

Inżynieria ontologii dla potrzeb integracji systemów.

Boiński Tomasz Orłowski Piotr Szpryngier Piotr

7 września 2012

Streszczenie. Inżynieria ontologii to gałąź inżynierii wiedzy, która zajmuje się metodami i metodologiami budowy ontologii. Zajmuje się m. in. procesem wytwarzania, cyklem życia oraz narzędziami wspomagającymi wytwarzanie ontologii. W publikacji przedstawiono różne metodologie wytwarzania ontologii oraz wspierające te metodologie narzędzia. Omówiono edytor ontologii OCS, opracowany przez Katedrę ASK Wydziału ETI PG, oraz wykorzystaną w nim, przygotowaną z myślą o pracy grupowej, metodologię wytwarzania ontologii.

Słowa kluczowe. inżynieria ontologii, integracja systemów

1 Ontologie jako podstawa w komunikacji pomiędzy nowoczesnymi systemami informatycznymi

Powszechnie wiadomo, że podstawą dobrej komunikacji jest wspólny język. Wiedza, którą przez wieki gromadził człowiek, mogła być przyswajana i rozwijana przez kolejne pokolenia właśnie dzięki temu czynnikowi. Niestety to, co dla człowieka wydaje się proste i oczywiste, nie zawsze jest takie proste do zaimplementowania w programach komputerowych. Tak właśnie dzieje się między innymi ze zrozumieniem informacji zgromadzonej w formie tekstowej. Przetwarzanie języka naturalnego, nawet w wypadku języków o tak mało skomplikowanej składni jak język angielski, nadal jest zagadnieniem bardzo złożonym. Dostępne algorytmy mają wykładniczą złożoność obliczeniową [36], co powoduje, że programy z nich korzystające nadal wykonują się zbyt długo. Do podstawowych problemów można tu zaliczyć, między innymi, kontekstowość języka, elastyczność gramatyki i składni, a także wieloznaczność słów i pojęć.

Termin ontologia został zaadaptowany do informatyki w latach 70. i 80. ubiegłego wieku przez badaczy zajmujących się sztuczną inteligencją [24, 20].

Zadanie Badawcze SYNAT pt.: „Utworzenie uniwersalnej, otwartej, repozytoryjnej platformy hostingowej i komunikacyjnej dla sieciowych zasobów wiedzy dla nauki, edukacji i otwartego społeczeństwa wiedzy”

Program strategiczny: Interdyscyplinarny system interaktywnej informacji naukowej i naukowo technicznej, finansowany przez Narodowe Centrum Badań i Rozwoju, Nr Umowy SP/I/1/77065/10

Niniejsza praca jest finansowana przez Narodowe Centrum Badań i Rozwoju nr umowy SP/I/1/77065/10 w ramach programu strategicznego: Interdyscyplinarny system interaktywnej informacji naukowej i naukowo technicznej.

Zauważyli oni, że zastosowanie ontologii jako modelu danych pozwala na użycie logiki matematycznej do automatycznego wnioskowania. Wtedy termin ten oznaczał zarówno teorię modelowania świata, jak i komponent systemu opartego na wiedzy, a wielu badaczy uważało ontologię komputerową za gałąź filozofii stosowanej.

Początkowo bardzo szerokie pojęcie ontologii w informatyce zawęziło się i na początku lat 90. obejmowało już tylko dowolną skategoryzowaną listę rzeczy i pojęć (słownik), które program komputerowy miał przyjąć za istniejące [8].

W połowie lat 90. ubiegłego stulecia definicja ontologii skryształizowała się. Główne środowiska badawcze przyjęły zaproponowaną przez Studera [43], a opartą na pracach Grubera [16] szczegółową definicję brzmiącą „ontologia to sformalizowana i jawna specyfikacja współdzielonej konceptualizacji”.

W trakcie prac nad definicją ontologii, często ignorowanym jej aspektem jest nacisk na ich współdzielenie [18]. Przykładowo, przez długi czas uznawana za obowiązującą definicja ontologii Grubera [16] w ogóle nie uwzględniała tego atrybutu. Dopiero wraz z zastosowaniem ontologii w integracji heterogenicznych usług sieciowych badacze doszli do wniosku, że sposoby reprezentacji informacji nie biorące pod uwagę tego atrybutu ontologiami nie są.

Zastosowanie języków formalnych, czyli takich, które mają z góry ustalony słownik i składnię logiczną, pozwala na znaczące ograniczenie złożoności obliczeniowej przetwarzania języka. I tutaj właśnie ontologia, jako jasna, sformalizowana specyfikacja wspólnej konceptualizacji, znajduje swoje zastosowanie. Użycie ontologii pozwala w znacznym stopniu nadać tekstowi znaczenie łatwo zrozumiałe również dla maszyn. Ułatwia to wyeliminowanie występujących w komunikacji pomiędzy programami problemów związanych z niejednoznacznością pojęć. Ponadto, przy odpowiednim skomponowaniu języka ontologii, umożliwia wnioskowanie na podstawie poznanego zbioru faktów.

2 Czym jest inżynieria ontologii?

Inżynieria ontologii to gałąź inżynierii wiedzy, która zajmuje się metodami i metodologiami budowy ontologii. Zajmuje się m. in. procesem wytwarzania, cyklem życia oraz narzędziami wspomagającymi wytwarzanie ontologii [9, 46].

Do połowy lat 90. proces wytwarzania ontologii polegał na żmudnym, dość chaotycznym, modelowaniu pojęć i łączeniu ich relacjami. Poszczególne ośrodki badawcze używały w tym celu własnych, często opartych na logice matematycznej, metod i metodologii, a samo wytwarzanie ontologii miało więcej wspólnego ze sztuką niż z inżynierią [11]. Za autorów pierwszych prac, które opisywały właściwą inżynierię ontologii, przyjmuje się Uscholda i Gruningera [17, 48, 49] oraz Grubera [15]. Gruninger [17] spopularyzował stosowanie pytań CQ (ang. Competency Questions). Uschold [49] upowszechnił metody specyfikacji, konceptualizacji, oceny jakości i dokumentacji ontologii. Gruber [15] poddał pod dyskusję jakość procesu wytwarzania oraz samej ontologii wynikowej. Prace te stanowią bazę dla wielu następujących po nich rozwiązań.

Projekty z ostatnich lat XX i pierwszych lat XXI wieku, np. mające na celu stworzenie ontologii wysokiego poziomu, takich jak CyC [23], wielkich repozytoriów ontologii jak OntoWeb [39] czy Obofoundry [30], oraz wzrost zainteresowania komputerowymi ontologiami w środowisku bioinformatyków i chemików [42] zaowocowały wykształceniem się współczesnej inżynierii ontologii, która charak-



teryzuje się wielością stosowanych technik, metod, procesów, metodologii [13]. Niestety, inżynieria ontologii nie posiada jeszcze ustandaryzowanego aparatu pojęciowego [45]. Interesujące jest chociażby to, że niektóre środowiska ogół procesów związanych z wytwarzaniem ontologii nazywają budowaniem ontologii [9], inne inżynierią ontologii [12, 41], a jeszcze inne inżynierią ontologiczną [35].

W ciągu ostatnich kilku lat w ramach ogólnie pojętych prac W3C nad Siecią Semantyczną, a także kosztownych projektów Unii Europejskiej, takich jak NeOn (Networked Ontologies [28]) czy Athena [3], czy też działań OMG (Object Management Group) na rzecz stworzenia standardowego języka modelowania ontologii [12, 32] inżynieria ontologii prężnie się rozwijała. Systematycznie pojawiały się publikacje-kompedia z nią związane, np. *Ontology Engineering* Gómeza-Pérez [14], dwa wydania *Handbook on Ontologies* Staaba i Studera [41], *Ontology Matching* Euzenata i Shvaiko [10] czy *Model Driven Ontology Engineering* [12]. W chwili pisania tej pracy zauważalny jest trend do ujednoczenia stosowanych aparatów pojęciowych oraz rozwoju narzędzi wspierających różne fazy cyklu życia ontologii.

3 Elementy inżynierii ontologii

W tej części zaprezentowano wybrane metodologie wytwarzania ontologii, dostępne narzędzia oraz metodologię testowania poprawności ontologii. Zebrano również fazy procesu wytwarzania ontologii wspólne dla wszystkich metodologii. W trakcie wytwarzania ontologii można stosować dowolną kombinację tych elementów, niezależnie od używanego języka zapisu ontologii, jednak z uwagi na dostępność i jakość narzędzi wspierających wytwarzanie ontologii, w niektórych językach może to być znacznie utrudnione.

3.1 Metodologie wytwarzania ontologii

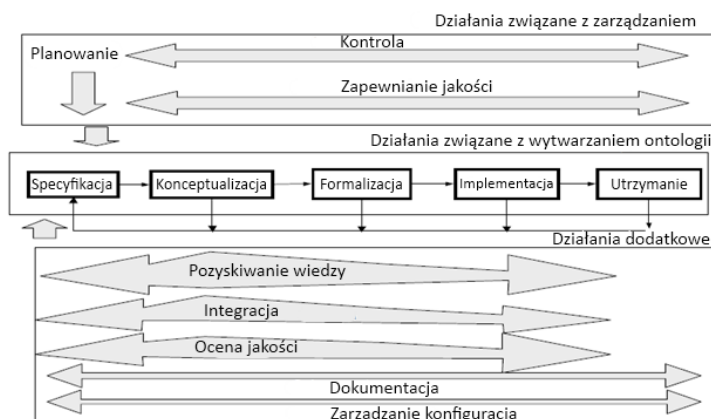
Gasevic [12] definiuje metodologie wytwarzania ontologii jako „przyjęte zasady, procesy, praktyki, metody i czynności stosowane do projektowania, konstruowania, oceny jakości i wdrażania ontologii”. Publikacje Corcho i in. [8], Staaba i Studera [40] oraz Glińskiego [13] zawierają przegląd dostępnych rozwiązań. Nie obejmują one jednak nowszych metodologii, takich jak NeOn [46] i UPON [9] czy zespołowej metody wytwarzania stosowanej przez system OCS [31, 5].

Methontology [11] przez długi czas była uważana za najbardziej kompletną metodologię wytwarzania ontologii [12, 13]. Środowiska wytwarzania ontologii ODE i WebODE [1] zostały zbudowane w celu dostarczenia technicznego wsparcia dla tej metodologii. Inne narzędzia ontologiczne i środowiska do budowy ontologii (Protégé, OntoEdit itd.) również częściowo wspierają tę metodologię wytwarzania ontologii. Była ona również rekomendowana przez nieistniejącą już fundację FIPA (Foundation for Intelligent Physical Agents) [8]. Szczegóły tej metodologii, dokładny opis procesu i poszczególnych faz znajdują się w artykułach Fernandez [11] i Glińskiego [13]. Na rys. 1 zilustrowano cykl życia ontologii stosowany w tej metodologii.

Podstawowe cechy tej metodologii obejmują:

- stosowanie aparatu pojęciowego inżynierii wiedzy i inżynierii oprogramowania,





Rysunek 1: Proces tworzenia ontologii w Methontology [13].

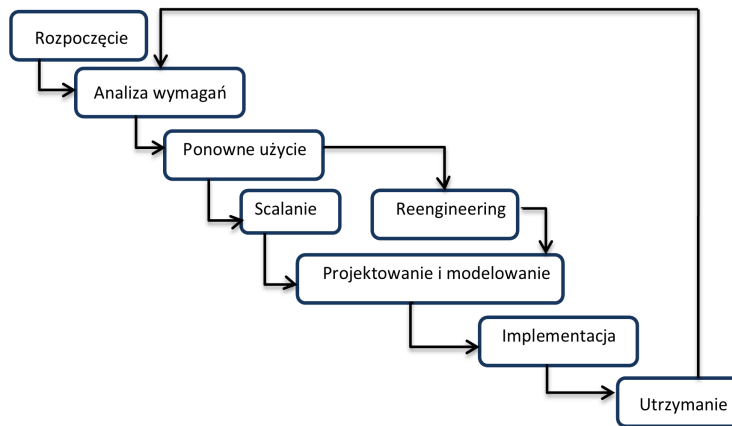
- pięcioletni cykl życia, nazwy faz pochodzą od nazw głównych czynności związanych z wytwarzaniem ontologii,
- budowa ontologii oparta na ewoluujących prototypach,
- wytwarzanie ontologii, zarządzanie projektem i czynności dodatkowe wykonywane są w tym samym czasie, co wymusza dość skomplikowane zarządzanie nimi,
- szczegółowa dokumentacja i wiele publikacji pokazujących jej zastosowanie,
- szczegółowy opis oczekiwanych produktów poszczególnych faz i sposobu ich oceny,
- duży nacisk na zarządzanie jakością,
- brak rozwiązań wspomagających współpracę.

Metodologia NeOn (od Networked Ontologies) została opracowana na potrzeby projektu Unii Europejskiej o takiej samej nazwie [26]. Jest to nowe rozwiązanie (opracowane w latach 2006-2010) współtworzone przez 14 ośrodków badawczych. Jej celem było podsumowanie dotychczasowego stanu wiedzy w dziedzinie inżynierii ontologii. Poza tworzeniem standardowych ontologii rozwinięto tu również zagadnienia związane z tworzeniem ontologii sieciowych (ang. Networked Ontologies), w szczególności ponowne użycie istniejących zasobów oraz wzorców projektowych. Narzędzia takie jak NeOn Toolkit i Protégé wspierają (bezpośrednio lub poprzez wtyczki) większość procesów i działań w niej opisanych.

Na rys. 2 przedstawiono model kaskadowy wytwarzania ontologii wg metodologii NeOn. Warto podkreślić, że fazy ponownego użycia, scalania i reengineeringu są opcjonalne. Poza modelem kaskadowym metodologia ta pozwala na korzystanie z modelu spiralnego i ewolucyjnego. Dokładny opis poszczególnych faz, czynności, scenariuszy użycia i narzędzi można znaleźć w dokumentacji projektu [44, 45, 46].

Oto podstawowe cechy tej metodologii:





Rysunek 2: Cykl kaskadowy w metodologii NeOn.

- została opracowana z myślą o ontologiach sieciowych,
- cykl życia ma zmienną liczbę faz,
- zaproponowano i opisano wiele scenariuszy wytwarzania ontologii,
- położono nacisk na ponowne użycie dostępnych komponentów i na wytwarzanie ontologii, których inni będą mogli również używać,
- opracowano szczegółową dokumentację opisującą poszczególne procesy,
- jest to rozwiązanie nowe, częściowo nadal teoretyczne,
- podstawowe procesy oparto na sprawdzonych rozwiązaniach (z OTK [46, 47], Methontology i DILIGENT [40]),
- istnieje częściowe wsparcie dla pracy grupowej i zdalnej współpracy (stosowanie narzędzi Wiki, propozycja użycia DILIGENT).

Na podstawie doświadczeń związanych z tworzeniem ontologii na potrzeby bioinformatyki Noy i McGuinness [29] zaproponowali swoją własną iteracyjną metodę wytwarzania ontologii opartą na inżynierii wiedzy. Metodę tę zamieszczono w samouczku do edytora Protégé, dzięki czemu cieszy się ona dużą popularnością wśród użytkowników tej aplikacji. Metoda ta składa się z 7 kroków:

- Ustalenie dziedziny oraz granic ontologii.
- Wykorzystanie istniejących ontologii.
- Ustalenie wszystkich najważniejszych terminów w projektowanej ontologii.
- Definiowanie klas i hierarchii klas.
- Definiowanie własności klas.
- Definiowanie cech własności klas.
- Tworzenie instancji klas.



Po wykonaniu powyższych kroków oceniamy wytworzoną ontologię i powtarzamy cały cykl, poprawiając znalezione błędy, aż do uzyskania satysfakcjonującej nas ontologii.

Oto podstawowe cechy tej metodologii:

- metoda praktyczna,
- iteracyjno-ewolucyjne tworzenie ontologii,
- szczegółowy opis większości kroków,
- procesy takie jak ponowne użycie komponentów i ocena jakości nie zostały szczegółowo omówione,
- brak wsparcia dla pracy grupowej.

Rozwinięciem rozwiązania Noy'a i McGuinness'a jest metodologia wytwarzania ontologii wykorzystana w narzędziu OCS [5]. Przyjęte przez oryginalnych autorów założenia rozszerzone zostały o pracę grupową.

3.2 Fazy i zadania inżynierii ontologii

Gasević i in. [12] podkreślają, że znane metodologie wytwarzania ontologii zawierają podobne fazy, czynności i procesy. W ramach projektu NeOn rozpoznano 59 procesów, technik i metod inżynierii ontologii. W punkcie tym krótko przedstawiono główne fazy tworzenia ontologii wspólne dla większości metodologii. Dokładny opis poszczególnych czynności znaleźć można w dokumentacji projektu NeOn [44, 45, 46].

3.2.1 Faza rozpoczęcia i analizy wymagań

Fazy te występują w każdej opisanej wcześniej metodologii wytwarzania ontologii. W ich ramach najczęściej wykonywane są 3 czynności:

- studium wykonalności,
- planowanie,
- specyfikacja wymagań.

3.2.2 Faza ponownego użycia

Faza ta występuje, gdy w procesie tworzenia naszej ontologii chcemy zastosować inne ontologie lub podobne do niej zasoby nieontologiczne (taksonomie, glosariusze). W fazie tej istotne są następujące czynności:

- wyszukiwanie i wybór przydatnych zasobów i ontologii (poprzez np. porównywanie ontologii),
- ponowne użycie znalezionych zasobów nieontologicznych, modułów ontologii, całych ontologii, wzorców projektowych, twierdzeń,
- adaptacja (ang. customization) dostępnych ontologii do potrzeb danego użytkownika.

Często w fazie tej (lub w fazie reengineeringu) warto rozpatrzyć zastosowanie wzorców projektowych. Przykładowe wzorce projektowe można znaleźć na portalu internetowym Ontology Design Patterns [33].

3.2.3 Faza scalania

Do fazy tej dochodzi, gdy chcemy połączyć dwie lub kilka ontologii w jedną całość. W fazie tej przeprowadza się następujące czynności:

- dopasowywanie ontologii (ang. Ontology Matching) – odnajdywanie elementów wspólnych między ontologiami,
- zestawianie ontologii (ang. Ontology Alignment) – zebranie zależności pozwalających na odwzorowanie elementów jednej ontologii w drugiej (zwane również mapowaniem ontologii),
- scalanie ontologii (ang. Ontology Merging) – właściwe scalenie dwóch lub kilku ontologii w jedną.

3.2.4 Faza reengineeringu

W ramach procesu tworzenia ontologii może zaistnieć konieczność przeprojektowania ontologii. NeOn wymienia tu szereg różnych czynności. Do istotniejszych czynności tej fazy należą:

- wzbogacanie ontologii w nowe terminy (ang. Ontology Enrichment),
- modularyzacja i podział ontologii,
- restrukturyzacja ontologii, czyli zmiana hierarchii konceptów,
- inżynieria wsteczna, czyli uzyskiwanie modeli relacji konceptów z kodu źródłowego.

3.2.5 Faza projektowania

Jest to jedna z najważniejszych faz całego procesu wytwarzania ontologii. Czynności tu występujące to:

- konceptualizacja (ang. Ontology Conceptualization), modelowanie wiedzy uzyskanej ze źródeł, przeprowadzane na podstawie specyfikacji wymagań,
- ewolucja ontologii (ang. Ontology Evolution), czyli ogół działań mających na celu zmodyfikowanie ontologii przy jednoczesnym zachowaniu jej spójności,
- formalizacja – jest to zamiana modelu uzyskanego w konceptualizacji w model formalny (zapisany np. z użyciem logiki opisowej lub UML),
- integracja ontologii – czynność włączenia jednej ontologii do innej,
- lokalizacja – adaptacja ontologii na potrzeby innego języka lub kultury,
- aktualizacja i drobne modyfikacje.

3.2.6 Faza implementacji

W fazie tej wytwarza się i wdraża do użycia modele ontologiczne. Czynności implementacyjne zależą bezpośrednio od języka implementacji. Zwykle języki te pozwalają na bezpośrednie odwzorowanie modelu formalnego.



3.2.7 Faza utrzymania

W fazie utrzymania wyróżniamy dwie podstawowe aktywności:

- wersjonowanie ontologii,
- unowocześnianie ontologii (ang. Ontology Upgrade) – zmiana używanej ontologii na jej nową wersję.

3.2.8 Czynności dodatkowe

Istnieje również wiele czynności i procesów, których nie można zaliczyć do pojedynczej fazy cyklu wytwarzania ontologii. Z ważniejszych warto wymienić:

- opisywanie i dokumentowanie ontologii,
- ocena ontologii,
- zarządzanie konfiguracją,
- zarządzanie jakością,
- pozyskiwanie wiedzy,
- uzyskiwanie wiedzy na podstawie dyskusji z ekspertami dziedzinowymi,
- ontologiczne uczenie się,
- wypełnianie ontologii konceptami,
- diagnozowanie i naprawa ontologii.

3.3 Ocena i testowanie ontologii

Jak każdy inny produkt inżynierski, ontologia powinna być dokładnie przetestowana i oceniona. Metodologie takie jak Methontology czy UPON [9] podkreślają ważność testów i oceny ontologii na wszystkich etapach jej wytwarzania. O testowaniu i ocenie ontologii pisze między innymi Studer [40]. Wyodrębnia kilka podstawowych zagadnień i kryteriów oceny ontologii:

- weryfikacja (ang. verification) – polega na odpowiedzeniu na pytanie, czy ontologia została zbudowana w prawidłowy sposób,
- stwierdzanie poprawności (ang. validation) – czy dana ontologia to ontologia, którą chciano stworzyć?
- ścisłość (ang. accuracy) – czy ontologia przedstawia rzeczywiste relacje?
- możliwość dostosowania (ang. adaptability) – czy ontologię można łatwo rozbudować?
- przejrzystość (ang. clarity) – czy ontologia jest zrozumiała i dostatecznie udokumentowana?
- kompletność (ang. completeness) – czy ontologia w sposób wystarczająco dokładny opisuje daną dziedzinę wiedzy?

- zwięzłość (ang. conciseness) – czy ontologia zawiera tylko terminy z danej dziedziny wiedzy?
- wydajność obliczeniowa – czy aplikacje wnioskujące przetwarzają ontologię wystarczająco szybko?
- spójność (ang. consistency i coherency) – czy wnioskowanie z ontologii daje sensowne i zgodne z oczekiwanymi rezultaty?
- przydatność i dostępność komercyjna (ang. organizational fitness i commercial accessibility) – czy ontologia została wytworzona odpowiednimi narzędziami i w odpowiedni sposób?

Kryteria te mogą wykluczać się między sobą (np. kompletność i zwięzłość), a wybór odpowiednich kryteriów należy do inżyniera wiedzy.

4 Wybrane narzędzia wspierające inżynierię ontologii

W ciągu ostatnich kilku lat nie pojawił się ogólny i obiektywny przegląd narzędzi wspierających wytwarzanie ontologii. Najbliższy ideału jest przegląd Gasevicia [12] zawierający narzędzia wspomagające modelowanie ontologii. Pozostałe przeglądy, w tym teoretycznie najnowszy przegląd Mizoguchiego i Kozakiego [25], są mało aktualne i przedstawiają narzędzia, których wsparcie porzucano kilka lat temu.

Kilka interesujących narzędzi inżynierii ontologii dostarczył projekt SEKT [37]. Zostały one później wykorzystane w NeOn Toolkit (środowisku do wytwarzania ontologii stworzonym w projekcie NeOn).

Najistotniejsze narzędzia, takie jak Protégé [12] czy NeOn Toolkit [19], wspierają cały cykl wytwarzania ontologii. W ich skład zazwyczaj wchodzi:

- edytor ontologii,
- aplikacja do wnioskowania,
- aplikacje wspomagające pracę grupową,
- aplikacja wizualizująca ontologię,
- moduły dodatkowe (często jako wtyczki).

Obok wymienionych wcześniej środowisk, powstaje kilka aplikacji aspirujących do miana zintegrowanego środowiska wytwarzania ontologii, jak np. SWOOP [21] czy OCS [31].

Protégé uznawane jest za wiodące środowisko wytwarzania ontologii [12]. Zostało opracowane na Uniwersytecie Stanforda jako edytor ontologii z rozbudowanym mechanizmem wtyczek i jest ciągle rozwijane. Podstawowymi jego zaletami są rozbudowany, dobrze udokumentowany edytor oraz duże zaangażowanie użytkowników w proces rozwijania i dodawania nowych funkcji do właściwej aplikacji. Wg oficjalnej strony posiada ponad 150 tysięcy zarejestrowanych użytkowników. Występuje w dwóch głównych odmianach: Protégé OWL i Protégé Frames. Opracowano również jego nieco uboższą wersję internetową, tzw.

WebProtege, z dostępem przez przeglądarkę internetową. Protégé oparte jest na własnym modelu obiektowym i w związku z tym umożliwia zapisywanie ontologii w różnych formatach, między innymi w OWL. Protégé OWL wykorzystuje OWL API [2].

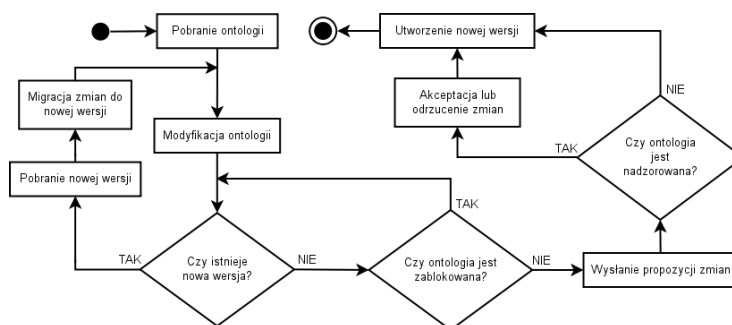
Stworzono do niego wtyczki, które wspierają niemal wszystkie wymienione wcześniej czynności związane z wytwarzaniem ontologii. W chwili pisania tej pracy do aktualnej wersji Protégé-OWL 4.02 dostępnych było ponad 30 wtyczek. Bibliotekę wtyczek wraz z opisem ich przeznaczenia można znaleźć na Protégé Wiki [34].

NeOn Toolkit [19] to środowisko wytwarzania ontologii z otwartym kodem źródłowym stworzone w ramach projektu NeOn [28]. Podobnie do OntoStudio jest ono oparte na platformie Eclipse. Pozwala na edycję ontologii w języku OWL. Występuje w dwóch znacząco różnych odmianach. Wersje 1.x oparte są na KAON API [6], natomiast wersje 2.x na OWL API [2]. Zastosowanie różnych API sprawia, że każda odmiana posiada odmienną listę wtyczek.

Wersja 1.2.3 posiada kilka wtyczek wspomagających automatyczne wytwarzanie ontologii na podstawie tekstu (aplikacje Text2Onto [7], LeDA i RoLExO [50]). Narzędzia te znalazły się jednak w „ślepych zaułku” inżynierii ontologii, z uwagi na zastosowanie KAON API, które od 2005 roku nie było rozwijane. Bibliotekę wtyczek z opisem ich przeznaczenia można znaleźć na stronie domowej projektu NeOn [27].

5 OCS - Ontology Creation System

Oparty na rozszerzonej metodologii Noy’a i McGuinness’a OCS (Ontology Creation System) [31] to system składowania i pozyskiwania ontologii powstający w Katedrze Architektury Systemów Komputerowych na Wydziale Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej.

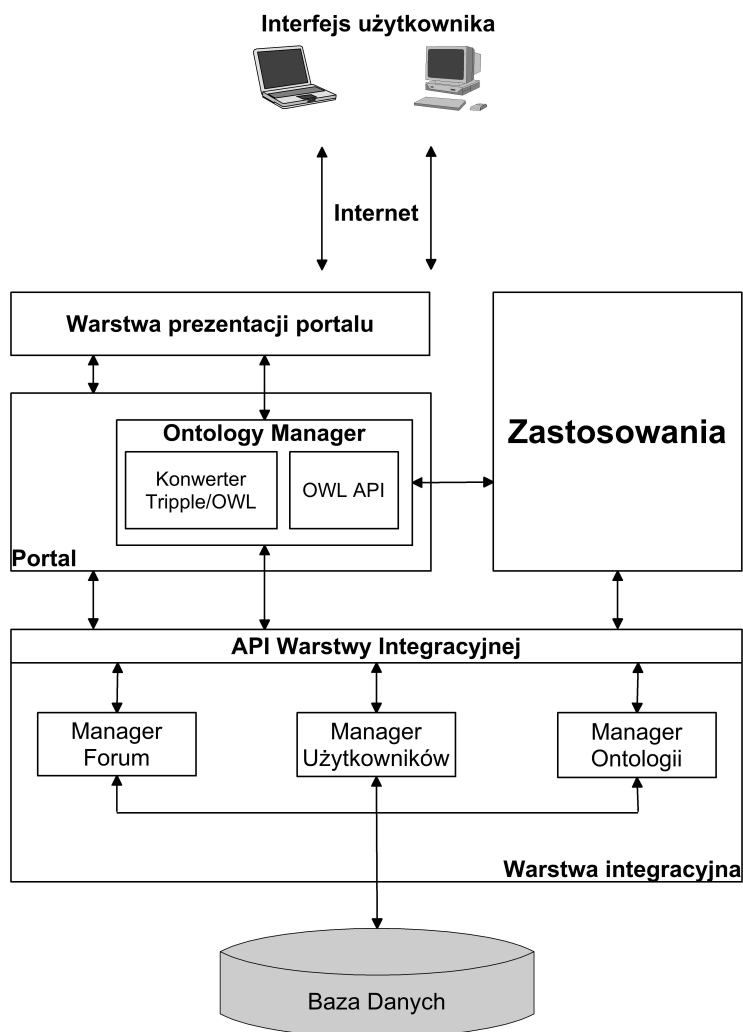


Rysunek 3: Proponowanie zmian oraz tworzenie nowych wersji ontologii w systemie OCS.

Podstawowa różnica pomiędzy narzędziem OCS a podobnymi rozwiązaniami wynika z założenia o pracy grupowej. Sam system, jak i metodologia, na jakiej został oparty, z założenia służyć ma zespołowemu, a nie indywidualnemu konstruowaniu ontologii. Na rys. 3 przedstawiono zastosowany model pracy grupowej [5]. Nad całym procesem wprowadzania zmian czuwa właściciel ontologii wraz z ekspertem dziedzinowym. Obaj mogą tworzyć nowe wersje, jak również blokować dostęp do ontologii.

Przyjęty model wytwarzania ontologii upraszcza późniejszą integrację systemów opartych na tak wytworzonych bazach wiedzy. Mnogość podmiotów wpływających na kształt ontologii gwarantuje szersze ujęcie opisywanego problemu i większą różnorodność w spojrzeniu na zagadnienie. Z kolei obecność pewnego ciała nadzorczego w postaci właściciela i ekspertów gwarantuje spójność tworzonego rozwiązania, a przez to jego użyteczność w praktycznych zastosowaniach.

OCS powstał na bazie platformy JEE. Ontologie zapisywane są w postaci plików RDF i udostępniane na zewnątrz w formie obiektów dzięki zastosowaniu OWL API. Architektura systemu przedstawiono na rys. 4.



Rysunek 4: Architektura systemu OCS.

W chwili pisania tej pracy OCS posiada wczesną wersję edytora ontologii i system wersjonowania i zarządzania ontologiami pozwalający na ich grupową edycję. Dodatkowo rozwijane są, między innymi, moduł SOVA [4] służący do wizualizacji ontologii oraz narzędzia do integracji ontologii. W ramach pracy



nad OCS stworzono również dwie wtyczki do Protégé. Jedna z nich udostępnia część funkcji wspierających współpracę [5], druga natomiast wizualizację z użyciem biblioteki SOVA [4, 38].

6 Podsumowanie

Ontologie i Sieć Semantyczna to zagadnienia, które od kilku lat zdobywają coraz większą popularność wśród badaczy i inżynierów zajmujących się sztuczną inteligencją i Internetem. Docelowo kolejnym etapem rozwoju Sieci Globalnej może być oparty na sztucznej inteligencji Web 3.0 [22]. Same ontologie mogą stać się w przyszłości podstawowym modelem danych czy też dostarczyć formalizmów niezbędnych do kontekstowej analizy dokumentów tekstowych. Przez długi czas ich opracowywanie przypominało sztukę i było dalekie od zasad inżynierii. Od pewnego czasu nauka o ich wytwarzaniu, zwana inżynierią ontologii, nabrała rozpędu. Aby wizja Sieci Semantycznej mogła się spełnić, konieczne jest tworzenie nowych i lepszych narzędzi oraz metod wytwarzania ontologii z uwzględnieniem zarówno ich bezpośredniego wykorzystania, jak i przyszłej integracji z globalną pajęczyną wiedzy, jaką stanie się Sieć Semantyczna.

Literatura

- [1] Arpírez, J.C., Corcho, O., Fernández-López, M. i Gómez-Pérez, A. 2001. WebODE: a scalable workbench for ontological engineering. Proceedings of the 1st international conference on Knowledge capture (2001), 6–13.
- [2] Bechhofer, S., Volz, R. i Lord, P. 2003. Cooking the Semantic Web with the OWL API. The SemanticWeb-ISWC 2003. (2003), 659–675.
- [3] Berre, A., Elvesaeter, B., Figay, N., Guglielmina, C., Johnsen, S., Karlsen, D., Knothe, T. i Lippe, S. 2007. The ATHENA Interoperability Framework. Enterprise Interoperability II. (2007), 569–580.
- [4] Boiński, T., Jaworska, A., Kleczkowski, R. i Kunowski, P. 2010. Ontology visualization. Information Technology (ICIT), 2010 2nd International Conference on (2010), 17-20.
- [5] Boiński, T., Jaworska, A., Kunowski, P. i Kleczkowski, R. 2010. Zespołowa budowa ontologii z wykorzystaniem systemu OCS oraz edytora Protégé. Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej. 8 (2010).
- [6] Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L. i others 2002. KAON - Towards a large scale semantic web. E-Commerce and Web Technologies. (2002), 231–248.
- [7] Cimiano, P. i Völker, J. 2005. Text2onto. Natural Language Processing and Information Systems. (2005), 227–238.
- [8] Corcho, O., Fernández-López, M. i Gómez-Pérez, A. 2003. Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & Knowledge Engineering. 46, 1 (2003), 41–64.

- [9] De Nicola, A., Missikoff, M. i Navigli, R. 2009. A software engineering approach to ontology building. *Information Systems*. 34, 2 (2009), 258–275.
- [10] Euzenat, J. i Shvaiko, P. 2007. *Ontology matching*. Springer-Verlag New York Inc.
- [11] Fernandez, M., Gomez-Perez, A. i Juristo, N. 1997. Methontology: from ontological art towards ontological engineering. *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering (1997)*, 33–40.
- [12] Gašević, D., Djurić, D. i Devedžić, V. 2009. *Model driven engineering and ontology development*. Springer-Verlag New York Inc.
- [13] Gliński, W. 2006. Wybrane metodologie i metody budowania ontologii. *Informacja w sieci. Problemy. Metody. Technologie*. Wydawnictwo Stowarzyszenia Bibliotekarzy Polskich. 157-207.
- [14] Gómez-Pérez, A., Fernández-López, M. i Corcho, O. 2004. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Verlag.
- [15] Gruber, T.R. i others 1993. A translation approach to portable ontology specifications. *Knowledge acquisition*. 5, (1993), 199–199.
- [16] Gruber, T.R. i others 1995. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*. 43, 5 (1995), 907–928.
- [17] Grüninger, M. i Fox, M.S. 1995. Methodology for the design and evaluation of ontologies. *Workshop on Basic Ontological Issues in Knowledge Sharing (1995)*, 6–1.
- [18] Guarino, N., Oberle, D. i Staab, S. 2009. What Is an Ontology? *Handbook on Ontologies*. 1-17.
- [19] Haase, P., Lewen, H., Studer, R., Tran, T., Erdmann, M., d’Aquin, M. i Motta, E. 2008. The neon ontology engineering toolkit. *Demo Session at WWW2008*. (2008), 21–25.
- [20] Hayes, P.J. 1985. The second naive physics manifesto. *PNAS*. 3, (1985), 4.
- [21] Kalyanpur, A., Parsia, B., Sirin, E., Grau, B.C. i Hendler, J. 2006. Swoop: A web ontology editing browser. *Web Semantics: Science, Services and Agents on the World Wide Web*. 4, 2 (2006), 144–153.
- [22] Lassila, O. i Hendler, J. 2007. Embracing "Web 3.0". *IEEE Internet Computing*. (2007), 90–93.
- [23] Matuszek, C., Cabral, J., Witbrock, M. i DeOliveira, J. 2006. An introduction to the syntax and content of Cyc. *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering (2006)*, 44–49.

- [24] McCarthy, J. 1980. Circumscription—a form of non-monotonic reasoning. *Artificial intelligence*. 13, 1-2 (1980), 27–39.
- [25] Mizoguchi, R. i Kozaki, K. 2009. *Ontology engineering environments. Handbook on Ontologies*. (2009), 315–336.
- [26] NeOn Book - NeOn Project. http://www.neon-project.org/nw/NeOn_Book. Dostęp dnia: 09-10-2010.
- [27] Neon Plugins - NeOn Wiki. http://neon-toolkit.org/wiki/Neon_Plugins. Dostęp dnia: 09-21-2010.
- [28] NeOn Project. <http://www.neon-project.org>. Dostęp dnia: 09-09-2010.
- [29] Noy, N.F., McGuinness, D.L. i others 2001. *Ontology development 101: A guide to creating your first ontology*. SML.
- [30] OBO Foundry - The Open Biological and Biomedical Ontologies. <http://www.obofoundry.org/>. Dostęp dnia: 09-11-2010.
- [31] OCS - Project Website. <http://www.ocs.kask.eti.pg.gda.pl/ocs2/pages/home.jsf>. Dostęp dnia: 05-05-2010.
- [32] ODM 1.0 - Ontology Definition Metamodel. <http://www.omg.org/spec/ODM/1.0/>. Dostęp dnia: 09-12-2010.
- [33] *Ontology Design Patterns ODP*. http://ontologydesignpatterns.org/wiki/Main_Page. Dostęp dnia: 09-18-2010.
- [34] *Protege Ontology Library - Protege Wiki*. http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library. Dostęp dnia: 09-18-2010.
- [35] Robert Trypuz i Paweł Garbacz 2007. Czym jest inżynieria ontologiczna? *Filozofia Nauki*. 3 (2007), 121-140.
- [36] Russel, S. i Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Prentice hall.
- [37] *Semantically-Enabled Knowledge Technologies Portal*. <http://www.sekt-project.com/>. Dostęp dnia: 09-21-2010.
- [38] SOVA - Protege Wiki. <http://protegewiki.stanford.edu/wiki/SOVA>. Dostęp dnia: 09-21-2010.
- [39] Spyns, P., Oberle, D., Volz, R., Zheng, J., Jarrar, M., Sure, Y., Studer, R. i Meersman, R. 2002. *OntoWeb - a semantic Web community portal. Practical Aspects of Knowledge Management. 4th International Conference, PAKM 2002. Proceedings, 2-3 Dec. 2002 (Berlin, Germany, 2002)*, 189-200.
- [40] Staab, S. 2009. *Handbook on Ontologies*. Springer.
- [41] Staab, S. i Studer, R. 2004. *Handbook on ontologies*. Birkhäuser.
- [42] Stevens, R. i Lord, P. 2009. *Application of Ontologies in Bioinformatics. Handbook on Ontologies*. 735-756.

- [43] Studer, R. 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*. 25, 1-2 (1998), 161-197.
- [44] Suárez-Figueroa, M.C. D5. 3.1 NeOn Development Process and Ontology Life Cycle.
- [45] Suárez-Figueroa, M.C. D5. 3.2 Revision and Extension of the NeOn Development Process and Ontology Life Cycle.
- [46] Suárez-Figueroa, M.C., UPM, J.A., Euzenat, J. i Le Duc, C. D5. 4.3. Revision and Extension of the NeOn Methodology for Building Contextualized Ontology Networks.
- [47] Sure, Y., Staab, S. i Studer, R. 2009. *Ontology Engineering Methodology. Handbook on Ontologies*. (2009), 135–152.
- [48] Uschold, M. i Gruninger, M. 1996. *Ontologies: Principles, methods and applications. The Knowledge Engineering Review*. 11, 02 (1996), 93–136.
- [49] Uschold, M. i King, M. 1995. *Towards a methodology for building ontologies. Workshop on Basic Ontological Issues in Knowledge Sharing (1995)*, 275–280.
- [50] Völker, J. 2008. NeOn Deliverable 3.8.1. *Prototype for Learning Networked Ontologies*.

