



# **ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**Факултет Компютърни системи и технологии  
Катедра Информационни технологии в индустрията**

**Маг. инж. Пламен Огнянов Наков**

## **МЕТОДИ И АЛГОРИТМИ ЗА АВТОМАТИЗИРАНЕ НА ПЛАНИРАНЕТО НА РЕСУРСИ В ИТ ПРОЕКТИ (МОДЕЛИРАНЕ НА ИНФОРМАЦИОННО- ТЕХНОЛОГИЧНИ РЕСУРСИ И АЛГОРИТМИ ЗА АВТОМАТИЗИРАНАТА ИМ ОБРАБОТКА)**

### **А В Т О Р Е Ф Е Р А Т**

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

Област: 5. Технически науки

Професионално направление: 5.3. Комуникационна и компютърна техника

Научна специалност: Автоматизирани системи за обработка на  
информация и управление

**Научен ръководител: Проф. д-р инж. Румен Трифонов**

СОФИЯ, 2023 г.

Дисертационният труд е обсъден и насочен за защита от Катедрения съвет на катедра „Информационни технологии в индустрията“ към Факултет Компютърни системи и технологии на ТУ-София на редовно заседание, проведено на 30.01.2023 г..

Публичната защита на дисертационния труд ще се състои на 02.05.2023 г. от 13.00 часа в Конферентната зала на БИЦ на Технически университет – София на открито заседание на научното жури, определено със заповед № ОЖ-5.3-23 от 08.02.2023 г. на Ректора на ТУ-София в състав:

1. Проф. д-р инж. Даниела Асенова Гоцева – председател
2. Доц. д-р инж. Даниела Велева Минковска – научен секретар
3. Проф. д-р инж. Стойчо Димитров Стойчев
4. Проф. д-р инж. Десислава Георгиева Петрова-Антонова
5. Доц. д-р Йорданка Найденова Анастасова

Материалите по защитата са на разположение на интересуващите се в канцеларията на Факултет Компютърни системи и технологии на ТУ-София, блок 1, кабинет № 1443-А.

Дисертантът е задочен докторант към катедра „Информационни технологии в индустрията“ на факултет Компютърни системи и технологии. Изследванията по дисертационната разработка са направени от автора, като някои от тях са подкрепени от научноизследователски проекти.

Автор: маг. инж. Пламен Наков

Заглавие: Изследване на Методи и алгоритми за автоматизиране на планирането на ресурси в ИТ проекти (Моделиране на информационно-технологични ресурси и алгоритми за автоматизираната им обработка)

Тираж: 30 броя

Отпечатано в ИПК на Технически университет – София

## **I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД**

### **Актуалност на проблема**

Едно от основните предизвикателства, които дигиталната трансформация поставя е дефинирането на правилния подход, чрез който ефективно да бъде извършена.

Навярно, към днешна дата, не съществуват ефективни бизнеси, които да не са преминали през процеси по дигитална трансформация. Някои организации преминават дори многократно през такива процеси, като целят да отстранят допуснати грешки, да отразят различни „научени уроци“ или да оптимизират дигиталните процеси.

Изграждането на правилен модел на данните на информационно-технологични (ИТ) ресурси е ключова предпоставка за осигуряване на модерна и ефективна поддръжка, създаване и доставяне на стойност, реализиране на разпределение във веригите на стойността (value chain), осигуряване на подобро качество, прозрачност, по-добро възприятие (insights), автоматизация и оптимизация на разходите.

Целта на дисертационния труд е да се изследват, проектират и оценят експериментално проблемите, пред които ИТ е изправено по отношение структурирането на данните и изграждането на правилен модел за съхранение и обработка на информация за ИТ ресурси, да се оптимизира работата с ИТ ресурсите, да се анализират различните ситуации и състояния, в които бизнеса бива поставен, да се обобщят и на база получените резултатите да бъдат създадени надеждни методи и алгоритми, чрез които автоматизирано и навременно да се постигне оптимизация, предотвратяване на дефекти и ефективен анализ на текущото състояние.

За изпълнението на целта на дисертационния труд са проучени и изследвани подходи, методи и алгоритми за управление на ресурси, алгоритми за оптимално разпределение.

### **Цел на дисертационния труд, основни задачи и методи за изследване**

Цел на дисертацията е създаването на стратегии и методологии за разработване на интерактивен дизайн базиран на потребителско поведение за мултифункционални приложения.

Основна цел пред дисертационния труд е предоставянето на модел за структуриране на данни и алгоритъм, които да позволят на бизнеса изграждането на правилен модел на данните на ИТ ресурси, осигуряване

на възможност за модерна и ефективна поддръжка, създаване и доставяне на стойност, реализиране на разпределение във веригите на стойността (value chain), осигуряване на подобро качество, прозрачност, по-добро възприятие (insights), автоматизация и оптимизация на разходите.

В резултат на извършения качествен и функционален анализ, пред дисертационния труд се формираха следните задачи:

1. Изграждане на модел на данни на информационно-технологични (ИТ) ресурси - материални активи, апаратни средства, софтуерно-технологични решения, приложен софтуер, бизнес логика, ИТ и бизнес услуги. Описание на йерархията и взаимовръзките между тях.
2. Създаване на алгоритъм с елементи на експертна среда за оптимално, според зададени критерии, локализиране на дефекти в описаната йерархична структура.
3. Имплементация на алгоритъма в програмна среда за управление и устойчивост на системата от ресурси
4. Валидиране на създадения концептуален модел

### **Научна новост**

В рамките на труда е предложен концептуален архитектурен модел за изграждане на база от данни на конфигурационни елементи в една организация, който модел да позволява достатъчно ниво на детайлност, позволяващо поддържането на CMDB с висок интегритет на данните. Дефинирани са основните класове и обекти, които предлагания концептуален модел следва да съдържа, взаимовръзките между тях и поддържаните атрибути.

Предложен е и алгоритъм, който извършва обработка и анализ на данните в предложения концептуален модел и спомага за бързото и ефективно идентифициране на свързани с дефектирал конфигурационен елемент обекти, с което да се оптимизира времето за идентифициране на проблем или промяна предизвикала даден дефект. Извършени са експериментални изследвания за работата на алгоритъма с извадка от реални данни, като са анализирани крайните резултатите, времето за обработка и въздействието на алгоритъма върху производителността на избрана софтуерна платформа.

### **Практическа приложимост**

Предложения модел за структуриране на данни и алгоритъм има основна приложимост при организации, които управляват множество активи и ИТ ресурси и целят оптимизиране управлението на информационно-технологични и бизнес услуги.

Резултатите постигнати в научния труд са обект на внедряване в процесите по управление на ИТ на водеща информационно-технологична компания. Потенциалния успех и приноси и резултатите на научния труд могат да бъдат възпроизведени и в други бизнес организации, които припознават приложимостта им, а и като част от учебни дисциплини изучвани в Технически университет – София и други академични институции.

### **Апробация**

Резултатите на научния труд са апробирани във водеща информационно-технологична компания.

### **Публикации**

Основни постижения и резултати от дисертационния труд са публикувани в 8 научни статии. Всички 8 научни статии са докладвани на международни научни конференции. Една от статиите е самостоятелна, четири са в съавторство с научния ръководител и три в съавторство с други автори. Седем от публикациите са индексирани в Scopus. Научните публикации са публикувани в IEEE, Springer и EDULEARN22.

### **Структура и обем на дисертационния труд**

Дисертационният труд е в обем от 157 страници, като включва увод, 4 глави за решаване на формулираните основни задачи, списък на основните приноси, списък на публикациите по дисертацията и използвана литература. Цитирани са общо 117 литературни източници. Работата включва общо 46 фигури и 12 таблици. Номерата на фигурите и таблиците в автореферата съответстват на тези в дисертационния труд.

## **II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД**

---

### **ГЛАВА 1. АНАЛИЗ НА СЪЩЕСТВУВАЩОТО СЪСТОЯНИЕ. ДЕФИНИРАНЕ НА ПОСТАВЕНИТЕ ЗАДАЧИ ПРЕД ДИСЕРТАЦИОННИЯ ТРУД. РАЗГЛЕЖДАНЕ НА КОНЦЕПТУАЛНИ МОДЕЛИ.**

Съществуват множество различни решения за съхранение на ИТ ресурси, за изграждане на модел и структура на данните, която позволява по-нататъшната им ефективна обработка. Към настоящия момент няма универсално решение, което да бъде използвано от различни организации и да гарантира изграждането и съхранението на информация за ИТ ресурси в най-правилен начин.

#### **Същност на бизнес системите**

Бизнес системите и процесите са основен фрагмент поддържащ една компания функционираща и просперираща. В годините бизнесът и икономиката са се променили многократно, но е останало едно важно правилно валидно и днес – успешният бизнес се изгражда върху успешни бизнес системи.

Целите на бизнес системата са да спомогне постигането на целите на една бизнес организация. Сама по себе си, бизнес системата е комбинация от политики, служители, оборудване, правила, всички от които устремите в доставянето на някаква услуга или продукт на крайния потребител. Бизнес системите са набрали огромна популярност в съвременния бизнес и са се превърнали в неразделна част от него. Всяка една бизнес система доставя напълно или частично автоматизирано решение спомагащо дейността на бизнес организация или предприятие.

Предлаганите към момента бизнес системи не предлагат решение само на един или няколко проблема, а предлагат решение за множество утвърдени и масово използвани процеси и практики. В повечето случаи, предлаганите на пазара бизнес системи са разработени въз основа на библиотеки съдържащи най-добрите практики, тоест такива практики, които са се утвърдили през последните години, а понякога и десетилетия. Някои от популярните библиотеки/стандарти обединяващи множество добри практики са ITIL (IT Infrastructure Library), COBIT (Control Objectives for Information and Related Technologies), MOF (Microsoft Operations Framework), TOGAF (The Open Group Architecture Framework) и много други.

#### **Информацията в бизнес системите**

В наши дни притежанието на информация е ключов фактор, гарантиращ успех и контрол над множество процеси и дейности. Всяка една компания работи с различни типове информация, нужна за постигане нейните цели. Информация бихме могли да считаме всичко записано в договорите на една компания с нейните служители (в частност техните възнаграждения и бонуси), с нейните клиенти и партньори, както и техническата информация гарантираща дейността на организацията/бизнеса.

### **Проблеми в сигурността**

Възможността да съхраняваме информация в системи е направила много компании успешни. Някои компании като Google, Facebook, eBay и други не биха съществували без тази възможност, което показва каква първостепенна важност е заела информацията.

Предвид огромното значение на информацията и влиянието, което би могло да се постигне чрез нейното притежание, тя е обект на множество интереси – на различни хора, организации и правителства.

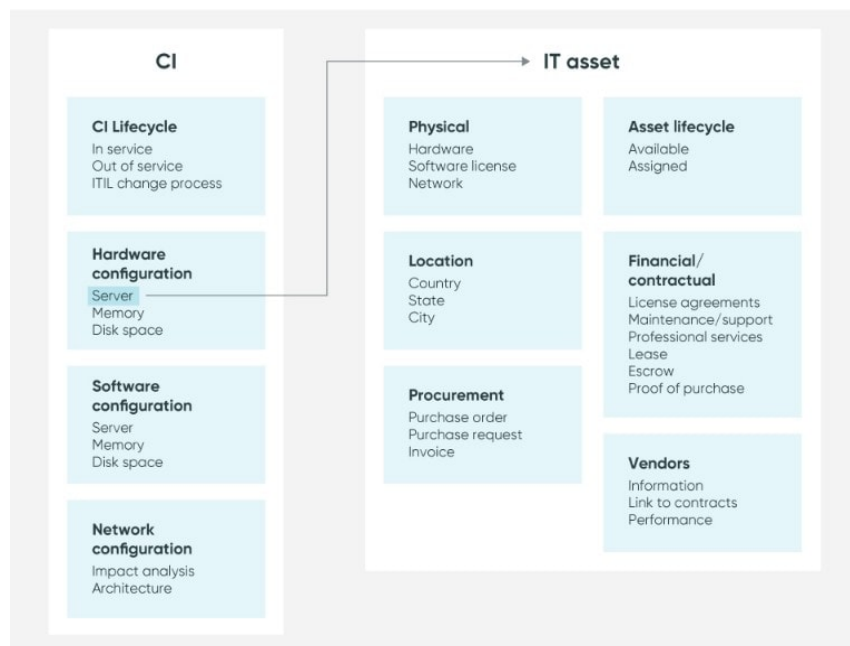
От банкови атаки и зловредни мобилни приложения до изнасяне на вътрешна информация, всяка година се превръща в още по-голямо предизвикателство за екипите гарантиращи сигурността на информация и хората изпълняващи длъжността Chief Information Security Officer.

Сигурността на системите за управление на информацията са тези мерки, които са предвидени и заложили за да защитят ресурсите в една информационна система от неотризиран достъп или от това да бъдат компрометирани. Рисковете за сигурността са тези слабости в една компютърна система, софтуер или хардуер, които биват използвани от „атакуващия“ с цел получаване на неотризиран достъп до информация или компютърна система.

### **Какво е база данни за управление на конфигурациите (CMDB)?**

CMDB е централно хранилище, което служи за склад на данни (data warehouse), съхраняващ информация за ИТ среди и ресурси.

CMDB предоставя възможност за съхранение на данни за ИТ ресурси и конфигурационни елементи (CIs). Изграждащите елементи на CMDB са CI-ите, като все от тях представлява различен ресурс (напр. рутер, сървър, приложение, виртуален сървър, компютър, услуга, а дори и логически елементи като портфолио или продуктова линия). [1]



Фиг. 7 – Примерни данни съхранявани в CMDB

Употребата на CMDB дава множество предимства за организациите. Събирането на информация за всички ресурси необходими за работата на една организация на едно място позволява по-добра видимост на ИТ операциите, предотвратява необходимостта данните да се събират от множество източници, а това от своя страна позволява проактивната работа по отстраняване на прекъсвания в услугите, намалява времето за отстраняване на прекъсванията, гарантира съответствие с вътрешните и външни нормативни изисквания за сигурност и одит и позволява по-лесно и информирано взимане на решения.

### **Аспекти на сигурността в изграждане и поддържане на данни. Преглед на стандарти за сигурност.**

Съгласно дефиницията на Национална база данни за уязвимости (NVD), програмата за често срещани уязвимости и експозиции (CVE), дава следната дефиниция за уязвимост - Слабост в изчислителната логика (напр. код), открита в софтуерните и хардуерните компоненти, която, когато се използва, води до отрицателно въздействие върху поверителността, целостта или наличността. Намаляването на уязвимостите в този контекст обикновено включва промени в кодирането, но може да включва и промени в спецификацията или дори отмяна на спецификацията (напр. премахване на засегнатите протоколи или функционалност в тяхната цялост). [30]

Създаването на абсолютно сигурен, не податлив към уязвимости софтуер е невъзможна задача. Дигиталния свят е така структуриран, че



със създаването на нов софтуерен подход или решение, нова метод за повишаване на сигурността, други групи от хората се активизират моментално в откриването на уязвимости и дефекти и възползването от тях. Индустрията за подобряване сигурността на ИТ решенията е индустрия с милиарди долари оборот, постоянно фокусирани върху грижата за информацията и нейната сигурност.

NIST е Националният институт за стандарти и технологии на САЩ и неговата Национална база данни за уязвимости (NVD) е хранилище на често срещани уязвимости и експозиции (CVE). Като един от най-доверените източници на информация за специалистите по ИТ и сигурността по света, NVD помага на екипите по сигурността по целия свят да бъдат в крак с уязвимостите в сигурността, когато бъдат открити. [29]

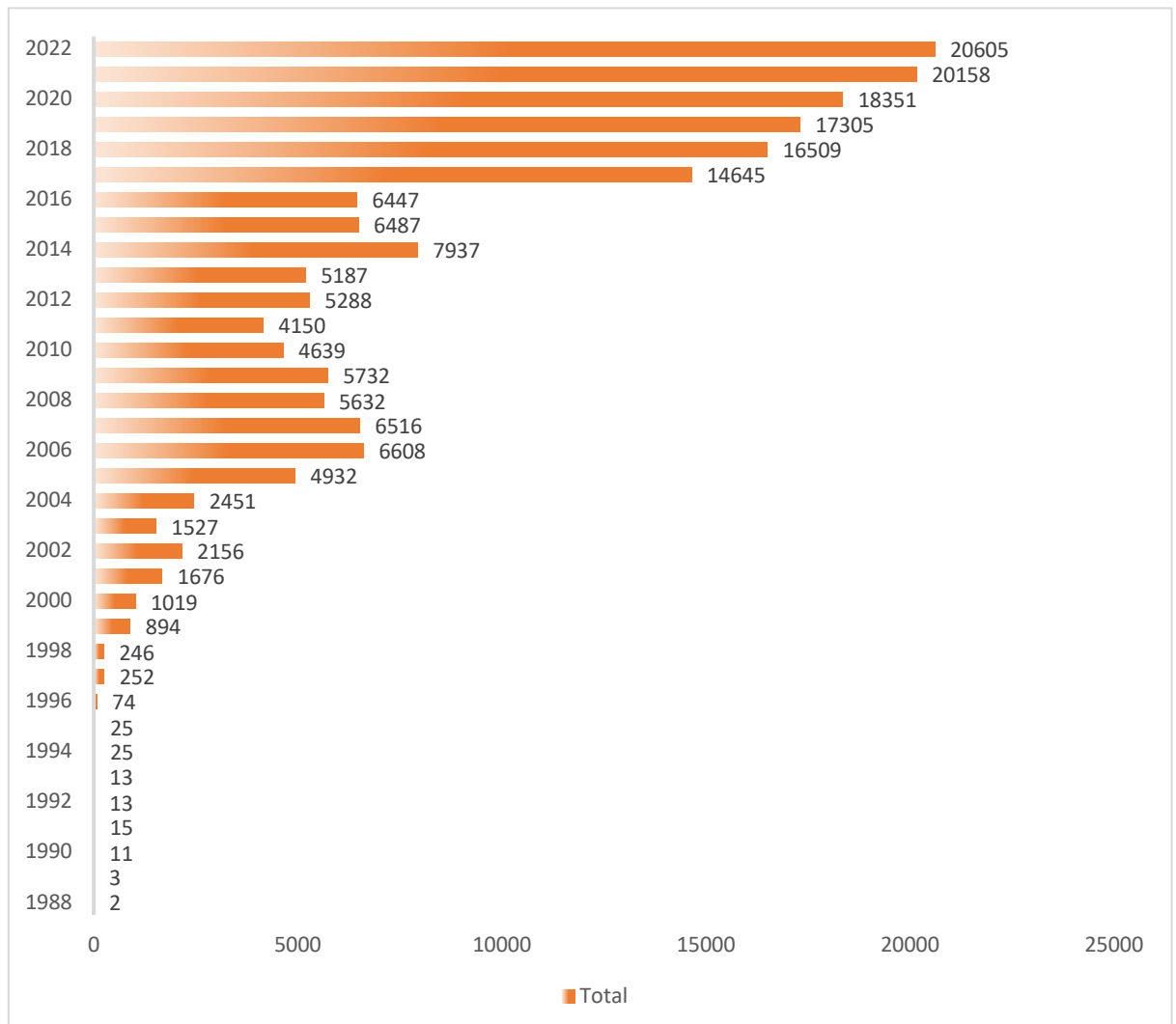
По данни на Redscan, 2021 година е една от най-тежките години по отношение сигурност на информация и системите. Наблюдава се изключителен ръст на ransomware атаките и нарастващата нужда от осигуряване на отдалечена работна сила. Има повече уязвимости в сигурността, разкрити през 2021 г. (18 439) (Данните са към 09.00 часа (GMT) на 8 Декември 2021 и се извадка от Национална база данни за уязвимости на NIST (NVD) на <https://nvd.nist.gov/> въз основа на CVSS v3.x.), отколкото през която и да е друга година до момента – средно повече от 50 CVE, регистрирани всеки ден.

Анализ извършен от Redscan показва:

- 90% от всички CVE, открити през 2021 г. досега, могат да бъдат използвани от нападатели с ограничени технически умения
- CVE, които не изискват взаимодействие с потребителя, като щракване върху връзка, изтегляне на файл или споделяне на техните идентификационни данни, представляват 61% от общия обем до момента
- 54% от уязвимостите към 08.12.2021 са класифицирани като имащи „висока“ наличност, което означава, че са лесно достъпни/използвани от нападатели [29]

По данни на NVD, 2021 поставя рекорд за брой регистрирани уязвимости в рамките на една година. Важно е да се отбележи, че през последните 5 години, всяка година е поставяла рекорд спрямо предходните. 2021 година приключва с 20 158 открити уязвимости, което е ръст от 9,3% процента спрямо предходната 2020 години и най-висок резултат за всички времена.

Една от изненадите за 2021 година е, че броя уязвимости класифицирани като „high severity” (висока тежест) е по-нисък от броя за предходната година. По данни на NVD 2021 записва 4 040 high severity уязвимости, докато през 2020 те са 4 381. [31]



Фиг. 10 Общ брой регистрирани уязвимости от 1988 до 29.10.2022 по данни на NVD [32]

Гореописаните числа могат да накарат всяка организация да се замисли и притесни за собствената си сигурност и колко приложения има в продуктивни среди с уязвимости. Има редица прости мерки, които една организация може да предприеме, за да подобри позицията си за сигурност на уеб приложенията. Първото започва в самото начало на разработването на приложения и това е да се уверите, че архитектите и разработчиците вземат предвид сигурността, когато разработват и кодират приложения. Второ, е да се гарантира, че софтуерът и операционните системи се поддържат актуални, с най-новите актуализации и корекции, за да се гарантира, че известните уязвимости, които имат корекции, не се

използват. Все пак, въпреки всички положени усилия, софтуери с уязвимости ще продължи да има, за това е важно организациите да имат изградена система и правилна структура на данните, която да ги подпомогне в откриването на дефекти и уязвимости на време, вземане на адекватни мерки и идентифициране на рискове.

В настоящия труд ще се концентрирам, както върху принципите за изграждане на правилна структура/модел на ресурси, така и за методи, по които оптимално и автоматизирано те могат да бъдат идентифицирани и обработени.

### **Анализ на съществуващи софтуерни решения**

В настоящия труд правя анализ на няколко съществуващи решения – платформи предоставяне под формата на услуга (PaaS) и софтуерни предоставяни под формата на услуга (SaaS), които често са използвани от бизнеса и ентърпрайз организациите за структурирано съхранение на информация за ИТ ресурси, а и не само. Този тип платформи са предвидени има за цел да осигурят на ползвателя лесен достъп до информацията, анализ, преглед и взимане на решения.

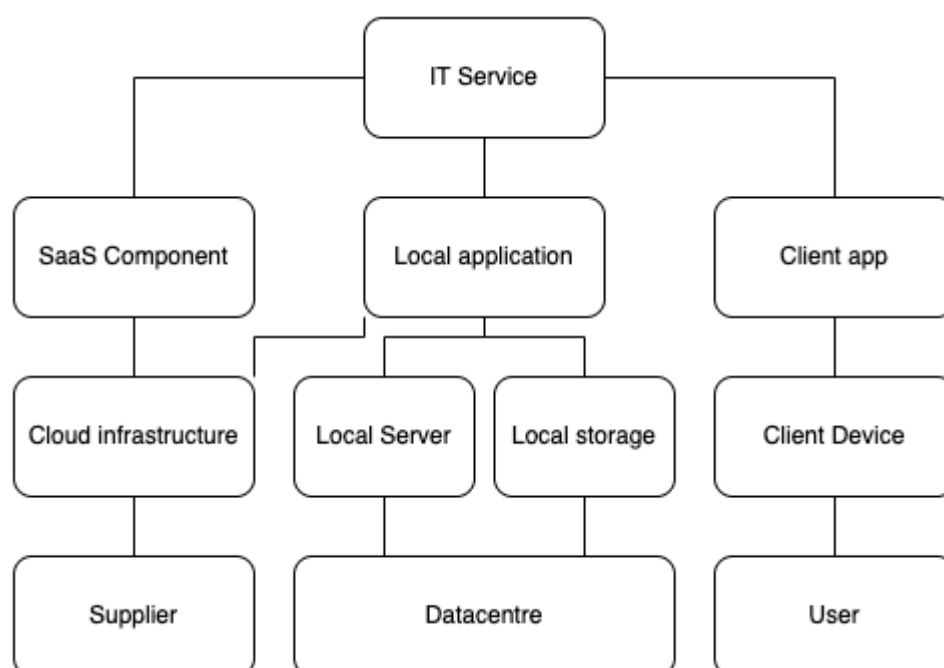
Предвид факта, че CMDB решенията често са припознавани като част от ITSM, в следващите пасажи ще разгледам проучване направено от Gartner Inc. Gartner Inc. е технологична изследователска и консултантска компания, която повежда проучвания за различни технологии. Клиенти на Gartner са различни големи корпорации, правителствени организации, технологични и инвестиционни компании. През 2018, Gartner отчитат, че тяхната клиентска база се състои от над 12 000 организации в над 100 държави. През 2022, компанията има над 15 000 служителя разположение в над 100 офиса по света [6]. Gartner е част от S&P 500. [7][8]

В рамките на дисертационния труд разгледам някои от платформите с най-добре развити ITSM процеси и CMDB възможности – ServiceNow, BMC и Ivanti.

### **Приложен характер на структурите на ресурси**

За да се гарантира, че няма влошаване на услугата и прекъсвания, организациите често внедряват процеси и системи за наблюдение и проактивен отговор на всякакви прекъсвания или потенциални проблеми. Повечето организации поддържат база данни за управление на конфигурацията (CMDB), която обикновено е под формата на база данни, която съдържа цялата необходима информация за хардуерни и софтуерни компоненти, използвани в услугата на информационните технологии (ИТ) на организацията и връзката между тези компоненти. Използването на CMDB предоставя организиран изглед на данните за конфигурацията на организацията и позволява изследване на данните от желана гледна точка.

Управлението на конфигурациите предоставя информация за CI, които допринасят за всяка услуга и техните взаимоотношения: как си взаимодействат; как взаимодействат и зависят един от друг, за да създадат стойност за клиентите и потребителите. Това включва информация за зависимостите между услугите. Този изглед от високо ниво често се нарича карта на услугата (service map) или модел на услугата (service model) и представлява част от архитектурата на услугата. [23] Фиг. 13 представлява опростена диаграма, показваща как множество CI допринасят за работата на ИТ услуга.



Фиг. 18 Опростена карта на услугата (service model) за типична ИТ услуга

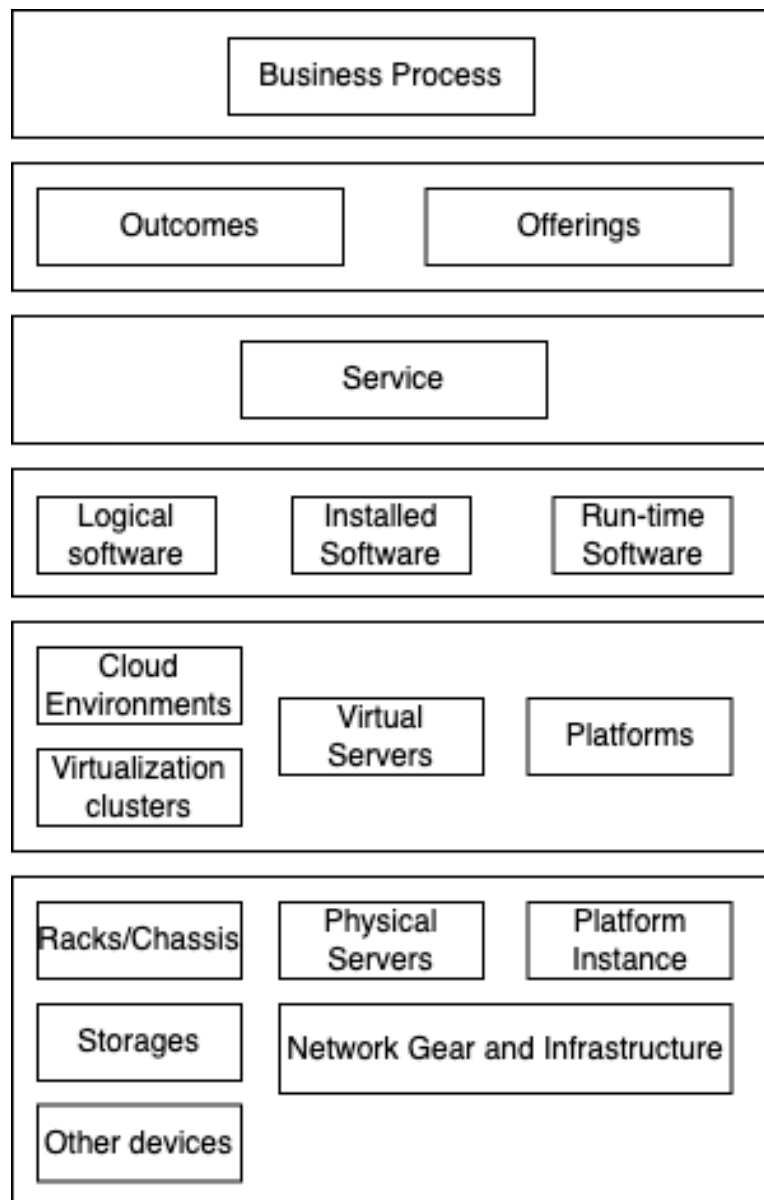
### Многослойна структура на ресурси

Структурата на ресурсите на корпоративните организации е изключително сложна. Броят на CI може да бъде от стотици хиляди до милиони елементи. Правилната структура на CMDB е от решаващо значение за лесната поддръжка и текущото използване на данните за активите. Незрелите организации не осъзнават важността и често не успяват да внедрят оптимална структура на своите данни за активи, осигурявайки солидна основа за бъдещи проблеми с CMDB и безпроблемно изпълнение на ITSM или други процеси, разчитащи на CMDB.

За да се гарантира, че организацията може да приеме добра структура на своите данни за ресурси, организациите трябва да приемат подходящи физически и логически модели на своите данни.

Илюстрираната по-долу концепция на физически модел е ориентирана към нива. Организациите могат да групират ресурси в 6 нива. Структурата на нивата обхваща бизнес процеси, резултати, услуги, софтуер, логическа инфраструктура и физическа инфраструктура.

Както е показано на Фиг. 19 Примерен концептуален модел, има множество компоненти и зависимости под всеки бизнес процес или бизнес резултати. Бизнес процесите разчитат на резултатите и предлагането на други услуги, за да постигнат своите резултати. Услугите се поддържат от софтуер, който може да включва, но не само, приложни пакети и внедрявания, приложен и инфраструктурен софтуер, екземпляри на бази данни и т.н., съответно компонентите на софтуерното ниво разчитат на логическа инфраструктура (облачни среди, кълстери за виртуализация, логически кълстери – преодоляване при отказ, балансиращи натоварването, виртуални сървъри и други. Нивото на най-долното ниво е физическата инфраструктура, където работи всичко - физически сървъри, стелажи, мрежово оборудване, комуникационни крайни точки, VLAN, IP адреси/мрежа. Във физическата инфраструктура можем също да намерим принтери, гласово и видео оборудване, хранилища и всякакви други устройства, които поддържат бизнеса.



Фиг 19. Концептуална структура на ресурси

Често компонентите от един модел зависят от/работят върху/абонират се за компоненти от друг/и модел/и, което прави структурата на ресурса доста сложна.

Поради сложността на модела или CMDB данните, често организациите не успяват да информират или включат всички заинтересовани страни, когато възникнат инциденти или искания за промени, като по този начин наличието на алгоритъм за идентифициране на всички зависимости, съответно всички заинтересовани страни, е важна задача за осигуряване на висока наличност на услуги.

Изграждането на CDMB с правилен и ефективен модел на данните е ключов аспект за последващото управление на ИТ организацията и ИТ услугите.

Проучване на Gartner от 2013 година показва, че най-честата „причина за смърт“ на CMDB е неговата комплексност. Всяка организация има свой собствен дизайн на CMDB модела на данни, въпреки опитите на водещите индустриални компании да предложат общ модел на данните, приложим за повечето организации.

Създаването на не предписващи, адаптивни и прагматични насоки относно дизайна на CMDB е основна задача в индустрията.

С настоящия труд изследвам концептуален модел на данни, с приложим обхват за ентърпрайз организации, който да улесни управлението на ИТ организацията и ИТ услугите и да даде ефикасен метод за управление на CMDB и изграждащите го данни.

## **ГЛАВА 2. ЙЕРАРХИЯ НА РЕСУРСИ**

Предвид броя на системите, които могат да участват в събирането и обработката на данните, от изключителна важната е информацията да има ясно определен собственик (owner). Трябва да е ясно, къде се съхранява основната информация, а в случай че информация бива възпроизвеждана, то това трябва да се случва контролирано и да се използва централизиран подход, където е възможно.

Изгражданото CMDB следва да съдържа всички конфигурационни елементи (CI) и техните взаимовръзки.

### **Цели на модела**

Моделът трябва да има достатъчно ниво на детайлност, за да отговори на изискванията на CMDB, но не толкова подробно, че да възпроизвежда функционалност на други системи. CMDB трябва да предостави единна система за запис на данните на дадената организация. Нивото на детайлност ще се определя до известна степен от откритите данни, но не всички данни ще трябва да бъдат моделирани и може просто да бъдат съхранени.

### **Общи данни**

Съществуват няколко атрибута, които са общи за повечето компоненти на структурата на CMDB, ето някои от тях:

- CIA оценка (Confidentiality, Integrity, Availability) - *Поверителността, целостта и наличността (Confidentiality, Integrity, Availability) известна още като CIA триадата, е модел, предназначен да ръководи политиките за информационна сигурност в организацията. CIA елементите са основополагащите и решаващи нужди за киберсигурност*

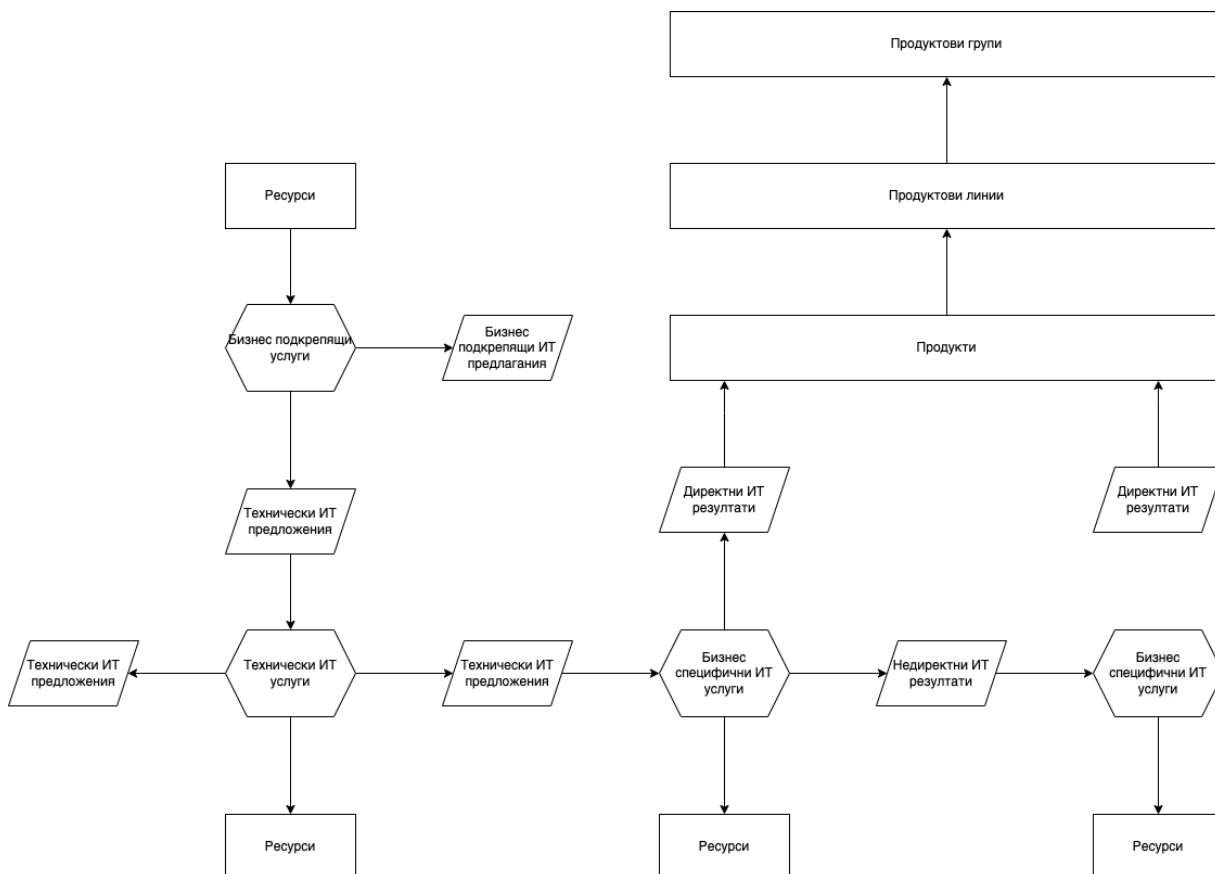
- Местоположение (локация) - *Съхранява информация къде физически са намира даден СИ. Структурата е в зависимост от големината на дадена организация, глобалните организации биха имали по-сложна структура от представената.*

## **ИТ Услуги**

Определението за ресурси може да включва хора, процеси, технологии, данни и т.н. Въпреки това, в тази версия на модела Ресурсите са ограничени до Конфигурационни елементи. Трябва да се отбележи, че няма пряка връзка между ресурс (включително Приложения/Applications) и ИТ предложения или ИТ резултат. ИТ услугата използва всички ресурси, които притежава, за да предостави едно или повече ИТ предложения/ИТ резултати.

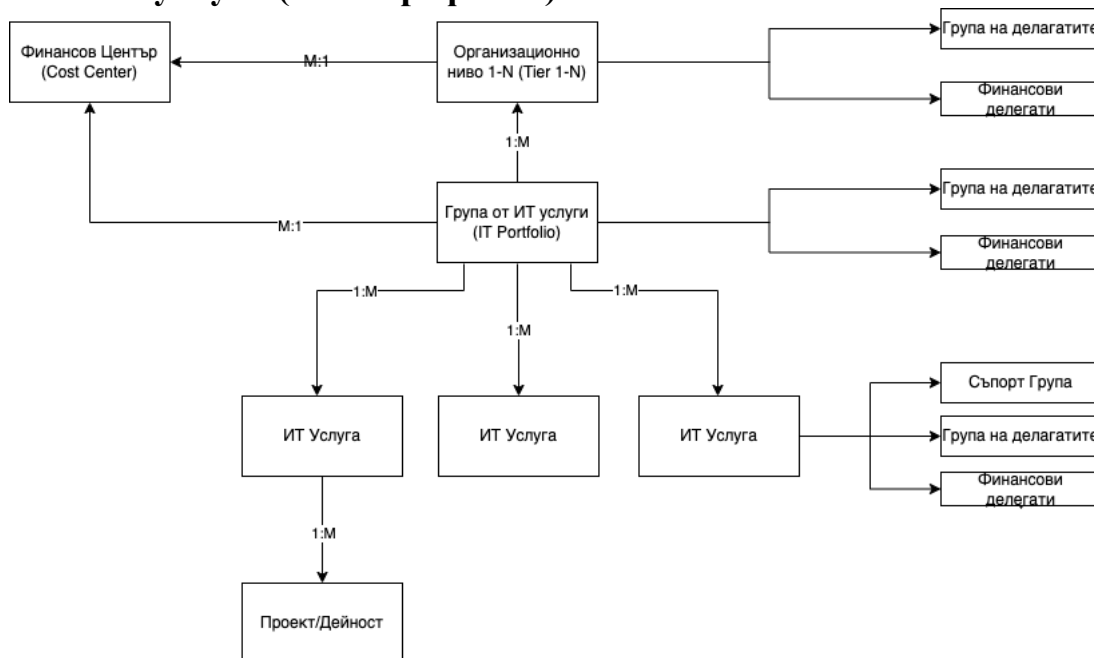
Връзката компонент/агрегация между ИТ услуга и ресурс предполага собственост. Когато даден ресурс (напр. приложение) се използва от множество ИТ услуги, тогава се използва отделна техническа ИТ услуга за доставяне на технически ИТ предложения до абониращите се ИТ услуги. Това е необходимо, за да се осигури правилна собственост и управление (особено контрол на промените). Техническите ИТ предложения може да са еднакви, но има вероятност да са различни. Трябва да се използва техническа ИТ услуга, тъй като само техническа ИТ услуга може да предоставя технически ИТ предложения на други ИТ услуги.





Фиг 21. - Модел на ИТ услуга

### Групи от ИТ услуги (ИТ Портфолио)



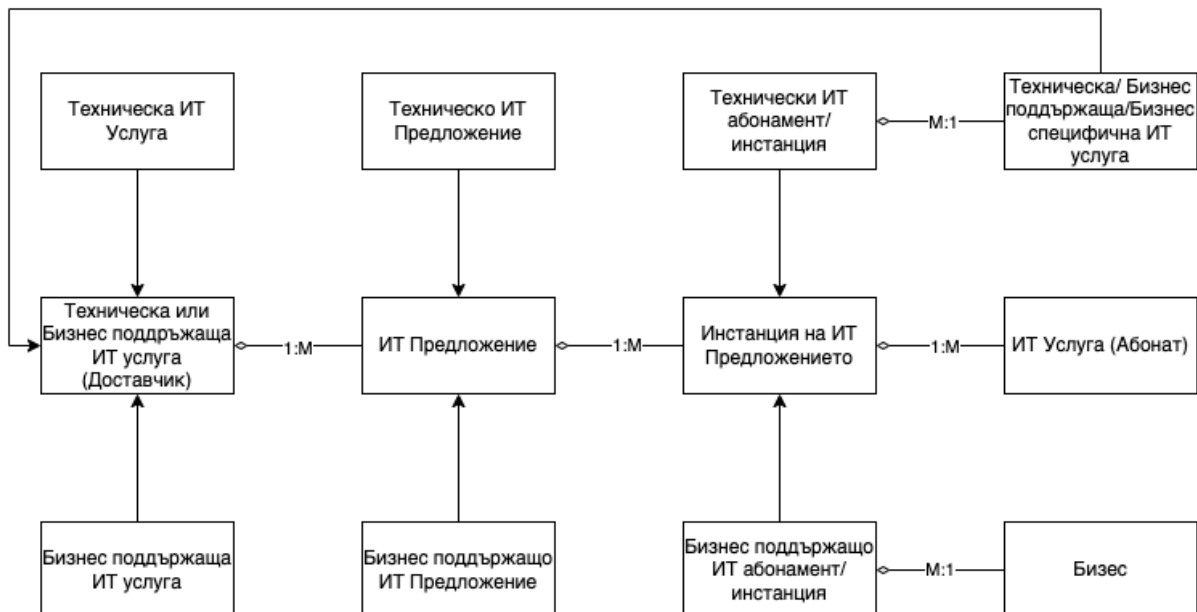
Фиг. 23 Групи от ИТ услуги

С оглед оптимално управление и контрол, ИТ услугите се организират в групи, наричани ИТ Портфолия. Това е независимо от каквато и да е официална организационна структура.

## ИТ Предлагания (IT Offerings)

Техническа или бизнес подкрепяща ИТ услуга се определя от ИТ предложенията, които предоставя. Една техническа или бизнес подкрепяща ИТ услуга трябва да има поне едно ИТ предложение, а обикновено те са много. Едно ИТ предложение не може да съществува без точно една ИТ услуга, която да го предоставя.

Техническите ИТ услуги съществуват, за да поддържат други ИТ услуги (технически, поддържащи бизнеса или специфични за бизнеса). Бизнесът не може да се абонира за технически ИТ предложения; те са невидими за бизнеса, но същевременно критична за неговата дейност.



Фиг. 24 Модел на ИТ Предлагания

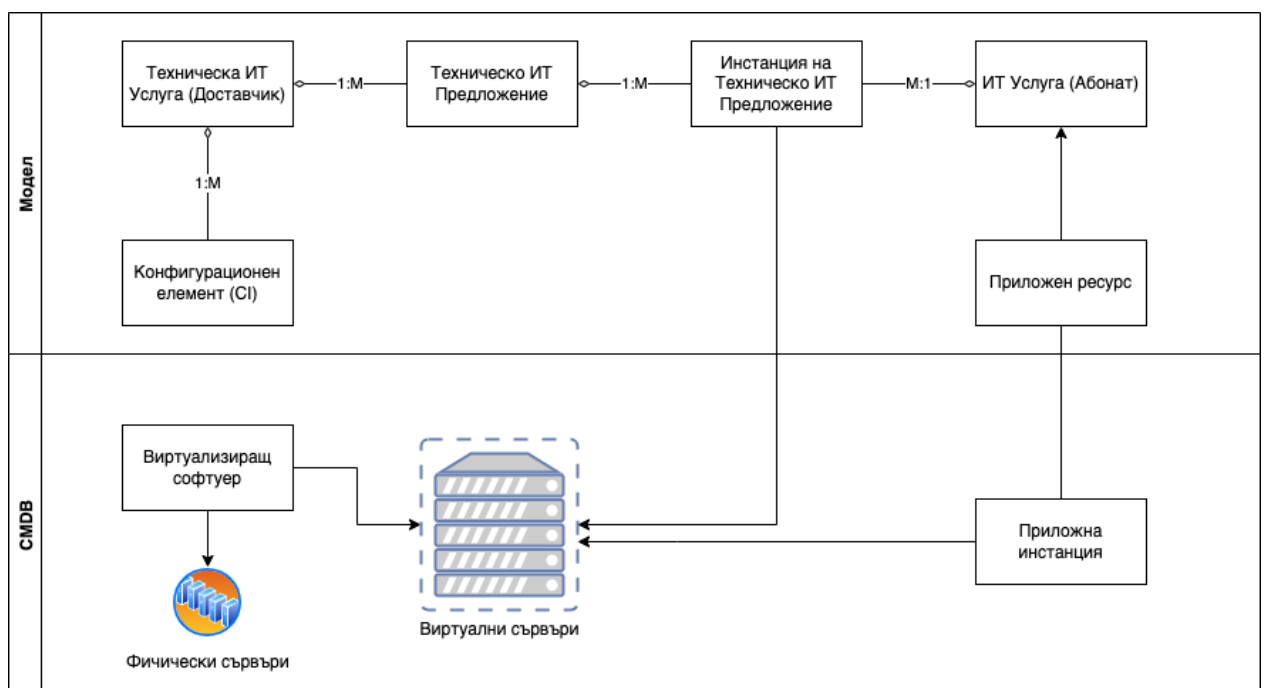
## Технически ресурси

Някои технически ИТ услуги съществуват, за да предоставят ресурс на абонатите, които на свой ред ще го използват, за да създадат стойност. Например, cloud техническа ИТ услуга, може да предоставя място за създаване на виртуален сървър, виртуален суич, база данни или др., на която е инсталирано приложение като част от друга специфична за бизнеса ИТ услуга.



Фиг. 25 Технически ресурси

Следващата фигура илюстрира типичен пример на приложение (част от Бизнес специфична услуга) качено (hosted) на виртуален сървър, част от Техническа ИТ услуга.



Фиг. 27 Пример на приложение, част от Технически ИТ услуга

## Приложения и софтуер

Връзките между Приложения и Софтуер са изобразени на следващата графика.



Фиг. 28 – Приложения и софтуер

Приложните системи предоставят логически бизнес-ориентиран изглед на една или повече характеристики на софтуерните продукти. Приложна система предоставя ресурс на ИТ услуга, която доставя Предложения

(Offering) на бизнеса. В допълнение към приложните системи, ИТ услугите могат да използват други ресурси като компютърни системи, помощен персонал и др.

### **Софтуерни компоненти**

Софтуерните компоненти трябва да бъдат част от концептуалния модел за управление на активи и/или зависимости и могат да бъдат част (или всички) от софтуерен продукт. Те обаче нямат дефинирани софтуерни функции или софтуерни елементи.

### **Софтуерни функционалности**

Софтуерните функционалности представляват възможностите на даден софтуерен продукт, т.е. какво може да прави, независимо дали е конфигуриран да го прави или не.

### **Софтуерни среди**

Приложни системи и софтуерни продукти са независими от софтуерната среда (производствена, тестване, UAT, staging, разработка и т.н.), тъй като те включват пълния набор от всички необходими софтуерни функционалности.

## **ГЛАВА 3. СИНТЕЗ НА ПРАВИЛА, МЕТОДИ И АЛГОРИТМИ ЗА АВТОМАТИЗИРАНА ОБРАБОТКА НА ИТ РЕСУРСИ**

В трета глава са описани предпоставките и правилата за дефиниране на ефективни методи и алгоритми за автоматизирана обработка на ИТ ресурси. Предложените методи и алгоритъм са представени като псевдо-код и формално описание. Разгледани са предимствата, недостатъците и спецификите на предложените решения.

### **Предпоставки за провеждане на изследванията**

Дигитализирането на всички сфери на бизнеса и индустрията предполага работата с все по-големи масиви от цифрови данни. Технологиите са в основата на всеки етап от производствените процеси на бизнеса, както в сектори като производство, така и в сектори като ИТ, здравеопазване, образование, отбрана и много други. От изключителна важност за оцеляването и ефективността на икономиката и индустрията е същата, чрез ИТ процеси, методи и алгоритми оптимално да обработва информацията за наличните ИТ ресурси, да оптимизира работата с тях, да анализира различните ситуации и състояния, в които бизнеса бива поставен и да общи на база получените резултати данни и изводи за по-нататъшното подобрене. За целта следва да бъдат създадени надеждни методи и алгоритми, чрез които автоматизирано и навременно да се постигне

оптимизация, предотвратяване на дефекти и ефективен анализ на текущото състояние.

От гледна точка на бизнеса, употребата на подобни методи и алгоритми не е нещо ново и непознато. Въпреки това, за бизнеса остава основно предизвикателство превантивното и навременно откриване на дефекти и постигане на оптимизация в процеси и ресурси, преди дефекти свързани с тях да нанесат вреди за бизнеса – икономически, репутационни, технологични.

### **Критерии за ефективност**

Ефективност на алгоритъма е процеса по определяне колко „добър“ е един алгоритъм. От изключителна важността е всеки алгоритъм да работи за всички типове дадени входни данни и трябва да избягва условия, които ще го принудят да влезе в безкраен цикъл (infinite loop).

При определяне ефективността на алгоритъма има два основни аспекта, които следва да се вземат в предвид:

- Време – измерване необходимото време за успешното изпълнение на алгоритъма
- Пространство – измерва необходимия обем от памет за изпълнение на алгоритъма [72]

Трябва да се има предвид, че пространството и времето са обратно пропорционални в алгоритмичен смисъл.

### **Описание на предложените методи и алгоритъм**

Предложените методи и алгоритъм целят да разрешат:

- Автоматизирано идентифициране на отговорната услуга за даден ресурс, чрез извършване на алгоритмично обхождане на таблици с взаимовръзки между множество ресурси. Връзките между ресурсите обикновено имат вложеност (nested), с неопределена дълбочина
- Автоматизирано определяне на отговорни звена в дадена организация за разрешаване на дефекта
- Автоматизирано определяне на заинтересованите страни и страни, които имат пряко или косвено отношение към дефектирания ресурс
- Оптимизация, предотвратяване на дефекти и ефективен анализ на текущото състояние.

### **Псевдо-код на предложените методи и алгоритъм**

Основна цел на предложения алгоритъм е определянето на собственика/отговорните страни на дефектирания ресурс, съгласно предложени в настоящия труд концептуален логически модел – ИТ услугата притежаваща ресурса.

Алгоритъма стартира с подаване от потребителя на информация за дефектирания ресурс (Configuration item), с което се извиква метод за идентифициране на ИТ услугата притежаваща ресурса – `getOwningITService`.

### ***Метод `getOwningITService`***

Метода `getOwningITService` получава като входен параметър уникално ID (GUID) на дефектирания ресурс и го открива в `cmdb_ci` таблицата на платформата. След като записът бъде открит се прави нова заявка до таблицата, в която е създаден записът (CI), тъй като в нея съхраняваме релевантни за коректния тип ресурс данни.

При подаден входен CI от тип `IT Service` не е необходима последваща алгоритмична обработка. Метода приключва като връща обект с резултати за `Owning IT Service` и GUID към запис на отговорната група за разрешаване на дефекта.

При подаден входен CI различен от `IT Service`, метода взема йерархичната структурата на клас на дефектирания обект и прави циклична проверка за откриване на топ обхождане, приложим за клас на дефектирания обект. Необходимостта от циклична проверка е породена от желание клас от по-ниско ниво да зададе свой собствен тип на обхождане, различен от тип на обхождане приложим за клас от по-високо ниво.

CMDB Table ▾	Tree Walk Route
<input type="text" value="Search"/>	<input type="text" value="Search"/>
Application Software [cmdb_ci_application_software]	Component Route
Server Chassis [cmdb_ci_chassis_server]	Component Route
Database [cmdb_ci_database]	Application Route
DB2 Instance [cmdb_ci_db_db2_instance]	Component Route
Database Instance [cmdb_ci_db_instance]	Component Route
MSFT SQL Instance [cmdb_ci_db_mssql_instance]	Component Route
Oracle Instance [cmdb_ci_db_ora_instance]	Component Route
Sybase Instance [cmdb_ci_db_syb_instance]	Component Route
EC2 Virtual Machine Instance [cmdb_ci_ec2_instance]	Application Route
ESX Resource Pool [cmdb_ci_esx_resource_pool]	Component Route
ESX Server [cmdb_ci_esx_server]	Component Route
Hardware [cmdb_ci_hardware]	Component Route
Infrastructure Software [cmdb_ci_inf_software]	Component Route
IP Address [cmdb_ci_ip_address]	Component Route
IP Router [cmdb_ci_ip_router]	Component Route
IP Switch [cmdb_ci_ip_switch]	Component Route

Фиг. 35 Таблица с информация за класовете на обектите в CMDB и типа на обхождане

На база върнатия резултат за тип на обхождане, метода извършва различни операции:

1. При тип на обхождане component
  - a. Проверява за директна релация между подадени CI и IT Service, в случай че има такава, връща обект с резултати за Owning IT Service и GUID към запис на отговорната група за разрешаване на дефекта.
  - b. При липса на директна връзка между подадени CI и IT Service, извиква метод `_checkDirectRelationship` за проверка на взаимовръзка (relationship) в таблицата за взаимовръзките – `cmdb_rel_ci`
2. При тип на обхождане application извиква метод `_getCIRoute` за по нататъшна алгоритмична обработка.

### ***Метод `_checkDirectRelationship`***

Метода цели да открие връзка (relationship), в таблицата с взаимовръзки на платформата `cmdb_rel_ci`, между дефектирания CI и притежаващия го IT Service.

За целта се прави заявка към база, в която се търси по `child` обект, дефектирания CI, и тип на връзката `TechnicalResource:TechnicalResource`, която е характерна за връзка на CI с притежаващия го IT Service.

При открит IT Service се връща обект с резултати за `Owning IT Service` и `GUID` към запис на отговорната група за разрешаване на дефекта.

### ***Метод \_getCIRoute***

На база базовия клас на дефектирания CI, алгоритмично се извършва обхождане на свързаните обектите с дефектирания CI.

В зависимост от подадения базов клас, алгоритъмът извършва някои от следните операции:

1. Проверява за директна релация между подадени CI и IT Service, в случай че има такава, връща обект с резултати за `Owning IT Service` и `GUID` към запис на отговорната група за разрешаване на дефекта.
2. При липса на директна връзка между подадени CI и IT Service, извиква метод `_checkDirectRelationship` за проверка на взаимовръзка (relationship) в таблицата за взаимовръзките – `cmdb_rel_ci`
3. Извлича йерархичната структура на класовете на ИТ услугите и извлича всички взаимовръзки на дефектирания CI и други CI от `cmdb_rel_ci`. За всеки върнат CI се
  - a. Извършва проверка дали е обект от клас на йерархичната структура на класовете на ИТ услугите, в случай че да, то този обект получава и връща като `Owning IT Service` с `GUID` към запис на отговорната група за разрешаване на дефекта.
  - b. Извършва проверка за пряка релация с ИТ услуга. При наличие на такава се връща резултата.
  - c. В случай че предходните две хипотези не върнат резултат за открита ИТ услуга, рекурсивно се извиква метода описан в т.3, до откриване на ИТ услугата.
4. Приложимо за `Application Deployments`. Извиква се метод `_getApplicationDeployment` и проверява за директна връзка през `parent Application Package` с ИТ услуга
  - a. В случай че има връзка връща резултат
  - b. В противен случай извиква метод `_getApplicationService` с подаден `parent Application package`
5. Приложимо за `Application Packages`. Метода `_getApplicationService` проверява за директна връзка през `parent Application Package` с ИТ услуга
  - a. В случай че има такава връща резултат
  - b. В противен случай рекурсивно се самоизвиква до откриване на най-високо ниво `Application package`, към когото има свързана



ИТ услуга. Тази хипотеза е валидна, в случай че имаме вложени Application packages.

- с. Ако не се открие резултат се извиква метода `_checkDirectRelationship` за проверка на взаимовръзки в `cmdb_rel_ci`.

### ***Алгоритмичен подход при обработка на заявки за промяна (change requests)***

Заявките за промяна (change requests) са по-специфичен тип заявки. За разлика от инцидентите и service requests, те са обвързани с промяна по инфраструктура на организацията и изисква по-внимателен подход, за да се избегне създаването на нов проблем/дефект, свързан с извършваната промяна.

При обработката на такъв тип заявки, предложениия метод се различава от основния подход за откриване на ИТ услуга.

Използва се `_changeRoute`, чрез който определяме подхода за обработка в зависимост от базовия клас на дефектирания CI, както следва:

- При обекти от тип ИТ Предлагания се извиква метод `_getProvidingService`
- При обекти от тип Application deployments се извиква метод `_getApplicationDeployment`
- При всички останали обекти се извиква метод `_standardChangeRoute`

### ***Метод \_getProvidingService***

Метода проверява за директна връзка между дефектирания CI и ИТ услуга.

### ***Метод \_standardChangeRoute***

Този подход се използва при standard change requests. Обработката му извършва следното:

1. Проверява връзки на дефектирания CI в `cmdb_rel_ci`, едновременно като parent и child участник във взаимовръзката.
  - a. В случай че parent или child обекта е Application deployment, за всяка открита взаимовръзка, без значение дали участва като parent или child обект, чрез метода `_getApplicationDeployment` извлича свързаната ИТ услуга и я записва в масив `serviceArray`.
2. В случай, че `serviceArray` е празен
  - a. Проверява за директна релация между подадени CI и IT Service, в случай че има такава, връща обект с резултати за Owning IT Service и GUID към запис на отговорната група за разрешаване на дефекта.
  - b. При липса на директна връзка между подадени CI и IT Service, извиква метод `_checkDirectRelationship` за проверка на

взаимовръзка (relationship) в таблицата за взаимовръзките –  
cmdb\_rel\_ci

3. В случай, че serviceArray има записани обекти, извиква javascript метод reduce върху масива serviceArray  

```
if (!!serviceArray.reduce(function (a, b) { return (a === b) ? a : NaN; })) {  
    return objectService;  
}
```
4. Ако и той не открие ИТ услугата
  - a. Проверява за директна релация между подадени CI и IT Service, в случай че има такава, връща обект с резултати за Owning IT Service и GUID към запис на отговорната група за разрешаване на дефекта.
  - b. При липса на директна връзка между подадени CI и IT Service, извиква метод \_checkDirectRelationship за проверка на взаимовръзка (relationship) в таблицата за взаимовръзките – cmdb\_rel\_ci

Във всички останали случаи, в които алгоритъма не е открил owning IT Service или запис на отговорната група за разрешаване на дефекта, се генерира нотификация до отговорна група с информация за edge case изключение). По този начин се подпомага организацията, в подобряване качеството и интегритета на данните.

## **ГЛАВА 4. ЕКСПЕРИМЕНТИ И РЕЗУЛТАТИТЕ (РЕАЛИЗАЦИЯ В СОФТУЕРНА СРЕДА)**

В четвърта глава разглеждаме проведените експериментални изследвания – реализиране в софтуерна среда, тестови сценарии, валидации, точността на изведените резултати.

Експерименти проведени в рамките на настоящия труд са реализирани в платформата ServiceNow, версия San Diego, и е използван JavaScript. ServiceNow San Diego използва JavaScript engine, който поддържа ECMAScript 5, базиран на Rhino версия 1.7 R5. [104]

Обект на изследването са предложените в Глава 3 методи и алгоритми за автоматизирано откриване на ресурс в голяма Configuration management database (CMDB) база от данни, с цел оптимизация, предотвратяване на дефекти и ефективен анализ на текущото състояние.

### **Цели на експерименталното изследване**

Основни цели на експеримента е да се:

- Изследват в реална среда предложените методи и алгоритъм
- Анализират резултатите от предложените методи и алгоритъм

- Валидират изходните данни от предложените методи и алгоритъм
- Валидира създадения концептуален модел
- Направи анализ на времето необходимо на алгоритъма за обработка на данните и извеждане на ресурс
- Докаже приложението и ефективност на предложените методи и алгоритъм в реална среда

### **Задачи на експерименталното изследване**

Задачите поставени пред експерименталното изследване могат да се обхванат в няколко групи:

- Изграждане в реална среда на модел на данни на информационно-технологични (ИТ) ресурси - материални активи, апаратни средства, софтуерно-технологични решения, приложен софтуер, бизнес логика, ИТ и бизнес услуги.
- Създаване на програмен алгоритъм за оптимално, според зададени критерии, локализиране на дефекти в описаната йерархична среда
- Имплементация на алгоритъма в програмна среда за управление и устойчивост на системата от ресурси
- Експериментално изследване на изходните данни от обработката на алгоритъма

### **Подбор на данните за изследването**

Данните, използвани в рамките на настоящото експериментално изследване са синтезирани от публично налични примерни данни за ИТ ресурси в бизнеса, както и данни събрани в хода на дисертационния труд. Експериментално изследване е извършено върху извадка от над 2 330 000 конфигурационни елемента (CI) в CMDB структура. Наличните взаимовръзки (relationships) са над 3 570 000 броя, без да се включват директни релации на един конфигурационен елемент с друг.

Използваните данни са подбрани, така че да отговарят на данните, с които разполагам големи (ентърпрайз) организации, те са разнородни, в някои случаи не консистентни, с нисък интегритет.

### **Експериментално изследване**

В рамките на експерименталното изследване, предложеният алгоритъм е изпробван върху 10 различни групи от ресурси (описани по-долу), като за всяка група е изпълнен върху минимум 10 000 конфигурационни елемента (configuration items), за всяка от двете пътеки на алгоритъма – incident и change пътеките. При изпълнението си алгоритъма автоматизирано открива оптимално, според зададени критерии, локализиране на дефекти в описаната йерархична среда, като обработва общо над 200 000 уникални обекта.

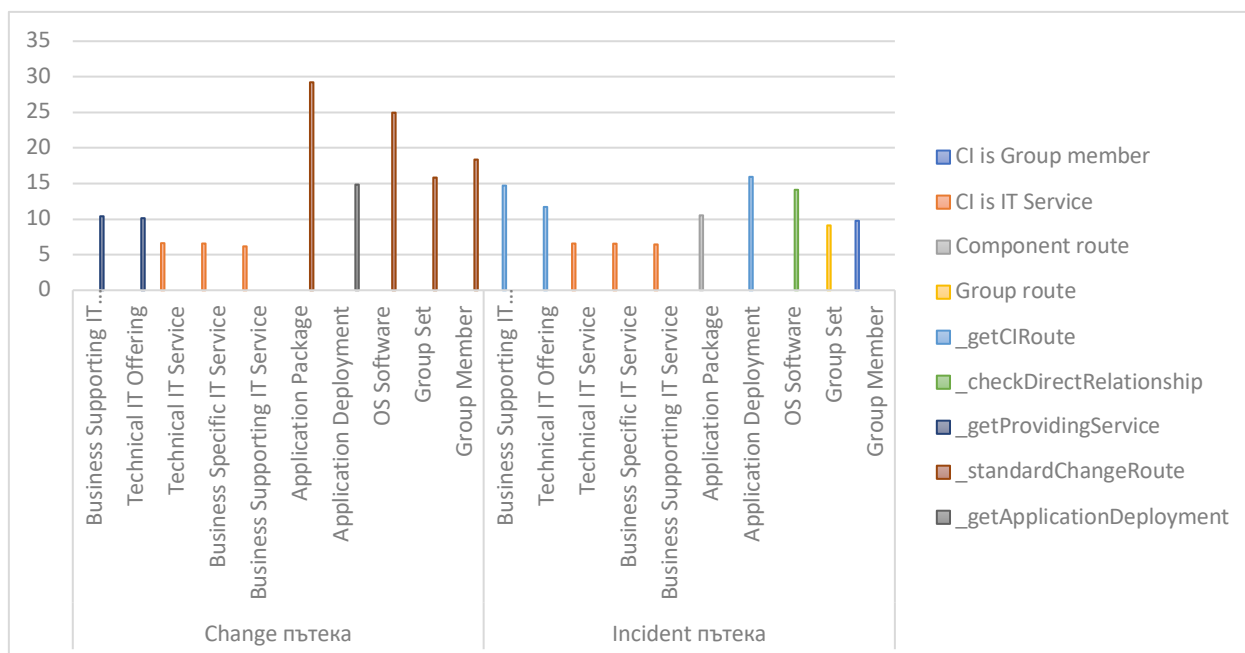
### **Поведение на експерименталната среда**

Експерименталното изпълнение на алгоритъма цели да изследва въздействието му върху експериментална платформа и по какъв начин нейната производителност и параметри са повлияния от изпълнението на алгоритъма и експерименталното обхождане на над 200 00 уникални обекта.

### Експериментални резултати

В рамките на тази точка ще разгледаме резултатите от обработката на алгоритъма, в частност необходимото време за откриване на търсения обект и съпоставка между времената за откриване по различните пътеки на алгоритъма и в различните класове.

Бърз поглед на резултатите показва, че времето необходимо за откриване на притежаваща (owning) ИТ услуга за дефектирал конфигурационен елемент варира от 6 до малко над 29 милисекунди за всеки обработен обект. Експеримента показва, че средното време за обработка на всички обекти от различни класове е 12.5 милисекунди. Алгоритмично обхождане и обработка за средно време от 12.5 милисекунди е изключително добър резултат, в контекста на конфигурационна база от данни с обем от над 2 330 000 конфигурационни елемента (CI) в CMDB структура и взаимовръзки (relationships) от над 3 570 000 броя, без да се включват директни релации на един конфигурационен елемент с друг.



Фиг. 45 Графично изобразени експериментални резултати на база средно време за изпълнение (ms) - групирана по пътека клас на обект

При анализ на резултатите групирани по change/incident пътека и клас на обект се вижда, че времето необходимо за откриване на притежаваща ИТ услуга е по-голямо при обработка по change пътеката, това е така тъй като

change пътеката изпълнява няколко по-сложни операции, с по-висока сложност на алгоритъма, съответно броя обхождания е значително по-висок. Причината за подобен тип подход е нужда от идентифициране не само на притежаващата ИТ услуга, но и идентифициране на всички останали свързани/заинтересовани обекти и групи от обекти. Резултатите показват, че забавянето между обработката по двете пътеки не е съществено от гледна точка производителност и време за откриване на резултат, но е възможно алгоритъма да трябва да бъде оптимизиран при обработка на многократно по-големи бази от данни. Подобна оптимизация на change пътеката ще бъде разглеждане и обект на допълнителен труд.

В зависимост от класовете на обработените обекти и пътеката, по която са извършени изчисленията, могат да се направят няколко извода на база разликата във времето необходимо за обработка по двете пътеки:

- Под 1 милисекунда – разликата се наблюдава основно при обработката на обекти от под клас на ИТ услугите. Този резултат е напълно очакван, тъй като при класифицирането на обектите не се очаква задълбочена обработка и обхождане на базата за никоя от пътеките.
- До 5 милисекунди – такъв резултат дават обектите от под класове на ИТ предлагания и Инстанции на приложенията. Този резултат отново е очакван и приемлив, тъй като този тип обекти имат ясно разположение в модела на данни и необходимия брой обхождания е лимитиран. При change пътеката е необходим допълнителен ресурс за обработка до идентифициране на всички свързани обекти.
- Над 5 милисекунди – този резултат се наблюдава при софтуерните пакети, group set-овете и group member-ите. При този тип обекти, модела не дава директна връзка с притежаващата ИТ услуга, поради което е необходимо по-сериозно обхождане на базата. При change пътеката се изисква пълно идентифициране на всички свързани обекти в модела, което предполага по-сложната алгоритмична обработка.

## **НАУЧНИ, НАУЧНО-ПРИЛОЖНИ И ПРИЛОЖНИ ПРИНОСИ**

### **Приноси с научен характер**

1. Предложен е концептуален модел за структура на данни на конфигурационни елементи в големи (ентърпрайз) организации, който модел спомага за оптимално идентифициране на дефектирали елементи в инфраструктурата на организацията
2. Дефинирано е описание на йерархии на разнородни обекти обуславящи създаването на качествена и ефективна база от данни на конфигурационни елементи (configuration management database)

3. Предложен е алгоритъм с елементи на експертна система за автоматизирана обработка на ИТ ресурси, идентифициране на свързани с дефектирал конфигурационен елементи обекти и таксономия за последваща класификация и действия.

### **Приноси с научно-приложено характер**

1. Анализирани са наличните международни стандарти и специфики при изграждане на CMDB база от данни и управление на инфраструктурните елементи в различни организации с цел изграждане на концептуален модел на конфигурационна база от данни, съобразен с актуалните тенденции и изисквания
2. Анализирани са най-често срещаните проблеми, пред бизнес структури и компании по отношение оптималното откриване, анализ и превенция на информационно-технологични проблеми.
3. Резултат на изследванията и създадения продукт, са анализирани и класифицирани обекти в голяма база от данни, които са използвани при усъвършенстването и финализирането на предложените методи и алгоритъм с оглед създаване на експертна среда.

### **Приноси с приложен характер**

1. Предложени са средства за локализиране по оптимален начин на дефекти в йерархична ресурсна система
2. Разработени са методи и алгоритъм, които позволяват лесна имплементация в реална софтуерна среда

## **СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИОННИЯ ТРУД**

1. Nakov, O., Mladenov, V., Mihaylova, E., Nakov, P. (2021). Analysis and Evaluation of Distance Schooling and Learning with Respect to ICT Usage During COVID-19 Period in Bulgaria. In: Rocha, Á., Adeli, H., Dzemyda, G., Moreira, F., Ramalho Correia, A.M. (eds) Trends and Applications in Information Systems and Technologies . WorldCIST 2021. Advances in Intelligent Systems and Computing, vol 1367. Springer, Cham. [https://doi.org/10.1007/978-3-030-72660-7\\_19](https://doi.org/10.1007/978-3-030-72660-7_19)
2. R. Trifonov, O. Nakov, G. Pavlova, S. Manolov, G. Tsochev and P. Nakov, "Analysis of the Principles and Criteria for Secure Software Development," 2020 28th National Conference with International Participation (TELECOM), 2020, pp. 125-128, doi: 10.1109/TELECOM50385.2020.9299567.
3. B. Prebreza, D. Gotseva and P. Nakov, "A Study of Documents Management System Based on Web, Case Study: University," 2021 29th National Conference with International Participation (TELECOM), 2021, pp. 85-89, doi: 10.1109/TELECOM53156.2021.9659663.
4. O. Nakov, R. Trifonov, G. Pavlova and P. Nakov, "Analysis of Software Vulnerabilities, Measures for Prevention and Protection and Security Testing," 2021 29th National Conference with International Participation (TELECOM), 2021, pp. 73-76, doi: 10.1109/TELECOM53156.2021.9659585.

5. O. Nakov, R. Trifonov, G. Pavlova and P. Nakov, "Comparative Analysis of the Interoperability Assessment Methods and Approaches in the Industry 4.0," 2022 10th International Scientific Conference on Computer Science (COMSCI), 2022, pp. 1-4, doi: 10.1109/COMSCI55378.2022.9912606.
6. R. Trifonov, O. Nakov, G. Tsochev and P. Nakov, "The Use of Intelligent Methods to Protect Industrial Systems," 2022 10th International Scientific Conference on Computer Science (COMSCI), 2022, pp. 1-4, doi: 10.1109/COMSCI55378.2022.9912603.
7. P. Nakov, "Structure of resources in an enterprise IT organization subject to optimal defect detection," 2022 10th International Scientific Conference on Computer Science (COMSCI), 2022, pp. 1-3, doi: 10.1109/COMSCI55378.2022.9912586.
8. P. Nakov, E. Mihaylova, M. Lazarova, O. Nakov, V. Mladenov (2022) A SOFTWARE PLATFORM FOR DIGITAL ASSESSMENT OF PEDAGOGICAL SPECIALISTS' QUALIFICATION PROGRESS IN BULGARIA, *EDULEARN22 Proceedings*, pp. 832-837.

## SUMMARY

### METHODS AND ALGORITHMS FOR AUTOMATED PLANNING OF RESOURCE IN IT PROJECTS

#### (MODELLING OF INFORMATION-TECHNOLOGY RESOURCES AND ALGORITHMS FOR AUTOMATED PROCESSING)

Author: Plamen Nakov  
Prof. PhD Eng. Roumen Trifonov

The dissertation consists of 157 pages, including an introduction, 4 chapters for solving the formulated main tasks, a list of the main contributions, a list of publications on the dissertation and used literature. A total of 117 literary sources are cited. The work includes a total of 46 figures and 12 tables. The numbers of the figures and tables in the author's reference correspond to those in the dissertation.

The aim of the dissertation is the creation of strategies and methodologies for the development of interactive design based on user behavior for multifunctional applications.

The main goal of the dissertation is to provide a data structuring model and an algorithm that will allow businesses to build a correct data model of IT resources, provide an opportunity for modern and effective maintenance, create, and deliver value, realize distribution in value chains, ensuring improved quality, transparency, better insights, automation, and cost optimization.

As a result of the performed qualitative and functional analysis, following tasks were formed before the dissertation work:

1. Building a data model of information technology (IT) resources - material assets, hardware, software technology solutions, application software, business logic, IT, and business services. A description of the hierarchy and the interrelationships between them.
2. Creation of an algorithm with elements of an expert environment for optimal, according to set criteria, localization of defects in the described hierarchical structure.
3. Implementation of the algorithm in a program environment for management and sustainability of the resource system
4. Validation of the created conceptual model

Dissertations covers 4 main chapters, as follows:

**Chapter 1** - Analysis of Existing Condition. Defining the tasks set for the dissertation work. Examining conceptual models – *chapter focuses on analysis of different solutions for building and implementing a CMDB. It also looks at several modern platforms that focus on maintaining a CMDB and providing the means to manage, monitor, and analyze data from and about IT resources.*

**Chapter 2** - Resource Hierarchy – *chapter focuses on defining resource data structuring model that provides sufficient level of detail to meet CMDB requirements, but not so detailed as to replicate functionality of other systems*

**Chapter 3** - Synthesis of rules, methods, and algorithms for automated processing of IT resources – *chapter describes the prerequisites and rules for defining effective methods and algorithms for automated processing of IT resources. The proposed methods and algorithm are presented as pseudo-code and formal description. The advantages, disadvantages and specifics of the proposed solutions are considered*

**Chapter 4** – Experiments and Results (Implementation in a Software Environment) - *chapter considers the conducted experimental studies - implementation in a software environment, test scenarios, validations, the accuracy of the derived results.*