Generator de Sisteme Informatice WEB Orientate Utilizator

Vasile Racoviţa [1]*, Gheorghe Căpăţână[1]
[1] *Universitatea de Stat din Moldova*
* vasilii.racovitsa@gmail.com

***Abstract* — It was showed up the concept of the user orientated WEB based Informational System Generator. There are the characteristics of WEB based applications for database driven informational management, build for final users. There are delimited relational database field types that demands automation in the process of final user interfaces generation and the modules used to upgrade the convenience of the Human-Computer-Interaction. There is developed an on-line application and a test case for it.**

***Index Terms* — database driven informational management, Informational System Generator, meta-informational types, user interfaces generation, WEB based applications.**

I. INTRODUCERE

În *sistemele informatice* (abreviat *SI*) păstrarea informaţiilor se face cu ajutorul *sistemelor de gestiune a bazelor de date* (abreviat *SGBD*).

În aplicaţiile *WEB* administrarea informaţiei, cât şi a structurii *bazei de date* (abreviat *BD*) este organizată de către utilizatorii finali cu ajutorul *interfeţelor*. Interfeţele prezintă un produs program intermediar, care asigură dialogul între utilizatorul final şi *SGBD SQL*. Componenta *SQL* administrează *BD* ale aplicaţiilor.

Pentru a rezolva probleme pe calculatorul electronic utilizatorul final trebuie să piloteze componentele software menţionate, adică să demonstreze un anumit nivel de cunoştinţe din domeniul informaticii.

Interfeţele de administrare a *BD* prezintă nişte *forme* (*eng., forms*). În procesul elaborării unei noi *forme*, fiecărui element structural al *BD* i se pune în corespondenţă un element de tipul: *textbox, textare, radio button, check box* etc. Dacă butoanele sunt elaborate cu ajutorul limbajului *HTML*, atunci acest proces poate fi anevoios şi din cauza limitărilor funcţionale ale acestui limbaj.

Elaborarea sistemelor de administrare a informaţiei unei *BD* se face într-un oarecare limbaj de programare sau în limbajul *SGBD*. Concomitent cu dezvoltarea aplicaţiilor *WEB* – iar acestea apelează la *SGBD* - creşte considerabil numărul utilizatorilor finali, care sunt nevoiţi să administreze *SGBD* şi să elaboreze noi interfeţe şi forme. Limbajele *HTML* şi *SQL* sunt imperative. De aceea, pentru a permite utilizatorului final (expertului în domeniul de aplicaţie dar şi non-informaticianul) exploatarea şi dezvoltarea *SI* cu minim de cunoştinţe speciale în informatică, *SI* trebuie elaborate în aşa mod, ca să asigure funcţia de traducător al comenzilor de administrare a *BD*, expuse de către utilizatorului final din *limbajul utilizatorului final* în *comenzile SGBD*.

Autorii şi-au pus scopul de a propune un model de generator de *SI* pentru *WEB*, o structură a acestuia şi mecanisme de funcţionare atât a generatorului cât şi a produselor program generate. Aceste exigenţe au fost realizate cu scopul de a permite dezvoltarea şi mentenanţa generatorului şi a produselor program *WEB*, elaborate cu ajutorul acestui generator de către însăşi utilizatorul final fără asistenţă obligatorie din partea informaticianului.

În lucrare este expus conceptul elaborării *Generatorului de Sisteme Informatice* (abreviat *GSI*) *WEB* orientate utilizator.

GSI oferă utilizatorului final concomitent următoarele abilităţi: a) *un mediu software* pentru exploatarea unei familii de *SI WEB* orientate utilizator; b) un *mediu instrumental WEB*, care asistă utilizatorul final în procesul dezvoltării *SI* pentru noi aplicaţii şi adaptarea acestora la evoluţia domeniului de aplicaţie.

II. DESTINGEREA TIPURILOR META-INFORMAŢIONALE ALE BD WEB

Tipurile *meta-informaţionale* diferă de tipurile câmpurilor tabelelor unei *BD*. *BD* sunt destinate pentru a păstra informaţia în anumite tipuri primare (numeric, textual, binar, calendaristic etc.) şi, de regulă, nu oferă utilizatorului final informaţii suficiente pentru organizarea procesului de adăugare/editare a informaţiei păstrate în ele. Spre deosebire de tipurile primare, tipurile *meta-informaţionale* furnizează *GSI* cunoştinţe despre însăşi informaţia păstrată în *BD*. Aceste cunoştinţe sunt utilizate în *GSI* pentru organizarea procesului de administrare şi dezvoltare a produsului program de către însăşi utilizatorul final fără asistenţa obligatorie a informaticianului.

Pentru a elucida situaţia descrisă prezentăm unele exemple.

1. Deseori, sistemele de management al conţinutului site-urilor (*eng., Content Management System,* abreviat: *CMS*) păstrează în *BD* conţinutul paginilor statice ale site-ului în formatul *HTML*. În acest caz, tipul câmpului tabelei va fi „*Text*” sau un derivat al acestuia. De asemenea, în câmpurile *BD* de tipul *Text* pot fi păstrate şi alte tipuri de date (spre ex.: *XML*, *CSV*, *text neformatat* etc.). Deci, tipul câmpului nu identifică tipul informaţiei păstrată în el, iar organizarea comodă (pentru utilizator) a procesului de administrare a informaţiei nu este posibilă fără informaţii adăugătoare (*meta-informaţii*) sau implicarea informaticienilor.

2. În unele cazuri, când totuşi este posibil de obţinut cunoştinţe despre tipul informaţiei păstrate în câmpurile tabelelor (spre exemplu, câmpurile de tip „*data*”), nu

întotdeauna este posibil de generat interfeţele *WEB* standard pentru administrarea comodă a *BD*. Şi aceasta este din cauza lipsei elementelor limbajului *HTML* adecvate necesităţilor de administrare.

Spre exemplu, dacă într-un câmp al tabelei este nevoie de păstrat o *dată calendaristică*, este firesc, că tipul câmpului dat să fie „*data*" (sau o derivată a acesteia). Ştiind acest fapt, sistemul de administrare a informaţiei poate să delimiteze tipul datelor păstrate. În asemenea caz un mod comod de administrare ar putea fi considerat acela, care la o eventuală inserare sau editare a informaţiei dintr-un câmp de acest tip, utilizatorului i s-ar propune un calendar stilizat pentru culegerea datei. În plus, această metodă de introducere a datei ar minimiza erorile de înserare a datelor provenite de la diferenţele în standardele de reprezentare a datelor calendaristice. Remarcăm, că limbajul *HTML* nu poate propune un astfel de element standard al formei.

Tipurile *meta-informaţionale* utilizate des în administrarea informaţiei în aplicaţiile *WEB*, sunt:

Data calendaristică. Utilizarea acestui tip a fost discutată în exemplul de mai sus.

Conţinutul HTML. În domeniul existent al aplicaţiilor on-line, există aşa numitele aplicaţii *WYSIWYG* (*eng., WYSIWYG - What you see is what you get - ceea ce vezi aceea şi primeşti*), principala menire a cărora este de a oferi facilităţi utilizatorilor neiniţiaţi în limbajul *HTML* pentru crearea paginilor *WEB* sau a unor componente ale acestora. Acestea aplicaţii sunt dotate cu interfeţe clare pentru utilizator. Rezultatul utilizării aplicaţiilor *WYSIWYG* este un bloc *HTML*, care poate fi utilizat în continuare independent şi în alte aplicaţii.

Culoarea în formatul RGB. Deseori, în aplicaţiile *WEB*, utilizatorii solicită selectarea culorii cu care se stilizează o parte sau un element al site-ului. În limbajul *HTML*, culoarea este definită cu un număr hexazecimal din 6 cifre în intervalul valorilor 000000 – FFFFFF. Primele 2 poziţii revin culorii roşii, poziţia 3 şi 4 culorii verzi iar ultimele 2 poziţii - culorii albastre. Unui utilizator neexperimentat (dar şi cel experimentat depune efort, atenţie şi timp suplimentar în comparaţie cu varianta discutată mai jos) îi vine destul de greu să indice culoarea în formatul dat. O soluţie comodă de alegere a culorii ar fi existenţa pe pagina *WEB* a unei porţiuni, în care să fie indicate culorile posibile din care utilizatorul ar alege culoarea potrivită.

Valoarea cheii externe. Majoritatea *SGBD*-urilor utilizate în aplicaţiile *WEB* sunt relaţionale. *BD* relaţionale, aduse la o stare suficientă de normalizare, au un număr enorm (în dependenţă de structura tabelelor şi domeniul informaţiilor păstrate) de câmpuri pentru cheile externe. În procesul lucrului cu aceste tabele, în locul valorii cheii este necesar de afişat valoarea câmpului informativ din tabelul respectiv. În momentul inserării sau editării unei valori a unui oarecare câmp ar fi comod de afişat lista valorilor admisibile.

Câmpurile booleene. Deseori în cadrul *SI* pentru *WEB* este necesar de oferit utilizatorului final facilităţi de administrare a câmpurilor logice, valoarea cărora este 0 sau 1 (*true* şi *false*). Aceste câmpuri sunt utilizate atunci, când este nevoie de indicat, de exemplu, dacă o înscriere în tabel este activă sau nu; ori când este nevoie de indicat că un rând din tabel este şters la nivelul logic fără a fi şters la nivelul fizic (adică pentru ca acest rând să nu fie afişat în interfaţa de administrare, dar toate legăturile istorice cu el să fie păstrate). În situaţiile descrise, ar fi comod ca utilizatorului să i se ofere posibilitatea de a administra informaţia din câmpul dat, fără condiţionarea cunoaşterii valorilor.

GSI are în dotare aceste tipuri informaţionale ale câmpurilor tabelei unei *BD* şi, în caz de necesitate, utilizând tehnicile de *generalizare şi concretizare* [1, 2], permite utilizatorului final declararea şi administrarea şi a altor tipuri informaţionale de date.

III. STRUCTURA GSI

Proiectarea *GSI* consideră unele exigenţe. Datele de intrare ale *GSI* sunt *fişierele text cu meta-date.* Meta-datele sunt datele de descriere a structurii *BD*, care urmează a fi gestionate de către aplicaţia dezvoltată. Tabelele acestei *BD* trebuie să fie create în *SGBD*. Ieşirea *GSI* o constituie *interfeţele de administrare a informaţiei aplicaţiei* dezvoltate. Aceste interfeţe trebuie să asigure atât posibilităţile funcţionale de adăugare, editare şi ştergere a informaţiei din tabelele respective, cât şi facilităţile de operare cu câmpurile, care se referă la unul din meta-tipurile descrise mai sus.

Nivelul de confort este o noţiune subiectivă, asigurată de modulele funcţionale independente de sistem şi care, în caz de necesitate, pot fi dezvoltate. Prin confort se înţeleg facilităţile pe care şi le doreşte utilizatorul final.

În consecinţă, *GSI* permite dezvoltarea unei familii de aplicaţii.

Structura *GSI* este prezentată în Figura 1.

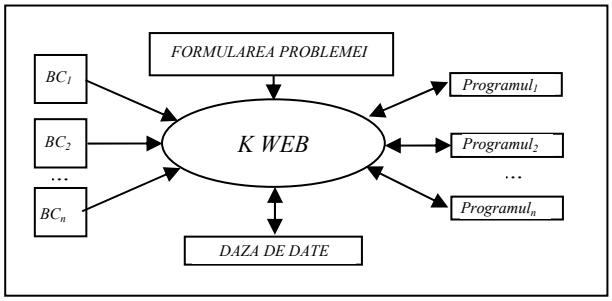

Fig 1. Structura *GSI*

GSI constă din următoarele componente:

BC1 – BCn sunt bazele de cunoştinţe ale *GSI*. Acestea furnizează cunoştinţele despre modul de prelucrare a fiecărui tip meta-informaţional oportun în fiecare aplicaţie.

Formularea problemei este partea sistemului în care se configurează aplicaţia şi sunt definite relaţiile dintre câmpurile tabelelor *BD* şi tipurile meta-informaţionale care le corespund.

Baza de date păstrează informaţiile aplicaţiilor.

Programul1 – Programuln sunt programele generate de *GSI*. Acestea reprezintă interfeţe ale utilizatorului final de administrare a informaţiei din *BD*.

K WEB este constructorul aplicaţiilor *WEB* al *GSI*.

IV. INSTRUMENTELE UTILIZATE LA ELABORAREA *GSI*

La elaborarea *GSI* a fost utilizat limbajul scriptic de programare *PHP 5.2.* [3]. În calitate de server al bazelor de date este utilizat *SGBD MySQL* [4] iar în calitate de server

WEB este utilizat *Apache 2.2* [5]. Pentru facilitarea lucrului cu limbajul *JavaScript* şi tehnologia *AJAX*, este utilizată biblioteca cu sursă deschisă *jQuery* [6]. Asigurarea editării *on-line* a conţinutului *HTML* se face cu aplicaţia liberă pentru utilizare *FCKEditor* [7].

La generarea interfeţelor corespunzătoare *GSI* utilizează în calitate de date de intrare, informaţiile despre structura tabelelor separate în fişiere aparte pentru fiecare tabel. Fişierele respective conţin masive asociative ale limbajului *PHP*. Descrierea câmpurilor masivului este redată în *tabelul 1*.

Tabelul 1.

Descrierea câmpurilor tabelei

Cheia masivului	***Valorile posibile***
Type – descrie tipul câmpului	***Id*** – cheia primară a tabelei
	Key – cheia externă
	Active – valoarea booleană (0 sau 1)
	Text – secvenţa scurtă de text simplu
	TextArea – secvenţa lungă de text simplu
	FCKEditor – editor *on-line* de blocuri *HTML*
	Date_picker – modul de selecţie a datei calendaristice
	Color_picker – modul de selecţie a culorii în formatul *RGB*
Name – denumirea câmpului	**Caractere alfa-numerice**
Insert_type – tipul datelor pentru inserare	***Int*** – pentru date numerice
	Str – pentru date textuale

V. DESCRIEREA MODELULUI DE TESTARE

Expunem un test al *GSI* folosit pentru validarea principiilor de elaborare a generatorului de aplicaţii *WEB*, descrise anterior. Diagrama tabelelor proiectului este reprezentată în Figura 2. Administratorul a creat fişierele de configurare corespunzătoare tabelelor proiectului. Un exemplu al conţinutului acestora este prezentat în *tabelul 2*. Utilizând configurările din fişierele respective, *GSI* a generat o aplicaţie *WEB* ce reprezintă interfeţele de administrare a informaţiei din *BD* şi corespunde principiilor *SI* orientat utilizator.

Tabelul 2.

Conţinutul fişierelor de configurare

Table_pages.cfg.php

```
<? $cfg= Array('table_data'=>Array(
      'table_name'=>'Pages Main'),
'fields'=>Array('id'=>Array (
      'type'=>'id','name'=>'Nr',
      'insert_type'=>'int'),
'url_name'=>Array('type'=>'text',
      'name'=>'URL Name',
      'insert_type'=>'str'),
'title_ru'=>Array('type'=>'text',
      'name'=>'Titlul Ru',
      'insert_type'=>'str'),
'keywords_ru'=>Array('type'=>'text',
      'name'=>'Keywords Ru',
      'insert_type'=>'str'),
'description_ru'=>Array('type'=>'text',
      'name'=>'Description Ru',
      'insert_type'=>'str'),
'content_ru'=>Array(
      'type'=>'fckeditor',
      'name'=>'Content Ru',
      'insert_type'=>'str',),
'title_md'=>Array('type'=>'text',
      'name'=>'Titlul Md',
      'insert_type'=>'str'),
'keywords_md'=>Array('type'=>'text',
      'name'=>'Keywords Md',
      'insert_type'=>'str'),
'description_md'=>Array('type'=>'text',
      'name'=>'Description Md',
      'insert_type'=>'str'),
'content_md'=>Array(
      'type'=>'fckeditor',
      'name'=>'Content Md',
      'insert_type'=>'str'),
'title_en'=>Array('type'=>'text',
      'name'=>'Titlul En',
      'insert_type'=>'str'),
'keywords_en'=>Array('type'=>'text',
      'name'=>'Keywords En',
      'insert_type'=>'str'),
'description_en'=>Array('type'=>'text',
      'name'=>'Description En',
      'insert_type'=>'str'),
'content_en'=>Array(
      'type'=>'fckeditor',
      'name'=>'Content En',
      'insert_type'=>'str'),
'include_path'=>Array('type'=>'text',
      'name'=>'Path if type=script',
      'insert_type'=>'str'),
'id_banner_right'=>Array('type'=>'key',
      'name'=>'Right Side Banner',
      'insert_type'=>'int'),
'id_banner_bottom'=>Array(
      'type'=>'key',
      'name'=>'Bottom Side Banner',
      'insert_type'=>'int'),
'active'=>Array('type'=>'active',
      'name'=>'Activ',
      'insert_type'=>'int'),),
'keys'=>Array('id_banner_right'=>Array(
    'data'=>Array(
    'table_name'=>'banners_right_main',
    'id_field_name'=>'id',
    'show_field_name'=>'name')),
'id_banner_bottom'=>Array(
    'data'=>Array('table_name'=>
                  'banners_bottom_main',
    'id_field_name'=>'id',
    'show_field_name'=>'name',
    'filter_field_name'=>'')))))?>
```

VI. CONCLUZII

Pe parcursul cercetărilor au fost delimitate: a) tipurile câmpurilor, care solicită automatizare în procesul de elaborare a interfeţelor utilizatorilor finali şi b) bibliotecile existente, utilizarea cărora permite un spor de confort

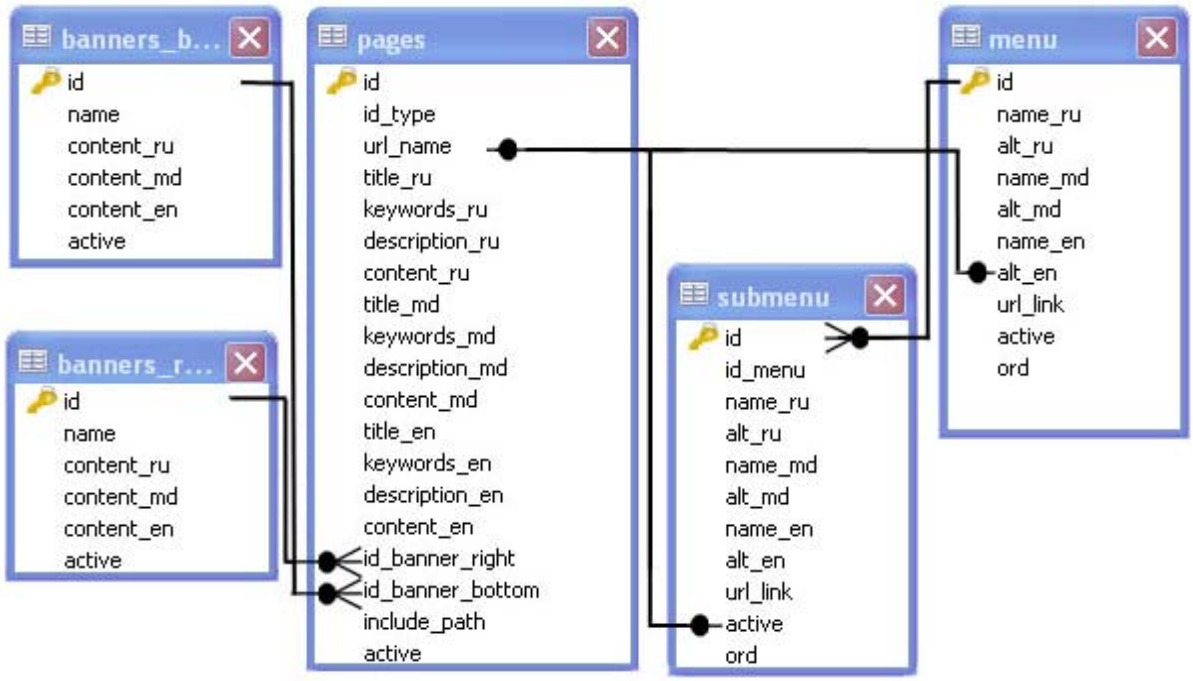

Fig. 2 . Diagrama tabelelor proiectului

utilizatorului final în procesul asistării rezolvării problemelor de profil.

A fost elaborat un *GSI WEB*, care permite utilizatorului final generarea interfeţelor comode de administrat a informaţiilor tabelelor bazei de date în baza fişierelor de configurare, de dezvoltare şi adaptare a aplicaţiilor *WEB*.

A fost expusă o metodă originală de proiectare a unui generator de aplicaţii *WEB*, orientate utilizator şi cu ajutorul lui de proiectare a unei familii de aplicaţii *WEB*.

GSI a demonstrat realizarea exigenţelor formulate şi viabilitatea principiilor funcţionale şi tehnologice descrise în lucrare.

REFERINŢE

[1] Căpăţână Gh. Experienţa elaborării sistemelor informatice orientate la problemă. În: Studia Universitatis, Nr.2. – Chişinău: CEP USM, 2007, pp. 23-28.

[2] Căpăţână Gh., Organ A. Sistem informatic orientat pe probleme. In: „Microectronics and Computer Science”, Int. conf. (5, 2007, Chişinău). Proceeding of the 5tm International Conference on „Microectronics and Computer Science”, sept. 19-21, 2007, Chişinău, Moldova / com. org: I. Balmuş, V. Ababii. - Chişinău: U.T.M., 2007, - p. 225-228.

[3] http://www.php.net/docs.php

[4] http://dev.mysql.com/doc/

[5] http://apache.org/

[6] http://docs.jquery.com/Main_Page

[7] http://docs.fckeditor.net/