

Rewolucja w sieci

Cel: Ta sekcja obejmuje podstawowe pojęcia związane z siecią, w tym historię sieci, różne etapy ewolucji sieci itp.

Podstawowe pojęcia

Był czas, kiedy uczeni i naukowcy pokonywali kilometry, aby dotrzeć do biblioteki publicznej, która była jedynym skarbcem wiedzy. Młodsze pokolenie jest na tyle sprytnie, by zadać pytanie „Dlaczego nie wygooglują?” To jest XXI wiek, o którym mówimy. Dzisiejszy świat kręci się wokół internetu. Ale ciemną stroną tej prawdy jest to, że bardzo niewielu z nas wie, co dokładnie dzieje się w świecie World Wide Web. Spróbujmy więc zrozumieć tło sieci. Oto niektóre z podstawowych terminów związanych z Internetem:

World Wide Web: World Wide Web (w skrócie WWW lub W3, powszechnie znany jako Web) to system powiązanych ze sobą dokumentów hipertekstowych, do których dostęp uzyskuje się przez Internet. Za pomocą przeglądarki internetowej można przeglądać strony internetowe, które mogą zawierać tekst, obrazy, pliki wideo i inne multimedia oraz nawigować między nimi za pomocą hiperłączy.

Hiperłącze: hiperłącze to najbardziej podstawowy element składowy sieci WWW. Jest to łącze z jednego dokumentu, obrazu, słowa lub strony internetowej do innego w sieci.

Hipertekst: W 1965 r. Ted Nelson wymyślił termin „hipertekst” na określenie złożonej, zmieniającej się i nieokreślonej struktury plików. Hipertekst to dowolny blok treści zawierający hiperłącza do innych dokumentów, obrazów lub treści multimedialnych.

Hypertext Transfer Protocol: Hypertext Transfer Protocol, popularnie w skrócie HTTP, to język, którego komputery używają do komunikacji dokumentów hipertekstowych przez Internet.

Uniform Resource Locator: Uniform Resource Locator, popularnie w skrócie URL, to globalny adres dokumentów i innych zasobów w sieci WWW.

Hypertext Markup Language: HTML to skrócona forma Hypertext Markup Language. Jest to język używany do tworzenia dokumentów elektronicznych, zwłaszcza stron w sieci WWW, które zawierają połączenia zwane hiperłączami do innych stron. Każda strona internetowa, którą widzisz w Internecie, zawiera kod HTML, który wyświetla tekst / obrazy w łatwym do odczytania formacie. Bez HTML przeglądarka nie będzie miała żadnego układu strony i dlatego wyświetli tylko zwykły tekst bez formatowania.

Przeglądarka internetowa: Przeglądarka internetowa (powszechnie określana jako przeglądarka) to aplikacja służąca do pobierania, prezentowania i przeglądania zasobów informacyjnych w sieci WWW.

Serwer WWW: serwer WWW to dowolny serwer internetowy, który odpowiada na żądania HTTP w celu dostarczenia treści i usług. Może to być sprzęt lub oprogramowanie.

Internet: Internet to pojedyncza ogólnosiwiatowa sieć komputerowa, która łączy inne sieci komputerowe, w których znajdują się usługi dla użytkowników końcowych, takie jak witryny sieci Web lub archiwa danych. W ten sposób umożliwia wymianę danych i innych informacji.

W odpowiedzi na uruchomienie Sputnika, Departament Obrony USA powołał Advanced Research Projects Agency (ARPA), która ostatecznie skupiła się na sieciach komputerowych i technologiach komunikacyjnych. Tak więc ARPANET był pierwotnie eksperymentem przeprowadzonym w celu ustalenia, w jaki sposób wojsko USA może utrzymywać łączność w przypadku możliwego uderzenia nuklearnego. Ale później ARPANET stał się cywilnym eksperymentem, który połączył komputery

mainframe uniwersytetów do celów akademickich. Oryginalny ARPANET wyrósł na Internet. Internet opierał się na założeniu, że będzie istniało wiele niezależnych sieci o raczej dowolnym projekcie, począwszy od ARPANET jako pionierskiej sieci przełączania pakietów. Dziś internet stał się magazynem bilionów komputerów osobistych, rządowych i komercyjnych połączonych ze sobą kablami i sygnałami bezprzewodowymi.

Aplikacja internetowa: aplikacja internetowa to dowolna aplikacja, która znajduje się na serwerze, ale jest przeznaczona do użytku przez ludzi. Aplikacje internetowe wykorzystują strony internetowe jako warstwę prezentacji. Interaktywność użytkownika (graficzny interfejs użytkownika) odbywa się za pośrednictwem stron internetowych, ale dane są przechowywane i (w większości) przetwarzane na serwerze. Strony internetowe mogą być statyczne, dynamiczne lub aktywne:

Statyczne strony internetowe

Statyczne strony internetowe zawierają tę samą wstępnie zbudowaną zawartość za każdym razem, gdy strona jest ładowana. Standardowe strony HTML to statyczne strony internetowe. Zawierają kod HTML, który definiuje strukturę i zawartość strony internetowej. Za każdym razem, gdy ładowana jest strona HTML, wygląda tak samo. Możesz sprawdzić, czy strona jest statyczna, czy dynamiczna, patrząc na rozszerzenie pliku strony w adresie URL. Jeśli jest to „.htm” lub „.html”, strona prawdopodobnie jest statyczna.

Dynamiczne strony internetowe

Dynamiczny oznacza zmianę lub ożywienie. Dynamiczna strona internetowa po stronie serwera to strona internetowa, której konstrukcja jest kontrolowana przez serwer aplikacji, przetwarzający skrypty po stronie serwera. Strony internetowe, takie jak PHP, ASP i JSP, to dynamiczne strony internetowe. Strony te zawierają kod „po stronie serwera”, który umożliwia serwerowi generowanie unikalnej zawartości za każdym razem, gdy strona jest ładowana. Na przykład serwer może wyświetlać aktualną godzinę i datę na stronie internetowej. Wiele stron dynamicznych używa kodu po stronie serwera w celu uzyskania dostępu do bazy danych i generowania treści na podstawie informacji przechowywanych w bazie danych. Witryny, które generują strony internetowe na podstawie informacji z baz danych, są często nazywane witrynami opartymi na bazach danych. Jeśli rozszerzenie pliku to „.php”, „.asp” lub „.jsp”, strona jest najprawdopodobniej dynamiczna.

Aktywne strony

Aktywne strony są bardziej dynamiczne niż „dynamiczne” strony internetowe. Interakcja wewnętrzna ma miejsce po stronie klienta i nie jest zależna od serwera do interakcji.

Usługi internetowe: Usługi sieci Web to aplikacje serwerowe, do których można uzyskać dostęp przez sieć WWW za pośrednictwem protokołu HTTP, ale są one przeznaczone głównie do interakcji z innymi programami. Usługi internetowe to składniki aplikacji lub „biblioteki”, z których mogą korzystać inne aplikacje. Po wdrożeniu usługi sieci Web inne aplikacje mogą wykrywać i wywoływać wdrożoną usługę.

Uwaga: Ludzie powszechnie myślą, że Internet i „WWW” to to samo. Ale są to dwie różne technologie, które są ze sobą częściowo powiązane. Internet to sieć sieci, które łączą ze sobą miliony komputerów na całym świecie, tworząc sieć, w której każdy komputer może komunikować się z dowolnym innym komputerem, o ile są one połączone z Internetem. WWW to sposób uzyskiwania dostępu do informacji za pośrednictwem Internetu.

Jak to się wszystko stało

World Wide Web (WWW) został wynaleziony przez Tima Bernersa-Lee, brytyjskiego informatyka w 1990 roku. Przed wynalezieniem WWW pojawiła się kolejka wynalazków technicznych, które ostatecznie doprowadziły do wynalezienia WWW. W 1945 roku Vannevar Bush napisał w „Atlantic Monthly” o rozszerzeniu pamięci zwanym „Memex”, które było urządzeniem fotoelektryczno-mechanicznym, które łączyło dokumenty na mikrofilmach. W 1962 roku Doug Engelbart opracował NLS, czyli „system online” do przeglądania i edycji informacji. W tym czasie wynalazł mysz komputerową. W 1965 roku Ted Nelson ukuł termin hipertekst dla złożonej, zmieniającej się, nieokreślonej struktury plików. Tim Berners-Lee z CERN w Szwajcarii napisał projekt oprogramowania o nazwie INQUIRE. Był to prosty program hipertekstowy, który miał niektóre z tych samych pomysłów, co sieć i sieć semantyczna, ale różniły się pod kilkoma ważnymi względami. Łącząc prace Vannevara Busha, Teda Nelsona i Douga Engelbarta, Tim Berners-Lee napisał Hypertext Transfer Protocol (HTTP). Wdrożył również schemat lokalizacji dokumentów. Zgodnie ze schematem każdemu dokumentowi przypisano Universal Resource Locator, czyli adres URL, który służył jako jego adres. Pod koniec 1990 roku Berners-Lee napisał pierwszą przeglądarkę lub program kliencki do pobierania i przeglądania dokumentów znanych jako www. Bezpośrednimi dwoma rezultatami Tima Berners-Lee po stronie www były oprogramowanie serwera WWW i HTML. Umieszczając wszystkie te komponenty we właściwym miejscu, w 1991 roku udostępnił swoją przeglądarkę i oprogramowanie serwera WWW w Internecie. Krótko mówiąc, to jest historia „www”.

Działanie serwera WWW

Podajmy krótki przykład, aby zrozumieć działanie serwera WWW. Załóżmy, że chcesz odwiedzić Bookboon.com. Więc wpisujesz adres URL odpowiadający Bookboon.com w pasku adresu i naciśnij klawisz Enter. Bez względu na to, gdzie na świecie znajduje się żądana strona, pojawia się ona przed Tobą na ekranie w ułamku sekundy. Ta wykonywana czynność jest podstawowym sposobem realizacji działania sieci. Wyjaśniając to w jednym zdaniu, można powiedzieć, że akcja jest inicjowana przez komputer kliencki, na którym działa przeglądarka internetowa, poprzez żądanie strony. Serwer lokalizuje stronę i odsyła ją z powrotem do klienta. W ten sposób odpowiadając na żądanie

Ewolucja sieci

Zawsze, gdy próbuję wyjaśnić swoim współpracownikom sieć semantyczną, pierwsze pytanie, jakie otrzymuję, brzmi: „W takim razie czym jest sieć 1.0?” Niewielu z nas zdaje sobie sprawę z ewolucyjnej hierarchii. Sieć przeszła ogromne zmiany, zanim otrzymała obecną formę. Na początku spróbujmy zrozumieć ewolucję sieci.

Web 1.0

Pierwotną formą sieci była Web 1.0. Web 1.0 został wymyślony przez Tima Bernersa-Lee. To była platforma tylko do odczytu. Zgodnie z filozofią Web 1.0 firmy tworzą aplikacje, które użytkownicy mogą pobierać, ale nie widzą, jak ta aplikacja działa. Na przykład Netscape Navigator był zastrzeżoną przeglądarką internetową ery Web 1.0. Załóżmy słownik online. Ma to na celu dostarczenie nam znaczenia mnóstwa słów. Ma dane statyczne. Użytkownik może go używać tylko do czytania, ale nie może współtworzyć. To najlepszy przykład Web 1.0. Wadą Web 1.0 jest to, że reprezentuje jednokierunkową komunikację, w której użytkownicy nie mogą współtworzyć sieci. To zmusiło świat do przejścia na Web 2.0.

Web 2.0

Tradycyjny Web 1.0 przeszedł ostatnio transformację i przekształcił się w Web 2.0, w którym nacisk kładzie się na folksonomię i zbiorową inteligencję. Wszystko, co jest dziś znane w świecie sieci, to Web

2.0. Począwszy od Facebooka po YouTube, wszystko jest Web 2.0. Krótko mówiąc, można nazwać obecną sieć Web 2.0. W Web 2.0 użytkownicy nie tylko czytają informacje z Internetu, ale także udostępniają je w sieci, aby dzielić się nimi z innymi. Na przykład na Facebooku możesz pisać swoje opinie, przysyłać zdjęcia i tak dalej. Druga generacja sieci World Wide Web koncentruje się na zdolności ludzi do współpracy i udostępniania informacji online. Web 2.0 to interaktywna sieć. Dlatego nazywa się go siecią do odczytu / zapisu. Cechy Web 2.0 są następujące:

*Możliwość współdzielenia poglądów: użytkownicy sieci mogą wносить swój wkład w Web 2.0. Na przykład, korzystając z formularza online, odwiedzający może dodać informacje do stron Amazon, które przyszli odwiedzający będą mogli przeczytać.

*Używanie stron internetowych do łączenia się z ludźmi: serwisy społecznościowe, takie jak Facebook i MySpace, są popularne, ponieważ ułatwiają użytkownikom znajdowanie się i utrzymywanie kontaktu.

*Szybkie i wydajne sposoby udostępniania treści: YouTube jest doskonałym przykładem. Członek YouTube może stworzyć film i przesłać go do serwisu, aby inni mogli go obejrzeć.

*Nowe sposoby uzyskiwania informacji: mamy niezliczoną liczbę witryn z wiadomościami, na których można znaleźć informacje. Na przykład Wikipedia zawiera szczegółowe informacje o prawie wszystkim na świecie.

*Rozszerzanie dostępu do Internetu: Obecnie ludzie uzyskują dostęp do Internetu nie tylko za pośrednictwem komputera, ale także telefonów komórkowych, tabletów itp.

Typowe cechy aplikacji Web 2.0:

*Na zawartość ma wpływ użytkownik.

*Treści są często generowane przez użytkownika.

*Aplikacje wykorzystują sieć jako platformę.

*Popularne trendy obecnej generacji, w tym Facebook, Twitter, YouTube itd., Są wykorzystywane w Web 2.0.

*Obejmują nowe technologie internetowe, w tym Ruby on Rails, RSS, API itp.

*Współużytkowane i edytowalne struktury w formie zorientowanych na użytkownika do tworzenia własnych interfejsów API.

W obecnej sieci dane są prezentowane w taki sposób, aby były czytelne tylko dla człowieka i niezrozumiałe dla maszyny. Dlatego eksperci zasugerowali przejście na Web 3.0 w celu uczynienia treści maszynami zrozumiałymi.

Widżety: widżety to małe aplikacje, które można wstawiać na strony internetowe, kopiując i osadzając kod widżetu w kodzie strony internetowej. Mogą to być gry, kanały informacyjne, odtwarzacze wideo, itp. Niektórzy internetowi prognostycy uważają, że Web 3.0 pozwoli użytkownikom łączyć widżety w celu tworzenia mash-upów, po prostu klikając i przeciągając kilka ikon do ramki na stronie internetowej. Na przykład, jeśli potrzebujesz aplikacji, która pokazuje, gdzie dzieją się wiadomości, po prostu połącz ikonę kanału wiadomości z ikoną Google Earth, a Web 3.0 zajmie się resztą.

Web 3.0

Sieć 3.0 to witryna internetowa do odczytu, zapisu i wykonywania. Sieć 3.0 jest popularnie nazywana siecią semantyczną. Niektórzy nawet zakładają, że Web 3.0 to połączenie Web 2.0 i Semantic Web.

Strona www drastycznie poprawiła dostęp do informacji przechowywanych cyfrowo. Jednak zawartość strony www była do tej pory tylko do odczytu maszynowego, ale nie była zrozumiała dla komputera. Informacje na stronie www są przedstawiane głównie w języku naturalnym; dostępne dokumenty są w pełni zrozumiałe tylko dla ludzi. Sieć semantyczna jest oparta na zorientowanej na treść opis dokumentów cyfrowych ze znormalizowanymi słownikami, które zapewniają semantykę zrozumiałą dla maszyny. Jest to „wykonywalna” faza programu World Wide Web z dynamicznymi aplikacjami, usługami interaktywnymi i interakcją „maszyna-maszyna”. Sieć 3.0 to sieć semantyczna, która odnosi się do przyszłości. W sieci 3.0 komputery mogą interpretować informacje, tak jak ludzie, oraz inteligentnie generować i rozpowszechniać użyteczne treści dostosowane do potrzeb użytkowników. Web 3.0 można scharakteryzować w następujący sposób:

*Zawiera połączone dane lub hiperdane, w których obiekty danych są połączone z innymi obiektami danych (podobnie do dzisiejszych linków do stron internetowych)

*Zawiera duże zbiory danych hiperdanych, takie jak DBpedia (wysiętek społeczności mający na celu wyodrębnienie ustrukturyzowanych informacji z Wikipedii i udostępnienie ich w Internecie)

*Potrzebny jest język zapytań dla hiperdanych zdolny do traktowania całej sieci jako pojedynczego centrum danych, zwany SPARQL.

*Jest to tak zwany „Internet rzeczy”, w którym miliardy podmiotów niebędących ludźmi (w tym domy, samochody i urządzenia) generuje i publikuje własne hiperdane.

Jeśli semantyka jest badaniem znaczenia, pomyśl o sieci 3.0 jako o znaczącej sieci. Ogólnie rzecz biorąc, rzeczy w Internecie zostaną opisane za pomocą języków deskryptorów, aby komputery mogły zrozumieć, czym one są. Komputery będą mogły korzystać z danych znajdujących się na stronach internetowych. Kiedy więc szukasz czegoś, osoby, restauracji, hotelu, maszyna trafia do swojej rozległej sieci znaczących połączonych danych, tworzy dla Ciebie połączenia i sugeruje przydatne linki, których twój ludzki umysł nigdy by nie znalazł. Z prędkością warp!

Wniosek:

Czy zastanawiałeś się kiedyś, w jaki sposób Google pobiera informacje? Czy sieć jest naprawdę wystarczająco inteligentna, aby przekazywać za jej pomocą nasze wymagania? Jeśli chodzi o obecny scenariusz, odpowiedź jest duża - nie. To właśnie problem niepełnosprawności sieci jest przewyżniany w sieci nowej generacji, czyli w sieci 3.0. W przeciwieństwie do Web 2.0, który koncentruje się na ludziach, Web 3.0 koncentruje się na maszynach. Krótko mówiąc, aby wyjaśnić w jednym zdaniu, sieć staje się inteligentna, aby rozumieć potrzeby użytkowników i lepiej im służyć.

Potrzeba sieci semantycznej

Cel:

Omówimy następujące tematy:

- Działanie aktualnej sieci.
- roboty sieciowe.
- Zalety sieci semantycznej.

Wprowadzenie

„Dlaczego potrzebujemy sieci semantycznej?” To kolejne pytanie, które może uderzyć każdego, kto czyta o sieci semantycznej. Kiedy ludzie czują się dobrze z obecnym internetem, to jaka jest potrzeba przejścia na nową platformę.

Sieć WWW nie jest statycznym zbiornikiem informacji, ale nieustannie rozszerzającym się oceanem faktów. Każdego roku średnio do sieci trafia 51 milionów witryn. Ta liczba ma nikłe szanse na utratę wartości w przyszłości. Udowodniono, że w nadchodzących latach wzrośnie. Obecnie prawie wszystkie organizacje wspierają otwarte dane i udostępniają swoje dane w Internecie. Był czas, kiedy innowacje ograniczały się do czterech drzwi laboratoriów innowacji. Teraz jest czas, kiedy drzwi są otwarte dla wszystkich za pośrednictwem danych open source. Bez wątplenia coraz więcej informacji dodawanych do sieci uczyni ją bardziej zaradną, ale będzie to również stanowić poważny problem w najbliższej przyszłości. Możemy nie wiedzieć, które z nich są poprawnymi danymi do wykorzystania, gdy mamy

Prosta czynność

Otwórz Google i wpisz następujące dwa słowa-

- Radość (naciśnij „Enter” i zobacz wynik)
- Zachwył (naciśnij „Enter” i zobacz wynik)

W obu przypadkach otrzymasz różne zestawy wyników, chociaż oba słowa oznaczają to samo. Dzieje się tak, ponieważ sieć nie może zrozumieć, że oba słowa oznaczają to samo. Na najbardziej podstawowym poziomie strony internetowe są traktowane jako ciągi słów i przetwarzane. Załóżmy teraz, że przesyłam następujące dwa zapytania do dowolnej z obecnych wyszukiwarek internetowych:

Zapytanie 1: Domino dobrze sprawdza się również przy dużym obciążeniu.

Zapytanie 2: Domino dobrze sprawdza się również w przypadku dużego popytu.

Zapytanie 1 dotyczy oprogramowania o nazwie Domino (platforma aplikacji serwera IBM używana do obsługi poczty e-mail w przedsiębiorstwie, przesyłania wiadomości, planowania i współpracy), które jest zdolne do pracy również pod dużym obciążeniem, podczas gdy Zapytanie 2 dotyczy punktu sprzedaży pizzy, który dobrze służy nawet pod wysoki popyt. Po przestaniu do przetwarzania wyszukiwarka nie mogła dostarczyć odpowiedniego wyniku dla żadnego z zapytań. Powodem tego jest to, że system nie jest inteligentny. Sieć semantyczna zapewnia tę inteligencję.

Podejście Web 2.0

Sieć semantyczna jest bardzo złożonym tematem, dlatego najlepszym sposobem na jej zrozumienie jest porównanie i studiowanie aplikacji i technologii, które są nam znane. Przyjrzyjmy się więc tematowi sieci semantycznej za pomocą naszej ulubionej wyszukiwarki, czyli Google. Zanim przejdziemy do algorytmu i techniki używanej przez Google, spróbujmy zrozumieć niektóre podstawowe pojęcia z nim związane.

Robot sieciowy

Głównym celem projektowania przeszukiwacza sieci jest umożliwienie pobierania stron internetowych i dodanie ich reprezentacji do lokalnego repozytorium. Robot to program, który odwiedza witryny internetowe i odczytuje ich strony oraz inne informacje w celu tworzenia wpisów do indeksu wyszukiwarki. Roboty internetowe są również nazywane „pająkami”.

Rola robota internetowego:

- Crawlery internetowe wędrują po sieci w celu zautomatyzowania określonych zadań związanych z siecią.
- Są odpowiedzialni za zbieranie treści internetowych.

Podstawowy algorytm stosowany przez roboty sieciowe:

- Rozpocznij od strony „seed”.
- Utwórz wiersz / kolejkę dla powiązanych stron.
- Pobierz stronę źródłową i przetwórz ją.
- Wyodrębnij adresy URL, na które wskazują.
- Utwórz wpis w repozytorium.
- Umieść wyodrębnione adresy URL w kolejce.
- Pobieranie każdego adresu URL z kolejki jeden po drugim.
- Dla każdego pobranego adresu URL powtórz powyższy krok.

Rodzaje Crawlerów:

- Crawler wsadowy: ten typ przeszukiwaczy przeszukuje migawkę swojej przestrzeni przeszukiwania, aż osiągną określony rozmiar lub limit czasu.
- Crawler przyrostowy: ten typ robotów indeksuje swoją przestrzeń indeksowania w sposób ciągły, ponownie odwiedzając adres URL, aby zapewnić aktualność.
- Skupiony crawler: ten typ robotów indeksujących, jak sama nazwa wskazuje, jest dość skoncentrowany na przeszukiwanych stronach. Próbuje zaindeksować strony związane z określonym tematem i zminimalizować liczbę tych stron są poza tematem i są zbierane.
- Rozproszony crawler indeksujący: rozproszone przeszukiwanie sieci WWW jest techniką przetwarzania rozproszonego. Wiele robotów współpracuje ze sobą w celu rozpowszechniania w procesie indeksowania sieci, aby objąć jak największą część sieci. Centralny serwer zarządza komunikacją i synchronizacją węzłów, ponieważ jest rozproszony geograficznie. Zasadniczo wykorzystuje algorytm Page Rank w celu zwiększenia wydajności i jakości wyszukiwania. Zaletą rozproszonego robota indeksującego jest to, że jest odporny na awarie systemu i inne zdarzenia oraz może być dostosowany do różnych aplikacji indeksujących.
- Crawler równoległy: Wiele robotów pracujących równolegle nazywane jest pełzakami równoległymi. Przeszukiwacz równoległy składa się z wielu procesów przeszukiwania zwanych C-proc, które mogą działać w sieci stacji roboczych. Roboty równoległe zależą od aktualności strony i wyboru strony. Robot równoległy może znajdować się w sieci lokalnej lub być rozprowadzany w odległych geograficznie lokalizacjach. Zrównoleglenie systemu crawlingu jest bardzo istotne z punktu widzenia pobierania dokumentów w rozsądnym czasie.

Normalizacja adresów URL:

Roboty zazwyczaj normalizują adresy URL, aby uniknąć wielokrotnego indeksowania tego samego zasobu. Zasadniczo ma na celu modyfikację i standaryzację adresu URL w spójny sposób.

Przykłady robotów:

- Skuter
- WebRACE
- RBSE
- Google Crawler
- Www Worm
- Fontanna internetowa

Pozycja strony

Google używa PageRank do określenia ważności strony internetowej. Jest to jeden z wielu czynników określających, jakie strony muszą pojawić się w wynikach wyszukiwania. PageRank mierzy znaczenie strony internetowej. Zgodnie z teorią Page i Brina najważniejszymi stronami w Internecie są strony z maksymalną liczbą linków do nich prowadzących. PageRank traktuje linki jako głosy, w przypadku gdy strona prowadząca do innej strony oddaje głos. Ma to sens, ponieważ ludzie mają tendencję do umieszczania linków do odpowiednich treści, a strony zawierające więcej linków są zwykle lepszymi zasobami niż strony, do których nikt nie prowadzi. PageRank na tym się nie kończy. Sprawdza również znaczenie strony zawierającej link. Strony z wyższą pozycją PageRank mają większą wagę w „głosowaniu” z linkami niż strony z niższą pozycją PageRank. Sprawdza również liczbę linków na stronie oddających „głos”. Strony zawierające więcej linków mają mniejszą wagę. Ma to również pewien sens. Strony, które są ważne, są prawdopodobnie lepszymi autorytetami w prowadzeniu internautów do lepszych źródeł, a strony, które mają więcej linków, prawdopodobnie mniej dyskryminują, gdzie prowadzą linki. PageRank jest mierzony w skali od jednego do dziesięciu i przypisywany do poszczególnych stron w serwisie, a nie do całej witryny. Aby znaleźć PageRank strony, użyj paska narzędzi Google Toolbar. Bardzo niewiele stron ma PageRank 10, zwłaszcza w miarę wzrostu liczby stron w Internecie.

Jak działa wyszukiwarka?

Krok 1:

Crawlers przeszukują sieć w celu okresowego gromadzenia informacji o witrynie, która została zmieniona lub o nowej witrynie, która została dodana. Jak wspomniano powyżej, ta praca jest wykonywana okresowo, a nie dla każdego przesłanego zapytania. Prawda jest taka, że żadna wyszukiwarka nie działa w czasie rzeczywistym.

Uwaga: roboty nie obejmują całej sieci. Część sieci, która nie jest zakryta, nazywana jest niewidoczną siecią.

Krok 2:

Po zebraniu informacji kolejnym krokiem jest uporządkowanie informacji. Strony zebrane podczas procesu indeksowania są uporządkowane poprzez utworzenie indeksu, dzięki czemu wiemy, jak je wyszukać, gdy zajdzie taka potrzeba. Indeks może zawierać informacje o słowach i ich lokalizacji. Podczas wyszukiwania wyszukiwane terminy są przeszukiwane w indeksie w celu znalezienia odpowiednich stron.

Krok 3:

Za każdym razem, gdy wysyłasz zapytanie, wyszukiwarka wraca do swojej gigantycznej biblioteki indeksów, aby pobrać wymagane informacje. Ponieważ wyszukiwarka znajduje miliony pasujących informacji, używa algorytmu do decydowania o kolejności, w jakiej mają być wyświetlane wyniki.

Jak działa Google?

Google działa w rozproszonej sieci tysięcy komputerów i dlatego może przeprowadzać szybkie przetwarzanie równoległe. Przetwarzanie równoległe to metoda obliczeniowa, w której można wykonywać wiele obliczeń jednocześnie, co znacznie przyspiesza przetwarzanie danych. Google działa w trzech częściach:

- Googlebot, robot sieciowy, który wyszukuje i pobiera strony internetowe.
- Indeksator, który sortuje każde słowo na każdej stronie i przechowuje wynikowy indeks słów w ogromnej bazie danych.
- Procesor zapytań, który porównuje zapytanie wyszukiwania z indeksem i rekomenduje dokumenty, które uważa za najbardziej odpowiednie.

Googlebot

Googlebot to robot Google indeksujący sieć, który wyszukuje i pobiera strony w sieci oraz przekazuje je indeksatorowi Google. Łatwo jest wyobrazić sobie Googlebota jako małego pająka przemierzającego cyberprzestrzeń, ale w rzeczywistości Googlebot w ogóle nie przemierza sieci. Działa podobnie jak przeglądarka internetowa, wysyłając żądanie do serwera WWW w celu uzyskania strony internetowej, pobierając całą stronę, a następnie przekazując ją do indeksatora Google. Googlebot składa się z wielu komputerów żądających i pobierających strony znacznie szybciej niż jest to możliwe za pomocą przeglądarki internetowej. W rzeczywistości Googlebot może jednocześnie żądać tysięcy różnych stron. Aby uniknąć przytłaczającej liczby serwerów internetowych, Googlebot celowo wysyła żądania do każdego serwera internetowego wolniej, niż jest w stanie zrobić.

Gdy Googlebot pobiera stronę, pobiera wszystkie linki pojawiające się na stronie i dodaje je do kolejki w celu późniejszego indeksowania. Googlebot ma tendencję do napotykania niewielkiej ilości spamu, ponieważ większość autorów łączy linki tylko do stron o wysokiej jakości. Gromadząc linki z każdej napotkanej strony, Googlebot może szybko utworzyć listę linków, która obejmuje większą część sieci. Ta technika, znana jako głębokie indeksowanie, umożliwia również Googlebotowi badanie w głębi poszczególnych witryn. Ze względu na ogromną skalę głębokie indeksowanie może dotrzeć do prawie każdej strony w sieci. Ponieważ internet jest rozległy, może to zająć trochę czasu, więc niektóre strony mogą być indeksowane tylko raz w miesiącu. Chociaż jego funkcja jest prosta, Googlebot musi być zaprogramowany do obsługi kilku wyzwań. Po pierwsze, ponieważ Googlebot wysyła jednocześnie żądania dotyczące tysięcy stron, kolejka adresów URL typu „odwiedź wkrótce” musi być stale sprawdzana i porównywana z adresami URL znajdującymi się już w indeksie Google. Należy usunąć duplikaty w kolejce, aby uniemożliwić Googlebotowi ponowne pobranie tej samej strony. Googlebot musi określić, jak często ma ponownie odwiedzać stronę. Z jednej strony ponowne indeksowanie niezmiętej strony to strata zasobów. Z drugiej strony Google chce ponownie zindeksować zmienione strony, aby dostarczać aktualne wyniki.

Indeksator Google

Googlebot przekazuje indeksatorowi pełny tekst znalezionych stron. Te strony są przechowywane w bazie danych indeksów Google. Indeks ten jest sortowany alfabetycznie według wyszukiwanego hasła, a każdy wpis indeksu zawiera listę dokumentów, w których termin ten występuje, oraz lokalizację w

tekście, w której występuje. Taka struktura danych umożliwia szybki dostęp do dokumentów zawierających terminy zapytań użytkownika. Aby poprawić wydajność wyszukiwania, Google ignoruje (nie indeksuje) typowe słowa zwane słowami pomijanymi (takie jak, jest, on lub, jak, dlaczego, a także niektóre pojedyncze cyfry i pojedyncze litery). Słowa pomijane są tak powszechne, że niewiele robią, aby zawęzić wyszukiwanie, dlatego można je bezpiecznie odrzucić. Indeksator ignoruje również niektóre znaki interpunkcyjne i wiele spacji, a także konwertuje wszystkie litery na małe, aby poprawić wydajność Google.

Procesor zapytań Google

Procesor zapytań składa się z kilku części, w tym interfejsu użytkownika (pola wyszukiwania), „mechanizmu”, który ocenia zapytania i dopasowuje je do odpowiednich dokumentów oraz programu formatującego wyniki. PageRank to system Google do pozycjonowania stron internetowych. Strona z wyższym PageRank jest uważana za ważniejszą i jest bardziej prawdopodobne, że zostanie wymieniona nad stroną z niższą pozycją PageRank. Google bierze pod uwagę ponad sto czynników podczas obliczania rankingu PageRank i określania, które dokumenty są najbardziej odpowiednie dla zapytania, w tym popularność strony, pozycja i rozmiar wyszukiwanych haseł na stronie oraz bliskość wyszukiwanych haseł względem siebie. na stronie. We wniosku patentowym omówiono inne czynniki, które Google bierze pod uwagę podczas oceniania strony. Odwiedź raport SEOmoz.org, aby zapoznać się z interpretacją pojęć i praktycznych zastosowań zawartych w zgłoszeniu patentowym Google. Google stosuje również techniki uczenia maszynowego, aby automatycznie zwiększać wydajność poprzez uczenie się relacji i skojarzeń w ramach przechowywanych danych. Google ściśle strzeże formuł, których używa do obliczania trafności; są modyfikowane w celu poprawy jakości i wydajności oraz przechytrzenia najnowszych przebiegłych technik wykorzystywanych przez spamatorów. Indeksowanie pełnego tekstu sieci pozwala Google wyjść poza proste dopasowywanie pojedynczych wyszukiwanych haseł. Google nadaje wyższy priorytet stronom, które mają wyszukiwane hasła blisko siebie i w tej samej kolejności co zapytanie. Google może również dopasowywać frazy i zdania składające się z wielu słów. Ponieważ Google indeksuje kod HTML oprócz tekstu na stronie, użytkownicy mogą ograniczyć wyszukiwanie na podstawie tego, gdzie pojawiają się słowa zapytania, np. W tytule, w adresie URL, w treści i linkach do strony, oferowane opcje za pomocą formularza wyszukiwania zaawansowanego Google i za pomocą operatorów wyszukiwania (operatorów zaawansowanych).

Podejście do sieci semantycznej

- Wiedza zostanie zorganizowana w konceptualne przestrzenie zgodnie z jej znaczeniem.
- Zautomatyzowane narzędzia będą wspierać konserwację, sprawdzając niespójności i wydobywając nową wiedzę.
- Wyszukiwanie oparte na słowach kluczowych zostanie zastąpione odpowiadaniem na zapytania, tj. Żądana wiedza zostanie pobrana, wyodrębniona i zaprezentowana w sposób przyjazny dla człowieka.
- Obsługiwane będą odpowiedzi na zapytania dotyczące kilku dokumentów.
- Zdefiniowanie, kto może przeglądać określone części informacji (nawet części dokumentów) będzie możliwe.

Oprócz pobierania wyników z wyszukiwania, tak jak robi to teraz komputer (to znaczy systematycznie, zadając jedno pytanie i łącząc je ze słowami kluczowymi, aby uzyskać miliony odpowiedzi użytkownika), sieć semantyczna przeprowadziłaby bardziej ludzkie -podobny sposób rozwiązywania problemów. Łączyłby się nie tylko od A do Z, ale także od A do B do C i tak dalej, aż dotrze do Z. Sieć semantyczna

reorganizuje ogromną ilość informacji, które są dostępne dla nas w Internecie w sposób podobny do tego naszego umysłu. Byłoby to jak szkolenie internetu w zakresie zrozumienia kontekstu otaczającego wyszukiwane słowo lub frazę za pomocą tagów, które osoba wyszukująca dołącza do tematu. Sieć semantyczna służyłaby jako połączenie między człowiekiem a komputerem, sprawiając, że komputer myśli bardziej jak człowiek, jednocześnie pozwalając ludziom na prawdziwe myślenie. To jest ekscytujące i przerażające na wielu poziomach. Załóżmy na przykład, że chciałeś zjeść obiad ze swoim przyjacielem Hydiem. Następnie możesz przeprowadzić serię rozmów z Hydiem. Weźmy na przykład pod uwagę następujący zestaw rozmów:

„Mam spotkanie w moim biurze, więc nie mogę iść jutro po południu, ale po 15:00. Jestem wolny.”

"To wystarczy. Chodźmy do Meluha? "

„Jestem wegetarianinem, więc to nie działa na mnie”. (I tak dalej...)

Gdyby w tej transakcji wykorzystano technologię sieci semantycznej, Hydie miałyby „agenta”, który miałby dostęp do wszelkiego rodzaju informacji o niej, w tym jej kalendarza, wszelkich preferencji żywieniowych lub alergii, które mogła mieć, a także ocen restauracji, które otrzymała. Twój własny agent miałby dostęp do podobnych informacji na Twój temat. Ci dwaj agenci komunikowałiby się ze sobą, a następnie automatycznie sugerowałiby coś, co ma sens dla was obojga. Mogą nawet dokonać rezerwacji za Ciebie! Mówiąc bardziej praktycznie, naukowcy wykorzystują technologie sieci semantycznej, aby umożliwić maszynom wnioskowanie o nowych faktach na podstawie istniejących faktów i danych. Oznacza to, że technologie sieci semantycznej umożliwiają komputerom nie tylko przechowywanie i pobieranie informacji, ale także samodzielne tworzenie zupełnie nowych informacji.

Korzyści z sieci semantycznej

- Komputery mogą działać automatycznie. Ponieważ komputery mogą podejmować decyzje tak jak ludzie, mogą automatycznie wykonywać pracę, oszczędzając w ten sposób dużo energii i pieniędzy.
- Komputery mogą również dostosowywać systemy biznesowe, a firmy mogą prowadzić działalność bardziej ekonomicznie, wymagając mniejszego wysiłku ludzkiego.
- Możemy wykorzystać ustandaryzowany sposób efektywnego przechowywania i wyszukiwania informacji.
- Udostępnianie danych może być łatwiejsze dzięki sieci semantycznej, ponieważ hurtownie danych mogą być dystrybuowane. Odpowiednie informacje mogą pomóc ludziom w podejmowaniu natychmiastowych i właściwych decyzji.
- Ułatwia wymianę treści i przedmiotów nauczania.
- Umożliwia uczniom wyszukiwanie zasobów edukacyjnych na podstawie semantyki, ułatwiając w ten sposób przeszukiwanie wiedzy docelowej.
- Poprawia kontekstowe semantyczne środowiska e-learningowe, dostarczając semantyczne modele do modelowania kontekstowego.
- Skalowalna, wielokrotnego użytku, współdzielona treść kursu.
- Możliwość znalezienia i przesunięcia całego kursu.
- Gromadź treść, aby zaspokoić potrzeby ucznia.

Wprowadzenie do sieci semantycznej

Cel:

Tu przedstawiono różne technologie, które są postrzegane jako elementy składowe sieci semantycznej. Rozdział koncentruje się na zdefiniowaniu znaczenia każdego terminu i terminologii odgrywającej kluczową rolę w tworzeniu sieci semantycznej.

Definiowanie sieci semantycznej

Sieć semantyczna ma wiele dobrze znanych definicji. Poniżej wymieniono kilka z nich:

Definicja sieci semantycznej autorstwa Tima Bernera Lee:

„Ludzie wciąż pytają, czym jest Web 3.0. Myślę, że może, kiedy masz nakładkę skalowalnej grafiki wektorowej - wszystko faluje, zwiija się i wygląda mgliście w sieci Web 2.0 oraz dostęp do sieci semantycznej zintegrowane w ogromnej przestrzeni danych, będziesz mieć dostęp do niewiarygodnych zasobów danych”.

Dyrektor generalny Google, Eric Schmidt, stwierdził:

„Web3.0 to seria połączonych aplikacji. Podstawową technologią oprogramowania Web 3.0 jest sztuczna inteligencja, która może inteligentnie uczyć się i rozumieć semantykę. Dlatego zastosowanie technologii Web3.0 umożliwi bardziej spersonalizowany, dokładniejszy i inteligentny Internet”.

Założyciel Netflix, Reed Hastings, uważa, że Web 3.0 byłby pełną siecią wideo, jak podano poniżej:

„Web 1.0 to połączenie dial-up, średnia przepustowość 50 000; Web 2.0 to średnio 1 megabit przepustowości, a Web 3.0 będzie przez cały czas mieć 10 megabitów przepustowości, co będzie pełnowymiarową siecią wideo i będzie wyglądać jak sieć 3.0”.

Założyciel Yahoo, Jerry Yang, powiedział na TechNet Summit w listopadzie 2006 roku:

„Web 2.0 jest dobrze udokumentowany i omawiany. Moc sieci osiągnęła masę krytyczną, z możliwościami, które można wykonać na poziomie sieci. Widzimy również bogatsze urządzenia w ciągu ostatnich czterech lat i bogatsze sposoby interakcji z siecią, nie tylko w sprzęcie, takim jak konsole do gier i urządzenia mobilne, ale także w warstwie oprogramowania. Nie musisz być informatykiem, aby stworzyć program. Widzimy, że manifest w Web 2.0 i Web 3.0 będzie wspaniałym rozszerzeniem tego, prawdziwym społecznym medium... różnica między profesjonalistami, półprofesjonalnymi i konsumentami zostanie zatarta, tworząc efekt sieciowy biznesu i aplikacji.”

Wyjaśnienie:

Sieć semantyczna to ewolucja i rozszerzenie istniejącej sieci, która umożliwia komputerom manipulowanie danymi i informacjami. Idea sieci semantycznej jest wciąż w fazie badań i rozwoju. Nie ma niezawodnego sposobu przetwarzania semantyki, który kwestionuje cel opracowania tak dużego projektu. Jednak za koncepcją takiego rozszerzenia istniejącej wcześniej dokumentacji informacji, która jest obecnie rozpowszechniana w sieci WWW, kryje się silny apel. W sieci istnieje ogromna atrakcja, która ma potencjalną możliwość „poznania” i „rozumienia” danych. Dodaje to bardziej humanistyczną jakość do standardowego przetwarzania danych, ponieważ sieć semantyczna stara się zniwelować lukę między zwykłym dostarczaniem dokumentów ludziom a automatycznym przetwarzaniem danych i informacji. Jednak, aby dojść do tego punktu, programiści muszą zapewnić język, który może wyrazić zarówno dane, jak i reguły rozumowania.

Charakterystyka sieci semantycznej

Inteligencja:

Ekspertsi uważają, że jedną z najbardziej obiecujących funkcji Web 3.0 będzie sieć z inteligencją, czyli inteligentną siecią. Aplikacje będą działać inteligentnie z wykorzystaniem interakcji i inteligencji człowiek-komputer. Różne narzędzia i techniki oparte na sztucznej inteligencji (AI) (takie jak zbiory przybliżone, zbiory rozmyte, sieci neuronowe, uczenie maszynowe itp.) Zostaną włączone do aplikacji w celu inteligentnego działania. Oznacza to, że aplikacja oparta na sieci Web 3.0 może bezpośrednio przeprowadzać inteligentną analizę, a wtedy optymalne wyniki byłyby możliwe bez dużej interwencji użytkownika. Dokumenty w różnych językach można inteligentnie przetłumaczyć na inne języki w Web 3.0. Powinien umożliwić nam posługiwanie się językiem naturalnym. Dlatego użytkownicy mogą używać swojego języka ojczystego do komunikacji z innymi dookoła świata.

Personalizacja

Inną cechą Web 3.0 jest Personalizacja. Osobiste lub indywidualne preferencje byłyby brane pod uwagę podczas różnych czynności, takich jak przetwarzanie informacji, wyszukiwanie, tworzenie spersonalizowanego portalu w sieci, itp. Sieć semantyczna byłaby podstawową technologią personalizacji w sieci 3.0.

Rozumowanie

Sieć semantyczna umożliwia wyszukiwanie, integrację, odpowiadanie na złożone zapytania, połączenia i analizę (ścieżki, sub-grafy), wyszukiwanie wzorców, eksplorację, walidację hipotez, odkrywanie, wizualizację itp.

Interoperacyjność

Interoperacyjność odnosi się do takich aspektów, jak płynna integracja danych z heterogenicznych źródeł, dynamiczna kompozycja, współdziałanie usług internetowych i wyszukiwarek nowej generacji. Aplikacje Web 3.0 byłyby łatwe do dostosowania i mogą niezależnie pracować na różnego rodzaju urządzeniach. Aplikacja oparta na Web 3.0 mogłaby działać na wielu typach komputerów, urządzeniach mikrofalowych, urządzeniach przenośnych, telefonach komórkowych, telewizorach, samochodach i wielu innych.

Użyteczność

Użyteczność obejmuje nowe paradygmaty wyszukiwania informacji, interfejsy użytkownika, techniki interakcji i wizualizacji, które z kolei wymagają metod radzenia sobie między innymi z zależnością kontekstową, personalizacją, zaufaniem i pochodzeniem, jednocześnie ukrywając przed użytkownikiem podstawowe problemy obliczeniowe.

Możliwość zastosowania

Możliwość zastosowania odnosi się do szybko rozwijających się obszarów zastosowań technologii i metod sieci semantycznej, kwestii przenoszenia najnowocześniejszych wyników badań do rzeczywistych zastosowań oraz do rozwoju nowych metod i podstaw opartych na rzeczywistych potrzebach aplikacji z różnych dziedzin.

Uwaga 1:

- Semantyka z metadanymi i ontologiami dla heterogenicznych dokumentów i wielu repozytoriów danych, w tym w Internecie, została omówiona w latach 90.

- Tim Berners-Lee użył terminu „sieć semantyczna” w swojej książce z 1999 roku.
- Pierwsze 5 lat badań nad siecią semantyczną, przyniosło zbyt dużo sztucznej inteligencji / DL, ale w ostatnich latach dominowały prace bardziej praktyczne / stosowane.

Uwaga 2:

Wszechstronna sieć - Wszechobecna sieć to termin używany do opisanie zjawiska, w którym sieć jest obsługiwana przez wiele różnych urządzeń elektronicznych.

Sieć semantyczna kontra sztuczna inteligencja (AI)

W rzeczywistości technologie sieci semantycznej dotyczą w takim samym stopniu danych, jak rozumowania i logiki. RDF, podstawowa technologia w stosie sieci semantycznej, to elastyczny model danych grafowych, który w żaden sposób nie obejmuje logiki ani rozumowania. Realizacja wizji sieci semantycznej nie opiera się na inteligencji na poziomie ludzkim. W rzeczywistości do wyzwań podchodzi się w inny sposób. Cały problem AI jest głęboko naukowy, być może porównywalny z głównymi problemami fizyki (wyjaśnij świat fizyczny) lub biologii (wyjaśnij świat żyjący). W AI częściowe rozwiązania mogą nie działać.

Ale w sieci semantycznej częściowe rozwiązania będą działać. Nawet jeśli inteligentny agent nie jest w stanie dojść do wszystkich wniosków, które mógłby wyciągnąć użytkownik, agent nadal będzie wnosił wkład do sieci znacznie lepiej niż obecna sieć. Jeśli ostatecznym celem sztucznej inteligencji jest zbudowanie inteligentnego agenta, wykazującego inteligencję na poziomie ludzkim (i wyższym), celem sieci semantycznej jest pomoc użytkownikom w ich codziennych działaniach online. Oczywiście jest, że sieć semantyczna będzie szeroko wykorzystywać obecną technologię sztucznej inteligencji, a postęp w tej technologii doprowadzi do lepszej sieci semantycznej. Nie trzeba jednak czekać, aż sztuczna inteligencja osiągnie wyższy poziom osiągnięć; Obecna technologia sztucznej inteligencji jest wystarczająca, aby przejść długą drogę w kierunku realizacji wizji sieci semantycznej.

SDLC - przegląd

SDLC oznacza cykl życia oprogramowania.

- Planowanie: Najważniejsza część tworzenia oprogramowania, tj. zbieranie wymagań lub analiza wymagań jest wykonywana przez najbardziej wykwalifikowanych i doświadczonych inżynierów oprogramowania w organizacji. Po zebraniu wymagań od klienta tworzony jest dokument zakresu, w którym określa się i dokumentuje zakres projektu.
- Wdrożenie: Implementacja odnosi się do kodowania wykonanego przez inżynierów oprogramowania w celu wdrożenia wymagań klienta.
- Testowanie: jest to proces znajdowania defektów lub błędów w tworzonym oprogramowaniu. Istnieją różne typy testów, wykonywane na różnych etapach tworzenia oprogramowania. Na przykład testy jednostkowe i testy regresyjne są zwykle wykonywane przez programistów, podczas gdy testy dymne i testy białoskrzynkowe są wykonywane przez testerów.
- Dokumentacja: każdy etap projektu jest dokumentowany w celu wykorzystania w przyszłości i ulepszenia oprogramowania w procesie rozwoju. Dokumentacja projektowa może obejmować napisanie interfejsu programowania aplikacji (API), wymagań biznesowych, docelowej grupy odbiorców itp.
- Wdrożenie i utrzymanie: na tym etapie oprogramowanie jest wdrażane po zatwierdzeniu do wydania.

- Utrzymanie: To jest ostatni etap SDLC. Ulepszanie oprogramowania i nowe wymagania (żądania zmian) mogą zająć więcej czasu niż czas potrzebny do utworzenia początkowego rozwoju oprogramowania.

Istnieje kilka modeli rozwoju oprogramowania, które są stosowane przez różne organizacje:

Model wodospadu: Model ten obejmuje całkowite zakończenie pierwszej fazy przed rozpoczęciem następnej. Po pomyślnym zakończeniu każdej fazy następuje przegląd, aby sprawdzić, czy projekt przebiega zgodnie z planem i czy kontynuacja jest wykonalna.

Model w kształcie litery V: Model ten koncentruje się na wykonywaniu procesów w sposób sekwencyjny, podobnie jak w modelu kaskadowym, ale z większym naciskiem na testowanie. Procedury testowe są pisane jeszcze przed rozpoczęciem kodowania. Systematyczny plan jest generowany przed rozpoczęciem fazy rozwoju.

Model przyrostowy: ten model cyklu życia obejmuje wiele cykli rozwojowych. Cykle są podzielone na mniejsze iteracje. Te iteracje przechodzą przez zestaw faz, w tym wymagania, projekt, implementację i testowanie. Działająca wersja oprogramowania jest tworzona podczas pierwszej iteracji, więc działające oprogramowanie jest tworzone na wczesnym etapie procesu rozwoju.

Bloki konstrukcyjne sieci semantycznej

Ontologia

Teraz, po głębokim wprowadzeniu do sieci semantycznej, spróbujmy zrozumieć jej główne elementy składowe. Zaczniemy od najważniejszego w swoim rodzaju, czyli ontologii. Dzięki ontologii komputery mogą czasami zachowywać się tak, jakby „rozumiały” informacje, które niosą. Tu pojawia się termin „semantyczny”. W tej sieci staramy się, aby znaczenia były tak jasne, że nawet komputer może je zrozumieć. Aby mieć naprawdę inteligentne systemy, wiedza musi być gromadzona, przetwarzana, ponownie wykorzystywana i przekazywana. Ontologie obsługują wszystkie te zadania. Termin „ontologia” można zdefiniować jako wyraźną specyfikację konceptualizacji. Dokładne znaczenie zależy od zrozumienia terminów „specyfikacja” i „konceptualizacja”. Jawna specyfikacja konceptualizacji oznacza, że ontologia jest opisem (podobnie jak formalna specyfikacja programu) pojęć i relacji, które mogą istnieć dla agenta lub wspólnoty agentów. Definicja ta jest zgodna z zastosowaniem ontologii jako zbioru definicji pojęć. Podstawą ontologii jest często taksonomia. Taksonomia to klasyfikacja rzeczy w formie hierarchicznej. Zwykle jest to drzewo lub krata, która wyraża relację pod-założeń - tj. A podsuwa B, co oznacza, że wszystko, co jest w A, jest również w B. Przykładem jest klasyfikacja organizmów żywych. Taksonomia zwykle ogranicza zamierzone użycie klas, gdzie klasy są podzbiorami zbioru wszystkich możliwych indywidualów w domenie. Ontologie są uważane za jeden z filarów sieci semantycznej, chociaż nie mają powszechnie akceptowanej definicji. Słownictwo (sieci semantycznej) można uznać za specjalną formę (zwykle lekkiej) ontologii. Aby dzielić się wiedzą między agentami, musi istnieć porozumienie w sprawie przekazywanych tematów. Rodzi to kwestię zaangażowania ontologicznego. Zobowiązania ontologiczne pozwalają wielu podmiotom na znaczące komunikowanie się na temat domeny, niekoniecznie indywidualnie działając w oparciu o globalną teorię. W kontekście wielu agentów, wspólna ontologia służy jako specyfikacja na poziomie wiedzy zobowiązań ontologicznych zbioru uczestniczących agentów. Wspólna ontologia definiuje słownictwo, za pomocą którego zapytania i twierdzenia są wymieniane między agentami, zapewniając w ten sposób środki do wypełnienia semantycznej luki istniejącej między leksykalnymi reprezentacjami informacji a jej nieleksykalną konceptualizacją.

RDF / OWL

RDF to specyfikacja, która definiuje model reprezentacji świata i składnię serializacji i wymiany tego modelu. W3C opracowało serializację XML dla RDF. RDF XML jest standardowym formatem wymiany RDF w sieci semantycznej, chociaż nie jest to jedyny format. Na przykład Notation3 to doskonała alternatywna serializacja RDF XML w postaci zwykłego tekstu. RDF zapewnia spójny, ustandaryzowany sposób opisywania i przeszukiwania zasobów internetowych, od stron tekstowych i grafiki po pliki audio i klipy wideo. Oferuje interoperacyjność syntaktyczną i stanowi podstawę do budowy sieci semantycznej. RDF definiuje skierowany wykres relacji. Są one reprezentowane przez trójki obiekt-atrybut-wartość, to znaczy obiekt O ma atrybut A o wartości V.

SPARQL

SPARQL to język zapytań RDF, który może pobierać i manipulować danymi przechowywanymi w formacie Resource Description Framework. Został znormalizowany przez grupę roboczą RDF Data Access (DAWG) Konsorcjum World Wide Web i jest uznawany za jedną z kluczowych technologii Semantic Web. SPARQL umożliwia kwerendę składającą się z potrójnych wzorców, spójników, rozłączeń i opcjonalnych wzorców.

Ontologia

Cel:

Szczegółowo wyjaśniono rolę ontologii w budowaniu Web 3.0 oraz przewagę ontologii nad tradycyjnymi strukturami hierarchicznymi. Sekcja omawia również pokrótce Protégé, który jest narzędziem do generowania ontologii.

Wprowadzenie do ontologii

Kierując się tradycyjną definicją, „ontologię można zdefiniować jako gałąź metafizyki zajmującą się naturą i relacjami bytu”. Niektóre pytania zawsze wiążą się z ontologią, ilekroć próbuje się wywnioskować relację bytów. Poniżej przedstawiono główne pytania:

- „Co można powiedzieć, że istnieje?”
- „Do jakich kategorii, jeśli w ogóle, możemy posortować istniejące rzeczy?”
- „Jakie jest znaczenie bycia bytem?”
- „Jakie są różne sposoby bycia bytem?”

Możemy użyć tej samej strategii do budowania ontologii w sieci semantycznej, tj. Musisz upewnić się, że ontologia, którą zbudowałeś, odpowiada na powyższe podstawowe pytania. Pomoże ci to wywnioskować, że uwzględniłeś wszystkie istotne elementy wymagane przez twoją maszynę, aby zrozumieć fakt, który próbujesz przedstawić. Logika powyższego stwierdzenia nabierze większego znaczenia, gdy w pełni zrozumiesz pojęcie ontologii.

Definicja ontologii

Ontologia jest jawną i abstrakcyjną, modelowaną reprezentacją już zdefiniowanych, skończonych zbiorów terminów i pojęć, zaangażowaną w inżynierię wiedzy, zarządzanie wiedzą i inteligentną integrację informacji. Mówiąc dokładniej, mogą zdefiniować ontologię jako „wyraźną specyfikację konceptualizacji” (sformułowaną przez Thomasa Grubera). Podczas gdy specyfikacja terminów i konceptualizacja wywołały wiele debat, zasadnicze punkty tej definicji ontologii to:

- Ontologia definiuje (określa) pojęcia, relacje i inne rozróżnienia, które są istotne dla modelowania domeny.

- Specyfikacja ma postać definicji słownictwa reprezentacyjnego (klas, relacji itd.), które zapewniają znaczenie słownictwa i formalne ograniczenia jego spójnego użycia.

Konceptualizacja

Konceptualizacja zawiera obiekty, pojęcia i inne byty, które, jak się zakłada, istnieją w pewnym obszarze zainteresowań oraz relacje, które je utrzymują. Konceptualizacja to abstrakcyjny, uproszczony pogląd na świat, który w jakimś celu chcielibyśmy przedstawić.

Konceptualizację można zdefiniować jako krotkę (U, R) , gdzie,

- U to zbiór nazywany wszechświatem dyskursu.
- R jest zbiorem relacji na U .

Konceptualizacja to abstrakcyjny, uproszczony pogląd na świat, który w jakimś celu chcielibyśmy przedstawić. Każda baza wiedzy, system oparty na wiedzy lub agent na poziomie wiedzy jest zaangażowana pewną konceptualizacją, jawnie lub niejawnie.

Uwaga: W kontekście ontologii formalny oznacza zrozumiały maszynowo. Dzielenie się oznacza wiedzę zaakceptowaną przez grupę.

Zakres ontologii

Ontologia definiuje wspólne słownictwo dla badaczy, którzy muszą udostępniać informacje w domenie. Zawiera interpretowalne maszynowo definicje podstawowych pojęć w tej dziedzinie i relacji między nimi. Ontologia to opis (podobnie jak formalna specyfikacja programu) pojęć i relacji, które mogą istnieć dla agenta lub społeczności agentów. Ogólnie rzecz biorąc, przedmiotem ontologii jest badanie kategorii rzeczy, które istnieją lub mogą istnieć w jakiejś dziedzinie. Produktem takiego badania jest katalog typów rzeczy, które z założenia istnieją w dziedzinie zainteresowania D z perspektywy osoby używającej języka L w celu mówienia o D . Typy w ontologii reprezentują predykaty, znaczenia słów lub pojęcia i typy relacji języka L używane do omawiania tematów w dziedzinie D . Logika, taka jak rachunek predykatów, grafy pojęciowe, ontologia lub KIF, która nie jest interpretowana, jest ontologicznie neutralna. Nie nakłada żadnych ograniczeń dotyczących przedmiotu ani sposobu, w jaki można go scharakteryzować. Jako specyfikacja interfejsu ontologia zapewnia język do komunikacji z agentem. Agent obsługujący ten interfejs nie jest zobowiązany do używania terminów ontologii jako wewnętrznego kodowania swojej wiedzy. Niemniej jednak definicje i formalne ograniczenia ontologii nakładają ograniczenia na to, co można sensownie wyrazić w tym języku. Zasadniczo zobowiązanie się do ontologii (np. Wspieranie interfejsu przy użyciu słownika ontologii) wymaga, aby stwierdzenia, które są przypisywane do danych wejściowych i wyjściowych, były logicznie zgodne z definicjami i ograniczeniami ontologii. Jest to analogiczne do wymagania, aby wiersze tabeli bazy danych (lub instrukcje wstawiania w SQL) były zgodne z ograniczeniami integralności, które są podawane deklaracyjnie i niezależnie od wewnętrznych formatów danych.

Podobnie, chociaż ontologia musi być sformułowana w jakimś języku reprezentacji, ma być specyfikacją poziomu semantycznego, tj. Jest niezależna od strategii modelowania danych lub implementacji. Na przykład konwencjonalny model bazy danych może reprezentować tożsamość osób przy użyciu klucza podstawowego, który przypisuje każdej osobie niepowtarzalny identyfikator. Jednak identyfikator klucza podstawowego jest artefaktem procesu modelowania i nie oznacza niczego w domenie. Ontologie są zwykle formułowane w językach, które są bardziej wyraziste niż formalizm logiczny, taki jak rachunek predykatów. Pozwala to projektantowi ontologii na ustalanie ograniczeń semantycznych bez wymuszania określonej strategii kodowania. Podobnie w ontologii można

przedstawić ograniczenia zachodzące między relacjami w prostej deklaracji (A jest podklasą B), która może być zakodowana jako sprzężenie kluczy obcych w modelu relacyjnym.

Przejście z bazy danych do ontologii

Dzięki determinizmowi technologicznemu zawsze jesteśmy nastawieni na kolejną błyszczącą innowację. Na czele tej kolejki innowacji stoi ontologia. Będąc studentami inżynierii komputerowej, tworzenie front-endu i back-endu aplikacji internetowej nie jest już wielką sprawą, ale z pewnością trzeba stawić czoła wyzwaniu, jeśli chodzi o uczynienie systemu bardziej inteligentnym. W takich okolicznościach najlepszym sposobem byłaby migracja z bazy danych do ontologii. Od czasu wprowadzenia Web 1.0, która w rzeczywistości jest platformą informacji tylko do odczytu, do Web 2.0, która ma być platformą uczestnictwa, zawsze kładziono nacisk na opracowanie bardziej zniuansowanego sposobu organizacji informacji. Zasadniczo ontologie służą do porządkowania informacji. Bez względu na dziedzinę lub zakres, ontologia jest opisem światopoglądu przy użyciu połączonej lub sieciowej struktury grafów. Trochę inaczej, przyjrzyjmy się bardziej powszechnemu terminowi, tj. „System zarządzania relacyjnymi bazami danych (RDBMS)”. Mamy wiele systemów opartych na RDBMS. W rzeczywistości większość współczesnych gospodarek używa RDBMS jako podstawy. Najbardziej oczywistym powodem takiego stanu rzeczy jest fakt, że system zarządzania bazą danych jest standardowym sposobem przechowywania danych w sposób trwały, a ich ekstrakcję można łatwo przeprowadzić za pomocą SQL. Ale rozważ przypadek, w którym masz kilka baz danych reprezentowanych w różnych formatach, a Twoja aplikacja nalega na integrację tych baz danych, napotkasz kilka problemów z powodu ich różnych formatów. Jest to obszar, w którym ontologia zyskuje na znaczeniu. Ze względu na strukturę relacji leżącą u podstaw ontologii są one doskonałymi narzędziami do odkrywania i łączenia. Analiza tego wykresu relacji jest podstawą Eksploratora pojęć. Oddzielenie wiedzy dziedzinowej od wiedzy operacyjnej i umożliwienie jej ponownego wykorzystania, wspólne rozumienie struktury informacji między agentami oprogramowania to tylko niektóre z ważnych celów realizowanych za pośrednictwem ontologii. Tak jak na przykład, jeśli istnieje kilka różnych witryn internetowych zawierających informacje o lekach i jeśli te witryny internetowe mają wspólną i publikują tę samą podstawową ontologię terminów, których używają, agenci komputerowi mogą wydobywać i agregować informacje z tych różnych witryn. Agenci mogą wykorzystywać te zagregowane informacje do odpowiadania na zapytania użytkowników lub jako dane wejściowe do innych aplikacji. W ten sposób zapewniają dzielenie się wiedzą i ponowne wykorzystanie zarówno między ludźmi, jak i agentami komputerowymi, ze względu na ich zdolność do przeplatania zrozumienia ludzi i maszyn za pomocą semantyki formalnej i rzeczywistej.

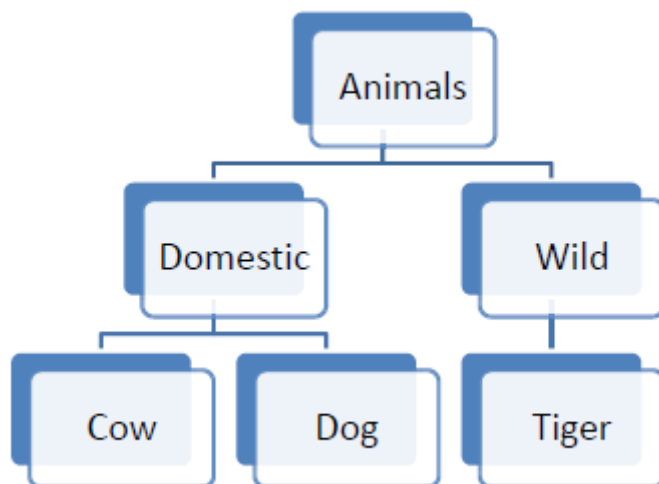
Taksonomia jako prekursor ontologii

Nauka koncepcji w sieci semantycznej nie jest łatwym zadaniem. Dzieje się tak, ponieważ większość tematów jest zorientowana na badania i aby właściwie zrozumieć definicje tych pojęć, konieczne jest dogłębne zrozumienie terminów, które tworzą definicję. Najłatwiej to zrobić, próbując powiązać te terminy techniczne z odpowiadającym im znaczeniem słownikowym. Słowo taksonomia pochodzi od dwóch greckich słów - Taxis i Nomos, które oznaczają odpowiednio „układ i porządek rzeczy” oraz „cokolwiek przypisane, użycie lub zwyczaj, prawo lub zarządzenie”. W sposób formalny taksonomię można zdefiniować jako klasyfikację przedmiotową, która porządkuje termin w kontrolowanym słowniku i umożliwia grupowanie powiązanych terminów i kategoryzowanie ich w sposób ułatwiający znalezienie właściwego terminu do użycia. Wiele taksonomii ma strukturę hierarchiczną, ale nie jest to wymagane. Taksonomie można wyjaśnić w prostych słowach jako graficzną reprezentację klasyfikacji rzeczy, idei itp. Zgodnie z pewnym schematem taksonomicznym można sklasyfikować prawie wszystko, o ile ma logiczną hierarchię. Pracują nad porządkowaniem informacji. Podstawą ontologii jest często taksonomia. Taksonomia to klasyfikacja rzeczy w formie hierarchicznej. Zwykle jest to drzewo lub krata,

która wyraża relację podporządkowania. Przykładem jest klasyfikacja organizmów żywych. Taksonomia zwykle ogranicza zamierzone użycie klas, gdzie klasy są podzbiorami zbioru wszystkich możliwych indywiduów w domenie.

Przykład taksonomii:

Kiedy spróbujemy porównać pojęcie taksonomii i ontologii, przekonamy się, że oba działają na podobnych pojęciach. Dla laika lub użytkowników końcowych to prawie ta sama koncepcja. Ale technokraci są zobowiązani postrzegać je oddzielnie. Dzieje się tak, ponieważ ontologia implikuje szerszą perspektywę taksonomii. Taksonomia to zwykła klasyfikacja obiektów w celu lepszego zrozumienia obiektów i ich podtypów, podczas gdy ontologia reprezentuje złożoną sieć połączeń między węzłami, która koncentruje się głównie na strukturze umożliwiającej wnioskowanie i logiczne wnioski.



Różnica między ontologią a taksonomią

Ontologia: Taksonomia

Ontologie można zdefiniować jako wyraźną specyfikację konceptualizacji. : Taksonomię definiuje się jako klasyfikację elementów w uporządkowanym systemie, który wskazuje na naturalne związki.

Ontologie zazwyczaj zawierają terminy opisowe w celu określenia słownictwa zawartości węzła. : Taksonomie zazwyczaj nie zawierają opisowych słów kluczowych dla każdego elementu.

Ontologie reprezentują bardzo złożoną formę wzajemnych połączeń. : Taksonomie są stosunkowo prostsze i dobrze się rozwijają, aby uporządkować temat w określony sposób.

Relacje w ontologii podlegają subiektywnej interpretacji, a tym samym wyzwalają wnioskowanie. : Taksonomie wydają się być dość pobłażliwe, jeśli chodzi o rodzaj relacji między rodzicami a węzłem potomnym w drzewie klasyfikacyjnym.

Typy ontologii

Proponuje się kilka typów klasyfikacji dla ontologii w oparciu o charakterystykę komponentów ontologii. Poniżej wyjaśniono kilka ważnych typów klasyfikacji proponowanych dla ontologii

Klasyfikacja według celu

Ontologia aplikacji : Ontologia aplikacji jest wykorzystywana w określonych aplikacjach, w których rozumownik jest implementowany w oparciu o ontologię.

Ontologia referencyjna : Ontologia referencyjna jest wykorzystywana podczas opracowywania aplikacji, do wzajemnego zrozumienia i wyjaśnienia między podmiotami należącymi do różnych społeczności, do ustalenia konsensusu w społeczności, która musi przyjąć nowy termin lub uprościć termin w celu wyjaśnienia jego znaczenia komuś nowemu w społeczności .

Klasyfikacja według wyrazistości

Ontologia ciężka : Ontologie o dużej masie są silnie aksjomatyzowane, a zatem wyraźnie reprezentują zobowiązanie ontologiczne. Celem aksjomatyzacji jest wykluczenie niejasności terminologicznych i pojęciowych wynikających z niezamierzonych interpretacji.

Lekka ontologia : Lekkie ontologie to proste struktury taksonomiczne terminów pierwotnych lub złożonych wraz z powiązаныmi definicjami. Trudno je aksjomatyzować, ponieważ zamierzone znaczenie terminów używanych przez społeczność jest mniej lub bardziej znane z góry wszystkim członkom, a ontologię można ograniczyć do tych strukturalnych relacji między terminami, które są uważane za istotne.

Klasyfikacja według specyfiki

Ontologia ogólna : Pojęcia zdefiniowane przez tę warstwę są uważane za ogólne w wielu dziedzinach. Zazwyczaj ontologie ogólne definiują pojęcia, takie jak stan, zdarzenie, proces itp.

Ontologia rdzenia : Ontologie podstawowe definiują pojęcia, które są ogólne w zbiorze domen. Dlatego znajdują się pomiędzy dwoma skrajnościami ontologii rodzajowej i domenowej. Granica między ontologiami rodzajowymi i rdzeniowymi nie jest jasno określona, ponieważ nie ma wyczerpującego wyliczenia pól i ich konceptualizacji. Jednak to rozróżnienie jest intuicyjnie znaczące i przydatne przy budowaniu bibliotek.

Ontologia domeny : Ontologie dziedzinowe wyrażają konceptualizacje specyficzne dla konkretnego uniwersum dyskursu. Pojęcia w ontologiach domeny są często definiowane jako specjalizacja pojęć w ontologiach ogólnych i rdzeniowych. Granica między ontologiami rdzeniowymi i domenowymi nie jest jasno określona, ponieważ ontologie podstawowe mają być generyczne w ramach domeny. W związku z tym zwykle trudno jest dokonać wyraźnego podziału między ontologiami rodzajowymi i rdzeniowymi, a także między ontologiami rdzeniowymi i dziedzinowymi. Pojęcia takie jak komponent oprogramowania zostałyby umieszczone w rdzeniowej ontologii serwerów aplikacji.

Dlaczego warto rozwijać ontologię?

Niektóre z powodów rozwoju ontologii to:

- Aby dzielić się wspólnym rozumieniem struktury informacji wśród ludzi lub agentów oprogramowania.
- Aby umożliwić ponowne wykorzystanie wiedzy domeny.
- Sprecyzować założenia domeny.
- Oddzielenie wiedzy dziedzinowej od wiedzy operacyjnej.
- Analiza wiedzy dziedzinowej.

Dzielenie się wspólnym rozumieniem struktury informacji wśród ludzi lub agentów oprogramowania:

Jest to jeden z najczęstszych celów w rozwoju ontologii. Załóżmy na przykład, że kilka różnych witryn internetowych zawiera informacje medyczne lub świadczy usługi e-handlu medycznego. Jeśli te strony internetowe współużytkują i publikują tę samą podstawową ontologię terminów, których używają, wówczas agenci komputerowi mogą wydobywać i agregować informacje z tych różnych witryn. Agenci mogą wykorzystywać te zagregowane informacje do odpowiadania na zapytania użytkowników lub jako dane wejściowe do innych aplikacji.

Umożliwienie ponownego wykorzystania wiedzy domeny:

To była jedna z sił napędowych niedawnego gwałtownego wzrostu badań ontologicznych. Na przykład modele dla wielu różnych dziedzin muszą reprezentować pojęcie czasu. Ta reprezentacja obejmuje pojęcia przedziałów czasu, punktów w czasie, względnych miar czasu i tak dalej. Jeśli jedna grupa badaczy opracuje szczegółowo ontologię, inni mogą ją po prostu ponownie wykorzystać w swoich dziedzinach. Dodatkowo, jeśli potrzebujemy zbudować dużą ontologię, możemy zintegrować kilka istniejących ontologii opisujących części dużej domeny. Możemy również ponownie użyć ogólnej ontologii, takiej jak ontologia UNSPSC, i rozszerzyć ją, aby opisała naszą dziedzinę, która nas interesuje.

Dokonywanie jednoznacznych założeń dotyczących domeny

Jawne założenia dziedzinowe, na których opiera się implementacja, umożliwiają łatwą zmianę założeń, jeśli zmieni się nasza wiedza o domenie. Zakodowanie na sztywno założeń dotyczących świata w kodzie języka programowania sprawia, że założenia te są nie tylko trudne do znalezienia i zrozumienia, ale także trudne do zmiany, szczególnie dla kogoś bez wiedzy programistycznej. Ponadto wyraźne określenie wiedzy domeny jest przydatne dla nowych użytkowników, którzy muszą poznać znaczenie terminów w domenie.

Oddzielenie wiedzy dziedzinowej od operacyjnej

Potrafimy opisać zadanie konfiguracji produktu z jego komponentów, zgodnie z wymaganą specyfikacją i zaimplementować program, który dokonuje tej konfiguracji niezależnie od samych produktów i komponentów. Następnie możemy opracować ontologię komponentów PC i właściwości i zastosować algorytm do konfiguracji komputerów na zamówienie. Możemy również użyć tego samego algorytmu do skonfigurowania wind, jeśli wprowadzimy do niego ontologię komponentu windy.

Analiza wiedzy dziedzinowej

Analiza wiedzy dziedzinowej jest możliwa po udostępnieniu deklaratywnego określenia warunków. Formalna analiza terminów jest niezwykle cenna przy próbie ponownego wykorzystania istniejących ontologii i przy ich rozszerzaniu. Często ontologia domeny nie jest celem sama w sobie. Opracowywanie ontologii jest podobne do definiowania zbioru danych i ich struktury do wykorzystania przez inne programy. Metody rozwiązywania problemów, aplikacje niezależne od domeny i agenty oprogramowania wykorzystują ontologie i bazy wiedzy zbudowane przy użyciu ontologii jako danych.

Cykl rozwoju ontologii

Podczas budowania ontologii pomocne będą następujące pytania, aby zrozumieć, gdzie rozpocząć i zakończyć proces rozwoju ontologii.

- Jakie działania są zaangażowane w proces rozwoju ontologii?
- Jaki jest cel każdego działania?
- Kiedy powinienem przeprowadzić każde ćwiczenie?

- Jaki jest związek jednej czynności z innymi?
- Gdzie mogę znaleźć ontologie w celu ich ponownego wykorzystania?
- Jak mogę zbudować ontologię dla mojej aplikacji?
- Czy potrzebuję pojedynczej ontologii czy sieci ontologii?

Poniżej wymieniono różne etapy związane z cyklem rozwoju ontologii:

Określ cel i zakres:

Każdy projekt, który ma być rozwijany z wykorzystaniem ontologii jako podstawy, będzie miał swój zakres. Można to dobrze zrozumieć, porównując ten etap z etapem analizy wymagań w tworzeniu aplikacji. Opracowanie specyfikacji wymagań dla ontologii poprzez zidentyfikowanie zamierzonego zakresu i celu ontologii jest początkowym etapem cyklu życia rozwoju ontologii. Dobrze scharakteryzowana specyfikacja wymagań jest ważna dla projektowania, oceny i ponownego wykorzystania ontologii.

Zdobywanie wiedzy:

Akwizycję wiedzy można zdefiniować jako proces zdobywania wiedzy dziedzinowej, przy pomocy którego można budować ontologię. Identyfikowane i łączone są różne możliwe scenariusze. Za jego pomocą tworzone są nieformalne pytania kompetencyjne.

Konceptualizacja:

Konceptualizację można zdefiniować jako proces identyfikacji kluczowych pojęć, które istnieją w domenie, ich właściwości i relacji, które między nimi zachodzą; identyfikowanie terminów języka naturalnego w celu odniesienia się do takich pojęć, relacji i atrybutów oraz strukturyzowanie wiedzy dziedzinowej w wyraźne modele pojęciowe. Jest to proces poruszony na początku tego rozdziału, w którym zostały uchwycone pojęcia i relacje opisujące tę dziedzinę. Ontologia jest zwykle opisywana przy użyciu nieformalnej terminologii. Gruber sugeruje spisanie list pojęć, które mają zostać włączone do ontologii i zbadanie innych ontologii w celu ponownego wykorzystania całości lub części ich konceptualizacji i terminologii.

Kodowanie:

Kodowanie to proces przedstawiania konceptualizacji w jakimś języku formalnym, np. ramki, modele obiektów lub logika. Obejmuje to tworzenie formalnych pytań kompetencyjnych pod kątem wybranego języka specyfikacji terminologicznej (zwykle logiki pierwszego rzędu).

Integracja:

Integracja oznacza użycie lub specjalizację w istniejącej ontologii. Zadanie to jest często utrudnione z powodu nieodpowiedniej dokumentacji istniejących ontologii, zwłaszcza ich niejawnych założeń. Używając generycznej ontologii, daje głębszą definicję pojęć w wybranej dziedzinie.

Dokumentacja:

Dokumentacja zawiera nieformalne i formalne kompletne definicje, założenia i przykłady, które są niezbędne do promowania właściwego stosowania i ponownego wykorzystywania ontologii. Dokumentacja jest ważna dla określenia dokładnego znaczenia terminów w ontologii.

Ocena:

Ewaluacja to określenie adekwatności ontologii do jej zamierzonego zastosowania. Ocena odbywa się pragmatycznie, poprzez ocenę kompetencji ontologii do spełnienia wymagań jej zastosowania, w tym określenia spójności, kompletności i zwięzłości ontologii. Zwięzłość oznacza brak redundancji w definicjach ontologii i odpowiednią szczegółowość.

Wykorzystanie ontologii

Następujące niezależne od aplikacji czynności cyklu życia są wykonywane w czasie wykonywania w celu obsługi ontologii:

Populacja ontologii:

Aby wypełnić bazę wiedzy (KB), instancje mogą być pobierane od użytkownika, np. za pośrednictwem formularzy. Jeśli wszystkie wystąpienia danych muszą być utworzone ręcznie, na użytkownika może zostać nałożony znaczny narzut. To obciążenie może zostać złagodzone przez (pół) -automatyczną populację KB. Część tego etapu tworzenia wiedzy obejmuje również manipulację i usuwanie instancji.

Oczyszczanie i fuzja:

Nie można założyć, że wiedza uzyskana automatycznie ma pożądaną jakość. Ulepszanie instancji danych może obejmować identyfikację i łączenie koncepcyjnie identycznych instancji, które są różnie oznakowane (identyfikacja obiektów).

Zobowiązania ontologiczne

Aby móc dzielić się wiedzą między agentami, musi istnieć porozumienie co do przekazywanych tematów. Rodzi to kwestię zaangażowania ontologicznego. Na to pozwalają zobowiązania ontologiczne pewnej liczbie agentów, aby w znaczący sposób komunikować się na temat domeny, niekoniecznie działając w oparciu o ogólnoswiatową teorię. W kontekście wielu agentów, wspólna ontologia służy jako specyfikacja na poziomie wiedzy zobowiązań ontologicznych zbioru uczestniczących agentów. Wspólna ontologia definiuje słownictwo, za pomocą którego zapytania i twierdzenia są wymieniane między agentami, zapewniając w ten sposób środki do wypełnienia semantycznej luki istniejącej między leksykalnymi reprezentacjami informacji a jej nieleksykalną konceptualizacją.

Uczenie się ontologii

Uczenie się ontologii definiuje się jako zbiór metod i technik wykorzystywanych do budowania ontologii od podstaw, wzbogacania lub dostosowywania istniejącej ontologii w sposób półautomatyczny z wykorzystaniem rozproszonej i heterogenicznej wiedzy i źródeł informacji, co pozwala na redukcję czasu i wysiłku potrzebnego w ontologii proces rozwoju. Chociaż w pełni automatyczne zdobywanie wiedzy pozostaje niedostępne, cały proces jest uważany za półautomatyczny, tj. W niektórych częściach procesu uczenia się konieczna jest interwencja człowieka. Nowe koncepcje są identyfikowane przy użyciu technik analizy języka naturalnego w odniesieniu do zasobów wcześniej zidentyfikowanych przez użytkownika. Wynikowa ontologia jest przycinana, a następnie skupiana na określonej dziedzinie za pomocą kilku podejść opartych na statystykach. Wreszcie związek między pojęciami ustala się poprzez zastosowanie metod uczenia się.

Dopasowanie ontologii

Dopasowanie ontologii polega na ustanowieniu różnego rodzaju odwzorowań (lub połączeń) między dwiema ontologiami, a tym samym zachowaniu oryginalnych ontologii. Scalanie ontologii proponuje wygenerowanie unikalnej ontologii z oryginalnych ontologii. Przyjmujemy, że mapowanie między

ontologiami oznacza przepisanie zbioru reguł wiążących terminy i wyrażenia zdefiniowane w ontologii źródłowej z terminami i wyrażeniami docelowej ontologii.

Zalety ontologii

Dobra ontologia oferuje złożony zestaw korzyści, które nie są dostępne w taksonomiach, schemacie relacyjnej bazy danych ani w innych standardowych sposobach strukturyzacji informacji. Niektóre z tych korzyści to:

- Ontologie promują spójną nawigację, umożliwiając przejście od koncepcji do koncepcji w strukturze ontologii.
- Mają elastyczne punkty wejścia, ponieważ każdą określoną perspektywę w ontologii można prześledzić i powiązać ze wszystkimi powiązаныmi z nią koncepcjami; nie ma określonej struktury zbioru ani sposobu interakcji z ontologią.
- Połączenia wyróżniają powiązane informacje i pomagają w szybkim wyszukiwaniu bez konieczności posiadania wcześniejszej znajomości domeny lub terminologii.
- Zdolność do reprezentowania dowolnej formy informacji, w tym danych nieustrukturyzowanych (powiedzmy, dokumenty lub tekst), częściowo ustrukturyzowanych (powiedzmy, XML lub strony internetowe) i ustrukturyzowanych (powiedzmy, konwencjonalne bazy danych).
- Wnioskowanie, dzięki któremu określając jedno pojęcie (powiedzmy, ssaki) wiemy, że odnosimy się również do pojęcia pokrewnego (powiedzmy, że ssaki są rodzajem zwierzęcia).
- Dopasowanie pojęć, co oznacza, że nawet jeśli możemy opisywać rzeczy nieco inaczej, nadal możemy dopasować do tego samego pomysłu (np. Zadowolony lub szczęśliwy, oba odnoszące się do koncepcji przyjemnego stanu umysłu).
- Oznacza to zatem, że możemy również zintegrować treści zewnętrzne poprzez odpowiednie dopasowanie i mapowanie tych pojęć.
- Rozumowanie, czyli umiejętność wykorzystania spójności i struktury jej w celu uzyskania informacji na temat pokrewieństwa lub udzielenia odpowiedzi na pytania; ta ostatnia korzyść jest bardziej związana z uczeniem maszynowym lub sztuczną inteligencją i ogólnie nie jest wyrażana w prostszych, standardowych ontologiach.

Ograniczenia ontologii

Chociaż ontologia przyczynia się do rozwoju sieci semantycznej, ma również pewne ograniczenia.

- Ontologia tworzy abstrakcyjny model określonej domeny w oparciu o zbiór danych i struktur, ale nie określa granic modelu.
- Rozmiar ontologii różni się w zależności od liczby klas i instancji; jeśli liczba instancji zostanie znacznie zwiększona, bardzo trudno będzie zarządzać nią ręcznie, a obecnie nie ma mechanizmu do automatycznego zarządzania.
- Proces ręcznego generowania ontologii staje się czasami bardzo złożony i czasochłonny, szczególnie w przypadku dużej ilości danych.

RDF i SPARQL

Cel:

Gdy dane są przechowywane w bazie danych, oczywistym jest, że należy je odpytywać, aby je pobrać i wykorzystać w aplikacji. W tej sekcji omówiono model danych grafów, tj. RDF i SPARQL, który jest językiem zapytań dla RDF.

Wprowadzenie do RDF

RDF to skrót od Resource Description Framework. RDF to rodzina specyfikacji World Wide Web Consortium (W3C), pierwotnie zaprojektowana jako model danych metadanych. Jest używana jako ogólna metoda opisu koncepcyjnego lub modelowania informacji, które są implementowane w zasobach sieci Web, przy użyciu różnych notacji składni i formatów serializacji danych.

Wyjaśnienie

RDF to format, w którym zapisywany jest model danych wykresu używany przez sieć semantyczną do przechowywania danych. RDF to ogólna metoda rozkładania wiedzy na małe części, z pewnymi regułami dotyczącymi semantyki lub znaczenia tych fragmentów. Chodzi o to, aby mieć metodę tak prostą, że może wyrazić każdy fakt, a jednocześnie tak skonstruowaną, że aplikacje komputerowe mogą robić użyteczne rzeczy z wiedzą wyrażoną w RDF. Nazywa się to w szczególności "metodą", a nie formatem, ponieważ można zapisać te fragmenty na wiele sposobów i nadal zachować oryginalne informacje i strukturę, tak jak można wyrazić to samo znaczenie w różnych ludzkich językach lub wdrożyć tę samą strukturę danych na wiele sposobów. RDF można zdefiniować za pomocą trzech prostych reguł:

- Fakt jest wyrażony jako potrójna część formy (podmiot, orzeczenie i przedmiot). To prawie jak angielskie zdanie.
- Podmioty, predykaty i obiekty to nazwy bytów, zarówno konkretne, jak i abstrakcyjne, w świecie rzeczywistym. Nazwy to:
 - 1) Globalne i odnoszą się do tego samego podmiotu w każdym dokumencie RDF, w którym się pojawiają, lub
 - 2) Lokalny i podmiot, do którego się odnosi, nie może być bezpośrednio powiązany poza dokumentem RDF.
- Obiekty mogą być również wartościami tekstowymi, zwanymi wartościami dosłownymi.

Ponowne odwiedzanie XML

XML to skrót od Extensible Markup Language. XML to specyfikacja dokumentów do odczytu komputerowego. Znacznik oznacza, że określone sekwencje znaków w dokumencie zawierają informacje wskazujące na rolę treści dokumentu. Znaczniki opisują układ danych dokumentu i strukturę logiczną oraz sprawiają, że informacje są w pewnym sensie samoopisujące. Przyjmuje formę słów między tagami - na przykład <nazwa> lub <age>.

Pod tym względem XML wygląda bardzo podobnie do dobrze znanego języka HTML. Przykład:

```
<? xml version = „1.0”?>
```

```
<pracownicy>
```

Lista osób w firmie:

```
<person name = „Nandini”>
```

```
<phone> 123456789 </phone>
```

</person>

</employees>

XML nie implikuje określonej interpretacji danych. Oczywiście, ze względu na nazwy znaczników, znaczenie poprzedniego fragmentu XML wydaje się oczywiste dla ludzi, ale nie jest formalnie określone. Jediną uzasadnioną interpretacją jest to, że kod XML zawiera nazwane jednostki z podelementami i wartościami; to znaczy, każdy dokument XML tworzy uporządkowane drzewo z etykietami. Ta ogólność jest zarówno mocną, jak i słabą stroną XML. Możesz zakodować wszystkie rodzaje struktur danych w jednoznacznej składni, ale XML nie określa zastosowania i semantyki danych. Strony, które używają XML do wymiany danych, muszą wcześniej uzgodnić słownictwo, jego użycie i znaczenie.

DTD i schemat XML

DTD to skrót od definicji typu dokumentu. Schematy DTD i XML częściowo odpowiedziały na wady XML. Chociaż DTD i schematy XML nie określają znaczenia danych, określają nazwy elementów i atrybutów (słownictwo) oraz ich zastosowanie w dokumentach. Oba są mechanizmami, za pomocą których można określić strukturę dokumentów XML. Następnie można sprawdzić poprawność określonych dokumentów w odniesieniu do zaleceń struktury określonych w DTD lub schemacie XML. DTD określają dozwolone zagnieżdżanie elementów, możliwe atrybuty elementu i lokalizacje, w których dozwolony jest zwykły tekst. Na przykład DTD może nakazać, że każdy element „person” musi mieć atrybut „name” i może mieć element potomny o nazwie „phone”, którego treść musi być tekstem. Składnia DTD wygląda nieco niezręcznie, ale w rzeczywistości jest dość prosta. Schematy XML są proponowanym następcą DTD. Definicja schematu XML jest nadal kandydatem na zalecenie W3C (World Wide Web Consortium), co oznacza, że chociaż jest uważane za stabilne, może nadal podlegać niewielkim zmianom. Schematy XML mają kilka zalet w porównaniu z DTD. Po pierwsze, mechanizm schematu XML zapewnia bogatszą gramatykę do określania struktury elementów. Na przykład możesz określić dokładną liczbę dozwolonych wystąpień elementów podrzędnych, możesz określić wartości domyślne i możesz umieścić elementy w grupie wyboru, co oznacza, że dokładnie jeden z elementów w tej grupie jest dozwolony w określonej lokalizacji. Po drugie, zapewnia wpisywanie danych. Ostatnią różnicą w stosunku do DTD jest to, że w przepisach schematu XML jako składni kodowania używany jest język XML. Upraszcza to tworzenie narzędzi, ponieważ zarówno zalecenie dotyczące struktury, jak i wymagane dokumenty używają tej samej składni. Twórcy specyfikacji XML Schema wykorzystali tę funkcję, używając dokumentu XML Schema do zdefiniowania klasy dokumentów XML Schema. W końcu, ponieważ recepta schematu XML jest aplikacją XML, musi być zgodna z regułami dotyczącymi jej struktury, które można zdefiniować za pomocą innej recepty schematu XML. Jednak ta rekurencyjna definicja może być nieco myląca.

Jednostki XML

Encja XML może pełnić kilka ról, takich jak symbol zastępczy dla powtarzalnych znaków (rodzaj skrótu), sekcja danych zewnętrznych (np. XML lub inne) lub jako część deklaracji elementów. Załóżmy na przykład, że dokument zawiera kilka informacji o prawach autorskich, które odnoszą się do bieżącego roku. Wtedy warto zadeklarować jednostkę <! ENTITY w tym roku „2007”>. Następnie w każdym miejscu, w którym należy uwzględnić bieżący rok, możemy użyć odniesienia do encji & thisyear; zamiast. W ten sposób aktualizacja wartości roku na „2008” dla całego dokumentu będzie oznaczała jedynie zmianę deklaracji podmiotu.

Po co zwracać sobie głowę XML w RDF

XML zapewnia składnię do kodowania danych. Oznacza to, że struktura opisu zasobów jest mechanizmem służącym do informowania o danych. Jak nazwa wskazuje, nie jest to język, ale model reprezentacji danych o „rzeczach w sieci”. Ten typ danych o danych nazywa się metadanymi. „Rzeczy” są zasobami w słowniku RDF. Podstawowy model danych RDF jest prosty: oprócz zasobów zawiera właściwości i instrukcje. Właściwość to określony aspekt, cecha, atrybut lub relacja opisująca zasób. Instrukcja składa się z określonego zasobu z nazwaną właściwością oraz wartości tej właściwości dla tego zasobu. Ta wartość może być innym zasobem lub wartością literalną, taką jak dowolny tekst. W sumie opis RDF to lista trójek: obiekt (zasób), atrybut (właściwość) i wartość (zasób lub dowolny tekst).

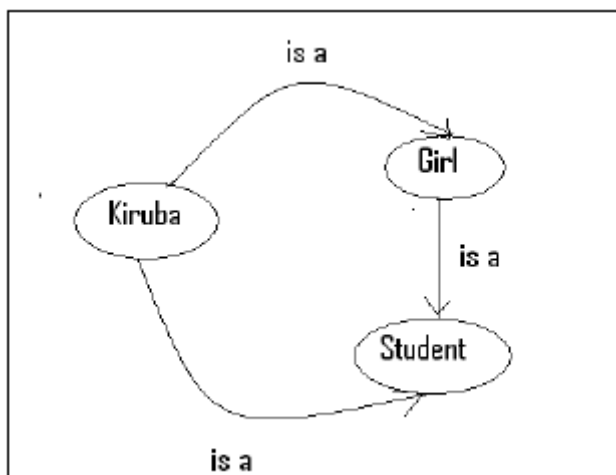
Uwaga: Ludzie często uważają, że RDF i XML to to samo. Ale faktem jest, że RDF jest modelem danych, który można przedstawić w XML (RDF / XML). RDF to wykresowy model danych, który wykorzystuje identyfikatory URI, podczas gdy XML jest drzewnym modelem danych i nie przejmuje się identyfikatorami URI.

Schemat RDF

RDF Schema (RDF) jest rozszerzeniem słownictwa RDF, umożliwiającym taksonomiczny opis klas i ich właściwości. Rozszerza również definicje niektórych elementów RDF, na przykład określa dziedzinę i zakres właściwości oraz łączy klasy i właściwości RDF z taksonomiami przy użyciu słownika RDFS.

Wykres RDF

RDF można zdefiniować jako koncepcję dwuwymiarową. Może być postrzegany jako wykres lub zestaw zdań. Zaczniemy od tego pierwszego. Zgodnie z klasyczną definicją, wykres jest reprezentacją zbioru obiektów, w których niektóre pary obiektów są połączone linkami. Połączone obiekty są reprezentowane przez matematyczne abstrakcje zwane wierzchołkami, a połączenia, które łączą niektóre pary wierzchołków, nazywane są krawędziami.



Nazwa na początku strzałki to temat instrukcji, nazwa na końcu strzałki to obiekt instrukcji, a nazwa, która zawiera etykietę strzałki, to predykat. RDF jako wykres wyraża dokładnie te same informacje, co RDF zapisane jako potrójne, ale forma wykresu ułatwia nam zobaczenie struktury danych. Stąd możemy wywnioskować:

Subject = (Kiruba, Girl)

Object = (Girl, Student)

Predicate = (is a)

Podstawy RDF

Zasoby

Możemy myśleć o zasobie jako o przedmiocie, „rzeczy”, o której chcemy porozmawiać. Zasobami mogą być autorzy, książki, wydawcy, miejsca, ludzie, hotele, pokoje, wyszukiwane hasła itd. Każdy zasób ma URI, jednolity identyfikator zasobu. Identyfikatorem URI może być adres URL (Uniform Resource Locator lub adres sieci Web) lub inny rodzaj unikalnego identyfikatora (należy pamiętać, że identyfikator niekoniecznie umożliwi dostęp do zasobu). Schematy URI zostały zdefiniowane nie tylko dla lokalizacji internetowych, ale także dla tak różnych obiektów, jak numery telefonów, numery ISBN i lokalizacje geograficzne. Odbyła się długa dyskusja na temat natury identyfikatorów URI, dotycząc nawet kwestii filozoficznych (na przykład, jaki jest odpowiedni unikalny identyfikator osoby?). Ale nie będziemy tutaj wchodzić w szczegóły. Ogólnie zakładamy, że identyfikator URI jest identyfikatorem zasób sieciowy.

Właściwości

Nieruchomości to szczególny rodzaj zasobów. Opisują relacje między zasobami, na przykład „napisane przez”, „wiek”, „tytuł” i tak dalej. Właściwości w RDF są również identyfikowane przez URI (i w praktyce przez adresy URL). Pomysł wykorzystania identyfikatorów URI do identyfikowania „rzeczy” i relacji między nimi jest dość ważny. Ten wybór za jednym pociągnięciem daje nam globalny, światowy, unikalny schemat nazewnictwa. Zastosowanie takiego schematu znacznie ogranicza problem homonimów, który do tej pory nękał rozproszoną reprezentacją danych.

Sprawozdania:

Instrukcje potwierdzają właściwości zasobów. Instrukcja to obiekt, potrójna wartość atrybutu, składająca się z zasobu, właściwości i wartości. Wartości mogą być zasobami lub literałami. Literały to wartości atomowe (łańcuchy), których struktury nie będziemy dalej omawiać.

Konstruowanie RDF

RDF zapewnia ogólną, elastyczną metodę rozłożenia wiedzy na małe fragmenty, zwane trojkami, z pewnymi regułami dotyczącymi semantyki (znaczenia) tych fragmentów. Podstawę kładzie się poprzez rozbitcie wiedzy na tak zwany ukierunkowany wykres z etykietami, jak omówiono w poprzedniej sekcji.

Potrójne RDF

Zacznijmy od standardowej składni trójek RDF. Poniżej podano standardową składnię

```
<rdf:Description rdf:about="subject">
```

```
<predicate rdf:resource="object" />
```

```
<predicate>literal value</predicate>
```

```
<rdf:Description>
```

Aby właściwie zrozumieć tę koncepcję, zacznijmy od przykładu. Rozważmy następujący przykład

Przykład 1:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<rdf:RDF
```

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:feature="http://www.bookboon.com/cartoon-features#">
<rdf:Description rdf:about="http://www.bookboon.com/cartoon#mickey-mouse">
<feature:character rdf:resource="http://www.bookboon.com/character#funny"/>
</rdf:Description>
</rdf:RDF>
```

Wyjaśnienie:

RDF potrójny

W przykładzie 1 RDF między tagami <rdf: Description> jest nazywany instrukcją RDF lub potrójną wersją RDF. W potrójnym RDF, jak sama nazwa wskazuje, stwierdzenie jest podzielone na trzy części:

- Przedmiot
- Obiekt
- Predykat wypowiedzi

Konstruowanie instrukcji w RDF:

Dokument RDF może zawierać więcej niż jedną instrukcję. Znacznik rdf: Description opisuje temat i nadaje mu unikalny identyfikator, a mianowicie <http://www.bookboon.com/cartoon#mickey-mouse>. Stąd tutaj temat to „Myszka Miki”.

Konstruowanie predykatu w RDF:

Stwierdzenia RDF opisują cechy ich przedmiotów za pomocą właściwości lub predykatów w terminologii RDF. Temat ma właściwość z nazwą: postać, która ma wartość dosłowną „zabawny”, tj. Postać z kreskówki Myszka Miki ma zabawną cechę. Zatem tutaj predykatem jest „znak”.

Konstruowanie obiektu w RDF:

Cecha, czyli charakter, w

```
„<Feature: character rdf: resource =“ http://www.bookboon.com/character#funny ”/>”
```

, odnosi się do tematu (ID) innego stwierdzenia. Oznacza to, że obiekty w RDF mogą odnosić się do podmiotów innych stwierdzeń. Przedmiot tutaj jest „zabawny”.

Wprowadzenie do SPARQL

SPARQL oznacza „protokół SPARQL i język zapytań RDF”. Podobnie jak w przypadku zapytań w tabelach relacyjnej bazy danych przy użyciu języka SQL, zapytania dotyczące trzech danych RDF są wykonywane przy użyciu protokołu SPARQL. SPARQL to język zapytań zaprojektowany specjalnie do wykonywania zapytań w bazach danych RDF. Zapytania SPARQL są wysyłane od klienta do usługi znanej jako punkt końcowy SPARQL przy użyciu protokołu HTTP. Interakcja między klientem a punktem końcowym jest zdefiniowany w przyjaznym dla maszyny protokole, który nie jest przeznaczony do interpretacji przez ludzi, dlatego użycie SPARQL wymaga interfejsu, który pozwala użytkownikowi wprowadzać zapytania i wyświetlać wyniki w znaczący sposób. Podobnie jak w przypadku tradycyjnych języków baz danych, takich jak SQL, interfejsy są zwykle konstruowane w taki sposób, że zapytania są konstruowane i

uruchamiane za pomocą formularzy, które nie wymagają od użytkownika znajomości RDF i SPARQL. Oprócz języka W3C zdefiniowało również:

- Protokół SPARQL dla specyfikacji RDF: definiuje zdalny protokół do wysyłania zapytań SPARQL i otrzymywania wyników.
- Specyfikacja formatu XML wyników zapytań SPARQL: definiuje format dokumentu XML do reprezentowania wyników zapytań SPARQL.

SQL

SQL to skrót od Structured Query Language. SQL to język komputerowy do przechowywania, manipulowania i pobierania danych przechowywanych w relacyjnej bazie danych. Student inżynierii komputerowej uzna ten termin za zbyt znajomy. Dzieje się tak, ponieważ większość istniejących aplikacji używa języka SQL do wykonywania zapytań w bazie danych. SQL jest standardowym językiem dla systemu relacyjnych baz danych. Wszystkie systemy zarządzania relacyjnymi bazami danych, takie jak MySQL, MS Access, Oracle, Sybase, Informix, postgres i SQL Server używają SQL jako standardowego języka bazy danych. Po SQL następuje unikalny zestaw reguł i wskazówek zwany składnią. Wszystkie instrukcje SQL zaczynają się od słów kluczowych, takich jak SELECT, INSERT, UPDATE, DELETE, ALTER, DROP, CREATE, USE, SHOW, a całe instrukcje kończą się średnikiem. Omówmy niektóre z ważnych słów kluczowych w SQL, które będą przydatne w zrozumieniu SPARQL:

CREATE

Tworzenie podstawowej tabeli obejmuje nazwanie tabeli i zdefiniowanie jej kolumn oraz typu danych każdej kolumny. Instrukcja SQL CREATE TABLE służy do tworzenia nowej tabeli. Przykład:

Utwórz klienta tabeli (

Emp_no int not null,

Nazwa varchar (25)

);

SELECT

Instrukcja SQL SELECT służy do pobierania danych z tabeli bazy danych, która zwraca dane w postaci tabeli wynikowej. Te tabele wynikowe nazywane są zestawami wyników.

Przykład:

Wybierz * z customer_information;

Spowoduje to pobranie wszystkich danych z tabeli „customer_information”.

INSERT

Instrukcja SQL INSERT służy do wstawiania danych do tabeli bazy danych.

Przykład:

Wstaw do wartości customer_information (A, B, C);

Spowoduje to wstawienie danych do tabeli „customer_information”.

SPARQL Vs SQL

Wiele osób pyta, co można zrobić za pomocą SPARQL, czego nie można zrobić za pomocą SQL. Zarówno SQL, jak i SPARQL dają użytkownikowi dostęp do tworzenia, łączenia i wykorzystywania ustrukturyzowanych danych. SQL robi to, uzyskując dostęp do tabel w relacyjnych bazach danych, a SPARQL robi to, uzyskując dostęp do sieci danych połączonych. RDF przechwytyje zarówno atrybuty encji, jak i relacje między jednostkami, jako oświadczenia w postaci „jednostka1 ma właściwość .Powiązanie z jednostką2” w języku o nazwie Turtle. Można powiedzieć, że jest osoba o imieniu „Mickey” z adresem w Disney w USA, jak poniżej:

```
<PersonaA> a <Persona>.
```

```
<AddressB> a <Address>.
```

```
<PersonA> <Person # fname> „Mickey”.
```

```
<AddressB> <Address # city> „Disney”.
```

```
<PersonA><Person#addr> <AddressB>.
```

```
<AddressB> <Address # state> „USA”.
```

Możemy też mieć inną osobę, „Plutona”, ale nie powiemy nic o jej adresie, ponieważ go nie znamy. Stąd reprezentujemy w następujący sposób:

```
<PersonF> a <Person>.
```

```
<Person> <Person # fname> „Pluton”.
```

Terminy użyte w powyższych stwierdzeniach to względne adresy URL w nawiasach ostrych (<> s), literały w cudzysłowie („” s) i słowo kluczowe „a”, które jest tylko skrótem do adresu URL używanego do identyfikacji „ma typ” związek. W RDF nie ma koncepcji odpowiadającej wartości NULL języka SQL, ponieważ nie ma wymagania RDF odpowiadającego ograniczeniu strukturalnemu SQL, że każdy wiersz w relacyjnej bazie danych musi być zgodny z tym samym schematem. Przedmiot jednego stwierdzenia, np. <AddressB> powyżej, może być podmiotem lub przedmiotem innych twierdzeń. W ten sposób zestaw instrukcji RDF łączy się, tworząc „wykres” (w sensie matematycznym). Często można usłyszeć termin „wykres RDF”. Te wykresy mogą być cykliczne, np. Stwierdzające, że Mickey mieszka w jakimś miejscu, którego jest również właścicielem:

```
<PersonC><Person#homeAddress> <AddressK>.
```

```
<PersonC> <Person # fname> „Mickey”.
```

```
<AddressK><Address#owner> <PersonC>.
```

Powyższe przykłady ilustrują niektóre strukturalne podobieństwa i różnice między RDF a danymi relacyjnymi. Podstawowa różnica filozoficzna polega na tym, że RDF jest językiem post-internetowym; to znaczy, pozwala używać identyfikatorów internetowych dla jednostek, które chcemy opisać, oraz dla atrybutów i relacji, których używamy do ich opisu. Jeśli ufam, że wydawcy nie okłamują mnie, mogą połączyć informacje z różnych imprez trywialnych. Na koniec rozważ następujący przykład:

- SELECT Person.fname, Address.city FROM Person, Address WHERE Person.addr = Address.

```
ID AND Address.state = „Mumbai”
```

Koncepcyjnie wybieramy listę atrybutów z zestawu tabel, w których spełnione są pewne ograniczenia. Te ograniczenia wychwytyją relacje niejawne w schemacie, Person.addr = Addresses.ID i kryteria wyboru, np. Address.state = „Mumbai”

Zapytanie SPARQL dotyczące tych samych danych mogłoby wyglądać następująco:

```
SELECT? Name? City
WHERE{
? who <Person # fname>? name;
<Person # addr>? Adr.
? adr <Adres # miasto>? miasto;
<Address # state> „Mumbai”
}
```

Co więcej, SPARQL ponownie wykorzystuje niektóre słowa kluczowe znane użytkownikom SQL: SELECT, FROM, WHERE, UNION, GROUP BY, HAVING i większość nazw funkcji agregujących.

Konstruowanie zapytania SPARQL

Język zapytań SPARQL jest oparty na dopasowanych wzorcach wykresów. Najprostszym wzorcem grafowym jest potrójny wzorzec, który jest podobny do potrójnego RDF, ale z możliwością zmiennej zamiast terminu RDF w temacie, predykanie lub pozycjach obiektu. Połączenie potrójnych wzorców daje podstawowy wzorzec wykresu, w którym dokładne dopasowanie do wykresu jest potrzebne do wypełnienia wzorca. Jako prosty przykład rozważ następujące zapytanie:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT? C
WHERE
{
? c rdf: wpisz rdfs: Class.
}
```

To zapytanie pobiera wszystkie potrójne wzorce, w których właściwość to rdf: type, a obiekt to rdfs: Class. Innymi słowy, to zapytanie po wykonaniu zwróci wszystkie klasy.

Format konstruowania zapytania SPARQL:

PREFIX (Namespace Prefixes)

e.g. PREFIX plant: <http://www.linkeddatatools.com/plants>

SELECT (Result Set)

e.g. SELECT ?name.

FROM (Data Set)

e.g. FROM <http://www.linkeddatatools.com/plantsdata/plants.rdf>

WHERE (Query Triple Pattern)

e.g. WHERE (?planttype plant:planttype ?name)

ORDER BY, DISTINCT etc (Modifiers)

e.g. ORDER BY ?name

Przykład:

```
PREFIX school-info: <http://education.data.gov.india/def/school/>
```

```
SELECT ? Name WHERE {
```

```
? school a school-info: School.
```

```
? szkoła school-info: establishmentName? name.
```

```
? school school-info: districtAdministrative <http://statistics.data.gov.uk/id/local-authority-district/  
Bombaj>.
```

```
}
```

```
ORDER BY ? Imię
```

To zapytanie, po wykonaniu, zwróci nazwy wszystkich szkół w Indiach w okręgu administracyjnym „Bombaj” i uporządkuje wyniki w kolejności alfabetycznej. Znajomość SQL będzie bardzo przydatna w zrozumieniu formatu i konstrukcji SPARQL. Instrukcja SELECT żąda zwrócenia nazwy zmiennej. ? nazwa zwraca wszystkie nazwy szkół, które pasują do trzech wzorców wyszukiwania podanych w zapytaniu.

Protégé

Cel:

Protégé jest najbardziej znanym narzędziem w społeczności Semantic Web do budowania ontologii. Omówimy szczegółowo metodę tworzenia ontologii przy użyciu Protégé oraz sposoby eksploracji tej koncepcji klas i jednostek w ontologii.

Wprowadzenie do Protégé

Protege to narzędzie typu open source opracowane w Stanford Medical Informatics. Ma społeczność tysięcy użytkowników. Chociaż rozwój Protege w przeszłości był napędzany głównie przez zastosowania biomedyczne, system jest niezależny od domeny i był z powodzeniem używany również w wielu innych obszarach zastosowań. Podobnie jak większość innych narzędzi do modelowania, architektura Protege jest wyraźnie podzielona na część „modelową” i część „widokową”. Model

Protege jest wewnętrznym mechanizmem reprezentacji ontologii i baz wiedzy. Komponenty widoku Protege zapewniają interfejs użytkownika do wyświetlania i manipulowania podstawowym modelem. Model protegowanego oparty jest na prostym, ale elastycznym metamodelu, który jest porównywalny z systemami zorientowanymi obiektowo i opartymi na ramkach. Zasadniczo może reprezentować ontologie składające się z klas, właściwości (szczelin), właściwości właściwości (aspektów i ograniczeń) oraz instancji. Protege zapewnia otwarty interfejs API języka Java do wykonywania zapytań i manipulowania modelami. Ważną zaletą Protege jest to, że sam metamodel Protege jest ontologią Protege, z klasami reprezentującymi klasy, właściwości, i tak dalej. Na przykład, domyślna klasa w systemie podstawowym Protege nazywa się STANDARD-CLASS i ma takie właściwości, jak: NAME i: DIRECT-SUPERCLASSES. Taka struktura metamodelu umożliwia łatwe rozszerzenie i adaptację do innych reprezentacji. Na przykład ten metamodel został rozszerzony o obsługę UML i OWL. Korzystając z widoków interfejsu użytkownika Protege, projektanci ontologii zasadniczo tworzą klasy, przypisują właściwości do klas, a następnie ograniczają aspekty właściwości w niektórych klasach. Korzystając z powstałych ontologii, Protege jest w stanie automatycznie generować interfejsy użytkownika, które wspierają tworzenie jednostek (instancji). Dla każdej klasy w ontologii system tworzy jeden formularz z edytowalnymi komponentami (widżetami) dla każdej właściwości klasy. Na przykład dla właściwości, które mogą przyjmować pojedyncze wartości łańcuchowe, system domyślnie udostępnia widżet pola tekstowego. Wygenerowane formularze można dodatkowo dostosować za pomocą edytora formularzy Protege, w którym użytkownicy mogą wybrać alternatywne widżety interfejsu użytkownika dla swojego projektu. Oprócz predefiniowanej biblioteki widżetów interfejsu użytkownika Protege ma elastyczną architekturę, która umożliwia programistom tworzenie niestandardowych widżetów, które można następnie podłączyć do podstawowego systemu. Inny typ wtyczki obsługuje pełnowymiarowe panele interfejsu użytkownika (zakładki), które mogą zawierać dowolne inne komponenty. Oprócz zbioru standardowych zakładek do edycji klas, właściwości, formularzy i instancji istnieje biblioteka innych zakładek, które wykonują zapytania, uzyskują dostęp do repozytoriów danych, wizualizują ontologie graficznie i zarządzają wersjami ontologii. Obecnie Protégé może być używany do ładowania, edytowania i zapisywania ontologii w różnych formatach, w tym CLIPS, RDF, XML, UML i relacyjnych bazach danych. Niedawno dodano obsługę OWL. Ponieważ ontologie odegrały ważną rolę w aplikacjach sieci semantycznej, łatwo było wziąć za punkt wyjścia istniejące środowisko programistyczne ontologii. Rozszerzenia Protege mogą korzystać z ogólnych usług świadczonych przez podstawową platformę, takich jak mechanizm zdarzeń, możliwości cofania i mechanizm wtyczek. Opierając wtyczkę OWL na Protege, mogliśmy również ponownie wykorzystać tryb wielu użytkowników oparty na kliencie i serwerze Protege, który umożliwia wielu osobom edycję tej samej ontologii w tym samym czasie. Protege zapewnia również wysoce skalowalne zaplecze bazy danych, umożliwiające użytkownikom tworzenie ontologii z setkami tysięcy klas. Ponadto istnieje już pokaźna biblioteka wtyczek, które mogą być bezpośrednio użyte w OWL lub przystosowane do OWL przy niewielkim wysiłku. Co więcej, fakt, że Protege jest oprogramowaniem typu open source, również zachęca do tworzenia wtyczek. Wreszcie, Protege jest wspierany przez dużą społeczność aktywnych użytkowników i programistów, a opinie tej społeczności okazały się nieocenione dla rozwoju wtyczki OWL. Nasza decyzja o oparciu wtyczki OWL na Protege również wiązała się z pewnym ryzykiem. Aby móc ponownie wykorzystać jak najwięcej istniejących funkcji Protege, musieliśmy stworzyć staranne mapowanie między metamodelem Protege a OWL, które w miarę możliwości zachowuje tradycyjną semantykę Protege. Co więcej, żaden z ogólnych widżetów i zakładek Protege nie jest zoptymalizowany pod kątem OWL, a nie wszystkie metafory edycji dla systemów opartych na ramkach są odpowiednie dla OWL. W szczególności bogate funkcje logiki opisu OWL, takie jak definicje klas logicznych, wymagały szczególnej uwagi. Poniższe sekcje pokażą, jak rozwiązaliśmy te problemy.

Jak rozwinąć ontologię?

Rozwijanie ontologii w ujęciu teoretycznym zostało już omówione w poprzednim rozdziale. Poniższe kroki pomogą w praktycznym rozwinięciu ontologii:

1. Zdefiniować zajęcia z ontologii
2. Ułóż klasy w hierarchii podklasy-nadklasy
3. Zdefiniuj gniazda i opisz dozwolone wartości dla tych gniazd
4. Wypełnij wartości slotów dla instancji

Pliki w Protégé:

Kiedy używasz Protégé do tworzenia i edycji ontologii, wygenerujesz co najmniej dwa pliki:

- Plik projektu

Plik projektu ma rozszerzenie .pprj. Plik projektu przechowuje informacje związane z dowolnymi dostosowaniami interfejsu lub wybranymi opcjami edytora. W wielu przypadkach nie musisz wysyłać tego pliku do współpracowników ze swoją ontologią, chyba że dostosowywałeś formularze na karcie Formularze. Pliki projektów utworzone za pomocą starszych wersji Protégé-OWL (szczególnie przed wersją 3.0) mogą nie być zgodne z bieżącą wersją, w takim przypadku może być konieczne utworzenie nowego projektu od podstaw.

- Plik źródłowy

Plik źródłowy ma rozszerzenie .owl, .rdfs lub .rdf. W tym pliku są zdefiniowane klasy ontologii, osoby i właściwości. W zależności od sposobu zdefiniowania ontologii może istnieć kilka plików źródłowych. Jeśli jest modułowy i został poprawnie utworzony, Protégé-OWL znajdzie i załaduje wszystkie odpowiednie pliki źródłowe.

- Zajęcia

Klasy ontologii są bardzo podobne do klas w programie zorientowanym obiektowo. Podobnie jak programowanie obiektowe, klasy w ontologiach również tworzą hierarchię. Możesz wyświetlić tę hierarchię w panelu po prawej stronie, oznaczonym jako Hierarchia klas potwierdzonych. Klasą główną znajdującą się na szczycie hierarchii dziedziczenia jest Thing (dotyczy to wszystkich ontologii OWL).

Czym różnią się klasy ontologiczne od klas zorientowanych obiektowo?

Chociaż są one podobne pod wieloma względami, jedną ważną różnicą do zapamiętania jest to, że jednostka ontologii może należeć do zera lub większej liczby klas, oprócz dowolnej klasy odziedziczonej. Istnieją sposoby na ograniczenie klas, do których może należeć osoba należąca do jednej klasy, ale jeśli nie ma takich wyraźnych informacji, członkami mogą być jednostki, tak mało lub tak wiele klas, jak chce ich twórca ontologii.

- Równoważne klasy

W tej sekcji opisano inne klasy lub grupy, które są równoważne wybranej klasie.

- Superklasy

Nadklasy to klasa / klasy nadrzędne wybranej klasy (ponieważ możesz mieć wiele klas, możesz również mieć wiele klas nadrzędnych).

- Członkowie

Członkowie reprezentują osoby, które są członkami określonej klasy. Można je dodać jawnie lub wywnioskować później poprzez rozumowanie.

- Klasy rozłączne

Umożliwiają one jawne wybranie klas, do których nie mogą należeć członkowie wybranej klasy. W większości przypadków nie musisz jawnie oznaczać klas jako rozłącznych, ale może być pomocne, jeśli używasz jakiegoś zewnętrznego rozumowania lub aplikacji, aby definicje klas były wyraźne i dokładne. Przejrzyj dostępne klasy i spróbuj zrozumieć koncepcje, które próbują opisać. Po obejrzeniu zajęć przejdź do górnego menu i wybierz Reasoner -> Fakt ++. Spowoduje to, że osoba rozumująca będzie wnioskowała o dodatkowych faktach dotyczących klas.

- Właściwości obiektu i danych

Podczas dołączania właściwości do klas sensowne jest natychmiastowe dostarczenie oświadczeń dotyczących domeny i zakresu tych właściwości. Istnieje tutaj metodologiczne napięcie między ogólnością a szczegółowością. Warto zdefiniować dziedzinę i zakres tak wąsko, jak to możliwe, aby łatwo było wykryć potencjalne niespójności i nieporozumienia w ontologii poprzez wykrywanie naruszeń domeny i zakresu.

- Właściwości OWL reprezentują relacje między dwoma obiektami.

Istnieją dwie główne właściwości:

1. Właściwości obiektu łączą obiekt z obiektem.
2. Właściwości typu danych łączą obiekt z typem danych schematu XML lub rdf: literal

Inne ważne właściwości to:

Właściwości adnotacji

Ta właściwość może służyć do dodawania informacji adnotacji do klas, osób i właściwości.

Właściwości odwrotne:

1. Każda właściwość obiektu może mieć odpowiednią właściwość odwrotną.
2. Jeśli jakaś właściwość łączy osobę A z osobą B, to jej odwrotna właściwość łączy osobę B z indywidualną A.

Właściwości funkcjonalne:

1. Jeżeli nieruchomości jest funkcjonalna dla danej osoby, wówczas tylko jedna osoba może być powiązana poprzez jej własność.
2. Właściwości funkcjonalne nazywane są również właściwościami o pojedynczej wartości.

Odwrotne właściwości funkcjonalne:

Jeśli właściwość jest funkcją odwrotną, to jej właściwość odwrotna jest funkcjonalna.

Właściwości przechodnie:

Jeśli właściwość P jest przechodnia, a własność wiąże osobę A z indywidualną B, a także indywidualną B z indywidualną C, wówczas możemy wywnioskować, że osoba A jest powiązana z indywidualną C poprzez właściwość P.

Właściwości symetryczne:

Jeśli właściwość P jest symetryczna, a właściwość wiąże osobę A z osobą B, wówczas jednostka B jest również powiązana z osobą A poprzez właściwość P.

Kilka ważnych terminów

Liczność:

Liczność należy określić dla jak największej liczby właściwości, aby wskazać, czy dozwolone lub wymagane jest, aby miały określoną liczbę różnych wartości, czy nie. Często zdarzające się przypadki to „co najmniej jedna wartość” (tj. Wymagane właściwości) i „co najwyżej jedna wartość” (tj. Nieruchomości o jednej wartości).

Wymagane wartości:

Często klasy są definiowane na podstawie pewnej właściwości mającej określone wartości, a takie wymagane wartości można określić w OWL, używając owl: hasValue. Czasami wymagania są mniej rygorystyczne, na przykład właściwość-A może mieć pewne wartości z danej klasy (i niekoniecznie określoną wartość, owl: someValuesFrom).

Charakterystyka relacyjna:

Chodzi o relacyjne cechy własności, tj. Symetrię, przechodność, własności odwrotne, wartości funkcjonalne.

Dziedzina i zakresy własności

Właściwości łączą osoby z domeny z osobami z zakresu. OWL używa dziedziny i zakresu jako aksjomatów w rozumowaniu.

Ograniczenia własności:

W OWL właściwości są używane do tworzenia ograniczeń. Ograniczenia służą do ograniczania osób należących do klasy. Istnieje pięć ograniczeń:

- Ograniczenia kwantyfikatorów
- Egzystencjalny kwantyfikator
- Uniwersalny kwantyfikator
- Ograniczenia licznosci
- ograniczenia hasValue

Instancje

Używamy ontologii do organizowania zestawów instancji i istnieje oddzielny krok, aby wypełnić ontologie instancjami. Zazwyczaj liczba instancji jest większa niż liczba klas w ontologii. Ontologie mają różną wielkość, od kilkuset klas do dziesiątek tysięcy klas; liczba wystąpień waha się od setek do setek tysięcy, a nawet więcej. Z powodu tych dużych liczb zapełnianie ontologii instancjami zazwyczaj nie jest wykonywane ręcznie. Często wystąpienia są pobierane ze starszych źródeł danych, takich jak bazy danych. Inną, często używaną techniką jest automatyczne wyodrębnianie instancji z korpusu tekstu.

Tworzenie przykładowego projektu

Uwaga: Istnieją dwa główne sposoby modelowania ontologii:

- Oparte na ramkach
- SOWA

Każdy ma własny interfejs użytkownika.

1. Edytor Protege Frame: umożliwia użytkownikom budowanie i wypełnianie ontologii opartych na ramkach, zgodnie z OKBC (Open Knowledge Base Connectivity Protocol). Składa się ona z -

- Zajęcia
- Miejsca na nieruchomości i relacje
- Wystąpienia klasy

2. Redaktor Protege OWL:

Umożliwia użytkownikom budowanie ontologii dla sieci semantycznej. Składa się ona z-

- Zajęcia
- Właściwości
- Instancje
- rozumowanie

Krok 1:

1. Rozpocznij protegowanego
2. Przejdź do pliku
3. Wybierz Nowy
4. Wprowadź identyfikator URI ontologii (<http://www.bookboon.com/ontologies/bookstore.owl>)
5. Następnie wybierz RDF / XML
6. Wybierz Widok Właściwości

Powstał nowy, pusty projekt Protégé. Zapisz go w swoim lokalnym pliku jako bookstore.owl.

Krok 2:

Przejdź do zakładki Zajęcia. Przekonasz się, że puste drzewo klas zawiera jedną klasę zwaną owl: Thing, która jest nadklasą wszystkiego.

Utwórz podklasy:

Podklasy, które mają zostać skonstruowane to: Książki, BookType i BookAuthor. Są to podklasy sowy: Rzecz. Choć nie ma specjalnej konwencji nazewnictwa, ważne jest, aby zachować spójność. Zapoznaj się z częścią wyróżnioną na poniższym zrzucie ekranu, aby poznać ikonę używaną do tworzenia podklasy.

Krok 3:

Aby określić, że Book, BookType i BookAuthor są klasami rozłącznymi. Dodaj obie podklasy do panelu rozłącznych klas, wybierając je.

Krok 4:

Utwórz podklasy dla BookType i BookAuthor.

Na przykład,

BookType = (Romans, Thriller)

BookAuthor = (Szekspir, Sidney Sheldon)

Postępuj zgodnie z instrukcjami kreatora, aby utworzyć te rozłączne klasy.

Krok 5:

- Przejdź do zakładki Właściwości obiektu.
- Kliknij „Utwórz właściwości obiektu”.
- Utwórz właściwość obiektu „hasBelonger”.

Krok 6:

Teraz wybierz właściwość „hasBelonger” i dodaj do niej właściwości podrzędne. Utwórzmy właściwości podrzędne

„HasWriter” i „hasPublisher”.

Krok 7:

Utwórz właściwości odwrotne.

- Utwórz nową właściwość obiektu o nazwie isBelongerOf.
- Naciśnij przycisk „Ustaw właściwość odwrotną”.
- Wybierz „hasBelonger”.
- Odwrotna relacja została ustanowiona.
- Wybierz hasPublisher
- Utwórz isPublisherOf jako odwrotną właściwość hasPublisher.
- Stąd isPublisherOf jest własnością podrzędną isBelongerOf.
- Wybierz hasWriter
- Utwórz isWriterOf jako właściwość inverse.
- Stąd isWriterOf jest własnością podrzędną isBelongerOf.

Krok 8:

Utwórz właściwość przechodnią.

- Wybierz właściwość „hasBelonger”.
- Wybierz / zaznacz przechodnie pole wyboru.

- Wybierz właściwość „isBelongerOf”, upewnij się, że pole wyboru przechodnie jest zaznaczone.

Krok 9:

Twórz właściwości funkcjonalne.

- Wybierz właściwość „hasPublisher”
- Zaznacz pole wyboru „funkcjonalne”
- OWL-DL nie pozwala, aby właściwości typu danych były przechodnie, symetryczne lub miały właściwości odwrotne.

Krok 10:

Określ zakresy.

- Wybierz hasPublisher.
- Naciśnij przycisk zakresu.
- Wybierz BookType.
- Naciśnij przycisk OK.
- BookType musi być wyświetlany na liście zakresów.
- Gdy do zakresu dodaje się wiele klas, reprezentują one sumę wszystkich klas.
- Wybierz właściwość isPublisherOf.
- Ustaw domenę właściwości isPublisherOf na BookType.
- Ustaw zakres właściwości isPublisherOf na Book.
- Powyższe kroki zostaną teraz powtórzone dla hasWriter.
- Wybierz właściwość hasWriter.
- Określ zakres jako BookAuthor.
- Wybierz właściwość isWriterOf.
- Określ zakres jako Book.

Krok 11:

Określ domenę

- Wybierz właściwość hasPublisher.
- Naciśnij przycisk dodawania domeny.
- Wybierz Książka.
- Wciśnij OK.
- Książka jest wyświetlana na liście domen.
- Gdy wiele klas jest dodanych jako domeny, reprezentują one unię tych klas.

- Ustaw domenę właściwości isPublisherOf na BookType.
- Wybierz właściwość hasWriter.
- Określ domenę jako książkę.
- Wybierz właściwość isWriterOf.
- Określ domenę jako BookAuthor.

Krok 12:

Przywołaj rozum.

Ontologia opisana w OWL-DL może być przetwarzana przez rozumującego. Idź do owl -> preferencje, aby upewnić się, że wybrano OWL-DL. Główną usługą oferowaną przez myśliciela jest sprawdzenie, czy jedna klasa jest, czy nie podklasa innej klasy. Wykonując takie testy na wszystkich klasach, osoba rozumująca może obliczyć wywnioskowaną hierarchię klas ontologii. Inną usługą wnioskowania jest sprawdzanie spójności, tj. Sprawdzanie, czy klasa może mieć jakiekolwiek instancje. Klasa jest uważana za niespójną, jeśli nie może mieć żadnych instancji. Po zidentyfikowaniu niespójności są odpowiednio usuwane, aby zapewnić spójność.

Swoogle

Cel:

Ta sekcja ma na celu zbadanie Swoogle, który jest implementacją sieci semantycznej na poziomie badawczym.

Wprowadzenie

Sieć semantyczna to zasadniczo wszechświat sieci równoległy do sieci dokumentów online. Dokument sieci semantycznej (SWD) jest dobrze znany ze swoich semantycznych adnotacji i znaczących odniesień. Ponieważ żadna konwencjonalna wyszukiwarka nie może skorzystać z takich funkcji, wyszukiwarka dostosowana do SWD, specjalnie dla ontologii, jest potrzebna zarówno użytkownikom, jak i agentom oprogramowania i usługom. Na tym etapie oczekuje się, że użytkownicy będący ludźmi będą badaczami i programistami sieci semantycznej, którzy są zainteresowani uzyskiwaniem dostępu, eksploracją i przeszukiwaniem dokumentów RDF i OWL znalezionych w sieci.

Swoogle

Swoogle to wyszukiwarka ontologii, dokumentów, terminów i danych sieci semantycznej publikowanych w sieci. Swoogle korzysta z systemu robotów indeksujących do wykrywania dokumentów RDF i dokumentów HTML z osadzoną zawartością RDF. Swoogle uzasadnia te dokumenty i ich części składowe (np. Terminy i trójki) oraz rejestruje i indeksuje znaczące metadane na ich temat w swojej bazie danych. Swoogle to oparty na robotach system indeksowania i pobierania dla sieci semantycznej, tj. Dokumentów sieci Web w formacie RDF lub OWL. Wydobywa metadane dla każdego wykrytego dokumentu i oblicza relacje między dokumentami. Odkryte dokumenty są również indeksowane przez system wyszukiwania informacji, który może używać znaków N-Gram lub identyfikatorów URI jako słów kluczowych w celu znalezienia odpowiednich dokumentów i obliczenia podobieństwa w zestawie dokumentów. Jedną z interesujących właściwości, które obliczamy, jest ranga, miara ważności dokumentu sieci semantycznej.

Historia Swoogle

Swoogle to projekt badawczy prowadzony przez grupę badawczą Ebiquity z Wydziału Informatyki i Inżynierii Elektrycznej Uniwersytetu Maryland w hrabstwie Baltimore (UMBC). Częściowe wsparcie badawcze zapewnił kontrakt DARPA F30602-00-0591 oraz NSF, w postaci nagród NSF-ITRIS-0326460 i NSF-ITR-IDM-0219649. Współautorzy to Tim Finin, Li Ding, Rong Pan, Anupam Joshi, Pavan Reddivari, Joel Sachs, Pranam Kolari, Akshay Java, Lushan Han, Yun Peng, R. Scott Cost, Sandor Dornbush i Vishal Doshi. Swoogle indeksuje tylko dokumenty sieci semantycznej, obecnie w tym dokumenty napisane w RDF / XML, N-Triples, N3 (RDF) i niektóre dokumenty zawierające fragmenty RDF / XML. Dane prezentowane w Swoogle są zbierane z publicznie dostępnej sieci WWW. Swoogle ma politykę prywatności w języku angielskim i zasady dotyczące indeksowania.

Wdrożenie Swoogle

Swoogle stosuje podejście hybrydowe do zbierania sieci semantycznej, w tym ręczne przesyłanie, meta-indeksowanie oparte na Google, ograniczone indeksowanie HTML i indeksowanie RDF. Strona statystyk Swoogle jest obecnie w trakcie przebudowy. Swoogle ma zindeksowanych ponad 1,4 miliona dokumentów sieci semantycznej i 290 milionów trójek. Swoogle jest napisane głównie w Javie (JDK 1.4.2), a jego usługi internetowe są dostarczane przez serwer Apache Tomcat. Front-end witryny Swoogle jest napisany w PHP. Obecnie Swoogle przechowuje swoje dane w bazie danych MySQL (mysql 4.1.16). Cały system działa na platformie Linux (Fedor Core 4). Swoogle zostało zaprojektowane jako repozytorium adresów URL, ale nie jako potrójny magazyn, więc nie przechowuje wszystkich napotkanych trójek. Jednak aplikacje mogą być tworzone na podstawie Swoogle, które indeksują określoną klasę danych sieci semantycznej i zapewniają obsługę wnioskowania przy użyciu potrójnych magazynów. Swoogle próbuje ulepszyć następujące funkcje sieci semantycznej:

Znajdowanie odpowiednich ontologii:

Brak odpowiedniej ontologii zawsze prowadzi do powstania nowej ontologii, która jest często dostosowywana do ponownego wykorzystania. Swoogle pomaga użytkownikom znaleźć ontologie zawierające określone terminy, a użytkownicy mogą nawet zakwalifikować typ (klasę lub właściwość) terminu. Ponadto mechanizm rankingowy sortuje ontologie według ich popularności. Uważa się, że ta cecha nie tylko ułatwia oznaczanie danych, ale także przyczynia się do powstawania ontologii kanonicznych.

Znajdowanie danych instancji:

Aby pomóc użytkownikom w integracji danych Semantic Web rozproszonych w Internecie, Swoogle umożliwia odpytywanie SWD z ograniczeniami dotyczącymi używanych przez nie klas i właściwości.

Charakterystyka sieci semantycznej:

Zbierając metadane, zwłaszcza relacje między dokumentami, dotyczące sieci semantycznej, Swoogle ujawnia interesujące właściwości strukturalne, takie jak „w jaki sposób sieć semantyczna jest połączona?”, „W jaki sposób odwołuje się do ontologii?” Oraz „w jaki sposób ontologia jest modyfikowana zewnętrznie. ? ”.

Studium przypadku związane z siecią semantyczną

Cel:

Ti mamy studium przypadku, które można przekształcić w pełnoprawny projekt sieci semantycznej. Studenci chętni do podjęcia projektów badawczych w sieci semantycznej i studiowania tej książki w celu zbudowania projektu mogą pracować na studium przypadku.

Wprowadzenie do handlu elektronicznego

E-biznes

E-biznes to stosowanie technologii informacyjno-komunikacyjnych (ICT) do wspierania wszystkich działań biznesowych. E-biznes będzie bezpiecznym, elastycznym i zintegrowanym podejściem łączącym systemy i procesy, które obsługują podstawowe operacje biznesowe. Jest to proces wykorzystywania technologii sieci Web w celu usprawnienia procesów, poprawy produktywności, zwiększenia wydajności i umożliwienia organizacjom łatwej komunikacji z partnerami, dostawcami i klientami, łączenia systemów danych zaplecza i przeprowadzania transakcji handlowych w bezpieczny sposób.

Handel elektroniczny

Handel elektroniczny (WE) lub po prostu handel elektroniczny to nowa koncepcja opisująca kupowanie i sprzedawanie produktów, usług i informacji za pośrednictwem sieci komputerowych, w tym Internetu.

Wyzwania dla handlu elektronicznego

Podejście peer to peer odgrywa główną rolę w rozwijaniu pełnego potencjału handlu elektronicznego. Oto niektóre z wyzwań stojących przed rozwiązaniem handlu elektronicznego i sieci semantycznej:

- Potrzebne jest zmechanizowane wsparcie w wyszukiwaniu i porównywaniu dostawców i ich ofert. Obecnie prawie cała ta praca jest wykonywana ręcznie, co poważnie utrudnia skalowalność handlu elektronicznego. Technologia sieci semantycznej może uczynić ją przetwarzalną maszynowo.
- Potrzebne jest zmechanizowane wsparcie w radzeniu sobie z licznymi i niejednorodnymi formatami danych. Istnieją różne „normy” opisujące produkty i usługi, katalogi produktów i dokumenty biznesowe. Aby lepiej zdefiniować takie standardy, potrzebna jest technologia ontologiczna. Skuteczne pomosty między różnymi terminologiami są niezbędne dla otwartości i skalowalności.
- Potrzebne jest zmechanizowane wsparcie w radzeniu sobie z licznymi i niejednorodnymi logikami biznesowymi.

Obecny scenariusz

Model biznesowy e-commerce

Model to systematyczne lub ustrukturyzowane podejście, które instruuje, jak należy postępować w celu wykonywania działalności biznesowej. Model biznesowy określa, w jaki sposób firma planuje zarabiać pieniądze i jak jest konkurencyjna w branży.

- Model biznesowy to metoda prowadzenia działalności, dzięki której firma może generować dochody, aby się utrzymać. Organizacje muszą zdefiniować i wdrożyć strategię, aby odnieść sukces w handlu elektronicznym.
- Model biznesowy, który ma na celu wykorzystanie i wykorzystanie unikalnych cech Internetu i sieci WWW, nazywa się E-Business Model.
- Elementy zawarte w modelu biznesowym dotyczą wszystkich funkcji przedsiębiorstwa, w tym czynników, takich jak wydatki, przychody, strategie operacyjne, struktura korporacyjna i procedury sprzedaży i marketingu. (Polityka, operacje, technologia i ideologia firmy określają jej model biznesowy)

Model biznesowy e-commerce:

Model biznesowy, który ma na celu wykorzystanie i wykorzystanie wyjątkowych cech Internetu i sieci WWW, nazywany jest modelem biznesowym handlu elektronicznego.

Kluczowe składniki modelu biznesowego

- Propozycja wartości.
- Model przychodów.
- Okazja na rynku.
- Konkurencyjne środowisko.
- Przewaga konkurencyjna.
- Strategia rynkowa.
- Rozwój organizacyjny.
- Zespół zarządzający.

1. Propozycja wartości

Propozycja wartości określa, w jaki sposób produkt lub usługa firmy spełnia potrzeby klientów. Ma na celu udzielenie odpowiedzi na następujące pytania -

- Dlaczego klienci wybierają współpracę z Twoją firmą zamiast z inną firmą?
- Co zapewni Twoja firma, czego inne firmy nie mogą i nie mogą?

2. Model przychodów

Model przychodów opisuje, w jaki sposób firma będzie zarabiać, generować zyski i zapewniać wyższy zwrot z zainwestowanego kapitału. Oto dobrze znane modele przychodów e-commerce:

- Model reklamowy
- Model subskrypcji
- Model opłat transakcyjnych
- Model sprzedaży
- Model afiliacyjny
- Model przychodów z reklam

W modelu reklamowym firma zapewnia forum dla reklam i otrzymuje opłaty od reklamodawców.

Przykład: yahoo

- Model przychodów z subskrypcji

W modelu subskrypcyjnym firma oferuje swoim użytkownikom, treści lub usługi i pobiera opłatę abonamentową za dostęp do niektórych lub wszystkich swoich ofert. Przykład: Wall Street Journal

- Model przychodów z opłat transakcyjnych

W modelu opłat transakcyjnych firma otrzymuje opłatę za umożliwienie lub wykonanie transakcji.

Przykład: eBay

- Model przychodów ze sprzedaży

W modelu przychodów ze sprzedaży firma uzyskuje przychody ze sprzedaży towarów, informacji lub usług.

Przykład: Amazon

- Model przychodów partnera

W modelu przychodów stowarzyszonych firma kieruje działalność do podmiotu stowarzyszonego i otrzymuje opłatę za polecenie lub procent przychodów z jakiegokolwiek wynikającej ze sprzedaży sprzedaży.

Przykład: MyPoints

3. Okazja rynkowa

- Szansa rynkowa odnosi się do planowanej przestrzeni rynkowej firmy i ogólnych potencjalnych możliwości finansowych dostępnych dla firmy w tej przestrzeni rynkowej.
- Jest określony przez potencjał przychodów w każdej z nisz rynkowych, w których masz nadzieję konkurować.

Przestrzeń rynkowa

Przestrzeń rynkowa to obszar faktycznej lub potencjalnej wartości handlowej, w którym firma zamierza działać

4. Otoczenie konkurencyjne

Otoczenie konkurencyjne odnosi się do innych firm działających na tym samym rynku, które sprzedają podobne produkty. Wpływ na to mają następujące czynniki:

- Liczba aktywnych zawodników
- Jak duże są ich operacje?
- Udział w rynku każdego konkurenta
- Jak zyskowe są te firmy?
- Jak wyceniają swoje produkty?

5. Przewaga konkurencyjna

- Przewaga konkurencyjna jest osiągnięta przez firmę, kiedy może wytworzyć lepszy produkt i / lub wprowadzić produkt na rynek po niższej cenie niż większość lub wszyscy jej konkurenci
- Jest to osiągnięte, ponieważ firma była w stanie uzyskać zróżnicowany dostęp do czynników produkcji, którym odmawiają jej konkurenci.

6. Strategia rynkowa

- Opracowany przez Ciebie plan, który dokładnie określa, w jaki sposób zamierzasz wejść na nowy rynek i przyciągnąć nowych klientów, to strategia rynkowa.

- Najlepsze koncepcje biznesowe nie powiodą się, jeśli nie zostaną odpowiednio sprzedane potencjalnym klientom

7. Rozwój organizacyjny

- Opisuje, w jaki sposób firma zorganizuje pracę, którą należy wykonać.
- Praca jest zazwyczaj podzielona na działy funkcjonalne.
- W miarę rozwoju firmy przechodzi od lekarzy ogólnych do specjalistów.

8. Zespół zarządzający

- Pracownicy firmy, odpowiedzialni za model biznesowy, tworzą zespół zarządzający.
- Silny zespół zarządzający zapewnia natychmiastową wiarygodność inwestorom zewnętrznym. Silny zespół zarządzający może nie być w stanie uratować słabego modelu biznesowego
- Zespół zarządzający powinien być w stanie zmienić model i przedefiniować biznes w razie potrzeby.

8.3.1 Główny model e-biznesu

- Business-to-Consumer (B2C)
- Business-to-Business (B2B)

Model biznesowy między biznesem a konsumentem

- B2C (lub ekstranety) to po prostu relacje internetowe między istniejącymi partnerami. Prowadzone są przez jedną firmę, która stara się obniżyć koszty prowadzenia interesów z obecnymi dostawcami lub klientami indywidualnymi.

Przykłady:

Amazon.com, Egghead.com

Business to Business

- „B2B” to handel między przedsiębiorstwami prowadzony przez Internet (zwany przestrzenią handlu elektronicznego B2B lub rynkami elektronicznymi)

Aplikacje B2B:

- Reklama
- Licytacja
- Zaopatrzenie
- Zarządzanie kanałami
- Handel elektroniczny

Unikalne cechy technologii handlu elektronicznego

1. Wszechobecność

Handel elektroniczny zmienia strukturę branży, tworząc nowe kanały marketingowe i zwiększając rozmiar całego rynku. Tworzy nową efektywność w operacjach branżowych i obniża koszty operacji sprzedażowych firm. Umożliwia także nowe strategie różnicowania.

2. Globalny zasięg

Zmienia strukturę branży, obniżając bariery wejścia, ale jednocześnie znacznie rozszerza rynek. Obniża koszty działalności przemysłu i przedsiębiorstw poprzez wydajność produkcji i sprzedaży. Umożliwia także konkurencję w skali globalnej.

3. Standardy uniwersalne

Zmienia strukturę branży, obniżając bariery wejścia i nasilając konkurencję w branży. Obniża koszty działalności przemysłu i firm poprzez obniżenie kosztów obliczeniowych i komunikacyjnych oraz umożliwia strategię o szerokim zakresie.

4. Bogactwo

Zmienia strukturę branży, zmniejszając siłę potężnych kanałów dystrybucji. Zmienia branżę i koszty operacyjne firmy, zmniejszając zależność od siły roboczej i usprawniając sprzedaż po sprzedaży strategię wsparcia.

5. Interaktywność

Zmienia strukturę branży, zmniejszając zagrożenie ze strony substytutów poprzez ulepszoną personalizację. Zmniejsza koszty branżowe i firmowe, zmniejszając zależność od siły roboczej i umożliwiając strategię różnicowania.

6. Personalizacja / dostosowywanie

Zmienia strukturę branży, zmniejszając zagrożenia związane ze substytutami, podnosząc bariery wejścia i redukując koszty łańcucha wartości w przemyśle i firmie, zmniejszając zależność od sił sprzedaży.

7. Gęstość informacji

Zmienia strukturę branży, osłabiając potężne kanały sprzedaży, przenosząc siłę przetargową na konsumenta oraz redukując koszty przemysłu i działalności firm poprzez obniżenie kosztów pozyskiwania, przetwarzania i rozpowszechniania informacji o dostawcach i konsumentach.

Studium przypadku 1 - Wdrażanie wirtualnego biura podróży w sieci semantycznej

Cel:

Wdrożenie wirtualnego biura podróży w sieci semantycznej. Wirtualne biuro podróży (VTA) działa jako pośrednik między klientami a dostawcami usług turystycznych. Dostawcami usług turystycznych będą tutaj firmy komercyjne, które świadczą usługi lotnicze, hotelowe, wynajmu samochodów itp. Rezultat będzie dostarczał klientom pakiet turystyczny poprzez agregację usług różnych dostawców usług turystycznych.

Opis scenariusza projektu:

Wyobraź sobie VTA, która jest usługą dla użytkownika końcowego, zapewniającą klientom usługi e-turystyki. Usługi te mogą obejmować wszystkie rodzaje usług informacyjnych związanych z turystyką. Na przykład informacje o wydarzeniach i zabytkach w okolicy, usługach obsługujących rezerwację lotów, hoteli, wypożyczalni samochodów itp. Takie VTA istnieją już teraz na rynku. Obecnie portale te są dostępne za pośrednictwem witryn html. Partnerzy VTA są zintegrowani poprzez tradycyjną integrację B2B.

Dzięki zastosowaniu Semantic Web Services, VTA będzie w stanie łatwo konfigurować i wywoływać usługi sieciowe dostarczane przez kilku dostawców e-turystyki i łączyć je w nowe usługi dla klientów. Takie VTA dostarcza zautomatyzowanych usług e-turystyki użytkownikom końcowym za pośrednictwem różnych interfejsów i można je łatwiej rekonfigurować zgodnie z rzeczywistymi potrzebami. Nasze VTA agreguje usługi sieciowe różnych dostawców usług turystycznych. Krótko mówiąc, zapewnia następujące funkcje:

Klient korzysta z usługi VTA jako punktu wejścia dla swoich żądań. Usługi dla użytkowników końcowych są agregowane przez VTA poprzez wywoływanie i łączenie usług sieciowych oferowanych przez kilku dostawców usług turystycznych. Aby to ułatwić, może istnieć tak zwana umowa „parasolowa” między dostawcami usług a VTA, regulująca korzystanie z usług internetowych i ich uprawnienia.

Wada obecnego systemu

- Obecne VTA utrzymują dane we własnej bazie danych. Oferty, które otrzymujemy, pochodzą tylko z ustalonego zestawu usług, które nabywają dzięki współpracy z różnymi usługodawcami.
- Usługodawcy nie kontaktują się bezpośrednio z klientami z własnym zestawem ofert. Wynika to z różnic w formacie bazy danych używanej przez tych usługodawców. Na przykład nazwa pola dla „numeru identyfikacyjnego produktu” może mieć postać „prod_id” w bazie danych usługodawcy-1 i „product_id” w bazie danych usługodawców-2.
- W związku z tym trudno jest zintegrować te usługi razem i zapewnić klientowi skonsolidowany pakiet.

Co robi system docelowy?

- System docelowy pobierze usługi z bazy danych samego usługodawcy zamiast pobierać je z bazy danych utrzymywanej przez VTA.
- Odbywa się to za pomocą mapowania ontologicznego, które odwzorowuje nazwy pól z ich odpowiednim znaczeniem i różnymi możliwymi nazwami pól.

Usługi VTA:

Usługa sieciowa VTA oferuje użytkownikom końcowym usługi wyszukiwania i rezerwacji biletów hotelowych i lotniczych. Ta usługa internetowa składa się z innych usług internetowych, a mianowicie usług zakwaterowania i transportu, które są świadczone i publikowane przez różne firmy i są zarejestrowane w VTA.

Technologie do wdrożenia VTA:

- JADE
- HTML 5, CSS, skrypt Java
- Serwlety

JADE:

JADE (Java Agent Development Environment) to oprogramowanie pośredniczące ułatwiające tworzenie systemów wieloagentowych. Obejmuje

- Środowisko wykonawcze, w którym agenty JADE mogą „żyć” i które musi być aktywne na danym hoście, zanim jeden lub więcej agentów będzie mogło zostać uruchomionych na tym hoście.

- Biblioteka klas, z których programiści muszą / mogą korzystać (bezpośrednio lub specjalizując się w nich) do rozwijania swoich agentów.
- Zestaw narzędzi graficznych, który umożliwi administrowanie i monitorowanie działania uruchomionych agentów.

Każda działająca instancja środowiska wykonawczego JADE nazywana jest kontenerem, ponieważ może zawierać kilka agentów. Zestaw aktywnych kontenerów nazywa się Platformą. Gdy agent A komunikuje się z innym agentem B, pewna ilość informacji I jest przesyłana z A do B za pomocą komunikatu ACL. W wiadomości ACL I jest przedstawiane jako wyrażenie treści zgodne z odpowiednim językiem treści i zakodowane w odpowiednim formacie. Zarówno A, jak i B mają swój własny sposób wewnętrznej reprezentacji I. Stąd jasne jest, że za każdym razem, gdy agent A wysyła informację I do agenta B, A musi przekształcić swoją wewnętrzną reprezentację I na odpowiadającą jej reprezentację wyrażenia treści ACL, a B potrzebuje wykonać odwrotną konwersję. Ponadto B powinien również przeprowadzić szereg sprawdzeń semantycznych, aby zweryfikować, czy ja jest znaczącą informacją. Wsparcie dla języków i ontologii treści zapewniane przez JADE jest zaprojektowane tak, aby automatycznie wykonywać wszystkie powyższe operacje konwersji i sprawdzania

Proces:

W przeglądarce użytkownik tworzy zdarzenie i generuje komunikat POST. Obsługuje go serwlet i wywoływana jest akcja „sendmessage”. Akcja tworzy nowy obiekt BlackBoard, który będzie kanałem komunikatów między GatewayAgent a serwletem. GatewayAgent pobiera utworzony wcześniej obiekt pulpitu nawigacyjnego i wyodrębnia odbiorcę oraz wiadomość. Następnie wysyła wiadomość. ServerAgent, który jest teraz odbiorcą, odpowiada GatewayAgent. ServerAgents odwołując się do opisu ontologicznego wywołują odpowiednią usługę sieciową klienta, która z kolei wywołuje usługę sieciową w domenie dostawcy usług, która pobiera wynik ze swojej bazy danych i przekazuje go dalej. Kiedy osiągnie GatewayAgent, GatewayAgent pakuje odpowiedź i wysyła ją przez BlackBoard do serwletu. Aplet przekazuje go do przeglądarki.

Wniosek:

Możemy więc stwierdzić, że ta rozwijająca się technologia ma szeroki zakres. Jedynym problemem, z jakim się boryka, jest to, że wszystkie narzędzia i interfejsy opracowane w celu implementacji usług w sieci semantycznej, takie jak WSMT (Web Service Modeling Toolkit), są na podstawowym etapie rozwoju, w przeciwieństwie do narzędzi w Web 2.0, takich jak Netbeans, które już są dobrze - opracowany i gotowy do użycia. Stąd wniosek jest taki, że jeśli będziemy w stanie bardziej skoncentrować się na rozwijaniu tych narzędzi w pełnoprawny interfejs, ta technologia sieci semantycznej może przynieść wspaniałe wynalazki.