 2000.
- Gabbard G. O., *Psychiatria psychodynamiczna w praktyce klinicznej*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2009.

- Gadamer H.-G., *Człowiek i język*, [w:] *Rozum, słowo, dzieje*, red. M. Łukasiewicz, K. Michalski, PIW, Warszawa 2000, s. 52-63.
- Gadamer H.-G., *Język i rozumienie*, [w:] *Język i rozumienie*, Fundacja Aletheia, Warszawa 2003, s. 5-24.
- Gadamer H.-G., *Prawda i metoda*, Państwowe Wydawnictwo Naukowe, Warszawa 2013.
- Gander F., Proyer R., *Strength-Based Positive Interventions: Further Evidence for Their Potential in Enhancing Well-Being and Alleviating Depression*, „Journal of Happiness Studies” 14, 2013.
- Garbat M., *Historia niepełnosprawności*, Novae res, Gdynia 2015.
- Gardner R. A., Gardner B. T., *Teaching sign language to a chimpanzee*, „Science”, 165, 1969, s. 664-672.
- Gawkowska A., *Biorąc wspólnotę poważnie*, Wydawnictwo Instytut Filozofii i Socjologii PAN, Warszawa 2004.
- Geipel J., Hadjichristidis C., Surian L., *The Foreign Language Effect on Moral Judgment: The Role of Emotions and Norms*, “Plos one”, nr 10, 2015, s. 1-17.
- Gemel A., *Językowy model poznania. Kognitywne komponenty w kontynentalnej filozofii języka*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2015.
- Ghosh R., Glott R., Krieger B., Robles G., *Free/Libre and Open Source Software: Survey and Study. Part 4: Survey of Developers*, International Institute of Infonomics University of Maastricht, Maastricht 2002.
- Giddens A., *Nowoczesność i tożsamość*, Państwowe Wydawnictwo Naukowe, Warszawa 2001.
- Giddens A., *Stanowienie społeczeństwa*, Zysk, Poznań 2003.
- Gisel P., *Wstęp*, [w:] P. Ricœur, *Zło. Wyzwanie rzucone filozofii i teologii*, Instytut Wydawniczy PAX, Warszawa 1992, s. 3-10.
- Glendinning S., *The Idea of Continental Philosophy*, Edinburgh University Press, Edinburgh 2006.
- Głowiński M., *Słownik terminów literackich*, red. S. Sławiński, Zakład Narodowy im. Ossolińskich, Wrocław 1988.

- Gołaszewska M., *Estetyka współczesności*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2001.
- Gołaszewska M., *O naturze wartości estetycznych*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 1986.
- Gołaszewska M., *Śmieszność i komizm*, Polska Akademia Nauk, Wrocław 1987.
- Goldberg E., *Kreatywność. Mózg w dobie innowacji*, Państwowe Wydawnictwo Naukowe, Warszawa 2018.
- Gøtzsche P., *Citation bias: questionable research practice or scientific misconduct?*, „Journal of the Royal Society of Medicine”, 115, nr 1, 2022, s. 31-35,
<https://journals.sagepub.com/doi/full/10.1177/01410768221075881> (dostęp dnia 22.8.2024).
- Gousios G., Pinzger M., van Deursen A., *An exploratory study of the pull-based software development model*, [w:] *Proceedings of the 36th international conference on Software engineering*. ACM 2014, s. 345-355.
- Grabowski D., *Sigmunda Freuda teoria dowcipu, humoru i komiki*, Universitas, Kraków 2019.
- Gray A., Sallis P., MacDonell S., *IDENTIFIED (Integrated Dictionary-based Extraction of Non-language-dependent Token Information for Forensic Identification, Examination, and Discrimination): a dictionary-based system for extracting source code metrics for software forensics*, *Proceedings of Software Engineering, „Education & Practice”*, IEEE Computer Society Press, 1998.
- Greenstein S., F. Zhu, *Open Content, Linus' Law, and Neutral Point of View*, “Information Systems Research”, 27, nr 3, 2016, s. 618-635,
<https://pubsonline.informs.org/doi/abs/10.1287/isre.2016.0643> (dostęp dnia 8.08.2024).
- Griffin R., *Anonymity and Authorship*, „New Literary History”, 30, nr 4, 1999, 877-895,
<https://muse.jhu.edu/article/24485> (dostęp dnia 1.09.2024).
- Grob A., Meyer Ch., Hagmann-von Arx P., *Skale inteligencji i rozwoju dla dzieci w wieku 5-10 lat: IDS intelligence and development scales. Podręcznik*, adaptacja A. Jaworowska, A. Matczak, D. Fecenec, Pracownia Testów Psychologicznych Polskiego Towarzystwa Psychologicznego, Warszawa 2012.

- Grondin J., *Do Gadamer and Ricœur have the same understanding of hermeneutics?*, [w:] M. Xie, *The Agon of Interpretations: Towards a Critical Intercultural Hermeneutics*, Toronto University Press, Toronto 2014, s. 43-64.
- Grzegółowska-Klarkowska H. J., *Mechanizmy obronne osobowości*, Państwowe Wydawnictwo Naukowe, Warszawa 1986.
- Grzywacz R., *Dyskurs przekonań podmiotu a tożsamość narracyjnej jednostki i wspólnoty w filozofii Paula Ricœura. Autoreferat z rozprawy doktorskiej*, „Rocznik Filozoficzny Ignatianum”, 21, nr 1, 2016, s. 87-92.
- Gułkowski J., *O istocie i naturze kultury*, [w:] *Co to jest filozofia kultury*, red. Z. Rosińska, J. Michalik, Wydawnictwo Uniwersytetu Warszawskiego, Warszawa 2006, s. 30-48.
- Gurczyńska-Sady K., *Troska o świat. Antropologia filozoficzna Hannah Arendt*, Wydawnictwo Naukowe UP, Kraków 2019.
- Habermas J., *Filozofia analityczna i hermeneutyczna – dwie komplementarne odmiany zwrotu lingwistycznego*, [w:] *Język, dyskurs, społeczeństwo*, red. L. Rasiński, Państwowe Wydawnictwo Naukowe, Warszawa 2009, s. 244-268.
- Hacker S., *Pleasure, Power, and Technology: Some Tales of Gender, Engineering, and the Cooperative Workplace*, Unwin Hyman, Boston 1989.
- Hall E., *Poza kulturą*, Państwowe Wydawnictwo Naukowe, Warszawa 2001.
- Hańderek J., *Pojęcia i definicje kultury*, [w:] *Filozofia kultury*, red. P. Mróz, WUJ, Kraków 2015, s. 23-42.
- Haque R, Rubya S., *An Overview of Chatbot-Based Mobile Mental Health Apps: Insights From App Description and User Reviews*, „JMIR Mhealth Uhealth”, 11, 2023, <https://pmc.ncbi.nlm.nih.gov/articles/PMC10242473/> (dostęp dnia 17.10.2024).
- Harris C., Gleason J., Ayçiçeği A., *When is a first language more emotional? Psychophysiological evidence from bilingual speakers*, [w:] *Bilingual minds: Emotional experience, expression, and representation*, red. A. Pavlenko, Multilingual Matters, Clevedon 2006, s. 257-283.
- Heidegger M., *W drodze do języka*, Wydawnictwo Baran i Suszczyński, Kraków 2000.

- Heroux M., Sexton P., *Epetra developers coding guidelines*. No. SAND2003-4169. Sandia National Laboratories (SNL), Albuquerque 2003, <https://www.osti.gov/servlets/purl/918292> , (dostęp dnia: 05.05.2023).
- Heszen I., Sęk H., *Psychologia zdrowia*, Państwowe Wydawnictwo Naukowe, Warszawa 2007.
- Hiller M., *Signs of the times: the software of the philology and a philology of the software*, "Digital Culture and Society", 1, 2015, s. 151-163.
- Hinwar R., A Lambert, *Anauralia: The Silent Mind and Its Association With Aphantasia*, „Frontiers in Psychology”, 12, 2021, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8551557/> (dostęp dnia 10.01.2024).
- Hollanek T., Nowaczyk-Basińska K., *Griefbots, Deadbots, Postmortem Avatars: on Responsible Applications of Generative AI in the Digital Afterlife Industry*, „Philosophy and Technology”, 37, nr 63, 2024, <https://link.springer.com/article/10.1007/s13347-024-00744-w#citeas> (dostęp dnia 5.10.2024).
- Hopkins S., Corporation T., *Camels and needles: computer poetry meets the Perl Programming Language*, Usenix Winter 1992 Technical Conference, s. 1-15, http://www.digitalcraft.org/iloveyou/images/Sh.Hopkins_Perl_Poetry.pdf (dostęp dnia 01.03.2023).
- Hudziec M., *Antoni Kępiński — swoistość i granice poznania drugiego człowieka w perspektywie spotkania*, „Studia Philosophica Wratislaviensia”, 11, nr 4, 2016, s. 65-79.
- Hunt A., Thomas D., *Pragmatyczny programista. Od czeladnika do mistrza*, Helion, Gliwice 2011.
- Idialu O., Mathews N., Whodunit, R. *Maipradit: Classifying Code as Human Authored or GPT-4 generated - A case study on CodeChef problems*, MSR '24: 21st International Conference on Mining Software Repositories 2024, https://www.researchgate.net/publication/381950411_Whodunit_Classifying_Code_as_Human_Authored_or_GPT-4_Generated_-_A_case_study_on_CodeChef_problems (dostęp dnia 05.10.2024).
- Ingarden R., *Książeczka o człowieku*, Wydawnictwo literackie, Kraków 2017.

- Irwin L., *Implementing Phrase-Structure Productions in PL/I*, „Communications of the ACM”, 10, nr 7, 1967, s. 424, <https://dl.acm.org/doi/pdf/10.1145/363427.363445> (dostęp dnia 06.09.2024).
- Izert M., Pachocińska E., *Wstęp do językoznawstwa ogólnego*, ZG UW, Warszawa 1998.
- Jackson G., *Transcoding Sexuality: Computational Performativity and Queer Code Practices*, „A Journal in GLBTQ Worldmaking”, 4, nr 2, 2017, s. 1-25.
- Jalote P., *A Concise Introduction to Software Engineering*, Springer, Londyn 2008, https://link.springer.com/chapter/10.1007/978-1-84800-302-6_7 (dostęp 06.07.2024).
- Jarymowicz M., Imbir K., *Próba taksonomii ludzkich emocji*, „Przegląd Psychologiczny”, 53, nr 4, 2010, 439-461.
- Jarzyńska K., *Ernst Cassirer. Mit Państwa*, „Politeja”, nr 1, 2008, s. 514-519.
- Jasiński M., *Myślenie krytyczne i kreatywne*. Sylabus, WSB-NLU.
- Jaworski K., *Czy liczby istnieją? Rekonstrukcja sporu o istnienie przedmiotów matematycznych*, „Humanities and Social Sciences”, 23, nr 25, 2018, s. 99-113.
- Jerz D., *Somewhere Nearby is Colossal Cave: Examining Will Crowther's Original „Adventure” in Code and in Kentucky*, „Digital Humanities Quarterly”, 1, nr 2, 2007, <http://www.digitalhumanities.org/dhq/-vol/001/2/000009/000009.html> (dostęp dnia 17.07.2022).
- Judycki S., *Zachowanie i działanie*, „Diametros”, nr 7, 2006, s. 82 – 97, <https://diametros.uj.edu.pl/diametros/issue/view/9> (dostęp dnia 12.09.2024).
- Kalaga W., *Mgławice dyskursu*, Universitas, Kraków 2001.
- Kaufmann H., *Dzieje komputerów*, Państwowe Wydawnictwo Naukowe, Warszawa 1980.
- Kearney R., *Foreword to the New Edition*, [w:] *From text to action; Essays in Hermeneutics*, II, red. R. Kearney, Northwestern University Press, Evanston 2007, s. IX-X.
- Kemeny J., Kurtz T., *Dartmouth Timesharing*, „Science”, 162, nr 3850, 1968, s. 223–268.
- Kępiński A., *Poznanie chorego, Rytm życia*, Sagittarius, Warszawa 1992.

- Kirschenbaum M., *Where Computer Science and Cultural Studies Collide. The Chronicle of Higher Education*, 2009, <https://www.chronicle.com/article/where-computer-science-and-cultural-studies-collide/> (dostęp dnia 7.07.2022).
- Kiwak W., *Własność jako uprawnienie*, „Studia Ekonomiczne. Zeszyty Naukowe Uniwersytetu Ekonomicznego w Katowicach”, nr 260, 2016, s. 45-55.
- Knuth D. E., *Sztuka programowania. Algorytmy podstawowe*, T. 1, Wydawnictwo Naukowo-Techniczne, Warszawa, 2002.
- Kocikowski A., *Słów kilka o etyce komputerowej*, [w:] *Wartość i świadomość w filozoficznej drodze*, red. E. Czerwińska, Instytut Historii UAM, Poznań 1995, s. 207-215.
- Kosowska-Czapla E., *Ekofilozofia a filozofia prawa rozumianego jako forma symboliczna*, „Colloquia Theologica Ottoniana”, 38, 2022, s. 237-257.
- Kowalska E., *Wprowadzenie*, [w:] W. Humboldt, *Rozmaitość języków a rozwój umysłowy ludzkości*, przekł. i wstęp E. Kowalska, KUL, Lublin 2001, s. 5-64.
- Kowalska M., *Wstęp. Dialektyka bycia sobą*, [w:] P. Ricœur, *O sobie samym jako innym*, przeł. B. Chelstowski, Państwowe Wydawnictwo Naukowe, Warszawa 2003, s. VII-XXXVII.
- Kowalski A., *Mit a sztuka w Ernsta Cassirera filozofii form symbolicznych*, „Filo-Sofija”, 3, nr 3, 2003, s. 117-130.
- Krajewski S., *Czy matematyka jest nauką humanistyczną*, Copernicus Center Press, Kraków 2011.
- Krois J. M., *Cassirer: Symbolic Forms and History*, New Heaven, London 1987.
- Królikowski T., Susłow W., *Analityczny projekt informatyzacji przedsięwzięcia szkoleniowego*, „Zeszyty Naukowe Wydziału Elektroniki i Informatyki Politechniki Koszalińskiej”, nr 14, 2019, s. 5-21.
- Kuć M., *Prawne podstawy resocjalizacji*, Beck, Warszawa 2013.
- Kulczycki E., *Założenia filozoficznych języków apriorycznych*, [w:] *My w wieży Babel. Między przekleństwem a błogosławieństwem*, red. M. Cieszkowski, J. Szczepaniak, Wydawnictwo Uniwersytetu Kazimierza Wielkiego, Bydgoszcz 2011, s. 39-49.
- Kurata T., *Poza ontologią: Paul Ricœur i kwestia sztuki*, „Sztuka i Filozofia”, 2223, 2005, s. 51-63.

- Kurcz I., Tomaszewski P., *Powstanie i ewolucja języka*, [w:] *Język jako przedmiot badań psychologicznych. Psycholingwistyka ogólna i neurolingwistyka*, red. I. Kurcz, H. Okuniewska, Wydawnictwo Szkoły Wyższej Psychologii Społecznej „Academica”, Warszawa 2011, s. 45-65.
- Kwaśnik J., *Charakterystyka subkultury hackerów jako kluczowego elementu cyberprzestępczości*, „Internetowy Przegląd Prawniczy TBSP UJ”, nr 1, 2010, s. 24-39, <https://ruj.uj.edu.pl/entities/publication/9fe74712-86cf-4806-b5bf-932427347e43> (dostęp dnia 01.04.2024).
- Lakatos I., *Renesans empiryzmu we współczesnej filozofii matematyki*, [w:] *Współczesna filozofia matematyki: wybór tekstów*, red. R. Murawski, Państwowe Wydawnictwo Naukowe, Warszawa 2002, s. 215-243.
- Lane H., *Maska dobroczynności. Deprecjacja społeczności głuchych*, Wydawnictwo Szkolne i Pedagogiczne, Warszawa 1996.
- Lang H., *Język i nieświadomość. Podstawy teorii psychoanalitycznej Jacques’a Lacana*, Wydawnictwo Słowo/Obraz Terytoria, Gdańsk 2005.
- Ledóchowski Z., *Rola sztuki programowania w kształceniu informatycznym*, „Edukacyjna Analiza Transakcyjna”, nr 7, 2018, s. 195-207.
- Lee D., *Man jailed in UK's first treason case for 40 years*, 2023, BBC, <https://www.bbc.com/news/live/uk-66108009> (dostęp dnia 18.10.2024).
- Lennenberg E., *A Note on Cassirer's Philosophy of Language*, “Philosophy and Phenomenological Research”, 15, nr 4, 1955, s. 512-522, <https://www.jstor.org/stable/2103911> (dostęp dnia 2.10.2023)
- Levy S., *Hackers: Heroes of the Computer Underground*, 1984.
- Lewis M., Yarats D., Dauphin Y., Parikh D., Batra D., *Deal or No Deal? End-to-End Learning for Negotiation Dialogues*, 2017, s. 1-11, <https://arxiv.org/pdf/1706.05125> (dostęp dnia 4.09.2024).
- Lindstrom L., *Dyskurs*, [w:] *Encyklopedia antropologii społeczno-kulturowej*, red. A. Barnard, J. Spence, Oficyna Wydawnicza Volumen, Warszawa 2008, s. 179-180.
- Lis M., *Leksykon kieszonkowy PHP*, Helion, Gliwice 2012.

- Logika formalna*, red. W. Marciszewski, Państwowe Wydawnictwo Naukowe, Warszawa 1987.
- Lorenc I., *Ontologizacja języka a koncepcja źródłowego logosu w ujęciach Heideggera i Merleau-Ponty'ego*, „Ruch filozoficzny”, 73, nr 3, 2018, s. 97-110.
- Lorenc W., *Hermeneutyczne koncepcje człowieka, W kręgu inspiracji heideggerowskich*, Scholar, Warszawa 2003.
- Łosiak W., *Psychologia emocji*, Wydawnictwa Akademickie i Profesjonalne, Warszawa 2007.
- Lubecki M., *Przestrzenie otwartości w przyczynkach do filozofii Martina Heideggera*, „Zeszyty Naukowe Towarzystwa Doktorantów UJ. Nauki Humanistyczne”, nr 3, 2011, s. 53-69.
- Lubowicka G., *Refleksyjność człowieka wobec świata znaczeń*, [w:] *Codziennosc jako wyzwanie edukacyjne*, tom II, *Refleksyjność w codzienności edukacyjnej*, red. I. Paszenda, Instytut Pedagogiki Uniwersytetu Wrocławskiego, Wrocław 2017, s. 62-73.
- Luft S., *The space of culture: towards a neo-Kantian philosophy of culture (Cohen, Natorp, and Cassirer)*, Oxford University Press, Oxford 2015.
- Machado A., *In Defense of Piaget's Theory: A Reply to 10 Common Criticisms*, „Psychological Review”, 103, nr 1, 1996, s. 143-164.
- Macko A., *Umysł moralny: jak powstają oceny moralne*, Poltex, Warszawa 2018.
- Madison G. B., *Hermeneutyka: Gadamer i Ricœur*, [w:] *Historia filozofii zachodniej*, red. R.H. Popkin, Zysk i S-ka, Poznań 2003, s. 715-721.
- Maj F., *Psychoanalityczne i egzystencjalne problemy twórczości*, „Humanistyka i przyrodoznawstwo”, 15, 2009, s. 143-153, <https://czasopisma.uwm.edu.pl/index.php/hip/article/view/947/814> (dostęp dnia 25.10.2023).
- Malata P., *Podmiot w filozofii kultury od Ernsta Cassirera do Paula Ricœura. Antropologia symbolu*, nieopublikowana praca doktorska, Kraków 2017.
- Mallery J. C., Hurwitz R., Duffy G., *Hermeneutics: From Textual Explication to Computer Understanding? The Encyclopedia of Artificial Intelligence*, John Wiley & Sons, New York 1986.

- Mancuso S., *Software Craftsman. Profesjonalizm, czysty kod i techniczna perfekcja*, Helion, Gliwice 2016.
- Mäntylä M. V., Lassenius C., *Subjective Evaluation of Software Evolvability Using Code Smells: An Empirical Study*, „Journal of Empirical Software Engineering”, 11, nr 3, 2006, s. 395-431, <http://mikamantyla.eu/BadCodeSmellsTaxonomy.html> (dostęp dnia 20.04.2017).
- Mariański J., *Socjologia Moralności*, Wydawnictwo KUL, Lublin 2006.
- Marino M. C., *Critical Code Studies: Initial methods*, The MIT Press, Cambridge 2020.
- Marino M.C., *Critical code studies. Electronic Book Review*, 2006, <http://www.electronicbookreview.com/thread/electropoetics/codology> (dostęp dnia 2.07.2022).
- Marquard O., *Apologia przypadkowości*, Oficyna Naukowa, Warszawa 1994.
- Marriott H., Pitardi V., *One is the loneliest number... Two can be as bad as one. The influence of AI Friendship Apps on users' well-being and addiction*, „Psychology & marketing”, 41 nr 1, 2023, s. 86-101, <https://openresearch.surrey.ac.uk/esploro/outputs/journalArticle/One-is-the-loneliest-number-Two/99817566302346> (dostęp dnia 18.10.2024).
- Martin R., *Czysty kod*, Helion, Gliwice 2014.
- Maruszewski T., *Psychologia poznania. Umysł i świat*, Gdańskie Wydawnictwo Psychologiczne, Gdańsk 2011.
- Marzluff J., Angell T., *Gifts of the Crow. How Perception, Emotion, and Thought Allow Smart Birds to Behave Like Humans*, Simon and Schuster, New York 2013.
- Mateas M., Stern A., *Natural Language Processing in Facade: Surface Text Processing*. Paper presented at the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE), 2004, <http://users.soe.ucsc.edu/~mi-chaelm/publications/mateas-tidse2004.pdf> (dostęp dnia 7.07.2022).
- Matsumoto Y., *Traktując kod jako esej*, [w:] *Piękny kod. Tajemnice mistrzów programowania*, red. A Oram, G. Wilson, Helion, Warszawa 2008, 495-499.

- McConnell S., *Kod doskonały. Jak tworzyć oprogramowanie pozbawione błędów*, Helion, Gliwice 2010.
- McWilliams N., *Psychoterapia psychoanalityczna*, Wydawnictwo Harmonia, Gdańsk 2011.
- Melaney W. D., *Ricœur's transcendental concern: a hermeneutics of discourse*, [w:] *The Fullness of the Logos in the Key of Life: Book II. Christo-Logos: Metaphysical Rhapsodies of Faith. Itinerarium mentis in deo. Analecta Husserliana 111*, red. Anna-Teresa Tymieniecka, Springer, Hanover 2011, s. 495-513.
- Michałowski D., Otrocki M., *Hermeneutyka dyskursu według Paula Ricœura*, „Kwartalnik Filozoficzny”, 41, nr 4, 2013, s. 131-152.
- Michałowski D., *Paul Ricœura hermeneutyka symboli. Próba prezentacji*, „Kwartalnik Filozoficzny”, 40 nr 3, 2012, s. 91-114.
- Michalska P., Szymanik-Kostrzevska A., Gurba E., Trempała J., *Rozwój myślenia w dorosłości: o sposobach badania rozumowania postformalnego w codziennych sytuacjach*, „Psychologia rozwojowa”, 21, nr 2, 2016, s. 55-71.
- Michalski K., *Heidegger i filozofia współczesna*, PIW, Warszawa 1978.
- Miles H. L., *The cognitive foundation for reference in a signing orangutan*, [w:] *“Language” and Intelligence in Monkeys and Apes: Comparative Developmental Perspective*, red. S. T. Parker, K. Gibson, The MIT Press, Cambridge 1990, s. 356-378.
- Moor J., *What is computer ethics?*, "Metaphilosophy", 16, nr 4, 1985, s. 266-275, <http://mumelab01.amu.edu.pl/Wprowadzenie-HTML/KO-05-01.html> (dostęp dnia 01.04.2024).
- Moratis K.; T. Diamantopoulos; D. Nastos; A.Symeonidis, *Write me this Code: An Analysis of ChatGPT Quality for Producing Source Code*, 2024 IEEE/ACM 21st International Conference on Mining Software Repositories, Lisbon 2024, <https://ieeexplore.ieee.org/abstract/document/10555604> (dostęp dnia 25.10.2024).
- Morris D., *Philosophy of Mind*, [w:] *The Edinburgh Companion to Twentieth-Century Philosophers*, red. C.V. Boundas, Edinburgh University Press, Edinburgh 2007, s. 531-544.
- Morris S., *WADs, Bots and Mods: Multiplayer FPS Games as Co-creative Media*, 2003

- Mounier E., *Wprowadzenie do egzystencjalizmów oraz wybór innych prac*, Społeczny instytut Wydawniczy Znak, Warszawa 1964.
- Murawski R., *Filozofia matematyki. Zarys dziejów*, Państwowe Wydawnictwo Naukowe, Warszawa 1995.
- Murawski R., *Wstęp. Renesans empiryzmu we współczesnej filozofii matematyki*, [w:] *Współczesna filozofia matematyki: wybór tekstów*, red. R. Murawski, Państwowe Wydawnictwo Naukowe, Warszawa 2002, s. 215-217.
- Musioł A., *Dwa projekty antynaturalizmu metodologicznego na przykładzie filozofii Heinricha Rickerta i Ernsta Cassirera*, „Ruch filozoficzny”, 71, nr 2, 2014, s. 173-186.
- Musioł A., *Neokantyzm epistemologiczny i ontologiczny Ernst Cassirer – Martin Heidegger*, Wydawnictwo Uniwersytetu Śląskiego, Katowice 2020.
- Nęcka E., Orzechowski J., Szymura B., Wichary Sz., *Psychologia poznawcza*. Wydanie nowe, Państwowe Wydawnictwo Naukowe, Warszawa 2020.
- Nęcka E., Orzechowski J., Szymura B., *Psychologia poznawcza*, Państwowe Wydawnictwo Naukowe, Warszawa 2013.
- Nicolas A., *Spokil, langue internationale; grammaire, exercices, les deux dictionnaires*, Maloine, Paris 1904,
<https://archive.org/details/spokillanguageint00nicogoog/page/n9/mode/2up?q=aprio>
 (dostęp 07.04.2022).
- Nietzsche F., *Wola mocy. Próba przemiany wszystkich wartości*, Wydawnictwo „bis”, Warszawa 1993.
- Nissbett R., *Geografia myślenia. Dlaczego ludzie Wschodu i Zachodu myślą inaczej?*, Smak Słowa, Sopot 2009.
- Nixon R., *Ubuntu: Up and Running: A Power User's Desktop Guide*, O'Reilly, Sebastopol 2010.
- Nowaczyk A., *Poławianie sensu w filozoficznej głębi*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2006.
- Nowak T., *Język w świetle odkryć nauki*, Petrus, Kraków 2011.
- Odrawąż-Sypniewska J., *Przewodnik po filozofii języka*, Wydawnictwo WAM, Kraków 2016.

- Oldag D., DeLucchi M., *A Python Project Template for Healthy Scientific Software*, "Notes AAS", 8, nr 5, 2024, <https://iopscience.iop.org/article/10.3847/2515-5172/ad4da1/meta>,
- Opoczyńska M., *Pasaże egzystencji. Trony psychoterapii i inne strony*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2019.
- Ossowska M., *Ethos rycerski i jego odmiany*, Państwowe Wydawnictwo Naukowe, Warszawa 1986.
- Paivio A., *Mental Representations*. Oxford University Press 1990, oxford.universitypressscholarship.com (dostęp dnia 18.07.2022).
- Parszutowicz P., *Fenomenologia form symbolicznych. Podstawowe pojęcia i inspiracje „późnej” filozofii Ernsta Cassirera*, Wydawnictwo IFiS PAN, Kęty 2011.
- Parszutowicz P., *Wstęp*, [w:] E. Cassirer, *Substancja i funkcja. Badania nad podstawowymi problemami krytyki poznania*, tłum. i wstępem opatrzył P. Parszutowicz, Wydawnictwo Marek Derewiecki, Kęty 2008, s. 2-10.
- Pawelec A., *Romantyczna teoria języka i metafory*, [w:] Charlesa Taylora *wizja nowoczesności. Rekonstrukcje i interpretacje*, red. Ch. Grabowski, J. Hudzik, J. Kłos, Oficyna Wydawnicza Łośgraf, Warszawa 2012, s. 261-273.
- Pei M., *One language for the world*, Biblo & Tannen, New York 1958.
- Penn G., Kiparsky P., *On Panini and the Generative Capacity of Contextualized Replacement Systems*, „Conference: Proceedings of COLING”, 2012, s. 1-8, <https://web.stanford.edu/~kiparsky/Papers/panini-1.pdf> (dostęp dnia: 01.05.2023).
- Peterson P., *Koko's story*, Scholastic Press, New York 1987.
- Piaget J., *Psychologia i epistemologia*, Państwowe Wydawnictwo Naukowe, Warszawa 1977.
- Piechocki A., *Haker: Czarny, Szary, a może Biały. Co to jest "white hacking" i czy jest legalny?*, „Gazeta Prawna”, 2020 <https://prawo.gazetaprawna.pl/artykuly/1447091,white-hacking-odpowiedzialnosc-karna.html> (dostęp dnia 15.03.2024).
- Piękny kod. Tajemnice mistrzów programowania*, red. A. Oram, G. Wilson, Helion, Gliwice 2008.

- Piszkalski H., *Problem „sytuacji granicznych” w ujęciu Karla Jaspersa*, „*Analecta Cracoviensia*”, 10, 1973, s. 99-116, <https://core.ac.uk/download/pdf/299330303.pdf> (dostęp dnia 22.10.2024).
- Ples M., *O micie politycznym w kontekście symbolizmu Ernsta Cassirera*, „*Filo–Sofija*”, nr 11, 2010, s. 73-91.
- Postanalytic and Metacontinental Crossing Philosophical Divides*, red. J. Reynolds, J. Chase, J. Williams, E. Mares, Wydawnictwo Continuum, New York 2010.
- Precht P., *Wprowadzenie do filozofii języka*, Wydawnictwo WAM, Kraków 2007.
- Pryciak M., *Prawo do prywatności*, „*Studia Erasmiana Wratislaviensia*”, 4, 2010, s. 211-229.
- Psychologia kliniczna*, red. L. Cierpiałkowska, H. Sęk, Państwowe Wydawnictwo Naukowe, Warszawa 2016.
- Pyritz B., *Craftsmanship versus engineering: Computer programming—An art or a science?*, „*Bell Labs Technical Journal*”, 8, nr 3, 2003, s. 101-104.
- Rahimi A., Gupta R. K., *Hardware/Software Codesign for Energy Efficiency and Robustness: From Error-Tolerant Computing to Approximate Computing*, [w:] *Dependable Embedded Systems*, red. J. Henkel, N. Dutt, Springer, Cham 2020, s. 527-543, https://link.springer.com/chapter/10.1007/978-3-030-52017-5_22 (dostęp dnia 05.07.2024).
- Rastogi A., Nagappan N., Gousios G., van der Hoek A., *Relationship between geographical location and evaluation of developer contributions in github*, *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* 2018, s. 1-8, <https://dl.acm.org/doi/pdf/10.1145/3239235.3240504> (dostęp dnia 22.08.2024).
- Raube S., *Kultura jako ekwilibrium. Krytyczna filozofia Ernsta Cassirera*, „*IDEA – Studia nad strukturą i rozwojem pojęć filozoficznych*” 26, 2014, s. 143-155.
- Raymond E., *Katedra i bazar*, <https://web.archive.org/web/20180326202640/http://www.linux-community.pl/node/4> (dostęp dnia 22.08.2024).
- Raymond E., *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly, Cambridge 1999.

- Rhode M., *Enaction, Embodiment, Evolutionary Robotics. Simulation Models for a Post-Cognitivist Science of Mind*, Atlantis Press, Amsterdam-Paris 2010.
- Richards H., *Engineering Education after ChatGPT*, Paper presented at 14th Annual First-Year Engineering Experience (FYEE) Conference, University of Tennessee in Knoxville, Tennessee 2023.
- Richardson C., Rymer J., *Vendor Landscape: The Fractured, Fertile Terrain Of Low-Code Application Platforms*, „Forresster”, 2016.
- Ricœur P., *Prawda i kłamstwo*, [w:] *Podług nadziei*, P. Ricœur, PAX Warszawa 1991, s. 57-86.
- Ricœur P., *Czas i opowieść. Czas opowiadany*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2008.
- Ricœur P., *Czas i opowieść. Intryga i historyczna opowieść*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2008.
- Ricœur P., *Doświadczenie estetyczne*, [w:] P. Ricœur, *Krytyka i przekonanie. Rozmowy z Francois Azouvim i Markiem de Launay*, tłum. M. Drwięga, Wydawnictwo KR, Warszawa 2003, s. 245-266.
- Ricœur P., *Drogi rozpoznania*, Wydawnictwo Znak, Kraków 2004.
- Ricœur P., *Du texte à l'action, Essais d'herméneutique*, t. 2, Editions du Seuil, Paris 1986.
- Ricœur P., *Egzystencja i hermeneutyka*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 182-204.
- Ricœur P., *Fenomenologia i hermeneutyka. Wychodząc od Husserla*, tłum. M. Drwięga, „Kwartalnik Filozoficzny”, 24, nr 3, 1996, s. 149-181.
- P. Ricœur, *Freedom and Nature: The Voluntary and the Involuntary*, tłum. Erazim Kohák, Northwestern University Press, Evanston 1966.
- Ricœur P., *Hermeneutyczna funkcja dystansu*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 224-246.
- Ricœur P., *Hermeneutyka symboli a refleksja filozoficzna – II*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 105-124.

- Ricœur P., *Interpretacja a refleksja: konflikt hermeneutyczny*, „Studia Filozoficzne”, nr 9 (250), 1986, s. 137-161.
- Ricœur P., *Język jako dyskurs*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 65-96.
- Ricœur P., *Konflikt hermeneutyk: epistemologia interpretacji*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 124-147.
- Ricœur P., *Ku hermeneutyce krytycznej*, „Pamiętnik Literacki”, 83, nr 1, 1992, s. 182-192.
- Ricœur P., *Le conflit des interprétations. essais d'herméneutique*, Éditions du Seuil, Paris 1969.
- Ricœur P., *Le Modèle du texte: l'action sensée considérée comme un texte*, [w:] *Du texte à l'action, Essais d'herméneutique*, t. 2, Editions du Seuil, Paris 1986, s. 183-213.
- Ricœur P., *Mowa i pismo*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 96-123.
- Ricœur P., *O interpretacji. Esej o Freudzie*, Wydawnictwo KR, Warszawa 2008.
- Ricœur P., *O pewnej egzystencjalnej interpretacji Freuda*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 227-245.
- Ricœur P., *O sobie samym jako innym*, przeł. B. Chełstowski, Państwowe Wydawnictwo Naukowe, Warszawa 2003.
- Ricœur P., *On interpretation*, [w:] *From text to action; Essays in Hermeneutics*, II, red. R. Kearney, Northwestern University Press, Evanston 2007, s. 1-25.
- Ricœur P., *Osoba: struktura etyczna i moralna*, [w:] *Zawierzyć człowiekowi*, red. Zbigniew Brzeziński, Znak, Kraków 1991, s. 37-50.
- Ricœur P., *Pamięć, Historia, Zapomnienie*, Universitas, Kraków 2007, tłum Janusz Margański.
- Ricœur P., *Philosophie de la volonté I. Le volontaire et l'involontaire*, Editions du Seuil, Paris 1950.
- Ricœur P., *Przemoc i język*, „Ethos”, 27, nr 2, 2014, s. 141-148.

- Ricœur P., *Przyswojenie*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 272-290.
- Ricœur P., *Refleksja dokonana. Autobiografia intelektualna*, Wydawnictwo Antyk, Kęty 2005.
- Ricœur P., *Struktura, wyraz, zdarzenie* [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 296-314.
- Ricœur P., *Symbol daje do myślenia*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 58-76.
- Ricœur P., *Symbolika zła*, PAX, Warszawa 1986.
- Ricœur P., *The model of the text: meaningful action considered as a text*, [w:] *From text to action; Essays in Hermeneutics*, II, red. R. Kearney, Northwestern University Press, Evanston 2007, s. 144-168.
- Ricœur P., *Wydarzenie i sens w mowie*, tłum. E. Bieńkowska, „Teksty: teoria literatury, krytyka, interpretacja” nr 6, 1972, s. 103-118.
- Ricœur P., *Wyjaśnianie i rozumienie*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 156-180.
- Ricœur P., *Wyzwanie semiologiczne: problem podmiotu*, [w:] *Egzystencja i hermeneutyka. Rozprawa o metodzie*, red. S. Cichowicz, Instytut Wydawniczy PAX, Warszawa 1985, s. 245-276.
- Ricœur P., *Zadanie hermeneutyki*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 191-224.
- Ricœur P., *Zło. Wyzwanie rzucone filozofii i teologii*, Instytut Wydawniczy PAX, Warszawa 1992.
- Rocha M., *Cognitive, Embodied or Enacted? Contemporary Perspectives for HCI and Interaction*, Reader, Plymouth 2012.
- Rode B., *Czy zwierzęta mają język*, „Zoophilologica. Polish Journal of Animal Studies”, nr 3, 2017, s. 129-142.
- Rodriquez M., *Research: Quantifying GitHub Copilot's impact on code quality*, <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-code-quality/> (dostęp dnia 25.10.2024).

- Rojas R., Göktekin C., Friedland G., *Plankalkül: The First High-Level Programming Language and its Implementation*, Institut für Informatik, Freie Universität Berlin 2020, Technical Report B-3/2000, <http://ftp.mi.fu-berlin.de/pub/reports/TR-B-00-03.pdf>, (dostęp 06.07.2022).
- Rorty R., Introduction, [w:] *The linguistic turn: recent essays in philosophical method*, The University of Chicago Press, Chicago; London 1968, 1-41, <https://archive.org/details/linguisticturnre0000unse/page/n7/mode/2up?view=theater> (dostęp dnia 05.06.2023)
- Rosen J., *This is your brain on code: JHU researchers decipher neural mechanics of computer programming*, HUB Johns Hopkins University, 2020 https://hub.jhu.edu/2020/12/17/brain-activity-while-reading-code/?fbclid=IwAR1_Ri-3sNM0RpczAkltuH9hoqucOmF0k2J1aPkfYfN7vAVK4YY9NidtJ1o (dostęp dnia 7.10.2023).
- Rosner K., *Hermeneutyka jako krytyka kultury*, PIW, Warszawa 1991.
- Rosner K., *Paul Ricœur – filozoficzne źródła jego hermeneutyki*, [w:] *Język, tekst, interpretacja. Wybór pism*, red. K. Rosner, PIW, Warszawa 1989, s. 5-63.
- Rosner K., *Narracja, tożsamość, czas*, Universitas, Kraków 2003.
- Rowiński T., Kowalska-Dąbrowska M., Strus W., *Pomiar patologicznych cech osobowości według DSM-5: polska adaptacja Inwentarza Osobowości PID-5. Część I – podstawy teoretyczne*, „Psychiatria Polska”, 53, nr 1, 2019, s. 7-22.
- Rowlands M., *The New Science of the Mind: From Extended Mind to Embodied Phenomenology*, MIT Press, London 2010.
- Rutvik D., Riccardi N., *Cognitive Neuroscience of Language*, [w:] *The Routledge Handbook of Cognitive Linguistics*, red. J.R. Taylor, X. Wen, Routledge, New York 2021, s. 1-35, https://www.researchgate.net/publication/350495833_Cognitive_Neuroscience_of_Language (dostęp dnia 1.09.2023).
- Sacha K., *Inżynieria oprogramowania*, Państwowe Wydawnictwo Naukowe, Warszawa 2010.
- Sady W., *Spór o racjonalność naukową. Od Poincarégo do Laudana*, Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika, Toruń 2013.

- Sauer R., *Why develop open-source software? The role of non-pecuniary benefits, monetary rewards, and open-source licence type*, Oxford Review of Economic Policy”, 23, nr 4, 2007, s. 605-619.
- Schultz D. P., Schultz S. E., *Historia współczesnej psychologii*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 2008.
- Scott-Baumann A., *Ricreur and the Hermeneutics of Suspicion*, Bloomsbury Academic, London, 2009.
- Searle J. R., *Ontologia społeczna. Niektóre podstawowe zasady*, [w:] *Język, dyskurs, społeczeństwo: zwrot lingwistyczny w filozofii społecznej*, red. L. Rasiński, Państwowe Wydawnictwo Naukowe, Warszawa 2009, s. 202-227.
- Searle J., *Umysł, język, społeczeństwo*, WAB, Warszawa 1999.
- Segado-Boj F., Martín-Quevedo J., Prieto-Gutiérrez J., *Jumping over the paywall: Strategies and motivations for scholarly piracy and other alternatives*, „Information Development”, 40, nr 3, 2024, s. 442-460.
- Sendyka R., *Od kultury ja do kultury siebie. O zwrotnych formach w projektach tożsamościowych*, Universitas, Kraków 2015.
- Sentencing Remarks of Mr Justice Hilliard Central Criminal Court*, 5 October 2023. <https://www.judiciary.uk/wp-content/uploads/2023/10/R-v-Chail-sentencing-050923.pdf> (dostęp dnia 18.10.2024).
- Siekańska K., Nowaczyk-Basińska K., *Przeżywamy nie swoje żaloby. Wywiad z Katarzyną Nowaczyk-Basińską*, Weekend.Gazeta,
- Siewczyński B., *Natchnieni rzemieślnicy 4.0.*, „Architectus”, nr 64 ,2020, s. 89-96.
- Simms K., *Paul Ricœur*, Routledge, London 2003.
- Simonyi C., *Hungarian notation*. MSDN Library, 1999, <https://cld.pt/dl/download/4275816b-59bc-4fe9-96a3-f2c7a24e9246/GnuCOBOL/MISC/Manuals%20%26%20Guides/C-C%2B%2B%20Miscellaneous/Hungarian%20Notation%20%28by%20Charles%20Simonyi%29%20%281999%29.pdf> (dostęp dnia: 01.05.2023).
- Singleton T., Gerken T., MacMahon L., *How a chatbot encouraged a man who wanted to kill the Queen*, BBC, 2023, <https://www.bbc.com/news/technology-67012224> (dostęp dnia

18.10.2024).

Sinnott J. D., *The development of logic in adulthood: postformal thought and its applications*, Springer, New York 1998.

Słomski W., *Człowiek jako podmiot działań etycznych*, „Społczeństwo i Edukacja”, nr 1, 2008, s. 5-14.

Soćko-Mucha A., *Wspólnota śmiechu muzealników (Szkic do dalszych studiów)*, „Zeszyty Naukowe Uniwersytetu Jagiellońskiego Prace Etnograficzne”, 48, nr 3, 2000, s. 253-268.

Sójka J., *O koncepcji form symbolicznych Ernsta Cassirera*, Państwowe Wydawnictwo Naukowe, Warszawa 1988.

Sontag S., *Przeciw interpretacji i inne eseje*, S. Sontag, Wydawnictwo Karakter, Kraków 2018.

Stammel S., *Beyond syntax, towards subculture: on the power and potentiality of deep open source communities*, 2018, <https://www.steph.ai/opensource/beyond-syntax-towards-subculture-on-the-power-and-potentiality-of-deep-open-source-communities/> (dostęp dnia 8.08.2024).

Steiner G., *Po wieży Babel*, Wydawnictwo Aletheia, Warszawa 2018.

Steinhart J. E., *Sekretne życie programów. Zrozumieć komputery – tworzyć lepsze kody*, Państwowe Wydawnictwo Naukowe, Warszawa 2021.

Stemplewska-Żakowicz K., *Metody jakościowe, metody ilościowe: Hamletowski dylemat czy różnorodność wyboru?* „Roczniki Psychologiczne”, 13, nr 1, 2010, s. 87-96.

Stępień-Nycz M., *Emocje w języku, Komunikowanie i odczytywanie emocji wyrażanych w języku w biegu życia*, [w:] *Język w biegu życia* red. M. Kielar-Turska, Harmonia Universalis, Gdańsk 2019, s. 449-469.

Strawson G., *Against narrativity*, „Ratio”, 17, nr 4, 2004, s. 428-452.

Strelau J., Doliński D., *Psychologia T. 2*, Gdańskie Wydawnictwo Psychologiczne, Gdańsk 2008.

Stróżewski W., *Istnienie i wartość*, Znak, Kraków 1981.

Strzeszewski C., *Praca ludzka: zagadnienie społeczno-moralne*, Towarzystwo Naukowe KUL, Lublin 1978.

- Stuart T., *Understanding Computation: From Simple Machines to Impossible Programs*, O'Reilly Media Inc, Sebastopol 2013.
- Szadura J., O ludowej etiologii wrodzonego kalectwa, „Literatura ludowa”, nr 6, 2018, 70-82.
- Sztompka P., *Socjologia zmian społecznych*, Znak, Kraków 2010.
- Sztompka P., *Socjologia. Analiza społeczeństwa*, Kraków 2012, Znak.
- Szulakiewicz M., *Czas i to co ludzkie, Szkice z chronozofoii i kultury*, Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika, Toruń 2011.
- Szulakiewicz M., *Hermeneutyka - czyli filozofia jako rozumienie*, [w:] *Filozofia Współczesna*, red. L. Gawor, Z. Stachowski, Lublin 2006, s. 297-325.
- Szulakiewicz M., *Obecność filozofii transcendentalnej*, Wydawnictwo Uniwersytetu Mikołaja Kopernika, Toruń 2020.
- Szymańska B., *Co to jest „filozofia kultury”*, [w:] *Co to jest filozofia kultury*, red. Z. Rosińska, J. Michalik, Wydawnictwo Uniwersytetu Warszawskiego, Warszawa 2006, s. 15-30.
- Tatarkiewicz W., *Droga przez estetykę*, Warszawa 1972.
- Taylor Ch., *The Language Animal. The Full Shape of the Human Linguistic Capacity*, The Belknap Press Cambridge, London 2016.
- Taylor Ch., *Etyka autentyczności*, Znak, Kraków, 1996.
- Taylor Ch., *Teoria znaczenia*, [w:] *Język, dyskurs, społeczeństwo*, red. L. Rasiński, Państwowe Wydawnictwo Naukowe, Warszawa 2009, s. 147-187.
- Taylor Ch., *Źródła podmiotowości. Narodziny tożsamości nowoczesnej*, Państwowe Wydawnictwo Naukowe, Warszawa 2001.
- Temkin D., *Language Without Code: Intentionally Unusable, Uncomputable, or Conceptual Programming Languages*, „Journal of Science and Technology of the Arts”, 9, nr 3, 2017, s. 83-91.
- Terrell J., Kofink A., Middleton J., *Gender bias in open source: Pull request acceptance of women versus men*, „PeerJPreprints”, 2016
https://www.researchgate.net/publication/308716997_Gender_bias_in_open_source_Pull_request_acceptance_of_women_versus_men (dostęp dnia 22.08.2024).

- Tischner J., *Hermeneutyka i język*, „Analecta Cracoviensia”, 5, 1973, s. 347-358.
- Trochimska-Kubacka B., *Koncepcja sztuki w filozofii kultury Ernsta Cassirera*, „Studia Philosophica Wratislaviensia”, 14, nr 3, 2019, s. 81-99.
- Trompenaars F., Hampden-Turner Ch., *Siedem wymiarów kultury: znaczenie różnic kulturowych w działalności gospodarczej*, Oficyna Ekonomiczna Grupa Wolters Kluwer, Kraków 2002.
- Truwant S., *The concept of 'function' in Cassirer's historical, systematic, and ethical writings*, [w:], *The Philosophy of Ernst Cassirer*, red. T. Friedman, S. Lufts, Marquette University, Milwaukee 2015, s. 289-312.
- Tsay J., Dabbish L., Herbsleb J., *Influence of Social and Technical Factors for Evaluating Contribution in GitHub*, [w:] *Proceedings of the 36th international conference on Software engineering*. ACM 2014, s. 356-366.
- Tucholska S., *Mądrość jako przejaw dojrzałości osobowościowej*, „Rocznik Filozoficzny Ignatianum”, 22, nr 1, 2016, s. 230-253.
- Tymoczko T., *Problem czterech barw i jego znaczenie filozoficzne*, [w:] *Współczesna filozofia matematyki*, red. R. Murawski, Państwowe Wydawnictwo Naukowe, Warszawa 2002, s. 310-340.
- Vaillant G., *Pozytywne zdrowie psychiczne: czy istnieje definicja międzykulturowa?*, „Postępy Psychiatrii i Neurologii”, 21, nr 4, 2012, s. 229-239.
- Van Roy P., S. Haridi, *Programowanie. Koncepcje, techniki i modele*, Helion, Gliwice 2005.
- Vopel K. W., *Warsztaty – skuteczna forma nauki*, Jedność, Kielce 2004.
- Wadsworth B., *Teoria Piageta - poznawczy i emocjonalny rozwój dziecka*, WSiP, Warszawa 1988.
- Wall L., Christiansen T., Orwant J., *Programming Perl. Third Edition*, O'Reilly, Sebastopol 2000.
- Warmbier A., *Tożsamość, narracja i hermeneutyka siebie. Paula Ricœura filozofia człowieka*, Universitas, Kraków 2018.

- Wegner P., *Research paradigms in computer science*, Proceedings of the 2nd International Conference on Software Engineering, San Francisco 1979, <https://tiny.pl/whbfg> (dostęp dnia 23.07.2022).
- Węgrzyn M., *Ja-Ty-Dzieło jako źródło poznania. Między neokantyzmem a hermeneutyką: Cassirer i Ricœur*, „Hybris”, 50, nr 3, 2020, s. 19-43 <http://www.czasopisma.uni.lodz.pl/hybris/article/view/14467/13821> (dostęp dnia 22.08.2023).
- Whitman W., *Song of Myself* (1892 version), <https://www.poetryfoundation.org/poems/45477/song-of-myself-1892-version> (dostęp dnia 01.03.2023).
- Whitman W., *Pieśń o mnie i inne utwory*, tłum. A. Szuba, Wydawnictwo Miniatura, Kraków 1996.
- Wilder R., *Kulturowa baza matematyki*, [w:] *Współczesna filozofia matematyki. Wybór tekstów*, red. R. Murawski, Warszawa 2002, Państwowe Wydawnictwo Naukowe, s. 275-292.
- Witosz B., *Lingwistyczne koncepcje tekstu wobec wyzwań komunikacji wirtualnej*, [w:] *Język w internecie. Antologia*, red. M. Kita, I. Loewe, Wydawnictwo Uniwersytetu Śląskiego, Katowice 2016, s. 101-113.
- Wittgenstein L., *Tractatus logico-philosophicus*, Państwowe Wydawnictwo Naukowe, Warszawa 1997.
- Wojan K., *Języki sztuczne. Zapotrzebowanie społeczeństw czy fantazje jednostki*, Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 2025.
- Wolicka E., *Narracja i egzystencja*, Wydawnictwo KUL, Lublin 2010.
- Wolska D., *Odzyskać doświadczenie. Sporny temat humanistyki współczesnej*, Universitas, Kraków 2012.
- Wulf W. A., *Are We Scientists or Engineers?* *ACM Computer Surveys*, 27, nr 1, 1995, s. 55-57 <https://tiny.pl/whbjh> (dostęp dnia 23.07.2022).

- Yegge S., *Execution in the Kingdom of Nouns*, <https://steve-yegge.blogspot.com/2006/03/execution-in-kingdom-of-nouns.html> (dostęp dnia 11.01.2023).
- Zagrodzka J., *Neurofizjologiczne mechanizmy zachowania emocjonalnego*, [w:] *Mózg a zachowanie*, red. T. Górski, A. Grabowska, J. Zagrodzka, Państwowe Wydawnictwo Naukowe, Warszawa 2012, s. 396-417.
- Żardecka-Nowak M., *Autentyczność zawiedzionych nadziei, czyli o wygasaniu pewnego moralnego ideału w ponowoczesnym społeczeństwie masowym*, „Logos i Ethos”, 18, nr 1, 2012, s. 81-104.
- Zięba A., *Język a kultura. Komunikacja werbalna w wybranych polskich, brytyjskich i amerykańskich programach telewizyjnych*, „Investigationes Linguisticae”, 16, 2008, s. 250-263.
- Zimbardo P., R. Johnson, V. McCann, *Psychologia. Kluczowe koncepcje. Motywacja i uczenie się*, Państwowe Wydawnictwo Naukowe, Warszawa 2010.
- Zimbardo P., R. Johnson, V. McCann, *Psychologia. Kluczowe koncepcje. Struktura i funkcje świadomości*, Państwowe Wydawnictwo Naukowe, Warszawa 2010.
- Ziwei J., Nayeon L., *Survey of Hallucination in Natural Language Generation*, ACM Computing Surveys, <https://dl.acm.org/doi/pdf/10.1145/3571730>, (dostęp dnia 11.12.2024).
- Żydek-Bednarczuk U., *Hipertekst w perspektywie lingwistycznej, komunikacyjnej i kulturowej*, [w:] *Język w internecie. Antologia*, red. M. Kita, I. Loewe, Wydawnictwo Uniwersytetu Śląskiego, Katowice 2016, s. 113-123.

Spis rysunków

Rys. 1. Porównanie obszerności (rozbudowania wypowiedzi) przykładowego fragmentu kodu napisanego w Ruby (czerpiącego z języka naturalnego) i odpowiadającego mu znaczeniowo kodu w języku lambda calculus (system formalny).....	77
Rys. 2. Fragment profilu Linkedin Janusza Jeremiasza Filipiaka, dyrektora działu badawczo-rozwojowego w firmie Comarch.....	78
Rys. 3. Wiersz Sharon Hopkins	82
Rys. 4. Żartobliwa historyjka z krainy Javaland zapisana w formie przypominającej kod lecz nim nie będąca, zestawiona ze swoją inspiracją, czyli ludową wyliczanką, znaną jako “For want of a nail”	90
Rys. 5. Przykład użycia funkcji insanity (niepoczytalność)	101
Rys. 6. Przykład pojawienia się w kodzie źródłowym słowa cripple	102
Rys. 7. Nawiasy egipskie porównane do hieroglifu.....	107
Rys. 8. Uproszczony diagram rozwoju języków programowania	116
Rys. 9. Rozbudowany i coraz mniej czytelny diagram rozwoju języków programowania	117
Rys. 10. Pomniejszony i już nieczytelny, ze względu na mnogość powiązań i języków, diagram rozwoju języków programowania uwzględniający możliwie wiele, tych znanych autorowi drzewa - Pascalowi Rigaux	117
Rys. 11. Zmiany popularności języków programowania	118
Rys. 12. Przykład kodu napisanego w PHP z użyciem języka narodowego. Zawiera nazwę zmiennej „gżegżółka” (będącego ilustracją trudnego, polskiego słowa) oraz ideogram oznaczający miłość „<3”	131
Rys. 13. Podświetlona składnia programu napisanego w języku Whitespace. „Białe znaki” zostały zakolorowane, w celu ukazania niewidocznej składni	133
Rys. 14. Program „Hello,World!” zapisany w „zagadkowym” języku ezoterycznym brainfuck.....	134
Rys. 15. Fragment kodu w języku Unlambda autorstwa Davida Madore.....	135
Rys. 16. Przykład użycia try, throw, catch w kodzie C++	141
Rys. 17. Słynny bug (pol. robal), którego znalezienie w log book’u przypisuje się Grace Hopper – popularyzatorce terminu bug i debugging.....	144
Rys. 18. Fragmenty kodu źródłowego autorstwa Dana MacKinley’a, w których autor obraża poprzedniego twórcę kodu. Zostały wyboldowane i podkreślone fragmenty wskazujące na niechęć autora do byłego współpracownika.....	149
Rys. 19. Manifest rzemieślników piszących oprogramowanie.....	166
Rys. 20. Tak zwany „Jargon File” (słownik gwary hakerskiej), w którym spisano zasady etyki hakerskiej.....	183
Rys. 21. Zrzut ekranu przeprowadzonej konwersacji z ChatGPT	210
Rys. 22. Możliwość wyrażenia przez człowieka oceny dla udzielonej przez Microsoft Copilot odpowiedzi i trenowania między innymi w ten sposób AI	211
Rys. 23. Po lewej stronie kod wygenerowany przez Copilot w C, po prawej napisany przez programistę w C. Algorytm oblicza szybką odwrotność pierwiastka kwadratowego – tzw. szybki InvSqrt().....	214
Rys. 24. Kod wygenerowany przez Copilot w C i C#. Algorytm oblicza szybką odwrotność pierwiastka kwadratowego – tzw. szybki InvSqrt().....	214

Rys. 25. Kod wygenerowany przez Copilot w Pythonie i C. Algorytm oblicza kolejną liczbę ciągu Fibonacciego	215
Rys. 26. Zrzut ekranu przeprowadzonej konwersacji z ChatGPT na temat jego praw autorskich	218
Rys. 27. Rozmowa z botem konwersacyjnym, który traktowany jest jako wsparcie psychiczne	221