

Rafał Ilnicki

## Użytkownik modelowy – podmiotowość implikowana przez oprogramowanie<sup>1</sup>

Współcześnie podmiotowość użytkowników oprogramowania, ale także użytkowników kultury<sup>2</sup>, zostaje radykalnie przekształcona, w wyniku czego sama podporządkowuje się pewnym programom. Jednak nie oznacza to redukcji człowieka do narzędzia czy reżimu użytkowania tego narzędzia, lecz powoduje wytworzenie się nowych form podmiotowości, które zyskują odniesienie w formach – użytkownikach modelowych (kształtują oni podmiotowość poprzez wyznaczenie jej podstawowych właściwości). Należy zwrócić uwagę na fakt, że patrząc na użytkownika siedzącego przed komputerem, widzimy jego nieobecne bycie (odłączenie użytkownika od otoczenia) oraz przepływ informacji pomiędzy jego układem poznawczym a interfejsem (zakładany proces podczas obserwacji układu człowiek-komputer). Zachodzi tutaj jeszcze jeden proces – transfer

| 59

---

<sup>1</sup> Rafał Ilnicki jest stypendystą Fundacji im. Adama Mickiewicza na rok 2012.

<sup>2</sup> Użytkownik kultury zastępuje tutaj uczestnika. Dzięki temu możliwe staje się pomyślenie podstawowych relacji w kulturze z perspektywy uogólnionej narzędziowości użytkownika zasobów kultury. Użytkownik kultury tworzy własną narzędziowość oraz korzysta z tej wyprodukowanej przez innych użytkowników oraz przemysł programów, czyli przez przemysł wytwarzający programy ludzkiego doświadczenia. Człowiek jest sprzętem, na którym uruchamiany jest program doświadczenia. Równie uprawnione wydaje się także dopowiedzenie: człowiek jest oprogramowaniem, do którego dostawia się sprzęt, dzięki któremu może on funkcjonować. Są to dwie perspektywy odnoszące się do opisu tego samego procesu.

subiektywności użytkownika do teleobecnych terytoriów programowania własnego doświadczenia. Subiektywność pojmowana jest tutaj jako własność użytkownika, która nie jest całkowicie przynależna jego wnętrzu, będąc także właściwa otoczeniu (w tym przypadku terytoriom dostępnym jedynie zdalnie). Określając subiektywność w sposób negatywny, dochodzimy do zrozumienia mediów technicznych, a konkretnie do sytuacji, w której użytkownik przed monitorem produkuje subiektywność, ale wciąż jednak okreśmy go w perspektywie pewnego braku – nieobecności, braku kontaktu z otoczeniem, monadycznego zamknięcia w interfejsie. Perspektywa aktywnej produkcji subiektywności w zdalnym zapośredniczeniu wprowadza możliwość ujęcia działań użytkownika w całej rozległości jego akcji, to znaczy przyjrzenia się egzystencjalnym terytoriom<sup>3</sup>, które stanowią uogólnienie odniesienia wszelkiej możliwej aktywności podmiotowej.

Egzystencjalne terytoria nie stanowią przestrzeni dostępu, lecz są efektem ukierunkowania podmiotowości użytkowników poprzez figurę użytkownika modelowego. Istnienie samego terytorium w środowisku elektronicznym to za mało, by wyprodukować dostateczną ilość subiektywności zdolną odłożyć się w podmiotowości. Stąd też tak częste współcześnie funkcjonowanie użytkowników jako *pustych użytkowników modelowych* – korzystają oni z oprogramowania w sposób niepełny, to znaczy niepełny do tego, by ich subiektywność ustaliła się w podmiotowości, ta zaś została ukierunkowana przez jej model, który pełni rolę sterownika determinującego to ukierunkowanie. Korzystają oni z mediów, lecz nie są w stanie zaktualizować w nich egzystencjalnych terytoriów, aktywując spontanicznie płwy subiektywności, które nie zyskują ustalenia w podmiotowości. Niemożliwość zarządzania własną subiektywnością powoduje, że „nie-ludzka przed-osobowa część subiektywności”<sup>4</sup> jest tworzona przez techniczne interfejsy, które nie tyle pośredniczą w produkcji subiektywności, co produkują ją w programie antyprodukcji, to znaczy utrudniającym podmiotowe skupienie. Produkcja subiektywności zostaje wstrzymana poza oprogramowaniem, dlatego też jednostka,

60|

<sup>3</sup> Jest to termin Felixa Guattariego, który rozumiem jako przestrzeń instalacji wyprodukowanej subiektywności użytkownika. Tutaj: zdalnie dostępne terytorium dla aktywności egzystencjalnych podmiotu odnoszące się do produkcji subiektywności. Egzystencjalne terytoria mają własne oprogramowanie – wystarczy wprowadzić do nich określoną subiektywność, tak aby została ona poddana działaniu przez zainstalowane w otoczeniu oprogramowanie.

<sup>4</sup> F. Guattari, *Chaosmosis: an Ethico-Aesthetic Paradigm*, transl. P. Bains, J. Pefanis, Bloomington-Indianapolis 1995, s. 9.

by móc identyfikować własną podmiotowość, ponawia próby połączenia z interfejsami, które obsługuje – dopiero ten dostęp gwarantuje jej poczucie jednostkowości.

Gdy jesteśmy odłączeni od oprogramowania, to także myślimy oprogramowaniem: jak dane zdjęcie wyglądałoby na Facebooku, jak można by z tej sytuacji wykonać zdjęcie na demotywatory, jaki film można by ściągnąć, jakie oprogramowanie diagnostyczne pomoże nam w identyfikacji choroby. Seria takich zdarzeń uzmysławia, że oprogramowanie ze sfery technologicznej przechodzi do aktywnego formowania i formatowania subiektywności, co jest trwałym efektem. Pozostajemy zatem użytkownikiem modelowym nawet po wyłączeniu wszelkich urządzeń służących do zdalnej produkcji subiektywności. Mówiąc „jesteśmy użytkownikami modelowymi”, mamy na myśli „nasza podmiotowość jest nośnikiem dla modelu użytkownika oprogramowania”, czyli zachowuje się ona zgodnie z tym oprogramowaniem.

Techniczna produkcja subiektywności i jej odkładanie się w podmiotowości zgodnie z regułami użytkownika modelowego, który modeluje wyprodukowaną subiektywność, zapewniając rodzaj tymczasowej i sieciowej konsystencji podmiotowości, wyznacza sytuację wszelkich zasobów subiektywności i ich dalszych transformacji. „Technologiczne maszyny informacyjne i komunikacyjne operują w sercu ludzkiej subiektywności, nie tylko wewnątrz jego pamięci i inteligencji, ale także w jego wrażliwości”<sup>5</sup>. Oprogramowanie działa w sposób pozajęzykowy, to znaczy wykorzystuje języki programowania jako oparcie własnego działania, jednak są one nie-ludzkie, to znaczy ich geneza istotowo nie odpowiada człowiekowi.

## Subiektywność – podmiotowość – użytkownik modelowy

W zarysowanej sytuacji należy zatem podać wstępne rozumienie podstawowych terminów.

**Subiektywność** – oznacza najbardziej wewnętrzne zasoby użytkownika, to znaczy jego emocjonalne, afektywne, intelektualne podstawowe wyposażenie (najbardziej pierwotna sfera doznaniowości). Użytkownik ma częściowy dostęp do subiektywności, ponieważ myślenie refleksyjne ustępuje programowaniu własnych stanów mentalnych poprzez przyjmowanie gotowej świadomości z interfejsów oprogramowania. Oprogramowanie uzyskuje dostęp i wyłączność na pośredniczenie w dostępie podmiotu do własnej

---

<sup>5</sup> Ibidem, s. 4.

subiektywności. Subiektywność za Felixem Guattarim zostaje określona w następujący sposób: „Zespół warunków, które umożliwiają emergencję indywidualnych i/lub kolektywnych instancji jako odwołujące się do siebie egzystencjalne Terytoria, sąsiadujące lub pozostające w relacji ograniczenia, do inności, która jest sama subiektywna”<sup>6</sup>. Tworzy ona Kolektywne Wyposażenie Nieświadomości podzielonej na warstwy<sup>7</sup> zgodnie z określonym obowiązkiwaniem działania poszczególnych maszyn produkujących subiektywność, to znaczy określonego oprogramowania subiektywności człowieka, wprowadzając „gotowe wymiary subiektywności”<sup>8</sup> instalowane bezpośrednio w interfejsach maszyn technologicznych (transfer subiektywności zachodzi od człowieka i do człowieka). Są one przygotowywane na poziomie podmiotowym i poziomie modelowania. Technicyzacja subiektywności oznacza, że jest ona *ready-made*, poprzedza podmiot – tworzy nieświadomość jego egzystencjalnych terytoriów.

Taka subiektywizacja jest częściowa<sup>9</sup>, ponieważ nie istnieje oprogramowanie zdolne do zarządzania całą subiektywnością, dlatego też rozumienie oprogramowania jako „urządzenia subiektywizacji”<sup>10</sup> dotyczy częściowych zmian w subiektywności.

**Podmiotowość** – przejawia się pomiędzy subiektywnością a poziomem użytkownika modelowego. Użytkownik ma pełniejszy dostęp do własnej podmiotowości, która jest podmiotowością w wersji technicznej, to znaczy daną w zapośredniczeniu przez określone programy. Oznacza ona tymczasowe ustalenie subiektywności. Tu zachodzi relatywne spowolnienie subiektywności – odkładanie się jej w efektach pamięci, przeżyciach, doświadczeniach. Tak rozumiana podmiotowość to *mindware* (oprogramowanie umysłu). „Tradycyjnie podmiot był pojmowany jako ostateczna esencja indywiduacji, jak czyste, puste, przedrefleksyjne ujmowanie świata, jądra wrażliwości, ekspresywności – ujednolicony ze stanów świadomości”<sup>11</sup>. Podmiotowość jest systemem operacyjnym subiektywności. Podmiot ma dostęp sam do siebie – zyskuje samoświadomość własnego programowania. Człowiek komunikuje się ze swoją subiektywnością za pomocą oprogramowania – tylko w ten sposób zyskuje do niej dostęp. Język programowania subiektywności jest językiem używania własnej subiektywności.

62 |

---

<sup>6</sup> Ibidem, s. 9.

<sup>7</sup> Zob. ibidem, s. 12.

<sup>8</sup> Ibidem, s. 7.

<sup>9</sup> Zob. ibidem, s. 15.

<sup>10</sup> Ibidem, s. 20.

<sup>11</sup> Ibidem, s. 22.

Podmiotowość rozciąga się pomiędzy *mindware* a *wideware* (technologie poznawcze)<sup>12</sup>. Nie zyskuje ona skupienia w jednej określonej postaci, lecz umożliwia częściowe zarządzanie zasobami subiektywności, przez co zachodzi „relatywnie progresywna mutacja subiektywności”<sup>13</sup>; to znaczy, że przez jej oprogramowywanie (instalację działających programów) dochodzi do tego, iż podmiot rozwija zdalnie, a więc w zdalnym zapośredniczeniu, własną subiektywność.

**Model (użytkownik modelowy)** – abstrakcyjny wzór wytworzony w kulturze ukierunkowujący wielość stających się subiektywności poprzez wpływanie na działanie podmiotu (podmiot podłącza się do określonego użytkownika modelowego, tak aby mógł się z nim utożsamić w sposobie zarządzania własną subiektywnością). Subiektywności nie można w pełni kontrolować, dlatego musi ona zostać spowolniona i skupiona w podmiotowości. Model to abstrakcyjny wzór ukierunkowujący konkretną podmiotowość (skupienie subiektywności). Użytkownik modelowy oznacza już abstrakcyjną funkcję będącą operatorem kulturowych zachowań – koordynuje on zachowania podmiotu w taki sposób, żeby odpowiednio programował on własną subiektywność. Strukturyzuje mikrozachowanie (kliknięcia, uruchomienia programów, czynności włączania i wyłączania sprzętu) zgodnie z określonym profilem zachowania wchodzącym w skład określonej formy użytkownika modelowego. Użytkownik modelowy oznacza model programowania naszej podmiotowości. Kultura jest tutaj sprzętem (*hardware*), zaś podmiotowość oprogramowaniem (*mindware/wideware*). Użytkownik modelowy jest zatem abstrakcyjną maszyną służącą do koordynacji podmiotowości z subiektywnością w określonym, kulturowo wyprodukowanym wzorze eksploatacji oprogramowania.

Produkcja subiektywności (heterogeniczna) składa się na podmiotowość (względnie homogeniczna), która odpowiada obrazowi użytkownika modelowego (konstruowany kulturowo typ idealny z wielu programów, które wykorzystuje). Inżynierowie oprogramowania, informatycy oraz pozostałe podmioty biorące udział w kreowaniu kulturowego obrazu programów są „wynalazcami nowych sposobów bycia”<sup>14</sup>, czyli nowych wersji ludzkiej podmiotowości.

<sup>12</sup> Zob. A. Clark, *Mindware. An Introduction to the Philosophy of cognitive Science*, New York-Oxford, s. 147.

<sup>13</sup> F. Guattari, *Chaosmosis...*, s. 21.

<sup>14</sup> J.D.G. Bacca, *Science, Technology, History, and Philosophy*, w: C. Mitcham (ed.), *Philosophy of Technology in Spanish Speaking Countries*, Dordrecht-Boston-London 1993, s. 237.

Modele są modelami podmiotowego bycia<sup>15</sup>, czyli żywej subiektywności biotechnopolitycznie administrowanej przez *mindware*<sup>16</sup>. Użytkownicy modelowi tworzą także własne modele ukierunkowania podmiotowości i zarządzania subiektywnością. Pełna transcendencja modelu poza interfejs nie jest jednak możliwa. Częściowa transcendencja konstytuuje moment przełączenia, który wynika z wyczerpania aktualnego modelu. Użytkownik modelowy zmienia się w innego użytkownika modelowego, gdy zasoby jego subiektywności ulegną dezaktualizacji. Model podmiotowości jest zaprogramowany. Został zaprojektowany przez kogoś, dany przez kogoś, używany przez kogoś. Te elementy zapośredniczenia tworzą podmiotowość, gdzie człowiek zachowuje względną autonomię, to znaczy autonomię w zapośredniczeniu. Można bowiem wyróżnić interfejsy, na które ma on wpływ, oraz takie, które pozostają poza jego podmiotową kontrolą. Dlatego też sytuacje użytkownika modelowego są definiowane poprzez wydarzenia, którymi można zarządzać z perspektywy podmiotowej, bądź też zarządzają one subiektywnością użytkownika z poziomu użytkownika modelowego (jest on wtedy zarządzany).

Oprogramowanie działa poprzez ograniczenie subiektywności (program ma wyraźne granice wyznaczone liniami jego kodu<sup>17</sup>), co doprowadza do tego, że podmiotowość zostaje skupiona w określonych programach doświadczenia

<sup>15</sup> Por. M. Heidegger, *Ku rzeczy myślenia*, tłum. J. Mizera, Warszawa 1999, s. 70. Heidegger mówi tutaj o modelach ontycznych, co w sensie ogólnym odpowiada modelom podmiotowości. Istotny jest fakt, że korzystając z określonego oprogramowania, użytkownik zapośrednicza w nim swoje bycie, czyli w wypadku powtarzalności danej czynności – wzoru obsługi oprogramowania, doprowadza do krystalizacji modelu ontycznego. Sens użytkowania jest zatem projektowany – stąd tak istotna rola oprogramowania jako tego, co wyprzedza użytkowanie. Podmiotowość implikowana przez oprogramowanie jest autonomiczną częścią podmiotowości *per se*, nie będąc jakimś jej wyimkiem, czy też symulacyjnym fragmentem. Kwestię tę rozwijam w pojęciu cyberzystencji jako pometafizycznym sposobie wydarzenia się bycia w erze sieciowych mediów elektronicznych.

<sup>16</sup> Na *mindware* składa się oprogramowanie umysłu. Operuje ono pomiędzy subiektywnością a modelem użytkowania, który przyciąga podmiotowość. Oprogramowanie umysłu (*mindware*) ma częściowy dostęp do subiektywności, większy do podmiotowości (mając do niej dostęp, wnioskuje o własnej subiektywności). Podmiotowość to spowolniona technicznie subiektywność, zaś model użytkowania (wielu użytkowników modelowych) skutkujący określonym zachowaniem jest ponownym przyśpieszeniem.

<sup>17</sup> Także kodu dystrybucji i teleobecnego skupiania ludzkiej subiektywności w nie-ludzkim egzystencjalnym terytorium (np. jako zasobu afektywnego i zasobu uwagi do późniejszego wykorzystania), przez co zachodzi utożsamienie pracy urządzeń technicznych i człowieka na poziomie użytkownika modelowego (dzieje się to poza paradygmatem symulacji).

pozwalających na kontrolę dystrybucji subiektywności, ale już w reżimie operacyjnym tych programów. Zyskuje ono swoje przedłużeniu w modelu – działa on zarówno na poziomie subiektywności, jak i podmiotowości.

Wreszcie, podmiotowe modele użytkowników stabilizują nietrwałą tożsamość wirtualną, pozwalając na osiągnięcie względnej stałości i ciągłości przejścia pomiędzy jednym użytkownikiem modelowym a drugim i kolejnymi przełączeniami użytkowników modelowych. Modele użytkownika wyróżniamy zatem, odwołując się zarówno do zgeneralizowanych form programowania, jak i do oprogramowania konkretnego (kulturowego, a także w funkcji programów komputerowych).

Należy zatem „oderwać” użytkownika od narzędzi, za pomocą których utrzymuje on własną subiektywność w spektrum panowania danego oprogramowania definiującego jego podmiotowość, tak by sam użytkownik stał się dla techniki czystym wydarzeniem. Serii interakcji użytkownika odpowiada seria działań maszyny – są to heterogeniczne serie<sup>18</sup>. Naszą podmiotowość tworzą serie interakcji z oprogramowaniem, które komunikuje nam oraz przez nas serie odpowiedzi i próśb, wpływając na ukierunkowanie naszej subiektywności. To zarządzanie jaźnią na dwóch poziomach dokonywane przez oprogramowanie zawsze skutkuje powstaniem modelu użytkownika jako pewnego zestandaryzowanego systemu interakcji. Człowiek-Windows, człowiek-gra komputerowa, człowiek-edytor tekstu stanowią takie modele, bowiem ich cała podmiotowość, myślenie do pewnego stopnia wpisuje się w strukturę oraz logikę używanego przez nich oprogramowania.

## Oprogramowanie w technicznej współczesności

Oprogramowanie pełni współcześnie egzystencjalnie doniosłą rolę jako podstawowe medium ludzkiego doświadczenia. Komputery, telefony, wszelkie urządzenia elektroniczne, roboty, sprzęt przemysłowy – wszystkie wymagają istnienia odpowiedniego oprogramowania (*software*), które pozwala działać sprzętowi (*hardware*). Ta opozycja stanowi najczęściej umowne przeciwstawienie tego, co stanowi materialną podstawę istnienia sprzętu elektronicznego, temu, co pozostaje jedynie „miękką” nakładką. Dziś jednak oprogramowanie i sprzęt funkcjonują jako *everyware*<sup>19</sup>. Pojęcia, takie

|65

<sup>18</sup> G. Deleuze, *Logika sensu*, tłum. G. Wilczyński, Warszawa 2011, s. 63. Są one zatem niepasujące do siebie, mimo ciągłych prób komunikowania się.

<sup>19</sup> Termin Adama Greenfielda na oznaczenie wszechdostępności technologii informacyjnych zarówno w aspekcie sprzętowym, jak i tym dotyczącym oprogramowania.



jak *wetware*, *affective computing*, *cloud computing*, prowadzą z konieczności do przewyżnienia tej dychotomii i określenia wzajemnej relacji tych dwóch składowych jako wzajemnie się konstytuujących o niskim poziomie odrębności. Tylko dzięki połączeniu komponentów *soft* i *hard* możliwa jest realizacja ich funkcji w określonych programach. W tym sensie oprogramowanie stanowi podstawowe medium teleologii, wyznacza bowiem to, w jaki sposób, pod jakimi warunkami zrealizowanie celu jest możliwe. Ludzkie intencje i działania są ujmowane sekwencyjnie. Mówiąc o nim, nie można pomijać faktu, że stanowi ono medium codziennego doświadczenia, będąc zarówno pojemną metaforą wyrażającą stosunek człowieka do stechnicyzowanego świata, jak i przenośnią wskazującą na sposób funkcjonowania mózgu oraz umysłu<sup>20</sup>, a także konkretyzacją kultury, która staje się swoistym oprogramowaniem dla umysłu (*mindware*). Nie sposób pominąć także roli, jaką oprogramowanie odgrywa w kulturze, pełniąc funkcję narzędzia umożliwiającego podstawowy dostęp do świata dla ogromnej liczby użytkowników technologii informacyjnych. Ta wieloaspektowość i wielowymiarowość sprawia, że oprogramowanie stanowi kulturową technologię podstawową i jako takie wywiera modelujący wpływ na użytkowników, wytwarzając modele podmiotowości.

## Kultura – zbiór zgeneralizowanych języków programowania

Oprogramowanie zarządza podmiotowością; wpisuje się ono w struktury dostarczane przez konkretne rozwiązania instrukcji składających się na konkretne programy. Oprogramowanie jest rozumiane jako autonomiczny zestaw programów (rodzaj wirtualnego dyspozytywu<sup>21</sup> mocy) służący do tworzenia podstawowych relacji w świecie (bycie-w-świecie jest zastąpione przez bycie-w-programie lub bycie-w-programach<sup>22</sup>). W tym sensie możemy tutaj mówić o różnych formach zarządzania i urządzania ludzkiego doświadczenia przez oprogramowanie, które dokonuje się na różnych poziomach intensywności. Oprogramowanie stanowi dyspozytyw mocy, wprowadza relację władza-program, to znaczy program połączeń subiektywności do określonych

66|

<sup>20</sup> Mam na myśli obliczeniowe teorie umysłu. Paradygmat komputacyjny wymaga istnienia pewnych programów, które nadawałyby kierunek obliczeniom.

<sup>21</sup> Urządzenia w sensie teorii dyspozytywu formułowanej m.in. przez G. Deleuze'a, J.P. Baudry'ego oraz M. Foucaulta.

<sup>22</sup> Bycie-w-programie jest podporządkowane określonemu użytkownikowi modelowemu.



modeli. Sformułowanie tezy o modelującym wpływie oprogramowania na podmiotowość sprawia, że musimy brać pod uwagę perspektywę wspólnej indywiduacji człowieka z oprogramowaniem, zamiast rozpatrywać te relacje w separacji, podkreślając oraz porównując ich zakres wzajemnego wpływu. Dlatego nie można rozpatrywać oprogramowania w separacji od jego roli w modelowaniu ludzkiego doświadczenia, które prowadzi do wykształcenia się wzorów podmiotowości<sup>23</sup>. Pytanie zatem nie dotyczy samego obiektu technicznego (oprogramowania) w relacji do człowieka, tylko jego powstawania w perspektywie jego wykorzystania. Takie połączenie określane jest przez Bernarda Stieglera mianem transindywiduacji<sup>24</sup> (czyli połączenia indywiduacji w pewną strukturę oraz wzór w kulturze), która odnosi się do wspólnej podstawy – ustalonych zasobów w pamięci jako podstawy indywiduacji kolektywnej w kulturze. Człowiek indywiduuje się poprzez interakcje z sieciami relacji, które znajdują swoje zapośredniczenie w oprogramowaniu: komunikacyjnym, użytkowym, operacyjnym, rozrywkowym<sup>25</sup>. Zatem staje się sobą dzięki indywiduacji w programach (indywiduacja rozpoczyna się równocześnie w subiektywności i w użytkowniku modelowym – ta podwójna indywiduacja jest niezależna od podmiotu).

Oprogramowanie to głównie język (jako że jest ono wyrażone w określonym języku programowania), jednak nie ogranicza się on do kodów składających się na wewnętrzną strukturę programu wyrażoną w konkretnym języku programowania. Uwzględniając aspekt programów znajdujący swoje określenie w architekturze programów elektronicznych (funkcja abstrakcyjna<sup>26</sup>), należy odnieść go do efektów oprogramowania w kulturze i filozofii (funkcja konkretna). Przeporządkowanie tych aspektów jest zdecydowanie kontrintuicyjne, jako że oprogramowanie odczytujemy jako coś konkretnego, podczas gdy regulacje kultury oraz wzajemne stosunki składowych systemów filozoficznych odczytujemy jako pewne abstrakcje, jednak takie spojrzenie nie pozwala dostrzec dynamiki przemian między tymi dwiema płaszczyznami. Człowiek działa według zgeneralizowanych kulturowych języków programowania – jego podmiotowość wpisana jest w złożoną sieć instrukcji tworzących techniczne (a w nich symboliczne) ramy jego funkcjonowania. W tym aspekcie oprogramowanie jest konkretne, bowiem

<sup>23</sup> Jako „zastygłej” subiektywności.

<sup>24</sup> Jest to termin wykorzystywany przez Bernarda Stieglera.

<sup>25</sup> Typologia ta nie odnosi się do dziedzin przyjętych w klasyfikacji oprogramowania, lecz ma zwrócić uwagę na skalę zmian i ingerencji oprogramowania w ludzką podmiotowość.

<sup>26</sup> Poziom ten jest abstrakcyjny i konkretyzowany dopiero poprzez kulturową hermeneutykę cyfrowej technicyzacji.

związane z podjęciem działania, nawet jeśli dotyczy tylko obsługi zaprogramowanych urządzeń lub relacji. Oprogramowanie tworzy dyspozytyw mocy w charakterze skupienia oraz ukierunkowania zawartej w nim wirtualności (w leibnizjańskim sensie aktywnej siły sprawczej, wpływając na zachowanie użytkowników). Podmiotowość jest wyposażona w programy interakcji z dyspozytywami kultury (zbiorami programów doświadczenia). Performatywny (sprawczy) wymiar tej podmiotowości odnosi się do wykorzystania wirtualności programowania, czyli do określenia, co człowiek może zaprogramować, a czego nie<sup>27</sup>. W ten sposób myślenie o człowieku przebiega w ramach antropologii oprogramowania. Człowiek jednocześnie programuje i jest programowany – tak wyrażona formuła stawania się podmiotowości, czyli kulturowego programowania podmiotowości, która polega nie tylko na sterowaniu nią, lecz także na wyposażaniu do programowania kultury, dotyczy technologicznie gwarantowanej możliwości. Podstawową aktywnością człowieka staje się programowanie. Przenosi on bowiem schematy nabyte podczas użytkowania kulturowego oprogramowania<sup>28</sup> do własnej podstawowej aktywności, internalizując zasadę sekwencyjnego myślenia i działania właściwego programom.

Kultura zatem to złożony system języków programowania subiektywności poprzez wprowadzanie oprogramowania podmiotowości. Użytkownik modelowy jest powszechnym językiem programowania ludzkiej subiektywności przez podmiotowość.

## Podmiot i oprogramowanie

„Podmiot jest skupieniem wiedzy, tak samo jak pasmo górskie jest skupieniem gór. Subiektywność podmiotu jest jako taka skupieniem”<sup>29</sup>. Podmiot jest czymś już ustalonym, podczas gdy subiektywność jest wciąż indywidualująca się. Podmiot też się indywidualuje (nabywając autonomiczność i jednostkowość), jednak skupia on dotychczasową indywidualację z różnych

68|

<sup>27</sup> Programowanie człowieka skutkujące wytworzeniem się określonego modelu podmiotowości odnosi się zatem do stoickiego programowania, gdzie podstawowy podział – pierwotna różnica tego podejścia – dotyczy dokonania selekcji tych rzeczy, które można zmienić (programować), oraz tych, których nie można zmienić (są nieprogramowalne). Postacią pojęciową jest tutaj Epiktet jako wzór programisty, który potrafi rozróżniać kulturowe kody i zawarte w nich zgeneralizowane języki programowania.

<sup>28</sup> Termin, który stosuje Lev Manovich.

<sup>29</sup> M. Heidegger, *Powiedzenie Nietzschego „Bóg umarł”*, tłum. J. Gierasimiuk, w: idem, *Drogi lasu*, Aletheia, Warszawa 1997, s. 198.

programów doświadczenia rozumianą jako cybernetyczna organizacja w modelu, podczas gdy subiektywność wciąż rozpoczyna serię acentrycznych indywiduacji<sup>30</sup>, które swoim skupieniem ogarniają model, zwrotnie go programując. Subiektywność jest projektująco-sięgająca, podczas gdy podmiotowość oznacza cofanie się do wcześniejszych ustaleń (ustalenia w użytkowniku modelowym<sup>31</sup>). Odpowiada to relacji przeszłości jako ustaleniu w podmiotowym istnieniu do przyszłości pojmowanej jako chaotyzujący ruch subiektywności. Zatem, podążając za wskazaniem Heideggera, należy podkreślić, że z perspektywy systemowej podmiot jest systemem, zaś subiektywność stanowi czynnik różnicujący tego systemu (sterownik różnicowania). „Podmiot zawsze, o ile jest podmiotem, upewnia się o swoim zabezpieczeniu”<sup>32</sup>. Podmiot działa w ramach określonego modelu własnej podmiotowości, w tym wypadku implikowanej przez oprogramowanie komputerowe – zabezpieczenie jest realizowane przez ciągłe aktualizowanie programu potwierdzającego podstawowe instrukcje pozwalające na zachowanie autoregulacji w granicach podmiotowości. Mechanizmem tego zabezpieczania jest powtórzenie, które odpowiada procesowi powracania do tej samej aplikacji, gdzie następuje różnicowanie własnego zachowania w obrębie danego modelu poprzez produkcję subiektywności. Różnicowanie i chaotyzowanie oraz towarzyszące mu powtarzanie to ontologiczny program autoimmunizacji zabezpieczający skupioną subiektywność w modelu podmiotowości.

Użytkownik modelowy, to znaczy taki, który jest wbudowany w program własnego funkcjonowania, jest modelowany i modeluje siebie. Użytkownik modelowy (człowiek stający się określonym programem) jest niezależny od konkretnej subiektywności<sup>33</sup>. Poruszając tę kwestię, za konieczne uważam zaznaczenie, że omawiany jest tutaj pewien aspekt podmiotowego funkcjonowania, ponieważ zdecydowanie nie ogranicza się ona do danego modelu-typu użytkownika. Współcześnie odpowiednikiem dyspozytywu – jego

<sup>30</sup> Acentryczna indywiduacja polega na jednoczesnym indywiduowaniu się podmiotów (ale i wszelkich zasobów) z różnych źródeł. Dla indywiduacji zapośredniczonej w mediach technicznych przyjmuję pojęcie teleindywiduacji, czyli zdalnej indywiduacji z wielu centrów indywiduacji, co wyraża tę acentryczność.

<sup>31</sup> W przypadku oprogramowania możemy mówić, że to skupienie dokonuje się jako efekt programowania, to znaczy sekwencyjnego ukierunkowania subiektywności, która może chaotyzować się jedynie według zestandaryzowanych w programie instrukcji. W ten sposób chaotyczność jest ograniczona przez program ukierunkowania.

<sup>32</sup> Słowa Martina Heideggera.

<sup>33</sup> Użytkownicy internetu dzielą ze sobą nie tylko warstwę informacyjną, ale także subiektywność.

technologiczną aktualizacją – jest oprogramowanie. Możliwe jest mówienie o pewnym programie doświadczenia przez ten dyspozytyw wprowadzanego (jest on konkretyzacją modelu podmiotowości kształtowanej przez oprogramowanie). Oprócz konsekwencji natury praktycznej, użytkowej, ta sytuacja prowokuje pytania natury filozoficznej<sup>34</sup> – na ile ktoś jest zdolny przekroczyć schemat funkcjonowania założony przez kulturowych programistów<sup>35</sup> oraz wszystkie komercyjne stopnie internalizacji programu w subiektywności użytkowników<sup>36</sup>. Wyłaniają się zatem dwie podstawowe płaszczyzny: pierwsza dotyczy performatywności kodu, jego zdolności do podporządkowywania sobie użytkownika (jego formowania i programowania), druga – społecznych modeli funkcjonowania użytkownika, gdzie nie możemy mówić o programowaniu na poziomie informatycznym, tylko społecznym<sup>37</sup>. Skoro: „Komputery działają pod kontrolą programów, a funkcja programowania jest ulokowana na zewnątrz technologii w ludzkich agentach<sup>38</sup>, które zarządzają nią z góry”<sup>39</sup>, to podmiotowość modelowana przez oprogramowanie przyjmuje różne pozycje mocy w dyspozytywie mocy jako możliwej sile wywieranej na nie przez oprogramowanie.

## Performatywność kodu

W przypadku modelującej podmiotowość funkcji oprogramowania rozważany jest tutaj kod na co najmniej dwóch podstawowych płaszczyznach – kodu maszynowego oraz sposobu jego prezentacji na ekranie monitora. Oczywiście oba wzajemnie się warunkują, przy czym ten pierwszy jest nadrzędny w stosunku do drugiego. Przedmiotem zainteresowania jest tutaj pytanie o dystrybucję czasu – różne programy, w tym systemy operacyjne, mają różną architekturę, zatem w różnym czasie, w różnej kolejności wykonują różne sekwencje. Użytkownicy w inny sposób (na różnych poziomach kompetencji) oczekują wykonania tych sekwencji, często przyjmując je za oczywistość. Sama architektura komputera także wpływa na podmiotowość

70|

<sup>34</sup> Technologia stanowi źródło filozoficznej problematyki.

<sup>35</sup> Na obecnym poziomie rozwoju kultury oprogramowanie jest sprzężone w jej mechanizmy autoregulacji. Jednak aspekt podmiotowy wciąż jest tutaj aktualny.

<sup>36</sup> To także pytanie o transcendencję w obrębie danego programu – jak można przekroczyć za pomocą programu komputerowego własną podmiotowość.

<sup>37</sup> W ramach inżynierii społecznej.

<sup>38</sup> Przyjmuję rozumienie agenta jako podmiotu działania zarówno ludzkiego, jak i nieludzkiego (np. pod postacią bytów informatycznych).

<sup>39</sup> A. Feenberg, *Critical Theory of Technology*, New York 1991, s. 91.

użytkowników, bowiem przyzwyczajają do określonej reaktywności, określonego korzystania, specyficznego sposobu wykonywania komend internalizowanych bezpośrednio do subiektywności. Użytkownik modelowy nie musi mieć wglądu w kod, nie musi znać języków programowania (by programować<sup>40</sup>), choć gdyby miał, to jego samoświadomość zwiększyłaby się. Taki jego obraz postuluje Friedrich Kittler, który mówi, że aby sprawnie funkcjonować w kulturze medialnej i nie być cyfrowym analfabetą, należy znać co najmniej jeden język sztuczny. Taki stan rzeczy na pewno spowoduje większe uwrażliwienie na sam kod. Jego nieznanostwo niejednokrotnie uniemożliwia mu skuteczne poradzenie sobie z problemami (podkreślę, że komputery są wyjątkowo zawodne, a kody programistyczne niedoskonałe i pełne usterek<sup>41</sup>), które nie wymuszają na nim programistycznej interwencji, lecz poprzez zrozumienie sposobu, w jaki program działa, zastosowanie określonej procedury naprawiającej na poziomie programów, czego przykładem może być chociażby to, że dwa programy nie są ze sobą kompatybilne, tj. ich wzajemna obecność prowadzi do dysfunkcji. Istotną kwestią jest tutaj także błąd. Błędy istniejące na poziomie maszynowym skutecznie zmieniają sposób funkcjonowania programu. Są „oswajane” na poziomie kulturowym – np. „terror 404”<sup>42</sup>, sławny blue screen, wreszcie komunikaty o błędzie funkcjonują jako ikoniczne przedstawienie uniwersalnego błędu „łagodzony” poprzez komunikaty o możliwości przywrócenia poprzedniego stanu lub „miękkiego lądowania” w uruchomionej innej aplikacji.

Jednak użytkownik często spotyka się jedynie z efektem działania tego kodu. Prześledźmy to na przykładzie komunikatorów komputerowych oraz usług społecznościowych. Jeden program, jedna platforma programowa ma spełniać wszelkie funkcje dostępne w innych programach, tak aby nie doprowadzić użytkownika do znużenia. Zatem także posunięcia, takie jak wprowadzanie zmian potencjalnie na gorsze, mają tę przewagę, że aktywizują ludzi, którzy wystąpią jako nowa klasa wirtualna dzielona podług użytkowników modelowych, których przyjęcie umożliwia programowanie subiektywności, którzy zwracają się przeciw niekorzystnym dla nich zmianom. Takie posunięcia gwarantują minimalny koszt w postaci utraconego odsetka użytkowników przy jednoczesnym indukowaniu większego przywiązania do programu (obiektu). Program komputerowy wymusza komunikację nie mniej niż dzwoniący telefon, z tym że tutaj dzwonięcie najczęściej zastępowane jest

<sup>40</sup> W sensie, w jakim mówię o ogólnej teorii programowania.

<sup>41</sup> Podobnie jest też z językami programowania ludzkiego doświadczenia.

<sup>42</sup> Jest to metaforyczne określenie błędu 404, który sygnalizuje, że wskazana w przeglądarce internetowej strona nie może zostać odnaleziona.

przez dźwięk dostępności użytkownika (który można wyłączyć). Tutaj sama obecność, często jak w życiu społecznym, stanowi zaproszenie do rozmowy. W tym przypadku stosowanie znaczników „jestem zajęty” i „zaraz wracam” nie uważam za obowiązujące, bowiem są często ignorowane, a użytkownicy i tak zarzucani są komunikatami; zresztą stosowanie ich można uznać za komunikacyjną kokieterię, próbę zyskania atrakcyjności przez manifestowanie swojej niedostępności. Reguły programu komunikacji stanowią komunikacyjne programowanie użytkowników (ich komunikacyjnych wejść i wyjść), co stanowi *mindware* – ustalenie podmiotowości w określonym oprogramowaniu podmiotowości, czyli zbiorze programów doświadczenia regulowanych przez formę użytkownika modelowego.

Strategie użytkowników oprogramowania są zdecydowanie spluralizowane, lecz można wyróżnić pewne ich wzory. Zaznaczam, że ta klasyfikacja prezentuje pewne stereotypowe wyobrażenia użytkowników, zaś podążająca za nią strategia skupiająca się na sposobach użycia oraz bycia użytym przez program zostaje przeniesiona już na poziom programowanego w ten sposób doświadczenia. Profesjonaliści perfekcyjnie obsługują dany program oraz są na najwyższym poziomie kompetencji.

Na poziomie społecznym badamy to, co merkantylistycznie zostało skodyfikowane i zainstalowane w kulturze (rynek jest systemem operacyjnym kultury). Mam tu na myśli głównie fora i Web 2.0. Tutaj badamy społeczne i kulturowe życie programu. Dlaczego patrzymy w dane okienko, a nie w inne? To nie tylko pytanie z zakresu kognitywistyki oraz psychologii poznawczej, lecz także kulturowych ram widzenia skupionych w maszynach widzenia, czyli maszynach do programowania okulacentrycznych zasobów subiektywności. Przecież wszystko, co uruchamiamy na komputerze, jest programem, przeglądarka WWW także. Ona również została napisana w określonym języku, tak samo jak strony, nawet jeśli były utworzone za pomocą generatora.

72|

Podmiotowość w programie zмага się z podmiotowością użytkownika. To przestrzeń zarówno zawłaszczania, jak i negocjacji. Jest możliwe, że użytkownik modelowy działa jak wirus, infekując podmiotowość użytkownika i w końcu ją przejmując, tj. narzucając ją. Ten proces możemy sobie wyobrazić na przykładzie korzystania z bardzo małej rozpiętości mediów, ograniczających się do reaktualizowania już odczytanych danych. W ten sposób program komputerowy staje się programem-nawykiem ludzkiego doświadczenia.

Każdy program jest pod kogoś zaprogramowany, lecz nie oznacza to, że nie można go przeprogramować. Skuteczność takiego zabiegu określają warunki technologiczne oraz kulturowe, które pozostają ze sobą w stanie silnej

współzależności. Możemy wyróżnić różne wzorce podmiotowości, które wykraczają poza rozróżnienie na „kto” i „co”. Odpowiada to procesowi technicyzacji subiektywności skupionej w modelu podmiotowym konkretnego zestawu programów umysłu i doświadczenia (*mindware*).

Do programów zostają przeniesione funkcje umysłu – w ten sposób różne subiektywności tworzą pseudoumysł wynikły z ich połączenia. Tak powstały *mindware* odnosi się do działania podmiotowości, jednak z tego powodu, że nie może zapanować nad własnymi zasobami, dlatego też zleca zarządzanie nimi konkretnym programom, które zwalniają je z troski o siebie, wykonując techniki utrzymania siebie w ciągłej gotowości do funkcjonowania.

Nie istnieją programy komputerowe, które całkowicie modelowałyby zachowanie użytkownika. Należy wyróżnić kilka podstawowych sfer wpływu, którym odpowiadają stopnie intensywności oddziaływania programu komputerowego na podmiotowość. System operacyjny będzie oddziaływał w sposób zarówno bezpośredni, jako metaprogram oraz środowisko dla innych programów, w którym są one uruchamiane, a zatem pozostają w funkcjonalnej zależności od niego, jak i niebezpośredni, ponieważ obecne w nim aplikacje z konieczności będą musiały odwoływać się do rozwiązań w nim zastosowanych, aby móc działać. Analiza programów komputerowych w tej perspektywie musi być zatem wysokokontekstowa. Na potrzeby analiz dokonam szczególnych uproszczeń, by ukazać, w jaki sposób za pomocą oprogramowania dochodzi do produkowania (programowania) określonej subiektywności.

## Podmiotowość jako program działania i selekcji decyzji

Andrew Feenberg twierdzi, że „Skończony podmiot jest stworzony z jego działań podejmowanych w świecie”<sup>43</sup>, więc należy uznać, że stanowi on opozycję wobec nieskończoności spontanicznej dystrybucji subiektywności. Użytkownik modelowy jest wprowadzaniem linii podziału podług których subiektywność użytkownika zostaje dystrybuowana i eksploatowana przez określone oprogramowanie. Konieczne jest także rozpatrzenie tego, w jaki sposób oprogramowanie związane jest z działaniem. Może ono zarówno je wymuszać, faworyzować, jak i ograniczać i wstrzymywać. To właśnie oprogramowanie wyznacza daną racjonalność działania<sup>44</sup>, „ponieważ

| 73

<sup>43</sup> A. Feenberg, *Critical...*, s. 112.

<sup>44</sup> Andrew Feenberg widzi w komputeryzacji możliwość i ograniczenie pracy ludzkiej w zależności od konkretnej technologii i racjonalności, jaką ucieleśnia. Por. *ibidem*, s. 63.



technologia nie jest neutralna, lecz fundamentalnie stronnicza w kierunku poszczególnej hegemonii, wszelkie działania podjęte w jej ramach przyczyniają się do reprodukcji tej hegemonii<sup>45</sup>.

Programy komputerowe ucieleśniają pewne mikrotechniki władzy i są regulowane zgodnie z regułami operacyjnej autonomii<sup>46</sup>. Hegemonia programów wyznacza ramy funkcjonowania podmiotów, które są ograniczone przez ich działanie, poddane działaniu przemocy programów (zdolności programów do kontroli subiektywności, np. część subiektywności zostaje usunięta/wyłączona wraz z usunięciem danego programu doświadczenia). Mikrotechniki kontroli uwzględniają rozmieszczenie ikon, zasady używania interfejsu, możliwość dostępnych zmian poprzez konfigurację, animacje będące symboliczną reprezentacją procesów i działań wykonywanych przez komputer, odpowiednie usytuowanie ciała przed monitorem, ułożenie dłoni, szybkość reakcji użytkownika. Wszystko to sprawia, że podmiotowość jest warunkowana przez oprogramowanie – oczywiście biorąc pod uwagę skalę tego warunkowania, należałoby odnieść ją do badań kognitywnych i socjologicznych, które w sposób empiryczny (mierzalny) mogłyby określić ilość czasu spędzanego przed danym informatycznym interfejsem oraz jego wpływ na zachowanie. Uogólniając, możemy stwierdzić, że podmiotowość implikowana przez oprogramowanie jest formowana na zasadzie implementowania użytkownikom określonych zestawów nawyków obsługi urządzeń technicznych. W ten sposób skonstruowana zostaje podmiotowość podatna na podobne mikrotechniki kontroli w innych sferach – politycznej, społecznej, naukowej, światopoglądowej, rynkowej. Podmiotowość jest tam ukształtowana w taki sposób, że łatwiej reaguje z odpowiednio zaprogramowanymi interfejsami niż z innymi. Użytkownicy mogą jednak sami modelować swoją podmiotowość – tworząc lub w sposób alternatywny wykorzystując i modyfikując zastane oprogramowanie. W ten sposób wyłaniają się nowe formy podmiotowości: hakerska, prosumencka, programisty.

<sup>45</sup> Ibidem, s. 65.

<sup>46</sup> Operacyjna autonomia to pojęcie Andrew Feenberga, które wskazuje na bardzo istotny fakt obowiązywania mikrotechnik kontroli składających się na daną instytucję. Zawarte mikrotechniki reprodukują porządek władzy w niej zawarty, obejmując wszystkie podmiotowości. W ten sposób możemy odczytać oprogramowanie jako pewną formę instytucji – nasza percepcja, nasze zachowanie, sposób oraz warunki podejmowania działania są określone przez program. Nie determinują one jednak w sposób absolutny zachowania użytkownika, jednak sugerują, podpowiadają zgodnie z logiką funkcjonowania systemów eksperckich, jakie zadanie można czy też należy podjąć. Podsumowując, mikrotechniki kontroli zawarte w oprogramowaniu operacjonalizują i optymalizują kognitywne funkcje oraz zachowanie użytkownika. Por. A. Feenberg, *Critical...*, s. 79.

Wymienione typy są przykładami użytkowników modelowych ery informacyjnej, podobnie jak podmiotowość implikowana przez oprogramowanie wytwarza określone typy użytkowników modelowych. Technomerkantylizm<sup>47</sup> tworzy modele użytkowników, które stają się częścią projektowanego i programowanego oprogramowania. Wielość technicznych kodów<sup>48</sup> odpowiada wirtualnym podmiotowościom, które mogą zostać ucieleśnione w systemie technicznym, z którego przenoszą się na użytkownika, tworząc *mindware*.

## Pozorna neutralność modelu

Użytkownik modelowy oznacza pewną programowaną i programowaną podmiotowość. Model – wzór podmiotowości zawarty w technicznych kodach kultury nie jest tylko pewną formą, do której dopasowuje się użytkownik, lecz stanowi aktywnie działający program formatowania i kształtowania nawyków użytkownika podług określonego wzorca. Bez względu na rodzaj oprogramowania użytkownik pozostaje zależny od pewnego kodu obecnego w programie i w ten sposób ucieleśnia pewne właściwe mu zachowania, lub też aktywnie się im przeciwstawia poprzez przeprogramowywanie, hacking, instalowanie rozszerzeń, poprawek, „łatek” (*patch*), które pozwalają na warunkowanie odmiennych modeli podmiotowości zawartych w wykorzystywanym oprogramowaniu.

## Użytkownik modelowy i idealizm programistyczny

Model, będąc pewnym typem idealnym, który nigdy nie może zostać w pełni zaktualizowany w konkretnym użytkowniku, jest wpisany także jako pewien obraz myśli, element społecznego imaginarium, w kulturę. Funkcjonujący w przestrzeni dyskursywnej, audiowizualnej i audiowirtualnej obraz kodu technicznego wyposażony jest w mikrotechniki technomerkantylnej kontroli, które mają na celu przyciągnięcie użytkownika. Jednak użytkownik modelowy wciąż jest zakładany, wciąż istnieje w teleobecnych terytoriach egzystencjalnych. Jest to zatem idealizm programistyczny, w którym użytkownik zyskuje swoje potwierdzenie poprzez

| 75

---

<sup>47</sup> Pojęcie używane przez Wojciecha Chyłę na określenie form biotechnosystemowej ekonomii.

<sup>48</sup> Techniczny kod „jest zasadą, pod którą są dokonane techniczne wybory w perspektywie zachowania operacyjnej autonomii (np. wolności dokonania podobnych wyborów w przyszłości)”, A. Feenberg, *Critical...*, s. 80.

tożsamość własnej formy użytkownika modelowego wraz z zarządzaną podług niej subiektywnością.

Użytkownicy modelowi to użytkownicy zarówno modelowani abstrakcyjnie, jak i znajdujący swoje konkretyzacje w określonych typach i formach podmiotowości kształtowanej za pomocą mikrotechnik kontroli kodów technicznych. Wpisują się one w szerszy horyzont przemian człowieka spowodowany rozwojem technologii sieci i programów. Człowiek projektowany i programowany wyznacza przyszłość, jeszcze nadal – człowieka. Obraz programu wpływa na decyzje użytkowników, te natomiast do pewnego stopnia determinują złożoną sieć wzajemnych zależności pomiędzy kulturą, ekonomią, administracją i polityką. Ludzka podmiotowość znajduje się w centrum tych sporów – jest ona ich celem oraz narzędziem zmiany.

## Facebook i Google+ jako odmienne przykłady *mindware'u*

Zarówno Facebook, jak i Google+ wystarczą, by określić je mianem *mindware*, ponieważ odpowiadają podmiotowości użytkowników. Użytkownik modelowy portalu Facebook zarządza relacjami w sposób mający uzyskać największe skupienie, by następnie się rozproszyć – mechanizm skupienia-rozproszenia jest dla niego podstawowym sposobem myślenia i odczuwania. Użytkownik modelowy serwisu Google+ dystrybuje własną subiektywność w selekcji obszarów dystrybucji (kręgi).

Różnica pomiędzy tymi dwoma programami społecznościowymi polega na oferowaniu odmiennego modelu podmiotowości. Technologie obu aplikacji w sferze wizualnej i funkcjonalnej są do siebie zbliżone, różniąc się jedynie pewnymi szczególnymi rozwiązaniami. Należy zwrócić uwagę na to, że korzystanie z oprogramowania dotyczy zarówno sfery racjonalnej<sup>49</sup>, jak i sfery emocji. Podmiotowość na Facebooku z konieczności musi zarządzać swoimi emocjami, bowiem wysyłane przez nią komunikaty widzą wszyscy znajomi, podczas gdy na Google+ dostęp do nich mają jedynie określone przez użytkownika kręgi odbiorców. W ten sposób sama zmiana technologiczna wymusza inny model podmiotowości użytkownika. Nie oznacza to oczywiście skrajnego determinizmu w podejmowaniu działań, lecz wskazuje raczej, że przez sam fakt używania określonego oprogramowania kontroli poddawane są nie tylko komunikaty werbalne, ale także emocje (afekty) wpisane w cały program społecznościowy jako jego podstawowy wymiar

<sup>49</sup> Czy też postracjonalnej, jeśli pod uwagę bierzemy racjonalność technologiczną, która nie wpisuje się całkowicie w teorię racjonalności.

przejawiania się. Wybór pomiędzy określonymi technologiami społecznościowymi jest tożsamy z podjęciem decyzji dotyczącej funkcjonowania danej podmiotowości. Nie istnieje sprzeczność pomiędzy równoczesnym wykorzystaniem wielu programów społecznościowych, na przykład tych dwóch, które stanowią tutaj przykład analiz, jednak wtedy człowiek musi wykształcić odpowiednie techniki adaptacji – być wrażliwym na zmiany w perspektywie stosowanych przez siebie mikrotechnik, przełączając się między różnymi programami doświadczenia.

Podsumowując, Facebook należy do sfery zgeneralizowanego programu kulturowego doświadczenia, bowiem przekroczył aspekt własnego funkcjonowania jako programu komputerowego. Google+ pomimo większych możliwości personalizacji oraz innego szeregu rozwiązań technicznych nie został wpisany w te kulturowe programy i pełni rolę medium społecznościowego o zdecydowanie bardziej ograniczonym zakresie funkcjonowania w porównaniu z portalem Facebook. Umysły rozproszone – konsekwentny paradygmat oprogramowania umysłu funkcjonuje na portalu Facebook, podczas gdy na Google+ mówimy o symbolicznym paradygmacie *mindware'u*, bowiem celem nie jest łączenie wszystkiego w jeden zasób uwagi i wirtualnie dostępnej sfery interakcji, ale selekcja i dzielenie zasobów spośród sprofilowanych pod względem uwagi użytkowników. Strategią Google+ było przeprogramowanie podmiotowości użytkowników, tak by doskonale wpisywała się w ich program. Jednak w takim wypadku najczęściej nie są brane pod uwagę czynniki czysto ludzkie (pełniące w zależności od kontekstu rolę katalizatora lub inhibitora kulturowego programowania człowieka), jak nawyki użytkownika znajdujące swoją podstawę (skupienie) w danym modelu podmiotowości, czyli w konkretnym użytkowniku modelowym.

Facebook, zarządzając subiektywnością, ustala podmiotowość w swoim użytkowniku modelowym. Nie jest to możliwe w przypadku Google+, ponieważ produkuje on jedynie słabe subiektywności podporządkowane pustemu użytkownikowi modelowemu. Jest to spowodowane tym, że produkcja subiektywności nie wystarcza do tego, by wytworzyć wyposażenie podmiotowości. Użytkownik chce produkować subiektywność, ale nikt mu w tym nie pomaga, nikt nie zwraca na to uwagi. „Nikt” jest definiowany jako nieobecność energii subiektywizacji, czyli wirtualności subiektywizacji. Zachodzi także uaktualnienie podmiotowości poprzez instalację nowego *mindware*: przejście od użytkownika do administratora i od administratora do użytkownika. Podmiotowość tkwi w oprogramowaniu, jest z nim połączona, nie będąc jednak z nim zupełnie zintegrowana. Możliwe jest także przełączanie pomiędzy różnymi *mindware'ami*.

Subiektywności użytkowników nie dają jednej kolektywnej subiektywności, podobnie jak działające wspólnie inteligencje nie sumują się w postaci inteligencji kolektywnej (tym bardziej intelektu ogólnego – raczej można tutaj mówić o zmyśle wspólnym lub współtworzonym obrazie myśli). Dochodzi jednak do ich łączenia, wymiany, ale zawsze przez model-interfejs subiektywności, który zastępuje symboliczne relacje instytucjonalne (szkoły, rodziny, religii, polityki). Uwolniona przez technikę subiektywność nie jest zupełnie anarchistyczna (poza prawem), lecz poddaje się wtórnemu procesowi modelowania zgodnie z pewnymi wzorami, które zostały określone jako użytkownicy modelowi. Subiektywność dołącza się do takiego wzoru (przejmując nawyki użytkownika, pamięć oraz oczekiwania przyszłych wydarzeń), przez co wyprodukowany zostaje *użytkownik modelowy jako złudzenie autonomizacji i wolności techniki*. W modelu są zawarte sposoby stawiania się, podporządkowane konkretnym wzorom podmiotowym. Subiektywność jest uwalniana z jednego modelu, by przenieść się do kolejnego. Ten transfer subiektywności dokonujący się technicznymi środkami powoduje, że subiektywność rozsądza podmiotowość, by znaleźć opór w strukturze transcendentalnej modelu użytkownika. Przy odpowiedniej sile może go zmienić, zaproponować własną postać użytkownika modelowego, albo też zniszczyć go. Wszystkie te operacje znajdują się jednak na poziomie subiektywności – pozostałe są *tymczasowo ustalającymi nadbudowami immanentnie niestabilnego technicznego środowiska, skupieniem subiektywności w użytkowniku modelowym poprzez wprowadzenie chwilowych struktur podmiotowych zdolnych do uruchomienia i zarządzania (filtrowania) głównych produkcji subiektywności*. Człowiek-Windows staje się człowiekiem-grą komputerową, po czym staje się równocześnie człowiekiem-komunikatorem i człowiekiem-pocztą elektroniczną. Każda z tych form podmiotowości konkretyzuje się w użytkowniku modelowym, który stanowi ich przybliżające określenie. Człowiek staje się administratorem własnych podmiotowości, dzięki czemu może uruchomić własną subiektywność zdolną do przełączenia modelu użytkownika.

*Mindware* jest efektem synergii uruchomionych w subiektywności i zasilanych subiektywnością programów doświadczenia na poziomie podmiotowym. Użytkownik modelowy stanowi zasadę organizującą *mindware* – umożliwia on istnienie oprogramowania umysłu.

*Mindware* zostaje tutaj uruchomione w funkcji kulturowego systemu operacyjnego subiektywności. Staje się kulturą użytkowników modelowych, do których pretendują podmioty pozbawione zewnątrz. Jest to sztuczne zewnątrz, sztuczna granica ludzkiej tożsamości, która pozwala na zarządzanie subiektywnością.

Pora zakończyć metafizyczną konkluzją: języki programowania kultury jako uniwersalne języki sztuczne stają się powoli podstawą naszej ontyczności, fundują metafizykę oprogramowania subiektywności, wprowadzając nowego rodzaju skupienie umysłu w rozproszeniu programów doświadczenia, to znaczy wprowadzając użytkowników modelowych, zarządzają subiektywnością. Skrypty, możliwości konfiguracji programowanego bycia składają się na szereg procesów, co sprawia, że programowanie jest byciem zapośredniczonym przez media techniczne i realizuje się, nie wymagając określonych kompetencji. Jest ono wezwaniem do bycia-programowanym i bycia-programującym, gdzie odniesieniem jest sam podmiot oraz samo bycie. Użytkownik modelowy produkuje subiektywność – jest jej administratorem i zarazem użytkownikiem. Podmiot staje się użytkownikiem modelowym – zyskuje postać właściwą dla danego idealizmu połączeń, dzięki któremu zyskuje tożsamość. Dochodzimy zatem do podsumowującego wniosku: mój umysł jest tym oprogramowaniem, które uruchamiam i któremu powierzam troskę o siebie, dlatego też podmiotowość implikowana przez oprogramowania zyskuje skupienie w użytkowniku modelowym jako sterowniku poszczególnych programów doświadczenia. Podmiot w funkcji użytkownika modelowego jest zatem administratorem uruchomionych we własnej subiektywności programów doświadczenia.

