

Bazy Danych i Usługi Sieciowe

Drupal

Paweł Witkowski

Wydział Matematyki, Informatyki i Mechaniki

Jesień 2012



Plan wykładu

- 1 CMS, Framework, CMF
- 2 Czym jest Drupal?
- 3 Instalacja
- 4 Konfiguracja
- 5 Moduły
- 6 Szablony graficzne

CMS - System Zarządzania Treścią

- 1 Content Management System
- 2 Rozwiązanie wygodne
- 3 Od razu gotowe do użycia
- 4 Łatwa administracja

Przykłady

- ▶ Wordpress (wordpress.org)
- ▶ Joomla! (www.joomla.org/)
- ▶ DotNetNuke (www.dotnetnuke.com)
- ▶ ExpressionEngine (ellislab.com/expressionengine)

Framework

- 1 Framework
- 2 Dostarcza komponentów do budowy aplikacji
- 3 Wymaga budowania aplikacji od podstaw
- 4 Brak panelu administracyjnego

Przykłady

- ▶ Zend (framework.zend.com)
- ▶ Symfony (symfony.com)
- ▶ Django (www.djangoproject.com)
- ▶ CherryPy (www.cherrypy.org)
- ▶ Ruby on Rails (rubyonrails.org)
- ▶ CakePHP (cakephp.org)
- ▶ CodeIgniter (ellislab.com/codeigniter)

Content Management Framework

- 1 Content Management Framework
- 2 Narzędzie do tworzenia CMS
- 3 Dostarcza narzędzi do obsługi zawartości
- 4 Dostarcza obsługę użytkowników
- 5 Panel administracyjny

Przykłady

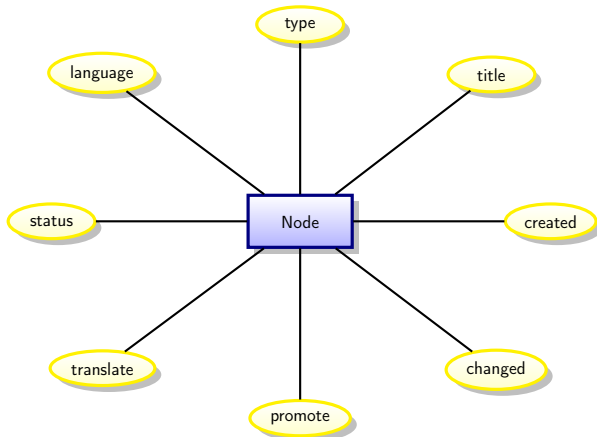
- ▶ Drupal (drupal.org)
- ▶ Typo3 (typo3.org)
- ▶ Symfony CMF (cmf.symfony.com)

Czym jest Drupal?

- 1 Elastyczna aplikacja do tworzenia CMS
- 2 Zawiera podstawową funkcjonalność CMS
- 3 Umożliwia implementację dowolnej funkcjonalności za pomocą
 - ▶ istniejących modułów
 - ▶ modułów własnych

Node

- 1 Node - segment, węzeł
- 2 Abstrakcyjny typ zawartości
- 3 Twórca tworzy specjalizacje - typy zawartości (bundle)



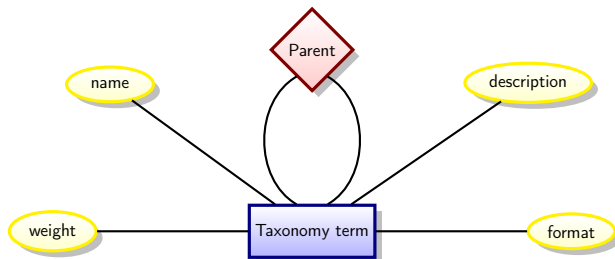
Rysunek : Encja node z wybranymi atrybutami

Przykłady typów zawartości

- Strona - zawartość w HTML z tytułem
- Artykuł - zawartość w HTML z tytułem i obrazkiem
- Wpis na blogu - zawartość w HTML z tytułem, obrazkiem i tagami
- Wydarzenie - zawartość w formacie tekstowym z tytułem i datami
- Produkt - zawartość w formacie tekstowym z obrazkiem i ceną
- Sklep - zawartość w formacie tekstowym z dwoma obrazkami, adresem pocztowym, adresem URL
- Strona podręcznika - zawartość w formacie HTML z zapisywanymi kolejnymi wersjami

Taxonomy

- 1 Taxonomy - kategorie
- 2 Metoda oznaczania
- 3 Klasyfikacja treści według różnego typu słowników
- 4 Terminy mogą mieć terminy nadrzędne (wiele)



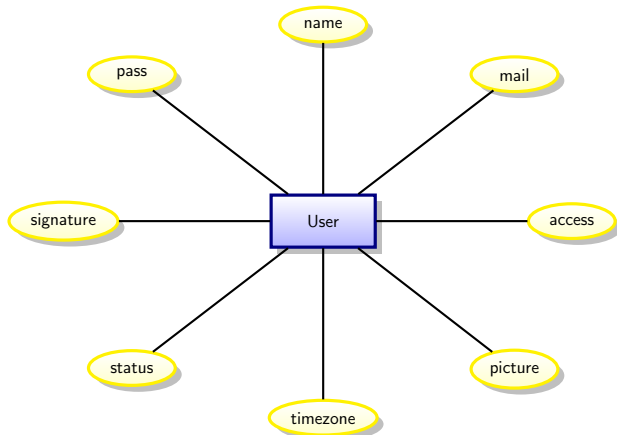
Rysunek : Encja Taxonomy term z wybranymi atrybutami

Przykłady słowników i kategorii

- Sport - snooker, szachy, brydż
- Żywność - nabiał, mięso, warzywa, owoce, używki
- Żywność z hierarchią
 - ▶ nabiał - sery, jogurty, jajka
 - ▶ warzywa - sałaty, fasole, ziemniaki
 - ▶ owoce - jabłka, gruszki, cytrusy
- Sport z hierarchią wielokrotną
 - ▶ zawodowy - snooker, szachy, brydż
 - ▶ amatorski - snooker, szachy, brydż, scrabble

User

- 1 User - użytkownik
- 2 Ma swoje konto i może dokonać uwierzytelnienia
- 3 Może mieć różne role i uprawnienia



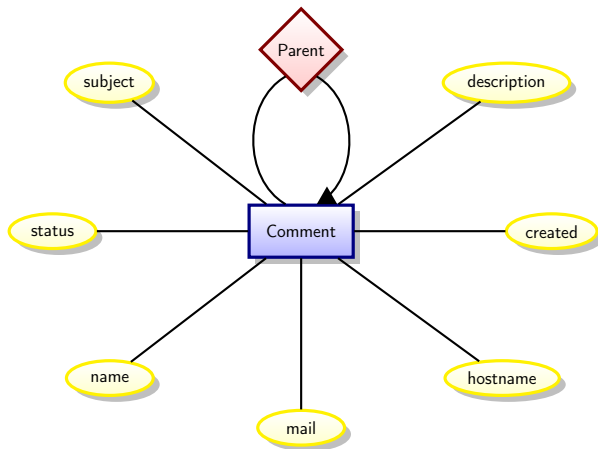
Rysunek : Encja User z wybranymi atrybutami

Przykłady ról i użytkowników

- Administrator - może wszystko
 - ▶ Ewa
- Content administrator - dodaje i edytuje treści
 - ▶ Jan
 - ▶ Olga
 - ▶ Sebastian
- Newsletter administrator - wysyła newsletter
 - ▶ Jan
 - ▶ Patrycja

Comment

- 1 Comment - komentarz
- 2 Zawartość dołączana do node przez użytkowników
- 3 Komentarze mogą mieć terminy nadrzędne (jeden)



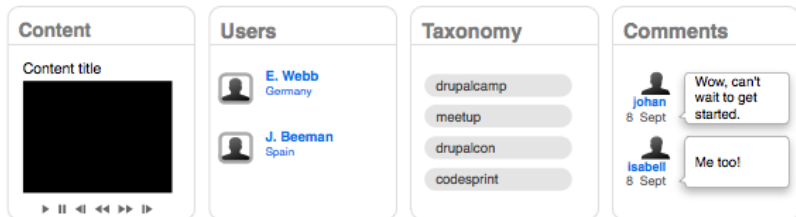
Rysunek : Encja Comment z wybranymi atrybutami

Przykłady komentarzy

- zwykły tekst ułożony w wątki
- tekst w HTML z kategoriami - używany do budowy forum
- zwykły tekst z gwiazdkami do oceniania artykułu
- tekst z adresem URL strony
- zapisy na wydarzenia

Entity

1 Abstrakcyjny typ encji w Drupalu



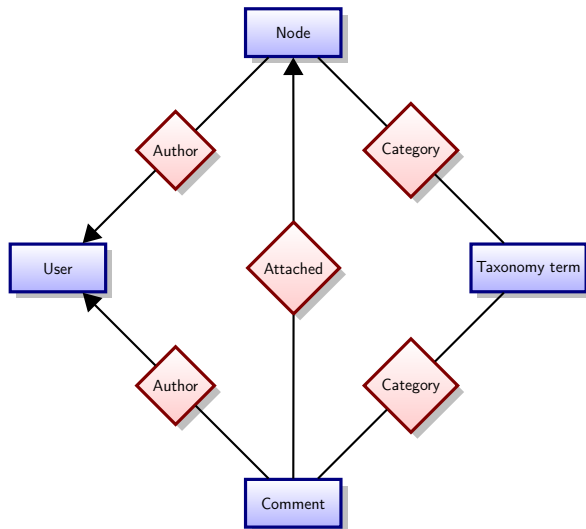
Entity

- ▶ Node
- ▶ User
- ▶ Comment
- ▶ Taxonomy term

Zalety abstrakcyjnych Entity

- ❶ **CRUD**: Create Read Update Delete
- ❷ Unifikacja operacji na danych serwisu
- ❸ Unikanie powtarzania tych samych fragmentów kodu
- ❹ Każda Entity może mieć
 - ▶ Wersje
 - ▶ Unikalny adres URL
 - ▶ Dołączane pola

Diagram związków entities



Rysunek : Diagram związków entities

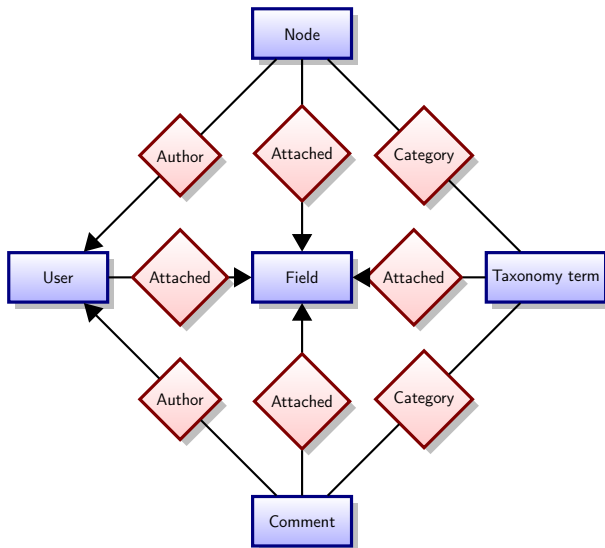
Field

- 1 Field - pole
- 2 Informacja dołączana do entity
- 3 Może być różnych typów, również złożonych
- 4 Może mieć różne kontrolki w formularzu dodawania treści
- 5 Można ją wykorzystywać w różnych entities

Przykłady pól i kontrolek

- Tekst
 - ▶ Lista rozwijana
 - ▶ Pole tekstowe
 - ▶ Pole tekstowe z walidacją
- Liczba całkowita
 - ▶ Lista rozwijana
 - ▶ Pole tekstowe
 - ▶ Suwak
- Lokalizacja
 - ▶ Pola tekstowe na kraj, miasto, ulica, numer
 - ▶ Mapka z możliwością zaznaczania
 - ▶ Pola tekstowe na współrzędne GPS

Diagram związków entities z polami



Rysunek : Encja Field reprezentuje pojedynczą instancję pola

Wymagane składniki systemu

- ❶ Serwer www: Apache, Nginx, IIS
- ❷ PHP
- ❸ Baza danych: MySQL, PostgreSQL, SQLite
- ❹ Można wykorzystać pakiety
 - ▶ XAMPP (www.apachefriends.org/en/xampp.html)
 - ▶ AMPPS (www.ampps.com)
- ❺ Wybrana wersja Drupal (drupal.org/download)

Instalacja

- 1 Założenie bazy danych wraz z użytkownikiem z pełnymi uprawnieniami do niej
- 2 Rozpakowanie plików aplikacji do katalogu serwera www
- 3 Uruchomienie instalatora przez adres URL
- 4 Podanie informacji na temat dostępu do bazy danych
- 5 Podanie podstawowych informacji o stronie i danych administratora

- 1 Views (drupal.org/project/views), CTools (drupal.org/project/ctools)
- 2 Panels (drupal.org/project/panels)
- 3 Pathauto (drupal.org/project/pathauto)
- 4 Token (drupal.org/project/token)

Moduły dla programisty

- 1 Devel (drupal.org/project/devel)
- 2 Theme developer (drupal.org/project/devel_themer)
- 3 Migrate (drupal.org/project/migrate)
- 4 Backup and Migrate (drupal.org/project/backup_migrate)

Moduły dla użytkowników

- 1 CKEditor (drupal.org/project/ckeditor)
- 2 IMCE (drupal.org/project/imce)
- 3 Link (drupal.org/project/link)
- 4 Printer, email and PDF versions (drupal.org/project/print)

Szablony graficzne

❶ Szablony gotowe

- ▶ Bartik
- ▶ Marinelli
- ▶ Danland
- ▶ BlueMasters

❷ Szablony bazowe do tworzenia własnej grafiki

- ▶ Zen
- ▶ Omega
- ▶ AdaptiveTheme