Web application refactoring and self-adaptation to hosting platform change: an educational web portal case study

Ljubica Kazi[1], Zoltan Kazi[1], Dragica Radosav[1], Dijana Karuovic[1], Ivana Berkovic[1], Biljana Radulovic[1] and Tatjana Lojovic[2]

[1]*University of Novi Sad, Technical Faculty "Mihajlo Pupin", Djure Djakovica bb, Zrenjanin,Serbia*
[2]*Preschool Institution, Karadziceva 3a, Zrenjanin, Serbia*

Abstract

This paper presents a web portal created for Preschool institution in Zrenjanin, Serbia. This portal is created to present most relevant information and it is based on underlying CMS (Content Management System). The web portal was created in 2014 and refactored in 2020, in aim to adjust to new PHP hosting platform. In solving problem of uncertain exact time of hosting upgrade (shift from PHP 5 to PHP 7 support), it was necessary to include automation in detecting and adjustments to hosting platform change. Therefore, self-adaptation mechanism was developed and included in the solution, to provide constant availability of web portal, regardless the hosting platform change. Self-adaptation mechanism is based on sensor-effector approach.

Keywords

Self-adaptive software, PHP, web portal, Content Management System, software refactoring

1. Introduction

Software changes are inevitable, since their application is closely related to constant advancements in hardware and software technology, as well as organizational transformations. Therefore, software maintenance becomes one of the most relevant areas in software lifecycle, which implies the need for systematic approach and standardization [1]. Lehman [2] has defined software evolution with eight laws that present the continual adjustments of software to user needs and requirements, which often leads to functional improvements and increase of complexity.

With introducing agile approach to software development [3], "responding to change over following a plan" became one of the most important approaches. One of particular areas of interest is research related to agile web development methodologies [4]. Prediction of future software change requests attracts attention, particularly related to frequency analysis [5], as well as in the context of Agile Software Engineering [6]. Usually software change requests have their roots in client requirements which are business-process changes elicited, while less frequently software changes are triggered by software and hardware technology change. User-induced change requests can be resolved more quickly (within existing technology stack), while technology-related software change requests sometimes require system reengineering, which is time consuming. Recent research endeavors are related towards automation of software change requests assignments [7], to improve efficiency in this area. Important aspect of software development includes selection of technology stack that could be used for software design and development of digital platforms [8], but this selection also affects software maintenance efficiency.

ITAMS 2021 – Information Technologies: Algorithms, Models, Systems, September 14, 2021, Irkutsk, Russia
EMAIL: ljubica.kazi@tfzr.rs (A. 1); zoltan.kazi@tfzr.rs (A. 2); dragica.radosav@tfzr.rs (A. 3); dijana.karuovic@tfzr.rs (A. 4); ivana.berkovic@tfzr.rs (A. 5); biljana.radulovic@tfzr.rs (A. 6); pomdir@predskolskazr.edu.rs (A. 7)
ORCID: 0000-0001-8089-3029 (A. 1); 0000-0001-6620-6384 (A. 2); 0000-0002-0540-6910 (A. 3); 0000-0002-2536-2273 (A. 4); 0000-0003-2366-3942 (A. 5); 0000-0002-5358-9010 (A. 6)

This paper presents an approach to solve the problem of web applications adjustments to hosting platform change, i.e. upgrade. It presents the concept and implementation of the self-adaptive software solution that is able to adjust to the change of hosting platform technology. The approach is illustrated with a case study of maintenance of the official web portal for Preschool Institution in Zrenjanin, Serbia http://www.predskolskazr.edu.rs/. Particularly, it describes refactoring and self-adaptive mechanism implementation in web application to support change of hosting platform from PHP5 to PHP7 without any web portal functionality impact.

The rest of the paper is organized as follows: second section presents background with theoretical concepts, section three related work, section four functional and structural design of developed web portal, section five the proposed approach to hosting platform related self-adaptation mechanism; section six conclusions and future work, followed by acknowledgements and references list.

2. Theoretical Background

The concept of adaptive software is defined as a software that dynamically adjusts behavior during execution, as a response to changes in working environment [9, 10]. In [11], self-adaptive software is defined as a system with closed feedback, where the control information is gathered from:

- The system itself ("self" component – relates to the state of the system, i.e. system architecture elements)
- The working environment ("context" component – represents all that is in the operative environment and that could influence behavior and characteristics of the system).

It is relevant to make distinction in terminology [11] between:

- Adaptive software (or self-adaptive) – adaptation mechanism is included in the software solution and automatically adjusts the behavior of software.
- Adaptable software – adaptation mechanism is placed externally in relation to the software core functionality, adaptation mechanism is usually used for manual adjustments from human user.

Figure 1. presents the "sensors-effectors" approach that is a basis for the process of self-adaptation, according to [11]. Sensors enable monitoring upon the events in the working environment, where they detect changes that may occur. System includes decision module, where information from sensors are a basis for initiating activities to respond to changes. Effectors get directions from the decision module and implement the changes.

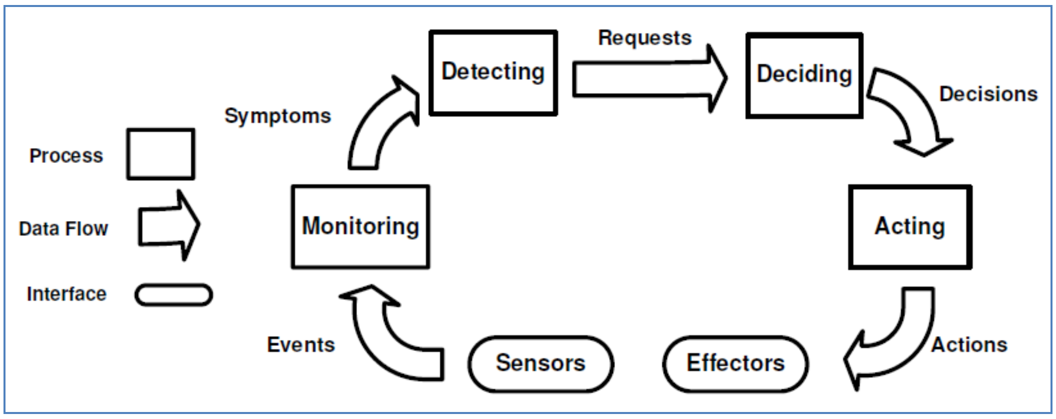


Figure 1. The "sensors-effectors" approach in self-adaptive systems [11]

3. Related work

Software adaptability and adaptivity has been developed by introducing appropriate concepts and practical implementations. Some of relevant terms that were introduced in related work include: basic adaptivity vs. intelligent adaptation [12], micro-adaptation (self-adaptation) vs. macro-adaptation (human-dependent adaptation) [13]

Implementation-related research of software adaptivity has addressed human perspective of software adaptivity (i.e. adaptive user interface), operative environment and working context, as well as software architecture [12, 13]. Diversity of approaches to support implementation of software adaptivity include: interactive programming [12], rule-based systems [13, 14], model-based systems [15, 16], graph-based adaptivity [13], software agents [14, 15]. Special aspects consider run-time adaptivity [13, 16] and software evolution [16].

Concept of adaptivity and self-adaptivity has been analyzed and implemented within variety of software architectures and platforms: multimedia web-based systems [17], distributed software systems [18] , Embedded Systems [19], software based on service-oriented architecture [20], mobile applications [21].

Self-adaptivity of web applications are mostly related to automated adaptation in relation to users and their behavior [22, 23, 24, 25]. Structure aspect of self-adaptive software, particularly web applications has been explored in [26, 27]. Diversity of techniques have been proposed and implemented, particularly for web applications adaptivity:

- Model-based adaptive web applications [25, 28]
- Component-based adaptive web applications [26]
- Active rules application in runtime adaptivity[29]
- Service-oriented adaptive web applications[30]
- Adaptivity for interoperability with devices and services [20].

Platform-related self-adaptivity has been addressed in PhD Thesis [31] that addresses context management and situation-aware smart software systems.

4. Design of Preschool Web Portal

4.1. Development History

First version of web portal of Preschool Institution Zrenjanin has been designed and implemented in 2014 as a PHP5/MySQL web application, in aim to present contents that were previously supported by WordPress solution. This solution has been presented in [31].

Refactoring of this solution from 2014 has been conducted in 2020, in aim to transform existing solution to be active at new hosting platform – PHP 7. Graphical design and functionality remained the same in 2020 version, but the underlying implementation was changed by refactoring to object-oriented solution and it now included hosting platform-related self-adaptive mechanism.

The similar idea of hosting platform-related self-adaptive mechanism was implemented in 2018, within the same institution web portal subsection, related to a particular project of supporting interinstitutional data integration and decision support (published in [32]), but it was not implemented for the primary web portal of Preschool Institution, until 2020. Therefore, the solution of implementing hosting platform-related self-adaptive mechanism for the primary web portal of Preschool institution Zrenjanin, Serbia (www.predskolskazr.edu.rs) [33] is presented in this paper.

4.2. The Functionality of Preschool Web Portal

Preschool web portal [33] was designed as a multi-page PHP-based web application, which has two mayor parts:

- Administrative module (personalized for web portal administrator) – Content management system, that is used to enable administrator to enter texts and upload documents and images, as additional material with texts.
- Presentation module (non-personalized, any web portal visitor) – with static pages (pure HTML) and dynamic pages (reading content recorded within the administrative module.

Following table presents list of web pages and their use:

Table 1. Structure of Preschool web portal with static and dynamic pages and CMS

PRESENTATIONAL PART		**ADMINISTRATIVE PART (Content Management System - CMS)**
STATIC (html)	**DYNAMIC (php / mysql)**	
Electronic children admissions	Home page / News	Adding news (data are categorized and presented in news, events and procurements pages)
About – institution history and work organization	Events	Tabular presentation of news
Kindergardens	Public procurements	News deletion
Advices		Upload of files (attachments to news)
Gallery		Attachments description data entry
Contact		Tabular presentation of all attachments
		Deleting attachments records

Next figure presents UML USE CASE diagram with presentation of software functions supported in both segments – presentational and CMS.

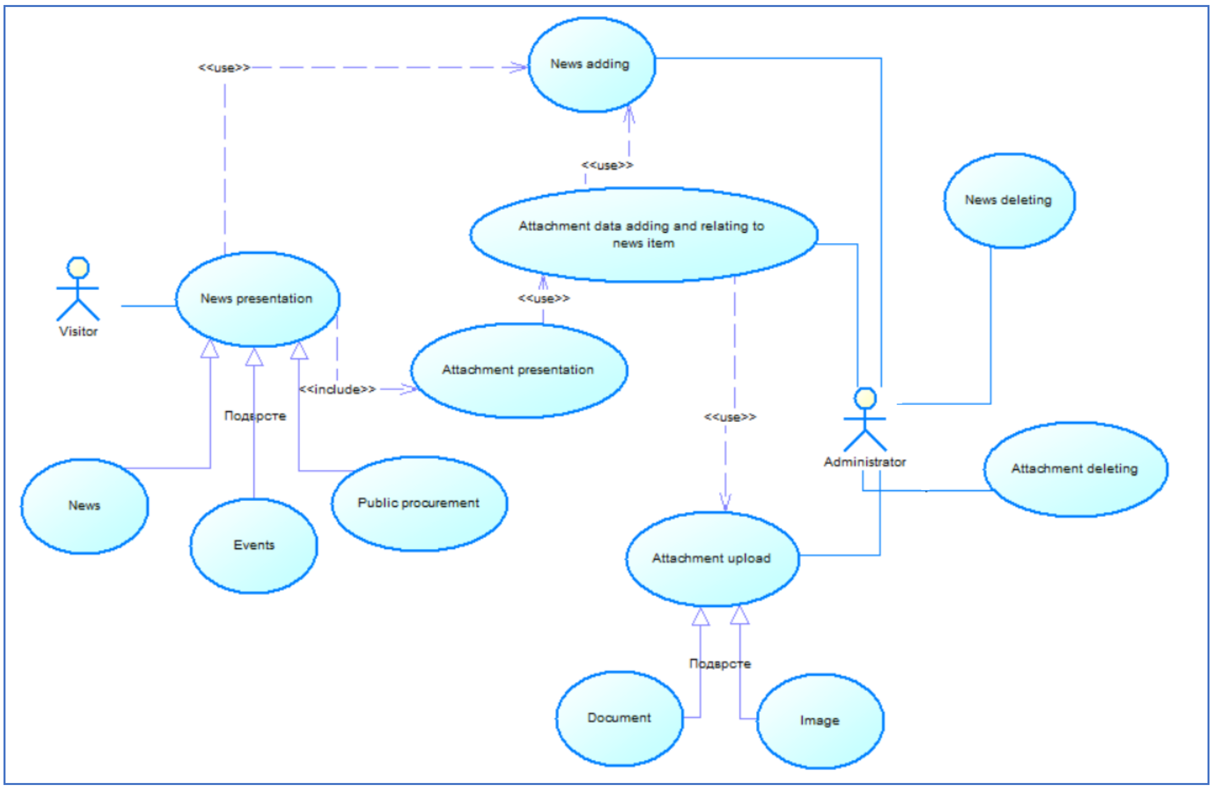
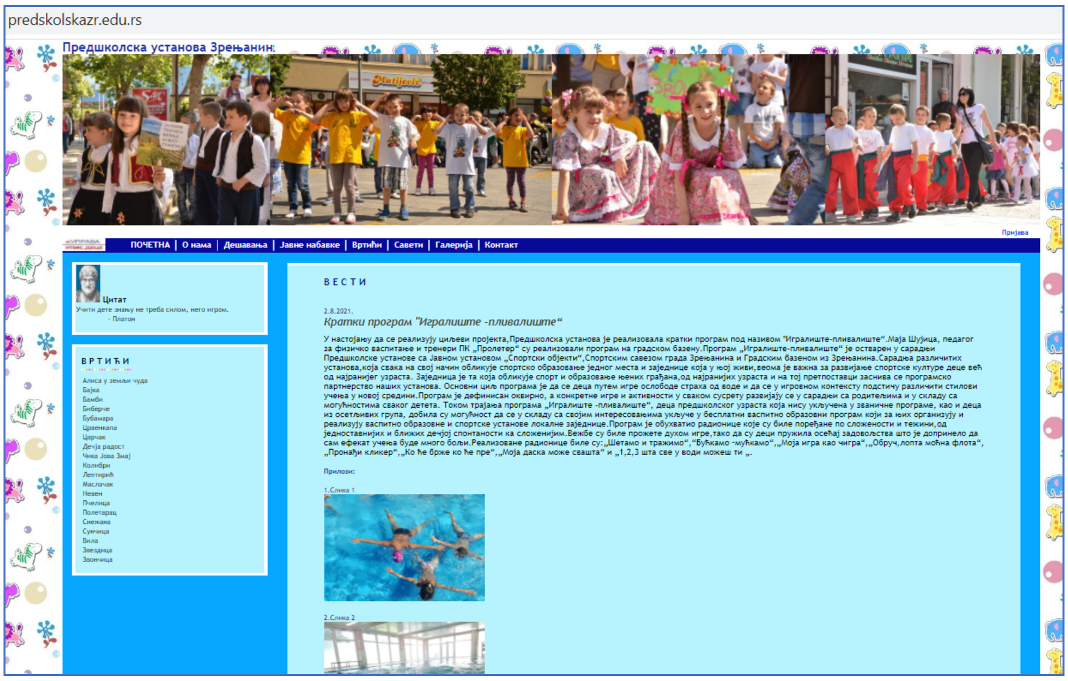


Figure 2. UML Use Case diagram with software functions of Zrenjanin Preschool Institution web portal [33]

Presentational part of web portal is presented at Figure 3, including news with images or documents as attachment to the main text of the news.


predskolskazr.edu.rs

Предшколска установа Зрењанин

Пријава

ПОЧЕТНА | О нама | Дешавања | Јавне набавке | Вртићи | Савети | Галерија | Контакт

Цитат
Учити дете знању не треба силом, него игром.
- Платон

ВРТИЋИ

Алиса у земљи чуда
Бајка
Бамби
Биберче
Бубамара
Црвенкапа
Цврчак
Дечја радост
Чика Јова Змај
Колибри
Лептирић
Маслачак
Невен
Пчелица
Полетарац
Снежана
Сунчица
Вила
Звездица
Звончица

ВЕСТИ

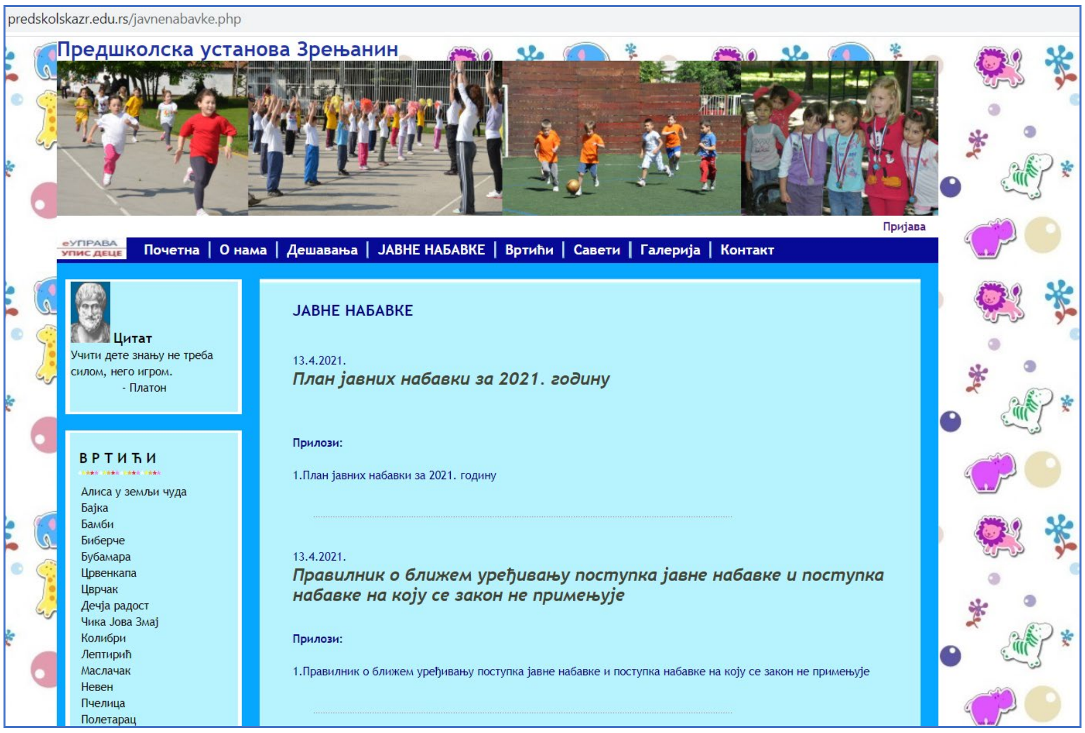
2.8.2021.
Кратки програм "Игралиште -пливалиште"

У настојању да се реализују циљеви пројекта,Предшколска установа је реализовала кратки програм под називом "Игралиште-пливалиште".Маја Шујица, педагог за физичко васпитање и тренери ПК „Пролетер“ су реализовали програм на градском базену.Програм „Игралиште-пливалиште“ је остварен у сарадњи Предшколске установе са Јавном установом „Спортски објекти“,Спортским савезом града Зрењанина и Градским базеном из Зрењанина.Сарадња различитих установа,која свака на свој начин обликује спортско образовање једног места и заједнице која у њој живи,веома је важна за развијање спортске културе деце већ од најранијег узраста. Заједница је та која обликује спорт и образовање њених грађана,од најранијих узраста и на тој претпоставци заснива се програмско партнерство наших установа. Основни циљ програма је да се деца путем игре ослободе страха од воде и да се у игровном контексту подстичу различити стилови учења у новој средини.Програм је дефинисан оквирно, а конкретне игре и активности у сваком сусрету развијају се у сарадњи са родитељима и у складу са могућностима сваког детета. Током трајања програма „Игралиште -пливалиште“, деца предшколског узраста која нису укључена у званичне програме, као и деца из осетљивих група, добила су могућност да се у складу са својим интересовањима укључе у бесплатни васпитно образовни програм који за њих организују и реализују васпитно образовне и спортске установе локалне заједнице.Програм је обухватио радионице које су биле поређане по сложености и тежини,од једноставнијих и ближих дечјој спонтаности ка сложенијим.Вежбе су биле прожете духом игре,тако да су деци пружила осећај задовољства што је допринело да сам ефекат учења буде много бољи.Реализоване радионице биле су:„Шетамо и тражимо“,“Буђкамо -мућкамо“,„Моја игра као чигра“,„Обруч,лопта моћна флота“, „Пронађи кликер“,„Ко ће брже ко ће пре“,„Моја даска може свашта“ и „1,2,3 шта све у води можеш ти „.

Прилози:

1.Слика 1

2.Слика 2


predskolskazr.edu.rs/javnenabavke.php

Предшколска установа Зрењанин

Пријава

Почетна | О нама | Дешавања | ЈАВНЕ НАБАВКЕ | Вртићи | Савети | Галерија | Контакт

Цитат
Учити дете знању не треба силом, него игром.
- Платон

ВРТИЋИ

Алиса у земљи чуда
Бајка
Бамби
Биберче
Бубамара
Црвенкапа
Цврчак
Дечја радост
Чика Јова Змај
Колибри
Лептирић
Маслачак
Невен
Пчелица
Полетарац

ЈАВНЕ НАБАВКЕ

13.4.2021.
План јавних набавки за 2021. годину

Прилози:

1.План јавних набавки за 2021. годину

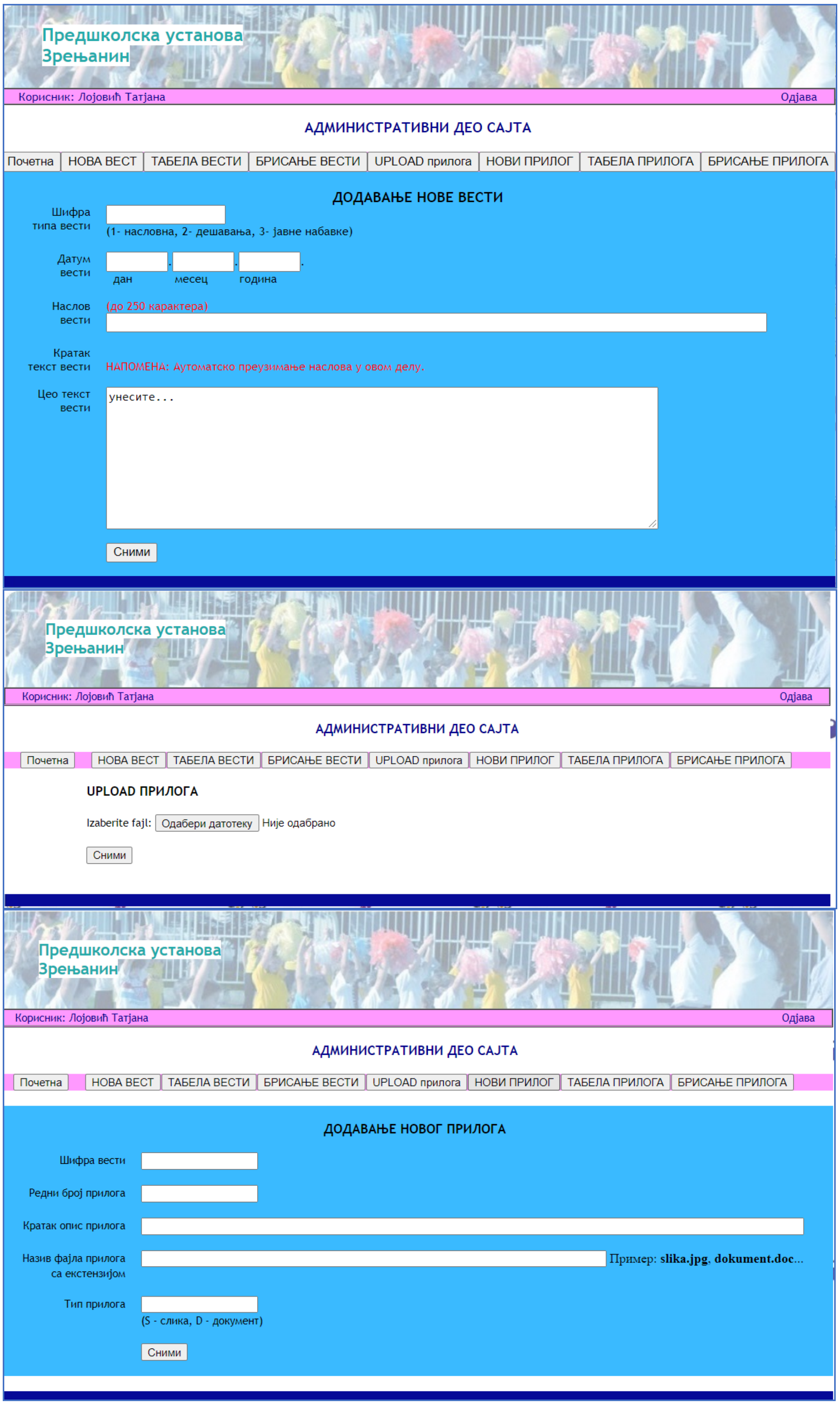
13.4.2021.
Правилник о ближем уређивању поступка јавне набавке и поступка набавке на коју се закон не примењује

Прилози:

1.Правилник о ближем уређивању поступка јавне набавке и поступка набавке на коју се закон не примењује

Figure 3. Presentational part of Zrenjanin Preschool web portal with news, images and documents attachments [33]

Administrative part of web portal is presented at Figure 4, with an example of web form for news data entry and recording, as well as attachment upload and web form for data entry about additional data about the attachment.



ДОДАВАЊЕ НОВЕ ВЕСТИ

Шифра типа вести (1- насловна, 2- дешавања, 3- јавне набавке)

Датум вести . . дан месец година

Наслов вести (до 250 карактера)

Кратак текст вести НАПОМЕНА: Аутоматско преузимање наслова у овом делу.

Цео текст вести унесите...

UPLOAD ПРИЛОГА

Izaberite fajl: Одабери датотеку Није одабрано

ДОДАВАЊЕ НОВОГ ПРИЛОГА

Шифра вести

Редни број прилога

Кратак опис прилога

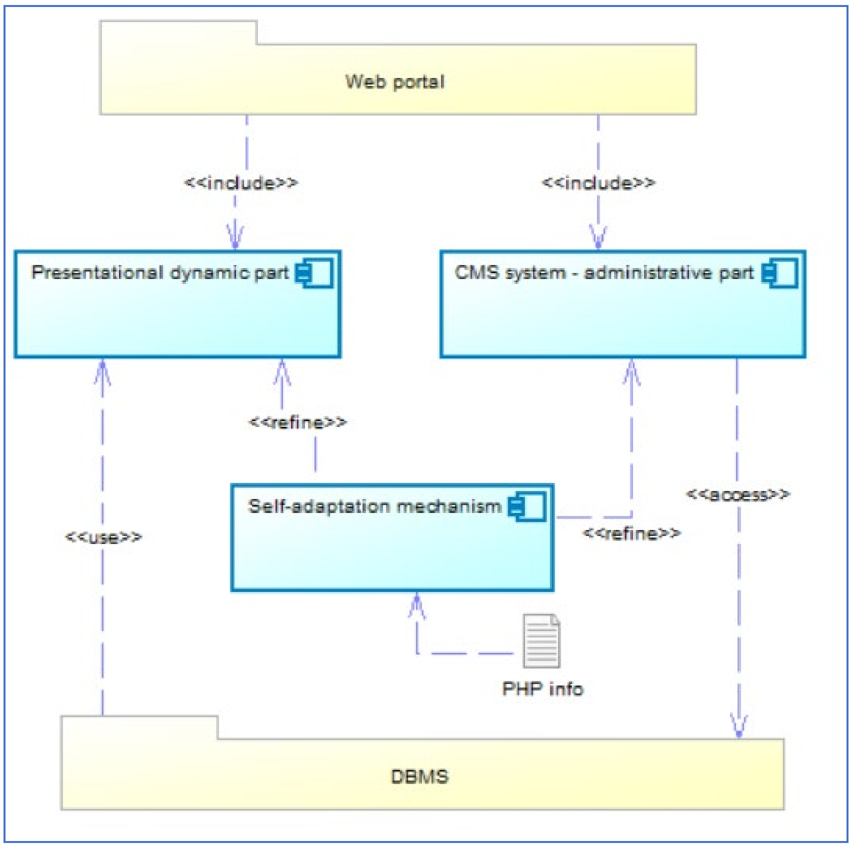
Назив фајла прилога са екстензијом Пример: **slika.jpg**, **dokument.doc**...

Тип прилога (S - слика, D - документ)

Figure 4. Essential pages of CMS part of Zrenjanin Preschool Institution web portal [33]

5. Refactoring and Proposed Hosting platform-related Self-adaptive Mechanism

The structure of web portal [33] is presented also with UML component diagram at Figure 5.



Web portal
<<include>>
<<include>>
Presentational dynamic part
CMS system - administrative part
<<refine>>
Self-adaptation mechanism
<<refine>>
<<access>>
<<use>>
PHP info
DBMS

Figure 5. UML component diagram of Zrenjanin Preschool Institution Web Portal [33]

The problem that this solution addressed was uncertainty of the exact time of shifting the hosting platform of the portal [33] from PHP5 to PHP7 support. Therefore, there should be a self-adaptation mechanism based on "sensors-effectors" approach, that could provide smooth transition from previous to upgraded hosting platform, without affecting the availability and functionality of the portal. The portal should have 24/7 availability.

To enable web application to have the self-adaptation mechanism included, it was necessary to refactor the code to provide centralization of the working environment sensing (i.e. to have this sensing code at a single place). Figure 6. presents the refactored object-oriented solution with class diagram (without attributes and methods, to be more readable), i.e. key classes hierarchy:

- At the top of hierarchy there is a class Connection ("Konekcija"), which has the role of SENSOR (detecting platform type with phpversion command and labeling the type of hosting for further use by the same and other classes) and DECISION/EFFECTOR (taling the platform label and selecting appropriate php command for connecting to the MYSQL database).
- Below the connection class there is class Table ("Tabela"), which is written as universal class (without any semantics). This class behaves as DECISION/EFFECTOR, since it receives platform label, selects and applies appropriate type of PHP command according to detected platform type. The role of this class is to perform all CRUD (Create, Read, Update, Delete) operations.

- At the lowest place in hierarchy there are semantically-rich classes that inherit class "Tabela". By inheritance, they receive technology part, but they use methods from Tabela by inserting semantics, i.e. SQL queries according to appropriate database table they refer to.

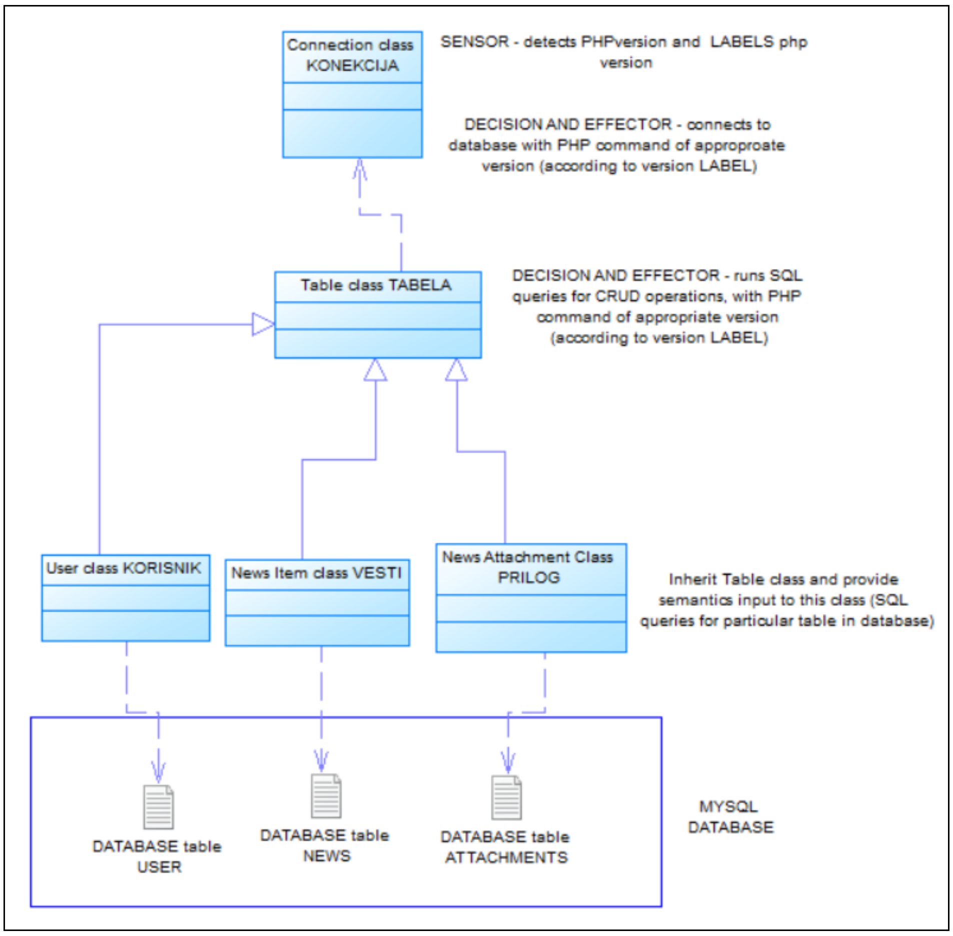
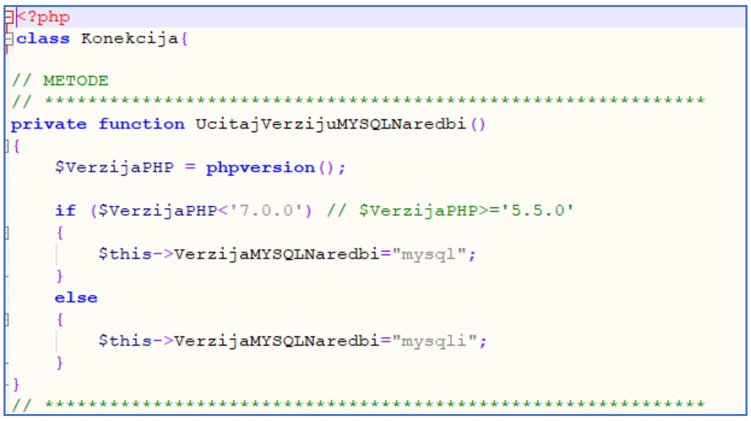

Connection class KONEKCIJA
SENSOR - detects PHPversion and LABELS php version
DECISION AND EFFECTOR - connects to database with PHP command of approproate version (according to version LABEL)
Table class TABELA
DECISION AND EFFECTOR - runs SQL queries for CRUD operations, with PHP command of appropriate version (according to version LABEL)
User class KORISNIK
News Item class VESTI
News Attachment Class PRILOG
Inherit Table class and provide semantics input to this class (SQL queries for particular table in database)
DATABASE table USER
DATABASE table NEWS
DATABASE table ATTACHMENTS
MYSQL DATABASE

Figure 6. Class diagram of the refactored solution

The hierarchy of classes at Figure 6 enables application of "sensor-effector approach" at self-adaptive mechanism, but also the distinction of technology and semantics enables better support to any future maintenance needs.

The proposed solution of the self-adaptation mechanism within web portal [33] is presented with parts and appropriate PHP code listings, to elaborate the approach:

- Sensor part – detecting PHP version (within the Connection class)



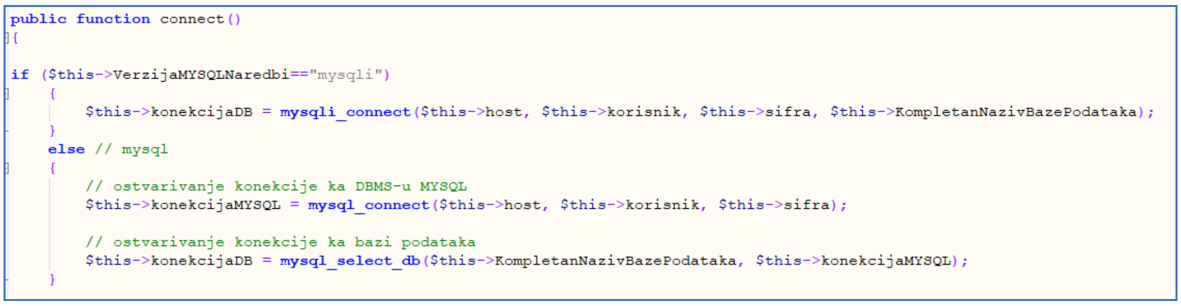
```php
<?php
class Konekcija{

// METODE
// ***************************************************************
private function UcitajVerzijuMYSQLNaredbi()
{
    $VerzijaPHP = phpversion();

    if ($VerzijaPHP<'7.0.0') // $VerzijaPHP>='5.5.0'
    {
        $this->VerzijaMYSQLNaredbi="mysql";
    }
    else
    {
        $this->VerzijaMYSQLNaredbi="mysqli";
    }
}
// ***************************************************************
```

Listing 1. The sensor part of self-adaptation mechanism

- Decision part – deciding which type of commands to use (PHP5 or PHP7)
- Effector part – apply the selected type of commands to regular work of data entry and data presentation

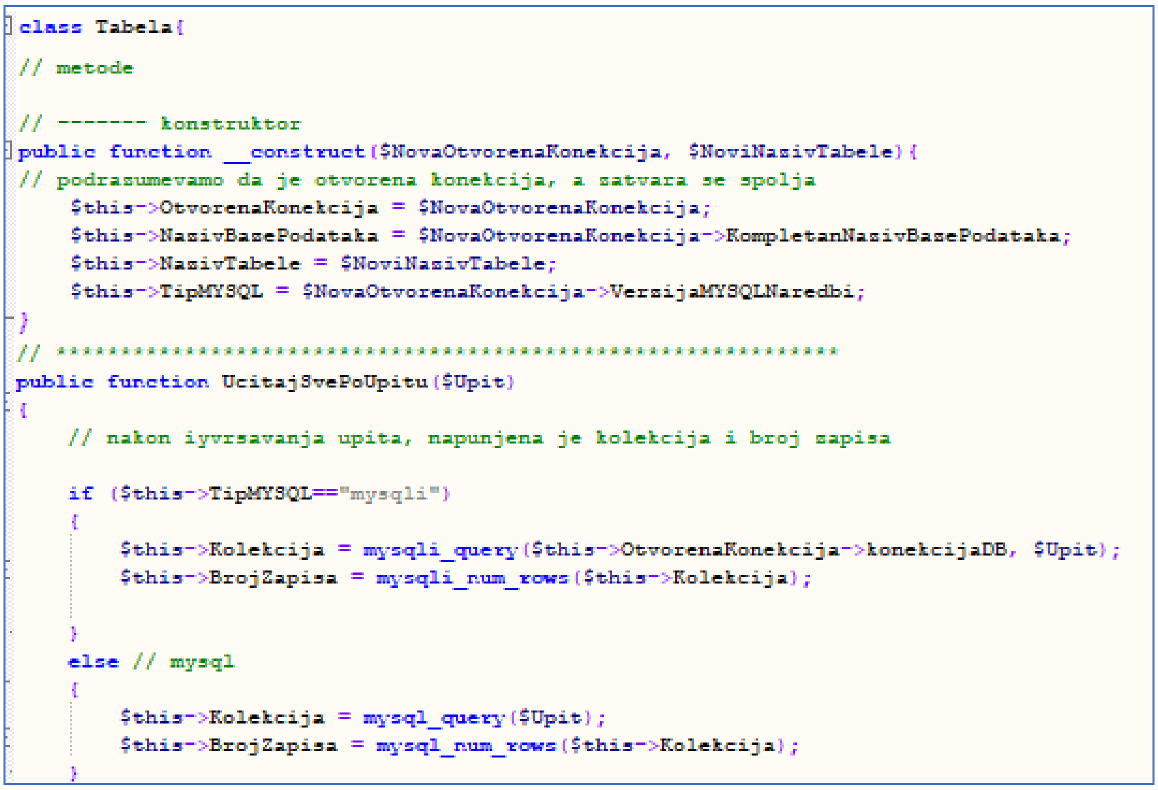


```php
public function connect()
{

if ($this->VerzijaMYSQLNaredbi=="mysqli")
    {
        $this->konekcijaDB = mysqli_connect($this->host, $this->korisnik, $this->sifra, $this->KompletanNazivBazePodataka);
    }
    else // mysql
    {
        // ostvarivanje konekcije ka DBMS-u MYSQL
        $this->konekcijaMYSQL = mysql_connect($this->host, $this->korisnik, $this->sifra);

        // ostvarivanje konekcije ka bazi podataka
        $this->konekcijaDB = mysql_select_db($this->KompletanNazivBazePodataka, $this->konekcijaMYSQL);
    }
```

Listing 2. The decision and effector part of self-adaptation mechanism for connecting to a database



```php
class Tabela{
// metode

// ------- konstruktor
public function __construct($NovaOtvorenaKonekcija, $NoviNazivTabele){
// podrazumevamo da je otvorena konekcija, a zatvara se spolja
    $this->OtvorenaKonekcija = $NovaOtvorenaKonekcija;
    $this->NazivBazePodataka = $NovaOtvorenaKonekcija->KompletanNazivBazePodataka;
    $this->NazivTabele = $NoviNazivTabele;
    $this->TipMYSQL = $NovaOtvorenaKonekcija->VerzijaMYSQLNaredbi;
}
// ****************************************************************
public function UcitajSvePoUpitu($Upit)
{
    // nakon iyvrsavanja upita, napunjena je kolekcija i broj zapisa

    if ($this->TipMYSQL=="mysqli")
    {
        $this->Kolekcija = mysqli_query($this->OtvorenaKonekcija->konekcijaDB, $Upit);
        $this->BrojZapisa = mysqli_num_rows($this->Kolekcija);

    }
    else // mysql
    {
        $this->Kolekcija = mysql_query($Upit);
        $this->BrojZapisa = mysql_num_rows($this->Kolekcija);
    }
```

Listing 3. The decision and effector part of self-adaptation mechanism for running queries upon database

6. Conclusions and Future Work

The research problem of this paper is related to self-adaptivity of web applications, regarding hosting platform technology change. This problem is particularly important during software maintenance, when shifting from one to another technology support, i.e. upgrading the hosting technology.

This paper presented the theoretical background related to the concepts of self-adaptive software, as well as related work with similar research results. It could be concluded that there are many research results related to adaptation of software to a user, as well as results in diversity of software adaptation techniques, methods and approaches. It is presented that there are also significant research results in the area of adaptive web application. However, there are not many previous research results related to changes of technology at web application hosting and approaches how to solve the problem of smooth transition between previous and upgraded hosting platform. In this context, this paper contributes with addressing these issues and proposes a simple solution based on “sensor-effector” approach.

Future research could be conducted to explore the impact of technology change frequency to software change requirements by classification and measurements of change requests according to the roots (technology, business needs, user requirements and needs). Other research could be related to approaches and techniques of automating software reengineering and creating self-adaptive mechanism, based on introducing new programming languages or their upgraded versions.

7. Acknowledgements

This research has been conducted within two projects:

1) “Platform adaptation of existing PHP web application as a administration system of official web site of Preschool institution, Zrenjanin (www.predskolskazr.edu.rs) and migration to new hosting platform“, financed by Preschool Institution Zrenjanin,
2) “Development of software tools for analysis and improvements of business processes”, project No: TR32044, financed by Ministry of science, education and technology development of Republic of Serbia.

8. References

[1] ISO/IEC 14764:2006 Software Engineering — Software Life Cycle Processes — Maintenance https://www.iso.org/standard/39064.html

[2] Manny M. Lehman,. “Laws of software evolution revisited”. Software process technology. Springer Berlin Heidelberg, (1996): 108-124.

[3] Manifesto for agile software development, https://agilemanifesto.org/ [visited: September 14, 2021.]

[4] Nasrin Ghasempour Maleki, Raman Ramsin: “Agile Web Development Methodologies: A survey and Evaluation”, In: R. Lee (ed.) Software Engineering Research, Management and Applications, Studies in Computational Intelligence 722 (2018)

[5] Samir Omanovic, Emir Buza. “Importance of Future Change Requests Frequency Analysis Factor for Solution Selection and Software Maintenance”, In: Damir Boras, Nives Mikelic Preradovic, Francisco Moya, Mohamed Roushdy, Abdel-Badeeh M. Salem (ed.) Recent Advances in Information Science, WSEAS (2013): 246-251.

[6] Samir Omanovic, Emir Buza. “Future Change Requests in Agile Software Engineering”. International Journal of Computers, Issue 3, Volume 7 (2013): 91-98.

[7] Yguaratã Cerqueira Cavalcanti, Ivan do Carmo Machado, Paulo Anselmo da Motal S.Neto, Eduardo Santana de Almeida. “Towards semi-automated assignment of software change requests”. Journal of Systems and Software, Volume 115 (2016): 82-101.

[8] Evgeny Nikulchev, Dmitry Illin, Alexander Gusev. “Technology Stack Selection Model for Software Design of Digital Platforms”. Mathematics, MDPI, Special issue “Applied and Computational Mathematics for Digital Environments”, Vol 9, No 4, 308 (2021)

[9] Zhang J, Cheng B.H.C. „Model-Based Development of Dynamically Adaptive Software”. ICSE’06, Shanghai, China. (2006)

[10] McKinley P. K., Sadjadi S. M., Kasten E. P., Cheng B. H. C. “Composing adaptive software”. IEEE Computer, vol. 37, no. 7. (2004): 56–64.

[11] Salehie M, Tahvildari L. “Self-Adaptive Software: Landscape and Research Challenges”. ACM Transactions on Autonomous and Adaptive Systems. (2009)

[12] Magnaudet, M, Chatty S.. “What should adaptivity mean to interactive software programmers?” EICS 2014, ACM SIGCHI symposium on Engineering interactive computing systems, Rome, Italy. (2014): 13-22.

[13] Derakhshanmannesh, M, Ebert J, Amoui M, Tahvildari L. “Introducing Adaptivity to Achieve Longevity for Software”. Lecture Notes in Informatics, Software Engineering 2011 Workshop, Bonn. (2011). 59-70.

[14] Xiao, L, Greer D. “Software Adaptivity through XML-based business rules and agents”, 17th Int. Conference on Software Engineering and Knowledge Engineering (SEKE'05),Taipei, China, (2005). 62-67.

[15] Xiao, L, Greer D. “Adaptive Agent Model: Software Adaptivity using an Agent-oriented Model-Driven Architecture”. Information and Software Technology 51, (2009): 109-137.

[16] Manesh M, Ebert J. “Software Evolution Towards Model-Centric Runtime Adaptivity.” Proceedings of the Euromicro Conference on Software Maintenance and Reengineering, CSMR · (2011).

[17] Michael Hinz, Zoltan Fiala: “AMACONT – A System Architecture for Adaptive Multimedia Web Applications”, Published in Berliner XML Tage. (2004)

[18] Guillaume Gauvrit, Erwan Daubert, Françoise André. SAFDIS: „A Framework to Bring Self-Adaptability to Service-Based Distributed Applications”, 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), Sep 2010, Lille, France. (2010): 211–218.

[19] Wong S, Brandon A, Anjam F, Seedorf R, Giorgi R, Yu Z, Puzovic N, Mckee S. A, Sjalander M, Carro L, Keramidas G: “Early Results from ERA –Embedded Reconfigurable Architectures” IEEE International Conference on Industrial Informatics (INDIN). (2011).

[20] Faical Felhi, Jalel Akaichi: “A new approach towards the self-adaptability of service-oriented architectures to the context based n workflow”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 12, (2012).

[21] Eoin Martino Grua; Ivano Malavolta; Patricia Lago: “Self-Adaptation in Mobile Apps: a Systematic Literature Study”, IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). (2019).

[22] Guntram Graef, Martin Gaedke: “Self-Adaptive Web Applications”, WCML: an Enabling Technology for the Reuse in Object-oriented web engineering,8th International World Wide Web Conference (WWW8), Toronto, Ontario, Canada, Elsevier, 102-103, (1999).

[23] Lorena Castaneda Bueno, Hausi A. Muller, Norha Villegas: “Self-Adaptive Applications: On the development of Personalized Web-Tasking Systems”, ACM, 9th ICSE International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2014). (2014).

[24] Vasconcelos, L.G., Baldochi, L.A. and Santos, R.D.C. "An approach to support the construction of adaptive Web applications", International Journal of Web Information Systems, Vol. 16 No. 2, (2020): 171-199.

[25] Mona Fadavi, Raman Ramsin, "Methodologies for Model-Driven Development of Adaptive Web Applications: An Analytical Survey," Journal of Software vol. 11, no. 1.(2016): 94-109.

[26] Zoltan Fiala: „Design and Development of Component-based Adaptive Web Applications‟ PhD Thesis, Technischen Universitƒat Dresden Fakultat, (2007)

[27] Karzan Wakl, Dayang N.A. Jawawi: ‟A New Adaptive Model for Web Engineering Methods to Develop Modern Web Applications”, ICSIM2018: Proceedings of the 2018 International Conference on Software Engineering and Information Management, (2018): 32–39.

[28] Tao Jiang; Jing Ying; Minghui Wu; Canghong Jin: A method for model-driven development of adaptive web applications, IEEE 12th International Conference on Computer Supported Cooperative Work in Design, (2008).

[29] Florian Daniel, Maristella Matera, Alessandro Morandi, Matteo Mortari, Giuseppe Pozzi: “Active Rules for Runtime Adaptivity Management”, AEWSE. (2007).

[30] David Parsons, Sebastian Schroeder: Adaptive Information Systems for the Web 2.0: Developing a Mobile Learning Architecture”, UKAIS (2006).
Norha M. Villegas: “Context Management and Self-Adaptivity for Situation-Aware Smart Software Systems”, Докторска дисертација, University of Victoria. (2013).

[31] Ljubica Kazi, Zoltan Kazi, Tatjana Lojovic: „Preschool web portal development“, Proceedings of International conference on Applied internet and information technologies 2017, 5 Octobar 2017, Proceedings, ISBN 978-86-7672-304-1, (2017): 145-152.

[32] Ljubica Kazi, Dijana Karuović, Dragica Radosav, Tatjana Lojović, Aleksandra Kalezić-Vignjević and Olga Lakićević: Adaptivity of Web Applications – Case of Preschool Web Portal for Interinstitutional Data Integration and Analysis, TIE2020, Cacak, (2020): 171-177.
[33] Preschool Institution Zrenjanin official web portal, www.predskolskazr.edu.rs [visited: 14.9.2021.]