

Bazy Danych i Usługi Sieciowe

Drupal, WeBIAS

Paweł Daniluk
Paweł Witkowski

Wydział Fizyki
Wydział Matematyki, Informatyki i Mechaniki

Jesień 2013



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

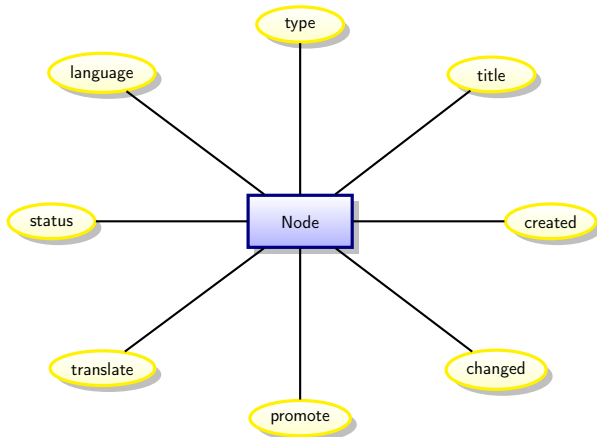
- ▶ Drupal (drupal.org)
- ▶ Typo3 (typo3.org)
- ▶ Symfony CMF (cmf.symfony.com)

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)



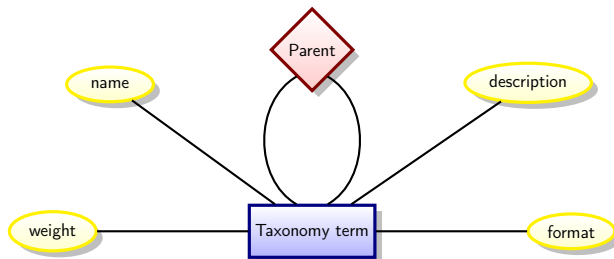
Rysunek : Encja node z wybranymi atrybutami

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)



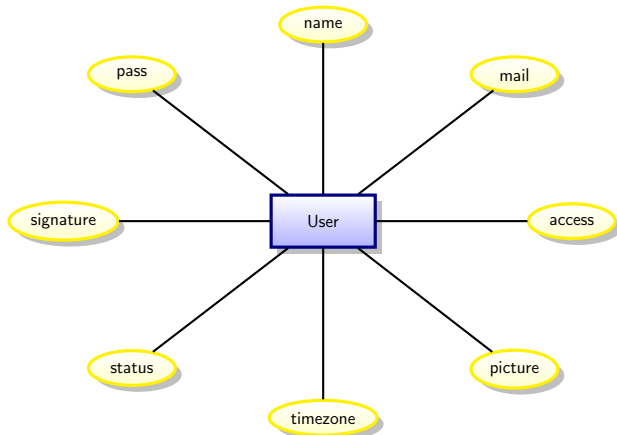
Rysunek : Encja Taxonomy term z wybranymi atrybutami

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



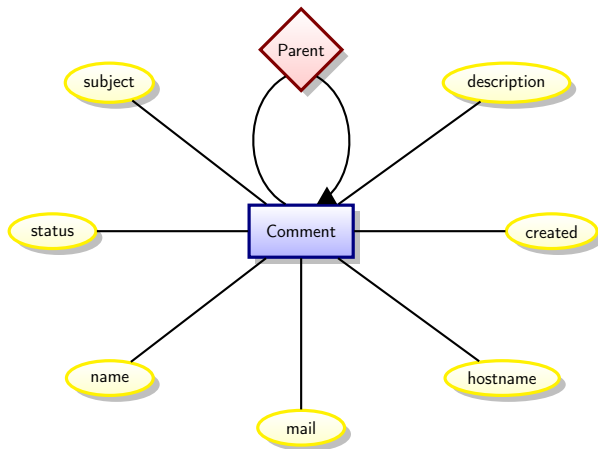
Rysunek : Encja User z wybranymi atrybutami

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)



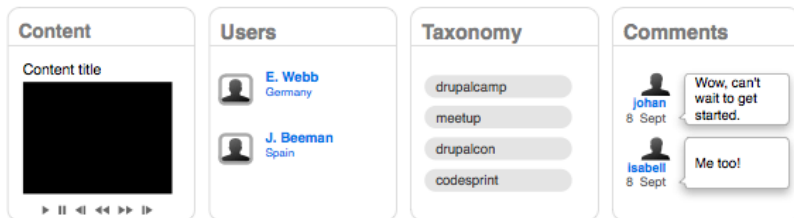
Rysunek : Encja Comment z wybranymi atrybutami

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



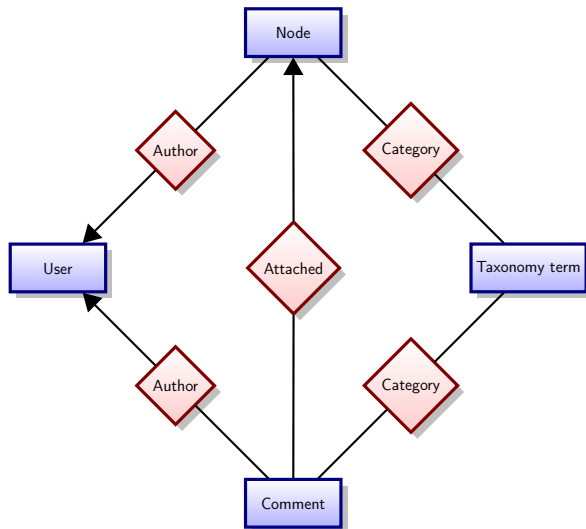
Entity

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

Zalety abstrakcyjnych Entity

- 1 **CRUD**: Create Read Update Delete
- 2 Unifikacja operacji na danych serwisu
- 3 Unikanie powtarzania tych samych fragmentów kodu
- 4 Każda Entity może mieć
 - ▶ Wersje
 - ▶ Unikalny adres URL
 - ▶ Dołączane pola

Diagram związków entities



Rysunek : Diagram związków entities

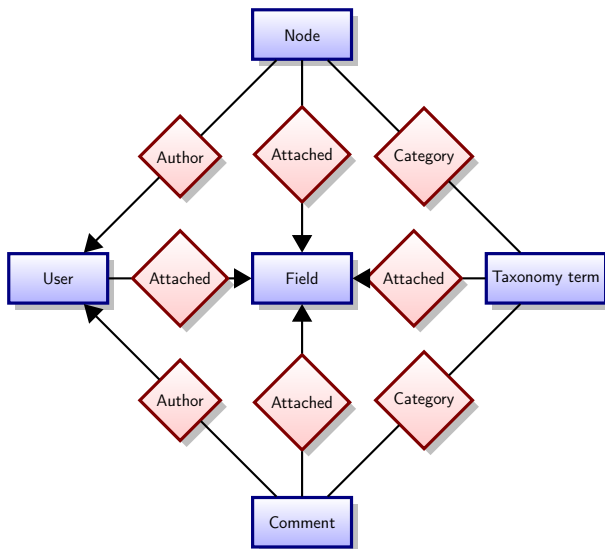
Field

- 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

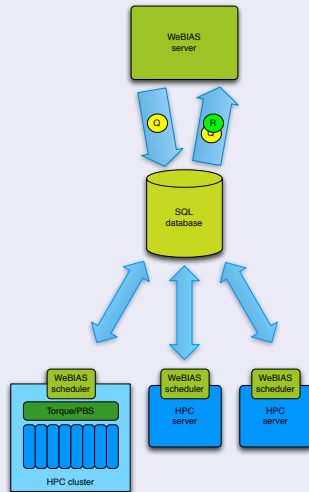
Wymagania

- ❶ Obsługa usług
 - ▶ przyjmowanie zapytań
 - ▶ zarządzanie zgłoszonymi zadaniami
 - ▶ udostępnianie wyników
- ❷ Rejestr użytkowników
- ❸ Obsługa administracyjna

Narzędzie do tworzenia serwisów udostępniających usługi.

Tworzenie serwisu – definicje

- 1 usług
- 2 zasobów obliczeniowych
- 3 konfiguracja (baza danych, katalogi robocze, etc.)



Opis usługi

```
<bias>
<application id="Test" name="Test" info="Hello World.">
  <description>This is the simplest application.</description>
  <author>Bioexploratorium</author>
  <email>pawel@bioexploratorium.pl</email>
  <setup>
    <help_url>/WeBIASDemo/page/Test/help</help_url>
    <param_template>/var/www/WeBIASDemo/examples/test.py --name="${name}">
    <param_table_template>examples/Test/param_table.genshi</param_table_template>
  </setup>

  <parameters>
    <email/>
    <section id="SectionInput" title="Test application"/>
    <text id="name" name="Name" tip="What is your name?" info="name" optional="no">

    <select id="mood" name="How are you?" info="mood" optional="no">
      <option value="0" text="Fine, thanks."/>
      <option value="1" text="Not bad."/>
      <option value="2" text="Don't even ask."/>
      <option value="3" text="Go away, you insensitive clod."/>
    </select>
  </parameters>
</application>
</bias>
```

Formularz zapytania

The screenshot shows a web browser window titled "WeBIAS demo site: Test". The address bar shows "bioinfo.imdik.pan.pl/WeBIASDemo/Examples/Test/". The page has a red header with login instructions: "You may log into the administrator account with the following credentials (login: **admin**, password: **admin**).". Below this is a blue banner with "WeBIAS demo site" and a "Log in" link. A dark blue section titled "Application list" contains the text "Test input form on WeBIAS demo site server." and "This is the simplest application." Below this is a form with a label "Your e-mail address:" and an input field. A blue section titled "Test application" contains labels "Name: *" and "How are you?: *", each with a dropdown menu. The "Name" dropdown is empty, and the "How are you?" dropdown shows "Fine, thanks.". A "Submit" button is at the bottom of the form. The footer says "Administered by Paweł Daniluk" and "Service powered by WeBIAS. Server source code is available under AGPL."

```
<parameters>
  <email/>
  <section id="SectionInput" title="Test application"/>
  <text id="name" name="Name" tip="What is your name?" info="name"
    optional="no" help="#name"/>
```

Formularz zapytania

The screenshot shows a web browser window titled "WeBIAS demo site: Test" with the URL "bioinfo.imdik.pan.pl/WeBIASDemo/Examples/Test/". A red banner at the top contains the text: "You may log into the administrator account with the following credentials (login: **admin**, password: **admin**).". Below this is a blue header with "WeBIAS demo site" and a "Log in" link. A dark blue section titled "Application list" contains the text "Test input form on WeBIAS demo site server." and "This is the simplest application." Below this is a form with a "Your e-mail address:" label and an input field. A blue section titled "Test application" contains a "Name:" label with a red asterisk and a blue help icon, an input field, a "How are you?:" label with a red asterisk and a blue help icon, a dropdown menu with "Fine, thanks." selected, and a "Submit" button. At the bottom, a blue footer contains the text "Administered by [Pawel Daniluk](#)" and "Service powered by [WeBIAS](#). [Server source code](#) is available under [AGPL](#)."

```
<select id="mood" name="How_are_you?" info="mood" optional="no"
  tip="C'mon, just tell me." help="#mood" >
  <option value="0" text="Fine, thanks."/>
  <option value="1" text="Not bad."/>
  <option value="2" text="Don't even ask."/>
  <option value="3" text="Go away you insensitive clod."/>
</select>
</parameters>
```

Reprezentacja zapytania i wyniku

Zapytanie

```
<query>
  <name type="value">John</name>
  <mood type="value">0</mood>
</query>
```

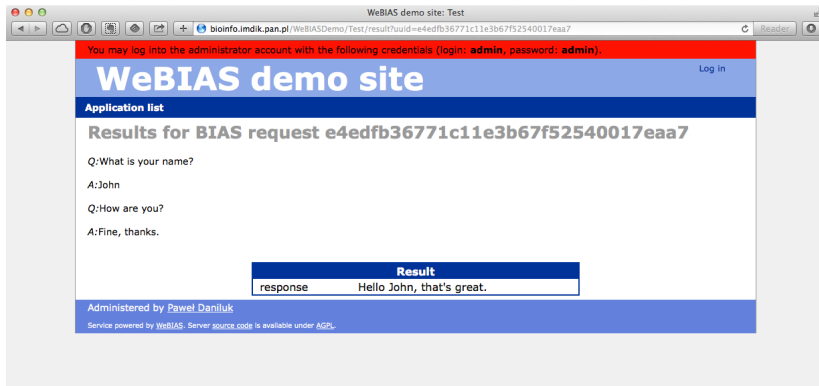
Wywołanie aplikacji

```
<param_template>
/var/www/WeBIASDemo/examples/test.py --name="${name.PCDATA}"
--mood=${mood.PCDATA}
</param_template>
```

Wynik

```
<TestResult>
  <response>Hello John, that's great.</response>
</TestResult>
```


Formatowanie wejścia/wyjścia



The screenshot shows a web browser window titled "WeBIAS demo site: Test" with the URL "bioinfo.imdik.pan.pl/WeBIASDemo/Test/result?uid=e4edfb36771c11e3b67f52540017eaa7". The page has a red header with login instructions: "You may log into the administrator account with the following credentials (login: **admin**, password: **admin**).". Below this is a blue banner with "WeBIAS demo site" and a "Log in" link. A dark blue bar labeled "Application list" precedes the main content. The main content area displays a chat log with the following text: "Q:What is your name?", "A:John", "Q:How are you?", and "A:Fine, thanks.". Below the chat log is a table with a blue header "Result" containing two columns: "response" and "Hello John, that's great.". The footer of the page states "Administered by Paweł Daniluk" and "Service powered by WeBIAS. Server source code is available under AGPL."

You may log into the administrator account with the following credentials (login: **admin**, password: **admin**).

WeBIAS demo site

Log in

Application list

Results for BIAS request e4edfb36771c11e3b67f52540017eaa7

Q:What is your name?

A:John

Q:How are you?

A:Fine, thanks.

Result	
response	Hello John, that's great.

Administered by Paweł Daniluk

Service powered by WeBIAS. Server source code is available under AGPL.

Formatowanie wejścia/wyjścia

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:py="http://genshi.edgewall.org/" py:strip="">
<div id="center">
  <?python
    dict=["Fine ,_thanks.", "Not_bad.",
          "Don't_even_ask.", "Go_away_you_insensitive_clod."]
  ?>

  <p><em>Q:</em>What is your name?</p>
  <p><em>A:</em>${query.tree.name.PCDATA}</p>

  <p><em>Q:</em>How are you?</p>
  <p><em>A:</em>${dict[int(query.tree.mood.PCDATA)]}</p>
</div>
</html>
```

Obsługa plików

- Aplikacje mogą generować pliki wynikowe (np. obrazki), których nie da się wygodnie zakodować w XMLu.
- Można wskazać jakie pliki znajdujące się w katalogu roboczym mają być zachowane.
- Pliki wejściowe i wyjściowe składowane są w bazie danych (jako BLOBy).
- Dostęp przez URL (np. https://bioexploratorium.pl/EP/DEDAL/file?uuid=735ce39c4de611e38a5d00163600000a&pathname=res_2_1.al&runid=129)

Dodatkowe funkcjonalności

Dla użytkownika

- rejestracja użytkowników
- dostęp do archiwalnych zapytań
- edycja zapytań i ponowne wysyłanie
- sprawdzanie poprawności formularza przed wysłaniem (AJAX)

Dla administratora

- raporty i statystyki
- możliwość włączania i wyłączania aplikacji
- kontrola dostępu przez użytkowników
- przeglądanie wszystkich zapytań