

Prof. dr hab. Józef Bremer SJ
Uniwersytet Ignatianum w Krakowie
Wydział Filozoficzny
ul. Kopernika 26
31-501 Kraków
josef.bremer@ignatianum.edu.pl

Kraków, 05 września 2025 roku

**P.T.
Uniwersytet Komisji Edukacji Narodowej w Krakowie
Wydział Nauk Humanistycznych
Instytut Socjologii
Kraków**

Recenzja rozprawy doktorskiej mgr Anny Bajer
*Język kodu źródłowego programów komputerowych w ujęciu hermeneutyki dyskursu
Paula Ricoeura i filozofii form symbolicznych Ernsta Cassirera,*
napisanej w Instytucie Socjologii Uniwersytetu Komisji Edukacji Narodowej
w Krakowie pod kierunkiem prof. dr hab. Katarzyny Gurczyńskiej-Sady
(Kraków 2025). Rozprawa liczy 266 stron, zawiera łącznie 1300 przypisów

Kod źródłowy to zestaw instrukcji, które programista pisze w celu stworzenia oprogramowania. Każda instrukcja (zwana również algorytmem) jest napisana w określonym języku programowania, takim jak np. Python, HTML, C++ lub Java. Innymi słowy, kod źródłowy jest niczym szczegółowy przepis, którego oprogramowanie komputerowe musi przestrzegać. Opisuje każdą czynność, jaka winna zostać wykonana, aby można było wykonać dane zadanie lub uruchomić program. I robi to w języku, który komputer „rozumie”. Niezależnie od tego, czy korzystamy z aplikacji na telefon komórkowy, odwiedzamy stronę internetową, czy używamy systemu operacyjnego na komputerze stacjonarnym lub laptopie – za wszystkim kryje się kod źródłowy. W swojej rozprawie doktorskiej Pani mgr Bajer podjęła się zadania przedstawienia języka kodu źródłowego programów komputerowych w ujęciu filozofii języka.

1. Treść rozprawy

Omawiana rozprawa składa się z Wstępu, czterech rozdziałów, Zakończenia, Streszczenia (Summary), Słowa kluczy (Keywords), obszernej bibliografii (głównie w języku angielskim) oraz Spisu rysunków.

We wstępie Doktorantka zaznacza, że „Celem niniejszej pracy jest zbadanie kodu źródłowego w oparciu o refleksję filozoficzną, a konkretnie – poprzez pryzmat dwóch podejść: hermeneutyki dyskursu Paula Ricoeura oraz filozofii form symbolicznych Ernsta Cassirera.” (s. 3). Poszerzając rozumienie celu swojej rozprawy, autorka dodaje, że starała się w możliwie pełny sposób ukazać językową działalność programistów w obszarze kodu źródłowego w jej trzech kluczowych wymiarach: przestrzeni publicznej, sferze indywidualnej oraz w kontekście standardów etycznych. Do każdego z tych wymiarów odwołuje się zarówno Cassirer, jak i Ricoeur (s. 7). Kod źródłowy zostaje przez autorkę przedstawiony jako artefakt kultury, poddający się interpretacji, a język rozumie nie jako narzędzie, ale jako przestrzeń twórczego bycia.

W rozdziale I („Zarys Problematyki”, s. 10–70) autorka wspomina rozróżnienie między filozofią analityczną a filozofią kontynentalną i odwołuje się do Charlesa Taylora podziału języka na dwie rodziny: deskrypcyjno-instrumentalne i konstytutywno-ekspresywne. W tej ostatniej rodzinie język traktowany jest jako „*locus* różnego rodzaju ujawnień” (s. 13). Zarówno Ricoeur, jak i Cassirer prezentują takie właśnie podejście, wychodząc poza funkcje opisywania oraz eksponując twórczy aspekt języka. Język staje się tym samym źródłem człowieczeństwa w przeciwieństwie do teorii instrumentalnych, gdzie służy procesom poznawczym. Autorka zaznacza, że od tekstów kodu źródłowego programów komputerowych wymaga się jednoznaczności, więc można by je uznać za wyraz spełnienia oczekiwań wobec języka z perspektywy podejścia deskrypcyjno-instrumentalnego. Proponowane przez Ricoeura i Cassirera podejścia mają pokazać, jak w języku konstytuuje się potrzeba bycia sobą, bycia z innymi oraz wyrażanie kwestii moralnych. Kwestie te Pani Bajer bada w kolejnych rozdziałach, śledząc wymiar ekspresji, ujawniania i konstytuowania w wybranych tekstach kodu źródłowego programów komputerowych. Po omówieniu rozumienia języka w hermeneutyce dyskursu Ricoeura i języka jako formy symbolicznej w filozofii Cassirera Doktorantka konkluduje, że podejścia te wpisują się w nurt filozofii kontynentalnej, wykraczając poza traktowanie języka wyłącznie jako narzędzia komunikacji i ukazując jego twórczy charakter. Z jednej strony Ricoeur i Cassirer różnią się pod względem oceny roli, jaką zajmuje język w strukturze ich refleksji filozoficznej, z drugiej strony łączy ich myślenie o ludzkiej potrzebie uczestniczenia w rzeczywistości. Tym samym podejścia Ricoeura i Cassirera uzupełniają się w narzędziowości, co autorka rozprawy stara się wykorzystać w rozważaniach odnośnie do kodu źródłowego programów komputerowych, powołując się na koncepcję języka dyskursu, jak i na koncepcję języka form symbolicznych. „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” (s. 50).

Wspomniane trzy kluczowe wymiary językowej działalności programistów w obszarze kodu źródłowego (publiczny, indywidualny i etyczny) zostają omówione w rozdziałach II–IV, z których każdy ma taką samą strukturę, podzieloną na trzy punkty: wymiar ekspresji, wymiar ujawniania i wymiar konstytuowania.

Rozdział II („Ustanowienie przestrzeni publicznej w języku”, s. 71–124) rozpoczyna się informacją, że autorka traktuje język kodu źródłowego programów komputerowych z perspektywy humanistycznej, zdając sobie sprawę, że nie można pomijać strukturalnych cech języka programowania, co podpada już pod nauki inżynieryjno-techniczne. Zajmuje się więc ekspresją językową, porównując możliwości wyrażania się w kodzie źródłowym z uwzględnieniem różnych języków programowania. Wychodząc poza stanowiska Ricoeura oraz Cassirera, skupia się na ekspresji artystycznej oraz na wartościach estetycznych obecnych w tekście kodu. Następnie autorka zaznacza, że „Analiza problemu [kodu źródłowego jako tekstu – przyp. J.B.] została dokonana Ricoeurowską «okrężną drogą» hermeneutycznego rozumienia, ale i z wykorzystaniem jednej z używanych przez Cassirera, kantowskich grup kategorii – mowa o «jakościowej» formie – czasie” (s. 101). Doktorantka kończy analizami wskazującymi na Cassirera społeczne, kulturowe elementy przyswajania zjawisk: „Poznanie świata kultury opiera się bowiem w ujęciu tego filozofa na trzech wymiarach rozumienia rzeczywistości zwanych fenomenami bazowymi lub inaczej prafenomenami (*Urphänomen*): nakierowaniu na ‘ja’, ‘ty’ i ‘to’. Ścieżka poznania rzeczywistości polegająca na wiedzy o sobie jest równorzędna i współzależna z pozostałymi wymiarami” (s. 123).

W rozdziale III („Artykulacja indywidualności”, s. 125–173) Doktorantka przedstawia indywidualny aspekt działań programistów, dostrzegalny w tym, że artykulacja ich własnego

sposobu widzenia świata jest możliwa również poprzez kod źródłowy. Widać to chociażby w wyborze stylu programistycznego, używanego słownictwa, a także w samym akcie kreacji ezoterycznych języków (*esolangów*) programowania (por. s. 132–134). W tej części rozprawy autorka przedstawia uwagi Cassirera związane z wyrażaniem afektu oraz Ricoeura tematem komunikowania doświadczenia. Konfrontując hermeneutykę dyskursu z myślą psychoanalityczną, ukazuje możliwość samoujawniania się programistów w tekstach, które oni tworzą. Zdaniem autorki, widoczne jest tu nawiązanie do mechanizmów psychicznych zaproponowanych w psychoanalizie (s. 151–153). Proces konstruowania tożsamości programistów zostaje przedstawiony poprzez ich działania językowe, w ramach hermeneutyki dyskursu i hermeneutyki narracji.

W Rozdziale IV („Rozpoznawanie standardów”, s. 174–225) Doktorantka analizuje wymiar etyczny działalności programistów w obszarze kodu źródłowego – tak istotny w okresie późnej twórczości zarówno Ricoeura, jak i Cassirera. Na przykładzie ekspresji językowej hakerów autorka podkreśla znaczenie świadomości standardów moralnych w środowisku programistycznym. „Ich działalność w językach programowania ma różnoraki charakter: przestępczy, rozrywkowy, ale także motywowany ideologicznie” (s. 177). Analizując działalność hakerów, Doktorantka posługuje się trójskładnikową strukturą działań etycznych według Ricoeura oraz Cassirerowską koncepcją ujarzmiania siły mitów (np. mitu hakera jako „kowboja klawiatury”). Wiąże się to namysłem nad źródłem działania programistów jako podmiotów etycznych, a tym samym nad etycznymi standardami wyrastającymi z wyznawanych przez nich wartości – troski o zdrowie, piękno i otwartość. Odwołując się do Terrella Bynuma definicji etyki komputerowej, autorka pisze, że rozważania etyki komputerowej 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” (s. 192). W tym kontekście ruch *open source* rozpatrywany jest przez autorkę w perspektywie fenomenologii człowieka działającego, autorstwa Ricoeura, natomiast kształtowanie się wokół tego ruchu subkultury – w świetle teorii form symbolicznych Cassirera. W trzeciej części rozdziału czytamy: „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” (s. 206). Poprzez porównanie biegłości językowych człowieka i sztucznej inteligencji mgr Bajer uwypukla zdolność programistów do posługiwania się językiem jako nośnikiem dyskursu. Jej zdaniem jedynie działania językowe podejmowane przez wolne, etyczne podmioty umożliwiają tworzenie wymiaru etycznego w języku kodu źródłowego, który z kolei „stanowi swoiste laboratorium, w którym można zaobserwować zdominowanie kultury przez formę nauki” (s. 223). Wychodząc od rozważań Cassirera, widzimy, zdaniem autorki, że ochrony przed tego typu zdominowaniem należy poszukiwać w obszarze refleksji etycznej.

W zakończeniu Doktorantka zaznacza, że filozofię Ricoeura i Cassirera potraktowała nie jako przedmiot egzegezy, lecz jako rzeczywistą perspektywę badawczą – narzędzie refleksji, które umożliwiło jej wgląd w działania językowe programistów (s. 229). Dwa zasadnicze motywy nieustannie przenikały rozważania jej rozprawy: działania programistów wyrażające ich doświadczenia oraz refleksyjność (odmiennie pojmowana przez Ricoeura i Cassirera). Swoje analizy Doktorantka domyka wskazaniem możliwych kierunków dalszych badań, zwłaszcza w kontekście wyzwań, jakie niesie ze sobą rozwój sztucznej inteligencji. Dodaje, że jej rozprawa nie rości sobie pretensji do wyczerpującego omówienia wszystkich zagadnień związanych z językową działalnością programistów w obszarze kodu źródłowego, a jej „Celem [...] było zbadanie, w jaki sposób język stwarza możliwość powstania kodu źródłowego jako artefaktu kulturowego” (s. 226), co sprowadza się do możliwie pełnego ukazania działalności

programistów w trzech kluczowych wymiarach: przestrzeni publicznej, sferze indywidualnej oraz w kontekście standardów etycznych, do których odwołują się zarówno Cassirer, jak i Ricoeur.

Pani mgr Anna Bajer analizuje teksty kodu źródłowego, traktując je jako językowy obszar ekspresji, konstytuowania i ujawniania kwestii kluczowych obszarze kulturowym, to jest w perspektywie społecznej, indywidualnej oraz etycznej. Obszar ten przedstawiła bazując na rozległych i zróżnicowanych podejściach do prac Ricouera i Cassirera. Sama zaznacza, że trudno określić, w jakim sensie sami programiści widzą swoje teksty jako istotne dla obszaru kulturowego, bowiem wobec tekstów kodu źródłowego programów komputerowych stawia się wymagania jednoznaczności, co mogłoby sugerować, że należy je uznać za realizujące oczekiwania wobec języka deskrypcyjno-instrumentalnego. Niemniej jednak, jej zdaniem, w ramach pracy programistów pojawiają się liczne przykłady kwestionujące założenia o jasności przedstawień, o analityczno-apriorycznej formie języków programowania. Stąd w swojej rozprawie Pani Bajer odwołuje się do konstytutywnego modelu języka, którego używa dla opisanego pozornie skomputeryzowanego obszaru rzeczywistości. Wskazując na Ricouera i Cassirera, pokazuje, jak w języku programowania konstytuują się indywidualne, społeczne i etyczne potrzeby człowieka.

2. Uwagi do treści rozprawy

(a) Zarówno refleksje Ricoeura, jak i Cassirera przynależą do nieanalitycznej tradycji filozofii języka. Odwołując się do rozróżnienia zaproponowanego przez Taylora, autorka zalicza owe refleksje do rodziny teorii konstytutywno-ekspresyjnych (zw. HHH), w przeciwieństwie do teorii deskryptywno-instrumentalnych (zw. HLC). Broniąc HHH Taylor zagłębia się w język metafor, symboli i rytuałów oraz w konstytutywną naturę języka w odniesieniu do moralności, religii i relacji międzyludzkich. Wszystkie te wymiary odzwierciedlają również jego długotrwałą troskę o obszar religii, etyki i wspólnoty. Należałoby wspomnieć, że obszary te były niegdyś silnie powiązane z religijnym językiem i praktykami etycznymi, które niestety są kwestionowane przez współczesną epistemologię i stają się czymś podejrzanym w racjonalnych rozważaniach. Proponowany przez Taylora encyklopedyczny opis języka sprawia ponadto, że jego prace są trudne w odbiorze, a coś z tej encyklopedyczności odnajdujemy w pracy mgr Bajer w jej krótkich odwołaniach do wielu różnych filozofów). Sama teoria Taylora jest skomplikowana, co utrudnia zrozumienie struktury jego argumentacji¹. Przykładowo, rzetelne porównanie analiz Taylora z fenomenologią hermeneutyczną Ricoeura i jego implikacje, np. dla badań nad naturą nauki chrześcijańskiej, mogłoby w znacznym stopniu wpłynąć na kierunek badań filozoficznych w nadchodzących latach – co oczywiście wychodzi poza zakres pracy Pani Bajer.

(b) Opracowania Taylora są przykładem przywiązania do hermeneutycznego stylu filozofii, który może wydawać się osobom nastawionym fenomenologicznie (nie wspominając o podejściu analitycznym) jako pozbawiony metody. Podczas gdy fenomenologia Husserla wykorzystuje metodę opisową w celu wyjaśnienia „esencji” lub niezmienności ejdetycznych, hermeneutyka Taylora dopuszcza stosowanie opisu anegdotycznego w każdym ogólnym projekcie nadawania sensu lub rozumienia. W szczególności podejście Taylora pozwala uwzględnić szereg zaskakujących cech pod sztandarem języka – uczuć, gestów, rytuałów, muzyki itp. – które są ewidentnie związane ze zjawiskiem języka w życiu codziennym (do

¹ Zob. np.: Andrew T.W. Hung, *Language, Meaning, Morality, and Community: A Philosophical Note on Charles Taylor's Phenomenology of Language*, „Philosophia Christi” 2017, t. 19, nr 1, s. 217–227, https://www.academia.edu/34701481/Philosophical_Note_on_Taylor_Phenomenology_of_Language; W. Clark Wolf, *Review of Charles Taylor, "The Language Animal: The Full Shape of the Human Linguistic Capacity"*, „Phenomenological Reviews” 2016, November 15, <https://doi.org/10.19079/pr.2016.11.wol>.

których także odwołuje się mgr Bajer), chociaż nie wydają się być niezbędne dla konceptualnej artykulacji języka. Taylor może zatem negować konkurencyjne ujęcia języka za brak uwzględnienia takich elementów w swoich rozważaniach, ale ich uwzględnienie wydaje się zależne od holizmu tak szerokiego, że jego granice zaczynają się rozpląwać. Czasami, moim zdaniem, krytyka Taylora pod adresem rywalizujących filozofów sprowadza się do skargi, że nie robią oni wszystkiego naraz. Wydaje się to przywracać stereotypowy kontrast między racjonalistyczną nauką a niejasnym humanizmem. Taylor nie dostrzega uzasadnionej racjonalnej warstwy w ludzkich znaczeniach, która jest właśnie niezbędna do krytycznej oceny tego rodzaju znaczeń.

(c) Pomimo przelotnych wzmianek Taylora o teorii „form symbolicznych” Cassirera, sam Cassirer rozwija bardziej systematyczny i prawdopodobnie bardziej przekonujący opis wyjątkowej funkcji „ludzkiego znaczenia” w konstytuowaniu kultury i wiedzy. Może to jedynie oznaczać, że teoria Taylora jest natury pedagogicznej, a być może nawet terapeutycznej (co pobrzmiewa w niektórych opisach autorki rozprawy). Można powiedzieć, że Taylor mówi nam to, co już wiemy, ale łatwo zapominamy, zwłaszcza pośród tak kuszącej dla filozofii tendencji do nadmiernego intelektualizowania. Próbuje pogodzić nas z naszym sposobem życia, chce abyśmy zauważyli, jak już żyjemy, a nie zmieniali naszego życia. Jego antynaturalistyczne stanowisko opiera się na poczuciu, że tak zwane teorie dehumanizujące domagają się zapomnienia o tym, że w swoich działaniach polegamy na strukturze ludzkich znaczeń, której nie da się wytłumaczyć z zewnątrz.

(d) Autorka zaznacza „że jej praca może stać się raportem o minionym świecie” (s. 172). Duże modele językowe (LLM), takie jak ChatGPT, Deepseek i inne, rewolucjonizują sposób tworzenia kodu, umożliwiając programistom zadawanie pytań w języku naturalnym i otrzymywanie wygenerowanych fragmentów kodu w ciągu kilku sekund. Obiecuje to zwiększenie wydajności tworzenia programów poprzez wyeliminowanie konieczności przeszukiwania dokumentacji lub forów internetowych w przypadku pytań lub napotkanych problemów. Możliwość ta jest również postrzegana jako narzędzie zwiększające dostępność programowania, umożliwiając początkującym programistom natychmiastowe otrzymywanie informacji zwrotnych i przykładów. Autorka rozprawy skupia się w tym miejscu na pytaniu o ewentualne zagrożenia tożsamości programistów, należy jednak dodać, że w tym kontekście na znaczeniu zyskuje głównie przegląd kodu i jego rewizja (*code review*, *peer review*). Stają się one ważniejsze niż kiedykolwiek wcześniej, nie tylko jako narzędzie do zapewnienia jakości i bezpieczeństwa kodu, ale także jako okazja do nauki dla programistów od swoich kolegów. Przegląd ludzki zapewnia dodatkową warstwę bezpieczeństwa, gwarantując, że kod nie tylko działa, ale jest również zgodny z najlepszymi praktykami i jest bezpieczny.

(e) Omawiana rozprawa ma nachylenie kulturoznawcze, warto więc zaznaczyć, że czynniki kulturowe mogą również wpływać na rozwlekłość i jednoznaczność składni języka. W kulturach ceniących bezpośredniość i przejrzystość komunikacji, takich jak przykładowo Stany Zjednoczone, Wielka Brytania czy Niemcy, języki programowania mogą skłaniać się ku bardziej jednoznacznej i rozwlekłej składni, z jasnymi deklaracjami zmiennych i definicjami funkcji. Z kolei kultury stawiające na zwięzłość i wydajność, takie jak przykładowo Japonia i Chiny, mogą preferować języki o bardziej zwartej i niejawnej składni, takie jak Ruby czy Perl. Różnice kulturowe w stylach komunikacji mogą również wpływać na sposób, w jaki języki programowania radzą sobie z błędami i wyjątkami w składni. W kulturach, w których ceni się zachowanie twarzy i unikanie konfrontacji, takich jak przykładowo Japonia i Korea, języki programowania mogą stosować bardziej pośrednie i subtelne sposoby sygnalizowania błędów, takie jak zwracanie uwagi na unikatowe wartości lub ustawianie globalnych priorytetów. W kulturach, w których priorytetem jest bezpośredniość i przejrzystość, języki mogą stosować bardziej jawną i agresywną obsługę błędów, na przykład zgłaszanie wyjątków i zawieszanie

programu². Istnieją języki eksperymentalne, takie jak EPL, ChinesePython, Yi Lang oraz Qi, które wykorzystują znaki chińskie jako słowa kluczowe lub składnię. Na przykład DeepSeek Coder, zaawansowany model sztucznej inteligencji opracowany przez DeepSeek AI, ma pomagać programistom w generowaniu, debugowaniu i optymalizacji kodu w ponad 80 językach. Wykorzystując zaawansowane sieci neuronowe, przetwarza zapytania tekstowe w celu generowania kodu, wyjaśniania go lub poprawiania już istniejącego. Składa się z serii modeli języka kodu, trenowanych od podstaw na 87% kodu i 13% języka naturalnego w języku angielskim i chińskim, przy czym każdy model jest wstępnie trenowany na tokenach 2T (na 2 bilionach tokenów w ponad 80 językach programowania), przy różnych rozmiarach modelu kodu, od 1 biliona do 33 bilionów (33B)³.

(f) Na str. 165–166 Doktorantka omawia „Manifest rzemieślników piszących oprogramowanie”, wspominając przy tym o regułach obowiązujących w średniowiecznych cechach rzemieślniczych⁴. Tego typu porównania spotyka się w literaturze przedmiotowej. Należałoby jednak nadmienić o krytyce sformułowań tych punktów manifestu, które nie rozróżniają programistów, którym naprawdę zależy na wartości, jaką wytwarzają swoją pracą, programistów, którym zależy na estetyce kodu, oraz programistów, którzy są błędnie przekonani o własnych umiejętnościach. Każdy programista – nawet naiwny inżynier tuż po studiach, bez głębszej wiedzy o projektowaniu programów i z niewielką wiedzą na temat stopniowej poprawy oprogramowania – mógłby podpisać się pod tym manifestem w błędnym przekonaniu, że robi to, co on głosi. Dyskusyjne jest stwierdzenie zawarte w Manifestie: „Innymi słowy, dążąc do osiągnięcia celów wymienionych po lewej stronie, odkryliśmy, że cele po prawej są niezbędne”. Gdyby to sformułowanie brzmiało, jak zaznaczają krytycy manifestu: „Innymi słowy, cele po lewej stronie to dobry początek i chcielibyśmy pomóc ci również w dążeniu do celów po prawej stronie”, to przesłanie Manifestu byłoby mniej elitarne. Dla tych programistów, którzy chcieliby być lepsi i którzy są świadomi, że się uczą, manifest stawia ich w roli „innych” (*helping others learn the craft*), którym Rzemieślnicy Oprogramowania „pomagają nauczyć się” rzemiosła. Wielu programistów znalazłoby się tym samym po stronie tych innych, którzy osiągają skutki wymienione po prawej stronie i z pewnością nie traktują siebie jak rzemieślników, którzy sformułowali manifest w jego obecnej formie.

(g) Autorka zaznacza, że działania Alexandry Elbakyan (s. 181–183), a w szczególności utworzenie i prowadzenie biblioteki cyfrowej Sci-Hub, są złożone pod względem etycznym i zawierają silne argumenty przemawiające zarówno za pozytywną, jak i negatywną oceną moralną jej działań. Choć wielu chwali ją jako orędowniczkę otwartego dostępu do wiedzy, zapewniającą bezpłatny dostęp do artykułów naukowych, to jej działania nie respektują praw autorskich, co jest naruszeniem prawa. Należałoby dodać, że ostatecznie „słuszność” jej działań zależy od perspektywy i priorytetów wartości: przykładowo od dostępu do wiedzy kontra poszanowanie praw własności intelektualnej.

(h) Autorka często przywołuje istotne dla współczesnego społeczeństwa zwroty „etyka komputerowa” (por. 4.2), „etyka szczegółowa”, skupiając się zasadniczo na „działaniach etycznych programistów”. Wskazuje przy tym, w Taylorowskim stylu, na rozważania etyczne Maxa Schelera, Józefa Tischnera, Isaiaha Berlina, Władysława Stróżewskiego, Paula Ricoeura, Andrzeja Kocikowskiego, Jamesa Moora, Michaela Feathersa, Romana Ingardena, Stanisława Judyckiego i innych. Często niezbyt wiadomo, dlaczego te różniące się między sobą

² Por. James D. Herbsleb, *Global Software Engineering: The Future of Socio-technical Coordination*, [w:] FOSE '07: 2007 Future of Software Engineering, Washington: IEEE Computer Society, s. 188–198, <https://doi.org/10.1109/FOSE.2007.11>.

³ DeepSeek Coder: Let the Code Write Itself, <https://deepseekcoder.github.io/>.

⁴ Manifesto for Software Craftsmanship. Raising the bar, <https://manifesto.softwarecraftsmanship.org/>.

ujęcia etyki są właśnie zestawiane. Ponadto należy pamiętać, że etyka komputerowa wyłoniła się w latach 80. XX wieku jako dziedzina zajmująca się analizą kwestii prywatności i bezpieczeństwa w technologii, odpowiadającą za wyważania obowiązków moralnych nie tylko programistów czy użytkowników komputerów i specjalistów, ale zwłaszcza za te dotyczące sfery społecznej i politycznej. Szybki rozwój internetu i technologii komputerowych prowadzi do wzrostu przestępczości internetowej, która może dotknąć nie tylko osoby prywatne, ale również każdy region świata lub w skrajnym przypadku cały kraj. Konieczne jest zatem podnoszenie świadomości i wprowadzanie odpowiednich instytucjonalnych, ponadnarodowych przepisów, aby zapobiegać przestępstwom związanym z używaniem komputerów i internetu. Wymagane jest wprowadzanie przepisów określających zbiór zasad, które zapewnią sprawną i niezawodną komunikację oraz transmisję informacji w tak rozumianym zakresie, co trudno znaleźć u wymienionych powyżej autorów. Przepisy te powinny również dyscyplinować osoby i organizacje dopuszczające się nadużyć oraz podejmować niezbędne działania w celu zapobiegania zdarzeniom mogącym wyrządzić krzywdę innym podmiotom. Brak rygorystycznych i kompleksowych przepisów dotyczących cyberprzestrzeni utrudnia przestrzeganie wymogów moralnych programistom, jak i użytkownikom prywatnym czy całym firmom. Rosnąca liczba cyberprzestępstw prowadzi do ogromnych strat społecznych i materialnych. Wraz z rozwojem technologii tak rozumiana etyka komputerowa ma kluczowe znaczenie w opracowywaniu i wdrażaniu różnorodnych strategii etycznych. Oprócz poprawy bezpieczeństwa ludzi jej przestrzeganie przyczyniłoby się do znaczącego postępu zarówno w zakresie tworzenia i implementacji prawa, jak i stosowanych technologii. Odpowiednie instancje muszą wprowadzać surowe zasady i regulacje, aby zapewnić, że wszyscy indywidualni i instytucjonalni użytkownicy komputerów ponoszą odpowiedzialność za swoje działania. Słusznie w tym kontekście mgr Bajor wskazuje na szerokie rozumienie terminu hakerzy, mówiąc o ich trzech grupach.

(i) Pojawienie się internetu wiązano z powstaniem utopijnej przestrzeni, w której użytkownicy będą mogli się komunikować i bez żadnych ograniczeń przekazywać sobie informacje. Tak rozumiana wolność w internecie jest od dawna zagrożona wraz z rosnącą liczbą państw budujących zapory sieciowe w celu cenzurowania treści w internecie i zakazujących usług internetowych niezgodnych z regulacjami rządowymi. To, co kiedyś było wolną przestrzenią, zostało teraz okiełzane, ograniczone, a nawet skrępowane. Autorka rozprawy nie wspomina w tym kontekście cenzury i ponownie, nie wiadomo, jak patrzą na nią filozofowie-etycy wymieniani w rozprawie. Sprawa jest skomplikowana, gdyż cenzura internetu nie zawsze jest etycznie zła. Istnieje wiele okoliczności, w których – w zależności od stanowiska – cenzura może być uznana za uzasadnioną. Wiele osób zgadza się z tym, ale jednocześnie nie zgadzają się one co do tego, kto decyduje, które treści są cenzurowane, a które nie. Decyzje przedstawicieli władz centralnych podejmowane w imieniu obywateli są często nie do przyjęcia dla większości użytkowników internetu. Wracając do treści omawianej rozprawy, cenzura nie „tkwi” w kodzie źródłowym, lecz w samym modelu sztucznej inteligencji. Został on wytrenowany, aby zajmować określone stanowiska polityczne i unikać określonych tematów. Co nie jest niczym niezwykłym, bowiem każda sztuczna inteligencja przeznaczona do użytku publicznego jest taka sama, to tylko kwestia tego, czyje poglądy polityczne i ideologiczne wyznaje. Model cenzurujący jest często nieprzejrzysty i nie da się „zajrzeć” do jego wnętrza i „zobaczyć” cenzury, podobnie jak nie da się zajrzeć do ludzkiego mózgu, aby zobaczyć, jakie są jego poglądy polityczne. Mgr Bajor wspomina jedynie o GitHub (s. 204, 223), opartej na chmurze platformie, na której programiści mogą przechowywać, udostępniać i współpracować z innymi nad pisanie kodu. Przechowywanie kodu w „repozytorium” w GitHubie umożliwia prezentowanie lub udostępnianie swojej pracy oraz śledzenie i zarządzanie zmianami w kodzie w czasie. Ponad 60 milionów programistów tworzy swoje projekty na platformie GitHub, dzięki czemu firma ma niemal monopol na rynku. W roku 2022 tysiące chińskich programistów

„obudziło się” i odkryło, że ich kod *open source*, hostowany w Gitee, wspieranym przez państwo chińskie i konkurującym z międzynarodową platformą repozytoriów kodu GitHub, został zablokowany i ukryty przed publicznym dostępem. Następnie, tego samego dnia, Gitee opublikowało oświadczenie, w którym wyjaśniło, że zablokowany kod jest poddawany ręcznej weryfikacji, tak jak każdy kod *open source*, który od tego momentu będzie musiał zostać poddany weryfikacji przed publikacją. Firma „nie miała wyboru” – napisano. Gitee nie odpowiedziało na pytanie „MIT Technology Review” o przyczynę tej zmiany, ale powszechnie uważa się, że chiński rząd wprowadził kolejną surową cenzurę⁵. Ostatecznie, jak obawiają się owi programiści, może to zniechęcić ludzi do wkładu w projekty *open source*, a chiński przemysł oprogramowania ucierpi z powodu braku współpracy. Z etycznego punktu widzenia cenzura internetu, podobnie jak działalność hakerów, to złożona kwestia, z silnymi argumentami zarówno za, jak i przeciw. Choć można ją postrzegać jako sposób na ochronę jednostek i społeczeństwa przed krzywdą, jak np. mowa nienawiści czy dezinformacja, budzi ona również obawy dotyczące wolności słowa i potencjalnych nadużyć władzy. Z jednej strony można postawić pytanie, na ile cenzura wpływa na tworzenie języka kodu źródłowego programów komputerowych, a z drugiej strony – na ile dałoby się ją przedstawić (opisać) w ujęciu hermeneutyki dyskursu Paula Ricoeura i filozofii form symbolicznych Ernsta Cassirera.

(j) Na s. 77 Doktorantka porównuje typy rozbudowania wypowiedzi przykładowego fragmentu kodu napisanego w *Ruby* (j. naturalny) i odpowiadającego mu znaczeniowo kodu w języku *lambda calculus* (s. formalny). Należałoby w kontekście *Ruby* wspomnieć, że języku *lambda calculus*, prawda jest definiowana jako funkcja typu Boolean, która przyjmuje dwa argumenty i zwraca pierwszy z nich, jeśli jest to „prawda”, lub drugi, jeśli jest to „fałsz”.

3. Uwagi formalne

Rozprawa została napisana jasnym językiem, z niewielką liczbą drobnych niedociągnięć (literówek, np. s. 4, 14, 51, 69, 71, 88, 200).

W spisie literatury brak pozycji: A. Pawelec, *Kontynentalna filozofia języka: wybrane stanowiska*, [w:] *Współczesna filozofia języka. Inspiracje i kierunki rozwoju*, red. P. Stalmaszczyk, Primum Verbum, Łódź 2013.

Konkluzja

Mgr Anna Bajer podjęła się przedstawienia złożonego - z perspektywy metodologicznej i treściowej - tematu: język kodu źródłowego programów komputerowych w ujęciu hermeneutyki dyskursu Paula Ricoeura i filozofii form symbolicznych Ernsta Cassirera. Opracowała go na sposób interdyscyplinarny, co często jest lub może być narażone na krytykę z dwóch stron: (a) powiązania oraz doboru treści empiryczno-naukowych i filozoficzno-psychologicznych, (b) tworzenia i operowania złożoną metodologią. Mimo wspomnianych powyżej niedociągnięć (z których część jest rodzajem dyskusji akademickiej), omawianą rozprawę doktorską mgr Anny Bajer oceniam pozytywnie. Prezentuje ona (zgodnie z art. 187 ustawy Prawo o szkolnictwie wyższym i nauce, Dz.U. z 2021 r., poz. 478) ogólną wiedzę teoretyczną w dyscyplinie filozofia oraz świadczy o umiejętności Doktorantki do samodzielnego prowadzenia pracy naukowej. Wnioskuje o dopuszczenie mgr Anny Bajer do dalszych etapów przewodu doktorskiego.



Prof. dr hab. Józef Bremer SJ

⁵ Zeyi Yang, *How censoring China's open-source coders might backfire*, „MIT Technology Review” 2022, May 30, <https://www.technologyreview.com/2022/05/30/1052879/censoring-china-open-source-backfire/>.