

**ВАРНЕНСКИ СВОБОДЕН УНИВЕРСИТЕТ "ЧЕРНОРИЗЕЦ ХРАБЪР"
ФАКУЛТЕТ "МЕЖДУНАРОДНА ИКОНОМИКА И
АДМИНИСТРАЦИЯ"
КАТЕДРА "ИНФОРМАТИКА"**

ЖАНАР ЕЛИБАЕВНА САРТАБАНОВА

**МОДЕЛИРАНЕ НА СИСТЕМАТА ОТ СЛАБОСТИ
НА СОФТУЕРА ПО CWE**

АВТОРЕФЕРАТ
на дисертационен труд
за получаване на образователна и научна степен "доктор",
професионално направление 4.6 Информатика и компютърни науки,
докторска програма "Информационни системи и технологии, информатика и
компютърни науки"

Научни ръководители:
проф. д-р Владимир Димитров,
доц. д-р Сауле Сарсимбаева

Варна, 2021

**ВАРНЕНСКИ СВОБОДЕН УНИВЕРСИТЕТ “ЧЕРНОРИЗЕЦ ХРАБЪР”
ФАКУЛТЕТ “МЕЖДУНАРОДНА ИКОНОМИКА И
АДМИНИСТРАЦИЯ”
КАТЕДРА “ИНФОРМАТИКА“**

ЖАНАР ЕЛИБАЕВНА САРТАБАНОВА

**МОДЕЛИРАНЕ НА СИСТЕМАТА ОТ СЛАБОСТИ
НА СОФТУЕРА ПО CWE**

АВТОРЕФЕРАТ
на дисертационен труд
за получаване на образователна и научна степен “доктор”,
професионално направление 4.6 Информатика и компютърни науки,
докторска програма “Информационни системи и технологии, информатика и
компютърни науки”

Научни ръководители:
проф. д-р Владимир Димитров,
доц. д-р Сауле Сарсимбаева

Рецензенти:
проф. д-р Росица Спасова Кузманова - Маринова
проф. д-р Калинка Михайлова Калоянова

Варна, 2021

Дисертационният труд, в размер на 112 страници, съдържа въведение, изложение в пет глави, заключение, списък на използваната литература и три приложения (20 стр.). Съдържанието на всяка от главите е разпределена в отделни части, като в края на всяка от главите са обобщени резултатите. Основният текст съдържа 9 фигури. Списъкът на използваните литературни източници съдържа 24 заглавия на руски и английски език.

Дисертационният труд е обсъден и насочен за защита пред научно жури от катедра „Информатика“ на факултета „Международна икономика и администрация“ към ВСУ „Черноризец Храбър“, гр. Варна.

Авторът на дисертационния труд е докторант на самостоятелна подготовка в катедра „Информатика“ на факултета „Международна икономика и администрация“ към ВСУ „Черноризец Храбър“ – гр. Варна.

Защитата на дисертационния труд пред научно жури ще се състои на 20.07.2021 г. в 14,00 часа, в Заседателната зала на ВСУ „Черноризец Храбър“ на заседание на научното жури. Материалите за запознаване са на разположение в канцеларията на секретаря на катедра „Информатика“ на факултет „Международна икономика и администрация“.

I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

1. Въведение. Актуалност и значимост.

Новите условия, в които работят като публични, така и частни организации по целия свят, в частност и в Република Казахстан, се нуждаят все по-нови подходи за подобряване на кибер сигурността.

Проблемът с кибер сигурността сега е един от най-обсъжданите и остри теми по целия свят. В резултат на развитието на цифровизация в почти всички области от дейността на човечеството, проблемът за защита на данните е важен въпрос. Хакерите по всякакви различни начини осъществяват атаки на сайтове на организациите, както държавните, така и на търговските, което води до проблеми на държавно и международно ниво.

Актуалността на темата на дисертация на труда е обособена от необходимостта от подобряване на работата на фирми, занимаващи се с проектиране и разработка на софтуер с оглед на отчитането на международния списък от слабости (CWE) на софтуера. Всяка една организация трябва да следи за сигурността закупения софтуер и разработката на такъв на всички нива. За защита на своите софтуерни продукти, компанията на всички етапи на разработка трябва да се придържа към препоръките и стандартите за информационна сигурност, както национални така и международни. За решаване на тези проблеми, общността на разработчиците и MITRE Corporation са създали международен списък на слабостите софтуера – CWE (Common Weakness Enumeration).

Значимост: Разработената онтология може бъде от полза на разработчиците на софтуер, изследователите в областта на разработката на софтуер и кибер сигурността, както и на преподавателите от учебните заведения, които водят курсове по софтуерни технологии и по информационна сигурност. На разработчиците, тази онтология може да послужи като справочник при проектирането на софтуер. Така слабите места в софтуера могат да бъдат отстранени в ранните етапи на разработка, а не впоследствие, когато софтуерът е завършен. За изследователите ще

са интересни въпросите свързани с изучаване и отстраняването на слабостите в софтуер. Преподавателите могат да използват онтологията като отправна точка при разглеждането и обсъждането на сигурността на слабите места при проектирането и архитектурата, а също така и по видовете грешки, които могат да бъдат допуснати по време на разработката на софтуер.

2. Обект и предмет на изследването

Обект на изследователската работа е системата от слабости софтуер CWE.

Предмет на изследване дисертация е архитектурната концепция в системата от слабости CWE.

3. Изследователски проблем

Изследователски проблемът. При разработването на софтуер най-често се допускат грешки при проектиране. За да се избегне появата на грешки, общността на програмистите е създала списък от слабостите на софтуера. Няма стандартна система за описание на слабости на софтуера.

4. Теза

Предполага се, че може да се разработи онтология на слабостите на софтуера, която да се използва от разработчиците на софтуер.

В рамките на този дисертационен труд е проектирана онтология за слабостите на софтуера по архитектурната концепция.

5. Цели и задачи на дисертационния труд

Основната цел на дисертационния труд е проучване и изследване на системата от слабости на софтуера на базата на CWE и разработването на онтология за тази система по архитектурната концепция.

За постигане на поставената цел е необходимо решаването на следните **изследователски задачи**:

- Проучване на структурата на системата от слабости CWE.
- Изследване на гледката: CWE VIEW: Architectural Concepts.

- Анализ на възможностите за използване на езика за онтологии OWL, RDF и средата за разработка на онтологии Protégé.
- Проектиране на OWL онтология на базата на CWE Architectural Concepts.
- Разработване на потребителски случаи за приложение на онтологията за архитекти, програмисти и университетски преподаватели.

6. Методика на научното изследване

Методологическата и теоретична основа на изследванията включва на общо научните техники за анализ и синтез, обобщение, абстрахиране, методи за групиране и класифициране на данни, логически методи, сравнителен, системен анализ, както и научните публикации на други автори в областта на компютърната сигурност, официалният сайт на международния списък от слабости на софтуера CWE, както и анализ на таксономията и синтез на схемата на онтологията, и изследване на търсенето в онтологии.

Ограничения в проблемни параметри на дисертационния труд

Дисертационният труд е ограничен в рамките на CWE и гледната точка на архитектурната концепция.

II. ОБЕМ И СТРУКТУРА НА ДИСЕРТАЦИОННИЯ ТРУД

Структура на дисертацията е – Увод, пет глави, Заключение, с общ обем 84 страници. Основният текст съдържа 9 фигури. Списък на използваната литература съдържа 53 заглавия от чуждестранни автори и Интернет източници. В допълнение са обособени 3 приложения (20 страници).

Структура на дисертационния труд:

ВЪВЕДЕНИЕ

Глава 1. Преглед на системата слабости софтуер Common Weakness Enumeration

- 1.1 Обща информация за CWE
- 1.2 Методология CWE
- 1.3 Концепцията CWE

Глава 2. Архитектурната концепция на Common Weakness Enumeration

- 2.1 Структура На Architectural Concepts
- 2.2 Структура на категориите
- 2.3 Структура на класове
- 2.4 Структура на базата
- 2.5 Структура възможности
- 2.6 Структура на съставните

Глава 3. Език за описание на онтологии OWL 2

- 3.1 Преглед на OWL 2
- 3.2 Синтаксис Manchester ARENA
- 3.3 Среда Protégé

Глава 4. Проектиране на база от знания

Глава 5. Използването На База От Знания За Софтуерен Архитект

- 5.1 Архитектурната концепция на Common Weakness Enumeration
- 5.2 Прилагането на база от знания

ЗАКЛЮЧЕНИЕ

Списък на използвани литературни източници

ПРИЛОЖЕНИЕ 1. Структурата на представяне на 1008 – Architectural Concepts

ПРИЛОЖЕНИЕ 2. Структура на категория 1019

ПРИЛОЖЕНИЕ 3. Един Пример за клас,

III. КРАТКО РЕЗЮМЕ НА ДИСЕРТАЦИЯ НА ТРУДА

В **Увода** е обоснована актуалността на избраната изследователска тема, представено е значението на дисертационния труд, определена е целта на изследването и са планирани изследователски задачи, също така са определени обектът и предмета на изследване в дисертацията, обосновано е значението на работата. Съдържа още кратка информация за всяка глава от дисертацията. Очертано е практическото прилагане на разработената онтология и са представени методическите и теоретичните основи на научните изследвания.

Първа глава „Преглед на Common Weakness Enumeration“ е посветена на теоретични аспекти на общия списък на слабости на софтуера (Common Weakness Enumeration) като обект на изследване.

Common Weakness Enumeration (CWE) е общ списък или речник на видове слабости на софтуера. Този речник е предназначен за програмисти, за специалисти в областта на информационната сигурност. Слабости могат да се появят в архитектурата, проекта, кода или в процеса на кодиране на софтуера, които впоследствие може да доведе до уязвимост в сигурността.

Слабостите (недостатъци) на софтуера са грешки или пропуски, които могат да доведат до уязвимости на софтуера. Уязвимостта на софтуера, като например тези от общия списък на уязвимостите (CVE), е грешка в софтуера, която един хакер може директно да се използва за получаване на достъп до системата или мрежата. По този начин, слабостта софтуер може да доведе до появата на уязвимост. Знаейки уязвимостта, хакерите могат да я използват при извършването на атаки на софтуер.

Формулирани са работни определения на следните понятия: слабост, уязвимост и атака на софтуер и е адаптиран придружаващия понятиен апарат съобразно спецификата на обхвата на кибер сигурността. Показана е връзка между понятията.

Представени са примери за слабости на софтуера. Посочена е основната цел на разработката на CWE, представени са

положителните страни на използването на списъка от слабости софтуера, отразени са някои исторически аспекти на CWE, подчертано е техническото въздействие на слабите страни на софтуера.

Анализирани са четири основни методология: точкуване на слабостите въз основа на мисията на организацията CWSS (Common Weakness Scoring System), CWRAF (Common Weakness Risk Analysis Framework) и Списък на CWE/SANS от 25-те най-опасни грешки в софтуера.

1. Точкуване на слабостите въз основа на мисията на организацията. Проектът на CWE предлага няколко подхода за определяне на приоритета на слабостите. Това позволява на служителите на компанията да се съсредоточат върху определена подгрупа от слабости въз основа на нуждите на организацията. Служителите на компанията могат да изучават, как да бъдат използвани тези техники, за подобряване на издръжливостта, надеждността и целостта на софтуера.
2. CWSS е система за точкуване на слабостите CWE. Тази система позволява на организациите да оценят честотата на откриваните грешки в кода на софтуерните приложения, като им дава възможност да бъдат смекчени тези недостатъци, а също така им позволява да се ориентират в качеството на бъдещите си покупки на софтуер и хардуер.
3. CWRAF е основата за анализ на риска на слабостите. Тази платформа позволява на организациите да анализират тези слабости CWE, които имат най-голямо отношение към спецификата на конкретния бизнес, мисия и внедряване на технологиите.
4. Списъкът на CWE/SANS от 25-те най-опасни грешки в софтуера. Това е списък на най-опасните грешки в софтуера. Списъкът предоставя на организациите приоритети за намаляване и избягване на слабостите.

Тук са дадени и исторически данни за CWE, посочени са кои организации и общности работят по CWE и ролята на компаниите и

изследователите в приложението на списъка при разработването на софтуер.

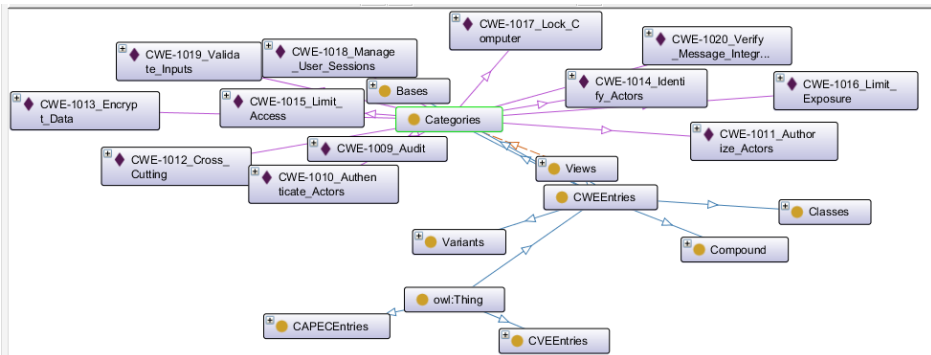
Има три основни концепции в CWE: проектиране, архитектура и изследвания.

1. Концепцията за разработка организира CWE слабостите в безопасността с помощта на принципи и понятия, които често се срещат при разработката; представянето е предназначено главно за разработчици и специалисти по оценка на качеството на софтуера.
2. Концепцията за архитектурата е за анализ на качеството на архитектурните решения на етапа на проектиране.
3. Концепцията за изследвания е предназначена да подпомага на академичните изследвания. Тя се отличава от първите две с високото ниво на абстракция. Фокусът на това представяне е формалното поведение на софтуера, специфичните примери за поведение са игнорирани.

Първата глава разглежда теоретичните аспекти на системата от слабости на софтуер CWE, включително методологията на системата, основните понятия и значими термини.

Във **Втората глава** “CWE VIEW: Architectural Concepts” се разглежда архитектурната концепция на CWE, т.е. как се анализират слабостите на софтуер от гледна точка на софтуерните архитекти.

Концепцията за архитектурата организира слабостите в съответствие с общата тактика архитектурата за сигурност. Тя е предназначена за подпомагане на архитектите в определяне на потенциалните грешки, които могат да бъдат допуснати при разработката на софтуера. Архитектурната концепция е представена като граф. Тя включва 12 категории, всяка от които се състои от класовете, бази, варианти и композити. Съставът на една категория е показан на фигура 1.



Фигура 1. Състав на категория

Архитектурната концепция (CWE 1008) се състои от следните категории:

- Одит (Audit) - (1009)
- Удостоверяване на актьорите (Authenticate Actors) - (1010)
- Даване на разрешения на актьорите (Authorize Actors) - (1011)
- Напречно разрязване (Cross Cutting) - (1012)
- Криптиране на данните (Encrypt Data) - (1013)
- Идентификация на актьорите (Identify Actors) - (1014)
- Ограничение на достъпа (Limit Access) - (1015)
- Ограничение на експозицията (Exposure Limit) - (1016)
- Заклучване на компютъра (Lock Computer) - (1017)
- Управление на сесиите на потребителя (Manage User Sessions) – (1018)
- Проверка на входните данни (Validate Inputs) – (1019)
- Проверка целостта на съобщенията (Verify Message Integrity) – (1020)

Изследвана е структурата на всеки елемент от системата, необходим за изграждането на онтологията.

Един запис в CWE включва следната информация:

- име на типа слабост
- описание на вида
- алтернативни термини за слабостта

- описание на поведението на слабостта
- описание на експлоатацията на слабостта
- вероятността за експлоатация на слабостта
- описание на последиците от експлоатацията
- потенциал за смекчаване на последиците от експлоатацията
- информация за отношенията с други възли (записи)
- първоначална таксономия
- код примери на езици и в архитектурури
- идентификатори на CVE уязвимости, за които има този тип слабост е приложима
- връзки

Направен е систематичен синтез на система за моделиране на слабости от гледна точка на архитектурната концепция.

Глава втора разглежда архитектурна концепция на системата от слабости софтуера: съдържание на концепцията, структура на записите в концепцията.

В глава Трета “OWL 2” е направена обосновка за избора на софтуерните средства за изграждане на онтологии, в частност анализирана е структурата на избрания език за онтологии.

За изпълнение на практическата част на проучване са избрани следните софтуерни средства за разработка на онтологии: редактор Protégé, който включва по презумпция езикът за описание на онтологии OWL 2, език за запитвания SPARQL и други технологии.

Показани предимствата на използването на езика за уеб онтологии OWL 2, и по-специално Манчестърския синтаксис на OWL 2, както и на средата Protégé.

Protégé е разработена от екип от учени, изследователи в медицинската информатика от Станфордския университет. Средата е предназначена за изграждане (създаване, редактиране и преглед) на онтологии в различни предметни области.

Езикът OWL 2 е език за онтологии в семантична мрежа с механизъм за формално описание на семантиката.

Езикът OWL дава възможност за изразяване на информацията от реалния света, а след това върху тази информация могат да се

правят изводи. Инструментите за OWL (машини за изводи) дават възможност автоматично да се правят изводи. При описанието на предметната област с езика OWL се предполага, че светът се състои от индивиди (предмети). Техните вътрешни свойства са описани чрез свойствата данни. Индивидите са взаимно свързани помежду си чрез обектните свойства. С помощта на OWL може да се групират в класове индивидите, които имат еднакви свойства.

Най-лесен за възприемане от хора е Манчестърският синтаксис на OWL 2.

Манчестърският синтаксис на OWL е приятелски синтаксис, разработен за писане изрази с класове (class expressions) в OWL, който, обаче, може да се използва и за пълното представяне на OWL-онтологията.

Структура на OWL-онтология:

- Всяка онтология има заглавие и тяло. В заглавната част се съдържа информация за самата онтология (версия, бележки) и импортираните в нея онтологии. След заглавието е тялото на онтология съдържащо описание на класовете, свойствата и индивидите.
- В OWL има две категории свойства: свойства-обекти и свойства-данни. Първите се свързват индивидите (екземпляри на класовете). Вторите свързват индивидите със стойности (данни№. И двата класа свойства са подкласове на клас `rdf:Property`.

За търсене на информация в онтологията се използва езика SPARQL. Към днешна дата език SPARQL е най-популярен сред езици за заявки към RDF-хранилищата.

SPARQL е език за заявки към данните представени от онтологиите с RDF, както и протокол за предаване на заявки и получаване на отговорите за тях. SPARQL е препоръката на W3C и една от технологиите на Семантичния Уеб.

Онтологията в Protégé се състои от следните елементи:

- Класове и подкласове. Класове са типове обекти, в някаква конкретна област. Класове и подкласове в Protégé се представят във вид на йерархия на наследяване.

- Свойствата данни са като атрибути на класове в онтологията и са свързани с определени типове данни (низ, число, дата и др.)
- Обектните свойства са като указатели към индивиди.
- Индивидът е екземпляр на клас. Онтологията, заедно с набора от индивиди формализира знанията в определена област.
- Домейнът е клас, за който е приложимо дадено свойство.
- Областта на стойностите задава типът на стойността, която може да приеме дадено свойство.
- Типът на данни е някои от стандартните типове.

Всяка онтология може да се представят във вид на граф.

Третата глава е посветена на средствата разработка на онтология – направен е преглед на средата Protégé, подробно е анализиран от езика за уеб онтологии OWL. Подробно е представена структурата на онтологията в Protégé и езика OWL.

В **Четвърта глава** “Проектиране на онтологията“ е представено решението, показан е процесът на изграждане на онтологията на системата от слабости на софтуера по отношение на архитектурната концепция.

Процесът на проектиране на онтологията е разделен на следните етапи.

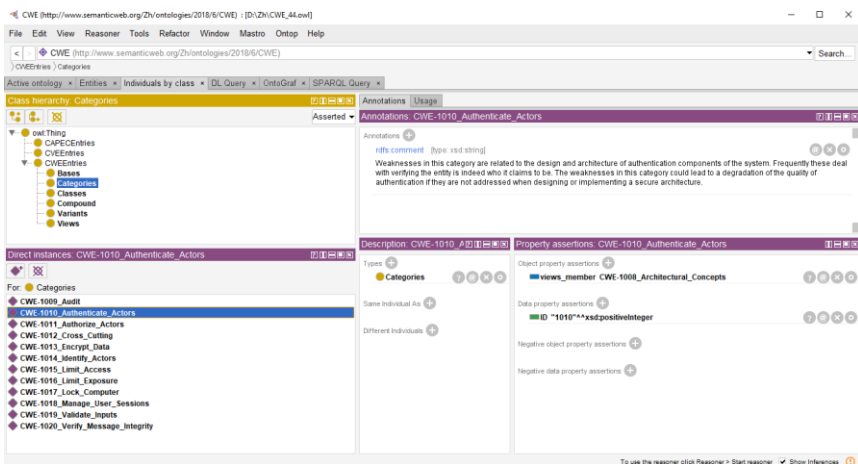
1. Проектиране на онтологията, а именно:
 - определяне на класове в онтологията;
 - организиране на класове в йерархия (базов клас → клас).
 - определяне на свойствата данни и техните допустимите стойности.
2. Създаване на форми за въвеждане на екземпляри.
3. Попълване с екземпляри на класа “CWEEEntries”.
4. Попълване с екземпляри на класа “CVEEntries”.
5. Попълване с екземпляри на класа “CAPECEEntries”.
6. Проверка на съгласуваността на онтологията.

Създадената онтология се състои от слабости, уязвимости в сигурността и образци на злонамерени атаки.

Онтологията е реализирана от гледната точка на архитектурни концепции. В нея основен клас е CWEEntries. Този клас е от тип View и включва 12 подкласа (category).

Класът Categories представя записи на CWE. Той съдържа набор от други CWE записи, които имат една общи характеристики.

На Фигура 2 е показано съдържанието на категория в средата Protégé.



Фигура 2. Структура на категория.

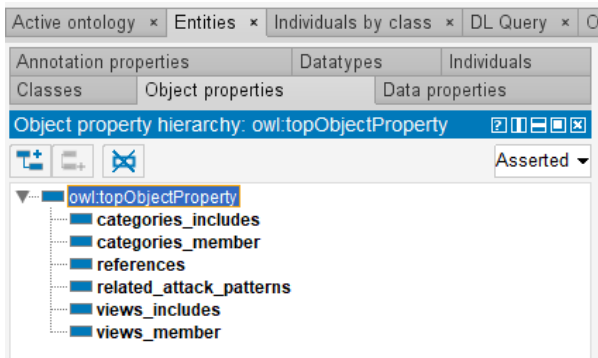
Категорията може да включва класове, бази, варианти и възможно композити.

Категорията има от описание (кратка характеристика – във вид на анотация), ID на категорията – свойство данни (data property), обектно свойство – views_member. Това свойство свързва категорията с нейните класове, бази, варианти и композити, т.е. с включени в нея слабости.

Основните обектни свойства за всички видове CWEEntries са:

- references – свързва слабостта с уязвимостите от този тип;
- categories_member – свързва категорията с нейните класове, бази, варианти и композити;
- related_attack_patterns – свързва слабостта с възможните образци на атаки.

На фигура 3 е представена йерархията на обектните свойства в разработената онтология.



Фигура 3. Йерархията на обектните свойства.

Обектните свойства имат домейн и област на стойностите. Те свързват индивидите от домейна с индивидите от областта на стойностите.

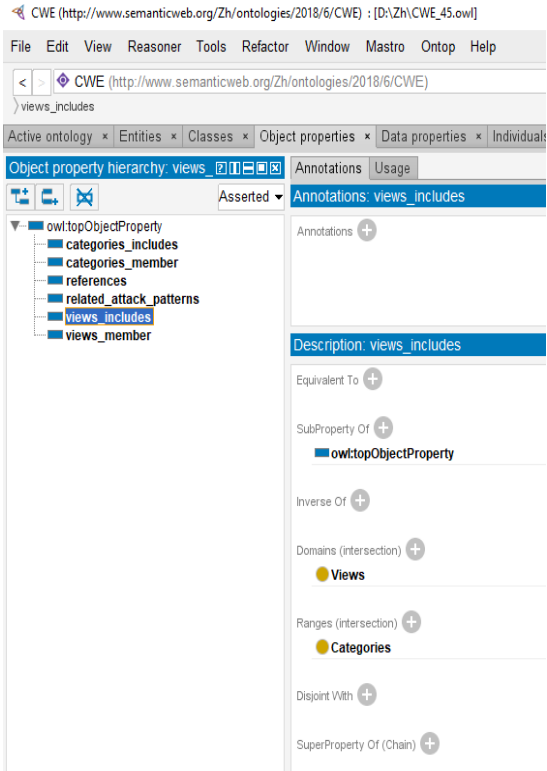
Да вземем примера от Фигура 4, `views_includes` свързва индивиди от клас `Views` (домейн) индивиди набор от клас `Categories`.

Основните свойства данни са:

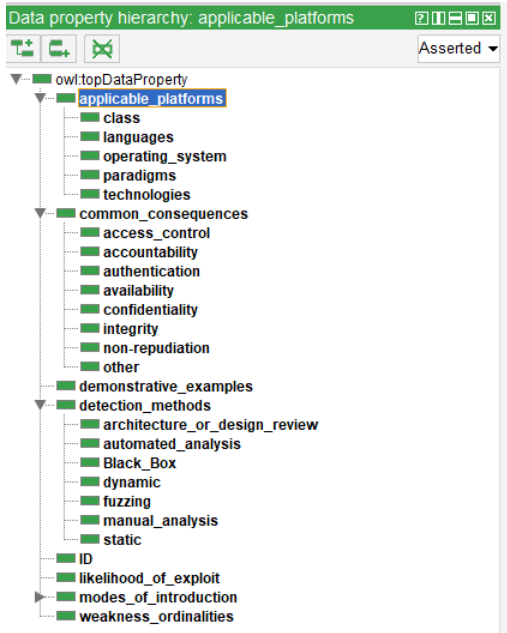
- `id`
- `modes_of_introduction`
- `common_consequences`
- `likelihood_of_exploit`
- `demonstrative_examples`
- `applicable_platforms`,
- `weakness_ordinalities`
- `detection_methods`.

Смисълът и значението на свойствата данни е определен от описаните по-горе полета в записите на CWE базата от данни.

На фигура 5 е представена йерархията на свойствата данни и в онтологията.



Фигура 4. Домейн и област от стойности в онтологията.



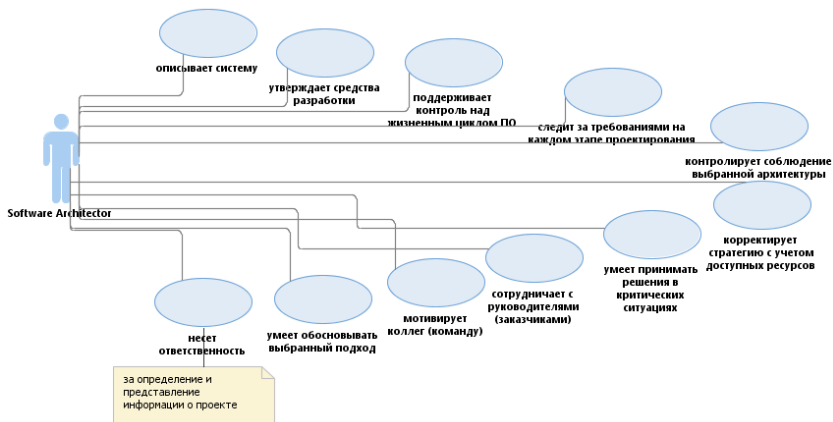
Фигура 5. Свойствата данни в онтологията.

В Пета глава „Използването на онтологията от софтуерния архитект“ са представени примери за приложението на разработената онтология за системата от слабости.

Анализирани основните задължения на софтуерния архитект и функциите са представени в диаграма на потребителските случаи.

Софтуерният архитект е лицето, чиято основна дейност е да ръководи процеса на проектиране, изграждане, разработка и поддръжка на софтуера.

Основните задължения на софтуерен архитект са показани на Фигура 6.



Фигура 6. Функции на софтуерния архитект

Основните функции на софтуерния архитект са:

1. Разработване на варианти на архитектура на софтуера;
2. Оценка на изискванията към софтуера и избор на вариант на архитектура на софтуера;
3. Документиране на архитектурата на софтуера и реализацията ѝ;
4. Оценка на възможностите за реализация на архитектурата и определяне на ключовите сценарии;
5. Управление на методите и начините за поддръжка на софтуера;
6. Контрол при избора на вариант на софтуерната архитектура и при реализацията и поддръжката ѝ.

В тази глава са демонстрирани примери за използването на онтологията от софтуерния архитект чрез различни видове заявки.

Има четири вида заявки:

- „по подразбиране“;
- с използване литерали;
- с индивиди на класове;
- с обектните свойства на клас.

Всяка SPARQL заявката има следната задължителна заглавна част следвана от SELECT шаблон генериран от Protégé:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object }

```

Изпълнението на тази заявка дава структурата онтологията в табличен вид:

	subject	object
Views		CWEEnties
Variants		CWEEnties
Compound		CWEEnties
Classes		CWEEnties
Bases		CWEEnties
Categories		CWEEnties

Заявка 1. Кои обекти са свързани със свойството categories_includes?

```

PREFIX db: <http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?x ?property ?y
WHERE {
    ?property rdfs:subPropertyOf* db:views_includes.
    ?property rdfs:domain ?x .
    ?property rdfs:range ?y . }

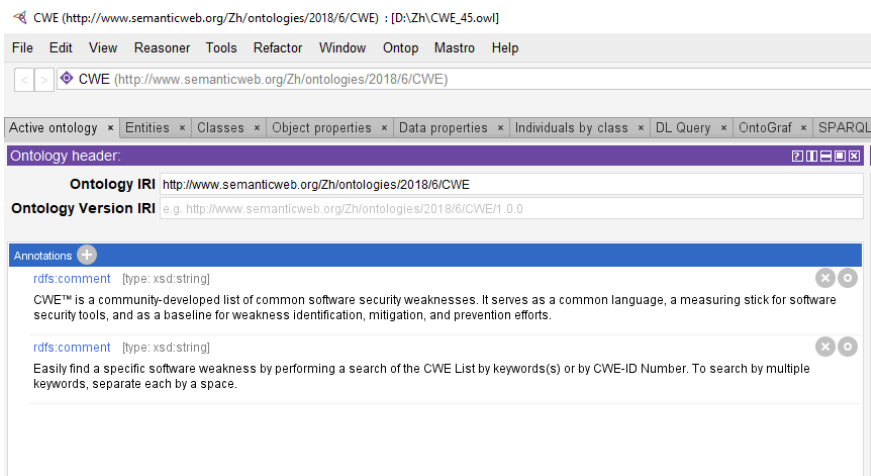
```

Резултат:

	x	property	y
Categories	categories_includes		● Bases or Classes or Compound or Variants

Заявките по-просто се задават когато се използват префикси към онтологията, в които се търси. Освен стандартния набор от

префикси може потребителят да дефинира свой за търсене в определена онтология, на примера тава е префиксът db.



Описание на префикса може да се вземе от адресната лента на онтологията.

Резултатите от изпълнението заявките се извеждат в табличен формат.

Заявка 2. Какви стойности има свойството за категории likelihood_of_exploit?

```
PREFIX db: <http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#>
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
SELECT ?t ?y
```

```
WHERE {
    ?x db:categories_member ?t.
    ?x db:likelihood_of_exploit ?y }
```

	t	y
CWE-1019_Validate_Inputs		"Medium""<http://www.w3.org/2001/XMLSchema#string"
CWE-1019_Validate_Inputs		"High""<http://www.w3.org/2001/XMLSchema#string"
CWE-1011_Authorize_Actors		"High""<http://www.w3.org/2001/XMLSchema#string"
CWE-1011_Authorize_Actors		"Medium""<http://www.w3.org/2001/XMLSchema#string"
CWE-1019_Validate_Inputs		"High""<http://www.w3.org/2001/XMLSchema#string"
CWE-1019_Validate_Inputs		"High""<http://www.w3.org/2001/XMLSchema#string"
CWE-1019_Validate_Inputs		"Low""<http://www.w3.org/2001/XMLSchema#string"
CWE-1010_Authenticate_Actors		"High""<http://www.w3.org/2001/XMLSchema#string"
CWE-1010_Authenticate_Actors		"High""<http://www.w3.org/2001/XMLSchema#string"
CWE-1010_Authenticate_Actors		"High""<http://www.w3.org/2001/XMLSchema#string"
CWE-1019_Validate_Inputs		"High""<http://www.w3.org/2001/XMLSchema#string"
CWE-1019_Validate_Inputs		"High""<http://www.w3.org/2001/XMLSchema#string"
CWE-1020_Verify_Message_Integrity		"Medium""<http://www.w3.org/2001/XMLSchema#string"

Execute

Следват примери с използване на разработената онтология.

Пример 1. Фирма желае да закупи нова система за управление на база данни. В краткия списък са останали DB2 и Oracle. Ръководството възлага на мениджъра по кибер сигурността да изследва и двете системи на уязвимости.

Първо, мениджърът проверява какви уязвимости са регистрирани в онтологията за DB2. Това той прави с помощта на следващата заявка:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#>
SELECT ?cve ?comment
WHERE {
    ?cve    a :CVEEntries;
           rdfs:comment ?comment
    FILTER regex(?comment, '.db2.', 'i')}
```

След това, той прави подобно търсене за система за управление на бази от данни Oracle със заявката:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#>
SELECT ?cve ?comment
WHERE {
    ?cve    a :CVEEntries;
```

```
    rdfs:comment ?comment
FILTER regex(?comment, '.oracle.', 'i')
```

Пример 2. Групата по кибер сигурността установява нахлуване в мрежата чрез HTTP протокола. Първо, групата извежда всички атаки на базата на HTTP протокола, като използва заявката:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#>
SELECT ?capec ?comment
WHERE {
    ?capec a :CAPECEntries;
          rdfs:comment ?comment
    FILTER regex(?comment, ".HTTP.", "i")}
```

По-нататък, групата проверява кои от тези атаки са свързани с идентифицирани уязвимости:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#>
SELECT ?cve ?description
WHERE {
    ?capec a :CAPECEntries;
          rdfs:comment ?comment
    FILTER regex(?comment, ".HTTP.", "i").
    ?cve :related_attack_patterns ?capec;
        :references ?cve.
    ?cve rdfs:comment ?description }
```

Списък на тези уязвимости е твърде голям и някои от тях не са свързани с конкретните системи.

В организацията е внедрена технологията PHP и групата се фокусира в по-нататъшните си изследвания върху тази технология:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#>
SELECT ?cve ?description
WHERE {
    ?capec a :CAPECEntries;
        rdfs:comment ?comment
    FILTER regex(?comment, ".HTTP.", "i").
    ?cwe :related_attack_patterns ?capec;
        :references ?cve.
    ?cve rdfs:comment ?description
    FILTER regex(?description, ".PHP.", "i").}

```

Сега групата разглежда списъка на откритите уязвимости:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#>
SELECT DISTINCT ?cwe
WHERE {
    ?capec a :CAPECEntries;
        rdfs:comment ?comment
    FILTER regex(?comment, ".HTTP.", "i").
    ?cwe :related_attack_patterns ?capec;
        :references ?cve.
    ?cve rdfs:comment ?description
    FILTER regex(?description, ".PHP.", "i"). }

```

Пример 3. На сайта е направена XSS атака. Сайтът е реализиран с Java технологията. Cross-Site Scripting (XSS) е атака на уеб приложението, която използва слабости в неправилна обработка на входните данни и това позволява да бъде изпълнен произволен скрипт (JavaScript, VBScript) в контекста на източника (origin) на засегнатото уеб приложение.

Група разработчици трябва да уточни видовете уязвимости използвани при XSS атаките:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#
SELECT DISTINCT ?cve ?description
WHERE {
    ?capec a :CAPECEntries;
          rdfs:comment ?comment
    FILTER regex(?comment, ".XSS.", "i").
    ?cve   :related_attack_patterns ?capec;
          :references ?cve.
    ?cve   rdfs:comment ?description
    FILTER regex(?description, ".Java.", "i"). }

```

Резултатът от изпълнението на заявката е:

CVE	description
CVE-2006-3211	"Cross-site scripting (XSS) vulnerability in sign.php in cjGuestbook 1.3 and earlier allows remote attackers to инжектирайте Javascript code via a javascript URI in an img bbcode tag in the comments parameter." ^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2006-4308	"Multiple cross-site scripting (XSS) vulnerabilities in Blackboard Learning System 6, Blackboard Learning and Community Portal Suite 6.2.3.23, and Blackboard Vista 4 allow remote attackers to инжектирайте arbitrary Javascript, VBScript, or HTML via (1) данни, (2), vbscript, and (3) malformed javascript URIs in various HTML tags when posting to the Discussion Board." ^^<http://www.w3.org/2001/XMLSchema#string>

Пример 4. В софтуерна компания е извършена атака чрез IP-адреси. Хакната е базата данни с информация за имена на клиентите, адресите им, ел. поща, пароли, снимки и други. Паролите в базата данни са криптирани с помощта на хеш-

функцията MD5. Програмистите трябва да анализират възможните уязвимости и слабости при създадената ситуация:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#
SELECT DISTINCT ?cve ?description
WHERE {
    ?capec a :CAPECEntries;
          rdfs:comment ?comment
    FILTER regex(?comment, ".П.П.", "i").
    ?cwe   :related_attack_patterns ?capec;
          :references ?cve.
    ?cve   rdfs:comment ?description
    FILTER regex(?description, ".MD5.", "i").}

```

Резултатът е:

CVE	description
CVE-2007-4961	"The login_to_simulator method in Linden Lab Second Life as used by the secondlife:// protocol handler and possibly other Second Life login mechanisms, sends an MD5 hash in cleartext in the passwd field, which allows remote attackers to login to an account by sniffing the network and then sending this hash to a Second Life authentication server." ^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2005-0408	"CitrusDB 0.3.6 and earlier generates easily predictable MD5 hashes of the user name for the id_hash cookie, which allows remote attackers to bypass authentication and gain privileges by calculating the MD5 checksum of the user name combined with the "boogaadeeboo" string which is hard-coded in the \$hidden_hash variable." ^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2002-2058	"TeeKai Проследяване Online 1.0 uses weak encryption of web usage statistics in data/userlog/log.txt, which allows remote attackers to identify IP's visiting the site by dividing each octet by the MD5 hash of '20'."

	^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2007-6013	"Wordpress 1.5 through 2.3.1 uses cookie values based on the MD5 hash of a password MD5 hash, which allows attackers to bypass authentication by obtaining the MD5 hash from the user database, then generating the authentication cookie from that hash." ^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2005-2946	"The default configuration on OpenSSL before 0.9.8 uses MD5 for creating message digests instead of a more cryptographically strong algorithm, which makes it easier for remote attackers to forge certificates with a valid signature certificate authority." ^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2008-3188	"libxcrypt in SUSE openSUSE 11.0 uses the DES algorithm when the configuration specifies the MD5 algorithm, which makes it easier for attackers to conduct на brute-force attacks against hashed passwords." ^^<http://www.w3.org/2001/XMLSchema#string >
CVE-2001-0497	"dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-MD5 shared secret key file used for DNS Transactional Signatures (TSIG), which allows attackers to obtain the keys and perform dynamic DNS updates." ^^<http://www.w3.org/2001/XMLSchema#string>

Пример 5. Компанията се сблъсква с нарушаване на сигурността на нейната база от данни. Информацията не е била надлежно защитена и на сайта на компанията не се поддържа безопасна SSL връзка. Независимо от факта, че при взлом не са били разкрити данни за кредитни карти и лични идентификационни данни, дейността на дружеството на фондовата борса е временно спряно, тъй като опасността от последиците на тази атака е твърде голяма. Паролите в базата данни са криптирани с помощта на хеш-функцията MD5. Програмистите трябва да се анализират възможните узвимости и слабости.

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

```

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/Zh/ontologies/2018/6/CWE#>
SELECT DISTINCT ?cve ?description
WHERE {
    ?capec a :CAPECEntries;
          rdfs:comment ?comment
    FILTER regex(?comment, ".SSL.", "i").
    ?cwe   :related_attack_patterns ?capec;
          :references ?cve.
    ?cve   rdfs:comment ?description
    FILTER regex(?description, ".MD5.", "i"). }

```

Резултатът е:

CVE-2005-0408	"CitrusDB 0.3.6 and earlier generates easily predictable MD5 hashes of the user name for the id_hash cookie, which allows remote attackers to bypass authentication and gain privileges by calculating the MD5 checksum of the user name combined with the "boogaadeeboo" string which is hard-coded in the \$hidden_hash variable." ^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2002-2058	"TeeKai Проследяване Online 1.0 uses weak encryption of web usage statistics in data/userlog/log.txt, which allows remote attackers to identify IP's visiting the site by dividing each octet by the MD5 hash of '20'." ^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2007-6013	"Wordpress 1.5 through 2.3.1 uses cookie values based on the MD5 hash of a password MD5 hash, which allows attackers to bypass authentication by obtaining the MD5 hash from the user database, then generating the authentication cookie from that hash." ^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2005-2946	"The default configuration on OpenSSL before 0.9.8 uses MD5 for creating message digests instead of a more cryptographically strong algorithm, which makes it easier for remote attackers to forge certificates with a valid signature certificate authority." ^^<http://www.w3.org/2001/XMLSchema#string>
CVE-2008-3188	"libxcrypt in SUSE openSUSE 11.0 uses the DES algorithm when the configuration specifies the MD5 algorithm, which makes it easier for attackers to conduct на brute-force attacks against hashed passwords."

Разгледани по-горе примери демонстрират начините за използване на разработена онтология на практика.

Онтологичният файл е публикуван на сайта на github чрез връзка: Онтологичният Файл е публикуван на сайта на github чрез връзка: Онтологичният Файл е публикуван на сайта на github чрез връзка: <https://github.com/Sartabanova/Ontology-project.git>.

ЗАКЛЮЧЕНИЕ

Проблемът със сигурността на компютърните системи е един от основните в нашето съвремие, тъй като наличието на уязвимости в компютърни системи дава възможност за извършване на компютърни атаки и вирусни епидемии, което може да нанесе големи поражения. С оглед на това и с нарастване сложността на компютърните програми и развитието на технологиите е необходимо непрекъснато подобряване на методите за контрол, изпитание и тестване. Изследванията в тази област винаги е актуален проблем.

Основната цел на дисертация в научните изследвания е постигната: това е изучена и изследвана е система от слабости на софтуера CWE и е разработена онтология за тази система от гледната точка на софтуерния архитект.

Също така са решени следните задачи:

1. Изследвана е структурата от слабости на CWE.
2. Задълбочено е изследвана гледната точка CWE VIEW: Architectural Concepts.
3. Изследвани са езика OWL за уеб онтологии, RDF и среда за разработка на онтологии Protégé.
4. Създадена е OWL онтология на базата на CWE Architectural Concepts.
5. Демонстрирани са възможностите за използване на разработената онтология от софтуерни архитекти, програмисти и университетски преподаватели.

НАУЧНИ ПРИНОСИ НА ДИСЕРТАЦИОННИЯ ТРУД

На базата на проведен теоретичен анализ на системата от слабости CWE предложена онтология от гледната точка на архитектурната концепция, което дава възможност за използването и от съответните заинтересовани лица.

Основни резултати дисертация на труда:

1. Изследвана е система от слабости CWE от гледната точка на архитектурата;
2. Анализирана система е системата от слабости CWE от гледна точка на архитектурата;
3. Изследвани средства са средствата за описание на онтологии на базата на RDF и по-специално на OWL;
4. Анализирани са средства за описание на онтологии на базата на RDF и специално на OWL. Избрана е средата Protégé за по-разработка на онтологията като пряко следствие от този анализ;
5. Създадена онтология с използване на OWL и Protégé за слабостите в CWE. Последната е одобрен от индустриален стандарт таксономизации слабости.
6. Показано е използването на онтология с редица примери по задачи с прилагането на SPARQL.

ПУБЛИКАЦИИ, СВЪРЗАНИ С ДИСЕРТАЦИОННИЯ ТРУД

1. Сартабанова Ж.Е., Димитров В.Т. Обзор системы слабостей программного обеспечения CWE // Труды международной научной конференции «Проблемы прикладной математики и информатики», Актобе, 10-11 ноября 2017 г.
2. Сартабанова Ж.Е., Димитров В.Т., Сарсимбаева С.М. О системе слабостей программного обеспечения // Материалы VIII Международной научной конференции «Проблемы дифференциальных уравнений, анализа и алгебры». Актобе, 1 ноября 2018, 244-247, <http://arsu.kz/media-files/kz/gylym-meni-innovaciylar/aomu-gylymi-basylymdary/Differential.pdf#2>
3. Sartabanova Zh, Dimitrov V, Modelling of CWEs on the CWE-287 example // CEUR Workshop Proceedings, Vol-2464, 2019.
4. Sartabanova Zh, Dimitrov V.T., Sarsimbaeva S.M., Applying the knowledge base of CWE weaknesses in software design // "Journal of Mathematics, Mechanics and Computer Science", Al-Farabi Kazakh National University, <https://bm.kaznu.kz/index.php/kaznu/issue/view/75>, 2020.