



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА

Факултет по изчислителна техника и автоматизация
Катедра "Компютърни науки и технологии"

маг. инж. Димитър Здравков Димитров

**ИЗСЛЕДВАНЕ, ПРОЕКТИРАНЕ И РЕАЛИЗАЦИЯ НА
НОВИ АРХИТЕКТУРИ НА СИСТЕМИ ЗА
СЪЗДАВАНЕ НА УЕБСАЙТОВЕ**

А В Т О Р Е Ф Е Р А Т

**на дисертация за присъждане на образователна и
научна степен "доктор"**

**по докторска програма „Системно Програмиране“ към
професионално направление „Компютърна и
Комуникационна Техника“**

Научен ръководител: доц. д-р инж. Елена Рачева

Рецензенти:

- 1. проф. Стойчо Стойчев**
- 2. доц. Христо Вълчанов**

**Варна
2018 г.**

**Дисертационният труд е обсъден на 07.11.2018 г. в катедра
“Компютърни Науки и Технологии” и насочен за защита.**

Автор: Димитър Здравков Димитров

**Заглавие: Изследване, проектиране и реализация на нови
архитектури на системи за създаване на уебсайтове**



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА

Факултет по изчислителна техника и автоматизация
Катедра "Компютърни науки и технологии"

маг. инж. Димитър Здравков Димитров

**ИЗСЛЕДВАНЕ, ПРОЕКТИРАНЕ И РЕАЛИЗАЦИЯ НА НОВИ
АРХИТЕКТУРИ НА СИСТЕМИ ЗА СЪЗДАВАНЕ НА
УЕБСАЙТОВЕ**

А В Т О Р Е Ф Е Р А Т

**на дисертация за присъждане на образователна и
научна степен "ДОКТОР"**

Варна 2018 г.

Дисертационният труд съдържа 153 страници, включително 50 фигури, 1 таблица и 8 приложения, оформени във въведение, 4 глави, заключение, общи изводи, научно-приложни приноси, списък с фигурите, списък с таблиците, списък с използваните термини и съкращения, списък с публикациите на автора, раздел приложения и списък с използваната литература от 191 заглавия, всички на латиница.

Защитата на дисертационния труд ще се състои на 2018 г. от ч. в стая на открито заседание на жури, сформирано със заповед на Ректора №. /2018г.

Материалите по защитата (дисертация, рецензии и становища) са на разположение на интересуващите се във ФД "Докторанти", стая 318НУК.

А. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

Актуалност на проблема

Настоящият дисертационен труд се фокусира върху онлайн системите за създаване на уебсайтове (ССУ) и, в частност, онлайн средствата за визуална редакция на съдържание и онлайн публикуване.

Статистическите данни на интернет потреблението показват среден ръст от около 50% на годишна база, а ръстът на интернет проникването се увеличава средно с 2.5% от световното население, на годишна база. Това го прави най-бързо развиващият се пазар, особено като тази тенденция на ръст е без промяна през последните 2 десетилетия.

С разрастването на интернет потреблението се разраства и нуждата от постоянно създаване на нови уебсайтове. Освен множеството дигитални агенции, които са се захванали с това призвание, се оформя изцяло нова ниша, това е нишата на ССУ. Това са системи, позволяващи автоматизацията на процеса на създаване на уебсайт, включително и неговото онлайн публикуване.

Първите ССУ са разработени още с първите стъпки на проникване на интернет, като представляват офлайн приложения. Подробно класифициране на различните ССУ е направено в Глава I. Настоящият дисертационен труд е фокусиран върху класовете на съвременните онлайн ССУ.

Ръстът на броя ССУ е тясно свързан с нарастващото проникване на интернет достъпа и използваемостта му, измерена в нарастващия обем на глобалния трафик. В изследването на ССУ, всяка от първите 10 системи, по брой потребители, има по над 40 млн. създадени уебсайта, като лидерите имат по над 100 млн. уебсайта.

Въпреки огромния брой уебсайтове, създадени и поддържани чрез онлайн ССУ, пазарно проучване на Шведския пазар, извършено за целите на дисертационния труд, показва че пазарната ниша е все още в своето начало и тепърва ще се наблюдава бум на броя на създадените ССУ, както и бум на потреблението им. Резултатите от проучването са приложени към дисертационния труд.

Към момента на защита на настоящия дисертационен труд не съществува цялостно описание на архитектура за ССУ, няма публикувани разработки, проучвания или статии, описващи как функционира една ССУ, какви са нужните подсистеми за оптимизиране на работата на една ССУ, която на практика трябва да поддържа стотици милиони уебсайтове, позволяваща паралелна редакция на живо на милиони потребители в един и същ момент, да позволява бързо пускане на нови версии, връщане към стари версии с нужното живо мигриране на данни, паралелна работа на няколко версии на една и съща функционалност и др. Единствените публикувани разработки са под формата на няколко патента от най-големите корпорации в бранша, които разкриват остарели технологии и методи.

Цел и задачи на дисертационния труд

Целта на дисертацията е да се изследва, проектира и докаже работоспособността на модел на архитектура за ССУ, която е предназначена за масовия потребител. Това са клас ССУ, характеризиращи се с нестандартно високи натоварвания и изискват една от най-сложните структури от подсистеми, работещи в синхрон за оптимално и безпроблемно опериране на ССУ с високо ниво на сигурност, и работоспособност.

За постигане на целта бяха поставени следните задачи:

1. Да се изследват тенденциите на растеж на интернет потребителите и на интернет потреблението за прогнозиране на растежа на нуждите от ССУ.
2. Да се изследват съществуващи ССУ, с фокус върху водещите ССУ.
3. Да се класифицират ССУ по основни критерии.
4. Да се изследват и анализират възможните проблеми при проектиране, създаване, мащабиране и опериране на ССУ.
5. Да се проучат и анализират публични изследвания относно технологии за ССУ и технологии, насочени към подсистеми на ССУ.
6. Да се проектират концептуални модели на ССУ архитектури и реализират експериментални ССУ, базирани на резултатите от проведените изследвания и на конкретни хипотези за решаване на основни проблеми при опериране на ССУ.

7. Да се създадат методи за конвертиране на съществуваща софтуерна система от еднонаемателна към многонаемателна архитектура.

8. Да се анализират проблемите при проектирането на експерименталните ССУ и да се отстранят, чрез въвеждане на допълнително технологична сложност в архитектурата.

9. Проектиране на концептуален модел за архитектура на ССУ, предназначена за масовия потребител, гарантираща стабилно опериране при високи натоварвания с възможност за безпроблемно паралелно опериране на стотици милиони уебсайтове.

10. Реализация на ССУ на базата на проектираната архитектура, оптимизация и отстраняване на проблемни точки по концепцията на предложената архитектура.

Обекти на изследване са онлайн системите за създаване на уебсайтове, в частност - системните им архитектури и методите на проектирането и разработването им.

Предмет на изследване е изследването, проектирането и експерименталната реализация на архитектура на ССУ за масовия потребител, хвърляща светлина върху технологичната сложност при създаване на ССУ, включваща в композицията си множество подсистеми.

Методи за изследване

За изпълнение за поставените цели и задачи за изследване са използвани следните методи:

- Теоретични изследвания на основата емпирични данни
- Експериментални изследвания в производствени условия
- Съчетаване на теоретични и експериментални изследвания

Научна новост

Основната за дисертационния труд научна новост е предложението на непубликуван до момента, цялостен концептуален модел на архитектура на ССУ, включвайки множество прилежащи подсистеми, който да положи публична основа за развитието на изследванията в сферата. Като основен проблем в развитието на новосъздадени ССУ е липсата на публичност при развитие на технологиите в областта. В раздел „Приноси“ са описани и останалите аспекти, които може да се считат за научна новост. Създадени са различни методи, върху които да се базират основни подсистеми на ССУ.

Практическа ценност и приложимост

Научно-приложната новост се състои в емпиричните резултати и общи изводи, в следствие на експерименталните изследвания, както и теорията и знанията, развити за нуждите на създаването на една ССУ. Резултатите от дисертацията би съкратила процеса за производство на една ССУ с години, нужни за развиване на необходимата теория, в следствие на което би спестила сериозния финансов ресурс, нужен за тези изследвания.

Създадени са методи за конвертиране на съществуваща софтуерна система от еднонаемателна към многонаемателна архитектура, които може да се използват за привеждане на една по-остаряла софтуерна архитектура към модел на доставка SaaS.

Основните аспекти на практическите ценности в дисертационния труд са описани подробно в секция „общи изводи“ и, от части - в секция „Приноси“.

Реализация и апробация на резултатите

Всички изследвания и резултати от настоящия дисертационен труд са реализирани в производствени системи, надлежно финансирани от български и чуждестранни компании.

Разгледаните концептуални модели на архитектури и описаните експериментални реализации от Глава II са реализирани в 4 отделни ССУ от различни класове. Всички са реализирани чрез нужното финансиране от отделни компании за доказване на зададени хипотези за проектиране на по-сложния концептуален модел на архитектура на ССУ за масовия потребител.

Основната за настоящия дисертационен труд архитектура, описана в Глава III и експерименталната ѝ реализация, разгледана в Глава IV, са финансирани с над 1 млн. евро от чуждестранни инвеститори. Реализираната система е с безплатен достъп на домейн oxxu.com.

Публикации

Основните резултати на дисертационния труд са представени в 9 публикации в България и чужбина, 7 от които - самостоятелни, 1 в съавторство с научния ръководител, 1 с двама съавтори.

Структура и обем на дисертацията

Дисертационният труд съдържа 153 страници, включително 50 фигури, 1 таблица и 8 приложения, оформени във въведение, 4 глави, заключение, общи изводи, научно-приложни приноси, списък с фигурите, списък с таблиците, списък с използваните термини и съкращения, списък с публикациите на автора, раздел приложения, и списък с използваната литература от 191 заглавия, всички на латиница.

Глава I: Изследвани са съществуващите ССУ, наличната публична информация, свързана с научни изследвания в областта, технологиите и системната им архитектура. Синтезирана е класификация на видовете ССУ по основни критерии. Дефинирани са обектът на изследване, предметът на изследване, целите и задачите на дисертацията. Във въведението и в Глава I се решават задачи 1 до 5.

Глава II: Предложени, проектирани и експериментално реализирани са архитектури, базирани на конкретни концептуални модели. Обобщени са подходи за създаване на различни видове архитектури за ССУ. Целта на тази глава е да изясни проблемите, които една ССУ трябва да решава, според нейната цел. Обобщени са резултатите, приложимостта и внедряването на първичните изследвания над ССУ. Оценени са силните и слабите страни на различните подходи за изграждане на ССУ архитектура, според определената класификация. В тази глава се решават задачи 6 до 8.

Глава III: Предложен е концептуален модел на архитектура на онлайн ССУ за масовия потребител, достигната след повече от 4 години предварителни изследвания, експерименти и разработки по видове архитектури от Глава II, и отделно още 4 години експериментални разработки по конкретно предложената архитектура на ССУ. Един период на подобрения и отстраняване на слабости в архитектурата след установяване на потенциални приложни проблеми. Главата разрешава основна задача номер 9 на дисертационния труд.

Глава IV: Разгледана е експерименталната реализация на предложената в Глава III архитектура, като са обобщени резултатите и срещнатите проблеми в процеса на проектиране. Описан е процесът на проектиране на ССУ от теоретична хипотеза и научни разработки до практическа реализация. Демонстрирани са различните части на имплементацията с визуализации, като самата система към този момент е достъпна безплатно онлайн, с десетки хиляди потребители създали своите уебсайтове. Главата разрешава задача 10 от дисертационния труд.

Б. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

ВЪВЕДЕНИЕ

Фокус на дисертационния труд

Онлайн системите за създаване на уебсайтове (ССУ), в частност онлайн средствата за визуална редакция на съдържание и онлайн публикуване, са фокуса на настоящия дисертационен труд.

Системи за създаване на уебсайтове (ССУ)

Първите ССУ са разработени още с първите стъпки на проникване на интернет, като са били офлайн приложения. Настоящият дисертационен труд е фокусиран върху съвременните онлайн ССУ. Ръстът на броя ССУ е тясно свързан с нарастващото проникване на интернет достъпа и използваемостта му. В изследването на ССУ, всяка от първите 10 системи, по брой потребители, има по над 40 млн. създадени уебсайта, като лидерите имат по над 100 млн. уебсайта.

Въпреки огромния брой уебсайтове, създадени и поддържани чрез онлайн ССУ, пазарно проучване, извършено за целите на дисертационния труд, показва, че пазарната ниша е все още в своето начало и тепърва ще се наблюдава бум на броя на създадените ССУ, както и бум на потреблението им. Резултатите от проучването са приложени към дисертационния труд.

Развитие на уеб браузърите

Новата ера на ССУ може да се зададе около 2012 г., с навлизането на поддръжката на стандартите за HTML5 и CSS3, поддръжка на стандартизираните на Javascript чрез приемане и въвеждане на последните стандарти за ECMAScript (ES) с фундаментални промени, приети 2009 г. с ES5, като към момента на писане на дисертацията е приета версия 8 на стандарта

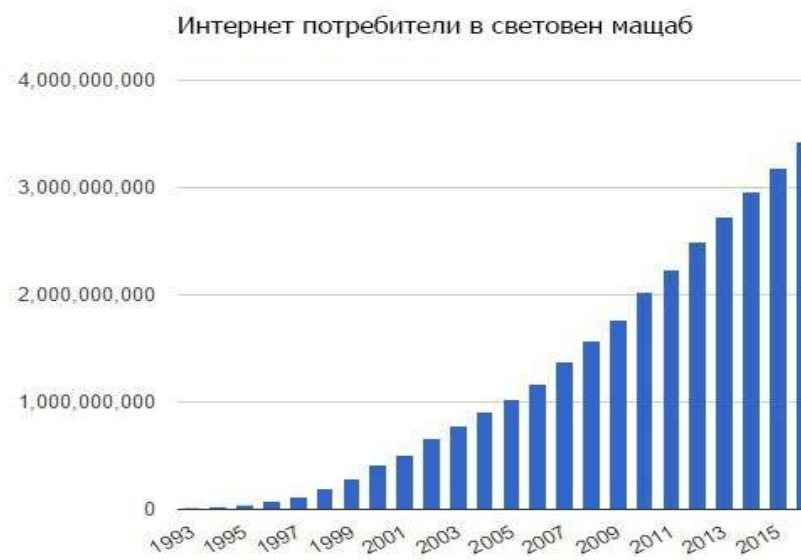
ES2017, като все още ES5 е последната най-поддържана версия. Възможността за създаване на иновативни ССУ подсигурава и драстичното подобряване на производителността на браузърите при рендериране и изпълнение на код, десетки стандарти за API, приети и масово поддържани от браузърите (WebRTC, WebGL, WebVR и мн. др.).

Развитие на интернет и съпътстващите технологии

В статистическите данни от изследвания върху интернет, показани на Фигура 1 се твърди, че към юни 2016 г. приблизителният брой интернет потребители е бил 3.425 млрд. (46.1% от световното население), удвоен от 2009-та година, когато е бил 1.766 млрд., докато през декември 1995 г. са били само 16 млн. по данни на IDC. Последното измерване от декември 2017 г. е 4.157 млрд., представляващо (54.4% от световното население), което е повишение с над 8% от населението само за година.

Глобален интернет трафик по години

Година	IP Трафик (PB/месец)	Фиксиран интернет трафик (PB/месец)	Мобилен интернет трафик (PB/месец)
1990	0.001	0.001	n/a
1991	0.002	0.002	n/a
1992	0.005	0.004	n/a
1993	0.01	0.01	n/a
1994	0.02	0.02	n/a
1995	0.18	0.17	n/a
1996	1.9	1.8	n/a
1997	5.4	5.0	n/a
1998	12	11	n/a
1999	28	26	n/a
2000	84	75	n/a
2001	197	175	n/a
2002	405	356	n/a
2003	784	681	n/a
2004	1,477	1,267	n/a
2005	2,426	2,055	0.9
2006	3,992	3,339	4
2007	6,430	5,219	15
2008	10,174	8,140	33
2009	14,686	10,942	91
2010	20,151	14,955	237
2011	30,734	23,288	597
2012	43,570	31,339	885
2013	51,168	34,952	1,480
2014	59,848	39,909	2,514
2015	72,521	49,494	3,685



Фигура 1. Статистика на броя на интернет потребителите

Таблица 1. Статистика на глобалния интернет трафик

Растеж на интернет проникването и потреблението

Обменът на информация през 1993 г. е бил 1% от общия обмен, през 2000 г. е бил 51%, а през 2007 г. е бил над 97%. Статистическите данни от Таблица 1 за интернет трафика, базирани на изследвания на Cisco Systems, показват глобалния трафик, растящ от 1TB (Terabytes) на месец през 1990 г. до 72.5EB (Exabytes) на месец през 2015 г. Предвижданията са общият IP трафик да се увеличи неколкостратно от 2016 г. до 2021 г. до 278EB на месец.

1. Анализ на растежа:

Използвайки линейна регресия на базата на времеви серии, чрез метода на най-малките квадрати, е създаден статистически модел, чрез който да се моделира тенденцията на растежа на интернет потребителите.

Ако приемем, че t е годината на измерване, $U(t)$ са потребителите за годината, $\Delta U(t)$ е промяната на размера на интернет потребителите на годишна база, R^2 е изравненият квадрат, показващ точността на модела.



Фигура 2. Статистически модел на размера на повишението на интернет потребителите

Уравнението, което представя модела, е:

$$\Delta U(t) = U(t) - U(t-1) = 15.623 * t - 31160 \quad (1.1)$$

$$R^2 = 0.8208 \quad (1.2)$$

За анализ на размера на повишението на интернет трафика (в Pb/месец) бяха използвани два подхода за създаване на статистически модел, които имат близка степен на обяснение. Първият е чрез експоненциална, а вторият чрез полиномиална регресия от трети ред.



Фигура 3. Статистически модел на размера на повишението на интернет потреблението, чрез метода на експоненциална регресия



Фигура 4. Статистически модел на размера на повишението на интернет потреблението в, чрез метода на полиномиална регресия

Ако приемем, че $I(t)$ е измереният интернет IP трафик в Pb на месец, годишното изменение на трафика е $\Delta I(t)$, то уравнението, с което може да се представи моделът, чрез експоненциална регресия, е:

$$\Delta I(t) = I(t) - I(t-1) = 1E - 17e^{0.2024t} \quad (1.3)$$

$$R^2 = 0.8212 \quad (1.4)$$

При полиномиална регресия от трети ред, уравнението представящо модела е:

$$\Delta I(t) = 2.27545t^3 - 16501t^2 + 3E + 07t - 2E + 10 \quad (1.5)$$

$$R^2 = 0.8907 \quad (1.6)$$

Експоненциалният модел е с по-малка точност и при него нараства грешката експоненциално. Подходящ е за представяне на тенденции в близките 5 години, но при по-дълъг период дава голямо отклонение. Полиномиалният модел от друга страна, показва по-голяма точност при проведени експерименти с вече известни данни и се представя по-стабилно при представяне на предсказания за по-дълъг период от време, например от 10г.

Моделите са полезни за анализиране на бъдещите тенденции на разрастване на проникването и потреблението, които пряко се отразяват към повишението за нуждите от системи, реализирани на базата на концептуалните модели на системни архитектури, предложени в настоящия дисертационен труд.

Изводи

Данните показват среден ръст от около 50% на годишна база на интернет потреблението, а средно 2.5% от световното население е ръстът на интернет потребителите на годишна база, като се наблюдават и по-големи скокове, до 8% на година. Това го прави най-бързо развиващият се пазар, като тази тенденция на бърз ръст е без промяна през последните 2 десетилетия. Масовостта, която предоставя интернет на постоянно развиващия се пазар, даде възможност на стартиращи компании да се конкурират с пазарни лидери. Докато десетина години по-рано, притежанието на интернет сайтове се асоциираше към големия бизнес или научните среди, в последните години интернет сайтовете са се превърнали в задължителен елемент от представянето на всяка една организация или дейност. Стартиращите предприятия намират за необходимост първият канал, през който да предоставят услугите или продуктите си, с възможно най-малка инвестиция, да е интернет. За много компании интернет е единственият канал за правене на бизнес. Такива компании например са за електронна търговия, за услуги, които могат да бъдат извършвани от разстояние, софтуерни компании и др.

Нуждата от бърза, достъпна и качествена услуга за производство на интернет страници за масовия потребител, е променила изцяло насоката на много софтуерни компании към разработката на уебсайтове.

Глава I: АНАЛИЗ НА СЪСТОЯНИЕТО НА ПРОБЛЕМА

За целите на дисертацията бяха изследвани над 100 съществуващи онлайн ССУ. Като резултат е синтезирана класификация на видовете ССУ по основни критерии. Изследваните ССУ са добавени, като Приложение 1. Класификацията е базирана на основни характеристики, които са фактор при определянето на технологичната сложност на системната архитектура при проектиране и реализиране на ССУ.

Проведено е проучване на степента на пазарно проникване на ССУ бранша на Шведския пазар.

Класификация на ССУ

1. Според средата на изпълнение
 - Офлайн ССУ
 - Онлайн ССУ
2. Според начина на публикуване и представяне на съдържание
 - ССУ с шаблонно публикуване
 - ССУ с фрагментно публикуване
3. Според начина на визуална редакция и вида UI/UX
 - Базирано на форми
 - Визуална редакция за публикуване на съдържание с адаптивен дизайн
 - Визуална редакция с релативно позициониране
 - Визуална редакция с абсолютно позициониране
4. Според вида уебсайтове, които публикува ССУ

- ССУ за създаване на лендинг страници
 - ССУ за създаване на специализирани нишови уебсайтове
 - ССУ, като заместител на CMS платформите за разработчици
5. Според целевата група потребители на ССУ
- Специализирани ССУ са предназначени за IT и дизайн професионалисти, уеб агенции
 - ССУ, насочени директно към масовите потребители
 - ССУ, непродващи директно до потребители, а само чрез посредници
 - ССУ без разпознаваем бранд сред крайните потребители - Whitelabeled
 - ССУ с ко-брандиране - Cobranded

Научни разработки, основни съпътстващи технологии и публични изследвания в областта

Почти всички проучвания в сферата на ССУ се провеждат на корпоративно ниво, не се публикуват, защото се третира като чувствителна интелектуална собственост (IP). Наблюденията в сферата на интелектуалната собственост показват, че по-големите корпорации в сферата на ССУ правят публични научни разработки, които вече видимо са развити и от други производители на ССУ, като това са технологии, които обикновено са базови или дори излезли от употреба, заместени от по-добри.

Тези публикации са под формата на патенти, като много малка част се публикуват отворено. Най-много одобрени патенти или кандидатури за патенти се наблюдават през 2016 г. Това е по-скоро тенденция и наблюденията показват, че е вероятно тази тенденция да се засилва през следващите години.

Много повече научни изследвания са налични, отнасящи се до различните части на ССУ и съпътстващите технологии, свързани с различните подсистеми, съпътстващи архитектури и модели, стандарти, шаблони за дизайн на софтуер и т.н. Водейки се от публичните изследвания в областите, засягащи подсистемите и съпътстващите технологии, в настоящия дисертационен труд са отсети най-добрите познати практики, както и са предложени най-добрите практики за сферите, в които публичните изследвания не достигат.

1. Изследвани научни разработки на съпътстващи технологии и публични изследвания
- Софтуер като услуга
 - Многонаемателност и автентикация
 - Мащабируемост, уеб сървъри, бази данни
 - Патенти и изследвания конкретно в сферата на ССУ

Изводи

- Ниско ниво на публичност
- Иновативни подходи в архитектури се държат като корпоративни тайни
- Публикуват се вече остарели технологии, в момент, в който дадена компания мигрира към по-нова технология
- Основна цел на публикуваните патенти е за повишаване на пазарна стойност на компаниите, както и за защита от нови конкуренти, тепърва развиващи се с предходни технологии
- Остарели технологии в сферата на ССУ се считат такива, навлезли преди повече от 3 години
- Въпреки засиленото навлизане на ССУ в последните 10 г., пазарното проникване е едва в началото си

Глава II: ИЗСЛЕДВАНИЯ И РАЗРАБОТКА НА ЕКСПЕРИМЕНТАЛНИ ССУ С ЦЕЛ СТРУКТУРЕН АНАЛИЗ И СИНТЕЗИРАНЕ НА ОПТИМАЛНА ССУ АРХИТЕКТУРА

Изследвани, проектирани и имплементирани са 4 концептуални модела на архитектура на ССУ от различни базови класове. Използвани са различни подходи, като основна цел на експерименталните реализации е доказване на основни хипотези при проектирането на концептуални модели на ССУ архитектури. Всеки предложен концептуален модел включва различен набор от авторски подсистеми за ССУ.

ССУ за създаване на многофункционални, еднотипни уебсайтове с автоматизирано взаимно обмяна на съдържание

Проектирането, изследванията и разработките са проведени в период от почти 1 година, от Април 2009г. до Март 2010г.

Целта на тази ССУ е да позволи на потребителите да създадат уебсайт от портален тип, съставен от множество модули за публикуване на съдържание, както от потребителите на ССУ, така и от потребителите на техните портални уебсайтове.

Уникалната част на тази ССУ е да открива тематично съдържание между различните уебсайтове и, чрез него, автоматично да генерира допълнително съдържание към уебсайтовете на потребителите, по начин, по който то няма да бъде засечено като дублиращо съдържание от търсачките. Резултатът е експоненциално подобрене на оптимизация на търсачките (SEO) на ССУ, относително към използваемостта на системата.

Системата е финансирана и изградена по поръчка на клиент от САЩ, с цел доказване на концепция за бизнес модел и подsigуряване на по-нататъшно финансиране, на по-мощно изграждане на нова ССУ на базата на тази концептуално доказана архитектура.

Системата се проектира за доказване на концепция за реализация на хипотетичната архитектура на ССУ, както и ефективността върху търсачките. Системата не цели оптимизация за масовост и високо натоварване, като ще разчита на повече хардуерни ресурси за периода си на работа. Като целта е при доказването на концепцията, да се препроектира и оптимизира за високо натоварване. Като основна задача при реализацията на тази ССУ, е да бъде създадена бързо и, в рамките на минимален бюджет.

1. Доказани хипотези:

- Възможно е създаването на метод/и за преобразуване на стандартен, инсталируем CMS в онлайн ССУ с възможност за многонаемателност;
- Имплементация, базирана на такъв CMS, ще намали производствените разходи и срокове;
- Система за оптимизация на рекламите може да оптимизира конверсиите, чрез контекстно позициониране;
- Експерименталната ССУ ще е възможно да поддържа поне 1000 уебсайта с органичен трафик на един уеб сървър, без нуждата от инфраструктурно разширяване, освен вертикално скалиране на самия сървър.

2. Създадени са следните методи и алгоритми, след което са експериментално доказани:

- Метод и алгоритъм за конвертиране на софтуер или модул от софтуер, от еднонаемателна в многонаемателна архитектура.
- Метод за капсулирано конвертиране от еднонаемателна към многонаемателна архитектура.
- Метод за конвертиране на еднонаемателна към многонаемателна архитектура чрез подsigуряване на многоинстантност.

3. Резултати

- Създаден е концептуален модел на архитектура за решаването на зададените проблеми и на негова база е реализирана експериментална ССУ, която изведе конкретни ясноти за възможна структура за ССУ;
- Проектирана е фундаментална система за рутиране на домейни, по които да се идентифицира уебсайта на потребителя на ССУ;
- Проектирана е фундаментална система за генериране на шаблони;
- Проектирана е система за автоматизирано съдържание, която може да открива релативно съдържание между уебсайтове, без да вдига флаг за дублирано съдържание при търсещите машини;
- Проектирана е система за автоматизирано управление на контекстна реклама според съдържанието на уебсайтовете на ССУ и е интегриран рекламен сървър;
- Интегрирана е Уеб ОС за управление на ССУ.

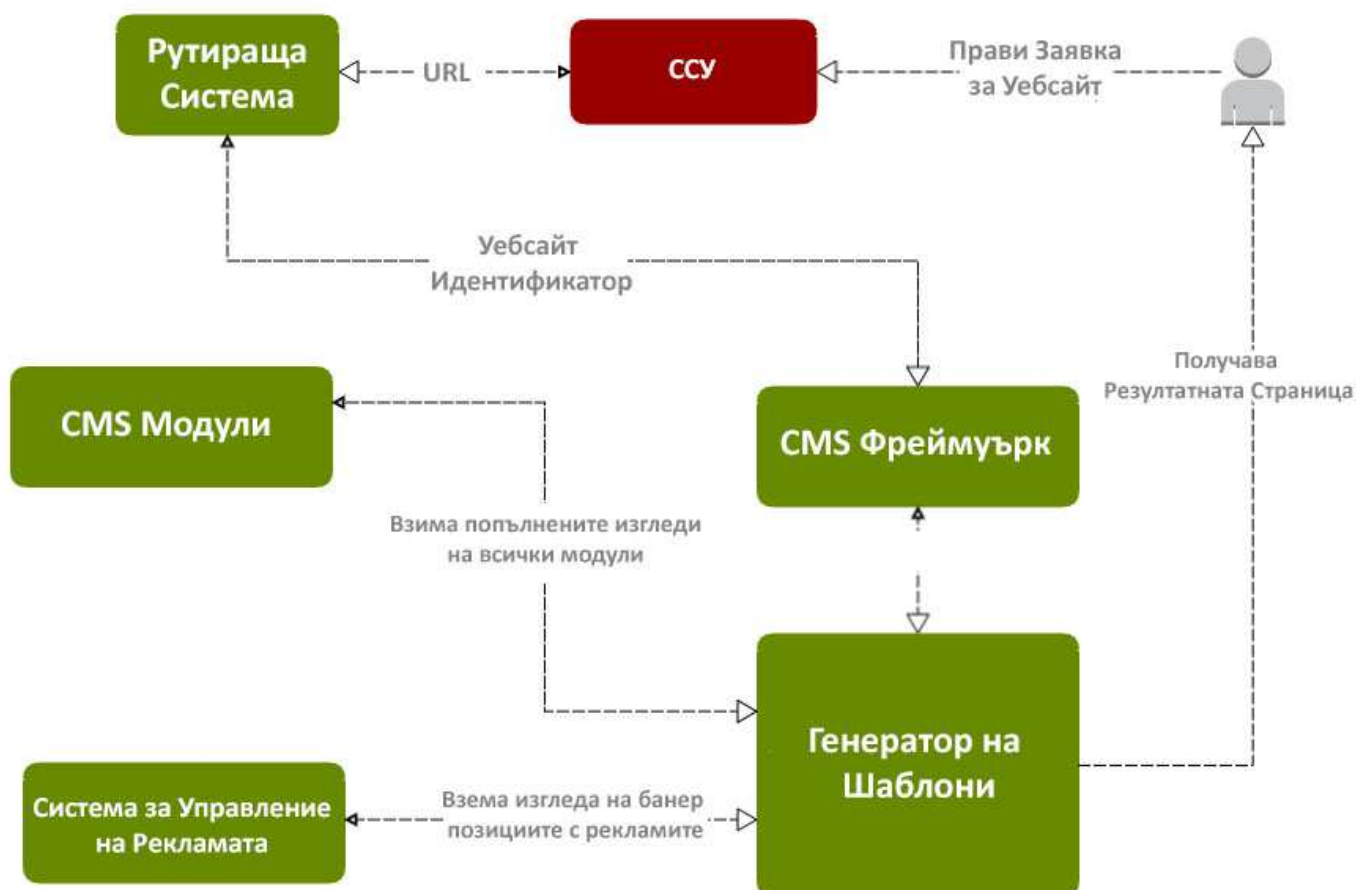
4. Архитектури

Фигура 5 демонстрира какво се случва при заявка за създаване на нов уебсайт. Потребителят въвежда 5 ключови фрази, те се подават на системата за предложения на ключови фрази. Тя имплементира услуги като Yahoo Keyword Suggest и сходни от други

търсещи машини. Като резултат намира по 10 ключови фрази за всяка от подадените, тоест общо 5 основни и 50 допълнителни ключови фрази. Те се подават към системата за генериране на уебсайтове.



Фигура 5. Архитектура на ССУ в частта на създаване на уебсайта



Фигура 6. Схема на процеса на опериране на експерименталната ССУ

Генераторът на уебсайтове подава домейна към рутиращата система, която на свой ред връща уникален идентификатор, който ще се използва при генерирането на структурата във всички останали системи. Генераторът има единен интерфейс, за създаване на структура (категории/секции/колекции), като към всеки един модул от CMS-а се създават 5 основни категории и по 10 подкатегории във всяка основна категория, с имената на ключовите фрази.

Както може да се види във Фигура 6, всяка една заявка от потребител към ССУ преминава през рутиращата система. Самата рутираща система е една развита обвивка над съществуващата рутираща система на CMS-а. Това от части я ограничава, но в същото време прави изграждането ѝ по-бързо. Рутиращата система идентифицира уебсайта според подадения URL, като подава към ССУ идентификатора, с който тя работи. CMS Фреймуърк (Framework – платформа/система за софтуерни разработки) намира нужния ресурс, отговарящ на заявения URL. Това например е началната страница или страницата на някой модул, като блог списък, чат и т.н. След това използва т.нар. Генератор на Шабини (Template Engine), който генерира резултата до клиента. В самия шаблон има определено място за всеки модул, като извиква изглед на всеки конкретен модул с данните на уебсайта, който се изисква. Същото става и с позициите на банерите, които се извикват от рекламния сървър. Целият този резултат се сглобява и се връща към клиента.

5. Изводи

Създаването на тази експериментална ССУ изведе конкретни ясноти за възможна структура за ССУ в години, когато публичната информация за ССУ архитектури е много ограничена. Основните публикации в областта се намират едва след 2012г. и по-сериозно едва след 2016г., като и към този момент са все още оскъдни.

Рутиращата система е базова, надградена върху рутираща система на самия прилежащ CMS. Въпреки това, може да се счита за основен компонент на ССУ архитектурите, като тя може да е част от друга система или самостоятелно изградена.

Хипотезите, заложи при разработването на експерименталната ССУ бяха емпирично доказани. Системите, които са концептуално релативни към ССУ и не допринасят за демонстрация на процесите на работа, не са разглеждани тук. Такива са системата за автоматизирано съдържание, която успя да докаже ефективността си с търсещите машини, системата за оптимизация на рекламите, чрез контекстно позициониране, с развитие на ССУ успя да демонстрира оптимизация. ССУ демонстрира добра работоспособност на над 1000 уебсайта едновременно, въпреки това експериментално се доказва, че това е близо до горния лимит на възможностите на системата, ако работи на едносървърна инфраструктура.

Системата за генериране на уебсайтове е една първа крачка към изграждането на цялостна архитектура за такава система. В момента тя е трудно мащабируема, има нужда от разделение на ключови подсистеми, което би било полезно при една ССУ за масови потребители. Тя, също така, не позволява редакция по вече генерираното съдържание, а разчита на конвертирания CMS за това.

ССУ за създаване на микросайтове за хотелиерство и туризъм, с портал за консолидирането им

Проектирането, изследванията и разработките са проведени в период от 6 месеца, от Ноември 2009 г. до Май 2010 г.

Проектът, който все още е активен под домейна hotelmap.bg, представлява портал за хотели, заведения, забележителности и събития. Обектите към този момент са над 7000, като представянето в портала се показват върху карта, чрез Google Maps интеграция или чрез стандартен списък в галерия.

За целите на настоящия дисертационен труд ще бъде разглеждана само ССУ частта на проекта и интеграцията ѝ с останалата част от проекта. Останалата част от реализацията е пример как може една ССУ да се интегрира или надгради към допълнителна система. Също така, разработките и изследванията, проведени по реализацията на тази експериментална система, са първа стъпка към автоматично създаване на портали на базата на ССУ.

Изследвани са предимствата при изграждане и поддържане на тематични онлайн портали, базирани върху ССУ.

Изследвани са зависимостите на концентрираната тематична информация с добър стандартизиран UX, като принос за повишаването на органичния трафик на портал, изграден върху ССУ.

Изследвани са технологичните нужди за SEO оптимизация, управление на онлайн репутацията (ORM) и повишаване на ранга (PR) на страниците в портал, изграден на базата на ССУ, като са изследвани възможностите за вътрешно свързване между уебсайтовете на ССУ, допринасящо за повишение на ключовите показатели от изброените характеристики.

Изследвана е практическата реализация на тематичните онлайн портали, изградени на базата на ССУ, като разработване на изцяло отделен пазарен сегмент.

1. Доказани хипотези

- Възможно е да се използва ССУ за източник на съдържание за тематичен онлайн портал;
- Концентрирането на тематично съдържание в онлайн портал повишава органичния трафик на портала и неговата онлайн репутация;
- Изграждането на тематични онлайн портали на базата на ССУ, освен директния трафик от портала, се повишава и трафикът от органични източници към отделните уебсайтове от системата;
- ССУ може да се използва като средство за създаване и поддържане на тематични онлайн портали.

2. Резултати

- Създаден е концептуален модел на архитектура за решаването на зададените проблеми и на негова база е реализирана експериментална ССУ;
- Създаден е т.нар. WYSIWYG (What You See Is What You Get) редактор за съдържание, като се използва един и същи шаблон за редактиране на информацията и презентиране на съществуващ уебсайт от ССУ. Проектирането на архитектурния модел на редактора е фундамент за развиването на по-сложни системи за визуална редакция;
- Тематичните онлайн портали, изградени на базата на ССУ, се разглеждат като отделен пазарен сегмент, като паралелно спомагат за по-бързото пласиране на самата ССУ;
- Проектирани са фундаментална за ССУ системи за управление на уебсайт данни и за кеширане на уебсайтове.

3. Архитектури



Фигура 7. Концептуален модел на архитектура на ССУ с интеграция към онлайн портал

В следствие на изследванията, проведени по време на имплементиране на експерименталната ССУ, бяха установени множество ползи от изграждането на тематични онлайн портали към ССУ, като към една ССУ може да съществуват множество онлайн портали.

С развитие на пласирането на ССУ, всеки онлайн портал, използващ данни от конкретното ССУ, повишава трафика си от органични източници и подобрява онлайн репутацията си. Ефекта е двустранен и влияе за оптимизация на уебсайтовете от ССУ, като те имат препратки от портали, които са развили висока репутация. Отделно - самите портали генерират и директен трафик към тях. Безспорно, изграждането на тематични онлайн портали е ускоряваща съставна част от архитектурата на една ССУ.

В тази система с обобщена ССУ архитектура, описана във Фигура 7, беше реализиран базов WYSIWYG редактор на съдържанието, който е първа крачка към по-сложни реализации на визуални уебсайт редактори.

Изследванията и публикациите, направени след имплементацията на експерименталната архитектура относно приложенията на ССУ в изграждане на портални уебсайтове и възможните оптимизации, следващи от тази интеграция, са финансирани и извършени в рамките на евро проект ОП 'РЧР', BG051PO001-3.3.06-0005 на ЕС.

ССУ за създаване на електронни магазини за Facebook

Проектирането, изследванията и разработките са проведени в период от 6 месеца от Октомври 2010 г. до Април 2011 г.

Системата представлява ССУ за създаване на електронни магазини, предвидени за вграждане във външни платформи. Системата използва Facebook като първа платформа, към която да се интегрира. Вграждането е автоматизирано към Facebook страниците на потребителите.

По време на провеждането на изследванията и имплементацията на тази ССУ не бяха открити сходни системи, които да реализират подобен краен продукт. Няколко години по-късно вече съществуват множество такива ССУ.

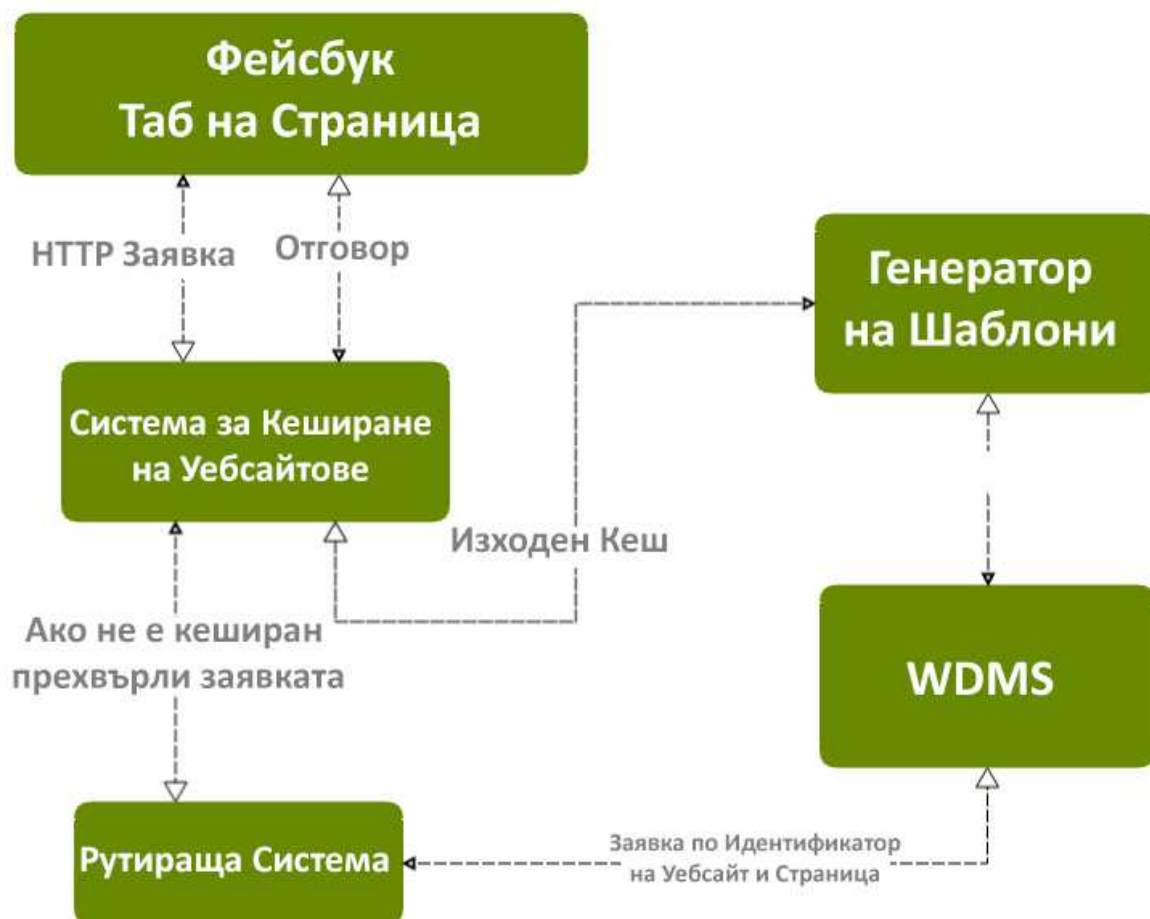
ССУ за електронна търговия би изисквала допълнително натоварване, поради спецификата си всеки електронен магазин да съдържа продукти от порядъка на десетки хиляди. По тази причина трябва да се идентифицират и въведат практики за оптимизация.

За времето на експеримента, системата функционира под домейна getamall.com.

1. Резултати и доказани хипотези

- Създаден е концептуален модел на архитектура за решаването на зададените проблеми и на негова база е реализирана експериментална ССУ под бранда GetAMall;
- Възможно е да се автоматизира интеграцията на ССУ с външна система, позволяваща вграждане на код (code embedding), като експерименталната ССУ от тази точка демонстрира работеща архитектура при интеграция на ССУ с Facebook, като вграждането на кода става чрез приложения;
- Възможно е всеки уебсайт да бъде разширяван по функционалност, активирана спрямо идентификатора си;
- Върху експерименталната система бяха изследвани, приложени и описани най-добри практики за оптимизация, с цел максимална разширяемост и възможност за мащабиране на ССУ;
- Проектирана е кешираща система с динамично кеширане на данните, която да работи адекватно за ССУ с динамични промени в БД;
- Към кеширащата система е проектирана подсистема за динамично инвалидиране на кеша, която да обновява само тази част от дадена страница, в която промяната на данните имат отражение, като запази кеширането на останалата част от дадената страница;
- Проектирана е фундаментална система за шаблони (Template Engine), която може да подава динамично избран шаблон, на който динамично да прилага нужни конфигурации;
- Емпирично доказано е, че системата работи адекватно при БД от 1 млн. продукта във всички магазини.

2. Архитектура на експерименталната ССУ



Фигура 8. Архитектура на експериментална ССУ за създаване на електронни магазини за вграждане във Facebook

За целите на интеграцията на ССУ към Facebook беше създадено Facebook приложение, което реално отваря в рамка уебсайт от ССУ директно във Facebook страницата. Описано във Фигура 8, приложението прави заявка с URL идентифициращ магазина, както и допълнителните параметри, като страница, параметри от търсене и т.н. Заявката пристига в кеширащата система, ако на този URL отговаря кеширан запис, то се връща директно, ако не се създава нов кеш, който се връща.

Създаването на нов кеш се случва чрез изпращане на URL адреса до рутирещата система, след което се идентифицира уебсайтът, страницата и всички нужни параметри за т.нар. части от страница (partials – паршъли). След което, системата за шаблони взема данните и ги вгражда в шаблона, който потребителят е избрал. След това преминава през преконфигуриране на избрани от потребителя параметри за цветове, фонове и т.н. След като се генерира, крайният изглед се подава на кеширащата система. Кеширащата система работи на няколко нива, а не плоско на ниво страница. Тя кешира всеки един паршъл от страницата по отделно.

3. Архитектура на системата за динамично инвалидиране на кеш

Описаната система за динамично инвалидиране не уеб кеша записва метаданни към всеки кеширан обект (паршъл или страница), които го обвързват с конкретен обект от WDMS. Системата за инвалидиране на кеша се закача за WDMS и слуша за промените по обектите, като когато един обект претърпи промяна, се вдига флаг за инвалидация във всички кешове, в които участва. Системата за кеширане съглобява страницата от кеширан общ изглед, като към него динамично вгражда всеки един паршъл на мястото му. Този резултат също може да се

кешира допълнително при по-бавно променящи се системи, но не се препоръчва при динамични такива.



Фигура 9. Архитектура на система за динамично инвалидиране на кеш, подсистема от системата за кеширане на уебсайтове

Например, както е описана предложената архитектура от Фигура 9, е сменена цената на Продукт 1, веднага след промяната се изпраща съобщение към системата за инвалидиране на кеша, която на свой ред намира всички кеширани обекти, в които участват данни от Продукт 1, като в случая са отделен паршъл, с изглед на продукта и продуктов списък. При по-сложни кеширани обекти, като например резултати от търсене, се залагат по-интелигентни подходи, като се преизчислява позицията на продукта спрямо заявката. Ако той пак се отрази на същия URL адрес, спрямо параметрите на търсене, то се инвалидира само тази страница, а ако се отразява на друг URL адрес, то се инвалидират всички страници, до които се случва пренареждането.

Подобен вид умно кеширане трябва да се развива спрямо функционалността и системата, в която се имплементира, но невинаги инвалидиране на цял набор от данни е нещо разумно, защото при натоварени системи би означавало допълнително натоварване. Поради сложността на изпълнение повечето изследвания и предложения за кеширащи системи предлагат залагането на динамични паршъли, които не се кешират, вградени в кеширана рамка.

4. Изводи

Експерименталната ССУ от тази точка демонстрира работеща архитектура при интеграция на ССУ с външна платформа, позволяваща вграждане на код чрез приложения.

Беше демонстрирана работеща система за управление на шаблони в експериментална ССУ, позволяваща освен избор между предефинирани шаблони, да се сваля и редактира всеки един от шаблоните за цялостно персонализиране. В допълнение, беше добавен слой за конфигурация, който се прилага динамично при построяване на изгледа и се управлява през

UI от потребителите на ССУ. Чрез тази ССУ се постигна и възможност на вграждане на собствен код и интеграция на други външни системи.

Системата е експериментално тествана с близо 1 милион записа на продукти, като оперирането не среща видими проблеми, забавяния и сринове.

Към системата беше изградена кешираща система, която, в допълнение на стандартни и предходно прилагани подобни системи, има включена към себе си подсистема за динамично инвалидиране на кеша.

Проектираната система за динамично инвалидиране на кеша свързва всеки обект от системата за управление на уебсайт обекти (WDMS) на ССУ към всеки кеширан обект, който може да е както цяла страница, така и част от страница (паршъл). Системата е разширяема за оптимизиране на динамична функционалност, позволяваща подобрение чрез инвалидиране само на минимално нужните кеширани обекти.

Върху експерименталната система бяха изследвани, приложени и описани най-добри практики за оптимизация, с цел максимална разширяемост и възможност за мащабиране на ССУ.

Специализирана ССУ за автоматизацията на работата на уеб отдел на компания

Проектирането, изследванията и разработките са проведени в период от 16 месеца от април 2011 г. до август 2012 г. Първичната интеграция се изпълни на четвъртия месец от старта на експерименталния проект, като анализите и подобренията се извършиха в следващата 1 година.

Създадената експериментална ССУ от тази точка, за разлика от предходните, е направена за вътрешна употреба на ИТ отдел с цел оптимизиране на цялостното опериране на отдела на компания, която освен за собствени нужди, създава типови уебсайтове и уеб приложения за свои контрагенти по различни договорености.

Създадената експериментална ССУ от тази точка, за разлика от предходните, е направена за вътрешна употреба на ИТ отдел, с цел оптимизиране на цялостното опериране на отдела на компания, която, освен за собствени нужди, създава типови уебсайтове и уеб приложения за свои контрагенти по различни договорености.

Целта на експеримента е да се изследва дали инвестиция в интеграция на ССУ за подобрение на оперативните процеси в компании или отдели с уеб разработки, е икономически оправдана, както и да се изследват и дефинират подобренията, до които тази интеграция води.

На практика целта на експеримента е да измести стандартни инсталации, настройки и преконфигурации на типови CMS платформи със ССУ и да измери разликата в ефективността и преимуществата в стандартизацията на бизнес процесите, които ССУ предоставя.

ССУ е интегрирана за производствените процеси на целия ИТ отдел, като разгледаните резултати са само за част от отдела, специализиращ в производството на малки маркетингови приложения. Резултатите от останалата част от отдела са сходни, но не се получи разрешение за споделянето им, поради спецификата на дейността на компанията.

Изследванията за стандартизация са проведени и финансирани в рамките на европейски проект BG051PO001-3.3.06-0005, Програма за Развитие на Човешките Ресурси. Имплементацията и останалите изследвания са финансирани от международна компания, в чиито отдел се извършват.

Резултатите са презентирани и публикувани на International Science, Technology And Engineering Conference, ISTECH – Dubai, Dec. 2012.

1. Резултати, доказани хипотези и изводи

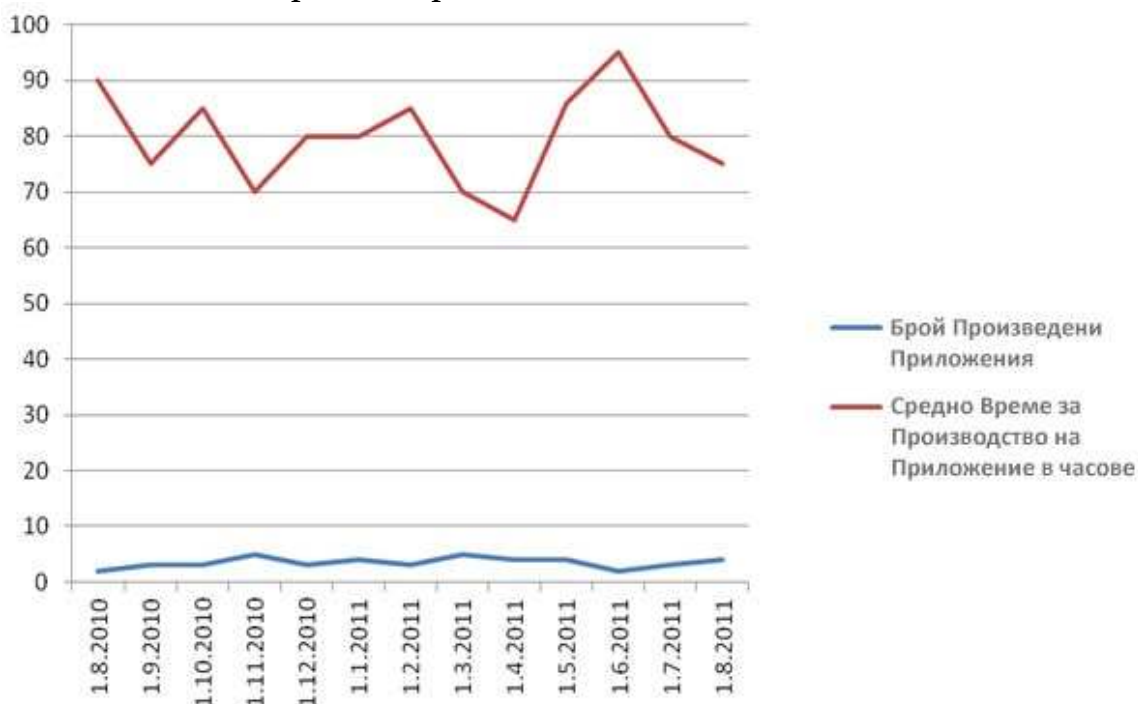
- Експериментите за проучването са извършени в реална работна среда;
- Създаден е концептуален модел на архитектура за решаването на зададените проблеми, автоматизиращ създаването на еднотипни уебсайтове и приложения;
- Създадена е експериментална ССУ на базата на концептуалния модел;
- ССУ може да служи за технологичен гръбнак на компанията, върху който да се изградят производствените процеси;
- Реализирана беше стандартизация на производствените процеси;

- След установяване на стриктни стандарти и методи на следене за спазването им, е отразена трайна тенденция на подобрена ефективност.

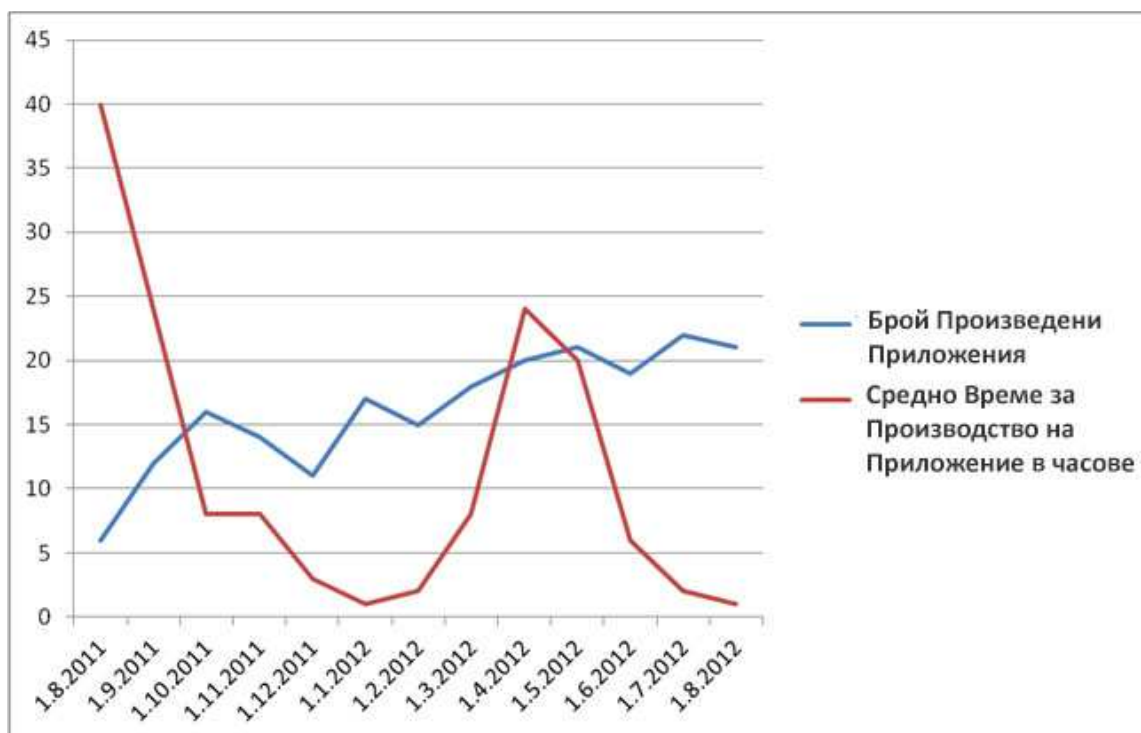
Чрез интеграцията на ССУ, като основа за разработки към IT отдел, може да се постигне:

- Налагане на стандарти в програмирането;
- Стандартизация: на процеса за създаване на уебсайтове и уеб приложения; в администрирането; в потребителския интерфейс и потребителското преживяване; в клиентската поддръжка; при обмена на данни;
- Улеснено придобиване на международни стандарти за качество;
- Намаляване на времето за производство, съответно - повишаване на броя завършени проекти на месечна база;
- Намаляване на разходите за производство, потребителска и техническа поддръжка, съответно - повишаване на печалбата.

2. Изследване на индикаторите на производителността



Фигура 10. Статистики на производството преди интеграция на специализирана ССУ



Фигура 11. Статистики на производството след интеграция на специализирана ССУ

Изследвайки статистиките, визуализирани на **Error! Reference source not found.** и Фигура 11, ако приемем, че средният „Брой Произведени Приложения“ преди прилагане на ССУ е $\overline{Nb}$, а след прилагане е $\overline{Na}$. $\overline{Nb}$ и $\overline{Na}$ са измерение на производителността на бизнес процеса, а функцията измерваща процентната оптимизация на показателя при прилагане на ССУ върху бизнес процеса е изразена като $WBS(Nb)$, то отношението с измерените статистически данни може да се формализира по следния начин:

$$WBS(\overline{Nb}) = \frac{\frac{1}{n} \sum_{j=1}^n Na_j}{\frac{1}{n} \sum_{j=1}^n Nb_j} = \frac{\overline{Na}}{\overline{Nb}} \approx \frac{16.30}{3.46} * 100\% = 471\% \quad (4.5.1)$$

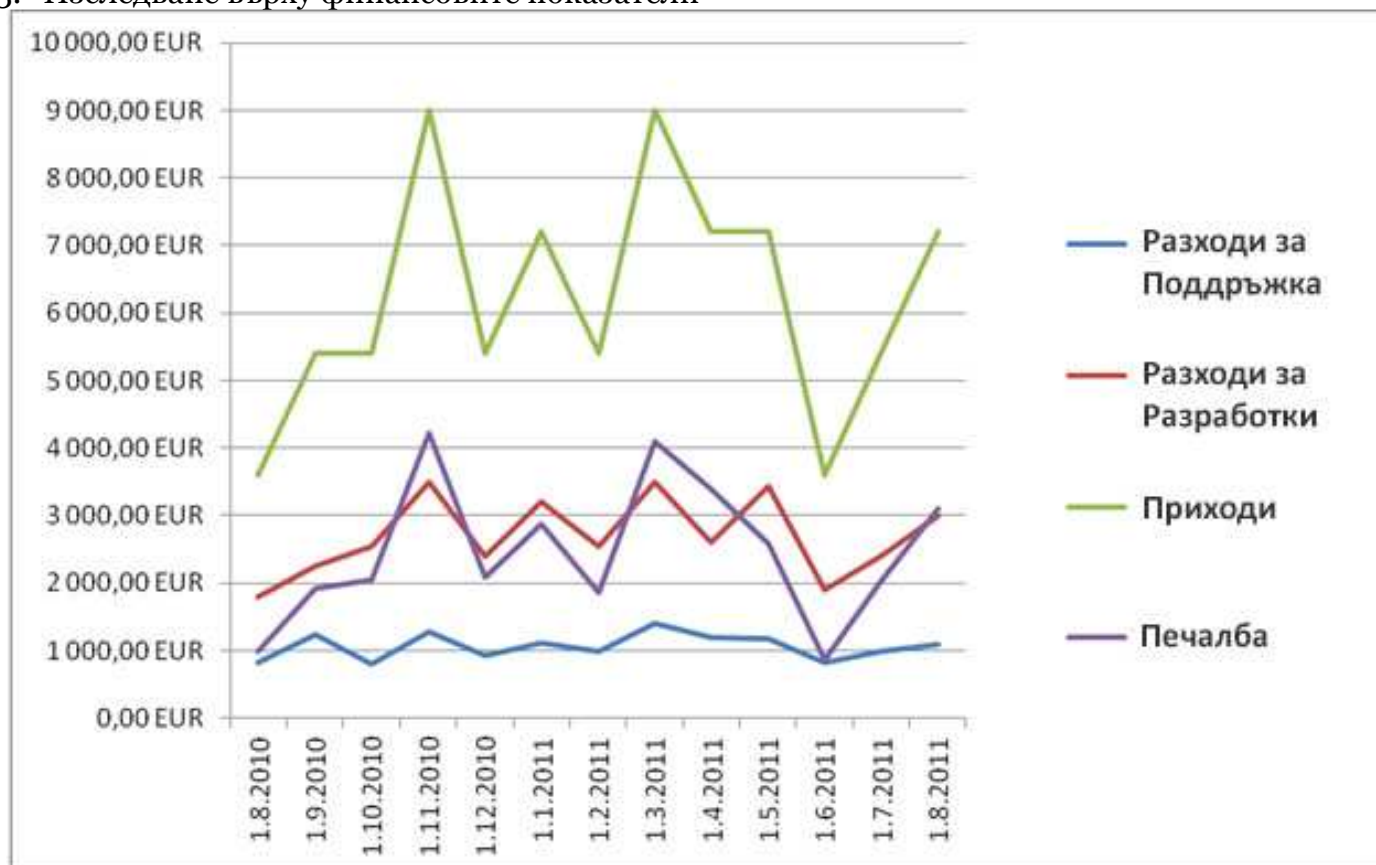
Средната оптимизация на това измерение на производителността е 471%. Във формализацията n е равните периоди на измерването, преди и след прилагане на ССУ.

Ако средната стойност на „Средно време за производство на приложение“, измерено в часове на месечна база, преди прилагане на ССУ е $\overline{PTb}$, а след $\overline{PTa}$, то средната оптимизация на това измерение от прилагането на ССУ може да се формализира така:

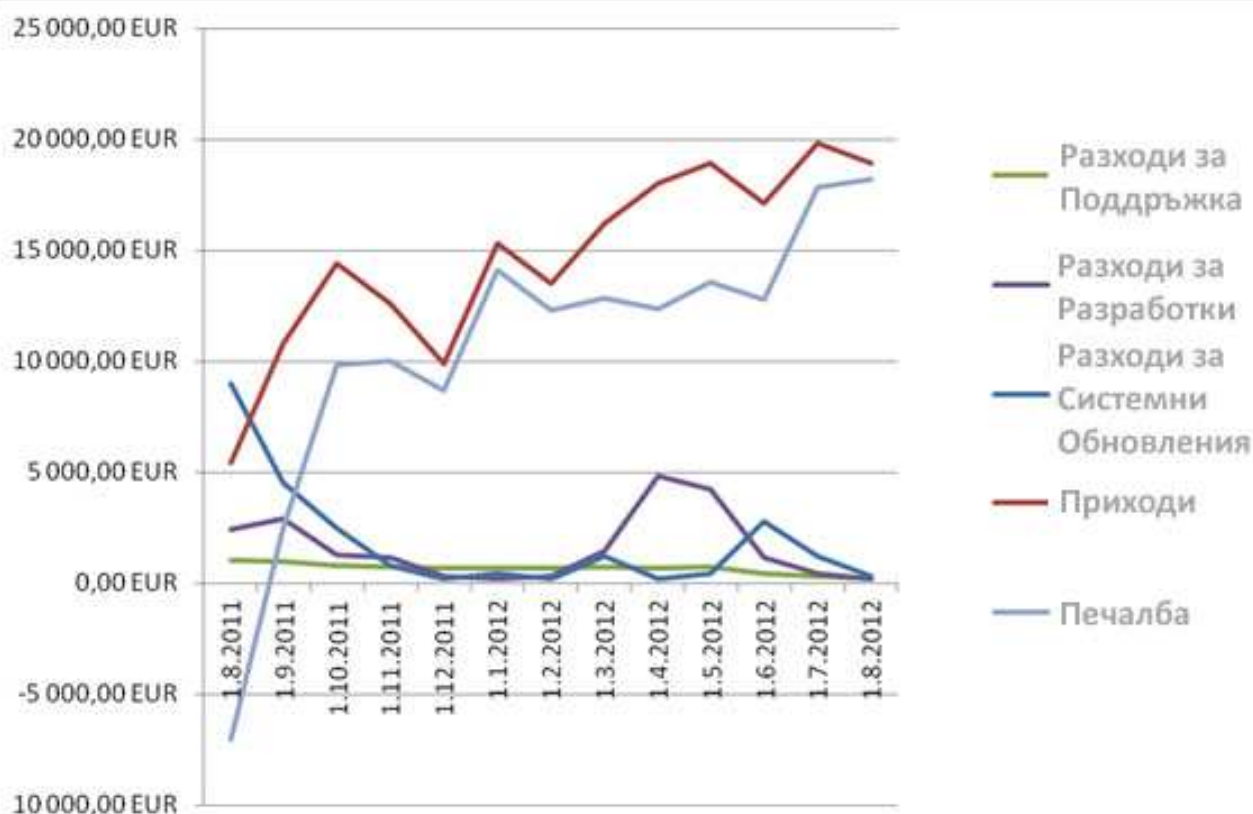
$$WBS(\overline{PTb}) = \frac{\frac{1}{n} \sum_{j=1}^n PTb_j}{\frac{1}{n} \sum_{j=1}^n PTa_j} = \frac{\overline{PTb}}{\overline{PTa}} \approx \frac{79.69}{11.30} * 100\% \approx 705\% \quad (4.5.2)$$

Средната оптимизация на това измерение на производителността е 705%. Тези две измерения са тясно зависими от типа производство, което се извършва, затова е избрано да се измерва целият период осреднено, вместо отделни пикови измервания на единица производство, като за целият период на експеримента се приема, че типа производство, дори и да е разнообразен през отделните дни, е сходен като дейности, за двата отчетни периода на експеримента.

3. Изследване върху финансовите показатели



Фигура 12. Обобщителен финансов отчет преди интеграция на специализирана ССУ



Фигура 13. Обобщителен финансов отчет след интеграция на специализирана ССУ

Изследвайки резултатите, показани на Фигура 12 и Фигура 13, ако приемем, че преди въвеждане на ССУ, разходът за поддръжка е $CSUPb$, разходът за разработки е $CDEVb$, разходът за системни обновления е $CSYSb$, приходите са Ib , а печалбата Pb , а след прилагане на ССУ съответно $CSUPa$, $CDEVa$, $CSYSa$, Ia , Pa . И, ако разгледаме прилагането на ССУ като функция за оптимизация на печалбата, то формално може да бъде описана по следният начин:

$$\begin{aligned}
 WBS(Pb) &= \frac{\sum_{j=1}^n (Ia_j - CSUPa_j + CDEVa_j + CSYSa_j)}{\sum_{j=1}^n (Ib_j - CSUPb_j + CDEVb_j + CSYSb_j)} = \\
 &= \frac{\sum_{j=1}^n Pa_j}{\sum_{j=1}^n Pb_j} = \frac{137\,890}{32\,070} * 100\% \approx 430\%
 \end{aligned} \tag{4.5.3}$$

От наличните данни по отделно може да бъде изчислен относителният ефект на прилагане на ССУ за отделните измервания на ключовите индикатори. Приблизително 430% е ръстът на печалбата в рамките на експеримента, спрямо предходно подадените данни, без чувствително изменение на основни параметри от бизнес процесите, освен прилагането на ССУ в производството.

4. Изводи

Чрез реализация на експериментална ССУ беше реализирана стандартизация на производствените и бизнес процеси. На практика, бяха повишени производствените възможности, като паралелно бяха понижени оперативните разходи, което допринесе за постигане на повишени финансови резултати за компанията, финансираше проекта. Във връзка с изследването, компанията разреши публикуването на обобщени данни на продукцията на част от ИТ отдела, която показва цялостната тенденция, отнасяща се пропорционално към общите данни за цялото производство.

Чрез експерименталния проект беше направено задълбочено проучване и практическото прилагане на множество стандарти за подобряване на производителността и качеството на продукцията. В проучването беше взето под внимание бъдещото сертифициране по

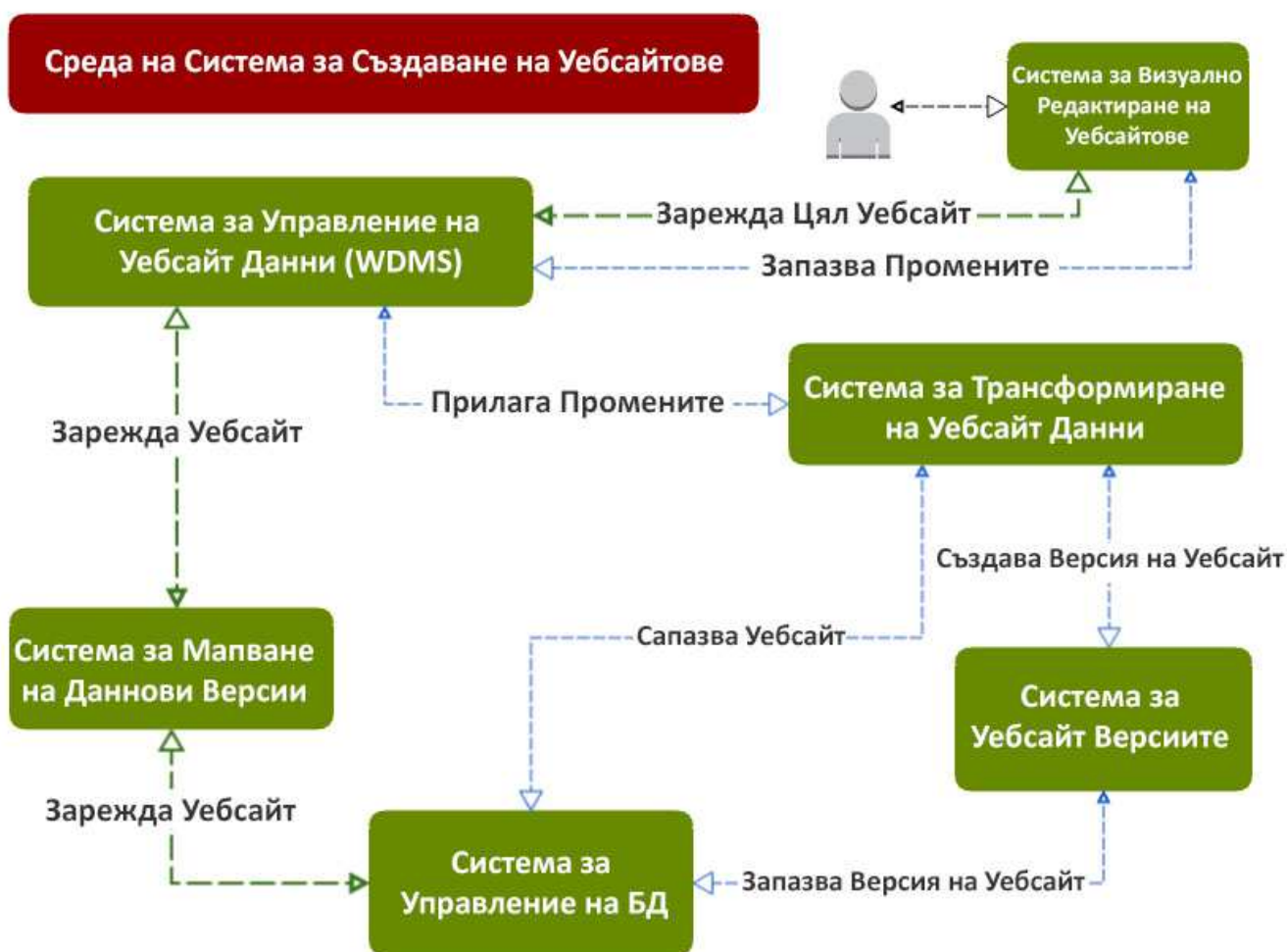
международни стандарти. Въведен е план за сертифициране, като самото сертифициране не влезе в рамките на проведения експериментален проект.

Глава III: ПРЕДЛОЖЕНИЕ ЗА АРХИТЕКТУРА НА ССУ

Предложената архитектура на ССУ в тази глава е базирана на изследвания и експерименти, продължили над 8 години, като част от тях са обобщени в Глава I и Глава II. Крайният вид на архитектурата е постигнат след над 4 години препроектиране и оптимизация чрез изследвания и разработка на експериментална ССУ, описана в Глава IV.

Архитектурата на ССУ е разделена на множество подсистеми, които взаимодействат помежду си. До този момент няма публично достъпна информация за архитектура на ССУ за масовия потребител, т.е. с изискване на усилено, динамично натоварване и разширяемост. Основната информация за разработки в сферата на ССУ се публикува предимно чрез патенти и кандидати за патенти, като става въпрос за единични подсистеми, които често са технологично изостанали и се публикуват само за увеличаване на IP на компанията вносител за защита от нова конкуренция, която тепърва развива ССУ.

1. Обобщение на основните подсистеми на ССУ, като ключови части на предложената архитектура:
 - Система за управление на уебсайт обекти (WDMS) в ССУ



Фигура 14. Архитектура и начин на работа на система за управление на уебсайт обекти

Описаната във Фигура 14 архитектура се използва от ССУ за управление на структурите от данни, които представляват даден уебсайт, който се създава от потребител на ССУ. Описани са методи за оптимизирана манипулация на уебсайт данни, предлагащи абстракции на данновите структури за оптимизиране на операциите в ССУ. Системата е основна при процеса за управление на всички операции по работа с уебсайт обектите.

Описаната система доставя уебсайт данните до останалите подсистеми на ССУ. Тя е отговорна например да предостави уебсайт данните до системата за визуално редактиране в структура, в която тя изисква, като трансформира данните през система за мапинг до изискваната

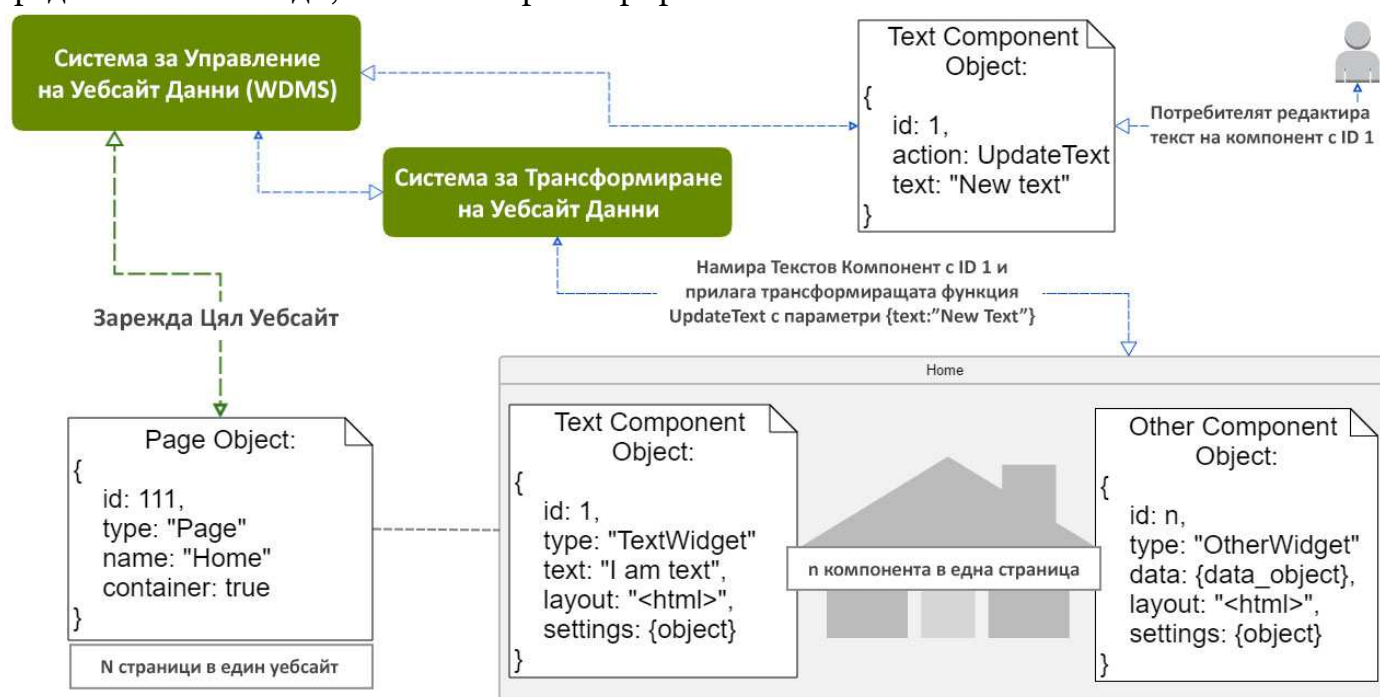
текущата версия. След това се грижи да приложи промените от визуалния редактор към уебсайт данните, да използва системата за версии на уебсайта за съхраняване на предходните варианти. Тя също управлява изпращането на данните за трансформиране в изходен код (HTML, CSS, JS и т.н.) през системата за кеширане на уебсайтове.

■ Система за управление на ДБ в ССУ

Системата е разделена в два основни слоя. Първият е слой за достъп до базата данни (DAL), който варира според вида БД, с който работи. Вторият слой е за т.нар. „мапване“ на данни към обекти (DOML), което означава обвързване на едно представяне на данни към друго представяне. В този слой са разположени ORM и ODM, които преобразуват данните в програмни обекти, съответно за релационни и нерелационни БД. Управлението на данните в ССУ има изискване за много висока производителност и натоварване, с бързо време за отговор и висока степен на наличност (availability) на системата.

■ Система за трансформиране на уебсайт данни

Това е основна подсистема на WDMS, като се използва за операциите за съхранение. Системата използва трансформиращи функции, които реално са всички възможни действия за редакция в една ССУ. Те се описват първоначално най-абстрактно, примерно функция за промяна на целия уебсайт обект и се оптимизират спрямо начините на използване на дадения клас ССУ, примерно до обновяване на едно поле, ако това е често срещано действие. В този случай при промяна на съответното поле, в системата се изпълнява само негова валидация и изпълняват само зависимостите, свързани с него. Изпратената за съхранение информация може да съдържа множество действия, за изпълнение в FIFO ред. Системата е основа и за други подсистеми, като например позволява чрез команден шаблон (command pattern) да се имплементира система за връщане на действията (undo-redo). Фигура 15 описва как тази система се вписва в общия концептуален модел на архитектура на ССУ, како и предложените методи, заложи при оперирането ѝ.



Фигура 15. Пример за начина на трансформиране на данните в процеса на запазване на уебсайт

$$W = \{W_1 \dots W_n\} \quad (2.3.1)$$

$$P = \{P_1 \dots P_m\} \quad (2.3.2)$$

$$C = \{C_1 \dots C_j\} \quad (2.3.3)$$

$$F = \{F_1 \dots F_i\} \quad (2.3.4)$$

$$F \in C_1, \quad C \in P_1, \quad P \in W_1 \quad (2.3.5)$$

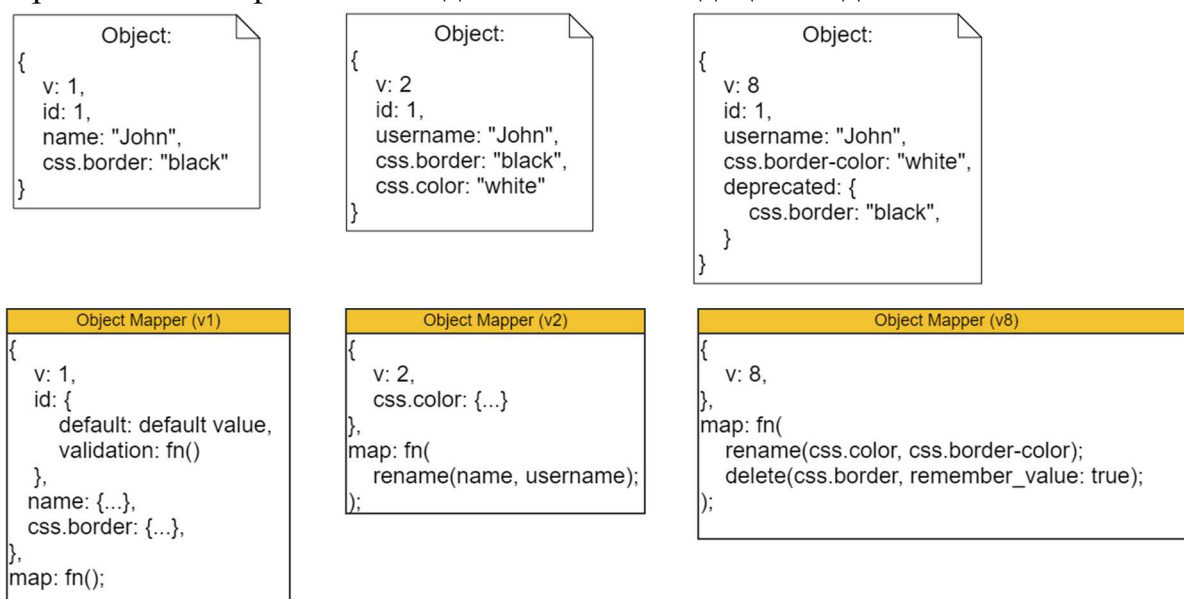
- Система за динамичен мапинг (mapping) на версии на данни в ССУ (DVM)

Това е основна подсистема на WDMS, като най-често се използва при зареждане на уебсайт от системата за визуално редактиране. Системата позволява съхранението на множество версии на дадена структура от данни (обект) на компонент, в зависимост от времето на създаването му, да мигрира и работи на всяка версия на ССУ, което на свой ред позволява ускорено разработване и кратки цикли на издаване (release cycles). Системата се базира на разработен метод за динамично мапване на версиите.

Производителите на ССУ избират различни подходи за процеса на управление на издаване на софтуер (release management). Най-масовият подход е да има дългосрочни цикли на издаване. В повечето случаи се наблюдава тези цикли да са от порядъка на 3 години за основно издаване на нова версия, с чести качвания на поправки (hot-fixes), които много по-рядко имат нужда от миграция на данни, а ако имат, тя се извършва по стандартните подходи, като при нужда дори се извежда системата офлайн, в режим на поддръжка за времето на миграция. За по-големи масиви от данни това може да означава неработен режим с часове, и при случаи на неуспешна миграция или проблеми при миграцията, може да се стигне до влошаване на услугата и неработни режими с дни.

Когато производител на ССУ избере динамични (rapid) процеси на разработване и издаване на версии, новите версии може да се издават от няколко пъти в годината до няколко пъти в седмицата. Тези динамични процеси изискват един много по-чувствителен подход за мигрирането на данните между изискваните версии. Производител на ССУ може да реши определени групи от потребители да използват различни версии на ССУ. В този вариант, различни версии на ССУ може да се деплойнат (инсталират на сървърна инфраструктура) и да работят в един и същ момент. Това на свой ред означава, че информацията трябва да е налична за всяка една версия на софтуера по едно и също време.

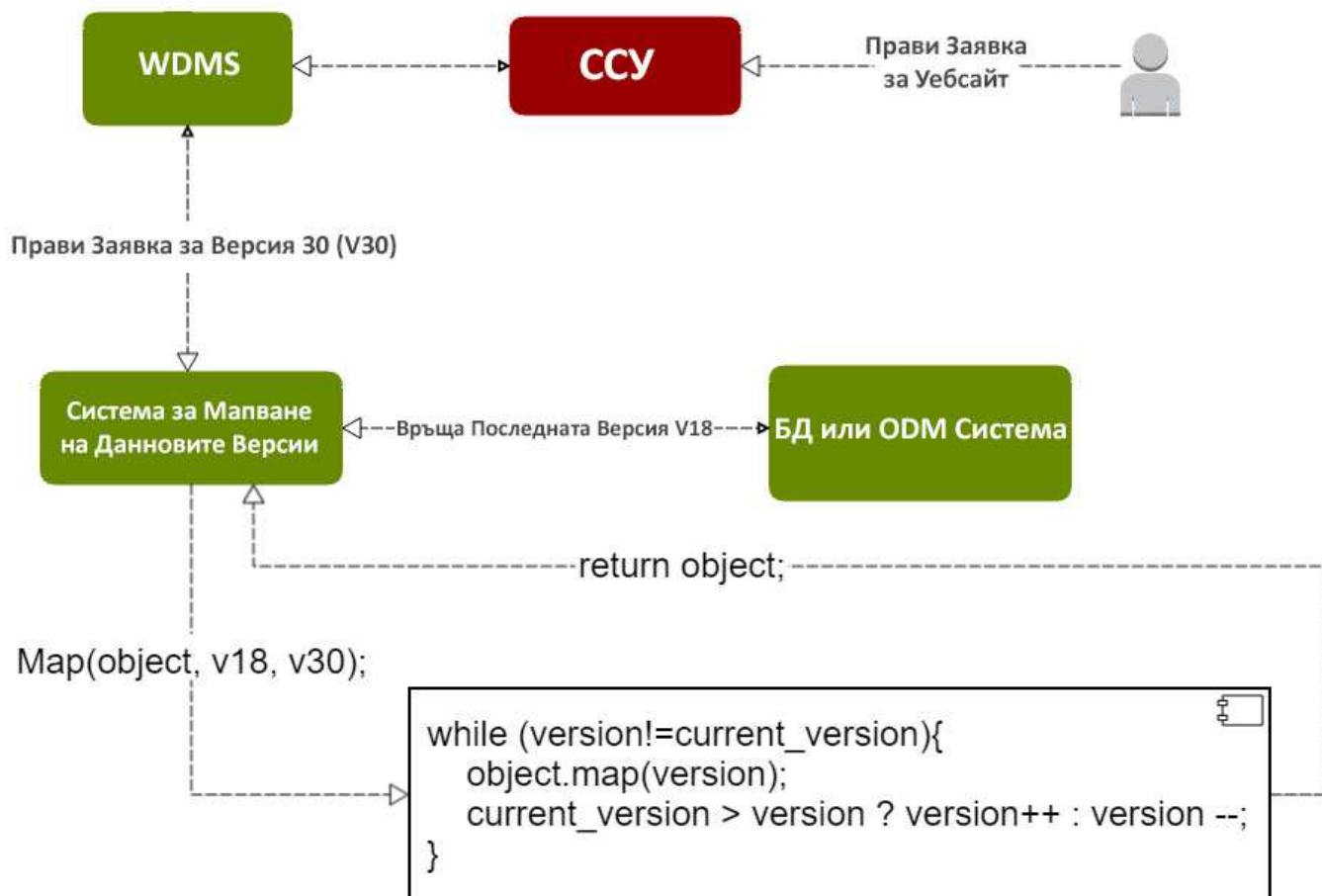
Методът за динамична миграция изисква първата версия на обектния мапинг клас да включва пълно описание на схемата и стойностите по подразбиране на всяко едно поле. Това помага при автоматизирането на подсистема за валидация на данните.



Фигура 16. Визуална репрезентация на версиите на примерен обект и обектен мапър клас за всяка от версиите му

Пример за реверсивната функция от Фигура 16, за промяна на име на поле, дадена за примера от Фигура 16, при която `field_name` е първото име на полето, `new_field_name` е новото име на полето, `revert_to_previous_version` обозначава, дали функцията се извиква в прав или обратен ред, т.е. когато се връща компонентът към по-стара версия:

```
function rename(field_name, new_field_name, revert_to_previous_version=false){
  !revert_to_previous_version ?
    this.field_name = new_field_name : this.field_name = field_name;      (2.4.3.1)
}
```



Фигура 17. Обща презентация на начина на работа на DVM системата

Фигура 17 показва концептуалния модел на архитектура на тази система, като част от CCY. При различните имплементации може да включва валидираща подсистема, която също може автоматично да се мигрира през описанието на мапър класовете на версиите.

Създаден е метод за динамично мапване на версии, който е в основата на системата.

- Система за оптимизация на снимки в CCY

Съвременните онлайн CCY са предимно SPA (Single Page Application), тоест приложения, работещи динамично на една страница. При създаване на системи за визуална редакция на уебсайтове, като SPA приложение, възникват редица проблеми за решаване. Представената система предлага решение за подобрене на потребителското преживяване (UX) във визуалната клиентска част на CCY.

При наличието на множество визуални елементи, анимации и ефекти, функциониращи в системата, дори и модерните браузъри изпитват затруднения да оперират леко при визуалното рендериране, както и при изпълнението на програмния код, който изпълнява анимирането и различните ефекти. Към всичко това се включат и тежки за рендериране и анимиране стилове, изискващи по-добро използване на ресурсите.

В частта с визуалното рендериране са идентифицирани няколко основни насоки, в които оптимизацията е задължителна, за да може приложението да оперира и при разрастване на системата. Текущо описаната система предлага решение на една от идентифицираните насоки за оптимизация. В системата за визуална редакция на една CCY се работи с множество снимки, като обикновено всички снимки от една страница се представят наведнъж, с възможности за промяна на снимката, стилизиране, позициониране, преоразмеряване, завъртане под ъгъл и др.

Системата предлага подход за оптимизация на графични изображения, което от своя страна оптимизира т.нар. изрисване на области в браузъра при различни операции, анимации, ефекти.

Текущо описаната система за оптимизация на снимки в CCY предлага един балансиран подход за оптимизиране на използването на сървърните ресурси за постигане на по-добро

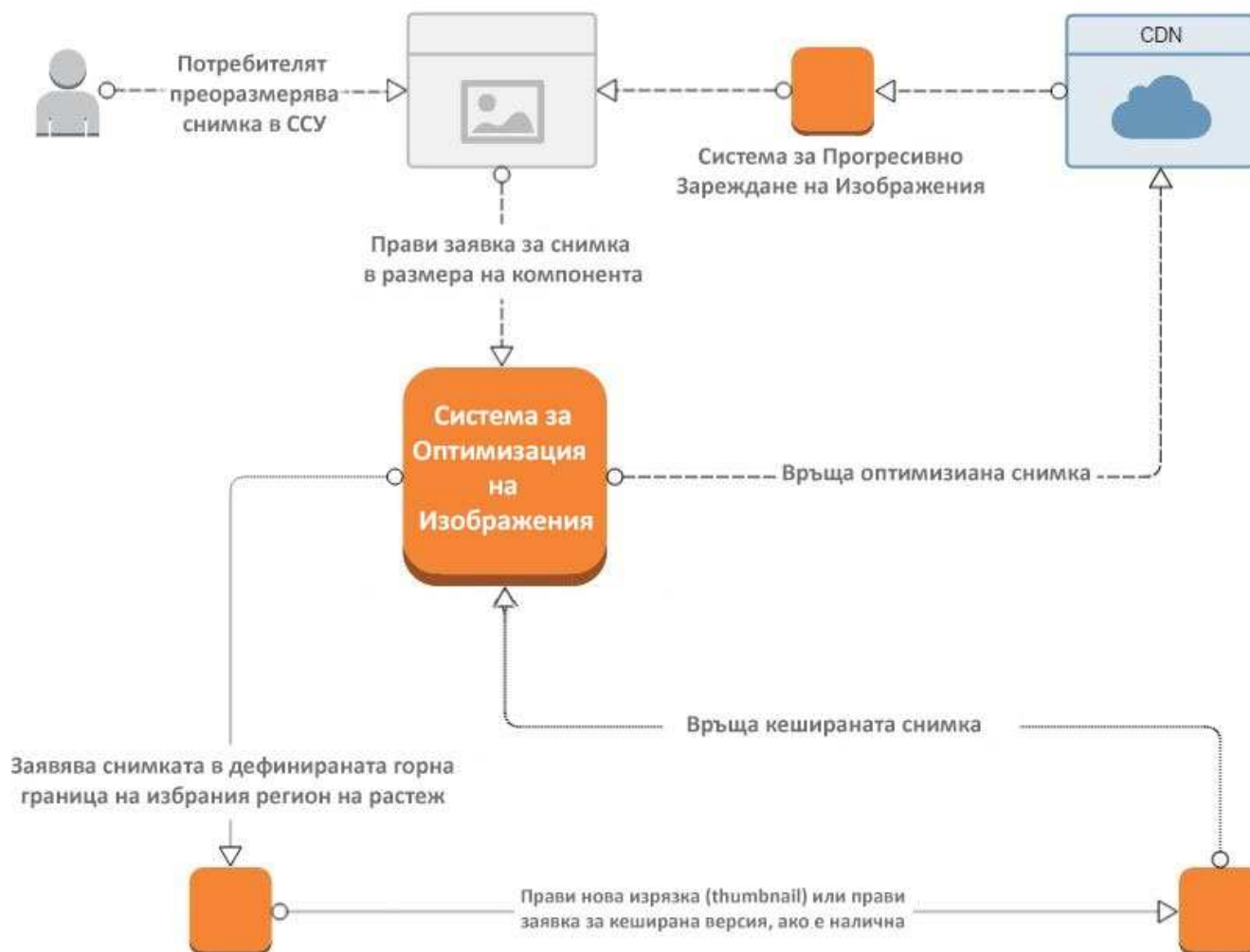
потребителско преживяване. Като краен резултат потребителят вижда по-бърза ССУ, снимки с максимално добро качество при всички размери както след, така и по време на самото оразмеряване.

1500x1500
1000x1000
500 x 500
200 x 200

Фигура 18. Региони на растеж на изображение при оразмеряване

Във Фигура 18 се представят региони на растеж на изображение при оразмеряване. Те могат динамично да се оптимизират спрямо нуждите на отделната система. Примерните региони са зададени спрямо типичните нужди при използването на една ССУ.

Според методите за обработка при имплементация на текущо описаната архитектура, регионите на растеж може да бъдат описани в двумерен масив, но също и като матрица. Броят на регионите на растеж може да бъде различен, като в този случай са предложени 4 региона, като синтезирано, оптимално решение за масово използване, базирано на публични статистики за най-използваните разделителни способности. При различните типове системи, тези региони може да бъдат оптимизирани допълнително при налични данни или хипотези за най-масово използване, като самата система може да бъде изградена автоматизирано да променя броя и размера на регионите, спрямо статистическите данни. Архитектурата на системата като част от предложени концептуален модел на архитектура на ССУ, е представена във Фигура 19.



Фигура 19. Архитектура и схема на работа на системата за оптимизация на снимки в ССУ

Фигура 20 и Фигура 21 демонстрират проблема във визуалния редактор на експерименталната ССУ, като при оразмеряване, за всеки един размер на компонента трябва да има подходящ размер на снимката, при който при всяка операция за увеличение на размера, снимката да е с най-добро качество, като в същото време е с най-оптимално работно представяне.



Фигура 20. Представа примерен компонент за снимка в ССУ, в първоначално състояние, разположен в най-малкия регион на растеж от 200 пиксела



Фигура 21. Демонстрира настройка на компонента на 100% ширина при адаптивен дизайн

■ Система за управление на зависимостите в ССУ

Тази система е основна система за оптимизираната работа на всички подсистеми в ССУ, особено за системата за визуално редактиране. Тази система не е централизирана, а се изпълнява от множество подсистеми в различните части от ССУ и се изпълнява на много нива.

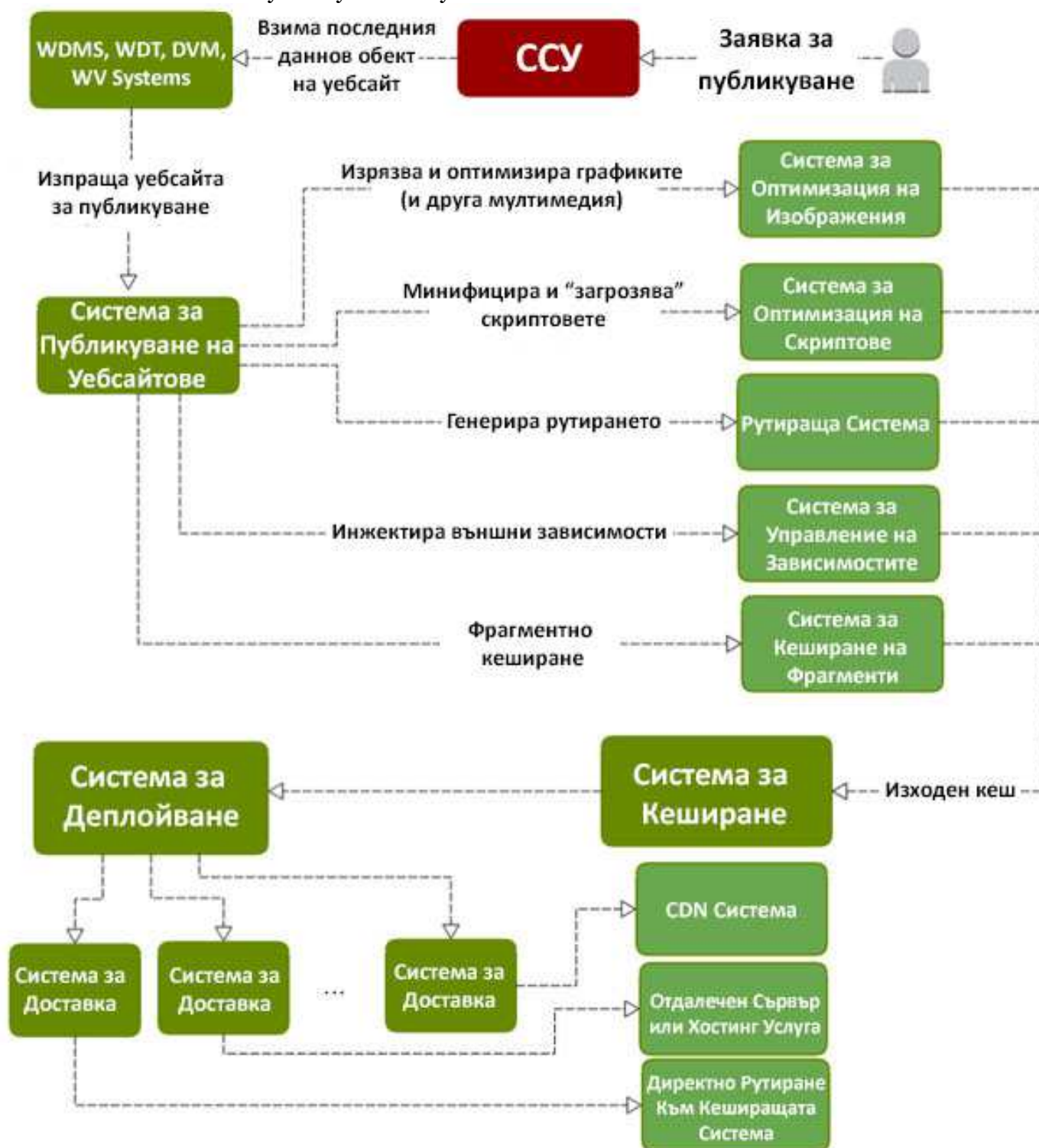
- Система за управление на уебсайт версиите и историята на промените в ССУ

Системата се използва за съхраняване на всяка една версия на уебсайт данните, като преди нанасяне на промените, се архивира предходната версия. Тази система може да се имплементира и като част от WDMS или като отделна система. Системата получава за вход уебсайт обект през уебсайт интерфейс, като създава копие, което се компресира и се съхранява в отделен архив в БД. Обикновено системата се извиква при всяко съхранение или публикуване на промени, след редакция на уебсайт и се използва като функционалност за история на уебсайта, която позволява да се възстанови уебсайтът към дадена версия от миналото.

- Система за рутиране в ССУ

Системата за рутиране е описана в различните конфигурации в предходната глава при четирите експериментални имплементации на ССУ.

- Система за онлайн публикуване на уебсайтове в ССУ



Фигура 22. Архитектура на система за онлайн публикуване на уебсайтове в ССУ

Системата, с концептуален модел на архитектура, описан във Фигура 22, управлява процеса на публикуване, използвайки свои подсистеми, като взема уебсайт обекта, който преди това е бил редактиран през системата за визуална редакция, трансформира го в изходен код (HTML, CSS, JS и т.н.) чрез система за кеширане, включва всички нужни библиотеки, според използваните компоненти чрез система за управление на зависимостите, оптимизира всички снимки и прави всичко налично онлайн.

- Система за динамично и умно кеширане в ССУ

Основно описана в експерименталните системи от Глава II, включващо съществуващите изследвания и технологии, предложеното умно инвалидиране на кеш, фрагментното кеширане и кеширането на динамични фрагменти.

- Система за управление на деплойването и доставката на уебсайтове в ССУ

Подсистемата за деплойване (дистанционно инсталиране) има задачата да деплойне кеширания уебсайт към подсистемата за доставка или до множество системи за доставка, ако това се изисква от архитектурата. Системите за доставка правят публичен уебсайта и достъпен до крайните потребители. Когато кешираният уебсайт се деплойне към система за доставка, може да се каже, че публикуването е завършено.

- Система за визуално редактиране в ССУ

Това е основна за ССУ система, освен от размерите си за имплементация, то и от гледна точка на факта, че това е системата, с която потребителят асоциира цялата ССУ. При изправност и оптимизация на всички останали системи, които работят на заден план, това е системата, която е фокус при продажбите и презентирането на ССУ. Това е и причината тази система да е от части уникална при различните ССУ, като някои основни подсистеми се припокриват, но обикновено с различен UI/UX. Потребителите реално работят директно с тази система, освен с маркетинговия уебсайт, през който се продава и навигира до самата ССУ.

При отваряне системата приема уебсайт обект за редакция, като разделя редактирането на набор от подсистеми, всяка - грижеща се за редакция на отделна част от уебсайт обекта. Например редакцията на страниците се осъществява от една система, а на компонентите - от друга система, която има функционалност за редакцията и визуалното рендериране на всеки един вид компоненти. При запазване отново се изпраща обратно към съпътстващите системи на ССУ вече промененият даннов обект на уебсайта.

- Система за управление на страници, контейнери и групите в ССУ

Когато се зареди уебсайт в системата, данните за страниците му се предават към системата за управление на страници, която предоставя интерфейс за редакция, преподреждане, добавяне, триене и т.н. След всяка една операция, например по време на редакция на име на страница, данновият обект на страницата е обвързан с начините на редакция, като по време на самото писане се изменя и самият обект на страницата.

След това компонентите от страницата, която се редактира в момента, се подават към системата за управление на редакцията на компоненти, която на свой ред подава паралелно всеки един компонент към съответните им управляващи функционалности, които се грижат за тяхното рендериране на страницата, показване на интерфейсите за редакция при конкретни събития, както и самата редакция на данновите обекти, които стоят зад компонентите.

Тази система се грижи и за операциите за смяна на страница, както и за подаването на обектите от текущата страница към системата за управление на компоненти. Също така, тя се грижи и за управлението на групите и другите видове контейнери, които може да съдържат компоненти.

- Система за управление на компоненти, редакция, репрезентация, стилизиране и настройки на визуални елементи в ССУ

Системата се грижи за визуалното рендериране на всеки компонент, като на базата на основното представяне на компонента постъпково, се прилагат съдържанието, стилизирането и настройките му.

Всеки компонент има тип, като за всеки тип има изградена уникална логика. Всеки компонент има уникални изглед и интерфейс/и и логика за редакция, следващи стандарта за UI/UX на цялата ССУ. Чрез интерфейсите за редакция се редактират различни части от данновия обект, стоящ зад компонента. Например чрез интерфейс за

оразмеряване, чрез дърпане на визуални стрелки, които са изрисувани около компонента, се променят стиловете му в частта с размерите му. Също така, например при текстовия компонент, при редакция, текстът му се отваря в текстов редактор, като TinyMCE или CKEditor (двата най-популярни уеб WYSIWYG редактора на текст).

- Система за управление на действията в системата за визуално редактиране в ССУ

Това е система, позволяваща премахване на извършени корекции и навигиране между действията (undo-redo). Това е система от изключителна важност за ССУ при прилагане на система за трансформиране на уебсайт обекти при запазване на уебсайта. Когато двете системи работят заедно единственото нещо, което се запазва, е изходът от системата за управление на действията. Тя изпраща извършените действия със съответните им трансформиращи функции и данните към тях.

Например, при преместване на компонент се създава действие `WidgetChangePosition`, което има функция, която изпълнява промяната в данния обект на компонента, като се запазват предишните стойности на променените полета в самото създадено действие. Трансформиращата функция е реверсивна, и например при извикване на `Undo` се възстановяват старите стойности, като новите се запазват в действието, и, ако последва операция `Redo`, отново се прилага действието.

Тази система имплементира широко познатия Команден Шаблон за софтуерен дизайн (Command Pattern). Самият шаблон не е новост, като съществуват множество имплементации и изследвания в посоката, затова не е включено детайлно описание към настоящия дисертационен труд.

Новост е обаче описание на приложението на подобна система в ССУ и обвързването ѝ със система за трансформации при оптимизиране на запазването на уебсайт след редакция в ССУ.

- Система за създаване и управление на мобилен/адаптивен изглед в ССУ

За управление на мобилен/адаптивен дизайн също се грижи отделна система, която реално прилага допълнителни версии на компонентите и страниците, които могат да се изменят за всяка една резолюция. В практическата имплементация на архитектурата се използва система с мобилен изглед, като на същия принцип може и потребителски да се дефинират отделни точки на промяна.

- Система за управление на езикови версии на уебсайтове в ССУ

При някои ССУ се предлага система за създаване на многоезични уебсайтове. На практика тази система е много сходна функционално със системата за адаптивен дизайн по това, че и тя създава допълнителни версии на компонентите, но в случая става дума само в частта на съдържанието им.

Тази система има интерфейс за смяна на моментно редактирания език, като той извиква пререндериране на страницата по стандартен начин през системата за управление на страници. Всеки компонент, обаче, зарежда съдържанието си на конкретната езикова версия. Съответно при редакция се редактира само тя.

Тази система също може да се обвърже с активатор през системата за рутиране. Например `bg.mydomain.com` или `mydomain.com/bg..`

- Система за автоматизирано SSL криптиране на уебсайтове в ССУ

Описаното решение е насочено за имплементация на масови ССУ с милиони уебсайтове, като за различни имплементации може да бъдат проектирани по-опростени решения.

Проблемът при SSL в ССУ идва от факта, че на една инфраструктура трябва да се инсталират милиони SSL сертификати, като това да не намалява производителността. При стандартна инсталация на SSL сертификати, те се съхраняват във файлове, като се прави асоциация към конкретните файлове от уеб сървъра. Когато една ССУ е балансирана на стотици уеб сървъри, то тези SSL-и трябва да се инсталират и подновяват на всеки един от тях. А когато станат много на брой, то това понижава и дори може да спре работоспособността на уеб сървърите. Решението на проблема е изнасянето на SSL сертификатите в отделна централизирана система, която се грижи да подава нужния

сертификат на всеки уеб сървър, който го поиска. Тази система също може да се балансира на много сървъри и да се разширява.

2. Изводи

Всяка от тези системи разрешава различен проблем и включва в себе си набор от различни подсистеми. Архитектурите разглеждат всяка система като логически затворена, с определен вход и изход. По този начин се позволява препроектиране и подобрене на концептуалния модел и въвеждане на допълнителна сложност за разрешаване на нови проблеми.

Комплексните системи, като ССУ са изградени от множество подсистеми, които с разрастването на ССУ стават с по-комплексна йерархия от системи. Всяка от тях има специфична задача, теоретично капсулирана логика, като на теория може да се разглеждат като логически затворени системи в изграждането на архитектурата на ССУ. Описаната архитектура на ССУ е комплексна съвкупност от архитектурите на описаните подсистеми, включваща начина им на взаимосвързване и съвместно опериране.

За голяма част от подсистемите са създадени авторски методи за справяне с конкретни проблеми, за които са създадени и съответните системи.

Към всяка от описаните подсистеми е проведено проучване на технологии в областта, описани са изследвания, технологии, подходи и методи свързани с теоретичното проектиране на всяка конкретно предложена подсистема. Системи, или части от системи, които не представляват научна новост и вече съществуват достатъчно изследвания за тях, не са разгледани. Някои от системите, като тези свързани с рутирането и кеширането, са вече разгледани и развити в Глава II, като в текущата глава са отнесени само като референции към съществуващото предходно описание.

Глава IV: ПРАКТИЧЕСКА ИМПЛЕМЕНТАЦИЯ И РЕАЛИЗАЦИЯ НА ПРЕДЛОЖЕНАТА АРХИТЕКТУРА НА ССУ

Проектирането, изследванията и разработките са проведени в период от над 4 години от Февруари 2013 г. до момента на писане и защита на настоящия дисертационен труд.

Проектът е първоначално финансиран на фаза концепция и получи последващо финансиране на фаза доказване на концепцията с общо над 1 милион евро от международен инвестиционен фонд, като в последствие компанията зад проекта Oxxu Group PLC е трансформирана в публична компания, като е листната за търгуване на Норвежката фондова борса в Осло.

Системата прилага на практика архитектурата, предложена в Глава III. Системата може да се класифицира, според класификациите дефинирани в Глава I, като ССУ насочена към масовата аудитория, система с фрагментно публикуване, абсолютно позициониране на визуалния редактор, WYSIWYG визуален редактор, визуално редактиране за адаптивен дизайн.

Като за система с тези класификации, предложената архитектура прилага неограничена разширяемост на всяка една подсистема. Чисто инфраструктурно тази системна разширяемост се имплементира при растежа на ССУ, като в самото начало само определени системи може да се мащабират автоматично, а други, които не са толкова натоварени, да са само на един сървър. Като при растежа на системата се стартира и автоматичното скалиране на всяка растяща подсистема.

За да се даде представа за реалните размери на практическата реализация на експерименталната ССУ, разгледана като собствен софтуерен код надминаващ 5 млн. реда, като в това число не са включени библиотеки или външно вградени готови проекти, с които системата надминава 10 млн. реда код.

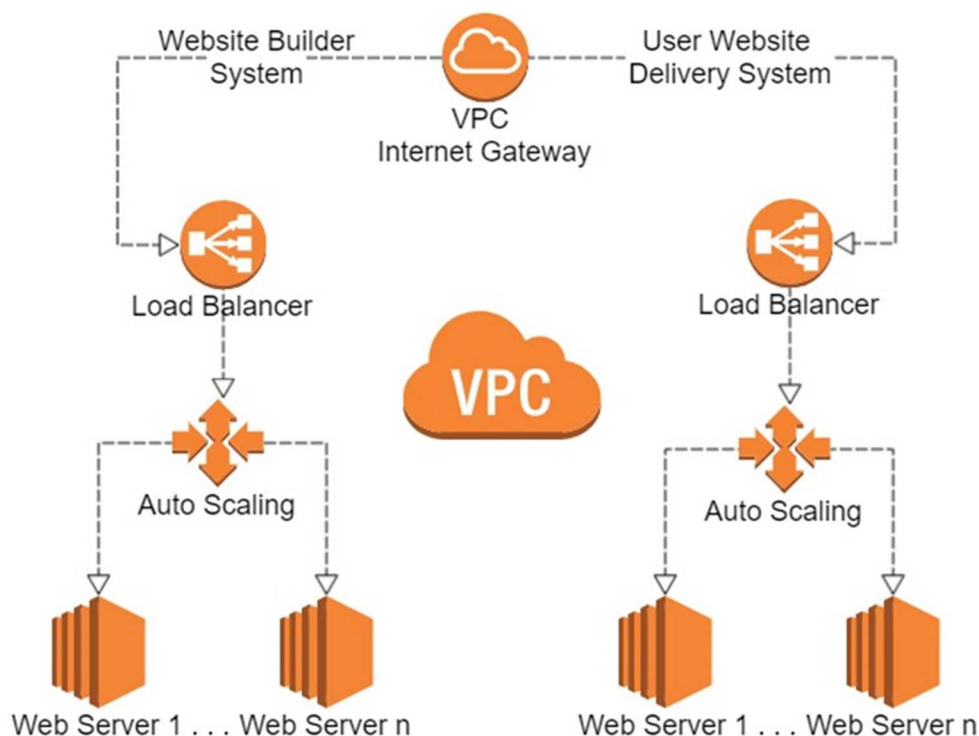
Описаните логически разделени системи от Глава III са резултат от теоретични изследвания, както и от емпирични доказателства на хипотези, чрез практическата им реализация в този експериментален проект, която допринесе за оптимизирането им.

Глава IV в същността си представлява допълнително описание на предложената архитектура от Глава III в две части. Първата е проектирането на сървърната архитектура и визуалният редактор. Във втората част се демонстрира начинът на имплементация на описаните системи, имайки предвид, че за всяка система може да се избере уникален начин за имплементация, съобразен със специфичните нужди на ССУ.

Системата е достъпна свободно на домейна oxhu.com. Позволява безплатен абонамент със свободен достъп, с който може да се разгледа и използва от всеки заинтересован.

1. Архитектура на сървърната инфраструктура

За целите на експерименталната система е изграден т.нар. виртуален частен облак – Virtual Private Cloud (VPC) в Amazon Web Services (AWS).



Фигура 23. Архитектура на БД инфраструктура, вкл. и кеширащата система

Фигура 23 демонстрира обобщената сървърна инфраструктура. ССУ е изградена изцяло в затворен VPC, с достъп през т.нар. Internet Gateway. Той открива към света само двата входящи разпределители на натоварването (LB). Единият е за визуалния редактор и останалата част от публично достъпния интерфейс на ССУ, а вторият е към системата за доставяне на уебсайтове, която в този пример е изградена като част от ССУ, като на потребителите се дава възможност да използват и външна система за доставяне.

Демонстриран е начинът на защита и капсулиране на инфраструктурата. Връзката между инфраструктурата на различните подсистеми става изцяло във вътрешна мрежа (VPC), без външен достъп.

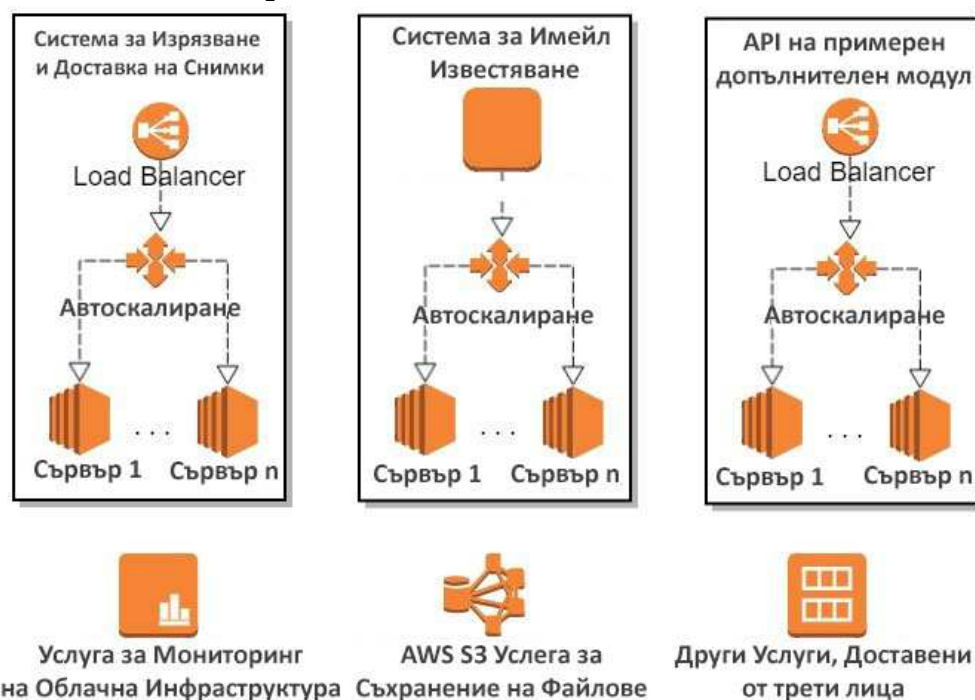
Ако в частност вземем входящия трафик към ССУ, LB разпределя трафика към набор от еднотипни уеб сървъри. Система за автоматично скалиране следи натовареността на уеб сървърите, като при нужда стартира нови сървъри автоматично. Това обикновено се случва в пикови часове, като също така се грижи за обратното скалиране и намалява сървърите, когато няма нужда от тях. Например, във вечерно време може да има само 1 уеб сървър, а през деня да бъдат 100.



Фигура 24. Архитектура на БД инфраструктура, вкл. и кеширащата система

Подобен е и принципът при автоматичното скалиране на БД и кеширащите системи, показани на Фигура 24. При тях обаче не се използва стандартен LB, а те се създават чрез стандартни за тях репликация и, съответно - чрез клъстеризация, за които стана въпрос в Глава II. Уеб сървърите се характеризират като „Stateless“, т.е. без текущо състояние и възможност за промяна на състоянието, като всичко свързано с текущо състояние, е изнесено на отделен сървър или услуга. Например сесиите на потребителите са изнесени в кешираща система, в случая Memcached. Скалирането при самите бази данни е много по-тромаво от веб сървърите, защото те се скалират на базата на имидж на текущото състояние, което обаче до момента на създаването на нова инстанция не е актуално и отнема процес на актуализиране. Затова при тях се прилага по-забавен метод за скалиране, който предполага по-плавен растеж и смаляване.

Експерименталната ССУ използва множество различни видове БД, всяка с различно предназначение. NoSql БД е една от най-основните, като е използвана MongoDB, поради най-голямата близост до начина на съхранение на данните в JSON. MySQL и PostgreSQL са сходни като работа, но за различни части от архитектурата са избрани различни качества на всяка, при които се представят по-добре. Системата за доставка на SSL сертификати доставя данните за сертификатите към веб сървърите. По този начин е възможно скалирането на данните за сертификати, като веб сървърът намира мигновено конкретния сертификат. При стандартния начин сертификатът трябва да се складира на веб сървъра, но при автоматичното скалиране това прави веб сървъра променлив, а не без статут (Stateless), което нарушава архитектурата, нужна за автоматичното скалиране.



Фигура 25. Архитектура на инфраструктурата от прилежащи услуги на ССУ, без директен достъп от потребителите.

Във Фигура 25 е демонстриран начинът на скалиране на различните видове прилежащи системи. Това например може да е подсистема, като тази за изрязване на снимки. Системата отново има една входяща точка, която обаче не е публично достъпна, а се достъпва само от веб сървъра през вътрешната мрежа. Тя се скалира автоматично на същия принцип, на който се скалират веб сървърите, като използва LB за разпределение на натоварването.

Демонстрирана е и система за обработка на задачи. За разлика от предходната, при този тип системи се работи с обемисти задачи, отнемащи ресурси. В примера е дадена система за изпращане на имейл нотификации или електронни бюлетени. Когато ССУ има милиони

потребители и в един момент трябва всички те да получат имейл, тази задача се подава на Контролера, който управлява процеса, като той също така може да играе роля на LB.

Към тази секция са описани също така и външни услуги, които се използват към инфраструктурата. Това например са услуги за следене на стабилността на сървърите, системи за складиране на файлове AWS S3, DNS системи, като AWS Route 53, CDN системи, като Cloudflare и други.

3. Основни стъпки при създаване и публикуване на уебсайт в ССУ

- 1) Влизане на основната входна точка на ССУ, която при описаната имплементация е уебсайтът <https://www.oxxy.com>;
- 2) Регистрация и/или автентикация в системата;
- 3) Избор на първоначален дизайн (заготовка) на уебсайта от списъка с готови дизайни, които предлага ССУ;
- 4) Редакция на структура, съдържание, дизайн, чрез визуален редактор;
- 5) (Опционално) Активиране и редакция на мобилен/адаптивен изглед;
- 6) (Опционално) Купуване/Свързване на домейн име (или използване на поддомейн на ССУ), добавяне на фирмена електронна поща;
- 7) (Опционално) Създаване на електронен магазин;
- 8) Публикуване на уебсайта.

Домейнът е моментално активен и всички промени са достъпни.

Към момента на писане на дисертацията на експерименталната ССУ, тя има десетки хиляди уникални потребители, създали и публикували поне по един уебсайт. Направени са експериментални уебсайтове за демонстрация, първият е за услуги за създаване на електронни магазини <https://www.magazinonline.com>, вторият е многоезичен, използван за маркетингов уебсайт на платформа за управление на резервации за заведения <https://www.reservation.tools>.

4. Изводи

Голяма част от описаните подсистеми на ССУ са подобрени значително, в следствие на експерименталната реализация, описана в текущата глава. Свободният достъп до експерименталната система <https://www.oxxy.com> позволява на всеки, заинтересован от изследванията в сферата, да разгледа как теоретичната архитектура, описана в Глава III и частично някои подсистеми в Глава II, е имплементирана и работи в една динамична среда, използвана към момента на писане на настоящия дисертационен труд от потребители от над 100 държави. При първичния вариант на концептуалния модел на архитектура бяха заложени множество хипотези, които при реализацията бяха опровергавани при конкретните експерименти, което допринесе за подобряване на предложената архитектура до варианта, в който е предложена в настоящия дисертационен труд.

Въпреки, че експерименталната реализация изискваше години разработки и изследвания, значителен финансов ресурс и екип от специалисти, тя допринесе за доказване на реализуемостта на предложения концептуален модел на архитектура. В следствие на експерименталната работа, се подобри този модел в частите му, при които предложени хипотези бяха нереализуеми. Също така, реалната експлоатация на системата позволи анализиране и дефиниране на конкретни слаби аспекти на архитектурата и показва необходимост от въвеждането на допълнителни подсистеми, които могат да увеличат нейната сложност, но решават проблеми, които неизбежно трябва да бъдат адресирани при бързо разрастване на една ССУ, работеща в реална среда.

ЗАКЛЮЧЕНИЕ

Дисертационният труд цели да даде светлина в сферата на ССУ, като разкрие цялостен концептуален модел на архитектура за изграждане на един от най-тежките класове ССУ. От започването на дисертацията до момента на завършването ѝ, нишата е развита значително, като много световни ССУ към този момент се използват от десетки милиони потребители всяка, като най-големите достигат до над 100 милиона потребители и над няколко милиона платени уебсайта всяка.

Въпреки това, публикуваните изследвания в областта са оскъдни, като се публикуват основно под формата на патенти. Тези патенти не дават светлина върху текущото развитие

на технологиите, а разкриват отминали такива, които технологичните корпорации искат да използват, като свое патентно право, за да се защитават срещу нови конкуренти, които тепърва ще минават през процеса на технологично развитие на своите ССУ.

Предложените 4 различни концептуални модела на архитектура на ССУ в Глава II са фокусирани върху специфичните части на различните видове ССУ, които хвърлят светлина върху конкретни основни аспекти от цялостната архитектура на ССУ системите.

В Глави III и IV се разглежда основната цел на дисертационния труд, концептуален модел на архитектура на ССУ за масовият потребител, както и успешната практическа реализация на предложената архитектура.

Изследваните над 100 ССУ, с обобщение описано в Глава I, показват че всички системи минават през процес на техническо надграждане, като всяка една от тях е срещала проблема с липсата на публични изследвания в тази посока. В следствие на това, всяка от тях е инвестирала време и финанси, които в конкурентна среда често достигат до неуспех.

Основна цел на дисертацията е да послужи за полагане на основа за бъдещо развитие на науката и практиката в тази насока. Тя следва да послужи за стартова отправна точка за създаване на нови ССУ и за оптимизиране на вече съществуващи ССУ, като към писането на настоящия дисертационен труд няма публично цялостно изследване на ССУ архитектура.

Приложени към дисертацията са:

1. Списък с изследваните ССУ;
2. Проучване за степента на пазарно проникване на ССУ бранша на Шведския пазар;
3. Четири референции от инвеститори за финансиране на проучванията и проектирането на 4те концептуални архитектури на ССУ от базови класове, както и на създаването на експерименталните системи към тях, описани в Гл. II точки 1 до 4;
4. Референция от инвеститор за финансиране на проучванията по основна теория в дисертацията, а именно - предложеният концептуален модел на архитектура в Глава III, както и за експерименталната ѝ реализация, описана в Глава IV;

ПРИНОСИ

1. Предложен, проектиран и експериментално реализиран е нов концептуален модел на архитектура (КМА) на Система за Създаване на Уебсайтове (ССУ) от клас с най-висока сложност спрямо синтезираната класификация. Това е първият публикуван концептуален модел на цялостна архитектура на ССУ.
2. Предложени, проектирани и експериментално реализирани са КМА на 17 авторски подсистеми за нуждите на предложения нов КМА на ССУ. Системите разрешават основните си проблеми, чрез предложени авторски методи за всяка от тях.
3. Предложена е класификация на ССУ, синтезирана на базата на над 100 изследвани, реални ССУ. Тя е съставена от 15 класа в 5 категории. Класификацията служи за определяне на нужното ниво на сложност при моделиране на архитектури за ССУ.
4. Създадени и експериментално доказани са три метода за конвертиране на съществуваща софтуерна система от еднонаемателна към многонаемателна архитектура.
5. Предложени са три статистически модела за предсказване на тенденцията за повишаване на нуждите на ССУ, базирани на бъдещото повишаване на интернет проникването и потреблението.
6. Предложени, проектирани и експериментално реализирани са 4 концептуални модела на ССУ архитектури, от различни по-прости за реализация класове.
7. Изследвани и емпирично доказани са методи и аспекти на оптимизация, мащабиране и опериране на ССУ, като е предложен подбор от методи с най-висока ефективност спрямо изследваните, съпоставяйки резултатите с нужната инвестиция за реализацията им.
8. Изследвана и емпирично доказана е ефективността на прилагането на методи за стандартизация чрез интеграция на ССУ, както и ползите от изграждане на тематични онлайн портали с основа ССУ.
9. Проведени са изследвания на потребителското поведение и пазарно проучване, целящи да изследват проникването на ССУ в целевата групата на масовия потребител, към която е насочена предложената архитектура. Резултатите са публикувани в приложение към дисертацията.

СПИСЪК С ПУБЛИКАЦИИТЕ СВЪРЗАНИ С ДИСЕРТАЦИОННИЯТ ТРУД

1. М. М. Митев, Е. В. Рачева, Д. З. Димитров: Применение текстометрических исследований при разработки специализированных интерпретаторов, Управляющие Системы и Машины, Национальная академия наук Украины, стр. 78-81, 2011
2. Dimitar Dimitrov: Software Automation Algorithms for user interface development, Международная конференция „Стратегия качества в промышленности и образовании”, р. 379-382, 2012
3. Димитър Димитров: Изграждане на тематични онлайн портали на базата на многосайтова система, Трети Международен Научен Конгрес, Технически Университет - Варна, vol.1, р. 139-142, 2012
4. Димитър Димитров: Оптимизация на многосайтови софтуерни системи, Трети Международен Научен Конгрес, Технически Университет - Варна, vol.1, р. 143-146, 2012
5. Dimitar Dimitrov: Standardization based on multiwebsite system integration, р. 1117-1122, International Science and Technology Conference (ISTEC) 2012, Dubai
6. Dimitar Dimitrov: System for website data management in a website building system, CompSysTech'2017 Proceedings, ACM ICPS, vol.1396, doi>10.1145/3134302.3134313, р. 211-218, 2017
7. Dimitar Dimitrov, Elena Racheva: System and method for dynamic data version mapping in a website building system, Computer Science and Technologies Journal, Technical University – Varna, №1/2017, р.77-82, 2017
8. Dimitar Dimitrov: A system for optimization of images in a website building system, Fifth Scientific International Conference for Computer Sciences and Engineering, vol.1, р. 112-119, Varna, 2018
9. Dimitar Dimitrov: A system for website publishing in a website building system, Fifth Scientific International Conference for Computer Sciences and Engineering, vol.1, р. 104-111, Varna, 2018

БЛАГОДАРНОСТИ

За ценното ръководене в процеса на писане на текущата дисертация, както и цялостната помощ за успешното завършване на докторантурата, бих искал да изкажа благодарности на моя научен ръководител доц. д-р Елена Рачева.

Въпреки, че научните изследвания, проектираната архитектура и описаните в настоящия дисертационен труд научно-приложни приноси са изцяло труд на автора, за практическата реализация на подобна архитектура е нужно немалко финансиране и екип от специалисти.

Инвестицията в проектиране на текущо предложената от Глава III архитектура до този момент е над 1 млн. евро, подсигурана от международен инвеститор. Имплементацията на научните изследвания по проектирането на архитектурата се подпомага от международен екип от няколко европейски държави (България, Норвегия, Швеция, Кипър, Малта, Люксембург, Англия). Дейностите по развитието на изследването и практическата му реализация са финансирани чрез публичната компания Охху Group PLC, листната на Норвежката фондова борса (Oslo Børs) и, в момента на писане на настоящия дисертационен труд, е с пазарна капитализация (Market Capitalization) от 312 млн. NOK или приблизително 31 млн. евро, като е в процес на вторично листване и на Виенската фондова борса.

Предходно на предложената система, финансирането за изследванията и разработките по ССУ от Глава II са направени във времето от различни заинтересовани български и международни компании. За проектиране и реализиране на експерименталните архитектури, обобщени в тази глава, бяха подсигурени инвестиции между 20 и 100 хиляди евро на ССУ.

RESEARCH, DESIGN AND DEVELOPMENT OF NEW ARCHITECTURES FOR WEBSITE BUILDING SYSTEMS

ABSTRACT

The field of research in this dissertation are the online website building systems (WBS), online visual content editing and online publishing tools. The main objective of this dissertation is to enlighten the field of WBS by publishing a complete conceptual model of a system architecture for developing one of the most complex classes of WBS. From the initiation of this dissertation till the moment of its completion, the niche has developed drastically, and many international WBS are being used by tens of millions of users each, and the biggest are used by over 100 million users. Despite this rapid development, the published research in the field is scarce, mainly in the form of patents. On top of that, the patents reveal obsolete technologies, and show no data on contemporary technologies, methods or system architectures, mainly used by companies to enrich their intellectual property actives, for being aggressive legally to competition, that still uses this technology. The current dissertation aims to be a base of future public researches in the field of WBS.

Chapter I describes a research of the technologies and system architecture of more than 100 existing WBS, existing research of supporting technologies, that can help reach the goal of this dissertation. A research resulted in synthesized classification of WBS by main characteristics, important to defining WBS system complexity, in the stage of designing a new system. A research on internet penetration and usage has been conducted and statistical models for predicting future trends have been suggested. A Swedish market research has been conducted to define the WBS penetration level.

Chapter II describes designed and developed experimental WBS in 4 different classes. All systems are used in live production environments. The research and development for the systems were funded by different Bulgarian and international companies. The goal of the conducted research and experiments described in this chapter is to clarify the issues, which a WBS architecture must solve, based on its class and purpose. The results, applicability, implementation of the initial research on WBS. The strong and weak sides of the different approaches for WBS architecture design and development are evaluated, based on the different classification.

Chapter III describes the main goal of the dissertation. A system architecture for online WBS has been designed, classified as targeting the mass user. The research has been conducted in over 4 years of initial research and development of the different classes of WBS architectures, described in Chapter II, and 4 more years in experimental development, further described in Chapter IV, based on the architecture described in this chapter. A total period of over 8 years of theoretical research and experimental development, leading to improvement of the currently presented WBS architecture.

Chapter IV describes the results of the experimental implementation of the proposed conceptual model of a WBS architecture in Chapter III, the major issues faced in the design and development process. The process of designing a WBS from a theoretical hypothesis and scientific research to real life implementation is described. The research of the proposed system architecture and the currently described implemented system are funded by an international company with currently over 1 million euro. The system is live and running in production, with a free access to use by everyone. The experimental WBS system is currently being used by users of over 100 countries, creating thousands of websites daily.