Architektura portalu dziedzinowego

Boiński T.

12 lutego 2008

Streszczenie. Zaprezentowano koncepcję architektury i funkcjonalności dziedzinowego portalu tworzenia ontologii realizowanego przez katedrę KASK. Portal podzielony został na niezależne komponenty, które z powodzeniem można realizować odrębnie a następnie połączyć za pomocą ich interfejsów. W dalszej części pracy omówione zostaną poszczególne elementy systemu i ich zadania. Następnie przedstawiona będzie sugerowana metodologia wytwarzania systemu. W ostatnim rozdziale pokrótce zestawiono również technologie wykorzystywane w trakcie tworzenia portalu.

Słowa kluczowe. architektura, portal

1 Wymagania dotyczące portalu

Podstawowe zadania projektowanego narzędzia wynikają bezpośrednio z wymagań zapewnienia kolaboracyjnego tworzenia ontologii dziedzinowych w zakresie medycyny, biznesu oraz bezpieczeństwa. Z tego powodu na podstawową funkcjonalność systemu składać będą się możliwości edycji ontologii, czyli tworzenia, modyfikacji oraz usuwania klas, atrybutów oraz relacji. Dodatkowo system musi umożliwiać trwałe przechowywanie za jego pomocą tworzonych i modyfikowanych ontologii, a także umożliwić podgląd i modyfikację na dowolnym etapie ich konstrukcji. Stąd portal musi dostarczać sprawnych mechanizmów wersjonowania, podobnych do tych znanych z subversion jednak operujących na poziomie semantycznym.

W celu zapewnienia prostej komunikacji z innymi systemami do tworzenia ontologii, dostępne powinny być również operacje importu do systemu przygotowanych poza systemem oraz eksportu przechowywanych ontologii. Operacje te muszą umożliwiać konwersje sposobu reprezentacji ontologii pomiędzy stosowanym w portalu a najbardziej popularnymi formatami zewnętrznymi.

Ważnym aspektem systemu jest możliwość pracy kolaboracyjnej. Przejawiać się ona będzie poprzez mechanizmy pracy grupowej, czyli możliwość pracy wielu osób równolegle nad jedną ontologią. Wszelkie zmiany w ontologiach rejestrowane będą jako sugestie zmian, które z kolei muszą zostać zaakceptowane przez użytkowników odpowiedzialnych za stan danej ontologii. Dopiero po takiej akceptacji sugerowane zmiany rzeczywiście wejdą w życie i staną się nową wersją ontologii. Konieczne jest tu zapewnienie odpowiednich mechanizmów wykrywania i rozwiązywania konfliktów, jakie mogą pojawić się w czasie od zgłoszenia sugestii do jej zaakceptowania. Proces korekty konfliktów wykonywany będzie ręcznie przez autora sugestii. System powinien jednak w miarę możliwości wspierać dokonywanie poprawek.

Drugim elementem wspomagającym pracę grupową nad ontologią jest forum dołączone do każdej z opracowywanych ontologii. Za jego pomocą użytkownicy systemu będą mogli prowadzić dyskusje nad kształtem poszczególnych elementów ontologii czy kierunku, w jakim zmierzać powinno docelowe rozwiązanie.

W dużej mierze sam proces tworzenia oraz edycji ontologii będzie wykonywany ręcznie prze specjalistów z danej dziedziny, gdyż to właśnie oni budować będą w systemie model posiadanej przez siebie wiedzy. System będzie jednak starał się w sposób półautomatyczny sugerować korektę części błędów poprzez wyszukiwanie potencjalnej nadmiarowości w modelu. System będzie również wspierał użytkownika w procesie odwzorowywania ontologii (czyli tworzenia powiązań między nimi) oraz łączenia ontologii (czyli tworzenia nowej dużej ontologii będącej wynikiem połączenia dwu lub więcej mniejszych).

2 Architektura portalu dziedzinowego

Rysunek 1: Architektura portalu dziedzinowego

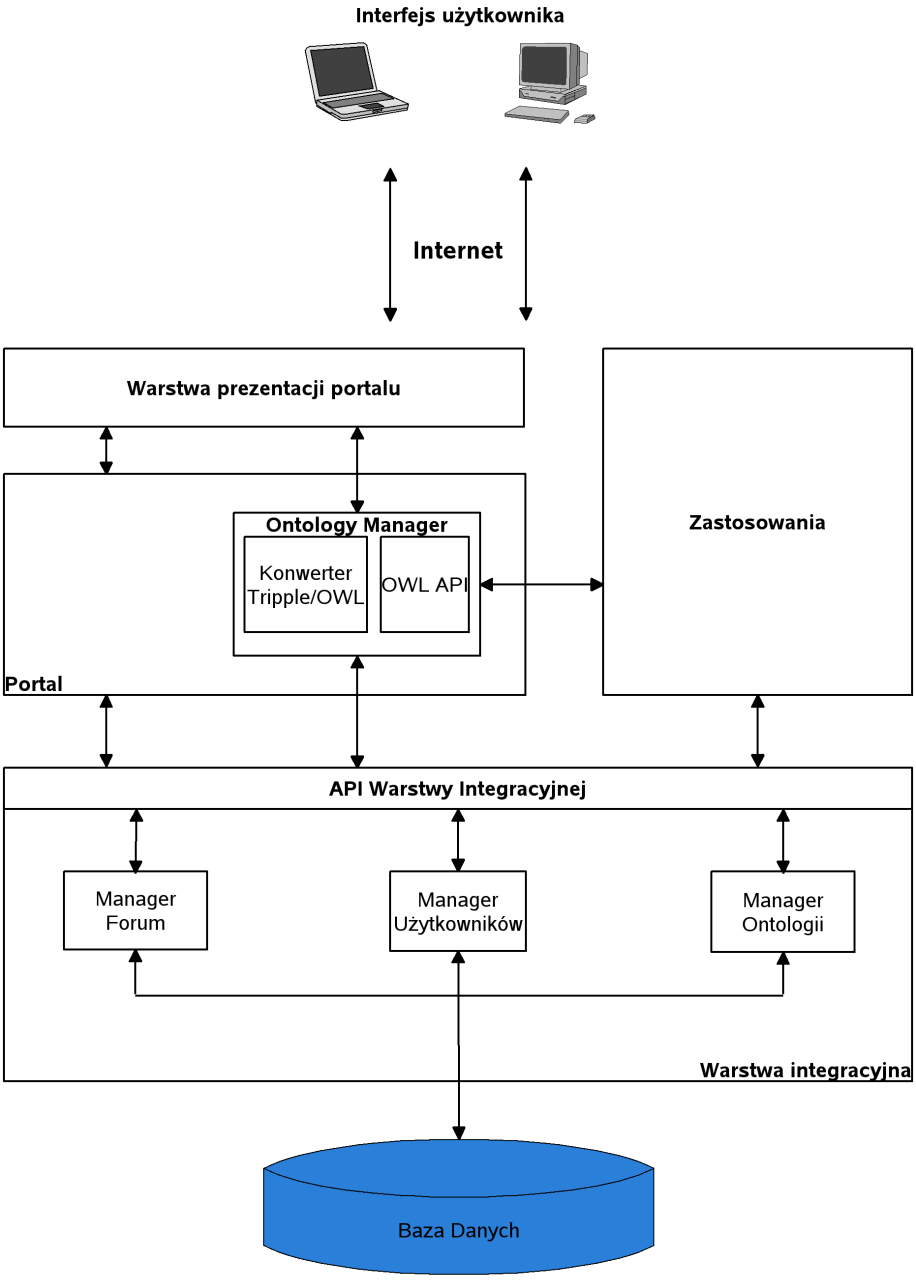
Wysokopoziomowa architektura portalu dziedzinowego zaprezentowana jest na Rys. 1. Można ją podzielić na 3 główne elementy:

- Warstwa integracji – zarządzającą dostępem do danych i ich zapisem/odczytem, realizującą politykę bezpieczeństwa systemu poprzez zarządzanie użytkownikami i ich uprawnieniami. Związane z jej funkcjami operacje dostarczane są pozostałym częściom systemu poprzez API tej warstwy,
- Portal – odpowiedzialny za dostarczenie elementów niezbędnych do tworzenia i edycji ontologii w poszczególnych dziedzinach, zapewnienie wszelkich niezbędnych operacji na modelu ontologii jak dodawania klas, relacji, atrybutów itp. oraz przyjaznego interfejsu użytkownika,
- Moduły zastosowań – zewnętrzne moduł odpowiedzialne za praktyczne zastosowanie ontologii zbudowanych za pomocą portalu.

Każdy z głównych elementów systemu podlega dalszej specjalizacji, co zostało zaprezentowane na Rys. 2. Bardziej szczegółowo zadania poszczególnych warstw omówione zostaną w dalszej części pracy.

3 Warstwa integracji

Warstwa integracji stanowi repozytorium ontologii dla pozostałych części systemu. Za pomocą tej części możliwe jest utrwalenie oraz późniejsze odczytanie



Rysunek 2: Architektura portalu dziedzinowego

gotowej ontologi a także zarządzanie użytkownikami. W tej części systemu zdefiniowany został sposób reprezentacji ontologii wspólny dla całego portalu. Jako obowiązujący przyjęto zapis w postaci trójek wywodzący się z RDF[6]. Pozostałe warstwy będą operować na ontologiach za pośrednictwem tej reprezentacji. W skład warstwy integracji wchodzą trzy moduły odpowiedzialne odpowiednio

za zarządzanie użytkownikami, ontologiami oraz forum. Moduły te zaprezentowano w dalszej części artykułu.

3.1 Moduł zarządzania użytkownikami - UserManager

Moduł ten odpowiedzialny jest za wszelkie operacje wykonywane na użytkownikach. Za jego pomocą można w systemie utworzyć bądź zmodyfikować użytkownika. Ta klasa pozwala też na przeglądanie wszystkich lub wybranego użytkownika.

W systemie każdy użytkownik jest jednoznacznie identyfikowany poprzez jego nazwę (username), będącą zwykłym ciągiem znaków z zakresu a-z oraz 0-9. Właściwe dane użytkownika przekazywane są poprzez obiekty typu User, zawierające dane dodatkowe takie, jak pełna nazwa użytkownika, hasło, e-mail, datę ostatniego logowania czy liczbę ontologii jaką ten użytkownik może utworzyć.

W systemie obowiązywać będą trzy poziomy uprawnień użytkowników:

- gość - użytkownik niezalogowany, posiada dostęp jedynie do publicznej części portalu
- administrator - użytkownik zarządzający onologiami oraz pozostałymi użytkownikami w systemie
- użytkownik zarejestrowany - zarejestrowany i zalogowany użytkownik w systemie, ma prawo zgłaszać sugestie do dowolnej ontologii, ma prawo czytać oraz pisać na forum. Dodatkowo może przyjąć jedną z dwu funkcji:
  - właściciel ontologii - jest twórcą i zarządcą ontologii, podejmuje wszelkie decyzje dotyczące jej kształtu, akceptuje bądź odrzuca sugestie innych użytkowników, tworzy nowe wersje ontologii
  - ekspert ontologii - wyznaczany przez właściciela ontologii, ma prawo do akceptacji sugestii zmian oraz tworzenia nowych wersji ontologii

Moduł ten umożliwia wykonanie następujących operacji:

- dodaj użytkownika
- usuń użytkownika
- zmodyfikuj użytkownika
- wyszukaj użytkownika
- wyświetl listę użytkowników
- uwierzytelnij użytkownika

3.2 Moduł zarządzania ontologiami - OntologyManager

Moduł ten odpowiedzialny jest za operacje na ontologiach. Należy tutaj zaznaczyć, że w zakres funkcjonalności warstwy logiki nie wchodzi modyfikowanie, przetwarzanie czy interpretowanie wiedzy zawartej w ontologiach. Zadaniem tej warstwy jest umożliwienie składowania, wersjonowania oraz zgłaszania poprawek do ontologii, nie zaś bezpośrednie na nich operowanie.

Za pomocą obiektu typu OntologyManager zalogowany użytkownik może stworzyć nową ontologię (zarówno pustą jak i na podstawie innej ontologii), pobrać już istniejącą, zablokować oraz odblokować ontologię a także zgłosić poprawki do ontologii oraz je zatwierdzić bądź odrzucić.

Obiekt typu OntologyManager pośredniczy we wszelkich operacjach dotyczących ontologii. Aby możliwe było wskazanie jakiej ontologii dotyczy dana operacja, ontologie w systemie identyfikowane są za pomocą URI, czyli unikatowego ciągu znakowego. Ciąg ten zawiera zarówno informacje o samej ontologii jak i jej wersji. W systemie identyfikator ten przechowywany jest w obiektach typu OntologyURI. Przechowywany jest on tam w pełnej postaci, czyli razem z informacją o wersji i podwersji. Obiekt ten umożliwia wyłuskanie URI zarówno w pełniej postaci jak i pozbawionej informacji o wersji (takie URI jest wspólne dla wszystkich wersji danej ontologii) a także informacji o wersji jak i podwersji. Przy określaniu ontologii wszędzie tam, gdzie mowa jest o konkretnej wersji jako jednoznaczny identyfikator stosowane jest OntologyURI. W przypadku, gdy chcemy wskazać daną ontologię we wszystkich wersjach posługiwać się należy parametrem baseOntologyURI wyłuskanym z OntologyURI za pomocą dostępnych w tej klasie metod.

Przy operacji zgłaszania propozycji zmian do ontologii moduł wspierany jest obiektami typu Proposition. Obiekty też zawierają informację o autorze zmiany a także właściwą listę zmian w postaci listy obiektów typu Change reprezentującą zmiany na poziomie trójek.

Moduł ten umożliwia wykonanie następujących operacji:

- utwórz nową ontologię
- wyszukaj ontologię
- pobierz listę ontologii
- pobierz listę dostępnych wersji danej ontologii
- odczytaj ontologię w wybranej wersji
- dodaj propozycję zmiany do ontologii
- usuń propozycję zmiany do ontologii
- zatwierdź propozycję zmiany
- odrzuć propozycję zmiany
- dodaj nową wersję ontologii
- eksportuj ontologię do zadanego formatu, np. RDF
- importuj ontologię zapisaną w zadanym formacie, np. RDF
- ustaw właściciela ontologii
- pobierz właściciela ontologii
- dodaj eksperta do ontologii
- usuń eksperta z ontologii
- pobierz ekspertów ontologii

3.3 Moduł zarządzania forum - ForumManager

Moduł umożliwiający dostęp do forum, będącego głównym pośrednikiem w dyskusji nad kształtem i kierunkiem zmian ontologii. Do każdej ontologii przypisane jest jedno forum wspólne dla wszystkich jej wersji. Moduł ten pozwala na pobranie listy tematów (Topic) przypisanych do danej ontologii (wskazanej poprzez BaseURI wyłuskane z OntologyURI) a także wiadomości (Message) przypisanych do danego tematu. Za pośrednictwem ForumManager użytkownicy mogą dodawać nowe tematy oraz wiadomości a także usuwać i modyfikować już istniejące wpisy.

Moduł ten umożliwia wykonanie następujących operacji:

- dodaj temat
- usuń temat
- zmodyfikuj temat
- wyszukaj temat
- dodaj wiadomość
- usuń wiadomość
- zmodyfikuj wiadomość
- wyszukaj wiadomość

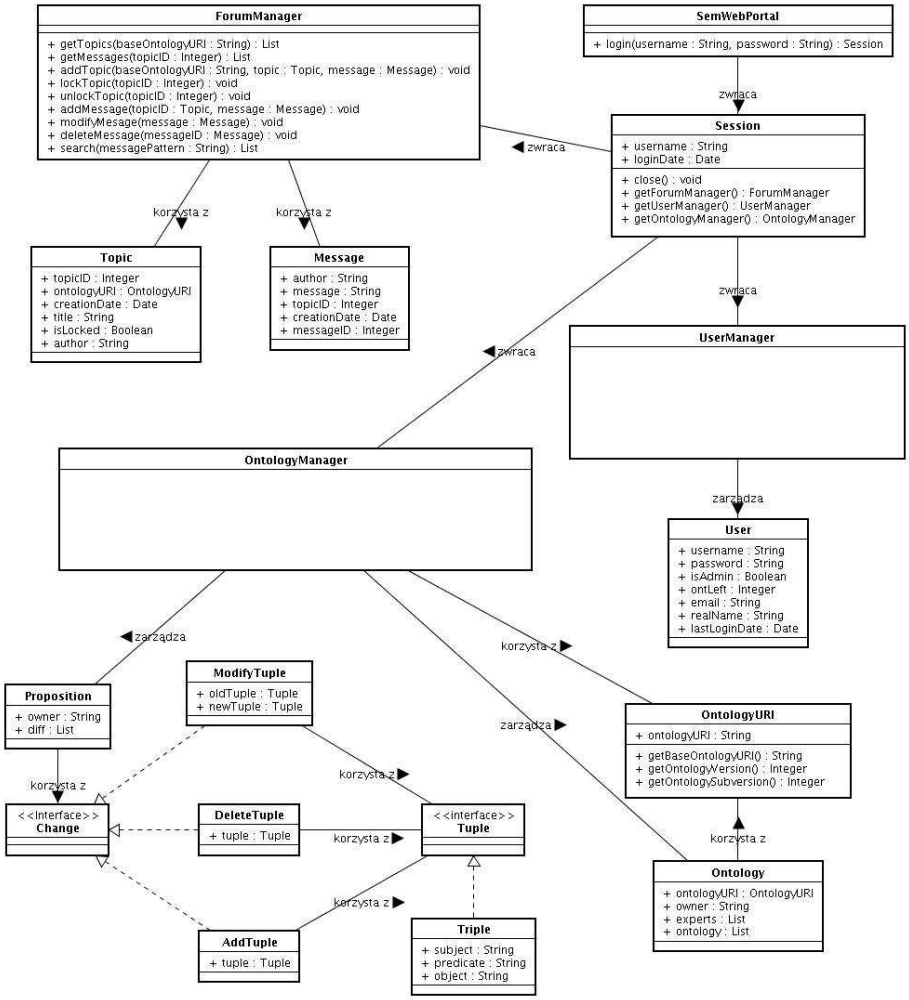
3.4 API Warstwy integracji

Funkcjonalność warstwy integracji dostępna będzie za pośrednictwem jej API przedstawionego na Rys. 3. W celu zachowania przejrzystości ukryte zostały metody klas OntologyManager oraz UserManager.

Rozpoczynając pracę z systemem użytkownik uwierzytelnia się informując system kim jest. W tym celu na głównej klasie systemu, czyli SemWebPortal wykonywana jest metoda login. Metoda ta zwraca obiekt typu Session będący kontekstem bieżącego użytkownika informującym kim on jest oraz kiedy się zalogował. Obiekt ten umożliwia również uzyskanie dostępu do 3 innych kluczowych obiektów opisanych w poprzednich podrozdziałach:

1. UserManager
2. OntologyManager
3. ForumManager

Wszystkie publiczne operacje dostępne będą zarówno jako wywołania JavaBeans jak i poprzez Web Services. Umożliwi to zarówno szybką komunikację z portalem jaki dostępność do danych zewnętrznych aplikacji w ustandaryzowany sposób.



Rysunek 3: API Warstwy integracji

4 Portal

Zadaniem tej części systemu jest bezpośrednia realizacja celów narzędzia, czyli umożliwienie tworzenia ontologii. Na tym poziomie ontologi reprezentowane będą za pomocą języka OWL. Główne funkcje jaka ta część będzie realizować to pośredniczenie pomiędzy warstwąintegracji a warstwą prezentacji. Moduł ten będzie tłumaczył akcje użytkownika na wywołaniafunkcji systemu. Ważnym zadaniem portalu jest dostarczenie przejrzystego i wygodnego interfejsu użytkownika umożliwiającego tworzenie ontologii ekspertom nie obeznanym z informatycznym zagadnieniem jakim są ontologie. Interfejs w dużej mierze oparty zostanie o technologię apletów Java[4], HTML oraz JavaScript.

4.1 Biblioteka operująca na ontologiach zapisanych w języku OWL

Głównym elementem tej warstwy jest moduł obsługi ontologii oparty o OWL API[7]. OWL API jest biblioteką pozwalającą na obsługę ontologii w języku OWL. Implementuje ona wszystkie występujące w specyfikacji operacje, a jej użycie pozwoli na znaczne przyspieszenie budowy portalu. W połączeniu z modułem konwersji reprezentacji ontologii z trójek na język OWL, biblioteka OWL API pozwoli na sprawną obróbkę ontologii bez konieczności implementacji rozbudowanej specyfikacji języka OWL.

Fragment portalu odpowiedzialny za obróbkę ontologii dostępny będzie w postaci biblioteki zarówno dla portalu jak i Modułu zastosowań poprzez API przedstawione na Rys. 4.

Głównym zadaniem tej biblioteki jest rozszerzenie poprzez interfejs PGOWLOntologyManager funkcjonalności zawartej w OWL API, by możliwa była realizacja założeń poczynionych w Warstwie integracji a nie było konieczne dublowanie możliwości zawartych w pakiecie. Użytkownik za pośrednictwem wspomnianego interfejsu będzie mógł operować na ontologiach nie martwiąc się o konwersję jej reprezentacji z języka OWL do postaci trójek. Drugim ważnym zadaniem biblioteki jest możliwość reprezentacji propozycji zmian zgłaszanych przez użytkowników w sposób zrozumiały zarówno przez Warstwę integracji jak i OWL API.

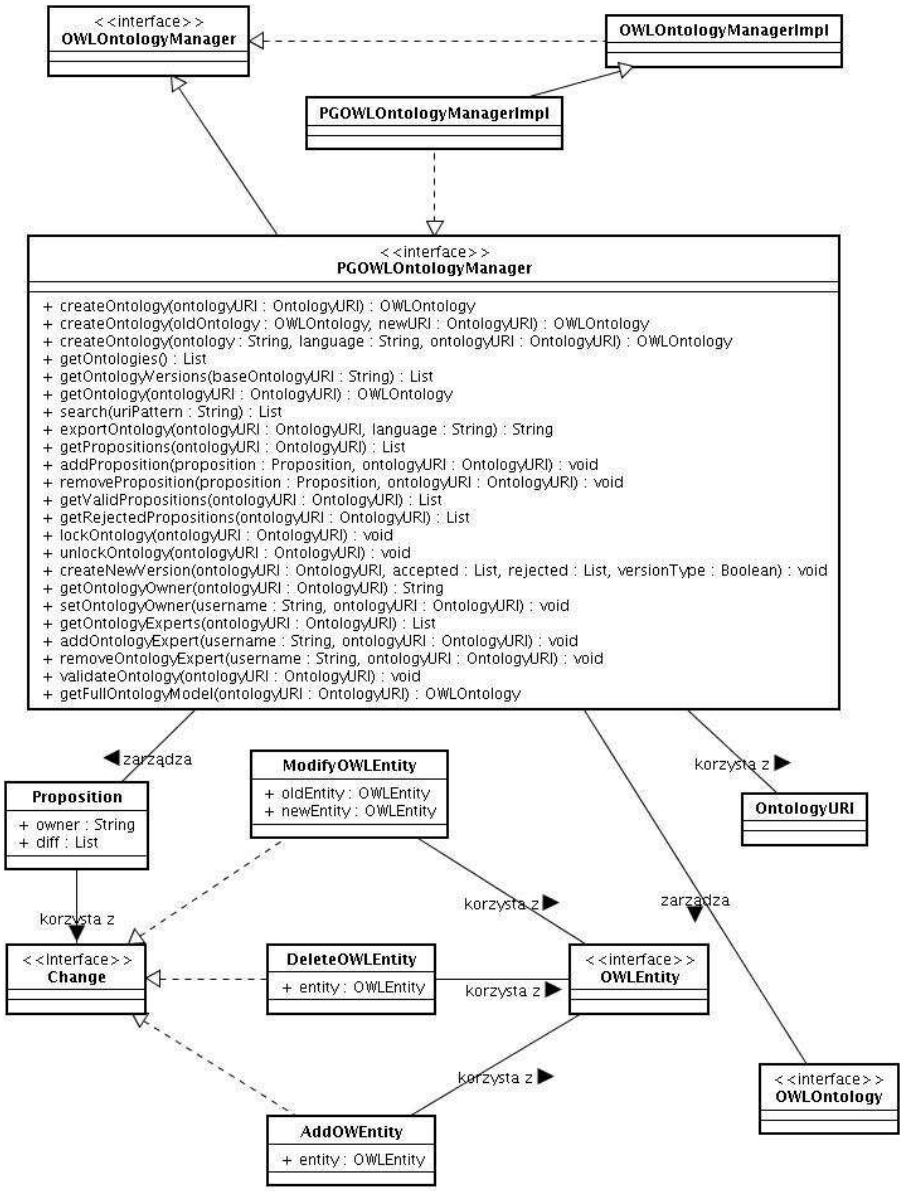
4.2 Łączenie oraz odwzorowywanie ontologii

Wartą wspomnienia jest również część systemu realizowana poza grantem w ramach pracy doktorskiej autora niniejszej publikacji. W trakcie konstrukcji ontologii zachodzi czasami konieczność odwzorowania (poprzez utworzenie wiązań między elementami ontologii) bądź połączenia ontologii w trzecią większą, w której skład wchodzą elementy łączonych ontologii. W ogólności portal realizowany w ramach grantu służyć będzie jako narzędzie testowe dla doświadczeń wykonywanych w ramach wspomnianej pracy doktorskiej.

Funkcjonalność taka może być potencjalnie bardzo ważna w procesie tworzenia ontologii ze względu na możliwość realizacji postulatu ponownego wykorzystania już gotowych elementów a tym samym redukcję nadmiarowości ukazującej się pod postacią wielu ontologii definiujących ten sam fragment wiedzy.

Częścią systemu realizująca powyższe zadania będzie część deweloperska. Odpowiednie metody zawarte zostaną w logice biznesowej a dostęp do nich odbywać się będzie poprzez odpowiednie formatki i aplety interfejsu użytkownika. W założeniu będzie to odrębny moduł integrujący się z portalem, jednak nie będący niezbędny do jego poprawnego funkcjonowania.

Opisywana funkcjonalność może posłużyć także jako wsparcie dla użytkownika, w trakcie tworzenia nowych elementów ontologii, poprzez automatyczne wyszukiwanie potencjalnych konfliktów czy nadmiarowości w redagowanej ontologii, już w momencie utworzenia nowej klasy czy atrybutu.



Rysunek 4: API biblioteki operującej na ontologiach w języku OWL

5 Moduły zastosowań

W przyszłości w tej części systemu znajdą się wszystkie rozwiązania umożliwiające praktyczne zastosowanie ontologii utworzonych za pomocą portalu. Jako repozytorium ontologii dla tej części służyć będzie Warstwa integracji portalu. Każde z rozwiązań stanowić będzie odrębny system komunikujący się z Warstwą integracji portalu ontologicznego poprzez jej API lub za pośrednictwem biblioteki operującej na ontologiach w języku OWL.

6 Metodologia wytwarzania systemu

Metodologia wytwarzania systemu w dużej mierze korzysta z doświadczeń nabytych w trakcie tworzenia systemu BeesyCluster[2]. Przebiegać będzie w kilku etapach. W pierwszym kroku zdefiniowana zostanie Specyfikacja Wymagań szczegółowo opisująca jakie zadania system, jako całość, ma realizować. Na tym etapie określone zostanie dokładnie co system ma robić jednak z pominięciem szczegółów implementacyjnych, czyli jak system ma to robić. Specyfikacja wymagań pozwoli również na doprecyzowanie celów jakie przyświecają przedsięwzięciu oraz jego ram.

Następnym krokiem będzie podział systemu na niezależne fragmenty, które mogą być realizowane w oderwaniu od pozostałych elementów. Po takim podziale należy również dokonać podziału Specyfikacji Wymagań na odrębne dokumenty opisujące zadania jakie realizować ma każda z części systemu.

Po podziale systemu na rozłączne obszary należy zdefiniować interfejsy pomiędzy nimi. Za ich pomocą poszczególne fragmenty będą się komunikować ze sobą na etapie konsolidacji projektu. Jakiekolwiek późniejsze zmiany w interfejsach powinny zostać ograniczone do minimum i uchwalane przy udziale przedstawicieli wszystkich zespołów pracujących nad systemem. W przeciwnym razie ostateczne scalenie systemu będzie niemożliwe do wykonania.

Na tym etapie prac poszczególne zespoły przystąpią do implementacji każdego fragmentu systemu. Respektowanie ustalonych wcześniej interfejsów pozwoli na proste scalenie oraz przetestowanie systemu na dowolnym etapie implementacji każdej z części.

7 Technologie wykorzystane w trakcie tworzenia portalu

Portal jako całość napisany zostanie w technologii J2EE[4] w wersji piątej. Wybór tej technologii pozwoli w znaczącym stopniu na uniezależnienie się od docelowej platformy sprzętowej i systemowej, na której działać będzie portal. Umożliwi również pełną swobodę w wyborze stosowanego przez twórców systemu środowiska tworzenia aplikacji. Ponadto, każdy z elementów systemu, w jaki największej części, utworzony powinien zostać z wykorzystaniem standardowych komponentów dostępnych w standardzie J2EE. Dzięki temu gotowy system spełniać będzie wszelkie standardy współczesnego wytwarzania oprogramowania a jego konstrukcja będzie przejrzysta i prosta w modyfikacji. Ograniczy to też znacznie liczbę linii kodu jaką należy samodzielnie wygenerować.

W celu uniknięcia problemów integracyjnych wynikających z różnic pomiędzy środowiskami wytwarzania aplikacji konieczne jest zastosowanie jednolitego narzędzia do budowy kodu. Najpowszechniej stosowanym rozwiązaniem tego typu jest Apache Ant[3]. Dzięki temu narzędziu kod portalu będzie można w prosty sposób budować i integrować bezkonieczności każdorazowego konwertowania projektu jednego środowiska w projekt drugiego. Apache Ant zapewni również obecność wszystkich niezbędnych bibliotek już na etapie budowy i integracji systemu. W przeciwnym wypadku nie będzie możliwa kompilacja kodu.

Konieczne jest także, by część integracyjna oferowała API dostępne zarówno poprzez zdalne wywołania JavaBeans[4] jak i poprzez Web Services[4][5]. Tworzenie aplikacji w ten sposób umożliwi proste rozwijanie dodatkowych modułów,

czyli zarówno części aplikacyjnej jak i deweloperskiej, a także proste połączenie z systemami zewnętrznymi, takimi jak BeesyCluster. Implementacja obu metod wywołania usług w systemie uchroni twórców przed koniecznością tworzenia w przyszłości dodatkowych warstw pośredniczących tłumaczących np. wywołania poprzez JavaBeans na wywołania zrozumiałe przez Web Services.

Portal utworzony na podstawie przedstawionej w tej publikacji architektury cechować się będzie dużą elastycznością. Ponadto przedstawiona tu propozycja nie narzuca żadnej z grup zajmującej się tworzeniem systemu żadnych konkretnych rozwiązań pozwalając na ich dopracowywanie i optymalizację. Jednoznaczne określenie elementów systemu i zadań każdego z nich, a także zastosowanie wspierającego modułowość standardu J2EE, pozwoli z kolei zintegrować wszystkie komponenty w jeden, kompletny produkt realizujący założenia a także cechującego się prostotą modyfikacji i rozbudowy.

Literatura

[1] Grant KBN N516 035 31/3499, Strategie i procedury tworzenia i negocjacji ontologii dziedzinowych

[2] Czarnul, P., 2006, Reaching and maintaining high quality of distributed J2EE applications - BeesyCluster case study, International Federation for Information Processing, Vol 227. S. 179-190, Springer, New York, U.S.A.

[3] The Apache Software Foundation, Apache Ant, http://ant.apache.org

[4] Sun Microsystems, Inc., 2006, Java(TM) Platform, Enterprise Edition 5 (Java(TM) EE 5) Specification, 4150 Network Circle, Santa Clara, California 95054, U.S.A.

[5] W3C Working Group, 2004, Web Services Architecture, W3C (MIT, ERCIM, Keio)

[6] W3C Working Group, 2004, RDF Primer, W3C (MIT, ERCIM, Keio), http://www.w3.org/TR/rdf-primer/

[7] Horridge, M., Bechhofer, S., Noppens, O., 2007, Igniting the OWL 1.1 Touch Paper: The OWL API. OWLED 2007, 3rd OWL Experienced and Directions Workshop, Innsbruck, Austria, June 2007