

06

Czynnościowe kształtowanie pojęć infotechnicznych

✍ Stanisław Ubermanowicz

Ważnym składnikiem kompetencji infotechnicznych, obok wiedzy merytorycznej i sprawności realizacyjnej, jest też posiadanie wspólnego dla przedstawicieli tej dziedziny aparatu językowego, niezbędnego do konstruktywnej komunikacji. Wynika to z konieczności porozumiewania się programistów i konstruktorów nie tylko w społecznych relacjach poziomych (pomiędzy specjalistami), ale też z potrzeby transferu zasobów wiedzy na pokolenia następców, wchodzących do obszaru działań inżynierskich. Dziedziny te posługują się dziś takim mnóstwem pojęć, że brakuje słów w języku polskim na adekwatne stosowanie nazw, dlatego dość często są to kalki zapożyczeń z języka angielskiego. Tym bardziej nieodzowne staje się właściwe kształtowanie **rozumienia słów**, a jeszcze istotniejsze jest przyswojenie pojęcia z owym specjalistycznym znaczeniem w dziedzinie informatyki, inżynierii oprogramowania czy mechatroniki. Wyraźnie rozróżnia się przy tym uogólniające definicje naukowe od rozumienia potocznego. Samo wyuczenie się definicji na pamięć nie jest równoznaczne z rozumieniem pojęcia.

Na zajęciach związanych z konkretną dziedziną inżynierską lub nauką muszą być formowane zasoby wiedzy pojęciowej, będącej dorobkiem danej dyscypliny. Uczniowie nie powinni jednak uczyć się pojęć na zasadzie odtwarzania czy reprodukcji materiału. Najskuteczniejszą formą uczenia się jest działanie praktyczne. Najlepiej utrwała się w pamięci ta czynność, którą uczeń nie tylko wykonał, ale także jeśli zrozumiał, co czynił i w jakim celu. Wykonanie zadania ułatwia zrozumienie istoty danego pojęcia. W takiej sytuacji przebiega rozwojowy proces **interioryzacji** (uwewnętrzniania), polegający na przekształcaniu czynności zewnętrznych w wewnętrzną funkcję budowania struktur, śladów i wzorców w umyśle.

Znaczenie pojęć w formowaniu kompetencji

Pojęcie, to taki rodzaj reprezentacji rzeczywistości w umyśle człowieka, w którym zostają odzwierciedlone ogólne właściwości rzeczy lub stosunki między zjawiskami. Pojęcie jest umysłową abstrakcją, symbolem bądź odwzorowaniem, odnoszącym się do obiektu, cechy, zjawiska lub relacji, które często także same nie mogą być bezpośrednio obserwowane.

Do opisywania pojęć służą definicje pojęciowe lub operacyjne. Istotą tych pierwszych jest odwzorowywanie za pomocą innych, już wcześniej przyswojonych pojęć. Zwróćmy jednak uwagę na trudność opisu za pomocą innych słów np. pojęcia ‘struktura’ w taki sposób, ażeby było to zrozumiałe przez uczniów. Lepiej jest wówczas posłużyć się definiowaniem operacyjnym, łączącym poziom *pojęciowo-teoretyczny* z *empiryczno-obszernym*. Owo teoretyczne pojęcie ‘struktura’, określające formę abstrakcyjną, zostaje przekształcone w pojęcie czynnościowe *strukturyzacja*, które łatwiej jest uczniom przyswoić poprzez osobiste przeprowadzenie takiego procesu.

Oczywiście czynnościowe kształtowanie pojęć rozpoczyna się od rozwiniętego opisu słownego z zastosowaniem wyrazów bliskoznacznych, a następnie przechodzi do opisu operacyjnego, zawierającego procedury takich działań, jakie należy podjąć, ażeby doświadczyć i zrozumieć konkretne pojęcie. Z tego względu w przygotowywaniu się do zajęć, na których wprowadzane będą zupełnie nowe, trudne pojęcia, warto bardzo starannie przygotować się do dialogu z uczniem. Oprócz planowania elementarnych czynności ucznia, jakie powinien wykonywać, aby osiągnąć zoperacjonalizowane cele, trzeba przewidzieć tok rozumowania i w czasie zajęć **akcentować fazy** najbardziej istotne dla percepcji danego pojęcia. W wielu przypadkach zalecane jest przedstawianie równoległe pojęć dodatkowych, silnie związanych z pojęciem głównym.

Informatyka, mechatronika, a zwłaszcza inżynieria oprogramowania wykształciła już tak duży zasób specjalistycznych pojęć oraz metod i technik budowania systemów, że opanowanie języka tych dziedzin wiedzy oraz nabycie umiejętności posługiwania się narzędziami i technologiami staje się kluczem dającym wstęp do elitarnej klasy, zwanej **społeczeństwem informacyjnym kognitariuszy**.

Do pierwotnych zadań programistów i projektantów systemów doszły nowe, znacznie trudniejsze. Nie wystarcza już przetwarzanie informacji – niezbędne jest szersze budowanie struktur wiedzy zarówno w formach reprezentacji ikonicznych, jak i symbolicznych, a także zaawansowanych systemów sztucznej inteligencji: potrafiących interpretować fakty, analizować wydarzenia, współtworzyć kierunki rozwoju, kreować przemiany społeczne, wspierać procesy decyzyjne itd.

Cyfrowe reprezentacje świata oraz wirtualne interpretacje otaczającej nas rzeczywistości niosą z sobą znaczną, interdyscyplinarną wartość poznawczą. Pojęcia pierwotnie ukształtowane przez inżynierów stają się ważne także w innych obszarach rozwoju osobistego i społecznego. Umiejętność selekcji, porządkowania, rangowania, szukania relacji, kreowania innowacji, obrazowania zjawisk, prezentacji, tworzenia procedur itp. powinna dotyczyć przecież postępowania w każdej sferze ludzkiej działalności. Ażeby jednak właśnie w sensowny sposób postępować, trzeba zrozumieć istotę owej swoistej logiki własnych działań. Każda czynność w umyśle ludzkim przekodowywana jest na jej werbalną reprezentację, stąd – dla postępowania efektywnego niezbędne jest poznawanie poprzez bezpośrednie działanie połączone z desygnowaniem nazw operacyjnych. W ten sposób zamyka się koło powiązań pomiędzy definicjami czynnościowymi a definicjami pojęciowymi. Mechanizm ten umożliwia kształtowanie zwłaszcza w młodym wieku nawyków do działania racjonalnego.

Kształtowania pojęć poprzez implementacje

Wypracowana już dawno i doskonała, zwłaszcza przez dydaktyków matematyki, efektywna forma czynnościowego kształtowania pojęć jest wielce użyteczna i w pełni może być zintegrowana

ze Strategią Wolnych i Otwartych Implementacji. Właśnie szczególnie na potrzeby systemowego przygotowywania uczniów do coraz bardziej samodzielnej realizacji własnych wytworów programistycznych lub mechatronicznych, potrzebne jest uspojnianie czynności konkretnych z czynnościami umysłowymi.

Czynności mogą być źródłem i katalizatorem uwewnętrzniania i przyswajania znaczeń (*interioryzacja pojęć*). Najczęściej przebiegają równolegle w interakcji z myśleniem (*autosprężenie zwrotne*), a czasami też z tak bardzo pożądanymi próbami wewnętrznego werbalizowania działań (*mowa cicha*). W programowaniu zachodzi ponadto ciekawy proces odwrotny, a mianowicie – podejmowana jest czynność implementacji pomysłu, polegająca na zweryfikowaniu operacji myślowych poprzez napisanie procedury, która wykonuje to, co przedtem było tylko koncepcją.

Implementacja najczęściej oznacza praktyczną realizację koncepcji programistycznej, co polega nie tylko na przejściu od algorytmów do procedur, lecz także docelowo na całościowym rozwiązaniu wszystkich zagadnień związanych z tworzeniem aplikacji, takich jak: interfejs użytkownika, funkcje menu, sterowanie obiektowo-zdarzeniowe, taktowanie procedur zmienną częstotliwością zegara, wizualizację graficzną i animację, symulację procesów i monitowanie parametrów, strukturyzację i archiwizację danych. Wynika stąd, że implementacja może być nie tylko opracowaniem drobnego fragmentu kodu czy pojedynczej procedury, która coś wykonuje, ale także zwieńczeniem zaawansowanego procesu kształcenia informatycznego i narzędziem weryfikacji założonych celów.

Przykładowym zadaniem, które zilustruje zalecany tu sposób integracji czynnościowego kształtowania pojęć ze strategią tworzenia implementacji, jest wykonanie imitacji urządzenia monitorującego stan pacjenta i przebieg jego podstawowych procesów życiowych. Ogólny zarys projektu w aspekcie tego, co ma być *wyświetlane* na ekranie (wartość tętna i ciśnienia), co *animowane* (obraz EKG), co i w jaki sposób ma być *sygnalizowane* (przekroczenie wartości granicznych), a zwłaszcza, co ma być *symulowane* (zmiany stanów) – wszystko to musi być uczniom precyzyjnie przedstawione w fazie wprowadzającej. Olbrzymie znaczenie mają w tej fazie procesy **asymilacji** uprzedniej wiedzy osobistej oraz **waloryzacji** sfery emocjonalnej. Nauczyciel zwraca szczególną uwagę na kompleksowość implementacji, ponieważ współczesne programowanie nie jest wyłącznie pisaniem procedur. Ważny jest dziś wystrój graficzny programu, dlatego warto, aby na wcześniejszych zajęciach z grafiki komputerowej uczniowie przygotowali obrazki ilustrujące kolejne cykle elektrokardiogramu (EKG).

W pierwszej fazie projektowania obiektów dopuszczalne są odmienne koncepcje graficzne, lecz z zachowaniem zgodnego nazewnictwa obiektów (kontrolki, *widzietów*), ażeby później przy tworzeniu procedur nie było zamieszania utrudniającego prowadzenie zajęć. Z tego samego względu w fazie definiowania zmiennych oraz w fazie deklarowania procedur warto także ustalić jednolite nazewnictwo, przy czym propozycje nazw powinny być ustalane metodą „burzy mózgów”. W tej fazie dominuje metoda projektów z pracą zespołową. W toku zajęć następuje stopniowe przechodzenie do formy pracy indywidualnej.

Wypełnianie procedur powinno być w miarę możliwości wykonywane częściowo przez ucznia – w postaci **małych porcji kodu**, z natychmiastowym korygowaniem ewentualnych błędów składniowych języka, sygnalizowanych podczas przejścia do edycji następnego wiersza. Należy

dopuszczyć różnorodne, własne koncepcje uczniowskich procedur i funkcji, jeśli tylko działają i spełniają kryteria poprawności. Dotyczy to przede wszystkim procedury symulacji, która ma imitować procesy rzeczywiste, a tym samym powinna być urozmaicona, gdyż takie są cechy realnego świata.

Warto, aby trener zaakcentował charakterystyczną dla komputera możliwość działania współbieżnego – jednoczesnej symulacji, monitorowania i sygnalizacji. Chodzi o generowanie przez procedury zarówno sygnałów, które w rzeczywistym pomiarze pochodzą z zewnątrz (tj. z czujników podłączonych do pacjenta), jak też o podział zadań procesora, z przydzielaniem czasu na wykonanie procedur animacji oraz testowanie wartości krytycznych. Zadanie jest przeznaczone na poziom zaawansowany, a jeszcze bardziej rozbudowaną i spektakularną realizacją byłoby połączenie tej implementacji z pracą modułu-interfejsu, który za pomocą czujników i przetworników A/C mierzyłby rzeczywiste parametry ucznia, np. tętno.

Przykład konstruktywnej interakcji trenera i uczniów

Przyjrzyjmy się, jak na zajęciach stacjonarnych mogłaby przebiegać zasadnicza faza tworzenia implementacji „Monitor życia”¹. Celowo używam sformułowania „mogłaby” (a nie „powinna”), gdyż przebieg zajęć zawsze ma elementy adaptacyjne, zależne od odbiorców i od dynamicznych sytuacji. Z tego powodu nie jest zalecane opracowywanie i sztywne realizowanie szczegółowego scenopisu.

Niniejszy opis jest jedynie ilustracją takiej formy interakcji z uczniami, jaką podejmuje w naturalny sposób kompetentny trener. Tak, jak jemu wystarczają konspekty-scenariusze i nie są potrzebne scenopisy, tak też i on nie powinien wydawać szczegółowych instrukcji, w jaki sposób uczeń ma coś wykonywać. Powinien inicjować mikrozadania i ukierunkowywać pracę uczniów, definiując kolejne fazy postępowania i stawiając uczniom pytania naprowadzające na rozwiązanie problemu.

Na potrzeby ilustracji techniki *inquiringu* (czyli niewyręczającego wsparcia poprzez pytania) podaję tu szkic interaktywnego trybu zajęć ćwiczebnych z realizacją zadań wytwórczych w fazie przejścia do konkretnego programowania. Przedstawiam to w formie listy dialogowej, wyróżniając kursywą istotne formy indagowania.

Trener	Uczniowie
Początek programowania obiektowo-zdarzeniowego – operacjonalizacja pojęć: ‘obiekt’, ‘zdarzenie’	
– <i>Co najpierw zaprojektujemy: obiekty czy zdarzenia?</i>	– Obiekty... Przygotowaliśmy już obrazki.
– <i>Jakiego narzędzia użyjemy do wytworzenia obiektów?</i>	– Kreatora obiektów... np. Qt Creator.

1 Ubermanowicz S., Bielawska H.: *Czynnościowe kształtowanie pojęć*, „Neodidagmata”2002/2003, nr 25/26, Wyd. Naukowe UAM

Monitor ma wyświetlać obraz oscylograficzny i trzy liczby.

– *Jak mógłby wyglądać taki Monitor?* Spróbujcie naszkicować.

– *Ustalmy, co powinno znaleźć się na ekranie.*

– *Jaki obiekt użyjemy do wyświetlania oscylogramu?*

– *Jakie obiekty użyjemy do wyświetlania napisów?*

– *Jakie inne elementy warto umieścić na obudowie monitora?*

Bardzo ważne jest uzgodnienie nazewnictwa obiektów.

– *Jakie proponujecie nazwy dla obiektów, elementów monitora?*

Uczniowie rysują swoje wyobrażenia.

– Okno, monitor, obudowa, jego elementy.

– Pole obrazowe, kontener do grafiki.

– Pola tekstowe.

– Lampkę, wyłącznik...

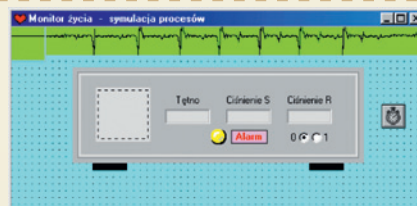
Uczniowie podają propozycje, np.: okno, obraz, panel, lampka, przycisk...

Projektowanie i konstruowanie obiektów ekranowych – operacjonalizacja pojęcia 'imitacja'

– Zbudujcie wszystkie obiekty z tymi ustalonymi nazwami.

Obudowa Monitora jest imitowana przez prostokąty i ramki. Na płycie czołowej umieszczone są pola obrazowe i tekstowe oraz napisy informacyjne. Jest też lampka sygnalizacyjna, napis alarmowy i wyłącznik. W polu okna umieszczone są wcześniej przygotowane obrazki imitujące przebieg wykresów EKG. Owe 10 pól obrazowych to ukryte kontenery, które przechowują obrazy służące animacji.

Uczniowie tworzą obiekty graficzne.



Deklarowanie wartości stałych i definiowanie zmiennych – operacjonalizacja pojęcia 'zmienna'

– Przyjmijmy nazwy i wartości określające kolory obiektów.

– *Jak nazwiemy zmienne, charakteryzujące parametry pacjenta?*

– Przyjmijmy nazwy i zadeklarujmy zmienne globalne

– *Jak nazwiemy zmienne pamiętające cykle, stany i tendencje?*

– Przyjmijmy nazwy i zadeklarujmy zmienne globalne

np. KolorAlarmu = &HFFC0FF

Uczniowie podają propozycje, np: Puls, CisnienieS, CisnienieR...

Uczniowie podają propozycje, np: Faza, Trend, Rytm, Alarm...

Programowanie obsługi ustawiania wartości początkowych – operacjonalizacja pojęcia 'wartość'

Przy uruchomieniu zakładamy losowe wartości tętna i ciśnień.

– *Jakich funkcji użyjemy do wygenerowania liczb losowych?*

– *Jakie realne wartości przyjmują u pacjenta tętno i ciśnienie?*

Omówienie struktury generowania wartości w danym przedziale.

Zależnie od języka, np. Randomize, Rnd...

Próba określenia (trener podpowiada)

Zapis i objaśnienie $Puls = \text{Int}(64 + 20 * \text{Rnd})$

Przygotowanie do zaplanowania struktury procedur – operacjonalizacja pojęcia 'proces'

Opracowanie struktury programu ułatwi ustalenie procesów.

Uczniowie wyobrażają sobie OIOM.

– Jak nazwalibyście procesy dozoru, jakie zachodzą na oddziale intensywnej opieki medycznej?

– Nie mamy tu pacjenta ani czujników pomiarowych.

– Jak nazwiemy sytuację, kiedy komputer imituje procesy, zastępując rzeczywiste środowisko i urządzenia?

– Uzgodnijmy nazewnictwo Procedur związanych z procesami

– Czuwanie, mierzenie, sygnalizacja, alarmowanie, monitorowanie, wyświetlanie...

– Symulacja, wirtualność...

– Symulator, Monitor, Sygnalizator...

Tworzenie struktury procedur obsługujących procesy

– operacjonalizacja pojęć: 'czas', 'struktura'

– Utwórzmy obsługę czasu, potrzebną do działania Monitora.

Uczniowie przypominają rolę Timera.

– Co zmienia się szybciej: obraz EKG, tętno czy ciśnienie?

Uczniowie dochodzą do tego, że aminacja EKG wymaga częstszej obsługi.

– Podzielmy procedurę obsługi Timera na 8 faz potrzebnych do animacji poprzez przełączanie 8 różnych obrazów EKG.

Uczniowie wpisują strukturę-instrukcję obsługi przypadków Select Case.

– W fazie 1 wpisujemy wywołania procedur obsługi procesów.

Symulator(); Monitor(); Sygnalizator()

Programowanie procedury Symulator(), imitującej procesy

– operacjonalizacja pojęcia 'symulacja'

– Napisz procedurę symulacji procesów życiowych, z losową zmianą tętna i ciśnienia, z przewagą tendencji do spadków.

Uczniowie próbują uzmysłowić sobie rytm pomiaru, tempo zmian i tendencje.

– W zależności od liczby losowej zmieniaj trend podstawowych parametrów życiowych.

If Rnd < .2 And Trend = 1 Then Trend = -Trend

If Rnd < .25 Then Puls = Puls + Trend

If Rnd < .3 And Trend = 1 Then Trend = -Trend

If Rnd < .35 Then CisnienieR = CisnienieS + Trend

If Rnd < .4 Then CisnienieS = CisnienieS + Trend

– Jaki wzór zastosować, aby przeliczyć częstotliwość tętna (ilość uderzeń serca na minutę) na interwał czasowy (okres pulsu wyrażony w milisekundach)?

If Puls > 0 Then Rytm = Int(60000 / Puls)

Programowanie procedury Monitor() do wizualizacji

– operacjonalizacja pojęcia 'komunikat'

– Napisz procedurę wyświetlania aktualnych wartości tętna oraz obu ciśnień w polach tekstowych na Monitorze.

To najłatwiejszy fragment kodu, który uczniowie powinni napisać samodzielnie.

Programowanie procedury Sygnalizator() do alarmowania

– operacjonalizacja pojęcia 'sygnalizacja'

– Napisz procedurę sprawdzającą przekroczenie dopuszczalnych wartości tętna lub ciśnienia (sygnał dźwiękowy, zapalenie lampki oraz zmiana koloru pól). Trener podaje wartości graniczne.

Ten fragment ma wiele funkcji warunkowych i wartości granicznych, dlatego uczniowie wpisują kod z pomocą trenera.

– Jaką strukturę posiadają funkcje warunkowe?

Uczniowie wyjaśniają logikę warunków.

Procedura musi obsłużyć reakcję EKG na śmierć pacjenta.

– Jak zamienić wykres EKG na linię ciągłą i jak wygasić dane?

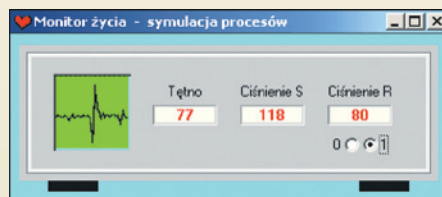
Uczniowie podają pomysły i wprowadzają wybrany sposób obsługi zdarzenia.

Programowanie procedury Aktywacja(), włączającej monitor – operacjonalizacja pojęcia 'aktywacja'

- | | |
|---|--|
| – Napisz procedurę imitującą włączenie urządzenia, z rozjaśnieniem barwy tła pól tekstowych i uaktywnieniem animacji EKG. | Uczniowie realizują pomysły na wywołanie wrażenia, że Monitor został włączony. |
| – <i>Jak uruchomić Timer wywołujący procedury cykliczne?</i> | – Trzeba ustawić interwał i stan aktywny. |
| Analogicznie napisz procedurę imitującą wyłączenie urządzenia. | Blokada Timera i wyczyszczenie pól. |

Uzupełnienie procedury obsługi Timera o kod animacji – operacjonalizacja pojęcia 'animacja'

- | | |
|--|--|
| – <i>W jaki sposób uzyskuje się efekty animacji graficznej?</i> | – Poprzez cykliczne zmiany obrazów. |
| – Uzupełnij procedurę Timera o kod, którego efektem będzie przewijanie wykresu elektrokardiogramu w polu obrazowym. | Wpisanie do obsługi 8 faz Timera poleceń zmiany sekwencji obrazków w polu EKG. |
| – Uzupełnij procedurę Timera o kod, którego efektem będzie migotanie lampki w rytmie aktualnego tętna. | Dodanie do fazy Timera poleceń zmiany obrazu lampki zapalanej/zgaszonej. |
| – Uruchom kod (debugger, kompilator lub interpreter), sprawdź prawidłowość i obserwuj efekty działania. | |
| – Zwróć uwagę na zróżnicowane tendencje do przyrostów bądź spadków wartości tętna i ciśnienia. | |
| – Porównaj zmienność tych parametrów z wartościami uzyskiwanymi na innych komputerach w pracowni. | |
| – <i>Zastanów się, jak rozbudować implementację, abyś miał możliwość działań podtrzymujących życie Twego pacjenta.</i> | |



Zadanie do wykonania samodzielnie po zajęciach – operacjonalizacja pojęcia 'implementacja'

- | | |
|--|---|
| – Wprowadź modyfikację implementacji w taki sposób, ażeby amplituda wykresu EKG zmieniała się proporcjonalnie wraz ze spadkiem lub wzrostem ciśnienia. | Jedno z najłatwiejszych rozwiązań tego zadania polega na wykorzystaniu funkcji automatycznego dopasowywania obrazu. |
|--|---|

Warto w oparciu o powyższy wzorzec konstruować autorskie jednostki zajęć z programowania lub informatyki, wprowadzając według tych samych zasad różne pojęcia z obszarów infotechniki, takie jak: organizacja, hierarchizacja, strukturyzacja, wizualizacja, konwersja, scalanie, eksploatacja, specyfikacja, definiowanie, deklaracja, algorytmizacja, iteracja, rekurencja, sortowanie, kompilacja, translacja, kodowanie, redundancja, logowanie, transmisja, interpretacja, predykcja błędów, interakcja, instalacja, animacja, ekspozycja i wiele innych. Wśród tych przykładowych pojęć znajdują się zarówno zagadnienia łatwe, jak i trudniejsze. Oczywiście należy zawsze brać pod uwagę cel zajęć, wiek uczniów i ich rzeczywiste możliwości percepcyjne.

Bibliografia

- » Ubermanowicz S., Bielawska H.: *Czynnościowe kształtowanie pojęć*, „Neodidagmata” 2002/2003, nr 25/26, Wyd. Naukowe UAM