

Prof. dr hab. Eugeniusz Wilk

Kraków, 10.02.2015 r.

Uniwersytet Jagielloński

Katedra Mediów Audiowizualnych

OCENA

rozprawy doktorskiej Pana mgra Jana K. Argasińskiego nt. *Software studies. Kultura, sztuka i oprogramowanie* napisanej pod opieką naukową profesora Andrzeja Pitrusa

Stuart S. Shapiro w dość znanym tekście zatytułowanym *Informatyka: badanie procedur* pisał: „Podobnie, jak nauki przyrodnicze, informatyka zawiera zarówno nauki podstawowe, jak i stosowane”. To zdanie z pewnością mogłoby zostać potraktowane przez Autora rozprawy doktorskiej jako motto wyznaczające główne kierunki jego rozumowania: z jednej strony bowiem mgr Argasiński z dużą skrupulatnością wprowadza czytelnika swojej dysertacji w obszar podstawowych pojęć i definicji wyznaczających ramy działania programów komputerowych, zwracając uwagę przede wszystkim na takie pojęcia, jak maszyna Turinga, dane i informacja, algorytm oraz komputacja i procesy z nią związane. Z drugiej strony główne zdanie Autora polegało przecież nie na prezentacji i przeglądzie ważniejszych wątków i pojęć informatycznych, lecz na podjęciu kluczowej kwestii opisu i interpretacji kulturowych konsekwencji działania w przestrzeni ludzkich aktywności nowych „technologii definiujących” (termin Boltera), jakimi dla Autora są technologie informatyczne, a w szczególności oprogramowanie. Jeśli poważnie potraktujemy słynne stwierdzenie Alexandra R. Galloway’a – czołowego przedstawiciela *software studies* – mówiące o tym, że „komputer jest podmiotowością”, to musimy przyjąć potrzebę radykalnych przemyśleń dotyczących kulturowych (i epistemologicznych) konsekwencji działania tych technologii. Pytanie zatem zasadnicze, jakie kieruje czytelnik w stronę Autora tej dysertacji naukowej brzmi: na ile udało się tutaj odkryć ujawnić, odszyfrować owe procesy zmian w sferze kultury oraz z jakimi

mechanizmami mamy tutaj do czynienia? Innymi słowy - na ile tytuł rozprawy jest adekwatny do jej zawartości? I tutaj muszę się przyznać, że Autor wystawia czytelnika swojej pracy na pewną próbę cierpliwości i pokory, która wynika głównie z przyjętej przez Doktoranta strategii konstruowania wywodu i eksplikacji głównych tez. Z pewnością ważnym obszarem przedstawianym w pracy jest – jak stwierdza sam Autor – działanie oprogramowania „które w swej istocie jest przecież wykonywanym przez komputery zbiorem poleceń, konstrukcją językową” (s. 42). Aby wyeksplikować główne tezy dotyczące tak sformułowanej problematyki Jan K. Argasiński przyjmuje historyczny punkt widzenia (na szczęście nie idzie tropem Umberta Eco, który narrację historyczną tego wątku rozpoczyna od *Księgi Rodzaju*) i wskazuje na koncepcję Johna Wilkinsa, biskupa Chester (1616 – 1672) przedstawioną w jego pracy *An Essay towards a Real Character and a Philosophical Language*. To w tym dziele Wilkins przedstawił ideę uniwersalnego, sztucznego języka, który – jak twierdzi Jan K. Argasiński – można potraktować jako pewien „przeźroczysty interfejs między światem umysłu a uniwersum empirycznych obiektów”. Kolejnym autorem omawianym przez Autora dysertacji jest Gottfried Wilhelm Leibniz i jego znana idea *języka ogólnego* oraz koncepcja *characteristica universalis*. Te historyczne wątki zostały poprzedzone w rozprawie omówieniem przełomu, jaki dokonał się w matematyce i logice pod koniec XIX wieku i na początku XX wieku za sprawą takich uczonych, jak: Bertrand Russell, Alfred North Whitehead, Kurt Godel i Gottlob Frege. Ten kierunek prezentacji prowadzi Jana K. Argasińskiego nieuchronnie w stronę koncepcji Alana Turinga, wyrażonej w jego przełomowej pracy *On Computable Numbers (O liczbach policzalnych)*, którą J. David Bolter nazywa „manifestem nowego elektronicznego ładu” (*Człowiek Turinga*, s. 41). Nim został wykonany pierwszy komputer – twierdzi Bolter – Turing zdefiniował działanie maszyn logicznych. Autor ocenianej dysertacji doktorskiej opisuje dość szczegółowo i przekonująco te kwestie starając się, w kolejnych partiach rozprawy, odpowiedzieć na pytania: czym jest teoria informacji?, jak rozumieć elementarne terminy: informacja, entropia, redundancja, sygnał etc. Te fragmenty tekstu, rozpatrywane łącznie z sygnalizowanymi wcześniej rozważaniami na temat języka, tworzą – jak ujmuje to sam Autor rozprawy „paradygmat teoretyczno-historyczny” (raczej część teoretyczno-

historyczną) i otwiera zestaw problemów związanych z „rzeczywistym oprogramowaniem działającym i aktywnym w świecie” (s. 59). W tych fragmentach tekstu pracy czytelnik znajdzie więc odpowiedź na pytanie: czym jest procesor, jakie są zasady jego działania, jaka jest architektura jednego z pierwszych komputerów (Johna von Neumanna, Johna Mauchly’ego i Johna P. Eckerta) i wreszcie jak rozumieć pojęcie: struktury danych. Te definicje dają Autorowi podstawy, by opisywać istotę i zastosowania języków programowania (dobrym przykładem jest tutaj wspomniany przez Doktoranta projekt *Processing*). To niewątpliwie wartościowa strona rozważań Autora: prezentacja i opis wybranych przykładów programów artystycznych oraz narzędzi humanistyki cyfrowej ujawniają istotne możliwości zastosowania specyficznych programów komputerowych w obszarach aktywności artystycznej i badań naukowych z dziedziny szeroko pojmowanej humanistyki. Przekonującą ilustracją takiego podejścia są opisywane przez Autora przypadki: *Gry o życie* oraz modelowania dynamicznych struktur narracyjnych za pomocą grafów. Z pewnością te fragmenty rozprawy, które poświęcone zostały prezentacji oraz interpretacji wybranych przykładów z obszaru *software art* należy uznać za interesujące i wartościowe. Przykłady te bowiem ujawniają zupełnie nowe możliwości obecne potencjalnie w całej sferze praktyk programowania. Opisana przez Jana K. Argasińskiego twórczość algorystów stanowi również dobrą ilustrację połączenia działalności stricte programistycznej z działaniami, które mają wyraźnie artystyczne cele. Dla czytelnika rozprawy interesujące są zapewne kolejne omawiane przez Autora zjawiska, takie, jak na przykład demoscena – ciekawa subkultura zrzeszająca użytkowników komputerów, którzy mają ambicje tworzenia tzw. dem (programów będących często rezultatem współpracy programistów, grafików, muzyków) uznawanych często za przykłady sztuki komputerowej. Aktywiści demosceny, jak słusznie zauważa Doktorant, to pasjonaci nowych technologii związani przeważnie z ruchami anarchizującymi i wnoszący nowe, wartościowe akcenty do świata technologii i kultury.

W swej dysertacji doktorskiej mgr Jan Argasiński stawia cały szereg kluczowych, istotnych kwestii z punktu widzenia medioznawstwa kulturoznawczego. I tak w części drugiej pracy znajduje się ustęp zatytułowany *Maszyny semiotyczne*, w

którym Autor twierdzi, że „oprogramowanie możemy lokować w uniwersum semiotycznym” (s. 51). Należy jedynie żałować, że Jan Argasiński nie pogłębił tego przecież tak istotnego, z punktu widzenia refleksji nad kulturą, wątku. Po pierwsze zatem w rozumowaniu zaprezentowanym w rozprawie wyraźnie ujawnia się redukcjonistyczne stanowisko polegające na utożsamieniu kwestii semiotycznych z zagadnieniami syntaktycznymi. Należy jednak zwrócić uwagę, że przywoływany przez Autora w tej części pracy Noam Chomsky istotnie w swoich *Syntactic Structures* koncentrował się przede wszystkim na kwestiach syntaktycznych, jednak w kolejnych publikacjach zwracał się już w stronę problematyki semantycznej proponując istotne rozwiązania z zakresu semantyki generatywnej. Jednym słowem Jan Argasiński winien czuć się chyba bardziej zobligowany tytułem swojej dysertacji i pokusić się o teoretyczne rozwinięcie tematów leżących w obszarach semantyki właśnie oraz pragmatyki – tym bardziej, że te zagadnienia ściśle łączą się naturalnie z kwestiami filozoficznymi, które pojawiają się w kontekście pytań o oprogramowanie. Warto w tym miejscu przywołać opinię Jamesa H. Fetzera, który jest przekonany o konieczności formułowania pytań natury filozoficznej (ściślej pytań ontologicznych i epistemologicznych). Jakimi obiektami są w gruncie rzeczy komputery i programy? – pyta wspomniany autor i odpowiadając podkreśla, że programy mają wprawdzie przede wszystkim charakter syntaktyczny, „to posiadają one pewne znaczenia semantyczne, którego nie mają dowody matematyczne”. To istotne spostrzeżenie, pozwalające odróżnić istotę programu komputerowego od działania stricte matematycznego, to może być zarazem punkt oparcia dla rozważań na temat pragmatycznych (kulturowych) odniesień sfery oprogramowania, rozważań, które Autor ocenianej rozprawy w pogłębiony sposób mógłby w przyszłości przeprowadzić. Gorąco Go do tego namawiam! O tym, że sprawa jest ważna i aktualna świadczy chociażby przypomniane przez Alexandra R. Galloway’a – niekwestionowanego przecież autorytetu w obrębie *software studies* (w jego książce *The Interface Effect*) – wystąpienie Warrena Sacka na Uniwersytecie w Amsterdamie, w którym wspomniany badacz zaproponował formalną listę cech pozwalającą odpowiedzieć na pytanie, czym w istocie rzeczy jest program komputerowy? Nie wchodząc tutaj (ze zrozumiałych względów) w szczegóły, można powiedzieć, że zarówno dla Sacka, jak i dla

Galloway'a, konteksty i odniesienia semantyczne są istotne przy charakterystyce oprogramowania. W tej samej książce Galloway poświęca jeden, obszerny rozdział relacjom między ideologią i oprogramowaniem (tekst ten był przez autora opublikowany wcześniej i został przetłumaczony i zamieszczony na łamach „Kultury Popularnej”). Formułowana w tym tekście teza mówiąca o istnieniu swoistej przyległości, podobieństwa między strukturą ideologii i strukturą oprogramowania jest punktem wyjścia przedstawianego rozumowania. Pojęcia rozważane przez Galloway'a tworzą frapujący i intrygujący trop myślowy - wszak rozważania tego autora, dotyczące istoty oprogramowania, krążą wokół takich pojęć, jak język, performatywność, narracyjność (w ujęciu Aarsetha), wizualność. Mapa analizowanych kwestii jest, jak widzimy, niesłychanie atrakcyjna dla kulturoznawcy. Galloway na przykład nawiązuje do książki W.J.T. Mitchella *Czego chcą obrazy?* ponieważ kontekst wizualności traktuje jako niezbywalną część rozważań na temat istoty oprogramowania. Ów element widzialny – opisywany przez Galloway'a – rozpada się niejako na dwie sfery: widzialnego postrzegania oraz poznawczego procesu konceptualizacji i rozumienia, które zyskują charakter wizualny właśnie. W tym właśnie punkcie Galloway dostrzega atrakcyjność oraz innowacyjność myśli Mitchella i podkreśla ważność jego stwierdzenia „że podstawową, genetyczną formacją kultury nie jest tekst, czy nawet idea, lecz obraz”. Oczywiście można pójść dalej tą drogą i przypomnieć, co – w ujęciu Mitchella – jest traktowane jako istota tzw. zwrotu obrazowego: przypomnijmy więc, że autor *Zwrotu piktorialnego* postuluje nowy rodzaj podejścia do sfery obrazu, które jest postlingwistyczne i postsemiotyczne oraz traktowane jest jako rodzaj „złożonej gry pomiędzy wizualnością, zmysłami, instytucjami, dyskursem, ciałem i figuratywnością”. Dlaczego przypominam te dobrze znane tropy myślowe? Otóż dlatego, że wydają się one istotne wtedy, gdy pytamy o kulturoznawcze aspekty *software studies*. Tym samym wymienione wyżej pojęcia mogłyby się stać dla Jana Argasińskiego istotnymi etapami rozumowania, dzięki którym możliwe byłoby bardziej wyraziste wyartykułowanie problemów zaznaczonych w podtytule dysertacji. Doktorant przyjął w swej pracy odmienną strategię postępowania, którą można nazwać historyczno-prezentacyjną. Tą drogą Autor artykułuje najważniejsze kwestie badawcze, narzucone przez temat rozprawy,

ale czyni to niejako „okrężnie”, poświęcając dużo miejsca na przykład zagadnieniom historycznym. Bardziej efektywnym sposobem byłoby postawienie określonych problemów badawczych, a następnie próba ich interpretacji i rozwiązania (przy uwzględnieniu naturalnie również perspektywy historycznej). Narracja w rozprawie z pewnością zyskałaby wtedy na klarowności i prawdopodobnie Autor uniknąłby wrażenia powierzchowności i arbitralności doboru kwestii merytorycznych. To wrażenie jest dodatkowo wzmacniane przez stylistykę pracy, sugerującą, że niekiedy główną troską piszącego jest wyjaśnienie czytelnikowi podstawowych pojęć z zakresu informatyki (odwoływanie się do leksykonów, podręczników oraz *Wikipedii* potwierdza niejako tę intencję). Jednym słowem główną troską Autora winny być w przyszłości starania o wypracowanie przejrzystego i koherentnego metajęzyka opisu. Przykładem pewnego niedookreślenia i niejasności jest na przykład miejsce koncepcji ANT w rozumowaniu Autora dysertacji. „Elementy teorii aktora-sieci pozwalają – pisze Jan Argasiński na s. 201 – na ujęcie dynamik relacji między aktorami wchodzącymi w relacje o charakterze zakotwiczonym w relacjach społecznych”. To z pewnością mógłby być bardzo inspirujący trop, gdyby Autor nie poprzestał na tym stwierdzeniu i próbował jednak nieco szerzej odnieść się do jednej z najbardziej „wpływowych” przecież koncepcji również na gruncie kulturoznawstwa! A zatem jasna i przekonująca deklaracja będąca podstawą do interpretacji ANT w duchu na przykład antyesencjalizmu i relacjonizmu, dałaby solidne podstawy do poszukiwania dalszych konsekwencji kulturowych funkcjonowania oprogramowania. W perspektywie ANT Latoura świat jawi się przecież jako niezdeterminowany, otwarty, heterogeniczny i hybrydalny – jednym słowem jest to świat, w którym „aktorzy” nie posiadają wcześniej zdefiniowanej tożsamości i w którym dualizm podmiotów i przedmiotów zostaje zanegowany. Uwaga tym bardziej na miejscu, że Jan Argasiński kilka stron dalej podejmuje „kwestie związane z elementami metafizyki Sieci” (s. 207) i pyta wprost „jaki status mają [te] obiekty?” Odpowiedź na to pytanie nie jest do końca satysfakcjonująca. Po pierwsza Autor pisze, że „filozoficzna spekulacja zbliża nas do internalizacji zjawisk”, a po drugie „kod i oprogramowanie [...] są nośnikami najprawdziwszej rzeczywistości”. Stwierdzenia o takiej wadze winny być jednak nieco szerzej rozwinięte i uzasadnione. Krótkie odwołanie się do koncepcji *realis*

Michała Ostrowickiego nie rozstrzyga sprawy. Z obowiązku recenzenta pragnę jeszcze zwrócić uwagę na niedostatki redakcyjne widoczne w rozprawie: zdarzają się niestety błędy językowe (niepoprawna forma „niemniej jednak” pojawia się konsekwentnie w całym tekście kilkanaście razy), błędna pisownia nazwisk (Shakespeare – s. 152), błędne używanie dużych liter („II wojna światowa” – s.60). Autor w przyszłości winien wykazać się jednak większą starannością przy ostatecznej edycji tekstu.

Wspomniane wyżej zastrzeżenia, mające niekiedy charakter polemiki naukowej, nie podważają naturalnie pozytywnej opinii o przedłożonej do oceny rozprawie Jana K. Argasińskiego. Dysertacja ta pozostaje, obok prac kulturoznawców lubelskich (przede wszystkim Piotra Celińskiego) jedną z nielicznych prac polskich autorów poświęconych kulturowym aspektom oprogramowania. Mapa problemów i analizowanych przykładów, które proponuje tutaj Autor, daje czytelnikom odpowiedź, jakie w gruncie rzeczy zadania stoją przed *software studies*.

Stwierdzam zatem, że rozprawa Jana K. Argasińskiego spełnia warunki stawiane dysertacjom doktorskim i wnoszę o dopuszczenie Doktoranta do dalszych etapów przewodu doktorskiego.

