



KOMPUTEROWE PRZETWARZANIE WIEDZY

**Kolekcja prac 2013/2014
pod redakcją Tomasza Kubika**



SPIS TREŚCI

1	Rozproszone bazy wiedzy	3
1.1.	Ontologie, bazy wiedzy	3
1.2.	Języki formalnej reprezentacji ontologii	5
1.2.1.	Język naturalny	6
1.2.2.	Logika deskrypcyjna	6
1.2.3.	Formalizmy obiektowe	7
1.2.4.	Grafy konceptualne	8
1.2.5.	Resource Description Framework	8
1.2.6.	Ontology Web Language	9
1.2.7.	Podsumowanie	10
1.3.	Cel projektu	10
1.4.	Narzędzia	11
1.5.	Semantic Beer Base	11
	Bibliografia	13

ROZPROSZONE BAZY WIEDZY

B. Kochanowski, M. Krupop

1.1. Ontologie, bazy wiedzy

Tematyka niniejszego projektu obejmuje zapoznanie się z różnymi metodami modelowania rzeczywistości. Znajomość różnych modeli reprezentacji wiedzy będzie stanowiło podstawę do zapoznania się z formalnymi sposobami opisu rzeczywistości. Niestety jest to jednak zadanie niezwykle trudne. Wynika to z dużej złożoności procesów zachodzących w przyrodzie oraz jej skłonności do uwidaczniania niuansów w wielu płaszczyznach. Z tego właśnie powodu próbie modelowania (czy zapisu) poddaje się jedynie pewne fakty lub fragmenty rzeczywistości. Skupienie uwagi wyłącznie na wąskiej dziedzinie lub domenie pozwala uzyskać z nich bardzo dużą liczbę informacji. Jednym ze sposobów takiego zapisu są właśnie ontologie. Służą one do budowania tzw. modeli konceptualnych reprezentujących pewien wąski obszar wiedzy. Opisywane w ten sposób zjawiska są najczęściej bardzo skomplikowane, dlatego głównym zadaniem modeli konceptualnych jest ich znaczące uproszczenie. Podejście to wymaga pewnego sposobu opisu, który umożliwia pełne zrozumienie skomplikowanych mechanizmów zachodzących w rzeczywistości. Niezwykle istotny jest również sposób reprezentacji tych modeli umożliwiający ich automatyczne przetwarzanie przez maszyny bądź specjalnych komputerowych agentów [1].

Istnieje bardzo wiele definicji pojęcia ontologia. Odwołując się do nich można wyróżnić 5 najważniejszych cech charakteryzujących każdą ontologię. Są to [2]:

- konceptualizacja - tworzenie bardzo uogólnionych modeli pewnego wycinka rzeczywistości,
- konsensus - dziedzina lub domena opisywana przy pomocy ontologii musi być prawdziwa i zgodna z przekonaniami osób z nimi związanych - konsensus społeczny,
- ograniczoność ontologii - wynika z ograniczonego zakresu opisywanej wiedzy; dla węższych dziedzin można stosować bardziej rozbudowane ontologie,
- formalny charakter - każda ontologia może zostać wyrażona przy pomocy specjalnych języków reprezentacji - możliwość komputerowego przetwarzania,

1. Rozproszona baza wiedzy

- jawność - jawne definiowanie pojęć z danej dziedziny i relacji między nimi z wykorzystaniem formalnych języków reprezentacji.

Ontologia nie jest pojęciem jednoznacznie zdefiniowanym. Przykładem może być stosunkowo duża liczba definicji zawartych w artykule [1]. Każda z nich bazuje na innej, uzupełniając ją lub poruszając aspekty związane z formalnym sposobem reprezentacji bazy wiedzy. Warto przytoczyć jedną z prostszych definicji, określających ontologię jako „uporządkowany zbiór, składający się ze zbiorów obiektów oraz relacji pomiędzy nimi (binarnych i unarnych)”. Jednym z najważniejszych założeń ontologii jest stworzenie reprezentacji łatwej do zrozumienia zarówno przez ludzi jak i maszyny (komputery).

Jak wspomniano wcześniej ontologia jest jawną oraz formalną metodą reprezentacji pewnego fragmentu rzeczywistości. Każda ontologia składa się z 3 najważniejszych części. Pierwszą z nich są klasy, uważane za centralną część ontologii. Dzięki nim możliwe jest opisywanie pojęć charakterystycznych dla danej dziedziny czy domeny. Jako przykład może tutaj posłużyć ontologia wykorzystana do realizacji zadania projektowego. Wykorzystując klasy opisuje ona najważniejsze pojęcia związane z produkcją piwa. Można wyróżnić tutaj następujące klasy: *piwo*, *style piwa*, *składniki*, *browary*, *wydarzenia*, itp. Każda klasa może posiadać również swoje podklasy, które opisują pojęcia w bardziej szczegółowy sposób, np. klasa *składniki* dzieli się na konkretne typy składników: *zboża*, *chmiel*, *słód*, *drożdże*, *woda*, itp. Idąc dalej w klasie *zboża* możemy wyróżnić *jęczmień* oraz *pszenicę*. Widać zatem wyraźnie, że klasy pozwalają na opisy pojęć lub obiektów należących do danej dziedziny. Warto również wspomnieć, że ontologia ta bazuje na innym słowniku, w którym klasa *piwo* jest podklasą klasy *napój alkoholowy*, a klasa *browar* podklasą klasy *organizacja*. Drugi element stanowią tzw. sloty, które określają właściwości, funkcjonalność lub atrybuty pojęć opisanych klasami oraz konkretne instancje tych klas. Przykładem może być tutaj klasa stylu piwa - *Lager* posiadająca właściwość *madeFrom*, określająca składniki potrzebne do uwarzenia danego stylu piwa. Zgodnie z informacjami z istniejącej ontologii do uwarzenia piwa w stylu *Lager* potrzeba chmielu *Saaz* oraz słodu *Pils*. Dodatkowym elementem uzupełniającym całość ontologii stanowią ograniczenia narzucone na sloty każdej klasy. Przykładowo, składnikiem potrzebnym do uwarzenia danego stylu piwa nie może być klasa *browar*, *wydarzenie*, czy inny *styl piwa*. Jest to szczególnie użyteczne w przypadku gdy pewna własność danej klasy jest ograniczona pewnym przedziałem. Ograniczenia chronią definiowaniem niepoprawnych wartości dla instancji określonej klasy [3].

Dzięki takiej budowie ontologie służą jako pewnego rodzaju szkielet bądź słownik, służący do definiowania konkretnych instancji danej klasy. Autorki artykułu [3], podają, iż nie istnieje wyraźna granica między ontologią a pojęciem baza wiedzy. Niemniej jednak, bazując na ogólnej definicji, terminem bazy wiedzy można objąć ontologię wraz ze wszystkimi instancjami klas korzystających z opisu umieszczonego w tej ontologii. Niniejszy projekt związany jest z tematyką rozproszonych baz wiedzy. Należy przez to rozumieć pewnego rodzaju odseparowanie ontologii oraz wszystkich instancji bazujących na tej ontologii. Według założeń projektowych, stworzona ontologia powinna znajdować się na ogólnie-

dostępnym serwerze w sieci. Korzystając z tego opisu, przedsiębiorstwa browarnicze mogą tworzyć instancje warzonego przez siebie asortymentu, tworząc w ten sposób firmowe bazy produktów. Rozproszone bazy wiedzy dają możliwość wielokrotnego wykorzystywania swoich zasobów, których umowna reprezentacja umożliwia ich komputerowe przetwarzanie.

Podsumowując, proces tworzenia ontologii wymaga zdefiniowania klas reprezentujących pojęcia z danej dziedziny w pewnej ustalonej hierarchii. Najpierw zaczyna się od pojęć najbardziej ogólnych, których podklasy będą zawierać bardziej szczegółowe elementy. Proces ten wieńczy stworzenie slotów interpretowanych jako właściwości lub atrybuty danych wraz z ich ograniczeniami. Tworzenie baz wiedzy jest procesem bardziej złożonym, gdyż wymaga stworzenia instancji klas opisanych przy pomocy ontologii.

1.2. Języki formalnej reprezentacji ontologii

W związku z bardzo szybkim rozwojem techniki komputerowej oraz jej stosowania w wielu dziedzinach życia, pojawiła się konieczność komputerowego przetwarzania wiedzy przechowywanej w specjalnych repozytoriach. Aby wiedza ta mogła być przesyłana i przetwarzana istnieje konieczność modelowania wiedzy w taki sposób, aby mogła ona w bardzo prosty i zrozumiały sposób zostać zapisana za pomocą pewnej formalnej reprezentacji. W tym celu powstało wiele metod lub języków opisu, dzięki którym wiedza o określonej dziedzinie lub domenie zawarta w bazie wiedzy, może zostać przekształcona w użyteczną informację [4].

Bardzo ważną własnością języków używanych do opisu ontologii jest stopień ich formalności. Określa on pewnego rodzaju sposób, semantykę wyrażania ontologii przez ten język. Jak podają M. Uschold i M. Gruninger [5] stopień formalności określający dany język reprezentacji jest bardzo ściśle związany ze specyfiką procesu wymagającego przetwarzania ontologii. Przykładem może być komunikacja międzyludzka. Autorzy zaznaczają, iż w takim przypadku wykorzystywany jest nieformalny sposób reprezentacji ontologii. W wielu przypadkach jest on wystarczający, aby w poprawny sposób wyczuć intencję drugiej osoby. Zupełnie inna sytuacja ma miejsce w przypadku przetwarzania ontologii przez komputery lub wirtualnych agentów. Niestety nie posiadają oni tak rozbudowanej inteligencji jak ludzie, dlatego do poprawnego wnioskowania wymagają bardzo formalnej reprezentacji. Zatem formalny sposób reprezentacji jest zarezerwowany dla ontologii poddawanej komputerowemu przetwarzaniu. Taka postać reprezentacji jest najczęściej bardzo skomplikowana i trudna do interpretacji przez człowieka. Dlatego najwygodniejszym językiem reprezentacji ontologii dla ludzi jest język pozbawiony formalizmów - język naturalny.

Uschold oraz Gruninger wprowadzają również krótki podział tych języków w zależności od stopnia ich formalizacji. Wyróżniają oni [5], [6]:

- języki nieformalne - brak jasnych reguł co do reprezentacji ontologii; języki charakteryzujące się brakiem jasności przekazu oraz wprowadzające wiele dwuznaczności,

1. Rozproszone bazy wiedzy

- języki semi-nieformalne - bardziej ograniczona i ustrukturyzowana wersja języków nieformalnych,
- języki semi-formalne - stanowią grupę tzw. sztucznych języków reprezentacji, dostosowanych do wymagań języków formalnych,
- języki formalne - języki z bardzo szczegółowo zdefiniowaną semantyką, teorematami oraz dowodami.

W niniejszym podrozdziale opisany zostanie krótki przegląd wybranych języków służących jako formalny sposób reprezentacji tworzonych ontologii.

1.2.1. Język naturalny

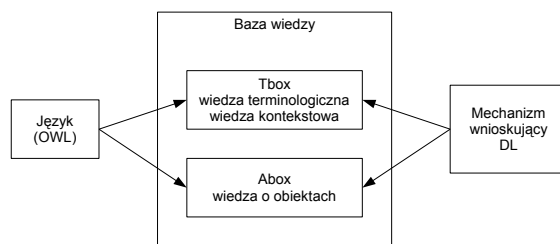
W przypadku modelowania rzeczywistości z wykorzystaniem języka naturalnego, wiedza reprezentowana jest za pomocą zdań lub różnego rodzaju wyrażeń. Bardzo dobrym przykładem takiego modelowania są instrukcje obsługi urządzeń. Dzięki zawartej tam wiedzy można łatwo dokonać szybkiej diagnozy wadliwie działającego urządzenia. Zaletą tego modelu jest możliwość bardzo szybkiego przekazu wiedzy. Niestety nadaje się on tylko do opisu bardzo prostego modelu rzeczywistości. Bardziej złożony fragment rzeczywistości będzie wymagał większej ilości struktur potrzebnych do jej zapisania. Reprezentacja wiedzy przy pomocy języka naturalnego posiada jedno, bardzo istotne ograniczenie. Mianowicie taka forma reprezentacji będzie zrozumiała jedynie wśród osób jednakowo interpretujących dany język oraz symbole. Powoduje to, że sposobu przedstawiania wiedzy przy pomocy języka naturalnego nie można uznać za uniwersalną formę reprezentacji [3].

1.2.2. Logika deskrypcyjna

Logika deskrypcyjna DL (ang. *Description Logic*) jest formalnym językiem reprezentacji ontologii opartym na założeniach związanych z logiką predykatów oraz ideą sieci semantycznych. W przypadku tego języka reprezentacji wiedza o danej dziedzinie reprezentowana jest przy pomocy następujących elementów: pojęć, ról oraz osobników. Pojęcia mogą być interpretowane jako pewien rodzaj odwzorowania rzeczywistych obiektów lub zdarzeń z opisywanej dziedziny. Z kolei osobnikiem będziemy nazywać konkretny przykład pojęcia. Natomiast rolą będziemy określać relację dwuargumentową zachodzącą między danymi osobnikami [6].

Strukturę systemu opartego o reprezentację wiedzy z wykorzystaniem logiki deskrypcyjnej przedstawia rysunek 1.1.

1.2. Języki formalnej reprezentacji ontologii



Rys. 1.1: Reprezentacja wiedzy przy pomocy logiki deskrypcyjnej [7].

Reprezentacja wiedzy oparta na logice deskrypcyjnej składa się z dwóch zasadniczych komponentów. Są to [7]:

- Komponent TBox - reprezentuje zestaw pojęć, które można podzielić na dwie grupy: pojęcia pierwotne oraz pojęcia zdefiniowane poprzez określony zestaw osobników i ról (relacji między osobnikami). Obiektami tego komponentu są klasy, dla których wyznacza on pewne dopuszczalne reprezentacje. Ponadto komponent TBox umożliwia stosowanie mechanizmów wnioskowania opartego na klasach.
- Komponent ABox - reprezentuje pewien skończony zbiór twierdzeń. Obiektami tego komponentu są konkretne instancje klas. Podobnie jak TBox ogranicza liczbę dopuszczalnych interpretacji dla swoich obiektów.

1.2.3. Formalizmy obiektowe

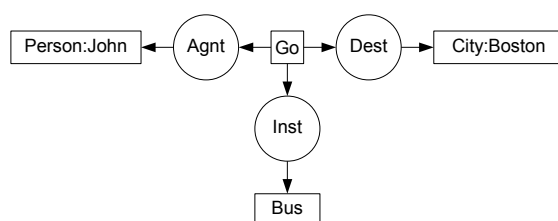
Formalizmy obiektowe pozwalają na reprezentowanie wiedzy z danej dziedziny w postaci wirtualnych modeli, odzwierciedlających rzeczywiste obiekty lub zdarzenia. Powstałe w ten sposób modele noszą nazwę klas. Określają one szereg właściwości lub atrybutów charakterystycznych dla wszystkich obiektów należących do danej klasy. Formalizmy obiektowe są ściśle związane z tzw. hierarchizacją klas. Wynika stąd, iż każda klasa może posiadać zbiór podklas, powstałych w wyniku jej dziedziczenia, podczas gdy sama powstała z klasy nadrzędnej. Własnością klas nadrzędnych jest fakt, iż dzięki hierarchizacji pozwalają one uogólnić pewne zbiory obiektów rzeczywistych. Podczas gdy jej klasy pochodne definiują bardziej szczegółowy podział. Z formalizmami obiektowymi jest również związane pojęcie instancji klasy. Są to konkretne obiekty o właściwościach i atrybutach zgodnych ze wzorcem, z którego powstały [6].

Przykładem języka umożliwiającego opis formalizmów obiektowych jest UML (Unified Modeling Language). Pozwala on na ich graficzną reprezentację w postaci tzw. diagramu klas - zobrazowanie statycznego związku pomiędzy klasami. Jak zaznacza A. Sobczak, UML posiada pewne ograniczenie, które nie spełnia wymogów co do formalnego sposobu reprezentacji ontologii. Niestety język ten nie posiada formalnej semantyki, czyli szczegółowo zdefiniowanych reguł zapisu ontologii przy pomocy tego języka [6].

1. Rozproszone bazy wiedzy

1.2.4. Grafy konceptualne

Grafy konceptualne są kolejnym przykładem formalnego sposobu reprezentacji ontologii. Ich idea powstała z połączenia sieci semantycznych oraz grafów Pierce'a (od nazwiska amerykańskiego logika Charlesa S. Pierce'a). Często są one przedstawiane jako rozbudowana forma logiki predykatów [6]. Sposób reprezentacji wiedzy przy pomocy tego języka prezentuje rysunek 1.2.



Rys. 1.2: Przykład grafu konceptualnego [8].

Grafy konceptualne są bardzo wygodnym narzędziem pozwalającym na przetwarzanie języka naturalnego. Potwierdza to powyższy graf prezentujący wiedzę ze zdania „John jedzie autobusem do Bostonu”, pochodzącego z języka naturalnego.

Taki sposób reprezentacji wiedzy pozwala za pomocą prostokątów przedstawiać istniejące obiekty lub zdarzenia z pewnego fragmentu opisywanej rzeczywistości. Obiekty te, podobnie jak poprzednio, nazwane zostały konceptami lub pojęciami. Natomiast okręgi określają relacje zachodzące pomiędzy tymi obiektami. W prezentowanym przykładzie *Agnt* oznacza agenta czyli głównego wykonawcę czynności, *Inst* określa narzędzie wykorzystane przy danej czynności oraz *Dest* określa jej cel [8].

Ogólnie rzecz biorąc sieci semantyczne służą do formalnej reprezentacji pewnego wąskiego fragmentu rzeczywistości. W tym celu wykorzystują one dwa rodzaje obiektów. Pierwszym z nich są koncepty reprezentujące pewne fizyczne byty, atrybuty lub zdarzenia charakterystyczne dla opisywanej dziedziny. Natomiast drugi typ określa relacje wyrażające zależności między obiektami. Związki między obiektami można określić mianem semantyki grafów konceptualnych. Opisywany język reprezentacji wykorzystuje również elementy logiki, które wyznaczają czy określony koncept jest elementem opisywanej rzeczywistości czy też nie. W tym celu używane są kwantyfikatory: ogólny oraz egzystencjalny, a także operatory: negacji, alternatywy i koniunkcji [8],[6].

1.2.5. Resource Description Framework

RDF jest przykładem języka o bardzo rygorystycznym stopniu formalizacji. Język ten opracowany został w celu stworzenia formalnego standardu dającego możliwość automatycznej interpretacji i przetwarzania wiedzy przez systemy komputerowe i wirtualnych agentów. Warto dodać, iż standard ten opracowany

1.2. Języki formalnej reprezentacji ontologii

został przez konsorcjum W3C [9].

Reprezentacja ontologii przy pomocy RDF wymaga użycia pewnego formalnego sposobu zapisu danych. W tym celu wykorzystuje się tzw. metajęzyki. Jedną z najprostszych definicji określa metajęzyki jako: „języki służące do definiowania innych języków” [10]. Jednym z najpopularniejszych i często stosowanych metajęzyków jest XML. Jest to język bazujący na znacznikach pozwalających na reprezentację danych w postaci specjalnie zdefiniowanej struktury. Obecnie większość języków lub standardów służących do opisu ontologii zbudowanych jest właśnie na bazie XML [6].

Główna idea RDF zakłada możliwość opisu pewnego bardzo wąskiego fragmentu rzeczywistości w postaci tzw. trójek lub trypletów. Zaliczamy do nich [9]:

- podmiot/klasa - określa zasoby czyli pewne obiekty lub zdarzenia występujące w modelowanej dziedzinie,
- predykat - może być interpretowany jako pewne własności lub atrybuty danego obiektu,
- obiekt - jest to wartość danego predykatu lub inaczej wartość określonego atrybutu obiektu.

Warto dodać, iż język RDF posiada bardzo ciekawy mechanizm identyfikacji trypletów. W tym celu wykorzystuje on standard URI (Uniform Resource Identifier). URI jest standardem wykorzystywanym do identyfikacji zasobów w sieciach komputerowych [9].

Jak wspomniano wcześniej RDF, jest językiem bazującym na języku znaczników XML. Pozwala to na reprezentację wiedzy w postaci struktury drzewiastej. RDF posiada niezwykle użyteczny mechanizm dziedziczenia umożliwiający praktycznie nieograniczoną możliwość rozszerzania klas oraz predykatów. W tym celu stosuje się polecenia „rdfs:subClassOf” oraz „rdfs:subPropertyOf” [9].

W kontekście języka RDF warto w kilku zdaniach odnieść się do standardu RDF Schema. RDFS jest językiem reprezentacji wiedzy opartym na RDF. Nie definiuje on nowych pojęć ale tworzy struktury zwane słownikami RDF. Są to różnego rodzaju mechanizmy służące do opisu klas oraz predykatów (właściwości, atrybutów). Dzięki RDFS możliwe jest zdefiniowanie zbioru klas i predykatów stosowanych łącznie do opisu określonego obiektu czy mechanizmu hierarchizacji klas. Definiując w tym języku klasę Samochód, można ją zdefiniować jako klasę podrzędną klasy pojazd kołowy, która stanowi podklasę klasy pojazd. Ta własność sprawia, że RDFS jest językiem bardzo zbliżonym do obiektowych języków programowania. Ponadto RDFS umożliwia tworzenie zależności pomiędzy klasami oraz predykatami, co znacznie usprawnia mechanizm wnioskowania na podstawie dostarczonej bazy wiedzy. Na koniec warto wspomnieć iż RDF Schema stanowi bazę dla języka OWL [11].

1.2.6. Ontology Web Language

Ostatnim prezentowanym przykładem języka stosowanego do formalnej reprezentacji ontologii jest język OWL. Stanowi on rozszerzenie języków RDF i RDF Schema. Należy on do języków o bardzo wysokim stopniu formalizacji. Ozna-

1. Rozproszone bazy wiedzy

cza to, że taki sposób reprezentacji wiedzy umożliwia jej interpretację oraz automatyczne przetwarzanie przez systemy komputerowe lub wirtualnych agentów. Podobnie jak RDF, standard ten, został opracowany przez konsorcjum W3C [12].

Składnia języka OWL jest zgodna ze standardem XML, natomiast jego semantyka bazuje na tzw. logice opisowej. Najprościej definiując terminem logiki opisowej można określić zbiór języków reprezentacji ontologii, umożliwiających (w różnym stopniu) przeprowadzenie procesu wnioskowania. Ponadto OWL jest znacznie bardziej rozbudowany niż RDF czy RDFS, gdyż posiada bogatszą składnię służącą do budowy klas, przypisywania im określonych własności, definiowania ich wzajemnej relacji, a także posiada metody pozwalające na reprezentację tego samego fragmentu rzeczywistości przy pomocy różnych wersji ontologii. Warto wspomnieć, iż istnieją 3 różne odmiany tego języka, charakteryzujące się różnym stopniem funkcjonalności [6]. Wyróżnia się [12]:

- OWL Lite - ograniczenia w stosunku do konstruktorów tego języka (brak klas wyliczeniowych czy rozłączności), łatwość implementacji, ograniczona ekspresyjność,
- OWL DL - mniejsze ograniczenia konstruktorów języka, zapewnia efektywne wnioskowanie, utrata kompatybilności z RDF
- OWL Full - dostępność wszystkich konstrukcji języka OWL, możliwość wykorzystywania konstrukcji z języków RDF i RDFS, ograniczone wsparcie dla mechanizmu wnioskowania.

1.2.7. Podsumowanie

W niniejszym rozdziale przedstawiono kilka wybranych formalnych metod reprezentacji ontologii. Noszą one nazwę języków reprezentacji. Jednym z najważniejszych kryteriów podziału tych języków jest stopień formalizacji. Jego stopień określa stopień formalizacji semantyki. Im wyższy stopień formalizacji tym większa podatność reprezentacji na jej automatyczne przetwarzanie przez programy komputerowe. Niestety taki sposób przedstawiania wiedzy jest bardzo nieczytelny dla ludzi. Niski stopień formalizacji związany jest z językiem naturalnym używanym przez ludzi. Ze względu na sporą liczbę niejasności oraz tendencji do dwuznaczności nie nadaje się on do komputerowego przetwarzania.

Wspólną cechą wielu języków reprezentacji jest sposób reprezentacji ontologii w postaci, tzw. trypletów czyli klas, predykatów (własności lub atrybutów) klas oraz wartości tych atrybutów wraz z ograniczeniami. Zaletą takiego sposobu reprezentacji jest stosunkowo duża łatwość dokonywania zmian, a także możliwość jej wielokrotnego przetwarzania w przypadku różnych zadań.

1.3. Cel projektu

Celem projektu jest wykorzystanie dostępnej bazy wiedzy dotyczącej wyspecjalizowanej dziedziny, w tym wypadku informacji opisujących różne style piwa. Baza wiedzy, znaleziona przy pomocy *Swoogle*, została wykorzystana jako zestaw pojęć do opisu ww. dziedziny. Dysponując tym modelem, można wykorzystać

go, importując do własnej bazy wiedzy, która na tej podstawie opisuje własne instancje. Zaletą takiego rozwiązania jest stworzenie architektury rozproszonej. Baza główna może znajdować się w dowolnym miejscu w sieci, wystarczy, że będzie dostępna. Wykorzystując tę bazę, dowolna firma produkująca piwo, będzie w stanie na jej podstawie stworzyć własną bazę, opisującą własne wyroby, tworząc instancje danych stylów piwa. Dysponując niezbędnymi elementami, czyli słownikiem oraz bazami firmowymi, wykorzystującymi ten słownik, można przystąpić do wykorzystania tych rozproszonych informacji w jednym miejscu.

Projekt ten można zrealizować na dwa sposoby. Pierwszym sposobem jest wykorzystanie zapytań w języku SPARQL z poziomu programu i uzyskiwanie w ten sposób potrzebnych informacji o właściwościach poszczególnych stylów piwa oraz ich instancjach. Podejście to, nie wymaga pobierania tych baz do pamięci komputera, gdyż zapytanie zwraca jedynie wymagane informacje. Można również dokonać agregacji poszczególnych grafów reprezentujących bazy wiedzy do jednej struktury grafowej w programie. Do wykonania takiej operacji służy Apache Jena - framework napisany w języku Java. Dopiero na tej zintegrowanej strukturze można odczytywać potrzebne informacje.

1.4. Narzędzia

W celu realizacji wybranego zadania projektowego wykorzystano następujące narzędzia:

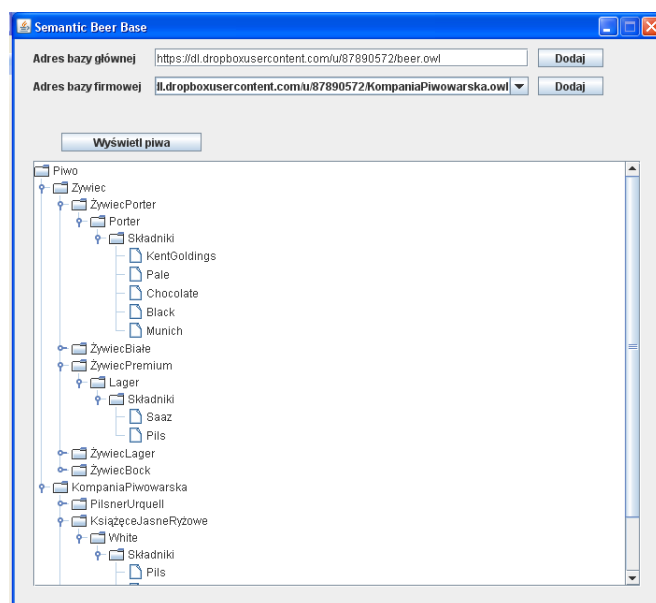
- Protege([www.http://protege.stanford.edu/](http://protege.stanford.edu/)) – edytor służący do tworzenia ontologii,
- Apache Jena ([www.http://jena.apache.org/](http://jena.apache.org/)) – jest to API umożliwiające operowanie na bazach wiedzy, ich agregację, wyświetlanie potrzebnych informacji,
- Swoogle ([www.http://swoogle.umbc.edu/](http://swoogle.umbc.edu/)) - wyszukiwarka pozwalająca na znalezienie dokumentów semantycznego Webu na podstawie słów kluczowych; dodatkowo z dokumentów tych wyodrębniane są metadane co służy do określenia relacji między nimi.[10].

1.5. Semantic Beer Base

W ramach projektu powstał program Semantic Beer Base, który umożliwia wczytywanie słownika oraz baz firmowych zawierających instancje różnych stylów piwa. Program, za pomocą Jeny łączy poszczególne bazy firmowe w jeden graf, a następnie wyświetla wszystkie te instancje, dodając właściwości reprezentowanego przez nie stylu piwa. Informacje te wyświetlane są za pomocą struktury drzewiastej. Przykład działania aplikacji, wyświetlającej wczytane informacje przedstawione zostały na rys. 1.3.

Program wyświetla instancje piw ontologii firmowych i dla każdej z nich dodaje nazwę stylu który reprezentuje dana instancja. Możliwe jest również wyświetlenie atrybutów związanych z danym stylem. W tym przypadku są to składniki, z

1. Rozproszone bazy wiedzy



Rys. 1.3: Widok działającej aplikacji

których można uwarzyć piwo w danym stylu. Informacje te pobierane są z głównej ontologii, a dokładniej, jest to, wspomniany już wcześniej, atrybut *madeFrom*.

Literatura

- [1] W. Gliński. Ontologie. próba uporządkowania terminologicznego chaosu. *Od informacji naukowej do technologii społeczeństwa informacyjnego: praca zbiorowa / pod red. Barbary Sosińskiej-Kalaty i Marii Przastek-Samokowej*. Wydaw. SBP, Warszawa, 2005.
- [2] Marcin Roszkowski. Do czego potrzebne są nam ontologie? charakterystyka funkcjonalna ontologii jako narzędzi reprezentacji wiedzy. 2013.
- [3] N. Noy, D. McGuinness. What is an ontology and why we need it? *Ontology Development 101: A Guide to Creating Your First Ontology*. http://liris.cnrs.fr/alain.mille/enseignements/Ecole_Centrale/.
- [4] G. Paquette. *Visual Knowledge Modeling for Semantic Web Technologies: Models and Ontologies*. Information Science Reference, 2010.
- [5] M. Uschold, M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review, Volume 11, Number 2*. 2014.
- [6] A. Sobczak. Analiza wybranych języków stosowanych do budowy ontologii. *Polskie Stowarzyszenie Zarządzania Wiedzą - Studia i Materiały nr 7*. Warszawa, 2006.
- [7] J. Józefowska. Ontologie. logiki deskrypcyjne. <http://www.cs.put.poznan.pl/jjozefowska/wyklady/ai/Ontologie.pdf>.

- [8] J. F. Sowa. Conceptual graphs. *Handbook of Knowledge Representation*. 2008.
- [9] World Wide Web Consortium. *RDF 1.1 Concepts and Abstract Syntax*, W3C Recommendation, 2004.
- [10] W. Gliński. Języki i narzędzia do tworzenia i wyszukiwania ontologii w kontekście semantycznego weba. *Od informacji naukowej do technologii społeczeństwa informacyjnego: praca zbiorowa / pod red. Barbary Sosińskiej-Kalaty i Marii Przastek-Samokowej*. Wydaw. SBP, Warszawa, 2005.
- [11] World Wide Web Consortium. *RDF Schema 1.1*, W3C Recommendation, 2004.
- [12] World Wide Web Consortium. *OWL Web Ontology Language Overview*, W3C Recommendation, 2004.