

Paweł Ziemba

Państwowa Wyższa Szkoła Zawodowa w Gorzowie Wielkopolskim
e-mail: pziemba@pwsz.pl

Jarosław Jankowski

Zachodniopomorski Uniwersytet Technologiczny w Szczecinie
e-mail: jjankowski@wi.zut.edu.pl

Waldemar Wolski

Uniwersytet Szczeciński
e-mail: wwolski@uoo.univ.szczecin.pl

DOBÓR JEZYKA REPREZENTACJI WIEDZY W ONTOLOGIACH DZIEDZINOWYCH

SELECTION OF KNOWLEDGE REPRESENTATION LANGUAGE IN THE DOMAIN ONTOLOGIES

DOI: 10.15611/ie.2015.1.07

Streszczenie: W artykule podjęto problem doboru języka reprezentacji wiedzy dla ontologii dziedzinowych. Przedstawiono w nim zastosowania ontologii jako narzędzia reprezentacji wiedzy oraz analizę formalizmów logicznych, takich jak: ramy, programowanie logiczne, logika opisowa, logika I rzędu i logika wspólna. Następnie analizie poddano szereg, klasycznych i opartych na znacznikach, języków reprezentacji wiedzy: Ontolingua, LOOM, OCML, FLogic, SHOE, RDF(S), OWL oraz OWL 2. Na podstawie analizy literatury usystematyzowano również związki i zależności pomiędzy poszczególnymi wersjami i profilami języka OWL. Artykuł kończy następująca konkluzja: język OWL 2 DL jest najbardziej ekspresywnym językiem zachowującym rozstrzygalność, w związku z czym charakteryzuje się on najwyższą stosowalnością w budowie ontologii dziedzinowych, umożliwiających wnioskowanie.

Słowa kluczowe: reprezentacja wiedzy, OWL, ontologia, baza wiedzy.

Summary: The article discusses the problem of knowledge representation language selection for domain ontologies. In the article the use of ontology as a tool of knowledge representation was presented and the analysis of logical formalisms such as frames, logic programs, description logic, first-order logic and common logic was carried out. Then a number of classic and markup based knowledge representation languages were analysed: Ontolingua, LOOM, OCML, FLogic, SHOE, RDF(S), OWL, OWL2. Based on the analysis of literature relationships and dependencies between versions and profiles of the OWL language were systematised. The article ends with the conclusion, according to which OWL 2 DL language is the most expressive language of retaining decidability, and therefore it is characterized by the highest applicability in the construction of domain ontologies allowing inference.

Keywords: knowledge representation, OWL, ontology, knowledge base.

1. Wstęp

W związku z dynamicznym rozwojem technologii informacyjnych i komunikacyjnych w drugiej połowie XX wieku zmianom uległ sposób funkcjonowania organizacji. Na skutek tego ich dotychczasowa struktura stała się niewystarczająca i uległa modyfikacjom prowadzącym w kierunku organizacji informacyjnej czy też opartej na wiedzy. Cechą takich organizacji jest położenie nacisku na: ciągłe uczenie się, pozyskiwanie wiedzy i jej efektywne wykorzystanie. Dzięki temu mają one zdolność adaptacji do szybko zmieniającego się otoczenia i konkurencji. Istotny dla takich organizacji jest wykorzystywany w nich system informacyjny, który powinien wspomagać zarządzanie wiedzą i umożliwiać podejmowanie skuteczniejszych decyzji [Brdulak 2004].

Sama wiedza jest podstawą sztucznej inteligencji, a więc kwestie związane z jej reprezentacją, zrozumieniem, projektowaniem i implementacją są bardzo istotne dla wszelkich systemów inteligentnych [Chaudhary i in. 2012]. O reprezentacji wiedzy mówi się ogólnie w kontekście jej przedstawiania w taki sposób, który umożliwia jej automatyczne przetwarzanie przez komputer [Hartley 1985]. W sztucznej inteligencji przez reprezentację wiedzy rozumie się natomiast połączenie struktur danych oraz procedur interpretacyjnych operujących na tych danych [Rao, Jain 1988]. Procedury te, właściwie użyte, powinny prowadzić do inteligentnego zachowania systemu. Oznacza to, że system taki powinien potrafić przeprowadzać wnioskowanie w sposób zbliżony do człowieka [Abraham 2003]. W związku ze wskazanymi cechami systemów inteligentnych, takich jak systemy ekspertowe i inteligentni agenci [Truemper 2004], od kilkudziesięciu lat widoczny jest ich dynamiczny rozwój [Wong, Monaco 1995; Shiau 2011]. Systemy ekspertowe, określane często jako systemy oparte na wiedzy [Tripathi 2011], do sprawnego działania wymagają odpowiedniej reprezentacji wiedzy [Rao, Jain 1988]. Wiedza ta coraz częściej jest w nich reprezentowana w postaci ontologii, których istotną przewagą nad systemami regułowymi jest większa ekspresywność [Czarnecki, Sitek 2013], zachowanie monotoniczności, wnioskowanie terminologiczne oraz ułatwione powtórne wykorzystanie [Waloszek 2010]. Same ontologie mogą być budowane przy użyciu różnych formalizmów i języków. W związku z tym ważnym zagadnieniem przy ich projektowaniu staje się właściwy dobór formalizmu oraz konkretnego języka reprezentacji wiedzy.

Celem podjętych i opisanych w artykule badań jest dobór języka reprezentacji wiedzy, charakteryzującego się wysoką stosowalnością w budowie ontologii dziedzinowych. W artykule przedstawiono przesłanki przemawiające za wykorzystaniem ontologii jako formalizmu reprezentacji wiedzy, wskazano cechy poszczególnych języków reprezentacji oraz przeprowadzono ich analizę. Artykuł kończy się wnioskami z badań określającymi najbardziej użyteczny język reprezentacji wiedzy dziedzinowej.

2. Przegląd literatury

W literaturze z dziedziny informatyki pojęcie ontologii pojawiło się po raz pierwszy w 1967 r. w pracy S.H. Mealy'ego w kontekście modelowania danych [Labu-

zek 2003]. Obecnie najczęściej przytaczana w literaturze informatycznej definicja ontologii mówi, że jest ona „formalną specyfikacją konceptualizacji”, która może uchwycić wiedzę dziedzinową, nadającą się do wielokrotnego użycia [Gruber 1993]. Podobna definicja mówi o tym, że w informatyce ontologia jest traktowana jako struktura danych i narzędzie do reprezentacji danych, pozwalające współdzielić i ponownie wykorzystywać wiedzę w systemach sztucznej inteligencji korzystających ze wspólnego słownictwa [Guzman-Arenas, Cuevas 2010]. Natomiast jedna z najbardziej rozbudowanych definicji stwierdza, że ontologia jest jednoznaczną specyfikacją konceptualizacji, w czym zawierają się cztery znaczenia: (1) konceptualizacja, która jest abstrakcyjnym modelem wywiedzionym z realnego świata do reprezentowania pewnych pojęć; (2) jednoznaczność, co oznacza, że ontologia ma jednoznacznie zdefiniowane koncepty, relacje, właściwości, funkcje, ograniczenia i aksjomaty; (3) formalizacja, która oznacza, że ontologia może być rozumiana i wykonywana przez maszynę; (4) współdzielenie, oznaczające, że ontologia jest odzwierciedleniem powszechnie uznanej wiedzy dziedzinowej i dostarcza ona usługi semantyczne [Zhu, Yang, Chen 2009]. Każda z tych definicji odnosi się do faktu, że za pomocą ontologii można przedstawić wiedzę dziedzinową, a następnie tę wiedzę współdzielić i wielokrotnie wykorzystywać. Najczęściej ontologie w informatyce są wykorzystywane do zapewnienia szeroko pojętego dostępu do zasobów wiedzy. Ogólnie rzecz ujmując, ich zastosowania obejmują:

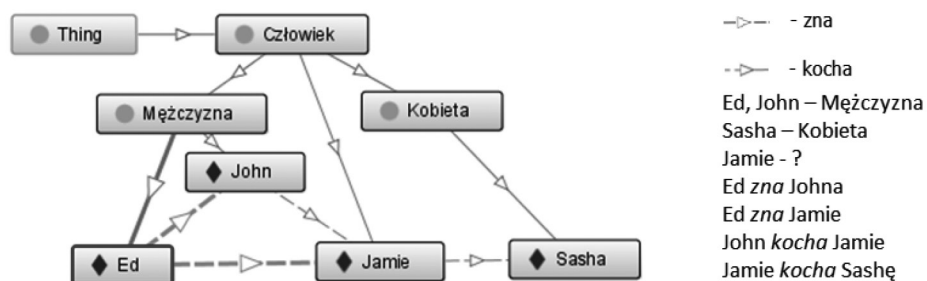
- komunikację – między realizowanymi systemami obliczeniowymi, między ludźmi oraz między systemami i ludźmi,
- wnioskowanie obliczeniowe – do wewnętrznej reprezentacji planów i planowania informacji oraz w celu dokonywania analizy: wewnętrznej struktury, algorytmów oraz wejść i wyjść realizowanych systemów,
- ponowne użycie i organizację wiedzy – dla strukturalizowanych lub organizowanych bibliotek i repozytoriów informacji dziedzinowych [Gruninger, Lee 2002].

Zarówno w wielu definicjach ontologii, jak i w charakterystykach ich zastosowań często przewija się sformułowanie „wiedza”. Istotnie ontologie są stosowane do reprezentacji wiedzy, a więc również do tworzenia baz wiedzy. Takie wykorzystanie ontologii pojawia się chociażby w pracy [Villanueva-Rosales, Dumontier 2008], gdzie przy użyciu ontologii utworzono bazę wiedzy biomedycznej. Możliwość wykorzystania ontologii jako baz wiedzy systemów ekspertowych potwierdzają m.in. prace: [Saa i in. 2012; Shue, Chen, Shiue 2009], w których ontologie zaimplementowano w bazach wiedzy funkcjonujących systemów, tj.: systemu ekspertowego do badania wskaźników finansowych przedsiębiorstwa [Shue, Chen, Shiue 2009] i systemu wspomagania decyzji podejmowanych przy budowie infrastruktury kolejowej [Saa i in. 2012].

Zastosowanie ontologii do tworzenia baz wiedzy wynika z podstawowej różnicy między bazami wiedzy a bazami danych, ze względu na którą do tworzenia baz wiedzy nie można zastosować schematów baz danych. Mianowicie: wydobywanie informacji z baz danych jest działaniem algorytmicznym, podczas gdy to samo

działanie, realizowane z wykorzystaniem bazy wiedzy, ma charakter wnioskowania logicznego. Ponadto najczęściej bazy danych są niemonotoniczne, a bazy wiedzy charakteryzują się monotonicznością. Oznacza to, że nowe dane dodane do bazy danych mogą zmienić wnioski wysnuwane przez bazę (odpowieź bazy na określone zapytanie). Z kolei nowa wiedza dodana do bazy wiedzy nie może zmienić uzyskiwanych z niej wniosków [Goczyła 2011]. Ponadto model ontologiczny może pozwalać na pełne i rozstrzygalne wnioskowanie w bazach wiedzy, nawet gdy nie dysponuje kompletną wiedzą. Cechę tę przedstawia przykład problemu hemafrodyty zawarty na rys. 1. Jest to zmodyfikowany przykład z rozprawy doktorskiej [Waloszek 2007]. Mając przedstawioną na rys. 1 bazę wiedzy o osobach, ich płci, a także znajomościach i sympatiach między nimi, można zadać bazie pytanie: „Czy jest człowiek znający mężczyznę, który kocha jakąś kobietę?”. Klasyczna baza danych nie ma takiej wiedzy, więc odpowiedzią byłby pusty rekord. Z kolei ontologiczna baza wiedzy przeprowadzi wnioskowanie, zgodnie z którym odpowiedzią będzie „Ed”, gdyż:

- jeżeli Jamie jest kobietą: Ed zna Johna (mężczyznę); John kocha Jamie,
- jeżeli Jamie jest mężczyzną: Ed zna Jamie; Jamie kocha Sashę (kobietę).

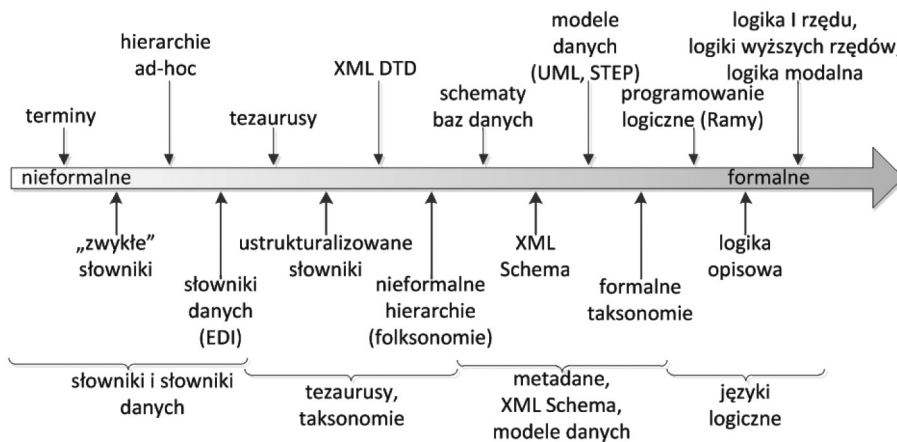


Rys. 1. Problem hemafrodyty jako przykład wnioskowania w ontologii przy niepełnej wiedzy

Źródło: opracowanie własne.

Rozważania na temat języków, za pomocą których ontologie mogą być reprezentowane, należy rozpocząć od pewnej dygresji. Mianowicie: ontologie mogą mieć postać formalną lub nieformalną. Do ontologii nieformalnych można zaliczyć wszelkie słowniki, tezaury i taksonomie. Ontologie formalne natomiast są to ontologie charakteryzujące się jasną semantyką języka używanego do definiowania konceptów, jasnymi motywacjami rozróżniania konceptów i rygorystycznymi zasadami dotyczącymi sposobu definiowania konceptów i relacji [Roussey i in. 2011]. Do ontologii formalnych zaliczane są więc wszelkie ontologie bazujące na: programowaniu logicznym (*Logic Programming*), ramach (*Frames*), logice opisowej (*Description Logic*), logice modalnej (*Modal Logic*), logice pierwszego rzędu (rachunku predykatów I rzędu) (*First-order Logic*) i logikach wyższych rzędów (*High-order Logic*).

W literaturze z zakresu ontologii nie jest natomiast jednoznacznie określone, do jakiej grupy ontologii należy zaliczyć ontologie oparte na modelach danych, schematach bazodanowych i standardzie XML Schema. Przykładowo [Goczyła 2011] zalicza tego typu ontologie do grupy ontologii formalnych (jednak z zastrzeżeniem, że nie spełniają one założenia o świecie otwartym, tzn. nie są monotoniczne). Z kolei autorzy pracy [Roussey i in. 2011] zaliczają je do grupy ontologii programowych (*Software Ontologies*), przy czym grupa ta jest odrębna od ontologii formalnych. Podobnie Uschold i Gruninger [2004] nie określają tego typu ontologii jako formalne. Natomiast Guarino i in. [Guarino, Oberle, Staab 2009] stwierdzają, że trudno jednoznacznie określić, gdzie przebiega granica między ontologiami formalnymi i nieformalnymi. Problem przynależności różnego rodzaju ontologii do kategorii ontologii nieformalnych i formalnych został przedstawiony na rys. 2. Za ontologie formalne bez wątplenia można uznać ontologie oparte na językach logicznych, umiejscowione na rys. 2 po prawej stronie.



Rys. 2. Problem podziału ontologii na formalne i nieformalne

Źródło: opracowanie własne na podstawie [Guarino, Oberle, Staab 2009; Uschold, Gruninger 2004].

Formalizmy, takie jak: ramy, logika opisowa i logika pierwszego rzędu, stanowią podstawę wielu języków ontologii. Obok nich można wymienić również wspólną logikę (*Common Logic*), która jest dopiero opracowywana, ale została już ustandaryzowana przez ISO [ISO/IEC 24707:2007(E) 2007]. Formalizmy te mają pewne cechy wspólne, ale mają też wiele cech je różnicujących.

Paradygmat ram jest ściśle powiązany z sieciami semantycznymi, gdyż są to dwa warianty notacji do zapisu takich samych konstrukcji [Cumings 2005; Hirst 1992]. Chociaż zaznaczyć należy, że ramy są bogatsze językowo niż sieci semantyczne, gdyż pozwalają np. na dynamiczne dodawanie wartości dla klatek poprzez rozszerzenia proceduralne [Chaudhary 2012]. Paradygmat ram spełnia założenia

o unikalności nazw, co oznacza, że dwie ramy mające inne nazwy są od siebie różne (chyba że jawnie zdefiniowano między nimi powiązanie mówiące o tym, że są to te same ramy). Ramy, poprzez wykorzystanie domniemanej negacji, spełniają również założenie o świecie zamkniętym. Ontologie oparte na ramach uwzględniają tylko jeden minimalny model (interpretację) świata. Ponadto z wykorzystaniem ram można definiować tylko warunki konieczne, a mechanizm wnioskujący działa na zasadzie sprawdzania ograniczeń [Wang i in. 2006]. Poza rozszerzeniami proceduralnymi i domniemaną negacją ramy wykorzystują również pewien fragment logiki I rzędu [Wu, Chen 2008], pokrywający się z częścią logiki opisowej [Nebel 2001]. Ponadto ramy, ze względu na zastosowanie metaklas, mogą się charakteryzować nierozstrzygalnością wnioskowania [Goczyła 2011].

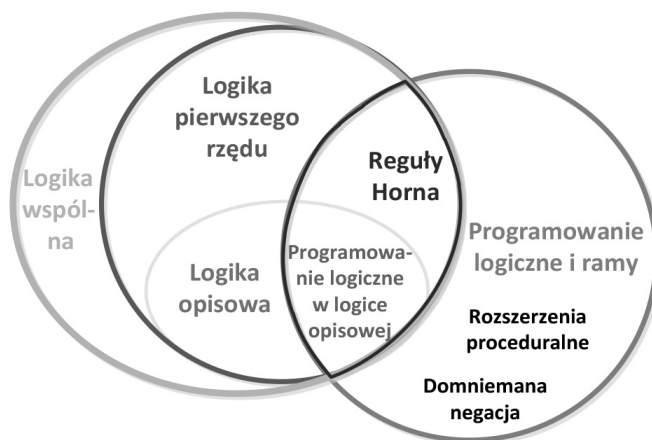
Porównując charakterystykę programowania logicznego zawartą w [Guarino, Oberle, Staab 2009] („programowanie logiczne uwzględnia tylko część modeli podczas oceny wiązań semantycznych” i „jest ono niemonotoniczne”) z charakterystyką ram i uwzględniając schematy opisujące stopień formalizmu ontologii zawarte w pracach [Guarino, Oberle, Staab 2009; Uschold, Gruninger 2004] (gdzie ramy i programowanie logiczne zostały użyte zamiennie), można wnioskować, że pojęcie ram jest bliskie rozumieniu pojęcia programowania logicznego. Potwierdza to analiza schematów zawartych w publikacji [Grosz, i in. 2003; Antoniou, van Harmelen 2009], w której programowaniu logicznemu przypisuje się domniemaną negację i rozszerzenia proceduralne oraz częściową przynależność do logiki I rzędu, w tym do logiki opisowej.

Logika opisowa pozwala reprezentować wiedzę w ustrukturalizowanej i sformalizowanej formie. Dopuszcza ona m.in. konstrukcje: przecięcia, sumy, klasycznej negacji, ograniczenia (egzystencjalne, uniwersalne i liczności) subsumpcji, role itd. Zaznaczyć należy, że istnieje wiele dialektów logiki opisowej charakteryzujących się różnym zakresem ekspresywności [Baader, Horrocks, Sattler 2009]. Logika opisowa, w przeciwieństwie do ram, jest monotoniczna i spełnia paradygmat o otwartości świata [Knorr 2011]. Jest ona również bardziej sformalizowana i ekspresywna od ram, ale jednocześnie charakteryzuje się mniejszą ekspresywnością od logiki I rzędu. Jest ona podzbiorem logiki I rzędu, bardzo bliskim logice modalnej, a większość dialektów logiki opisowej charakteryzuje się rozstrzygalnością [Nebel 2001; de Bruijn 2007].

Logika I rzędu obejmuje m.in. logikę opisową, modalną i reguły Horna. Logika ta, podobnie jak logika opisowa, jest monotoniczna i spełnia paradygmat o otwartości świata. Jednak logika I rzędu jest znacznie bardziej ekspresywna od ram i logiki opisowej. Wobec tego trudno w niej przeprowadzić wnioskowanie, a wiele przedstawionych w niej problemów jest nierozstrzygalnych [de Bruijn 2007].

Logika wspólna jest tzw. frameworkiem rodziny logik I rzędu. Dostarcza ona pełnej składni i semantyki logik I rzędu, wobec czego każda konwencjonalna logika I rzędu może być bezpośrednio przełożona na logikę wspólną, bez utraty informacji lub zmiany znaczenia [ISO/IEC 24707:2007(E) 2007]. Język wspólnej logiki jest

monotoniczny. Umożliwia on też uzyskanie trzech rodzajów odpowiedzi mechanizmu wnioskującego (prawda, fałsz, wartość nieznana) [Koide, Takeda 2011]. Jednak, ze względu na to, że język wspólnej logiki jest bardzo rozbudowany, bardzo trudne w nim jest przeprowadzenie automatycznego wnioskowania [Kutsia, Marin 2012]. Ponadto obecnie nie jest wyłoniony dialekt implementacji wspólnej logiki [Koide, Takeda 2011]. Wzajemne relacje pomiędzy omówionymi formalizmami, stanowiącymi podstawę istniejących języków ontologii, zawiera rys. 3.



Rys. 3. Relacje pomiędzy formalizmami będącymi podstawą języków ontologii

Źródło: opracowanie własne na podstawie [Goczyła 2011; Grosz i in. 2003; Antoniou, van Harmelen 2009; ISO/IEC 24707:2007(E) 2007].

3. Metodyka badań

Problem doboru języka reprezentacji wiedzy potraktowano w niniejszej pracy jako problem decyzyjny, którego rozwiązanie wymaga:

- 1) określenia celu decyzji,
- 2) zdefiniowania wariantów decyzyjnych,
- 3) analizy wariantów pod względem kryteriów decyzyjnych,
- 4) opracowania rekomendacji [Roy 1996].

Celem decyzji był wybór języka charakteryzującego się wysoką stosowalnością w budowie ontologii dziedzinowych. Precyzując cel, należy zaznaczyć, że język taki powinien oferować możliwie wysoką ekspresywność, dzięki czemu pozwalałby on na tworzenie konstrukcji, które precyzyjnie odzwierciedlają daną dziedzinę wiedzy, reprezentowaną w ontologii. Bardzo istotnym wymogiem jest także rozstrzygalność języka, a tym samym zapewnienie rozstrzygalności problemów reprezentowanych w ontologii.

W kroku drugim należało zdefiniować warianty decyzyjne, którymi są poszczególne języki reprezentacji ontologii. W tym celu dokonano analizy wielu języków reprezentacji wiedzy stosowanych do konstruowania ontologii dziedzinowych. W trakcie tej analizy oparto się na publikacjach naukowych, w których scharakteryzowano część języków. Dane zaczerpnięte z prac naukowych uzupełniono także o informacje zawarte w materiałach i dokumentach standaryzacyjnych, opisujących nowe języki wchodzące w skład standardu OWL 2. Istotnym punktem badań jest uporządkowanie wiedzy na temat poszczególnych dialektów języków OWL i OWL 2 oraz określenie relacji i zależności między ich dialektami.

W kolejnym etapie badań, na podstawie dokonanej analizy, zdefiniowano kryteria oceny oraz stopień ich spełnienia przez poszczególne języki. Określono tutaj stopień ich ekspresywności pod względem:

- definiowania conceptów,
- budowy taksonomii conceptów,
- konstruowania relacji,
- definiowania funkcji,
- możliwości wykorzystania reguł,
- definiowania aksjomatów formalnych,
- tworzenia instancji conceptów.

Dzięki przeprowadzonej analizie dla każdego z języków określono też stopień rozstrzygalności problemów przedstawionych w danym języku oraz zbadano ich monotoniczność. Informacje te posłużyły do przedstawienia analizy porównawczej poszczególnych języków reprezentacji ontologii.

Ostatnim etapem badań był przegląd poszczególnych wariantów (języków) i dokonanie rekomendacji ze względu na stopień spełnienia celu, którym był wybór języka o możliwie wysokiej ekspresywności, zachowującego jednocześnie rozstrzygalność wnioskowania.

4. Wyniki badań

Poszczególne języki oparte są na różnych, omówionych wcześniej formalizmach i ich kombinacjach. Wobec tego różnią się stopniem formalizacji, ekspresywnością i złożonością wnioskowania. Języki te można podzielić na dwie grupy: języki tradycyjne i języki znaczników. Do języków tradycyjnych zaliczają się m.in.: Ontolingua, LOOM, OCML, FLogic. Z kolei językami opartymi na znacznikach są m.in.: SHOE, RDF(S), OWL i OWL 2.

Język Ontolingua opiera się na logice I rzędu, ale został dodatkowo rozszerzony o ramy. Umożliwia on tworzenie takich elementów, jak: concepty, taksonomie conceptów, n -arne relacje, funkcje, aksjomaty, instancje i procedury. Ze względu na wysoką ekspresywność trudno skonstruować dla niego mechanizm wnioskujący [Gomez-Perez i in. 2004; Corcho, Fernandez-Lopez, Gomez-Perez 2006].

LOOM został opracowany jako język umożliwiający konstruowanie systemów ekspertowych, baz wiedzy i innych aplikacji inteligentnych. Jest on ustanowiony na logice opisowej, a składa się z dwóch podjęzyków, tj.: języka opisowego, którego odpowiednikiem w logice opisowej jest TBox, i języka powiązań, któremu w logice opisowej odpowiada ABox. LOOM jest językiem monotonicznym. Umożliwia on automatyczną klasyfikację konceptów, a zawiera takie elementy, jak: koncepty, taksonomie konceptów, n -arne relacje, funkcje, aksjomaty i reguły produkcyjne. Dodatkowo wspiera obliczenia predykatów i programowanie proceduralne. Wadą języka LOOM jest to, że jest on bardzo ekspresywny, w związku z czym jego mechanizm wnioskujący nie umożliwia pełnego wnioskowania (LOOM jest nierozstrzygalny) [Gomez-Perez, Fernandez-Lopez, Corcho 2004].

OCML (*Operational Conceptual Modeling Language*) jest oparty na języku Ontolingua (wykorzystuje więc logikę I rzędu i ramy), a został zbudowany w celu opracowywania wykonywalnych ontologii i modeli rozwiązywania problemów. Pozwala definiować klasy, relacje, funkcje, aksjomaty i instancje oraz definiować reguły produkcyjne i dedukcyjne oraz rozszerzenia proceduralne. Zawiera 12 podstawowych ontologii definiujących: typy danych, definicje prymitywów będących elementami ontologii (np. relacje), prymitywy używane do budowy metod rozwiązywania problemów (*Problem Solving Methods*) oraz opis języka OCML. Ontologie te są wykorzystywane przez system dowodzenia twierdzeń. Zaznaczyć należy jednak, że system dowodzenia twierdzeń języka OCML, podobnie jak mechanizmy wnioskujące innych języków opartych na logice I rzędu i ramach, nie zapewnia pełnego wnioskowania [Gomez-Perez, Fernandez-Lopez, Corcho 2004; Corcho, Fernandez-Lopez, Gomez-Perez 2006].

FLogic (*Frame Logic*) opiera się na paradygmacie programowania obiektowego, ramach i logice I rzędu. Początkowo zastosowaniem języka FLogic były dedukcyjne i obiektowe bazy danych, a dopiero potem został on zaadaptowany do zastosowania go w ontologiach. FLogic umożliwia definiowanie konceptów, taksonomii konceptów, relacji binarnych, funkcji, instancji, aksjomatów i reguł dedukcyjnych. Mechanizmy wnioskujące, przeznaczone dla tego języka, są w stanie zapewnić pełne wnioskowanie tylko dla jego monotonicznej części [Gomez-Perez, Fernandez-Lopez, Corcho 2004; Corcho, Fernandez-Lopez, Gomez-Perez 2006].

SHOE (*Simple HTML Ontology Extension*) miał na celu wprowadzenie wiedzy semantycznej do dokumentów Web. Został on oparty na ramach oraz regułach. Ponadto może korzystać ze składni HTML lub XML. Język SHOE zawiera reprezentację konceptów, ich taksonomii, n -arnych relacji, instancji i reguł dedukcyjnych. Mechanizm wnioskujący do pozyskania nowej wiedzy wykorzystuje właśnie reguły dedukcyjne [Gomez-Perez, Fernandez-Lopez, Corcho 2004; Corcho, Fernandez-Lopez, Gomez-Perez 2006].

RDF(S) (*Resource Description Framework + Schema*) jest połączeniem opartego na sieciach semantycznych języka RDF i wykorzystującego ramy języka RDFS. Zasadniczo język RDF(S) (podobnie jak jego przodkowie, tj. RDF i RDFS) nie po-

winien być rozumiany ściśle jako język ontologii, ale raczej jako ogólny język opisu danych w sieci Internet. W związku z tym język ten jest dużo mniej ekspresywny od niemal wszystkich wcześniej scharakteryzowanych. Dostarcza on reprezentacji konceptów, taksonomii konceptów i relacji binarnych oraz korzysta ze składni XML. Działanie mechanizmów wnioskujących dla tego języka głównie sprowadza się do sprawdzania ograniczeń [Gomez-Perez, Fernandez-Lopez, Corcho 2004; Corcho, Fernandez-Lopez, Gomez-Perez 2006].

OWL (*Web Ontology Language*) jest przeznaczony do publikowania i współdzielenia ontologii w sieci Internet oraz wykorzystuje składnię XML i RDF. OWL jest językiem monotonicznym [Gomez-Perez, Fernandez-Lopez, Corcho 2004]. Istnieją trzy odmiany języka OWL w wersji 1 różniące się od siebie ekspresywnością. Są to, oparte na logice opisowej, OWL Lite i OWL DL oraz, wykraczający poza logikę opisową, OWL Full.

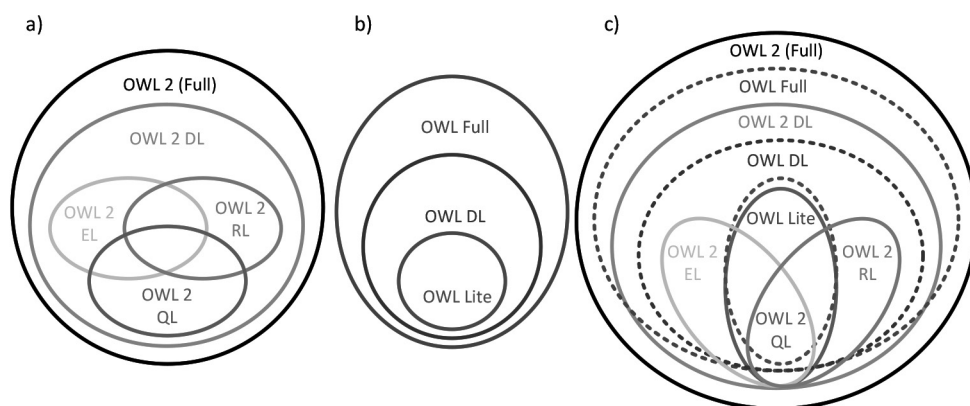
OWL Lite umożliwia tworzenie hierarchii i nakładanie prostych ograniczeń. Jest on zgodny z dialektem SHIF(D). OWL DL dostarcza wszystkich prymitywów języka OWL z wyłączeniem metamodelowania. Wykorzystuje on dialekt SHOIN(D). OWL Full jest najbardziej ekspresywną odmianą języka OWL 1, umożliwiającą m.in. metamodelowanie. OWL Full jest w pełni kompatybilny z językiem RDF, czyli każdy dokument RDF jest też dokumentem OWL Full i każdy dokument OWL Full jest dokumentem RDF. Odmiany OWL Lite i OWL DL są w pełni rozstrzygalne i umożliwiają pełne wnioskowanie, podczas gdy OWL Full może być nierozstrzygalny, a wnioskowanie z tego języka może być mało przewidywalne [Roussey i in. 2011; Grau i in. 2008; Antoniou, van Harmelen 2009].

Język OWL 2, w porównaniu z OWL w wersji 1, umożliwił tworzenie nowych konstrukcji, takich jak: klucze, nowe typy danych, kwalifikowane przez typ danych ograniczenia liczości, nowe właściwości (m.in. asymetryczność i zwrotność oraz rozłączność właściwości). Szczegółowo różnice między wersją 1 i 2 języka OWL zostały przedstawione w [Grau i in. 2008]. Poza pełnym językiem OWL 2 (który jest nierozstrzygalny) wyróżnia się także profile językowe: OWL 2 EL, OWL 2 QL i OWL 2 RL.

OWL 2 EL jest oparty na dialekcie EL++, a jego przeznaczeniem jest tworzenie dużych ontologii i klasyfikacja, czyli określanie relacji między klasami (konceptami) zawartymi w ontologii. Nie dopuszcza on stosowania: kwantyfikatora ogólnego, ograniczeń liczości, sumy, negacji, niektórych właściwości i typów danych. OWL 2 QL jest oparty na podzbiorze logiki opisowej DL-Lite, który to podzbiór jest z kolei bliski odmianie OWL Lite. Jest on przeznaczony do zastosowań wymagających klasycznych baz danych i ontologii OWL, a jego konstrukcje mogą być zapisane również za pomocą diagramów UML i diagramów związków encji. Nie dopuszcza on konstrukcji: kwantyfikatorów ogólnego i (w określonych sytuacjach) egzystencjalnego, ograniczeń pojedynczą wartością dla indywidualności i literalów, kluczy, lokalnej zwrotności, sumy, klas wyliczanych, ograniczeń liczości, niektórych właściwości i typów danych. OWL 2 RL przeznaczony jest dla aplikacji wymagających

skalowalnego wnioskowania bez zbyt dużej utraty ekspresywności języka. Opiera się on na podzbiorze będącym przecięciem logiki opisowej i reguł Horna, oznaczonym na rys. 3 jako programowanie logiczne w logice opisowej. Nie są w nim dopuszczalne konstrukcje, w których po prawej stronie relacji podrzędności klas występuje wyrażenie, nie dopuszcza również sumy klas rozłącznych i niektórych typów danych. Każdy z tych profili bazuje na podzbiorze logiki opisowej, więc również każdy z nich jest rozstrzygalny i umożliwia pełne wnioskowanie. Istnieje również nieformalny profil OWL 2 DL, ograniczający język OWL 2 do zbioru logiki opisowej. Wariant OWL 2 DL jest zgodny z dialektem SROIQ(D) [W3C OWL Working Group 2012; Casellas 2011; Grau i in. 2008; Yu 2011; Calvanese i in. 2007].

Zależności między pełnym językiem OWL 2 oraz jego profilami, a także pomiędzy odmianami języka OWL 1 zawarto na rys. 4.



Rys. 4. Zależności między profilami języka OWL 2 i odmianami języka OWL 1

Źródło: opracowanie własne na podstawie [Grau i in. 2008; Antoniou, van Harmelen 2009; Yu 2011; W3C OWL Working Group 2012].

Analizując rysunek 4 (a), można zauważyć, że nieformalny profil OWL 2 DL zawiera się w pełnym języku OWL 2 i zawiera on formalne profile OWL 2, które mają również część wspólną. Interpretując rysunek 4 (b), można stwierdzić, że odmiana OWL DL jest bardziej ekspresywna od OWL Lite, ale mniej ekspresywna od odmiany OWL Full. Najtrudniej jest ustalić zależności między profilami języka OWL 2 i odmianami OWL 1, przedstawione w części (c) rys. 4. Język OWL Full zawiera się w pełnym języku OWL 2, gdyż OWL 2 rozszerzył OWL Full o nowe konstrukcje. Z tego samego względu odmiana OWL DL zawiera się w profilu OWL 2 DL. Ponadto OWL 2 DL zawiera konstrukcje, które nie występowały w języku OWL Full (np. zwrotność), ale też nie zawiera wszystkich konstrukcji występujących w OWL Full. Wobec tego między tymi zbiorami nie zachodzi relacja zawierania się zbiorów. Profil OWL 2 EL zawiera np. klucze, niewystępujące w OWL Full, więc profil ten

wykracza poza zbiór OWL Full oraz OWL DL. Podobna sytuacja zachodzi dla pozostałych profili: OWL 2 QL zawiera np. asymetryczność, a OWL 2 RL – rozłączność właściwości. Konstrukcje te nie występowały w języku OWL Full, więc profile OWL 2 QL i OWL 2 RL wykraczają poza język OWL Full. Jednocześnie profile te, podobnie jak profil OWL 2 DL, nie zawierają wielu konstrukcji dostępnych w języku OWL Full, wobec czego nie zachodzi tutaj relacja zawierania się zbiorów. Na rysunku 4 odzwierciedlono także podobieństwo między profilem OWL 2 QL oraz odmianą OWL Lite.

Usystematyzowanie informacji o poszczególnych językach reprezentacji ontologii oraz odmianach i profilach języka OWL zawiera tab. 1. Opierając się na tab. 1, można stwierdzić, że każdy język ontologii pozwala reprezentować wiele jej składników. Ponadto nie są to tylko proste elementy (takie jak zmienne w klasycznych językach programowania), ale również konstrukcje złożone z wielu elementów podstawowych. Języki reprezentacji ontologii są więc językami o względnie dużej złożoności.

Aby zbudować ontologię reprezentującą wiedzę z określonej dziedziny, należy wybrać odpowiedni język jej reprezentacji. Wybrany język powinien się charakteryzować rozstrzygalnością oraz powinien zapewniać formalność ontologii w znaczeniu omówionym w punkcie 2 artykułu. Pozwoli to utworzyć bazę wiedzy, w której będzie można przeprowadzać pełne, rozstrzygalne wnioskowanie. Wobec tego z dalszych rozważań należy wyłączyć języki niezapewniające formalności ontologii, takie jak np. XML, języki i diagramy modelowania baz danych oraz diagramy UML. Ponadto odrzucić należy języki modelowania wiedzy i reprezentacji ontologii, które nie zapewniają rozstrzygalności. Wśród takich języków wymienić należy języki opierające się na ramach i logice I rzędu, tj.: Ontolingua i OCML, oraz LOOM bazujący na logice opisowej. Odrzucenie wymienionych języków związane jest też z faktem, że obecnie dynamicznie rozwijają się języki wykorzystywane w Sieci Semantycznej (*Semantic Web*), podczas gdy tradycyjne języki ontologii są coraz rzadziej stosowane.

Kolejnymi językami, których zastosowania w budowie ontologii dziedzinowych można nie brać pod uwagę, są języki o małej ekspresywności (co ujęto w tab. 1), tj.: SHOE, RDF, RDFS i RDF(S). Zaznaczyć również należy, że języki RDF, RDFS i RDF(S) nie są zazwyczaj wykorzystywane do budowy ontologii, stosowane są one natomiast do opisu danych w sieci. Ponadto, ze względu na fakt, że język OWL 2 jest następcą OWL i jest od niego bogatszy semantycznie i bardziej ekspresywny, należy wziąć pod uwagę zastosowanie nowszej wersji języka OWL, tj. OWL 2.

Po odrzuceniu wymienionych języków reprezentacji ontologii do rozważenia pozostały języki: FLogic oraz profile języka OWL 2. Rozważając zastosowania języków FLogic i OWL [Gomez-Perez, Fernandez-Lopez, Corcho 2004], zaznaczają, że język FLogic znajduje zastosowanie w budowie metod rozwiązywania problemów (*Problem Solving Methods*). Z kolei języki opierające się na logice opisowej (jak np. OWL) są często stosowane w integracji heterogenicznych źródeł danych

Tabela 1. Cechy poszczególnych języków ontologii

Cechy		Ontolingua	LOOM	OCML	FLogic	SHOE	RDF(S)	OWL Lite	OWL DL	OWL Full	OWL 2 RL	OWL 2 DL	OWL 2 Full	
Ekspresywność	Koncepty i ograniczenia	Atrybuty instancji	+	+	+	+	+	+	+	+	+	+	+	
		Atrybuty klas	+	-	+	+	-	p	p	p	p	+	+	+
		Ograniczenia typów	+	+	+	+	+	+	+	+	+	+/-	+	+
		Ograniczenia liczności	+	+	+	p	-	-	+/-	+	+	-	+	+
	Taksonomie konceptów	Podklasy	+	+	+	+	+	+	+	+	+	+	+	+
		Rozłączność-dekompozycja	+	+	+	p	-	-	p	+	+	-	+	+
		Podział zupełny – dekompozycja	+	-	p	p	-	-	p	p	p	-	+	+
		Partycje	+	+	+	p	-	-	p	p	p	-	+	+
	Relacje	Relacje binarne	+	+	+	+	+	+	+	+	+	+	+	+
		Relacje <i>n</i> -arne	+	+	+	p	+	p	-	-	p	p	p	+
		Hierarchia relacji	+	+	+	p	-	+	+	+	+	p	+	+
		Więzy integralności	+	+	+	+	-	-	-	-	-	-	-	-
	Funkcje	Funkcje binarne	+	+	+	+	-	-	-	+	+	-	+	+
		Funkcje <i>n</i> -arne	+	+	+	+	-	-	-	-	-	-	-	-
	Inne	Aksjomaty formalne	+	+	+	+	-	-	-	-	-	p	p	p
		Instancje	+	+	+	+	+	+	+	+	+	+	+	+
		Reguły	-	+	+	-	+	-	-	-	-	-	-	-
	Rozstrzygalność		-	-	-	*	-	-	+	+	-	+	+	-
	Monotoniczność		-	+	-	-	-	-	+	+	+	+	+	+

„+” – cechy wspierane przez język; „-” – cechy niewspierane; „+/-” – cechy wspierane tylko częściowo; „p” – cechy niewspierane przez dany język, ale względnie łatwe do zaimplementowania w sposób pośredni; „*” – rozstrzygalny tylko w jego monotonicznej części.

Źródło: opracowanie własne na podstawie [Corcho, Fernandez-Lopez, Gomez-Perez 2006; Casellas 2011].

i wyszukiwaniu informacji. Dodatkowo języki, takie jak OWL, mają silne wsparcie od społeczności, wobec czego szeroko dostępne są narzędzia do edycji i dokumentowania ontologii budowanych w tych językach. Ponadto mechanizmy wnioskujące dla języka FLogic zapewniają pełne wnioskowanie tylko dla jego monotonicznej

części, wobec czego, ogólnie rzecz ujmując, język ten może być nierozstrzygalny. Casellas [2011] w kontekście języka OWL stwierdza, że język ten jest obecnie często stosowany do opracowywania ontologii i może on stanowić podstawę do ponownego użycia i rozszerzania ontologii. Ponadto OWL stale ewoluuje, jest wzbogacany i rozszerzany, a edytorzy ontologii podążają za jego ewolucją. Według badań przeprowadzonych w 2007 r. w niemal 76% budowanych ontologii stosowany był język OWL, podczas gdy FLogic wykorzystano tylko w 12% przypadków [Cardoso 2007]. Pomimo braku nowszych danych, w związku z dynamicznym rozwojem języka OWL, można przypuszczać, że obecnie stosowalność tego języka jest jeszcze większa. Wobec tych argumentów można założyć, że wybór języka OWL 2 do budowy ontologii dziedzinowych będzie najwłaściwszy. Język ten oferuje dużą ekspresywność, liczną społeczność użytkowników oraz bogatą dokumentację zapewnianą przez konsorcjum W3C. Jest on (wraz z wersją OWL 1) również wspierany przez liczne edytorzy ontologii, a jego funkcjonalność jest rozszerzana dzięki stosowaniu reguł SWRL [O'Connor i in. 2005].

Aby zagwarantować rozstrzygalność języka OWL, należy zastosować jego wersję, dla której mechanizmy wnioskujące potrafią przeprowadzić pełne wnioskowanie. Najbardziej ekspresywną wersją OWL spełniającą ten warunek jest nieformalny profil OWL 2 DL. W związku z tym język ten charakteryzuje się wysoką stosowalnością w budowie ontologii dziedzinowych umożliwiających wnioskowanie.

5. Wnioski

Rosnące znaczenie zarządzania wiedzą powoduje, że konieczne staje się jednoznaczne zdefiniowanie i wybór formalnych mechanizmów opisu wiedzy. W warstwie praktycznej problem ten polega na właściwym ocenie i doborze języka reprezentacji wiedzy w ontologiach (to właśnie ontologie stanowią aktualnie podstawową formę inżynierii wiedzy).

Ocenę rozpoczęto od dyskusji formalizmów (ramy, programowanie logiczne, logika opisowa, logika pierwszego rzędu) stanowiących podstawę języków ontologii. Jednoznacznie wskazano, że poszczególne języki opierają się na różnych formalizmach i ich kombinacjach, różniąc się wzajemnie stopniem formalizacji, ekspresywnością i złożonością wnioskowania. Następnie przedstawiono analizę porównawczą dostępnych języków reprezentacji wiedzy, uwzględniając w niej zależności pomiędzy różnymi profilami i odmianami języków opartych na znacznikach.

Łatwo zauważyć, że poszczególne języki zdecydowanie różnią się stopniem ekspresywności. Jednocześnie należy wskazać, że w zakresie ekspresywności najbardziej uniwersalnym jest język OWL, który ponadto jest językiem monotonicznym. Jego poszczególne wersje uwzględniają wszystkie istotne struktury logiczne, takie jak koncepty i ograniczenia, relacje oraz instancje konceptów. Dodatkowo, język OWL jest uzupełniany przez regułowy język SWRL umożliwiający wspomaganie wnioskowania w ontologiach. Nie bez znaczenia pozostaje również fakt, że ważniej-

sze systemy wspierające budowę ontologii obsługują ten język, a w środowisku Web 3.0 (sieć Internet trzeciej generacji) stał się on standardem semantycznego opisu zawartości zasobów sieciowych.

Zgodnie z wynikami przedstawionej analizy język OWL, a w szczególności jego profil OWL 2 DL, charakteryzuje się najwyższą stosowalnością w konstruowaniu ontologii dziedzinowych. Wobec tego podczas budowy ontologii należy rozważyć przede wszystkim zastosowanie języka OWL 2 DL. Oczywiście dobór języka powinien być także dostosowany do potrzeb danej ontologii i jej ekspresywności. Jeżeli nie ma potrzeby zawierać w ontologii pewnych konstrukcji logicznych, wtedy wystarczający może okazać się inny, uboższy profil OWL 2 lub OWL 1, zgodnie z rys. 4 i tab. 1.

Jeżeli chodzi o kierunki dalszych badań, to wskazać należy, że przy budowie ontologii można stosować różne metodyki, a samą implementację można wykonać przy użyciu różnych narzędzi informatycznych. Istotnym zagadnieniem, które należy, zdaniem autorów, podjąć w dalszych badaniach, jest więc dobór odpowiedniej metodyki projektowania ontologii i edytora służącego do jej implementacji. Właściwy dobór i dopasowanie tych trzech elementów pozwoli w efekcie zoptymalizować proces opracowywania ontologii reprezentujących wiedzę dziedzinową.

Literatura

- Abraham A., 2003, *Intelligent systems: Architectures and perspectives*, "Studies in Fuzziness and Soft Computing", vol. 113, s. 1-35.
- Antoniou G., van Harmelen F., 2009, *Web Ontology Language: OWL*, [w:] S. Staab, R. Studer (red.), *Handbook On Ontologies*, second edition, International Handbooks on Information Systems, Springer, s. 91-110.
- Baader F., Horrocks I., Sattler U., 2009, *Description Logics*, [w:] S. Staab, R. Studer (red.), *Handbook On Ontologies*, second edition, International Handbooks on Information Systems, Springer, s. 21-43.
- Brdulak J.J., 2004, *Systemy informatyczne w zarządzaniu wiedzą*, "E-mentor", nr 2, s. 48-51.
- Calvanese D., De Giacomo G., Lembo D., Lenzerini M., 2007, *Tractable reasoning and efficient query answering in description logics: The DL-lite family*, "Journal of Automated Reasoning", vol. 39, no. 3, s. 385-429.
- Cardoso J., 2007, *The semantic web vision: Where are we?*, "IEEE Intelligent Systems", vol. 22, no. 5, s. 84-88.
- Casellas N., 2011, *Methodologies, tools and languages for ontology design*, [w:] N. Casellas (red.), *Legal Ontology Engineering. Methodologies, Modelling Trends, and the Ontology of Professional Judicial Knowledge*, Springer, s. 57-107.
- Chaudhary D., Yadaav P.K., Singh R.K., Mitra S., Ghaziabad S., 2012, *Integrated knowledge base: An approach to knowledge extraction*, "International Journal of Computer Applications", no. 6, s. 19-26.
- Corcho O., Fernandez-Lopez M., Gomez-Perez A., 2006, *Ontological engineering: Principles, methods, tools and languages*, [w:] C. Calero, F. Ruiz, M. Piattini (red.), *Ontologies for Software Engineering and Software Technology*, Springer, s. 1-48.
- Cummings L. (red.), 2005, *Pragmatics: A Multidisciplinary Perspective*, Edinburg University Press.

- Czarnecki A., Sitek T., 2013, *Ontologies vs. rules – comparison of methods of knowledge representation based on the example of IT services management*, [w:] L. Borzemski, A. Grzech, J. Świątek, Z. Wilimowska (red.), *Information Systems Architecture and Technology: Intelligent Information Systems, Knowledge Discovery, Big Data and High Performance Computing*, Oficyna Wydawnicza Politechniki Wrocławskiej, s. 99-109.
- De Bruijn J., 2007, *Logics for the semantic Web*, [w:] J. Cardoso (red.), *Semantics Web Services: Theory, Tools and Applications*, IGI Global, s. 24-43.
- Goczyła K. (red.), 2011, *Ontologie w systemach informatycznych*, Akademicka Oficyna Wydawnicza EXIT, Warszawa.
- Gomez-Perez A., Fernandez-Lopez M., Corcho O., 2004, *Languages for building ontologies*, [w:] A. Gomez-Perez, M. Fernandez-Lopez, O. Corcho (red.), *Ontological Engineering, With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*, Springer, s. 199-292.
- Grau B.C., Horrocks I., Motik B., Parsia B., Patel-Schneider P., Sattler U., 2008, *OWL 2: The next step for OWL*, “Web Semantics: Science, Services and Agents on the World Wide Web”, vol. 6, no. 4, s. 309-322.
- Grosz B.N., Horrocks I., Volz R., Decker S., 2003, *Description Logic Programs: Combining Logic Programs with Description Logic*, Proceedings of the 12th international conference on World Wide Web, ACM, s. 48-57.
- Gruber T.R., 1993, *A translation approach to portable ontology specifications*, “Knowledge Acquisition”, vol. 5, no. 2, s. 199-220.
- Gruninger M., Lee J., 2002, *Ontology applications and design*, “Communications of the ACM”, vol. 45, no. 2, s. 39-41.
- Guarino N., Oberle D., Staab S., 2009, *What is an ontology?*, [w:] S. Staab, R. Studer (red.), *Handbook on Ontologies*, second edition, International Handbooks on Information Systems, Springer, s. 1-17.
- Guzman-Arenas A., Cuevas A.D., 2010, *Knowledge accumulation through automatic merging of ontologies*, “Expert Systems with Applications”, vol. 37, no. 3, s. 1991-2005.
- Hartley R.T., 1985, *Representation of Procedural Knowledge for Expert Systems*, IEEE 2nd. Conf. on AI Applications.
- Hirst G., 1992, *Semantic Interpretation and the Resolution of Ambiguity*, Cambridge University Press.
- ISO/IEC 24707:2007(E), 2007, *Information technology – Common Logic (CL): A Framework for a Family of Logic-based Languages*.
- Knorr M., Alferes J.J., Hitzler P., 2011, *Local closed world reasoning with description logics under the well-founded semantics*, “Artificial Intelligence”, vol. 175, no. 9-10, s. 1528-1554.
- Koide S., Takeda H., 2011, *Common languages for web semantics*, [w:] L.A. Maciaszek, P. Loucopoulos (red.), *Evaluation of Novel Approaches to Software Engineering*, Springer, s. 148-162.
- Kutsia T., Marin M., 2012, *Solving, Reasoning, and Programming in Common Logic*, Proceedings of the 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, s. 119-126.
- Łabuzek M., 2003, *Wykorzystanie metamodelowania do specyfikacji ontologii znaczenia opisów rzeczywistości*, E-informatyka.pl, http://www.e-informatyka.pl/attach/Wykorzystanie_meta-modelowania_do_specyfikacji_ontologii_znaczenia_opisow_rzeczywistości/402.pdf (21.03.2015).
- Nebel B., 2001, *Frame-based systems*, [w:] R.A. Wilson, F. Keil (red.), *The MIT Encyclopedia of the Cognitive Sciences (MITECS)*, MIT Press, s. 324-326.
- O'Connor M., Knublauch H., Tu S., Grosz B., Dean M., Grosso W., Musen M., 2005, *Supporting rule system interoperability on the semantic Web with SWRL*, “Lecture Notes in Computer Science”, vol. 3729, s. 974-986.
- Rao A.R., Jain R., 1988, *Knowledge representation and control in computer vision systems*, “IEEE Expert”, vol. 3, no. 1, s. 64-79.

- Roussey C., Pinet F., Kang M.A., Corcho O., 2011, *An introduction to ontologies and ontology engineering*, [w:] G. Falquet, C. Metral, J. Teller, C. Tweed (red.), *Ontologies in Urban Development Projects*, Springer, s. 9-38.
- Roy B., 1996, *Multicriteria Methodology for Decision Aiding*, Springer, Dordrecht.
- Saa R., Garcia A., Gomez C., Carretero J., Garcia-Carballeira F., 2012, *An ontology-driven decision support system for high-performance and cost-optimized design of complex railway portal frames*, "Expert Systems with Applications", vol. 39, no. 10, s. 8784-8792.
- Shiau W.L., 2011, *A profile of information systems research published in expert systems with applications from 1995 to 2008*, "Expert Systems with Applications", vol. 38, no. 4, s. 3999-4005.
- Shue L.Y., Chen C.W., Shiue W., 2009, *The development of an ontology-based expert system for corporate financial rating*, "Expert Systems with Applications", vol. 36, no. 2, s. 2130-2142.
- Tripathi K.P., 2011, *Artificial intelligence techniques – Novel approaches & practical applications*, "International Journal of Computer Applications", no. 4, s. 19-23.
- Truemper K. (red.), 2004, *Design of Logic-based Intelligent Systems*, Wiley-Interscience, Hoboken.
- Uschold M., Gruninger M., 2004, *Ontologies and semantics for seamless connectivity*, "SIGMOD Record", vol. 33, no. 4, s. 58-64.
- Villanueva-Rosales N., Dumontier M., 2008, *yOWL: An ontology-driven knowledge base for yeast biologists*, "Journal of Biomedical Informatics", vol. 41, no. 5, s. 779-789.
- W3C OWL Working Group, 2012, *OWL 2 Web Ontology Language, Document Overview (Second Edition)*, <http://www.w3.org/TR/owl2-overview> (24.03.2015).
- Waloszek A. (red.), 2010, *Hierarchiczna kontekstualizacja baz wiedzy*, rozprawa doktorska, Politechnika Gdańska.
- Waloszek W. (red.), 2007, *Metody strukturalnej analizy ontologii opartych na logice opisowej*, rozprawa doktorska, Politechnika Gdańska.
- Wang H.H., Noy N., Rector A., Musen M., Redmont T., Rubin D., Tu S., Tudorache T., Drummond N., Horridge M., Seidenberg J., 2006, *Frames and OWL Side by Side*, Proceedings of the 9th international Protégé conference, s. 54-57.
- Wong B.K., Monaco J.A., 1995, *Expert system applications in business: A review and analysis of the literature (1977-1993)*, "Information & Management", vol. 29, no. 3, s. 141-152.
- Wu Z., Chen H. (red.), 2008, *Semantic Grid: Model, Methodology, and Applications*, Springer.
- Yu L., 2011, *OWL: Web ontology language*, [w:] L. Yu (red.), *A Developer's Guide to the Semantic Web*, Springer, s. 155-239.
- Zhu L., Yang Q., Chen W., 2009, *Research on Ontology Integration Combined with Machine Learning*, Proceedings of the Second International Conference on Intelligent Computation Technology and Automation, IEEE, vol. 1, s. 464-467.