

ANNA OLSZEWSKA  
MAREK DŁUGOSZ

# AUTONOMIA KODU W SZTUCE MEDIÓW

## PRZYKŁAD RZEŻBY CYBERNETYCZNEJ SENSTER EDWARDA IHNATOWICZA

### ANNA OLSZEWSKA

Doktor, humanistka, zajmuje się historią idei utrwaloną w porządku wizualnym. Doświadczenie historyka wykorzystuje w projektowaniu interaktywnych systemów cyfrowej analizy obrazu. Związana z Gabinetem Rycin Biblioteki Naukowej PAU i PAN w Krakowie oraz z Wydziałem Humanistycznym AGH. Angażuje się w działania kuratorskie, których celem jest eksponowanie performatywnej funkcji obrazu w nauce. Prowadzi projekt Re:SENSTER.

Urządzenia analogowe określają zakres i możliwości funkcjonowania kodu. Kod porządkuje zadany schemat interakcji. Te dwa elementy dzieła-systemu pozostają względem siebie zależne. Jednakże relacja pomiędzy analogowymi podzespołami sterowania i kodem nie jest symetryczna. Z jednej strony, im więcej elementów wykonawczych, im bogatszy zestaw sensorów, tym bardziej wszechstronny program możemy budować. Z drugiej zaś – sam program nie jest w stanie istotnie zmienić zakresu interakcji maszyny z otoczeniem.

W niniejszych rozważaniach powracamy do problemu ustanawiania relacji pomiędzy kodem a analogowymi elementami systemu sterowania<sup>1</sup>. Omawiamy tę kwestię w kontekście działań artystycznych<sup>2</sup>. Jesteśmy przekonani, iż ustanowienie takiej relacji nie powinno być rozpatrywane jedynie w kategoriach wyboru technologicznego. Chcemy wykazać, że ustanowienie tej zależności kształtuje ekspresję dzieła.

1 F. Kittler, *There is no software*, „Stanford Literature Review” 1(9)/1992, s. 81–90.

2 Por. B. Magrini, *Confronting the Machine: An Enquiry into the Subversive Drives of Computer-generated Art*, Walter de Gruyter, Berlin 2017, s. 109–129, 135–140.

**MAREK DŁUGOSZ**

Dr inż., absolwent Wydziału Elektrotechniki, Automatyki, Informatyki i Elektroniki AGH w Krakowie, na którym w 2001 ukończył studia magisterskie. Stopień doktora nauk technicznych w dyscyplinie automatyka i robotyka uzyskał w 2009. Obecnie jest pracownikiem Katedry Automatyki i Robotyki na stanowisku adiunkta. Jego zainteresowania naukowe dotyczą teorii sterowania, systemów dynamicznych, optymalizacji, automatyki pojazdowej, a w ostatnich latach także robotów i pojazdów autonomicznych. W swojej pracy naukowej zajmuje się teorią sterowania, optymalizacją, modelowaniem systemów dynamicznych informatyką. Razem z grupą studentów zrzeszonych w KN Integra (którego jest opiekunem) projektuje i realizuje z sukcesem różne konstrukcje robotów autonomicznych. Jest autorem i współautorem 4 monografii oraz ponad 50 prac naukowych.

Odnosząc się do kwestii zagnieżdżania kodu w analogowej strukturze systemu, chcemy również pośrednio argumentować za odrzuceniem kategorii dzieła software'owego. Nawiązując do teorii notacji Nelsona Goodmana, chcemy przypomnieć, że niemożliwe jest tworzenie dzieł-skryptów, natomiast możliwe są jedynie dzieła bazujące na skryptach, a sądowi estetycznemu podlega w tym przypadku zazwyczaj amalgamat formy, sensorium oraz kodu<sup>3</sup>. Wspomniana wyżej asymetria sprawia, że działający kod nieodmiennie pozostaje pochodną warstwy analogowej.

W rozważaniach o estetycznej wartości kodu oraz jego analogowej otuliny posłużymy się przykładem jednego z założycielskich dzieł sztuki mediów. *Senster* – wielkoformatowa rzeźba cybernetyczna zaprojektowana przez Edwarda Ihnatowicza – pozostaje w tym kontekście przykładem szczególnie stosownym z uwagi na swoją historię. Pierwszą wersję sensorium oraz oprogramowania rzeźby zrealizowano około 1970 roku, według konwencji, które obecnie należą już do minionego porządku inżynierii. Wersję współczesną opracowano natomiast w latach 2017–2018, podczas prac nad reaktywacją dzieła<sup>4</sup>. Porównanie tych dwóch realizacji: historycznej oraz współczesnej daje możliwość rozpatrzenia estetycznych aspektów oprogramowania w porządku archeologii mediów.

**SENSTER: IDEA I KONTEKSTY TECHNOLOGICZNE**

Współczesne doświadczenie kulturowe nauczyło nas traktować kod jako kategorię autonomiczną. W trakcie prac nad reaktywacją rzeźby często zadawano nam pytania o oryginalny program. Nas samych

3 N. Goodman, *Languages of Art: An Approach to a Theory of Symbols*, Hackett Publishing, Indianapolis 1968, s. 127–150.

4 J. Walewska, *Portret artysty jako inżyniera. Twórczość Edwarda Ihnatowicza*, Wydawnictwo Naukowe UMK, Toruń, s. 125–148; A. Zivanovic, *The development of a cybernetic sculptor: Edward Ihnatowicz and the Senster*, [w:] *Proceedings of the 5th Conference on Creativity & Cognition*, ACM, London 2005, s. 102–108; tenże, *The technologies of Edward Ihnatowicz*, [w:] *White Heat Cold Logic*, pod red. N. Lamberta, C. Mason, MIT Press, Boston 2005, s. 95–110; A. Zivanovic, D. Boyd, *Elegant motion: The Senster and other cybernetic sculptures by Edward Ihnatowicz*, „*Kybernetes*” 1/2(40)/2011, s. 47–62; R. Bomba, *Senster (1970): Edward Ihnatowicz*, [w:] *Klasyczne dzieła sztuki nowych mediów*, pod red. P. Zawojskiego, Instytucja Kultury Katowice – Miasto Ogrodów, Katowice 2015, s. 23–27; A. Olszewska, *Senster: o przedsięwzięciu*, <http://senster.agh.edu.pl> (27 września 2018).

frapowała możliwość emulowania archiwalnej wersji kodu napisanego w assemblerze i implementowanego na maszynie Philipsa. Kwestia kodu powracała w rozmowach znacznie częściej niż pytania o mechanikę czy problemy związane z funkcjonowaniem podzespołów elektronicznych. Archiwalna wersja programu nie została jednak ostatecznie wykorzystana w procesie reaktywacji rzeźby. Przyczynę zaistniałej sytuacji wytłumaczy przytoczona niżej historia. Uporządkowanie technologicznych kontekstów powstania *Senstera* pozwala wnosić, że program był marginalnym komponentem dzieła od samego początku.

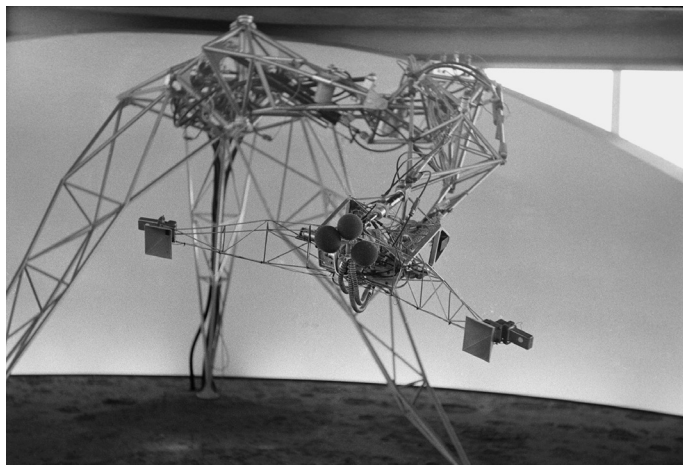
Dla realizacji rozbudowanego przedsięwzięcia, jakim był *Senster*, Edward Ihnatowicz – rzeźbiarz polskiego pochodzenia, działający w Londynie – znalazł wsparcie w osobie wpływowego kuratora sztuki Jamesa Gardnera. Pod koniec lat sześćdziesiątych Gardner zaangażowany był w projekt wystawy ulokowanej w pawilonie Evoluon, wznoszonym przez koncern Philipsa w Eindhoven<sup>5</sup>. Dla holenderskiej firmy rzeźba miała stać się trzecią poważną inwestycją w sztukę mediów – po tym, jak w 1956 roku inżynierowie koncernu zaangażowani zostali w powstanie kinetycznej instalacji Nicolasa Schöffera *CYSP 1*, a w 1958 roku Le Corbusier, Iannis Xenakis i Edgar Varèse zaprojektowali dla Holendrów słynny *Poème électronique*: pawilon-instalację immersyjną, prezentowaną w przestrzeniach brukselskiego Expo. *Senster* był więc u swojego zarania projektem artystycznym, ideą opracowaną przez rzeźbiarza. Niemniej, jego ostateczna wersja powstała w środowisku naukowo-technologicznym. W trakcie pracy nad rzeźbą Ihnatowiczowi asystowali inżynierowie Philipsa i Mullarda. Artysta został wprowadzony w środowisko londyńskiego UCL-u. Gdy ukończona rzeźba przetransportowana została do Eindhoven w połowie 1970 roku, Ihnatowicz, który na potrzeby realizacji przedsięwzięcia uczył się na bieżąco podstaw programowania, współdziałał z Peterem Lundahlem – informatykiem Philipsa oddelegowanym do pracy nad *Sensterem*<sup>6</sup>.

Z treści kontraktu zawartego wiosną 1968 roku na okoliczność realizacji projektu wynika, że ustalenia między artystą a Holendrami dotyczyły budowy „podstawy o wysokości około ośmiu stóp”, na której wsparte miało zostać „ramię» o długości piętnastu stóp, złożone z pięciu segmentów zespalanych ze stalowych rurek, poruszane elektro-hydraulicznymi serwami”. Ramię stanowiło „system transportowy dla organu sensorycznego, czyli «głowy»”<sup>7</sup>. Efektem prac miało być stworzenie „dzieła responsywno-sensorycznego” (*sensitivity-response feature*), nazywanego w dalszej części umowy „sensterem”. Wstępne warunki kontraktu określały sposób, w jaki sensoryczna „głowa” wykrywać będzie obecność ludzi w otoczeniu. Służyć temu miały dwa odrębne systemy, z których pierwszy opisano jako układ namierzania akustycznego działający na podstawie rozpoznania przesunięcia fazy pasma dźwięku (*an acoustic direction finding system operating on the basis of narrow-band-phase-shift discrimination*). Jego dopełnieniem miał być

5 J. Gardner, *The Artful Designer: Ideas off the Drawing Board*, Lavis Marketing, London 1993, s. 330–333.

6 A. Zivanovic, *The Senster*, <http://www.senster.com/ihnmatowicz/senster/> (17 września 2018).

7 C.H.M. Wirtz do Edwarda Ihnatowicza, załącznik listu określający warunki kontraktu (21 marca 1968), [w:] *James Gardner Archives*, University of Brighton, rękopis niepublikowany, 1968, s. 2.



Il. 1.  
*Senster* w *Evoluonie*  
 (Eindhoven), ok. 1970  
 (© James Gardner  
 Archive, źródło: Ja-  
 mes Gardner Archive,  
 University of Brighton  
 Design Archives).

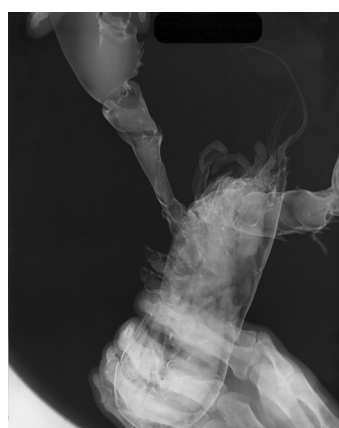
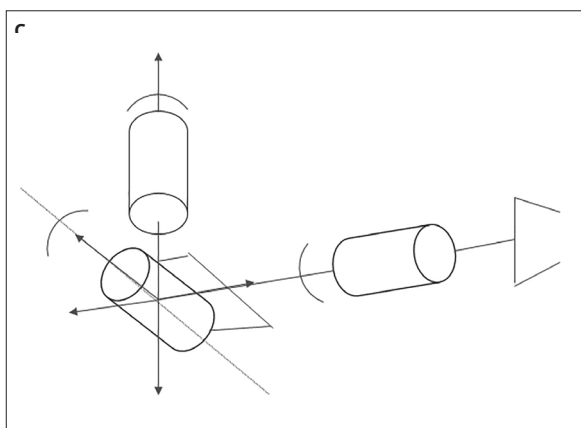
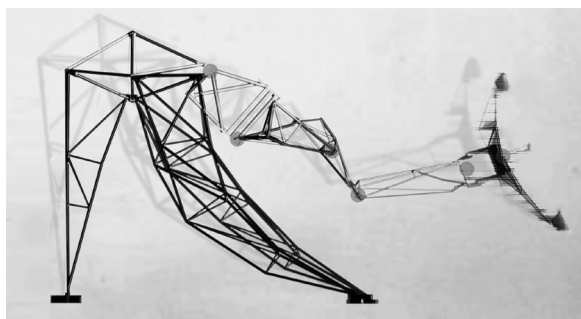
radar wykrywający bliskość i kierunek ruchu (*a radar proximity and direction detector using a semiconductor microwave generator and parabolic reflector*). Według dalszych zapisów kontraktu komputer miał odbierać sygnały zarejestrowane przez sensory w celu uruchomienia zakodowanych w maszynie procedur, które na bieżąco modyfikować miały zaprogramowany cykl ruchów rzeźby<sup>8</sup>. W odróżnieniu od szczegółowego opisu sensorium rzeźby, specyfikacja komputera nie została w kontrakcie dokładniej opisana.

Z dokumentu wynika również, że artysta nie był zobowiązany do tworzenia kodu. W myśl umowy Ihnatowicz miał przekazać zamawiającemu *Senstera* wyposażonego w elektroniczny system sterowania<sup>9</sup>. Przewidziany kontraktem model rzeźby działać miał na programie zapisanym na taśmie. W tym celu przewidywano zakup czytnika do taśm perforowanych. Opis pracy zawarty w kontrakcie traktuje kwestie programistyczne w sposób zdawkowy. W treści kontraktu problem przetwarzania sygnałów został odsunięty na dalszy plan. Sposób opisanego zamawianego dzieła wskazuje na to, że głównym kontekstem dla pracy nad wyposażoną w sensory rzeźbą była robotyka.

W czasie, gdy Ihnatowicz pracował nad *Sensterem*, robotyka uznawana była za dziedzinę integrującą najważniejsze problemy badań z zakresu AI. W laboratoriach uczelni technicznych powstawały wówczas prototypowe konstrukcje typu ramię–oko, roboty poruszające się w przestrzeni realnej, eksplorujące teren. Inteligentne maszyny stanowiły kontynuację starszej generacji projektów, które znalazły zastosowanie w przemyśle (manipulatory robotyczne) i działaniach militarnych (systemy radarowe). Prototypy służyły do prowadzenia badań podstawowych z zakresu problemów analizy obrazu, badań nad sieciami neuronowymi,

<sup>8</sup> Tamże, s. 2: „The sensory information will be fed to the computer which will process it and direct the immediate response of the head and alter by the means of pre-coded sub-routines the long cycle programme of movements of the main structure”.

<sup>9</sup> Tamże, s. 3.



II. 2.

a) schemat *Senstera* prezentujący umiejscowienie par kinematycznych (oprac. A. Olszewska); b) współczesny manipulator robotyczny (licencja CC, źródło: [https://commons.wikimedia.org/wiki/File:Uarm\\_metal\\_wiki2.jpg](https://commons.wikimedia.org/wiki/File:Uarm_metal_wiki2.jpg), fot. Siaopan: CC BY-SA 4.0); c) schemat manipulatora stawowego (oprac. A. Olszewska); d) rentgen odnóży homara, przechowywany wraz z dokumentacją *Senstera* (© James Gardner Archive, źródło: James Gardner Archive, University of Brighton Design Archives).

prób tworzenia sztucznych odpowiedników systemu nerwowego<sup>10</sup>. Brakuje informacji co do tego, czy londyńska uczelnia, w której ulokowano pracownię artysty, prowadziła pod koniec lat sześćdziesiątych projekty z zakresu robotyki. Niemniej, olbrzymie zainteresowanie, jakim cieszyły się wówczas te badania, pozwala przypuszczać, że wchodzący w środowisko inżynierów artysta miał okazję zapoznać się z najnowszymi osiągnięciami. Wysoce prawdopodobnym jest również, że Ilnatowicz zyskał za pośrednictwem UCL dostęp do literatury stanowiącej podstawy ówczesnych studiów inżynierskich w zakresie mechaniki i konstrukcji maszyn.

<sup>10</sup> N. Nilsson, *The Quest for Artificial Intelligence*, Cambridge University Press, Cambridge 2009, s. 189–198, 213–235.

Porównywany z ówczesnymi dokonaniem inżynierii *Senster* prezentuje się jako alternatywna wersja robota przemysłowego. Rzeźbiarską w swoim zamyśle formę wyróżnia szereg odwołań do kanonu inżynierii mechanicznej i kinematyki<sup>11</sup>. Ruchoma część szkieletu *Senstera* jest manipulatorem o sześciu stopniach swobody. Oznacza to, że mechanizm jest w stanie przenieść zakończenie ramienia w dowolny punkt trójwymiarowej przestrzeni. Rozwiązania tego typu wyróżniały się niską redundancją ruchu, a co za tym idzie, gwarantowały prostotę układu, ograniczenie do minimum liczby składających się na nią ogniw. Prototypy złożone z większej liczby ogniw, które w konsekwencji charakteryzują się trudnym do oprogramowania ruchem redundantnym (miękkim, elastycznym), przed nastaniem lat osiemdziesiątych stosowane były stosunkowo rzadko. Spotkać je można w tak zwanych urządzeniach sterowania ciągłego (*continuous control devices*)<sup>12</sup>. Ruch redundantny, realizowany jednak z pominięciem sterowania komputerowego, został przez Ihnatowicza wykorzystany w jego wcześniejszej pracy *SAM – Sound Activated Mobile* (1968). Decydując się na ograniczenie konstrukcji do sześciu stopni swobody, Ihnatowicz zyskiwał możliwość uproszczenia formy oraz sprawnej kontroli ruchu. Sprzyjał temu również prosty sposób zespolenia poszczególnych ogniw rzeźby. Połączone zawiasami człony tworzyły jeden ciąg kinematyczny. Układ zastosowanych połączeń pozwala zaliczyć *Senstera* do grupy manipulatorów antropomorficznych – powstałych ze złożenia przynajmniej trzech ogniw obrotowych. Układy tego rodzaju wzorowane są na anatomii kończyn organizmów żywych. Sam artysta podkreślał, że inspiracją dla konstrukcji była budowa odnóży homara. Tego rodzaju odniesienia do świata natury rekomendowano w traktatach mechanicznych przynajmniej od XIX wieku.

Specyfika kontekstu technologicznego rzeźby ujawnia się również w wyborze oraz sposobie wykorzystania czujników, które miały za zadanie otwierać zaprojektowany przez Ihnatowicza układ na interakcję ze światem zewnętrznym. W świetle współczesnych projektowi zasad wyposażania maszyn w inteligentne systemy interakcji wybór sensorium wykracza poza kanon inżynierski. Jak wspomnieliśmy wyżej, *Senster* powstał w czasach, gdy najbardziej intensywne badania prowadzone były na układach typu ramię–oko, czyli na prototypach wyposażonych w kamerę wizyjną oraz oprogramowanie analizujące obraz. Maszyny cybernetyczne tego typu określono wkrótce mianem robotów trzeciej generacji – w odróżnieniu od konstrukcji wyposażonych w prostsze, niewizyjne sensory. Zaopatrzony w zestaw mikrofonów oraz namierzające ruch diody Gunna *Senster* był zatem ukształtowany według starszego schematu. O takim wyborze zdecydować mogły ograniczenia sprzętowe systemu kontroli. Analiza obrazu w czasie rzeczywistym wymagała większej mocy obliczeniowej maszyn. Była również bardziej skomplikowana pod względem algorytmicznym. Niemniej, odwołanie do starszych, prostszych w opracowaniu schematów nie było w tym przypadku

11 R.N. Jazar, *Theory of Applied Robotics: Kinematics, Dynamics, and Control*, Springer, New York–Dordrecht–Heidelberg–London 2010, s. 8–14.

12 J. Calamia, *Artifacts from the first 2000 years of computing*, „IEEE Spectrum” 5(48)/2011, s. 34–40.

równoznaczne z powieleniem zastanych konwencji konstruowania systemów reponsywnych. Uwagę zwraca sposób wykorzystania sygnałów przesyłanych do programu sterującego ruchem rzeźby. Mikrofony wykorzystano do śledzenia źródła dźwięku – obecności ludzi. Sygnał przesyłany z sensorów – w momencie, gdy stawał się zbyt intensywny – uruchamiał tryb alarmowy. Schemat ten wydaje się kompletnie nie pasować do procedur rekomendowanych w ówczesnej inżynierii. Rozpięte na krańcach zwieńczenia rzeźby diody Gunna, które są rodzajem radarów wyposażonych w charakterystyczne anteny stożkowe, stosowano zazwyczaj do śledzenia obiektów świata zewnętrznego, mikrofony zaś przeznaczano do pracy w trybie alarmowym<sup>13</sup>. Sensorium rzeźby stanowi więc, podobnie jak i jej konstrukcja, przykład nieortodoksyjnego przetworzenia aktualnych konwencji projektowania inżynierskiego.

Oprogramowanie złożonego systemu, jakim był *Senster*, stanowiło ostatni etap prac nad rzeźbą. Korespondencja zachowana w archiwach Jamesa Gardnera świadczy o tym, że jeszcze w początkach 1970 roku nie było jasności co do tego, jak wyglądać będzie wspomniany w kontrakcie długi cykl ruchu rzeźby. Od Edwarda Ihnatowicza oczekiwano podania precyzyjnych – wyrażonych w zakresach liczbowych – parametrów ruchu. Właściwy skrypt powstał po przetransportowaniu *Senstera* do Holandii w połowie 1970 r. Podczas kilku miesięcy poprzedzających premierę zmagano się z konfiguracją przygotowanej przez artystę pracy z dostarczonym przez firmę komputerem. Kod był w tej sytuacji raczej wypadkową ograniczeń obu układów niż swobodnie ukształtowaną całością.

### SENSTER: PORÓWNANIE FUNKcjONALNOŚCI KODÓW

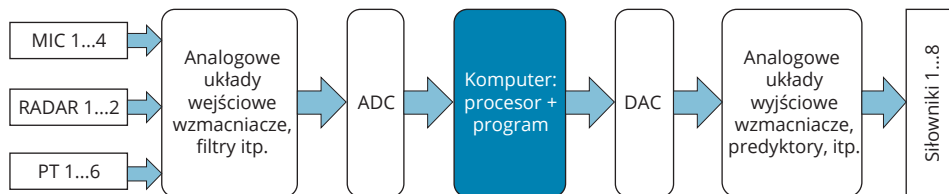
Zarówno podczas pierwszej instalacji rzeźby, jak i w czasie jej reaktywacji na ostateczną strukturę układu sterowania wpływ miały nie tylko rozwój i ewolucja samych języków programowania, ale również rozwój architektur komputerów.

W trakcie pierwszej instalacji pracy elektronicznym „mózgiem” *Senstera*, odpowiedzialnym za interpretację otoczenia oraz interakcje z nim, był najprawdopodobniej komputer firmy Philips P9201<sup>14</sup>. Był to *de facto* komputer typu DPP-416, produkowany przez firmę Computer Control Company, a sprzedawany do roku 1971 przez firmę Philips. P9201 był wyposażony w procesor 8-bitowy, długość rozkazu wynosiła 24 bity, dostępne było tylko 8kB pamięci (na program i dane), cztery rejestry wejściowe i cztery rejestry wyjściowe (obydwa rodzaje rejestrów 16-bitowych). Fizycznie komputer Philips P9201 ważył prawie 115 kg i zużywał prawie 1 kW energii elektrycznej. Integralną częścią systemu sterowania rzeźby były również analogowe układy elektroniczne wspomagające pracę komputera.

Lata sześćdziesiąte i siedemdziesiąte to początki upowszechniania się komputerów, które zaczęto wykorzystywać do obliczeń naukowych, a także w układach sterowania automatyki. Ze względu na ograniczenia technologiczne komputery z tamtego okresu wykorzystywane w systemach sterowania musiały być

<sup>13</sup> E.G. Johnsen, W.R. Corliss, *Teleoperators and human augmentation*, [w:] AEC Nasa Tech Survey (615), Washington (DC) 1967, s. 19.

<sup>14</sup> A. Zivanovic, D. Boyd, *Elegant motion...*, dz. cyt., s. 51.



Il. 3. Hybrydowy układ sterowania rzeźbą *Senster*, ok. 1970 (oprac. M. Długosz).

wspomagane różnego rodzaju urządzeniami analogowymi, takimi jak filtry czy układy kształtujące wyjściowy sygnał sterowania. W konsekwencji tworzyły one hybrydowe układy sterowania. Hybrydowe, ponieważ wykorzystywały one zarówno urządzenie dyskretnie, jakim był komputer, jak i współpracujące z nim analogowe układy elektroniczne.

Zamieszczona powyżej ilustracja przedstawia blokowy schemat hybrydowego układu sterowania rzeźby. Każdy z poszczególnych podzespołów (na przykład analogowe układy wejścia, przetworniki ADC/DCA itd.) stanowiły fizycznie odrębne urządzenia. Komputer wraz z oprogramowaniem był tylko jednym z elementów składowych w procesie sterowania, aczkolwiek bardzo istotnym.

Językiem wykorzystanym przy pisaniu pierwotnego programu dla *Senstera* był asembler. Język ten jest w zasadzie nakładką na język maszynowy (język bezpośrednio wykorzystywany przez procesory), która zamiast korzystania z numerycznych kodów instrukcji procesora (na przykład 34, 21, 12) umożliwia korzystanie z mnemonik rozkazów (na przykład ADD, MOVE, INC), co w znaczny sposób ułatwia programowanie przez człowieka. Języki asembler zaliczane są do grupy języków niskopoziomowych. Tworzenie programów z wykorzystaniem tego języka jest zadaniem żmudnym, wymagającym sporo wysiłku i wyobraźni od programisty. Pomimo korzystania z mnemonik rozkazów procesora, kod programu stworzony w języku asembler jest bardzo trudny do zrozumienia. Ponieważ każdy rozkaz (mnemonik) języka asembler odpowiada w zasadzie pojedynczemu rozkazowi języka maszynowego, to programy stworzone w tym języku są bardzo szybkie. Stąd też jest on wykorzystywany do pisania większych fragmentów programów krytycznych czasowo.

Współcześnie do programowania wykorzystuje się najczęściej języki wysokiego poziomu (na przykład C, C++, Java, Python, ST). Kod programu tworzony z wykorzystaniem jednego z języków wysokopoziomowych przypomina w znacznie większym stopniu naturalny język, jakim posługują się ludzie. W ostatnich latach można również zaobserwować rozwój graficznych języków programowania, w których działanie programu „rysuje się” w postaci odpowiednich bloków funkcjonalnych, a interakcje pomiędzy nimi określane są za pomocą odpowiednich połączeń. Języki wysokiego poziomu „tłumaczone” są w procesie kompilacji na język maszynowy procesora. Programy stworzone z wykorzystaniem języków wysokiego poziomu wymagają zdecydowanie większych mocy obliczeniowych procesora, jednak równoległy rozwój konstrukcji i architektury procesorów sprawia,



<pre> INA      '1500  READ CONTROL REGISTER JMP      INT2  INTERFACE 2 NOT ON STA      SAVA ANA      ='77  ARE ALL FFS OFF SNZ JMP      LOWP  YES CRA CALL     HOLD  NO - GO RESET SKP JMP      ERR1  ERROR RETURN LDA      SAVA ANA      ='4000 INTERRUPT STILL ON? SZE </pre>	<pre> ProgramV3State.CountSecondInterval :     IF SoundPos &lt; -0.65 THEN         intervalMin := -0.9;         intervalMax := -0.4;     ELSIF SoundPos &gt; 0.65 THEN         intervalMin := 0.4;         intervalMax := 0.9;     ELSE         intervalMin := SoundPos - 0.25;         intervalMax := SoundPos + 0.25;     END_IF     // Go to next state     State := ProgramV3State.SoundMap2; </pre>
---	--

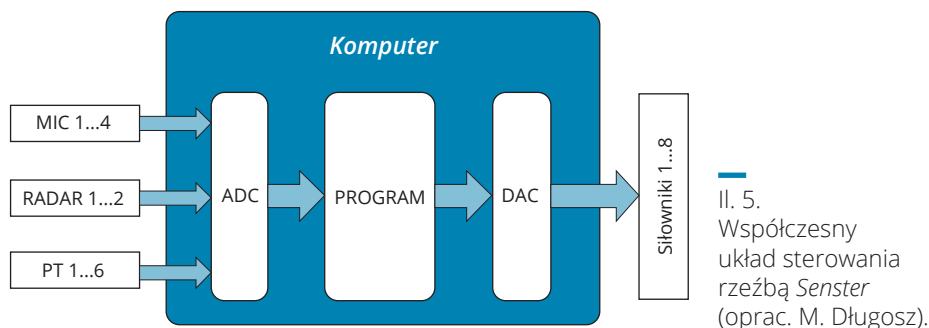
Il. 4 Fragmenty kodów *Senstera*: wersja z 1970 – przykład składni języka niskopoziomowego assembler (po lewej, źródło: <http://www.senster.com/ihnatowicz/senster/senstercomputer/index.htm>), wersja z 2018 ze składnią języka wysokopoziomowego ST (po prawej, oprac. M. Długosz).

że – pomimo wzrostu objętości i złożoności programów – działanie ich jest bardzo szybkie i wydajne.

Współczesne języki programowania oprócz możliwości funkcjonalnych posiadają jeszcze jedną bardzo ważną zaletę: możliwość uruchamiania programów zapisanych z ich wykorzystaniem na różnego typu komputerach i procesorach. Tej cechy pozbawiony jest język assembler – programy zapisane z jego wykorzystaniem czyni to hermetycznymi i ogranicza je tylko do jednego typu procesora lub niewielkiej rodziny typów procesorów.

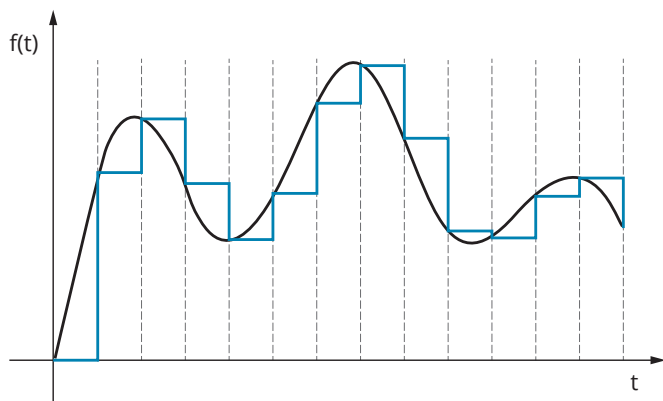
Tak więc, z uwagi na ograniczenia mocy obliczeniowej oraz ze względu na powiązaną bezpośrednio z architekturą sprzętu formułę assemblera, oryginalny kod *Senstera* ściśle „przylegał” do maszyny, na której był uruchomiony. Co za tym idzie, wprowadzenie zmian na przykład w cyklu określającym porządek ruchu rzeźby było raczej ograniczone. Sam program zaś był w gruncie rzeczy układem, który dość oszczędnie dystrybuował informacje przekazane z sensorów, chroniąc w ten sposób sam komputer przed przeładowaniem.

Podczas reaktywacji dzieła w 2018 roku postanowiono zastosować współczesny komputer przemysłowy, wraz z odpowiednimi układami wejścia-wyjścia. Do programowania wykorzystano język wysokiego poziomu ST (Structured Text), z wykorzystaniem którego starano się odtworzyć reakcje na bodźce oraz sposób ruchu *Senstera*. Część bloków funkcjonalnych układu sterowania przedstawionego na ilustracji 1 została zintegrowana z komputerem (na przykład przetworniki ADC/DAC). Wielokrotne zwiększenie mocy obliczeniowej dzisiejszych komputerów, wraz z postępem naukowym i rozwojem algorytmiki, pozwoliło na zastąpienie funkcjonalności bloków analogowych (takich jak filtry czy predyktor) odpowiednimi fragmentami programu, które działają identycznie – jeśli nie lepiej – w porównaniu z ich analogowymi odpowiednikami. W efekcie końcowym układ sterowania rzeźby uległ znacznemu uproszczeniu. Szczegóły przedstawiono na ilustracji 5.



Pomimo mniejszej liczby elementów w układzie sterowania, funkcjonalność całego systemu wzrosła. Zwiększyła się również funkcjonalność całego układu sterowania. W oryginalnym układzie sterowania rzeźby ostateczny kształt trajektorii, po której poruszały się poszczególne jej elementy, był generowany przez analogowe układy elektroniczne, tzw. predyktory. Wpływ na kształt generowanych trajektorii możliwy był jedynie w niewielkim zakresie poprzez zmianę wartości rezystancji lub pojemności w analogowych układach wyjściowych. W obecnym układzie sterowania rzeźby *Senster* kształt trajektorii ruchu jest generowany przez program i możliwości jej modyfikacji zależą jedynie od fizycznych ograniczeń układów wykonawczych (na przykład maksymalna prędkość lub przyspieszenie siłowników). Należy jednak pamiętać, że sygnał generowany przez komputer różni się od sygnału generowanego przez analogowy układ elektroniczny. Każdy komputer jest urządzeniem dyskretnym, co skutkuje tym, że generowany przez niego sygnał wyjściowy może zmieniać swoją wartość jedynie w ściśle określonych momentach, a w pozostałych – wartość sygnału jest stała (patrz ilustracja 6). Jeżeli komputer generuje sygnał wyjściowy w zbyt dużych odstępach czasu, wtedy obserwator może zauważyć znaczące różnice.

W przypadku współczesnego układu sterowania zastosowanego w rzeźbie *Senster* ten problem nie występuje. Komputer generuje dyskretny sygnał wyjściowy z tak dużą częstotliwością, że obserwator nie zauważa żadnej różnicy. Z przeprowadzonych eksperymentów i pomiarów wynika, że program sterujący rzeźbą *Senster* wykorzystuje niecałe dwadzieścia procent mocy obliczeniowej zastosowanego komputera. Paradoksalnie, stanowi to pewien problem dla odtworzenia oryginalnej ekspresji pracy. Oryginalny układ sterowania rzeźby z wykorzystaniem komputera Philips P9201 miał na tyle małą moc obliczeniową, że przy zbyt dużej liczbie danych wejściowych (na przykład wielu odczytach z mikrofonów) nie był w stanie wyliczyć poprawnego sterowania, przez co jego zachowanie było nieokreślone. Aby poradzić sobie z tym problemem w przypadku wystąpienia sytuacji przeciążenia komputera sterującego, główny program przerywał swoje działanie i ustawiał siłowniki w taki sposób, aby *Senster* „uciekł” na chwilę w inne położenie. W tym położeniu *Senster* czekał przez pewien czas, kiedy główny program znów zacznie poprawnie działać. W obecnym układzie sterowania rzeźby



Il. 6.  
Dyskretyzacja sygnału ciągłego sygnałem dyskretnym (kolor) generowanym przez komputer (oprac. M. Długosz).

*Senster* w zasadzie nie ma możliwości przeciążenia komputera i sytuacja taka musi zostać zasymulowana przez odpowiednią część programu.

Układ sterowania rzeźby *Senster*, wykorzystujący współczesny komputer wraz z wysokopoziomowym językiem programowania, daje również możliwości implementowania i testowania różnych algorytmów sterowania działających na bazie sieci neuronowych, programowania z wykorzystaniem logiki rozmytej, sieci głęboko uczonych, algorytmów genetycznych, algorytmów sztucznej inteligencji. Pierwotną inspiracją powstania wyżej wymienionych algorytmów była obserwacja działania i funkcjonowanie różnych organizmów żywych oraz zachodzących w nich procesów. Niemniej, modelowana w tak zróżnicowany sposób ekspresja pracy zasadniczo różni się od pierwotnej.

### KOD DOPASOWANY I KOD UKSZTAŁTOWANY SWOBODNIE: KONKLUZJE

Rekapitulacja historii powstania pracy Edwarda Ihnatowicza wskazuje wyraźnie, że kwestie przetwarzania sygnałów oprogramowania systemu schodziły w procesie tworzenia rzeźby na plan dalszy. Priorytetem stało się opracowanie materialnej struktury dzieła. Opisany wyżej przykład pokazuje również, że w pierwotnej realizacji systemu sterowania przeważająca część zadań była realizowana przez układy elektroniczne. Zadania programu były ograniczone do generowania podstawowych parametrów pracy układu. Historia reaktywacji rzeźby jest dobrym przykładem na to, że we współczesnej automatyce oprogramowanie odpowiedzialne jest za sterowanie w o wiele szerszym zakresie. W starszych systemach program generował jedynie statyczne stany układu. Współcześnie generuje również dynamikę układu sterowanego.

Uogólniając powyższe obserwacje, wykazać możemy, że program pozwala na działanie w formule sterowania adaptacyjnego (wysokopoziomowego). Nie ingeruje natomiast w niski poziom sterowania. W aktualnej wersji systemu sterowania niski poziom (moment, w którym przechodzimy od środowiska cyfrowego do elementów analogowych) jest ograniczony do przełożenia komend kodu na napięcia i prądy sterujące. Jako taki wynika więc jedynie z konstrukcji fizycznej komputera sterującego instalacją. Funkcjonalność i możliwości współczesnego

programu sterującego przejęły funkcje i zadania realizowane przez analogowe układy elektroniczne. W konsekwencji możliwość zmian ekspresji takiego układu (na przykład opisywanej rzeźby) sięga znacznie dalej w porównaniu z wersją realizowaną przy wykorzystaniu kodu ściśle dopasowanego do reszty systemu sterowania. Co istotne, zmiany związane z poszerzeniem mocy obliczeniowej maszyn nie objęły tych elementów systemu sterowania, które odpowiadają za interakcje z otoczeniem. Układami analogowymi koniecznymi do funkcjonowania dzieł interaktywnych pozostają wszelkiego rodzaju czujniki, takie jak mikrofony, sensory dopplerowskie, czujniki położenia i serwozawory. A zatem oddziaływanie oprogramowania na świat zewnętrzny nadal możliwe jest jedynie przy wykorzystaniu urządzeń analogowych.

Przykłady takie jak *Senster* pokazują, w jakim stopniu relacje między programem a resztą układu sterowania kształtują ekspresję dzieła. W pierwszej realizacji rzeźby ściśle dopasowany do analogowych elementów systemu sterowania kod odzwierciedlał logikę budowy układu (na przykład przeciążenie systemu skutkowało wycofaniem – zawieszeniem ruchów ramienia, niedoskonałe próbkowanie sygnału dźwiękowego czyniło reakcję nieprecyzyjną). Natomiast w wersji współczesnej swobodnie ukształtowany kod, angażujący niewielką ilość mocy obliczeniowej komputera, dopuszcza budowanie ekspresji konwencjonalnej, niemającej nic wspólnego z architekturą układu sterowania. W tych warunkach wspomniany wyżej alarm (charakterystyczne wycofanie ramienia rzeźby), opóźnienie lub brak precyzji w reakcji mogą być efektami przyjęcia pewnych wyznaczonych arbitralną decyzją twórcy założeń. Pojawienie się swobodnie ukształtowanego kodu sprawia, że ekspresja pracy traci walor odzwierciedlenia struktury układu sterowania. Swoboda działania niesie ze sobą pewne ryzyko. Operowanie kodami swobodnie ukształtowanymi może stać się estetyczną pułapką. Współczesne przesunięcie granicy między kodem a analogowymi elementami systemu sterowania doprowadzić może do zerwania ciągłości między strukturą dzieła a jego ekspresją.

Konkludując, podkreślić należy, iż swobodne kształtowanie kodu, możliwe współcześnie dzięki rozbudowaniu mocy obliczeniowych maszyn, skutkuje ujawnieniem się w ekspresji dzieła elementów (procedur) wynikających z arbitralnej decyzji projektanta. W tych warunkach ekspresja pracy rzadko kiedy bywa spójna z architekturą systemu sterowania, możliwe jest wprowadzanie różnorodnych, wymiennych wzorów interakcji. Tymczasem charakterystyczne dla starszych realizacji ograniczenia obliczeniowe skutkowały użyciem kodów ściśle dopasowanych do będących ich przedłużeniem struktur analogowych. Tego rodzaju relacja pozwalała na uruchomienie jedynie takich procedur, które pasowały do architektury systemu sterowania. Wprowadzanie ustalonych decyzją projektanta wzorów interakcji było w tych warunkach skutecznie ograniczone. Ekspresja dzieła pozostawała odbiciem jego wewnętrznej struktury.

Zaprezentowane powyżej rozważania ograniczyły się do jednego przykładu, są więc zaledwie podpozycją ukierunkowania szerszej zakrojonych studiów. Mamy jednak nadzieję, że obserwacje zebrane podczas pracy nad reaktywacją rzeźby Edwarda Ihnatowicza w nieznacznym choćby stopniu przyczynią się do

systematycznego opracowania problemu autonomii kodu w sztuce mediów. Bogaty materiał porównawczy dla przedłożonych wniosków odnaleźć można wśród współczesnych realizacji dzieł – instrumentów, sztuki odnoszącej się do fenomenu sztucznego życia. Kwestia konsekwentnego kształtowania ekspresji dzieł wykracza poza artystyczny kontekst rozwoju tej dziedziny.

## BIBLIOGRAFIA

- Bomba, Radosław. „Senster (1970): Edward Ihnatowicz”. W: *Klasyczne dzieła sztuki nowych mediów*, red. Piotr Zawojski. Katowice: Instytucja Kultury Katowice – Miasto Ogrodów, 2015.
- Calamia, Joseph. „Artifacts from the first 2000 years of computing”. *IEEE Spectrum* 5, 48 (2011).
- Gardner, James. *The Artful Designer: Ideas off the Drawing Board*. London: Lavis Marketing, 1993.
- Goodman, Nelson. *Languages of Art: An Approach to a Theory of Symbols*. Indianapolis: Hackett Publishing, 1968.
- Husbands, Phil, Paul Brown. „Not intelligent by design”. W: *Art Practice in a Digital Culture*. London–New York: Routledge, 2016.
- Johnsen, Edwin G., William R. Corliss. *Teleoperators and Human Augmentation. An AEC-NASA Technology Survey*. Washington (DC): NASA, 1968.
- Nilsson, Nils J. *The Quest for Artificial Intelligence*. Cambridge: Cambridge University Press, 2009.
- Olszewska, Anna. *Senster: dzienniki*. <http://senster.agh.edu.pl>.
- Olszewska, Anna. *Senster: o przedsięwzięciu*. <http://senster.agh.edu.pl>.
- Siciliano, Bruno. „Kinematic control of redundant robot manipulators: A tutorial”. *Journal of Intelligent and Robotic Systems* 3.3 (1990).
- Walewska, Joanna. *Portret artysty jako inżyniera. Twórczość Edwarda Ihnatowicza*. Toruń: Wydawnictwo Naukowe UMK, 2015.
- Webster, Robert J., Bryan A. Jones. „Design and kinematic modelling of constant curvature continuum robots: A review”. *The International Journal of Robotics Research* 29, 13 (2010).
- Zivanovic, Alexandar. *The Senster*. <http://www.senster.com/ihnatowicz/senster>.
- Zivanovic, Alexandar. „The development of a cybernetic sculptor: Edward Ihnatowicz and the Senster”. W: *Proceedings of the 5th Conference on Creativity & Cognition*. London: ACM 2005.
- Zivanovic, Alexandar, Boyd Davis. „Elegant motion: The Senster and other cybernetic sculptures by Edward Ihnatowicz”. *Kybernetes* 1/2, 40 (2011).
- Zivanovic, Alexandar. „The technologies of Edward Ihnatowicz”. W: *White Heat Cold Logic*, red. Nicholas Lambert, Catherine Mason. Boston: MIT Press 2009.

Data wplynięcia: 7 grudnia 2018 r. Data zatwierdzenia do druku: 26 marca 2019 r.

## AUTONOMY OF CODE IN MEDIA ART. EDWARD IHNATOWICZ'S *SENSTER* AS AN EXAMPLE OF A CYBERNETIC SCULPTURE

The authors of the text argue that any relationship established between the code and analogue elements of the operating system should be recognized as a matter of aesthetic choice. According to them with free code writing, enabled by rising computational power, certain elements (procedures) of the work may become part of artistic expression as a result of the designer's arbitrary decision. In contrast, in older installations codes had to be well-matched to ensure compatibility with the analogue components of the system because of constraints in computer performance. It was a relationship in which only these procedures could be applied which conformed to the architecture of the operating system. Consequently, the expression of such work was essentially a reflection of its structure.

**SŁOWA KLUCZOWE:** oprogramowanie, elektronika, ekspresja, sztuka mediów, *Senster*

**KEY WORDS:** software, electronics, expression, media art, *Senster*