

ANNA BOROWSKA

Uniwersytet Warszawski, Polska

Sposoby derywacji w polskim języku IT

Technologia informacyjna rozwija się bardzo szybko i efektywnie. Język, którym się posługuje do realizacji swoich celów, w skrócie nazywany językiem IT (od ang. *information technology*), należy niewątpliwie do grupy języków specjalistycznych. Jest on nie tylko podzbiorem języka naturalnego, różniącego się od języka potocznego głównie słownikiem (Bojar 1993), lecz także wariantem języka ogólnego używanym przez określone grupy zawodowe i przystosowanym do możliwie precyzyjnego opisu danej gałęzi wiedzy lub techniki (Szulc 1997). Należy na wstępie zaznaczyć, że język ten może być błędnie utożsamiany z językami programowania. Język IT nie jest językiem formalnym. Tak jak inne języki specjalistyczne pełni on trzy podstawowe funkcje: komunikacyjną, kognitywną i instrumentalną. Na gruncie polskim znany jest także jako *język komputerowców* (Cudak i Tambor 1995), *język informatyki* i *technolekt informatyki* (Borowska 2005, 2009). Zastanawiający jest jednak fakt, dlaczego język służący jednej z najprężniej rozwijających się dziedzin wiedzy nie jest przedmiotem regularnych i dogłębnych badań naukowych ani w Polsce, ani poza jej granicami. Jak wskazuje tytuł niniejszego artykułu, niżej przedstawione rozważania dotyczą obserwacji najczęstszych sposobów derywacji w polskim języku IT.

Dokonując podstawowej kategoryzacji języka IT, można wyróżnić dwa widoczne konteksty:

- 1) zastosowanie w środowisku pracy, także w dydaktyce – kontekst ściśle zawodowy;
- 2) zastosowanie poza informatyką, ale dotyczące informatyki: rozwiązywanie problemów, koordynacja pracy, przekład – kontekst „okołozawodowy”.

Istnieje także żargon zawodowy, którym posługują się zarówno informatycy, jak i zwykli użytkownicy komputerów. Stanowi on jednak odrębne wyzwanie dla badacza. Polski język IT nie jest tworem oryginalnym, lecz językiem, na który bardzo duży wpływ ma wciąż język angielski – wiodący w tej dziedzinie wiedzy

i uczestniczący w jej rozwoju dłużej niż polska wspólnota językowa. Budzi to sprzeciw tych użytkowników języka i kodyfikatorów norm językowych, którzy uznają potrzebę umiaru w korzystaniu z wyrażen obcych. Wyraźne różnice w charakterze rozpatrywanych języków nie pozwalają jednak na całkowite przejęcie lub kalkowanie leksykonu specjalistycznego z języka angielskiego. Można zatem zauważyć swoisty stopień oryginalności polskiego języka IT. Uznanie polskiego języka informatyki za oryginalny trzeba uzależnić przede wszystkim od tego, czy w składzie jego elementów leksykalnych istnieją takie, o których można powiedzieć, że są innowacyjne nie tylko po stronie wyrażeniowej, lecz także w zakresie funkcji znaczeniowych. Szanse polskiego języka IT na oryginalność wzrastają wraz ze stopniem kreatywności specjalistów informatyków i wyrażania za jego pomocą oryginalnych wyników realizowanych przez nich badań i osiągnięć. Często bywa to proces nieświadomy, ponieważ informatycy nie są lingwistami, a wszelkie tworzone przez nich nazwy mają charakter pragmatyczny. Zasadniczy zręb terminologii specjalistycznej przeznaczony jest wyłącznie dla fachowców, a przeciętny użytkownik języka nigdy danego terminu może nie użyć ani się z nim nie spotkać. Mimo to coraz częstszy kontakt z informatyką w życiu codziennym wymaga rozszerzenia zasobu słów. Poza tym, wśród tych nowych „informatycznych” wyrazów jest dużo słów obcych, które nie operują rodzimymi częstkami słowotwórczymi. Z czasem jednak każda obca pożyczka może zostać w jakiś sposób przyswojona do języka polskiego, a więc też do polskiej odmiany. Polski język IT cechuje przewaga rzeczowników i przymiotników nad czasownikami. Jednoznaczne i ściśle nazwy nasycone są maksymalnie informacją. Aby dokładnie zbadać powstawanie terminów w jakimkolwiek języku specjalistycznym, należy posłużyć się odpowiednimi parametrami. Celem niniejszego badania jest analiza sposobu tworzenia wyrazów w wybranym kontekście specjalistycznym.

Z przeglądu literatury przedmiotu wynika, że ani na polskim, ani na światowym gruncie badań naukowych analiza języka IT nie leży w kręgu zainteresowań lingwistycznych. W tym zakresie powstają tylko pojedyncze prace naukowe, głównie artykuły. Możemy zatem tylko się domyślać, na czym oparta jest derywacja w języku technologii informacyjnej. Przykłady terminów przedstawione w niniejszym tekście pochodzą z tekstów specjalistycznych dotyczących nauk informatycznych opublikowanych w latach 2014–2016 w czasopismach branżowych przeznaczonych głównie dla informatyków, tj. „Chip”, „PC World Komputer”, „Computerworld”, „Komputer Świat”, ale także z prac leksykograficznych, np. „Angielsko-polski słownik informatyczny” (2004), „Popularny angielsko-niemiecko-polski słownik terminów komputerowych i informatycznych” (2006). Przeprowadzona analiza miała na celu przedstawienie najsprawniejszych mechanizmów tworzenia terminów w polskim języku IT.

Derywacja w polskim języku informatyki pod pewnymi względami przebiega nieco odmiennie od pozostałych języków specjalistycznych ze względu na fakt, że często w tym kontekście decyduje czas, a raczej jego brak. Technologia

informacyjna nie tylko rozwija się bardzo szybko, lecz w ten sam sposób przedstawia swoje osiągnięcia. Obcy termin nazywający lub opisujący dany produkt jest już często w użyciu przed wprowadzeniem takiego produktu na rynek. Wybrany termin jest zatem często stosowany zanim pojawi się szansa znalezienia jego oryginalnego odpowiednika w języku wtórnym, jakim jest w tym wypadku język polski, nie mówiąc o jego dostosowaniu do danego systemu derywacyjnego. Gotowe terminy przechodzą więc na rynek polski i tu wiele zależy od osoby lub firmy, która pierwsza dany produkt opisze w celach naukowych lub komercyjnych. Od tego podmiotu zależy, jakich terminów użyje czy zastosuje proces kalkowania, czy też spróbuje zachować oryginalność językową. Polski język IT z jednej strony przejmuje obce jednostki terminologiczne, głównie z języka angielskiego, do swojego systemu terminologicznego, a z drugiej jego odrębność językowa nie pozwala na ich całkowitą asymilację. Polega to na tym, że leksemy obce, nieutrwalone dotychczas w leksykonie informatycznym, są automatycznie eliminowane i zastępowane rodzimymi. Proces taki przebiega bardzo szybko, bo to, co pojawia się w informatyce, musi być natychmiast nazwane. Bardzo często nowe terminy są asymilowane w języku polskim jako kalka, półkalka lub zapożyczenie z języka angielskiego.

Cudak i Tambor (1995) w swoim artykule wskazują na cytaty obce, zapożyczenia właściwe o różnym stopniu przyswojenia w języku docelowym i kalki: „czasem trudno orzec, dlaczego dana nazwa utrzymała się w języku na prawach cytatu [...]. Przykładem mogą być nazwy *write* i *paintbrush*”. Nie mają one rodzimych odpowiedników w języku polskim, choć w kontekście użytkownika otoczone są przez rodzimo brzmiące *kalendarz*, *kalkulator* oraz *zegar*. Autorzy wyjaśniają tę sytuację w sposób następujący: „W języku polskim były próby tłumaczenia nazw tych aplikacji przez abstrakcyjne nazwy czynności: *pisanie*, *malowanie*, czy nazwy subiektów agentywnych: *pisarz*, *malarz*. Jednak żadna z tych nazw nie zyskała powszechnej akceptacji i wciąż używane są cytaty z języka angielskiego, nawet w oficjalnej wersji lokalizowanej, co tę sytuację utrwala i normatywizuje. Nazwy te w języku mówionym (w specjalistycznym żargonie komputerowym) przeszły już jakby do następnego etapu przyswojenia. Wyrazy zachowują bowiem w piśmie oryginalną angielską ortografię i nieodmienność, natomiast w mowie mają niby angielską, bo spolonizowaną wymowę, nie wykazują aktywności derywacyjnej, ale już się po polsku odmieniają”.

Możemy też zauważyć wiele wyrazów, których częściowe spolszczenie morfologiczne polega na uzyskaniu przez nie zdolności derywacyjnych pomimo zachowania oryginalnej pisowni, częściowo wymowy i braku odmiany (Cudak i Tambor 1995), np. *save* – *sejwować*, *screenshot* – *zrób screenshota*. Do wyrazów spolszczonych pod względem fleksyjnym należą też *footery*, *fonty*. W celu zbadania derywacji form wyrazowych obecnie używanego przez specjalistów polskiego języka IT należy dokonać segmentacji takich form i zobrazować najczęstsze

morfemy jako podstawowe jednostki opisu systemu derywacyjnego, a także zobrazować inne sposoby derywacji wyrazów w wybranym kontekście.

1. Typologia morfemów języka IT

Forma językowa w wąskim aspekcie może być rozpatrywana jako złożona z morfemów dwojakiemu typu: morfemów tematu i morfemów końcówki, a w szerokim aspekcie jako złożona z morfemów trojakiemu typu: rdzenia, morfemów słowotwórczych i końcówki. Wyraz należy zatem rozłożyć na jego elementarne składniki, aby poznać mechanizm jego powstania. Pierwszy krok to analiza strukturalnej klasyfikacji morfemów języka IT. Zgodnie z teorią Tabakowskiej (2001), gdy w wyrazie ma miejsce stałe umiejscowienie morfemów, np. prefiksów, sufiksów, infiksów względem morfemów głównych, są one realizowane jako nieprzerwane ciągi elementów i możemy je nazwać morfemami ciągłymi (np. *minimalizować*, *minidysk*, *podsystemy*, *mikrokontroler*). Jeżeli jednak wykładnik funkcji gramatycznej występuje po obu stronach morfemu leksykalnego, mamy do czynienia z morfemami nieciągłymi, np. w angielskim języku IT: *under-rat-ed*. W języku polskim za morfemy nieciągłe uważa się prefiksalnie-postfiksalne morfemy słowotwórcze. Występują one także w języku IT i zawierają elementy takie, jak: 1) *roz-... się*, derywujący czasowniki, np. *rozkładać się*, *rozmagnesować się*, *rozprzestrzeniać się*; 2) *do-... się / de-... się* derywujący czasowniki, np. *dolączać się*, *debugować się*.

Ponadto realizację morfemu ze względu na kryterium strukturalne może stanowić także morf zerowy, który poza fleksją w języku polskim nie występuje. Wskazuje on, że w danej formie miejsce przeznaczone na morf o określonej funkcji nie jest zajęte, choć jest zajęte w innych formach tego samego leksemu (Bańko 2002:54). Morf zerowy funkcjonuje w ten sam sposób jak morfy niezerowe, różniąc się od nich jedynie typem reprezentacji fonologicznej. Paralelna jest także struktura morfologiczna form z niezerowymi i zerowymi morfami. Na przykład wyraz *urządzeniom* jest ciągiem morfów *urządzeni+om*. Tak samo złożoną strukturą morfologiczną jest wyraz *urządzeń+*. W angielskim języku IT taki morfem jest widoczny w następujących przykładach: *control – to control*, *format – to format*. Wyniki przedstawionej analizy ujęte zostały w poniższej tabeli:

Tabela 1: Strukturalne ujęcie morfemów języka IT

Typ morfemu	Angielski język IT	Polski język IT
nieciągły	<i>over...ed, under...ed</i>	<i>de...się, do...się, roz...się</i>
ciągły	<i>sub-, semi-, -er</i>	<i>mini-, pod-, -er</i>
zerowy	<i>to test</i>	<i>kod, szyn, magistral, plik</i>

Laskowski zwraca także uwagę na występowanie morfemów rdzennych: „morfem rdzenny to morfem konstytutywny formy gramatycznej, stanowiący jej struktural-

ny ośrodek, do którego mogą być dołączane inne morfemy” (1999:75). Stanowią one zatem zbiór otwarty, który, na przykład w drodze zapożyczeń, łatwo rozbudowywany jest o nowe elementy, np. *cyfr-owy*, *cykl-iczny*, *synchron-iczny*. Włączanie nowych elementów do takiego zbioru odbywa się przy przejmowaniu jednostek morfologicznych z języka prymarnego, np. *browser – przegląd-arka*, *sub-domain – sub-domena*. W przeciwieństwie do morfemów rdzennych afiksy stanowią w tym kontekście grupę zamkniętą, z trudem ulegającą poszerzeniu o nowe elementy. Język polski jest językiem fleksyjnym, a zatem wyrazy odmienne składają się z dwóch części: tematu fleksyjnego, który wyraża ich znaczenie leksykalne i jest stałym, niezmiennym członem wyrazu, oraz z morfemów gramatycznych, które charakteryzują się pod względem formalnym i są wymienne (ibid.:78).

Jeżeli rdzeń tworzący wyraz oznaczymy za pomocą litery *R*, a afiksację i fleksję za pomocą liter *d* i *f*, to dynamiczny charakter jego modelu funkcyjnego może być zarejestrowany za pomocą formuł przedstawiających poszczególne ciągi morfemów (por. Zmarzer 1992:28). Otrzymamy wówczas następujące zapisy, funkcjonujące jako swoiste formuły derywacyjne w badanym kontekście:

- *d-R*: typ prefiksalny: *pod-program*;
- *R-d*: typ sufiksalny: *program-ista*;
- *R-f*: typ fleksyjny: *program-ować*.

Powyższy schemat prezentuje najbardziej stałe typy słowotwórcze występujące w języku IT. Bardzo łatwo jest zatem wskazać więcej przykładów, stosując się do powyższych wzorów. Ponadto analizując poszczególne formanty, można zapisać inne kompozycje formuł, np. zero pozycyjne, *p* jako prefiks, *s* jako sufiks:

Tabela 2: Formuły derywacyjne języka IT

Formuła	Angielski język IT	Polski język IT
p-R-s	<i>re-programm-ed</i>	<i>za-program-ować</i> <i>prze-program-ować</i>
p-R	<i>re-program</i>	<i>pod-program</i>
R-s	<i>programm-able</i>	<i>program-owalny</i>
R-	<i>program</i>	<i>program</i>

Powyższe formuły ponownie pokazują wyraźny wpływ angielskiego języka IT na polski język IT. Ponadto, przy tworzeniu formacji czasownikowych w języku IT szczególną aktywnością wykazują się prefiksy, np. polskie prefiksy perfektywizujące (tworzące formy dokonane): *o-czyścić*, *s-pakować*, *s-kompresować*. Modyfikowanie znaczeń czasowników dokonuje się także za pomocą prefiksów obcych polszczyźnie. Jest to dość popularna tendencja. Bardzo często stosowany jest prefiks *re-* wnoszący znaczenie powtórnego wykonania czynności, np. *rede-*

finiować, reprodukować, restartować. W tej funkcji nie może być on zastąpiony przez żaden przyrostek rodzimy. Synonimiczny do polskiego *od-* jest obcy prefiks *de-*, którego dużą aktywność słowotwórczą w polskim języku IT potwierdzają następujące struktury: *dekodować, deszyfrować, deinstalator*. Do innych prefiksów zapożyczonych z języka angielskiego o wysokiej częstotliwości użycia należą: *super-* i *anty-*, np. *anti-aliasing, super-sampling*. Współcześnie w rozpatrywanym kontekście obserwuje się tendencję do tworzenia wyrazów raczej z elementem obcym, rzadziej rodzimym. Zabieg ten często widoczny jest w ściśle specjalistycznych tekstach poświęconych informatyce. Co więcej, kalkowanie terminów bezpośrednio z języka angielskiego wydaje się łatwiejsze i szybsze w życiu codziennym, np. *interaktywny, mikrokomputer, facebook, lajkować, smartfon*.

Z drugiej strony, polski język IT stara się w pewnym stopniu zachować swoją oryginalność. Wśród prefiksów rodzimych pozostają typowo polskie *wielo-*, *pod-*, *nie-*, *prze-*, *za-*, *pół-*, *między-*, *bez-* (np. *wielofunkcyjny, zablokować, bezpołączeniowy*). Wszystkie one podkreślają oryginalność polskiego języka IT, ponieważ należą do prefiksów produktywnych i są gotowe do tworzenia nowych formacji słowotwórczych. Można też stwierdzić, że w pewien sposób konkurują one z zapożyczonymi prefiksami takimi, jak: *multi-*, *sub-*, *de-*, *inter-*, *pre-*. Czasem możemy zauważyć, że ich znaczenia nakładają się na siebie, skutkując formami synonimicznymi, np. *debugować = odpluskwiać, interoperatywność = współdziałanie*.

Bardzo liczną grupę derywującą nowe leksemy stanowią również sufiksy. Bezpośrednio z języka angielskiego na grunt polski przeszły następujące sufiksy: *-ing*, *-er*, *-or*, *-yzm*. Wśród najbardziej produktywnych sufiksów rodzimych wyróżniamy: *-anie* (*programowanie*), *-ość* (*zawartość*), *-sja* (*kompresja*), *-ny* (*dostępny*), *-arka* (*przeglądarka*). Każdy z sufiksów produktywnych ma stałą i abstrakcyjną wartość, która konkretyzuje się zależnie od treści elementu, do którego sufiks został dodany.

2. Kompozycja

Mając na uwadze derywaty złożone, należy również wspomnieć o procesie słowotwórczym opartym na kompozycji. Cechą formacji utworzonych za pomocą kompozycji jest ich oparcie na dwóch lub więcej tematach, stanowiących łączną bazę słowotwórczą. Złożenia takie budowane są zazwyczaj na zasadzie hierarchicznej (Tabakowska 2001:83). Układy szeregowy, które powstają przez zespolenie co najmniej dwóch członów znaczeniowo równorzędnych, bo żaden z nich nie określa drugiego, a całe złożenie może być zastąpione przez grupę wyrazów związanych spójnikiem łącznikowym *i*, np. *kodujący i dekodujący*, raczej nie występują w języku IT. Możemy wymienić tylko kilka przykładów tego typu, np. urządzenie może być *kodująco-dekodujące, nadawczo-odbiorcze, kodowo-impulsowe*. Hierarchiczny układ członów złożenia polega na połączeniu semantycznie

nierównorzędnych członów w układzie podrzędno-nadrzędnym. Jeden z członów pełni funkcję utożsamiającą, a drugi odróżniającą. Oznacza to, że jeden utożsamia dane przedstawienie z wszystkimi przedstawieniami wchodzącymi w zakres pojęcia, drugi natomiast wyraża cechę szczegółową, która odróżnia oznaczane przedstawienie od innych przedstawień tego samego typu, a tym samym je określa (Milewski 2004:69). W zaobserwowanych złożeniach występujących w języku IT najpierw widnieje człon określający, a potem określany. Człony te mogą należeć do różnych kategorii leksykalnych, np. często obserwuje się następujące kombinacje:

- 1) przymiotnik + przymiotnik: *cienkowarstwowy, elektronopromieniowa,*
- 2) przysłówek + przymiotnik: *stałopozycyjny,*
- 3) liczebnik + przymiotnik: *jednoprzebiegowy, dwudzielny, współbieżny,*
- 4) przysłówek + rzeczownik: *półdupleks, półsumator, półprzewodnik.*

W przykładzie 2) przysłówek jest członem podrzędnym i modyfikuje znaczenie przymiotnika, określając stopień cechy wyrażanej przez przymiotnik. Procesowi kompozycji towarzyszy derywacja, tworząca za pomocą morfemu słowotwórczego *-owy, -ny* oraz interfiksów wyraz złożony z morfemów głównych zespolonych w jedną podstawę słowotwórczą. Typ interfiksów jest uzależniony od klasy gramatycznej pierwszego członu. Jest nim zazwyczaj postać interfiksów *-o-* występującego po członach rzeczownikowych, przymiotnikowych i liczebnikowych lub interfiksów *-u-* w członach liczebnikowych.

Wśród innych wyrazów złożonych występujących w języku IT można zaobserwować dodatkowo zrosty, w których brakuje wykładnika formalnego kompozycji. Między członami zachodzą wówczas relacje składniowe wyrażone fleksyjnie. Zrosty nie są jednak bardzo produktywne. Przykładem jest termin *słowo-klucz*, w którym człon wewnętrzny pozostaje nieodmienny. Człon drugi jest w tym przypadku określający (Grzegorzczkowska i Puzynina 1999). Stosowane złożenia mają charakter dwu- lub wielowyrazowych nazw jednego desygnatu, mających charakter utartych połączeń, o nieprzestawialnych członach, zazwyczaj zostają utworzone na zasadzie kalki z języka prymarnego, np. *znak kontrolny* od *check character*, *jednostka wkładalna* od *plug-in unit*.

3. Alternacja a skrótowiec

Kolejny zaobserwowany podczas badania tekstów informatycznych proces słowotwórczy to alternacja. Polega ona na zmianie ilościowej w wyrazie motywującym, a zazwyczaj na ucięciu elementów początku lub końca występujących w podstawie słowotwórczej. W języku IT proces ten jest stosowany do tworzenia tzw. wyrazowców, czyli wyrazów utworzonych przez skrócenie podstaw dwóch

i więcej wyrazów tworzących nazwę i połączenie ich w jeden wyraz. W języku IT najczęstszą grupę stanowią literowce powstałe w języku angielskim, np. *STP* – *shielded twisted pair* o tłumaczeniu *skrętka ekranowana*, ale stosowane w ich oryginalnej formie w polskim języku IT, a także głoskowce, np. *DRAM dynamic random access memory* / *dynamiczna pamięć o dostępie swobodnym* lub rzadkie sylabowce *LISP* – *list processing*. Polski język IT wiernie przejmując wszelkie skróty z języka angielskiego i nie tworzy własnych. Tendencja ta jest podyktowana skrótowością wypowiedzi, internacjonalizacją terminów, a także błyskawicznym postępem technologicznym, gdzie długie nazwy są niezbędne do precyzji informacji, ale zbyt skomplikowane w codziennym użyciu.

4. Gniazdo słotwórcze a derywacja

W celu lepszego zrozumienia procesów derywacyjnych każdego języka zaleca się jego analizę za pomocą wszelkich dostępnych narzędzi. W analizowanym korpusie leksemów możemy wyróżnić obiekty, które funkcjonują jako modele słotwórcze, rejestrujące typ podstawy słotwórczej, formant i znaczenie słotwórcze (Zmarzer 1992:86). Zgodnie z intuicją językową użytkowników od jednego wyrazu języka specjalistycznego zostaje utworzony derywat synchroniczny. W celu lepszego poznania mechanizmów tworzenia i funkcjonowania derywatów należy przyjrzeć się gniazdom słotwórczym wybranych elementów zbioru terminów. Tikhonov (1987:106) definiuje gniazdo słotwórcze jako zbiór wyrazów jednorodnych, uporządkowany relacjami pochodności synchronicznej, a Skarżyński (1999:161) jako „zbiór paradygmatów i łańcuchów słotwórczych o wspólnym wyrazie inicjalnym”. Za centrum gniazda uznaje się leksem synchronicznie niemotywowany słotwórczo (EWJP 1978). W strukturze gniazda najmniejszą jednostką jest para słotwórcza. Pary słotwórcze z kolei w obrębie gniazda tworzą łańcuchy słotwórcze i paradygmaty słotwórcze. Taki układ derywatów zawierających ten sam rdzeń i powiązanych relacjami motywacji może stanowić podstawę opisu słotwórstwa synchronicznego, w którym należałoby wykazać powiązania semantyczno-formalne całych grup wyrazowych, tworzące mniejsze i większe układy.

Gniazdowy opis słotwórczy stanowi kolejny krok ku lepszemu poznaniu systemu słotwórczego polskiego języka IT, ponieważ przedstawia derywaty tego języka w systemie wielostopniowej sieci relacji formalno-semantycznych do odpowiednich leksemów nie pochodnych. Każdy wyraz pochodny może zajmować w gnieździe tylko jedno miejsce w określonej odległości od wyrazu podstawowego. Lokalizacje w układzie poziomym wyznacza stopień komplikacji struktury słotwórczej, rozumianej tu jako suma taktów, czyli operacji derywacyjnych, dzielących analizowaną formację od centrum gniazda. Układ alfabetyczno-kategorialny (według części mowy) warunkuje pozycję danej formacji w układzie pio-

nowym. Tak skomponowane gniazdo stanowi strukturę hierarchiczną, w której ani równorzędność motywacyjna par derywatów, ani równouprawniona zależność od kilku podstaw nie są na tym poziomie istotne (Jadacka 1995). Zadaniem badacza jest scharakteryzować relacje motywacyjne w badanym systemie, aby uwydatnić przede wszystkim zależności typowe, o największym stopniu regularności w celu przedstawienia mechanizmu derywacyjnego badanego języka. Niezmiernie istotny jest wybór takich typów podstaw słowotwórczych, które pozwolą zinterpretować jak najwięcej formacji w celu nadania wykrytym zależnościom statusu relacji systemowych (Jadacka 2003:30). W tym celu derywaty można pogrupować w łańcuchy, posługując się symbolami odpowiednich części mowy i w ten sposób sporządzić schemat ich budowy słowotwórczej. Ustalenie listy możliwych typów modeli wykrywalnych np. w obrębie gniazd z centrum rzeczownikowym daje w rezultacie ilościowy i jakościowy obraz aktywności słowotwórczej. Pogrupowanie wyrazów motywowanych w gniazda słowotwórcze wokół wyrazów centralnych pozwala otrzymać zbiór wyrazów będących rezultatami kilku kolejnych operacji słowotwórczych, w których zastosowano różne techniki derywacyjne. Każdy łańcuch słowotwórczy ukazuje w ten sposób nie tylko wyrazy powstałe na różnych taktach, lecz także środki słowotwórcze uzyskane przy ich tworzeniu.

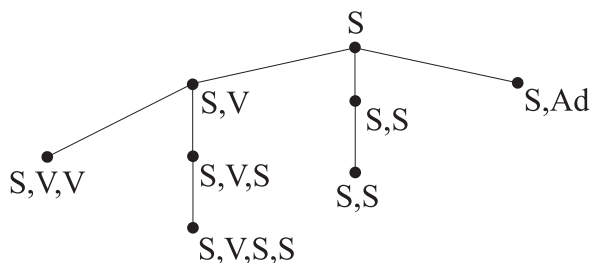
Poniżej przedstawiono dwa przykłady budowy gniazd słowotwórczych leksykonu IT (na podstawie SGSWJP 2001), w których można dostrzec wyrazy centralne, a następnie prześledzić proces derywacji, gdzie poprzez S oznaczono formę rzeczownika, V – formę czasownika, Ad – formę przymiotnika:

PROGRAM ‘uporządkowany ciąg rozkazów wprowadzony do maszyny cyfrowej’	S
[mikro-program]	S,S
[o-program-owanie]	S,S
[pod-program]	S,S
[program-ować]	S,V
[program-ator]	S,V,S
[program-ista]	S,V,S
[program-ist-ka]	S,V,S,S
[program-owanie]	
S,V,S	
[za-program-ować]	
S,V,V	
[prze-program-ować]	S,V,V
SPRZĘT	S
[o-sprzęt]	S
[u-sprzęt-owanie]	S,S
[sprzęt-owy]	S,Ad

(por. SGSWJP 2001)

W przypadku języka angielskiego możemy zbudować analogicznie gniazda, ale należy liczyć się z tym, że nie zawsze będzie to możliwe ze względu na charakter tego języka. Na przykład termin *hardware* nie ma swojego gniazda słowotwórczego.

Należy również pamiętać o ogólnie przyjętych zasadach porządkowania. W przypadku dwóch lub więcej derywatów od jednej podstawy o ich kolejności decyduje przynależność do kategorii części mowy według przyjętej kolejności dla języka polskiego w SGWJP, tj. rzeczowniki, czasowniki, przymiotniki, przysłówki, liczebniki, zaimki. Wśród gniazd opracowanych według przedstawionych w „Słowniku ...” zasad mogą znaleźć się zarówno duże, jak i małe hasła, czyli inaczej mówiąc — wielotaktowe i jednotaktowe. Każde gniazdo można przedstawić za pomocą grafu, co pozwala na syntetyczne ujęcie następujących cech gniazda: struktury morfologiczno-kategorialnej (udziału poszczególnych części mowy), układu modeli słowotwórczych oraz liczby taktów derywacyjnych. Za pomocą kropek przedstawione zostały miejsca, w których kończy się jeden model, a zaczyna kolejny, zawierający poprzedni w swojej strukturze:



Rys.1: Graficzne przedstawienie gniazda słowotwórczego terminu *program*

Kiedy derywat znajduje się najbliżej leksemu hasłowego, oznacza to, że jest przez niego bezpośrednio motywowany. Następnie przedstawione są formacje motywowane przez derywaty pierwszego stopnia (o drugim stopniu pochodności). Poszczególne stopnie pochodności to inaczej takty derywacyjne. Obserwując gniazda poszczególnych części mowy, możemy wysnuć wnioski dotyczące typów derywacji.

Derywację rzeczownikową cechuje różnorodność typów słowotwórczych i znaczeń komunikowanych przez derywaty. Wykładnikami derywacji w polskim języku IT są w większości formanty sufiksalne dodawane do podstaw czasownikowych. W ten sposób powstają nazwy urządzeń i przyrządów. Do najbardziej produktywnych formantów tej grupy należą *-arka*, *-nik*, *-er*, *-acz*, *-(t)or*, np. *wyszukiwarka*, *przeglądarka*, *przełącznik*, *akcelerator*, *sumator*. Grupa derywatów zakończonych na *-er* motywowana jest przez czasowniki obce i charakteryzuje się nieosobowością, np. *assembler*, *sampler*. Formant *-or* derywuje nazwy

narzędzi lub przyrządów od czynności, jakie wykonują, np. *rejestrator*, *regenerator*. Często są też one motywowane przez rzeczowniki zakończone na *-cja*, *-sja*.

Inna widoczna grupa terminów to takie, które mają podstawę słowotwórczą, ale nie mają wykładnika derywacji. Za podstawę motywuującą służą tu czasowniki, np. *rejestr*, *napęd*, *pamięć*. Podobne formy w większości przypadków możemy uznać za oryginalne, ponieważ polski język IT rozszerza w ten sposób znaczenie już istniejących rodzimych struktur.

Bardzo liczny zbiór stanowią też nazwy czynności. Przeważnie tworzone są one od tematu czasu przeszłego czasownika poprzez dodanie formantów *-anie* i *-enie*. Obserwuje się tu nominalizację czasowników o motywacji rzeczownikowej. Formant *-anie* stosowany jest w wyniku przekształceń podstaw słowotwórczych, utworzonych od tematu czasu przeszłego czasownika zakończonego na *-a*, np. *wywoływanie*, *przywracanie*; oraz na *-owa*, np. *logowanie*, *debugowanie*, *justowanie*. Derywacja rzeczowników z formantem *-enie* występuje, kiedy temat czasownika zakończony jest na *-i*, *-y*, np. *połączenie*, *sprawdzenie*.

W wielu przypadkach bardzo łatwo odnotować wyraźny bezpośredni lub pośredni wpływ języka angielskiego na derywację rzeczownikową, np. angielski formant *-ation* umożliwia otrzymanie polskich terminów z formantem *-acja*, np. *configuration* – *konfiguracja*, formant *-ing* umożliwia tworzenie terminów w języku polskim IT z formantem *-nie*, np. *coding* – *kodowanie*. Pozostałe często występujące w języku polskim IT formanty o zapożyczonej podstawie to: *-acja*, *-sja*, *-zja*, np. *defragmentacja*, *emulacja*, *digitalizacja*, *partycja*, *kompresja*, *dyfuzja*.

Warto również zwrócić uwagę na kategorię nazw cech abstrakcyjnych, do której należą terminy powstałe z przekształcenia przymiotnika na rzeczownik najczęściej za pomocą sufiksów angielskiego pochodzenia, np. *-ity*. Formantowi temu odpowiada zazwyczaj sufiks *-ość* w polskim języku IT: *compatibility* – *kompatybilność*, *reliability* – *pewność* (danych). Derywaty przymiotnikowe proste tworzone są zazwyczaj od czasowników i rzeczowników. Przynależność podstaw słowotwórczych do odpowiednich klas stanowi najogólniejsze kryterium podziału tych derywatów. Niektóre przymiotniki dopuszczają dwie interpretacje: odrzeczownikową i odczasownikową, np. *test*, *testować* motywuują przymiotnik *testujący*, a *start*, *startować* – *startowy*. Przymiotniki odczasownikowe nazywają właściwość przedmiotu lub zjawiska polegającą na związku z czynnością, procesem lub stanem, np. *blokujący*, *przyspieszony*.

Kolejna nieustannie pomnażana grupa derywatów przymiotnikowych to formacje odrzeczownikowe. Najczęściej tworzą ją formanty sufiksalne *-owy*: *wsadowy* i *-ny*: *numeryczny*. To, co decyduje o formancie, za pomocą którego derywowany jest przymiotnik, zależy od tego, jaki sufiks występuje w podstawowym rzeczowniku. Prawdopodobnie o aktywności słowotwórczej sufiksu *-owy* decydują pojemność znaczeniowa i skrótość formacji utworzonych za jego pomocą oraz duża łączliwość strukturalna przedrostka. Samogłoska na początku

umożliwia łączenie go z tematami zakończonymi na grupę spółgłoskową (por. Jadacka 1976:100).

Derywaty czasownikowe są motywowane przez czasowniki, np. *załadować*, *odczytać*, przymiotniki, np. *wyglądać*, *zagęszczać*, i rzeczowniki, np. *buforować*, *kalibrować*. Wykładnikami derywacji czasownikowej są przede wszystkim formanty prefiksalne *de-*, *do-*, *na-*, *nad-*, *po-*, *wy-*, *za-* oraz sufiksalne *-ać*, *-ować*, *-ywać*. Do kategorii o najwyższym stopniu częstotliwości w rozpatrywanym kontekście należą:

- formacje nazywające czynności prospektywne z formantem *za-*, oznaczające realizację akcji podstawy z myślą o wykorzystaniu przedmiotu lub zdarzenia w przyszłości, np. *zablokować*, *zapisać*;
- formacje reformatywne oznaczające zmianę cech przedmiotu, które są rezultatem akcji oznaczanej czasownikiem podstawowym. Tworzy je prefiks *prze-*: *przekompilować*, *przenumerować*;
- formacje anulatywne oznaczające likwidację stanu przedmiotu, który jest rezultatem akcji zawartej w podstawie. Znaczenie to wyrażają prefiksy *roz-*, *od-* i *de-*, np. *rozszyfrować*, *odblokować*, *deasemblować*.

Warto też zaznaczyć, że w polskim języku IT występują tzw. akcje pseudozwrotne, wyrażane przez derywaty motywowane głównie przez czasowniki nazywające czynności i działania. Wykładnikiem derywacji jest postfiks *się*, który komunikuje, że akcja podstawowa podlega takiej modyfikacji, iż ograniczona jest do sfery subiektu, np. *bootować się*, *zalogować się*.

Czasowniki odrzeczownikowe to wyrazy utworzone przy użyciu formantu *-ować*, w których bazę czasowników stanowią nazwy przedmiotów, za pomocą których agens działa na obiekt, a efektem działania jest jakaś zmiana, np. *filtrować*, *kompresować*;

- utworzone przy użyciu formantu *-ować* oznaczające wytworzenie niematerialnych przedmiotów, np. czasowniki motywowane przez nazwy wielkości matematycznych i oznaczające obliczanie: *logarytmować*, *sumować*;
- anulatywne z prefiksem *od-*, który likwiduje stan obiektu nazwanego podstawowym rzeczownikiem: *odwzorować*, *odpluskwiać*.

5. Wnioski

Każda wtórna odmiana języka specjalistycznego jest narażona na wpływy języka prymarnego wybranej dziedziny wiedzy. Jednak ze względu na dążenie do oryginalności i zachowanie poprawności językowej priorytetowe wydaje się badanie dotyczące wybranej odmiany języka niż porównywanie kilku odmian ję-

zykowych między sobą. System słotwórczy funkcjonuje poprzez wypełnianie pustych miejsc w obrębie kategorii morfologicznych swoistymi nowotworami. W polskim języku IT nowy wyraz powstaje zawsze w wyniku zabiegu morfologicznego, a nie tylko przekształcenia funkcji jak w języku angielskim. Morfologia słotwórcza ma na celu zbadanie jednostek słownikowych, których struktura formalna i funkcja są zdeterminowane przez ich relacje do innych jednostek słownikowych danego języka. W przypadku polskiego języka IT język angielski występuje jako podstawowe źródło internacjonalizacji narodowych leksykonów informatycznych, a zapożyczona z niego jednostka terminologiczna niejako automatycznie uzyskuje status internacjonalizmu. Z drugiej jednak strony odrębność językowa nie pozwala na całkowite przejście przez narodowy leksykon specjalistyczny prymarnego systemu terminologicznego: „Prototyp terminologiczny z jednej strony występuje jako swoisty regulator na drodze żywiołowego przejścia przez narodowy leksykon terminów z języka prymarnego, z drugiej zaś strony ukierunkowuje działalność indywidualną w zakresie tworzenia nowych jednostek terminologicznych” (STP 2005).

Polski język IT jest otwarty na wzbogacanie własnych zasobów drogą pożyczek z języka angielskiego. Niewątpliwie jest to też rezultat dążenia do unifikacji terminologii. Zanim to jednak nastąpi realnie, zapożyczone wyrazy muszą zostać przyjęte do systemu gramatycznego języka IT, czyli zyskać wskaźniki przynależności do określonych klas gramatycznych i paradygmatów odmiany. W języku polskim przejawia się to w otrzymywaniu przez nie końcówek fleksyjnych zgodnych ze wskaźnikami morfologicznymi polszczyzny. Wpływy angielskie wywołane zostały szybkim rozwojem techniki i rozpowszechnianiem się nowych narzędzi i aplikacji. Wiele z nowych wyrazów nie wchodzi do języka ogólnego, funkcjonując tylko w zamkniętych kręgach specjalistów, ale niektóre przedostają się do języka potocznego, np. *folder*, *przeglądarka*, *cookies*. Najwłaściwsze wydaje się wprowadzanie do języka polskiego wyrazów zgodnie z rodzimą fleksją. W tym zakresie polski język IT radzi sobie dobrze, dostosowując się do bieżących potrzeb jego użytkowników. Jednocześnie należy zaznaczyć, że sposoby derywacji polskiego języka IT powinny być dalej badane i precyzowane, także pod względem ilościowym. Ich analiza pozwoli na proponowanie możliwie jak najskuteczniejszych działań językowych podejmowanych przez polskich informatyków, a także terminologów i tłumaczy.

Bibliografia

- Angielsko-polski słownik informatyczny, 2004, Warszawa.
BAŃKO Mirosław, 2002, Wykłady z polskiej fleksji, Warszawa.
BAK Paweł, 2006, Popularny angielsko-niemiecko-polski słownik terminów komputerowych i informatycznych, Warszawa.

- BOJAR Bożenna (red.), 1993, Słownik encyklopedyczny terminologii języków i systemów informacyjno-wyszukiwaczy, Warszawa.
- BOROWSKA Anna Paulina, 2005, Obcość przekładu informatycznego, w: Lewandowski J./Kornacka M. (red.), Języki Specjalistyczne: Teksty specjalistyczne w kontekstach zawodowych i tłumaczeniach, Warszawa, s. 165 – 171.
- BOROWSKA Anna Paulina, 2009, W poszukiwaniu modelu analizy konfrontatywnej systemów syntaktycznych angielskiego i polskiego języka informatyki, w: Lukszyn J. (red.), W kręgu problematyki technolektalnej, Warszawa, s. 51 – 71.
- CUDAK Romuald / TAMBOR Jolanta, 1995, O języku ‘komputerowców’, w: Język Polski LXXV, z. 3, s. 197 – 204.
- Encyklopedia wiedzy o języku polskim [EWJP], 1978, Wrocław.
- GRZEGORCZYKOWA Renata / PUZYNINA Jadwiga, 1999, Problemy ogólne słowotwórstwa, Rzeczownik, w: Grzegorzyczkowa R./Laskowski R./Wróbel H. (red.), Morfologia, Warszawa.
- JADACKA Hanna, 1976, Termin techniczny – pojęcie, budowa, poprawność, Warszawa.
- JADACKA Hanna, 1995, Rzeczownik polski jako baza derywacyjna – opis gniazdowy, Warszawa.
- JADACKA Hanna, 2003, Zagadnienie motywacji słowotwórczej w opisie gniazdowym, w: Skarżyński M. (red.), Słowotwórstwo gniazdowe – historia, metoda, zastosowania, Kraków.
- LASKOWSKI Roman, 1999, Zagadnienia ogólne morfologii, w: Grzegorzyczkowa R./Laskowski R./Wróbel H. (red.), Morfologia, Warszawa.
- MILEWSKI Tadeusz, 2004, Językoznawstwo, Warszawa.
- SKARŻYŃSKI Mirosław, 1999, Słowotwórcze gniazda odczasownikowe, w: Skarżyński M. (red.), Słowotwórstwo gniazdowe – historia, metoda, zastosowania, Kraków.
- Słownik gniazd słowotwórczych współczesnego języka ogólnopolskiego [SGSWJP], 2001, Jadacka H. (red.), Kraków.
- Słownik terminologii przedmiotowej [STP], Lukszyn J. (red.), 2005, Warszawa.
- SZULC Aleksander, 1997, Słownik dydaktyki języków obcych, Warszawa.
- TABAKOWSKA Elżbieta, 2001, Kognitywne podstawy języka i językoznawstwa, Kraków.
- TIKHONOV Aleksander Nikołajewicz, 1987, Slovoobrazowatelnyj slovar russkogo jazyka, Moskwa.
- ZMARZER Wanda, 1992, Podstawy analizy konfrontatywnej języków pokrewnych, Warszawa.