



Colloquium 4(44)/2021
ISSN 2081-3813, e-ISSN 2658-0365
CC BY-NC-ND.4.0
DOI: <http://doi.org/10.34813/38coll2021>

MYŚLENIE KOMPUTACYJNE NA TLE TEORII UMYSŁU A DYDAKTYCZNE STRATEGIE JEGO ROZWIJANIA

**Computational thinking on the background of the theory of mind
and didactic strategies of its development**

Astrid Męczkowska-Christiansen ORCID  0000-0002-4966-7856
Janetta Charuta-Kojkoł ORCID  0000-0001-5948-022X
Joanna Zacniewska ORCID  0000-0001-9858-0768

Akademia Marynarki Wojennej im. Bohaterów Westerplatte w Gdyni
Wydział Nauk Humanistycznych i Społecznych

e-mail: a.meczowska-christiansen@amw.gdynia.pl

Streszczenie

Treść artykułu obejmuje problematykę teoretycznych aspektów myślenia komputacyjnego oraz dydaktycznych kontekstów jego kształtowania. Zagadnienia te są osadzone w kontekstach teorii umysłu i procesów poznawczych, gdzie śledzone są związki myślenia komputacyjnego z modelami umysłu i poszczególnymi typami myślenia. W rezultacie analiz zidentyfikowano dwie zasadnicze formy konceptualizacji CT: wąską – instrumentalną – oraz zintegrowaną w ramach procesów poznawczych, w tym metapoznawczych. Ich rozwijaniu towarzyszą odrębne strategie dydaktyczne: instrumentalno-behawioralna i konstruktywistyczna.

Słowa kluczowe: myślenie komputacyjne CT, dydaktyczne konteksty, konceptualizacja, teoria umysłu i procesów poznawczych.

Abstract

The paper involves problematization of theoretical aspects of computational thinking and didactic contexts of its formation. These issues are embedded in the contexts of theory of mind and cognitive processes, where the connections of computational thinking with models of mind and particular modes of thinking are traced. As a result of the analyses, two main forms of CT conceptualization are identified: narrow-instrumental and integrated within cognitive processes, including metacognitive. Their development is accompanied by distinct didactic strategies: instrumental-behavioral and constructivist.

Keywords: computational thinking CT, didactic contexts, conceptualization, theory of mind and cognitive processes.

Wprowadzenie

Problematyka dotycząca myślenia komputacyjnego jest stosunkowo od niedawna obecna w obszarze rodzimych rozważań edukacyjnych i jest podejmowana głównie w odniesieniu do kompetencji uczniów szkół średnich lub starszych klas szkół podstawowych a zarazem w powiązaniu z zagadnieniami edukacji informatycznej. W tym kontekście myślenie komputacyjne, jako jedna z treści kształcenia o charakterze formalnym, objętych podstawą programową na II i III etapie edukacyjnym, stała się obiektem zainteresowań różnorodnych dyskusji i opracowań metodycznych. W szczególności nawiązują one do pojmowania myślenia komputacyjnego jako algorytmizowania (Futschek, 2006), niekiedy wiążanego z samą umiejętnością korzystania z komputera. Tego rodzaju – *stricte* instrumentalne – odniesienia wyłoniły się już za sprawą pionierów informatyki, takich jak Alan Perlis i Donald Knuth (Denning, Trede, 2019).

Po raz pierwszy termin „myślenie komputacyjne” (*computational thinking*, CT) został użyty przez Seymoura Paperta (Papert 1980). Początkowo miał on charakter wyłącznie narzędziowy i nie powstał w związku z zaobserwowaniem jakichś specyficznych dla człowieka trybów myślenia uruchamianych w określonych sytuacjach (jak np. miało to miejsce w wypadku opisanego przez Johna Deweya procesu myślenia problemowego), lecz był odnoszony do zagadnienia umiejętności poznawczych wykorzystywanych w pracy z komputerem, w tym – umiejętności programowania (przede wszystkim w jego ograniczonym rozumieniu: jako kodowania). Tenże sposób konceptualizacji CT do dzisiejszego dnia dominuje w obszarze propozycji metodycznych, gdzie umiejętność myślenia komputacyjnego jest traktowana jako podstawa programowania, czy też rozpoczęcia nauki programowania. W odniesieniu do aktualnego globalnego dyskursu o myśleniu komputacyjnym, dynamicznie rozwijanemu głównie w ostatniej dekadzie, silnie zaznaczają się jego pozatechnologiczne odniesienia, w związku z akcentowaniem faktu, że myślenie komputacyjne nie jest tożsame z myśleniem algorytmicznym, to zaś z kolei nie dostarcza wystarczającej podstawy dla budowania kompetencji cyfrowych – nawet na elementarnym poziomie przeciętnych użytkowników IT, czyniących ich zdolnymi do wykorzystania możliwości urządzeń cyfrowych. W tym sensie, idea myślenia komputacyjnego jest wiązana nie tyle z określonymi umiejętnościami, rozumianymi w kategoriach czysto narzędziowych, lecz łączona z szerszym kontekstem procesów poznawczych człowieka, obejmującym także ich metapoznawczy wymiar. Ponadto jest ona tu traktowana jako pewien konstrukt teoretyczny, który – pomimo tego, iż wyrósł na bazie określonych spostrzeżeń empirycznych, związanych z zagadnieniami kompetencji cyfrowych i kształcenia zorientowanego na ich rozwój, wart jest włączenia w ramy szerszych rozważań o charakterze kognitywistycznym a zarazem edukacyjnym – tym razem w odniesieniu także i do pozatechnologicznych strategii rozwijania myślenia komputacyjnego, osadzanych na planie rozwoju poznawczego człowieka i nawiązujących do klasycznie już wyróżnianych (w ramach

dyskursu psychologii poznawczej) innych rodzajów myślenia, w tym – logicznego, problemowego, krytycznego i twórczego (zob. Denning, Tedre, 2019; Nardelli, 2019; Wing, 2008; Sysło, 2014; Pea, Kurland, 1984; Duncan, Bell, Tanimoto, 2014; Fessakis, Gouli, Mavroudi, 2013; Kalelioğlu, 2015).

Wielości kontekstualizacji, w których sytuowana jest obecnie problematyka myślenia komputacyjnego, towarzyszy różnorodność sposobów jego definiowania. W szerokim znaczeniu, nawiązującym do zagadnień kognitywnych, pojęcie myślenia komputacyjnego odnosi się do umiejętności systematycznego projektowania i rozwiązywania problemów poznawczych (jako zadań o złożonej i niepełnej strukturze danych, wymagających kreatywnego podejścia do konstruowania strategii rozwiązań). Powyższa konceptualizacja wydaje się stanowić pochodną naukowej dyskusji nad kompetencjami wykorzystywanymi w inżynierii oprogramowania, w której zwrócono uwagę na znaczenie szerszych kompetencji intelektualnych, w tym metakognitywnych. W takim poszerzonym znaczeniu, domeny myślenia komputacyjnego upatruje się jednak nie tylko w obszarze IT, lecz we wszystkich obszarach nauki (zarówno w domenie nauk humanistycznych jak i ścisłych), gdyż wymagają one analitycznego, logicznego, krytycznego i kreatywnego podejścia do rozwiązywania problemów poznawczych (Denning, Tedre 2019; Wing 2008; Nardelli 2019). Bywa, że myślenie komputacyjne jest definiowane wąsko – wyłącznie w nawiązaniu do myślenia algorytmicznego jako stosowanego w obrębie IT. Obydwa wyróżnione sposoby konceptualizacji odnoszą się do odmiennych (teoretycznych) modeli umysłu, na planie których osadza się problematykę myślenia komputacyjnego.

Myślenie komputacyjne na tle teorii umysłu

Źródłem pojęcia myślenia komputacyjnego są naturalistyczne, a zarazem funkcjonalistyczne teorie umysłu, w szczególności – stanowisko naturalistycznej filozofii umysłu. Ta ostatnia stawia pytania o to, czym jest umysł i na czym polega jego funkcjonowanie jako systemu przechowującego i przetwarzającego informacje. Zgodnie ze stanowiskiem naturalistycznym, umysł należy do świata fizycznego i stanowi on funkcję (biologicznego) mózgu; w tym sensie, w przeciwieństwie do mózgu, umysł ma charakter niesubstancjalny, stanowiąc skomplikowany układ funkcji, stanów i procesów realizowanych na bazie fizycznej substancji mózgu jako organu biologicznego.

Jedną z najwcześniejszych a zarazem najbardziej znanych wykładni teorii funkcjonalistycznej, obejmuje tzw. funkcjonalizm komputacyjny – określane także mianem obliczeniowego lub maszynowego – którego powstanie wiąże się z publikacją *Minds and Machines* Hilary'ego Putnama (1960). Ważnym pretekstem dla konceptualizacji komputacyjnej (obliczeniowej) teorii umysłu były także znacznie wcześniejsze dokonania Alana Turinga, odnoszące się do projektowania maszyn obliczeniowych, służących roz-

wiązywaniu określonych problemów matematycznych. Maszyna Turinga jest teoretycznym – abstrakcyjnym modelem urządzenia obliczeniowego. Urządzenie to manipuluje symbolami, podobnie jak człowiek manipuluje znakami, wykonując obliczenia arytmetyczne (Turing, 1936–37). W kontekście komputacyjnej teorii umysłu, mianem modelu turingowskiego (TMU) określa się takie informatyczne modele umysłu, które obejmują porównanie umysłu do systemów informatycznych, przetwarzających dane w sposób algorytmiczny. Na tej właśnie bazie powstała tzw. komputacyjna teoria umysłu, dominująca – m. in. za sprawą Putnama, który włączył dorobek Turinga w ramy swojej filozoficznej koncepcji umysłu w latach 60. i 70. XX w. W jej świetle – myślenie jest procesem przebiegającym z wykorzystaniem narzędzi logiki oraz symboli (określonego) języka – jest zatem procesem o charakterze obliczeniowym (komputacyjnym). Pomysły Alana Turinga i Henry’ego Putnama, a także Herberta Simona i Allena Newella, zaś na gruncie strukturalistycznego językoznawstwa – Noama Chomsky’ego, przedstawiają umysł jako „maszynę” w dużym stopniu niezależną od otoczenia, gdzie umysł pracuje mniej więcej tak jak program komputerowy. Owo oddzielenie poznania od jego kontekstu, łączy się z rozumieniem umysłu jako układu autonomicznego, zaś myślenia jako odrębnego od postrzegania i działania (Zahorodna, 2015, s. 52). Mamy tu do czynienia z pewnym teoretycznym modelem umysłu, który – choć popularyzowany m. in. w obszarze psychologii poznawczej w początkowym stadium jej rozwoju – zawiera jego redukcjonistyczną i technicystyczną wizję (zob. Hetmański 1998, s. 155–177). Warto jednak zaznaczyć, że obejmuje ono nowe podejście do problematyki poznania, określone mianem podejścia informacyjnego i całkowicie odmiennego od tych, które wcześniej były obecne w myśli filozoficznej, a także psychologicznej. Z tego też powodu twierdzi się, że za sprawą komputacjonizmu miała miejsce tzw. pierwsza rewolucja kognitywistyczna, rozgrywająca się na przełomie lat 50. i 60. XX w (Żegań, 2016, s. 43; Gardner, 1987).

Tezy wczesnej kognitywistyki, w tym Turinga i Putnama, wpisują się w dorobek tzw. klasycznego komputacjonizmu, łączonego ze stanowiskiem funkcjonalizmu maszynowego. Obecny tu tzw. komputerowy model umysłu” (*The Computer Model of the Mind*, Block, 1995) stał się jego pierwszym modelem wypracowanym w obszarze interdyscyplinarnych studiów nad poznaniem, tj. kognitywistyki. Jak wyjaśnia tę kwestię Józef Bremer, „zgodnie z nim mózg jest systemem przetwarzającym informację i zasadniczo pracuje podobnie do komputera. Rozróżnienie umysłu i mózgu da się rozumieć analogicznie do funkcjonalistycznego rozróżnienia software’u i hardware’u komputera. Podobnie jak software jest określony przez struktury danych i algorytmy, tak umysł jest określony przez reprezentacje mentalne i procesy obliczeniowe” (Bremer 2016, s. 18).

Komputacjonizm obejmuje dość prymitywny obraz reprezentacji poznawczych, w tym struktur wiedzy, tworzonych w ludzkich umysłach. Są one oparte na tzw. korelacyjnej teorii prawdy, co przekłada się na przekonanie, że umysł odzwierciedla

rzeczywistość – teza ta mieści się w obszarze reprezentacjonizmu jako obecnego w ramach niektórych stanowisk psychologiczno-pedagogicznych i epistemologicznych. Ujęcia te obecnie tracą na popularności. Pomimo tego, jak twierdzi Katarzyna Zahorodna (2015, s. 141), w dziedzinie multidyscyplinarnej kognitywistyki powszechnie używa się kategorii reprezentacji i obliczania, narzucających „komputacjonistyczne okulary” procesom postrzegania zjawisk. Współczesny komputacjonizm, jako klasyczne już stanowisko w ramach filozofii umysłu, jest określany mianem kognitywizmu (w ogóle, *sic!*) i stanowi, obok modelu koneksjonistycznego oraz teorii poznania ucieleśnionego, jeden z trzech wyróżnianych teoretycznych modeli umysłu.

Uwspółcześniona wersja komputacjonizmu uobecnia się w obszarze koneksjonizmu, rozwijającego się dzięki rozwojowi nauk o mózgu, akcentujących głównie jego biologiczny/neuronalny wymiar. Za sprawą podejścia koneksjonistycznego w latach 80. XX w. doszło do zmiany paradygmatu postrzegania umysłu, określanej mianem „drugiej rewolucji kognitywnej” (Pawelec, 2005, s. 265–278; Miller, 2003, s. 141–144). Podejście koneksjonistyczne także operuje modelem umysłu, którego działanie oparte jest na systemach obliczeniowych, jednak są one tu postrzegane jako rozproszone, działające równoległe, lokalnie w tzw. systemach sieci przetwarzania równoległe rozproszonego (Zahorodna, 2015, s. 78). Co więcej, koneksjonizm ostatecznie zrywa z tezą o algorytmicznym charakterze procesu myślenia, obejmując przekonanie, że myślenie nie polega na stosowaniu sztywnych reguł operowania symbolami; nie rządzi się twardą, a zarazem linearną logiką. Pomimo tego, że jest ono nadal traktowane jako proces komputacyjny, to podkreśla się skojarzeniowy i probablistyczny, nie zaś linearny i algorytmiczny, charakter tego procesu. Bywa ono nacechowane sprzecznościami, niejednoznacznością, niekonsekwencją, niekiedy nawet irracjonalnością (Jażyński, 2011, s. 398). Koneksjonizm odrzuca także reprezentacjonistyczne spojrzenie na kwestię treści umysłu. W miejsce nawiązań do kategorii reprezentacji, procesy poznawcze są często rozumiane w kategoriach dynamiki struktur neuronowych. Według Georga Theinera (2011, s. 5) w obszarze kognitywistyki, a przede wszystkim w odniesieniu do komputacjonistycznej teorii umysłu, pojęcie reprezentacji umysłowej jest zazwyczaj traktowane jako termin techniczny. „Odnosi się ono do występowania, przechowywania oraz transformacji struktur umysłowych obejmujących informacje” (Theiner, 2011, s. 5). (Struktury te, z naczelnym dla nich znaczeniem wiedzy i pojęć, stanowią podstawowe tworzywo dla procesów poznawczych, takich jak mowa i myślenie). Kolejnym aspektem zerwania z klasycznym modelem umysłu jest odejście od „metafory hardwarowej”, co łączy się z rezygnacją z myślenia o umyśle w kategoriach fizykalnych – jako o maszynie wykonującej określone operacje. W zamian – koneksjonizm koncentruje się na rozumieniu umysłu w terminach abstrakcyjnych sieci – co nawiązuje do zdobyczy neuro-nauk poznawczych (w ich aspekcie biologicznym), a także do intensywnych badań nad modelowaniem sieci neuronowych, prowadzonych w obszarze matematyki, informatyki i nauk pokrewnych. Koneksjonistyczny model umysłu ma ogromne znaczenie dla

współczesnych badań nad rozwojem poznawczym człowieka. Wyraża się ono przede wszystkim w tworzeniu takich modeli dynamiki rozwoju poznawczego, które nawiązują do zasad przetwarzania informacji przez sieci neuronowe (zob. Putko, 2002, Ooyen, 2003; Spencer, 2009).

Z jednej strony pojawienie się koneksjonizmu było traktowane jako wyrwanie się z kleszczy klasycznego paradygmatu komputacyjnego, z drugiej – pojawiają się tezy przekonujące, że koneksjonizm nie tyle stanowi zerwanie z komputacyjną teorią umysłu, ale jest raczej jej odmianą, akcentującą złożoność procesów przetwarzania informacji zachodzących w umyśle (zob. Pinker 2002).

Najświeższe podejście w zakresie teorii umysłu, operujące pojęciem „poznania ucieleśnionego”, nie daje podstawy dla ujmowania myślenia komputacyjnego wyłącznie przez myślenia algorytmicznego, przebiegającego w abstrakcyjnym – tj. czysto kognitywnym – trybie w oparciu na fizycznej lub biologicznej substancji centralnego układu nerwowego, traktowanego tu jedynie jako kartezyjańsko pomyślane zewnątrz dla procesów mentalnych. Podejście to uwzględnia całą egzostrukturę systemu poznawczego jako środowiska wielokrotnie złożonych i kontekstualnie osadzonych w egzosystemie (tj. cielesności człowieka, a także w jej otoczeniu) procesów i zjawisk. W obszarze teorii poznania ucieleśnionego formułuje się tezy o obecności – odmiennych od czystopoznawczych – formach reprezentacji umysłowych, obejmujących także (pomijane w klasycznym dyskursie psychologii poznawczej) reprezentacje ucieleśnione – tj. o charakterze sensomotorycznym. Dominujący sposób interpretacji tezy określa ją mianem antyrepresentacjonizmu. Podobnie jak podejście koneksjonistyczne, współczesne teorie dynamiczne w obszarze teorii umysłu stają się arcyważną podstawą dla post-piagetowskich analiz nad problematyką rozwoju poznawczego człowieka (zob. Thelen, Smith, 1994; Van der Maas, Hopkins, 1998), obejmujących zakwestionowanie niektórych też piagetowskiej teorii rozwoju poznawczego, z naciskiem na odrzucenie przekonania o przejściowym (tj. wyłącznie rozwojowym) charakterze sensomotorycznych reprezentacji poznawczych jako charakterystycznych wyłącznie dla dzieci w pierwszym stadium ich rozwoju (tj. do około 18. miesiąca życia).

Wydaje się, że najbardziej aktualne, a zarazem rozpowszechnione na świecie, dyskusje nad kształtowaniem CT w dużej mierze nawiązują do wykładni dostarczonej przez Jeanette M. Wing (2006, s. 33–35), niemal identyfikującej ten rodzaj myślenia z myśleniem w ogóle, a w szczególności z myśleniem analitycznym:

Myślenie obliczeniowe obejmuje rozwiązywanie problemów, projektowanie systemów i rozumienie ludzkich zachowań poprzez czerpanie z pojęć podstawowych dla informatyki (Wing, 2006, s. 33).

Pomimo względnej rozległości powyższej konceptualizacji, nie mamy w niej do czynienia z przekroczeniem ram paradygmatu komputacyjnego w odniesieniu do teorii umysłu. Nie wzięto bowiem pod uwagę ani sieciowego charakteru procesu myślenia,

ani też jego osadzenia w egzosferze. Można zatem powiedzieć, że dominujące w literaturze przedmiotu konceptualizacje myślenia komputacyjnego pozostają zakleszczone w obszarze searlowsko pojmowanej „syntaksy” (Searle, 2002) jako domenie myślenia maszyny (zgodnie z założeniem, że komputer nie myśli, ale mechanicznie postępuje według syntaktycznych reguł języka), co łączy się z tzw. słabą tezą w obszarze AI (*artificial intelligence*) – obejmuje ona przekonanie, że podmiot operujący myśleniem komputacyjnym zarazem wytwarza algorytmy, jak i dokonuje redukcji ekspresji własnych stanów mentalnych (nawiązując zarówno do tezy Searla, a także do hipotezy/testu Turinga). Innymi słowy: myślenie komputacyjne jako wymagające „interpretowania określonej sytuacji w kategoriach schematu, polega na dopasowaniu elementów sytuacji do ogólnych charakterystyk w ramach zredukowanej, schematycznej strukturze wiedzy” (Sarjoughian, Zeigler, 1996, s. 151). Pomimo dominacji wskazanego ograniczenia dominujących sposobów konceptualizacji myślenia komputacyjnego w domenie IT, a także w dyskursie dydaktycznym (tj. w obszarze dyskusji nad jego kształtowaniem), w obszarze kognitywistyki aktualnie dostrzega i poddaje się analizie jego związek z różnorodnymi typami myślenia i funkcjami poznawczymi, w tym: myśleniem problemowym;

- myśleniem krytycznym;
- myśleniem twórczym;
- operowaniem pojęciami przestrzennymi (por. Roman-Gonzalez et al., 2017).

Ponadto, zwłaszcza w domenie opracowań kognitywistyczno-dydaktycznych, zwraca się uwagę na efektywność jego rozwijania w domenie nietypowych (pozainformacyjnych, kooperacyjnych, etc.) środowisk dydaktycznych, obejmujących m.in. strategię *design thinking* (DT, myślenie projektowe), uczenie się przez działanie i kooperatywny model uczenia się. Tego typu próby wyraźnie idą nie tylko w kierunku tworzenia jego usieciowionej koncepcji (zob. Kiegeskorte, Douglas, 2018; Siew, Wulff, Beckage, Kenett, 2019), ale także uwzględnienia cielesnej, emocjonalnej i społecznej egzosfery dla przebiegu i rozwoju myślenia komputacyjnego. W tym sensie nawiązują one do teorii poznania ucieleśnionego.

Myślenie komputacyjne a myślenie problemowe

W świetle poszerzonej koncepcji CT, odnosi się ono do rozwiązań skomplikowanych problemów poznawczych o charakterze dywergencyjnym. Podobnie jak opracowana przez Johna Deweya (1988) struktura myślenia problemowego, także myślenie komputacyjne obejmuje określone fazy, na które składają się:

- formułowanie problemów poznawczych (tj. kompozycja) i ich identyfikacja;
- analiza określonego zbioru danych/sytuacji problemowej i jej dekompozycja, obejmująca subetapy: dekompozycji (rozkładu na składowe danego problemu) oraz zidentyfikowanie występujących w nim prawidłowości (analiza);

- abstrahowanie (eliminowania nieistotnych elementów);
- tworzenie algorytmu (rozwiązanie danego problemu) (Stańdo, Splawska-Murmyło, 2017), przy czym – odchodząc od wąsko podejmowanego kodowania jako domeny zastosowania myślenia komputacyjnego, jego ostateczny rezultat może odnosić się do tworzenia szerszego rozwiązania.

Myślenie komputacyjne jest procesem rozwiązywania problemów poznawczych o określonej specyfice (Voskoglou, Buckley, 2012; Doleck et al., 2017). W nawiązaniu do myśli amerykańskiego naukowca i wynalazcy – Charlesa Ketteringa – „problem dobrze ujęty to problem w połowie rozwiązany”. Warto zatem podkreślić, że efektywne myślenie komputacyjne wymaga szczególnej precyzji w analizowaniu struktury złożonych problemów (w sensie dydaktycznym – rozumianych jako sytuacja wymagająca od podmiotu wykreowania oryginalnego rozwiązania a zarazem sytuacja, w której zawiodą proste środki metodyczne prowadzące do rozwiązania). Myślenie komputacyjne, towarzyszące procesom rozwiązywania problemów poznawczych – bez względu na to, czy przebiega w środowisku cyfrowym, czy też nie – można scharakteryzować w następujący sposób (por. Stańdo, Splawska-Murmyło, 2017):

1. Analiza problemu wymaga logicznej organizacji danych i określenia pola niewiedzy; w zależności od typu problemu proces ten może mieć charakter bardziej lub mniej złożony i jest on powiązany z myśleniem analitycznym;
2. W ramach aktualnych podejść obejmujących poszerzoną koncepcję CT niekoniecznie zakłada się linearny charakter procesu rozwiązywania problemu – jako ciągu kroków, co niekiedy określa się mianem podejścia algorytmicznego;
3. Analiza i implementacja rozwiązania problemu prowadzą do otrzymania optymalnie efektywnego rozwiązania; w środowisku IT rozwiązanie to powinno uwzględniać jak najpełniejsze wykorzystanie możliwości urządzeń i zasobów cyfrowych;
4. Doświadczenia nabyte podczas rozwiązywania jednego problemu jest wykorzystywane do rozwiązywania innych problemów – zarówno pokrewnych, jak i pochodzących z innych dziedzin; zgodnie ze stanowiskiem Deweya, płynąca stąd wiedza ma charakter proceduralny, o charakterze niedeklaratywnym.

W niektórych propozycjach wyraźnie włącza się komponent myślenia problemowego w kontekście rozwiązywania problemów słabo-ustrukturyzowanych w definiowanie myślenia komputacyjnego (Rourke, Sweller, 2009).

Myślenie komputacyjne a myślenie krytyczne

Myślenie krytyczne jest uważane za wyższy, niealgorytmiczny, złożony sposób myślenia, który pozwalając na rzeczową ocenę propozycji strategii lub rozwiązań problemów poznawczych, przyczynia się do generowania ich oryginalnych a zarazem trafnych rozwiązań. Jest ono myśleniem celowym, zorientowanym przede wszystkim na realistyczną ocenę procesów lub wytworów intelektualnych, a także procesów motywacyjnych; może

mieć charakter autoewaluacyjny – gdy odnosi się do oceny własnych wytworów lub procesów (wówczas przypisuje mu się charakter metapoznawczy) (Nęcka, Orzechowski, Szymura, 2013, s. 428–429). Kompetencja myślenia krytycznego dostarcza metapoziomu dla – zaangażowanych w tryb myślenia komputacyjnego – takich czynności umysłowych jak analiza, synteza i ocenianie, a także stanowi warunek efektywności złożonych procesów wnioskowania, szacowania, przewidywania, uogólniania oraz – nade wszystko – twórczego wytwarzania rozwiązań. W tym sensie myślenie komputacyjne, osadzone w kontekście problemowym, jest sprzężone z myśleniem krytycznym (Kules, 2016; Denning, 2009).

Myślenie komputacyjne a myślenie twórcze

Myślenie twórcze nieusuwalnie towarzyszy myśleniu problemowemu. Jego obecność jest kluczowa zwłaszcza na etapie formułowania hipotez, gdyż efektywność ich konstruowania wymaga od podmiotu dywergencji (równoległego spojrzenia na problem z różnorodnych perspektyw) – stanowiącej podstawowy atrybut myślenia twórczego. Z tego samego powodu jest ono głęboko powiązane z myśleniem komputacyjnym – o ile to ostatnie jest rozumiane w kategoriach angażowania strategii poznawczych i metapoznawczych w aktywny proces projektowania rozwiązań, nie zaś jedynie w kategoriach myślenia stricte algorytmicznego (Romero, Lepage, Lille, 2017, s. 3). Myślenie komputacyjne jest także wiązane ze społecznym aspektem tworzenia rozwiązań w środowisku IT, rozumianym w kategoriach społecznego konstruowania wiedzy (por. Romero, Lepage, Lille, 2017, s. 3). Bywa, że samo pojęcie kreatywności zostaje bezpośrednio włączone w konceptualizację myślenia komputacyjnego (Brennan, Balch, Chung, 2014).

Koncepcja myślenia komputacyjnego, która wyłania się z prowadzonych analiz, stanowi alternatywę wobec jego wąskich ujęć dominujących do końca pierwszej dekady XX w. – akcentuje ona jego zintegrowany charakter w globalnych ramach procesów poznawczych, w tym metapoznawczych. Ponadto wyłania się tu taki obraz myślenia komputacyjnego, w którym przekracza on granice egzosfery systemu poznawczego, łącząc się ze społecznymi aspektami międzyludzkiej komunikacji w procesie rozwiązywania problemów poznawczych.

Pytanie o dydaktyczne strategie kształcenia myślenia komputacyjnego

W obszarze klasycznego dyskursu dydaktycznego odnoszącego się do myślenia komputacyjnego podkreśla się rolę, jaką ono odgrywa w procesie komunikacji z maszyną i temu zadaniu podporządkowuje się dydaktyczne strategie jego kształtowania. W tym wypadku opisuje się czynności intelektualne uczącego się podmiotu w kategoriach porządkowania danych i formułowania (klarownie ustrukturyzowanych) problemów oraz rozwiązań w taki sposób, by były zrozumiałe dla komputerów; opisuje się także czyn-

ności związane z tłumaczeniem rozwiązania na język zrozumiały dla komputera (tj. programowania, uwzględniającego kodowanie) oraz dotyczące znajdowania odmiennych sposobów zastosowania wypracowanego algorytmu/rozwiązania (tj. adaptacji do innych kontekstów). Taki sposób myślenia idzie w parze z wąskim sposobem pojmowania CT jako opartego na algorytmicznej strategii osiągania rozwiązań krok po kroku – strategii, która – w założeniu – ma charakter „wyuczalny” w odniesieniu do możliwości opanowania jej procedur w ramach modelu tzw. uczenia się bezpośredniego (Arends 1995), osadzonego w behawioralnym paradygmacie dydaktyki ogólnej. Powyższej konceptualizacji, systematycznie umacnianej od końca lat 80. XX. w., towarzyszy poszukiwanie domeny rozwoju CT wyłącznie w środowisku informatycznym, tzn. we współpracy z komputerem i określonymi typami oprogramowania. Z czasem zaczęto podejmować badania mające na celu weryfikację zasadności kształcenia myślenia komputacyjnego wyłącznie przy użyciu określonego oprogramowania (Bell et al., 2009; Brackmann et al., 2017; Torres-Torres, Roman-Gonzalez, Perez-Gonzalez, 2019), gdzie ostatecznie wykazano, że równie skuteczne jest rozwijanie myślenia komputacyjnego zarówno przy zastosowaniu specyficznego oprogramowania, jak i bez jego pomocy (Olmo-Muñoz, Cózar-Gutiérrez, González-Calero, 2020, s. 15)¹.

Aktualnie kładzie się coraz większy nacisk na kształtowanie myślenia komputacyjnego w sytuacjach problemowych – odnoszących się do problemów poznawczych o niskim stopniu strukturyzacji a zarazem w sytuacjach wymagających kreatywnego myślenia i współdziałania z innymi. Te koncepcje dydaktyczne wpisują się lub uwzględniają strategię rozwiązywania problemów określaną mianem *Design Thinking* (myślenie projektowe) (Romero, Lepage, Lille, 2017). *Design Thinking* nie stanowi odrębnego od innych trybu myślenia, lecz jest złożoną strategią osiągania celów w projektowaniu rozwiązań, obejmującą m.in. eksplorowanie, empatyczny wgląd (w ludzkie potrzeby w zakresie rozwiązań), definiowanie, ideację, prototypowanie i wykorzystanie kreatywności (zob. Brown, 2009; Skowrońska, 2019). W powyższym kontekście proces kształtowania myślenia komputacyjnego doskonale wpisuje się w koncepcję dydaktycznego konstruktywizmu, z jego komponentami społecznego konstruowania wiedzy, uczenia się przez działanie, wykorzystania kreatywności uczniów i ich „zanurzenia” w sytuacjach problemowych. Warto podkreślić, że kształtowanie komputacyjnego myślenia nie przebiega w izolacji od całościowego rozwoju poznawczych i metapoznawczych kompetencji uczniów. Integracja wielorakich czynności uczniów, uwzględnienie procesów motywacyjnych oraz różnorodnych procesów poznawczych zachodzących

¹ Niemniej hiszpańscy badacze wskazywali na minimalnie większą skuteczność rozwiązywania problemów wymagających zastosowania myślenia komputacyjnego przy użyciu stworzonego do tego oprogramowania; z drugiej strony – kształtowanie myślenia komputacyjnego na podstawie zadań nie związanych z wykorzystaniem IT łączyło się ze zwiększonym poziomem motywacji uczestników badania (Olmo-Muñoz, Cózar-Gutiérrez, González-Calero, 2020, s. 17).

w sytuacjach myślenia problemowego, a także wykorzystanie społecznego kontekstu uczenia się przy jednoczesnym zastosowaniu konstruktywistycznych strategii kształcenia w kształceniu myślenia komputacyjnego przewyciężają szablonowość i sztywność myślenia, rozwijają jego płynność i giętkość, umożliwiają wykorzystanie kompetencji i doświadczeń w nowych i niepowtarzalnych sytuacjach. Konieczność konsultowania różnych rozwiązań zadań umysłowych pobudza uczniów do intensywnego myślenia, ciągłej modyfikacji pomysłów i wypróbowywania różnorodnych sposobów działania.

BIBLIOGRAFIA

1. Arends, R. (1995). *Uczymy się nauczać*. Warszawa: WSiP.
2. Bell, T., Grimley, M., Alexander, J., Freemann, I. (2009). Computer Science Unplugged. Schools students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13.
3. Bennett, V. B., Kyu, H. K., Repenning A. (2013). Computing creativity: Divergence in computational thinking. In: *Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (359–364). SIGCSE.
4. Block, N. (1995). The Computer Model of the Mind. W: E.E. Smith, D.N. Osherson (Eds.). *An Invitation to Cognitive Science, t.3, Thinking* (247–289). Cambridge, Mass.: MIT Press.
5. Brackmann, C. P., Roman-Gonzalez, M., Robles, G., Moreno-Leon, J., Casali, A., Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. W: *Proceedings of the 12th workshop on primary and secondary computing education WiPSCe 17*. (65–72). New York: ACM Press,
6. Bremer, J. (2016). Wprowadzenie. W: J. Bremer (red.). *Przewodnik po kognitywistyce* (7–36). Kraków: Wyd. WAM.
7. Brennan, K., Balch, C., Chung, M. (2014). *Creative computing*. Harvard University Press: Cambridge.
8. Brown, T. (2009). *Change by design. How design thinking transforms organizations and inspires innovation*. New York: Harper Collins.
9. Denning, P. J. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM*, 52(6), 28–30.
10. Denning, P. J., Tedre, M. (2019). *Computational Thinking*. Cambridge-Massachusetts: MIT Press.
11. Dewey, J. (1988). *Jak myślimy?* Warszawa: PWN.
12. Doleck, T., Bazelais, P., Lemay, D., Saxena, A., Basnet, R. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, 4, 355–369.
13. Duncan, C., Bell, T., Tanimoto, S. (2014). Should Your 8-year-old Learn Coding? W: *Proceedings of the 9th Workshop in Primary and Secondary Computing Education. WiPSCe 14* (60–69). New York: ACM Press,.
14. Fessakis, G., Gouli, F., Mavroudi, E. (2013). Problem Solving by 5–6 Years Old Kindergarten Children in a Computer Programming Environment: A Case Study. *Computers & Education*, 63, 87–97.
15. Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. W: G. Futschek, R.T. Mittermeir (Eds.). *International Conference on Informatics in Secondary Schools-Evolution and Perspectives* (159–168). Berlin-Heidelberg: Springer.

16. Gardner, H. (1987). *The Mind's New Science. A History of the Cognitive Revolution*. New York: Basic Books.
17. Hershkovitz, A., Sitman, R., Israel-Fishelson, R., Eguíluz, A., Garaizar, P., Guenaga, M., (2019). Creativity in the acquisition of computational thinking. *Interactive Learning Environments*, 27, 628–644.
18. Hetmański, M. (1988). Maszyna Turinga a umysł ludzki. *Annales Universitatis Mariae Curie-Skłodowska. Sectio I. Philosophia-Sociologia*, 23, 155–177.
19. Jayźński, M. (2011). Hume, metafora mózgowia i myślenie życzeniowe. *Przegląd Filozoficzny – Nowa Seria*, 4(80), 394–406.
20. Kalelioğlu, F. (2015). A New Way of Teaching Programming Skills to K-12 Students. *Computers in Human Behavior*, 52, 200–210.
21. Kriegeskorte, N., Douglas, P. K. (2018). Cognitive computational neuroscience. *Nature Neuroscience*, 21, 1148–1160.
22. Kules, B. (2016). Computational Thinking is Critical Thinking: Connecting to University Discourse, Goals, and Learning Outcomes. *Proceedings of the Association for Information Science and Technology*, 53(1), 1–6.
23. Miller, G.A. (2003). The Cognitive Revolution: In a Historical Perspective. *Trends in Cognitive Science*, 7(3), 141–144.
24. Nardelli, E. (2019). Do we really need Computational Thinking? *Communications of the ACM*, 62(2), 32–35.
25. Nęcka, E., Orzechowski, J., Szymura, B. (2013). *Psychologia poznawcza*. Warszawa: PWN.
26. Olmo-Muñoz J., Cózar-Gutiérrez R., González-Calero J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computer & Education*, 150, 1–19.
27. Ooyen, A. (ed.). (2003). *Modeling Neural Development*. Cambridge, Massachusetts: The MIT Press.
28. Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books Inc.
29. Pawelec, A. (2005). Druga rewolucja kognitywistyczna, W: A. Kardela, Z. Muszyński, M. Rajewski (red.). *Kognitywistyka. Problemy i perspektywy*. Lublin: UMCS.
30. Pea, R. D., Kurland, D. M. (1984). On the Cognitive Effects of Learning Computer Programming. *New Ideas in Psychology*, 2(2), 137–168.
31. Pinker, S. (2002). *Jak działa umysł*. Warszawa: Książka i Wiedza.
32. Putko, A. (2002). Koneksjonistyczne modele mechanizmów rozwoju poznawczego. *Przegląd Psychologiczny*, 45(3), 361–376.
33. Putnam, H. (1960). Minds and Machines. W: S. Hook (ed.). *Dimensions of Mind* (148–180). New York: New York University Press.
34. Román-González, M. Pérez-González, Juan-Carlos & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691.
35. Romero, M., Lepage, A., Lille (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, 14.
36. Siew, C., Wulff, D., Beckage, N., Kenett, Y. (2019). *Cognitive Network Science: A review of research on cognition through the lens of network representations, processes, and dynamics. Complexity*. Pobrano z: <https://www.hindawi.com/journals/complexity/2019/2108423/> (dostęp 21.07.2021).
37. Skowrońska, M. (2019). Miasto i myślenie projektowe. Design thinking jako skrzynka narzędziowa. *Człowiek i Społeczeństwo*, 48, 75–90.

38. Spencer, J., Michael, S.C. Thomas, McClelland, J.L. (2009). *Toward a new unified theory of development: connectionism and dynamical systems theory re-considered*. Oxford: Oxford University Press.
39. Sarjoughian, H. S., Zeigler, B. P. (1996). Abstraction Mechanisms in discrete-event inductive modelling. W: *Proceedings of the Winter Simulation Conference, IEEE* (748–755). <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.414.7438&rep=rep1&type=pdf>
40. Searle, J. (2002). Can Computers Think? W: D. J. Chalmers (ed.). *Philosophy of Mind* (669–675). New York: Oxford University Press.
41. Stańdo, J., Splawska-Murmyło, M. (2017). *Sposoby kształtowania u uczniów zdolności algorytmicznego rozwiązywania problemów*. Warszawa: ORE.
42. Sysło, M. M. (2014). Myślenie komputacyjne. Nowe spojrzenie na kompetencje informatyczne. W: *Materiały konferencyjne „Informatyka w Edukacji” XI*. Toruń: UMK.
43. Theiner, G. (2011). *Res cogitans extensa. A Philosophical Defense of the Extended Mind Thesis*. European University Studies, Series XX, Philosophy, Vol. 744. Frankfurt nad Menem, Berlin, Berno, Bruksela, Nowy Jork, Oksford, Wiedeń: Peter Lang.
44. Thelen, E., & Smith, L. B. (1994). *A dynamic systems approach to the development of cognition and action*. Cambridge-Massachusetts: The MIT Press.
45. Torres-Torres, Y.-D., Roman-Gonzalez, M., & Perez-Gonzalez, J.-C., (2019). Implementation of unplugged teaching activities to foster computational thinking skills in primary school from a gender perspective. W: *TEEM'19: Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*. New York: ACM.
46. Turing, A. M. (1937). On Computable Numbers. With an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42), 230–265.
47. Van der Maas, H. L. J., Hopkins, B. (1998). Developmental transitions: So what's new? *British Journal of Developmental Psychology*, 16 (1), 1–13.
48. Voskoglou, M. G., & Buckley, S. (2012). Problem solving and computational thinking in a learning environment. *Egyptian Computer Science Journal*, 36(4), 28–46.
49. Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
50. Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society: Mathematical, Physical and Engineering Sciences*, 366, 3717e3725.
51. Wing, J. (2014). *Computational Thinking Benefits Society. 40th Anniversary Blog of Social Issues in Computing*. Pobrano z: <http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>, data dostępu: 04.10.2020.
52. Zahorodna, K. (2015). *Problem reprezentacji umysłowych w rozszerzonych systemach poznawczych*. Wrocław: Wyd. Projekt Nauka.
53. Żegań, U. (2016). Znaczenie filozofii dla kognitywistyki, W: J. Bremer (red.). *Przewodnik po kognitywistyce* (39–78). Kraków: Wydawnictwo WAM.