

Пловдивски университет „Паисий Хилендарски“
Факултет по математика и информатика
Катедра „Компютърни системи“

Нина Станчева Станчева

Семантични модели за Виртуално Образователно Пространство

АВТОРЕФЕРАТ

за присъждане на образователна и научна степен „доктор“
по област на висше образование

4. Природни науки, математика и информатика,
професионално направление

4.6. Информатика и компютърни науки,
Докторска програма Информатика

Научен ръководител: доц. д-р Ася Стоянова-Дойчева

Рецензенти: доц. д-р Ася Стоянова-Дойчева
проф. д-р Асен Кънчев Рахнев
проф. д-р Мария Петкова Христова
проф. д-р Аврам Моис Ескенази
доц. д-р Ирина Александровна Радева

Пловдив
2020

Дисертационният труд е обсъден и насочен за защита пред научно жури на заседание на катедра „Компютърни системи“ при Факултета по математика и информатика на ПУ „Паисий Хилендарски“ на 24.09.2020 г.

Дисертационният труд съдържа 159 страници. Библиографията включва 186 източника. Броят на авторските публикации е 6.

Защитата на дисертационния труд ще се състои на 22.12.2020 от 10:30 ч. в заседателната зала на Нова сграда на ПУ „Паисий Хилендарски“, гр. Пловдив.

Автор: Нина Станчева Станчева

Заглавие: Семантични модели за Виртуално Образователно Пространство

Тираж: 30 бр.

Пловдив, 2020 г.

СЪДЪРЖАНИЕ

ВЪВЕДЕНИЕ (Глава 1).....	4
Цели на дисертацията	4
Структура на дисертацията	6
ТЕКУЩО СЪСТОЯНИЕ В ОБЛАСТТА (Глава 2)	7
Семантични модели	7
Онтологиите като семантични модели	7
Необходимост от семантика в електронното обучение.....	7
Електронно тестване в електронното обучение	8
ViPS (Virtual Physical Space)	8
ФОРМАЛЕН МОДЕЛ ЗА АВТОМАТИЧНО ГЕНЕРИРАНЕ НА ТЕСТОВИ ВЪПРОСИ (Глава 3).....	9
СЕМАНТИЧНИ МОДЕЛИ ЗА ЕЛЕКТРОННИ ТЕСТОВЕ ВЪВ ВОП (Глава 4)	13
Реализация на онтология в областта на езика UML	13
Приобщаване на съществуващи онтологии за целите на електронното тестване	15
ИНТЕЛИГЕНТНА СРЕДА ЗА ЕЛЕКТРОННИ ТЕСТОВЕ	16
Архитектура	16
Автоматично генериране на въпроси.....	18
Съставяне на тестове	21
Проверка на потребителските отговори	21
Реализация на средата за електронни тестове	22
Апробация на средата за електронни тестове	24
ЗАКЛЮЧЕНИЕ (Глава 6).....	25
ПУБЛИКАЦИИ ПО ДИСЕРТАЦИОННИЯ ТРУД	27
БЛАГОДАРНОСТИ	28
БИБЛИОГРАФИЯ	29

ВЪВЕДЕНИЕ (Глава 1)

В наши дни компютърните технологии са неизменна част от ежедневието на хората. В съвременното електронното обучение присъства в голяма част от образователните институции и организации. За да бъдат полезни и ефективни, тези системи трябва да са гъвкави и автономни, да отговарят на нуждите на обучаващите и обучаемите. Основна цел на такъв вид софтуер е да предоставя възможности за съхраняване на знания, поднасяне на информация по правилен за обучението начин, подпомагане при усвояването на нов материал. Поради това представянето на учебния материал и проверката на наученото е необходимо да се модернизират чрез използването на автоматизирани системи и прилагането на семантика. Развитие в тази насока може да се постигне чрез системи, изградени от проактивни софтуерни единици, наречени агенти, които са способни на самоуправление. Чрез тях информацията, свързана с учебния процес, може да се обработва ефективно. За автоматизиране на основни учебни дейности, като проверка на знанията и провеждане на изпити, е необходимо учебният материал да бъде представен в подходяща форма. Семантичните модели са подходящ начин за структуриране на знанията, така че това да дава възможност за автоматична обработка и предоставяне на функционалности за електронното обучение, удовлетворяващи нуждите в образованието.

Цели на дисертацията

Целите на дисертацията са разделени в три основни групи, описани по-долу.

1. Дефиниране на формален модел за автоматично генериране на въпроси.

Първата цел на дисертацията се състои в създаването на формален модел за автоматично генериране на въпроси. Този модел включва формалното описание на знанията за домейна като хранилище за информация. Формулира структурата на това хранилище и как се представят знанията, за да бъдат в подходяща форма за използване при генериране на въпроси. Моделът дефинира целия процес за автоматично генериране на тестови въпроси. Това включва описание на процесите за извличане на знанията от хранилището и тяхната обработка. Дефинират се алгоритмите, необходими за генериране на различни типове въпроси от тези знания. Идеята на модела е да бъде използван за реализацията на тестови системи в електронното обучение.

2. Създаване на семантичен модел с цел използването му в електронното тестване.

Втората цел е да се създаде семантичен модел, който да бъде използван като база знания на среда за електронни тестове. Първоначално се дефинира концепция за създаването на такъв семантичен модел. За съставянето му се изгражда онтология в областта на езика UML (Unified Modeling Language). Реализирането на онтология се осъществява на базата на формалния модел, създаден за първата цел на дисертацията.

3. Създаване на среда за електронно тестване

Третата цел на дисертацията е реализиране на среда за електронни тестове. Средата използва семантичния модел, създаден за втората цел на дисертацията, като база знания. Чрез знанията от базата системата генерира автоматично въпроси в областта на езика UML. Реализира се функционалност за генериране на различни типове въпроси, взаймствани от стандарта QTI 2.1 (Question & Test Interoperability) [1]. Въпросите се съставят за тестове, разделени тематично, съобразено с поднасянето на учебния материал. Средата включва и генериране на тест с персонализирани въпроси, където персонализацията се основава на представянето на потребителя в предишните тестове, на които се е явил. Системата реализира и автоматична проверка на отговорите на всеки тест. Тя се осъществява независимо от генерирането на въпросите чрез използването на базата знания. Тук се включва планирането и имплементацията на средата. Средата реализира модела, дефиниран за първата цел на дисертацията. Третата цел се разделя на три под цели:

- автоматично генериране на въпроси – проектиране и създаване на агент, който реализира алгоритмите за генериране на различни типове въпроси от цел 1;
- автоматично проверяване на отговорите – проектиране и създаване на агент, който приема, обработва и проверява коректността на отговорите, дадени от потребителя;
- съставяне на тестове – реализация на комуникацията между двата агента; създаване на база данни за съхранение на резултатите от тестовете; съставяне на тематични тестове; имплементация на генериране на персонализиран тест чрез информацията от базата данни и базата знания.

Структура на дисертацията

Дисертационният труд се състои от шест глави, библиография и приложение:

- **Глава 1** запознава с темата и обхвата на дисертацията. Описва основния подход, използван в работата и основните цели в нея;
- **Глава 2** представя проучването в областта и прави преглед на съществуващите разработки и изследвания в областта. Дава кратък преглед на състоянието, в което се намира научната област;
- **Глава 3** дефинира формален модел за автоматично генериране въпроси;
- **Глава 4** представя семантичен модел за електронно тестване – представяне на концепция за създаване и описване на изграждането на онтология в областта на езика UML;
- **Глава 5** представя проектирането и разработката на прототип на среда за автоматично генериране на тестове за електронното обучение;
- **Глава 6** обобщава свършената работа върху поставените цели в увода на дисертацията.

ТЕКУЩО СЪСТОЯНИЕ В ОБЛАСТТА (Глава 2)

Семантични модели

Семантичен модел (Semantic data model – SDM) е описание от високо ниво на бази данни, които са семантично базирани. Това е структуриращ формализъм, който цели да обхване повече от същността и значението на областта на дадено приложение от колкото останалите съвременни модели за бази данни позволяват. Семантичното моделиране е метод за структуриране на данните, така че да включва семантична информация и смислово значение. Хъл и Кинг представят обширен обзор на семантичните модели към момента [3] като изтъкват три видни модела. ER (Entity-Relationship Model) [4] е смятан за един от първите семантични модели. В последствие моделът се превръща в абстрактен модел, дефиниращ структура от данни, от която може да се имплементира база данни или онтология. FDM (Functional Data Model) [5] е считан за първия семантичен модел, чийто подход е базиран върху функционални връзки (атрибути). SDM [2] използва групиращ конструктор и производна схема на компонентите.

Онтологиите като семантични модели

В компютърните науки с онтология се обозначава продукт, който е създаден с цел да стане възможно моделирането на знания за определена област. За пръв път понятието е използвано и дефинирано като технически термин от Том Грубер [6], който определя онтологията като спецификация на концептуализацията, като описание на понятията и взаимовръзките в даден домейн.

OWL (Web Ontology Language) [7] е една от последните препоръки на W3C (World Wide Web Consortium) за разработка на онтологии. OWL онтологията е семантичен модел, абстракция, създадена за целените знания.

Необходимост от семантика в електронното обучение

Системите за е-обучение трябва да се възползват от семантичните възможности, оперативна съвместимост, онтологии и семантични анотации [8]. Онтологиите в е-обучението спомагат за споделяне и използване на знанията – представяне на общото разбиране за преподаваната информация и достигането му до хора и системи [9]. Дават възможност за ясни предположения за областта и разделят учебните знания от оперативната информация.

Електронно тестване в електронното обучение

Тестовите са неизменна част в електронното обучение. Необходими са за улеснение на обучаващи и обучаеми при полагане на изпити, подготовка за изпити и самооценка. Съществува голямо разнообразие от системи, предлагащи, подобни услуги, които са широко използвани. Следваща стъпка за електронното тестване е изготвянето на въпросите да бъде превърнато в напълно автоматизиран процес, който отнема минимално време и усилия. Липсата на такава услуга налага нуждата от изготвяне на съдържанието и отговорите на всеки въпрос, ръчното му въвеждане и структуриране. Поради това автоматичното генериране на тестови въпроси е основна част от бъдещето на средите за електронно обучение. Създаването на качествени въпроси е трудна задача за разработката на подобни системи и изисква проучване и експериментиране с различни подходи. Първата стъпка към автоматичното генериране на въпроси за електронно тестване е определяне на най-подходящия ресурс от знания, който да бъде използван при създаването на тези въпроси. За тази цел съществуват различни възможности. Онтология в дадена предметна област, съдържаща знания необходими за обучение, може да се използва като ресурс за генериране на въпроси за електронни системи в образованието [10]. За да бъдат генерираните смислени и четими въпроси, онтологиите трябва да бъдат създадени с тази цел, да съдържат подходящи семантични връзки. Важно е структурата на онтологията да бъде съобразена със стратегията за съставянето на въпросите.

ViPS (Virtual Physical Space)

Virtual Physical Space [11] е пространство, разработено в катедра „Компютърни системи“ на Факултета по математика и информатика, което представя референтна архитектура приложима в различни области като например културно-историческо наследство [12], интелигентни градове [13] и електронно обучение, интелигентно селско стопанство [14]. Предложената архитектура на разработения прототип за генериране и проверка на въпроси на базата на онтология е приложение на референтната архитектура на ViPS.

ФОРМАЛЕН МОДЕЛ ЗА АВТОМАТИЧНО ГЕНЕРИРАНЕ НА ТЕСТОВИ ВЪПРОСИ (Глава 3)

Представен е базов подход за генериране на различни видове тестови въпроси като за целта се създава и използва структурирано учебно съдържание [15]. Подходът се състои от следните три етапа:

- етап 1: структурирано представяне на знанията, които са включени в учебното съдържание за дадена дисциплина. Извършва се така, че знанията да бъдат сведени до отделни учебни единици със смислови връзки между тях;
- етап 2: извличане на отделните компоненти, изграждащи структурата от учебно съдържание;
- етап 3: генериране на въпроси чрез използване на извлечените компоненти от етап 2.

В съответствие с представения подход е разработен формалния модел QGM (Question Generation Model). Той се състои от три нива като всяко от тях се основава на съответния етап от подхода:

- ниво 1: домейн – моделиране на базовите елементи и връзките между тях, изграждане на структурата на поддържаното учебно съдържание;
- ниво 2: екстрактори – моделиране на възможностите за извличане на елементи от структурата на учебния материал, на чиято основа ще се генерират въпросите;
- ниво 3: генератор – моделиране на генерацията на тестови въпроси.

В ниво 1 от формалния модел се представя домейнът – знанията в областта, за които ще се провежда тестът. Това ниво моделира знанията от домейна като хранилище. На Фигура 1 са показани множествата, дефинирани в него. Изгражданата структура се състои от концепции, атрибути и аксиоми. Концепциите са единици, представящи различните понятия, които съществуват в предметната област (домейна). Те са структурирани в типизирана йерархична структура, което дава информация за семантичния вид на всяка концепция. Същността на атрибутите е да показват връзките между понятията. Аксиомите са специфични за домейна правила. Тяхната цел е да дефинират концепциите и да описват домейна. Приемаме, че тези описания са коректни, т.е. аксиомите винаги са верни. Аксиомите се конструират от два основни типа елементи – базова концепция и рестрикция. Базовата концепция представя понятието, което се дефинира от аксиомата и се характеризира посредством рестрикцията.

Дефинираме следните множества:

- $A \equiv A_S \cup A_E \cup A_R \cup A_D$ – множество на аксиомите, описващи домейна, където:
 - A_S : множество на подклас аксиомите;
 - A_E : множество на аксиомите за еквивалентност;
 - A_R : множество на аксиомите за обхват;
 - A_D : множество на аксиомите за несъответствие;
- C : множество на концепциите;
- P : множество на атрибутите;
- R : множество на рестрикциите;
- Ann : множество на анотациите.

Дефинираме следните релации:

- $A \equiv C \times R$: структура от аксиоми $a \in A$, $a = (c, r)$, където $c \in C$ са базови концепции и $r \in R$ са рестрикции;
- $R \equiv 2^P \times C$: $r \in R$ се конструира от определен брой атрибути и концепции. наречени ограничителни концепции.

Фигура 1. Множества в ниво 1 на QGM

Множества:

- Axi – множество на избраните аксиоми по определен критерии;
- K – множество на критериите за избор на аксиоми;
- $T \equiv \{t_I, t_D\}$ – множество на видовете въпроси, което се състои от два елемента, където: t_I – въпросителен тип; t_D – съобщителен тип.

Екстрактори:

- $selectAxi: A \rightarrow Axi$ – извлича аксиоми по зададен критерии;
- $extrBaseConcept: A \rightarrow C$ – извлича базовата концепция от дадена аксиома;
- $extrProperty: A \rightarrow 2^P$ – извлича атрибута от дадена аксиома;
- $extrRestrConcept: A \rightarrow C$ – извлича ограничителните концепции от рестрикцията на дадена аксиома;
- $extrRange: P \rightarrow C$ – извлича концепцията обхват за атрибута на дадена аксиома;
- $extrAnnotations: A \cup P \cup C \cup Ann$ – извлича анотациите на даден елемент от структурата;
- $extrDisjointConcepts: C \rightarrow C$ – извлича несъвместимите концепции за дадена концепция.

Фигура 2. Дефиниции в ниво 2 на QGM

Ниво 2 от модела представя екстракторите, които определят информацията от учебния материал за генериране на въпросите. Тяхната цел е да извлекат от съществуващата структура подходящите знания, които да бъдат използвани при генерация на следващия въпрос. На Фигура 2 са показани дефинициите на използваните множества и операции в ниво 2. Тук дефинираме множеството на типовете изречения. В модела съществуват два типа изречения – съобщителни и въпросителни. Необходимо е типът на изреченията, които ще съставят въпрос да бъде ясен (генерацията на въпрос е обяснена в ниво 3). Някои екстрактори оперират в зависимост от типа на изречението.

Ниво 3 във формалния модел е генераторът. Той моделира спецификите за генерацията на въпросите. Генераторът успешно изпълнява генерирането на тестови въпроси в три стъпки:

- базов алгоритъм за генерация на въпросите (BQG – Basic Question Generation algorithm);
- генериране на отделните изречения, съставлящи въпроса (SG – Sentence Generation algorithm).
- генериране на възможни отговори за видовете въпроси, изискващи тази функционалност.

На Фигура 3 са представени дефинициите от ниво 3 на модела. Дефинирано е множеството на въпросите, където различаваме два основни типа – съобщителни и въпросителни. Класификация на тези типове се извършва въз основа на съставлящите ги изречения. Въпросителните въпроси са атомарни, съставени от едно въпросително изречение. Съобщителните въпроси могат да бъдат и съставни. Състоят се от едно или повече съобщителни изречения. За да дава моделът възможност за генериране на различни типове въпроси според начина на отговор, е дефинирано и множеството на възможните отговори. Това позволява определени типове въпроси да кореспондират с определен брой възможни отговори (правилни и неправилни), от които потребителят може да избира.

На Фигура 4 е описан алгоритъмът BQG, представящ една рамка за генериране на въпроси от структурата от знания. Резултатът от алгоритъма е тестови въпрос, състоящ се от генерираните изречения. Първа стъпка в алгоритъма е определянето на аксиомите за работа. За всяка от тях се прилагат екстракторите. Извлечените елементи се използват за генериране на изречение от всяка аксиома чрез SG функцията. Генерираните изречения формират въпроса.

Множества:

- Q – множество на въпросите. Освен това съществуват следните типове въпроси:
 - $Q_I \subseteq Q$ – множество на въпросителните въпроси;
 - $Q_D \subseteq Q^*$ – множество на съобщителните въпроси.
- Ans : множество на възможните отговорите.

Структури:

- $Pat \equiv \{(k_i : v_i) | k_i \in KW, v_i \in V\}$, където KW е множеството на ключовите думи, а V – множеството на променливи от допустими типове.

Фигура 3. Дефиниции в ниво 3 на QGM

function BQG (A, Pat, t, k) // връща множеството на генерираните изречения

Inputs:

- A ; // множество на аксиомите
- Pat ; // шаблон
- $t \in T$; // тип на изреченията
- $k \in K$; // критерий за избор на аксиомите

Output:

- $Q \leftarrow \emptyset$; // множество на генерираните изречения, формирани въпроса

$Axi \leftarrow selectAxi(A, k)$;

while $Axi \neq \emptyset$ **do**

for each $a \in Axi$ **do**

 // Извличане на нужните елементи

$bc \leftarrow extrBaseConcept(a)$;

$pro \leftarrow extrProperties(a)$;

$RC \leftarrow extrRestrConcepts(a)$;

$range \leftarrow extrRange(pro)$;

$q \leftarrow SG(bc, pro, RC, range, Pat, t)$;

$Q \leftarrow Q \cup \{q\}$;

return Q .

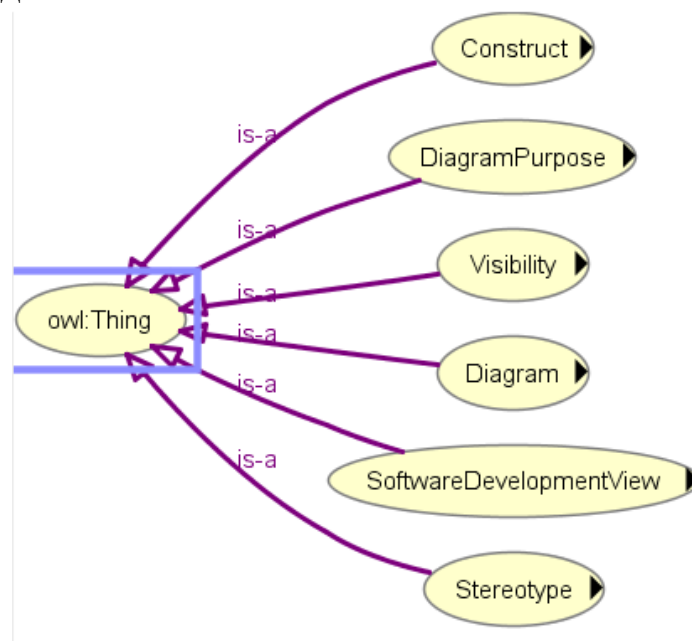
Фигура 4. Алгоритъмът BQG в QGM

Така построеният формален модел може да бъде приложен за реализацията на система, която автоматично генерира тестови въпроси от различен тип, според задаването на въпроса и изискванията за отговор.

СЕМАНТИЧНИ МОДЕЛИ ЗА ЕЛЕКТРОННИ ТЕСТОВЕ ВЪВ ВОП (Глава 4)

Реализация на онтология в областта на езика UML

За целите на дисертацията е създадена онтология в областта на софтуерните технологии и по-конкретно на езика UML (Unified Modelling Language) [16]. Онтологията е реализирана спрямо дефинираната по-горе концепция и върху формалния модел QGM, описан в Глава 3. Разработката на онтологията и резултатите от нея са публикувани в [23], [24], [25]. Тя съдържа основните елементи на езика. Чрез нея са описани видовете диаграми, които могат да се построяват в UML моделите. Важно е, че са включени знания, показващи за какво се използват те. Всяка диаграма е характеризирана и с конструкциите, които могат да се използват в нея, както и свойствата на тези конструкции. Всичко това представя направената онтология не просто като класификация на елементите, а като съвкупност от информация, която в покъсен етап може лесно да се обработва. Понятията от областта са описани чрез йерархия от класове, които са свързани помежду си чрез свойства. По този начин са образувани аксиоми, които представляват твърдения за езика UML. Така информацията се съхранява не само символно, но и семантично. Тя може да се използва не просто като текст за обработка и визуализация. Разполагаме с понятия, свързани смислово, което ни дава възможността да правим изводи за истинността на твърдения в областта.



Фигура 5. Диаграма на основните класове в онтологията на най-високо ниво йерархията

Всеки клас от онтологията отговаря на конкретно понятие – диаграма или конструкция в UML, както и понятия, описващи диаграмите. Йерархията е изградена по такъв начин, че да дава яснота за вида и същността на всяко от представените понятия. За да се организират подходящо знанията за UML се разделят на шест основни групи. Те са представени чрез шестте базови класа (Фигура 5), които са наследявани от всички останали. Всеки от основните класове има наследници, показващи подвидове на описваните понятия.

UML онтологията съдържа двата основни типа свойства, използвани за представяне на връзките на различните класове. Това са „object property“, които свързват класовете с други класове, и „data property“, които свързват класовете с дадени стойности.

Всеки клас от UML онтологията има своите ограничения (рестрикции). Те представляват връзката му с други класове или индивидуалност чрез свойствата. По този начин всеки клас се определя семантично.

При създаването на всеки клас, свойство или ограничение в онтологията чрез Protégé, образуваме нова аксиома. Тя представлява твърдение за областта, което съхраняваме и ще приемаме за вярно. В онтологията имаме OWL аксиоми, описващи класовете, индивидуалностите и свойствата. За целите на дисертацията, основно 4 вида аксиоми представляват интерес. Това са EquivalentClasses и SubClassesOf, DisjointClasses, ObjectPropertyRange.

В UML онтологията са добавени мета знания. Това са знания за знанията и дават допълнителна информация за отделните елементи в нея. Мета знанията в OWL са представени от анотациите. Анотациите могат да бъдат различни видове, което се определя чрез анотационно свойство – AnnotationProperty. В самия език съществуват няколко дефинирани анотационни свойства, както и възможността за анотационни свойства, дефинирани от потребителя. Всеки елемент от онтологията може да бъде анотиран и да бъде пояснен с мета знание. От вече дефинираните етикетът и коментарът са използвани за целите на UML онтологията. Етикетът (rdfs:label) предоставя информация за това как да бъде представен елементът от онтологията пред потребителя.

В UML онтологията са дефинирани още две анотационни свойства. Това са declarative и interrogative. Те се отнасят за свойствата и целта им е да разширят и допълнят техните етикети в зависимост от използването на свойството. Стойността на тези анотации дава правилните думи, нужни за използването му в съставяне на различни видове изречения. Може да се каже, че тези две анотации са видове ети-

кет, но са по-тясно специфицирани. В английския език при образуване на различни типове изречения (нужни за генериране на тестови въпроси) глаголите променят своята форма. За тази цел анотациите *declarative* и *interrogative* дават етикета на дадено свойство за използване в съответно на съобщително и въпросително изречение.

В UML онтологията е дефинирано анотационното свойство *test*. Неговата стойност обозначава темата в UML, към която принадлежи елемента от онтологията. Така имаме мета знания, които правят търсенето на информация по-ефективно. Чрез него лесно и удобно ще бъдат откривани знанията по теми. Използвайки анотацията, системата ще може да съставя тестове по зададени теми. Първоначално такава анотация беше добавена за всички аксиоми. В последствие, при промяна на алгоритъма за генериране на въпроси, тази анотация беше добавена за всички класове. Тази промяна отразява фактът, че всички знания за едно понятие принадлежат към една тема. Възможността за повече от една анотации, позволява знанията да бъдат поставяни в повече от една теми.

Приобщаване на съществуващи онтологии за целите на електронното тестване

Съществуващи онтологии в подходящи области (интересни или необходими за целите на електронното тестване) могат да бъдат обновявани, така че да бъде възможно използването им за генериране на тестване. Първоначално е необходим да се установи дали знанията в съответната онтология са достатъчни за желаното тестване. Ако това е така, в онтологията трябва да се добавят единствено подходящите мета знания. Това са анотациите описани в предходната секция. За целите на дисертацията беше направен експеримент за приобщаване на вече съществуваща онтология за генериране на електронни тестове. Използвана е онтология в областта на софтуерни технологии, съдържаща знания за базови понятия в областта. Оригиналната версия на тази онтология е разработена с цел генериране на електронно учебно съдържание по „Софтуерни технологии“. Процесът по разработката и резултатите са публикувани в [17]. За целите на електронното тестване онтологията е обогатена с новите анотации.

ИНТЕЛИГЕНТНА СРЕДА ЗА ЕЛЕКТРОННИ ТЕСТОВЕ

За целите на дисертационния труд е създадена интелигентна среда, която съставя за потребителя тест в областта на езика UML. Това включва автоматично генериране на тестови въпроси и оценяване на потребителските отговори. От потребителска гледна точка системата представлява тестова система, задаваща определен брой въпроси последователно като за всеки въпрос се предоставя определено време за отговор. Отговорът на всеки въпрос се проверява веднага след получаването му в системата. Непосредствено след завършване на теста резултатите се предоставят на потребителя.

Предоставена е възможност за генерация на различни типове въпроси. Типовете въпроси са избрани според стандарта QTI и включва шест вида въпроси според начина на задаване на въпроса и според възможностите за отговор. Те са: „Отворен въпрос“, „Липсващи думи“, „Вярно/Невярно“, „Затворен въпрос с един верен отговор“, „Затворен въпрос с няколко верни отговори“, „Текстови въпрос“.

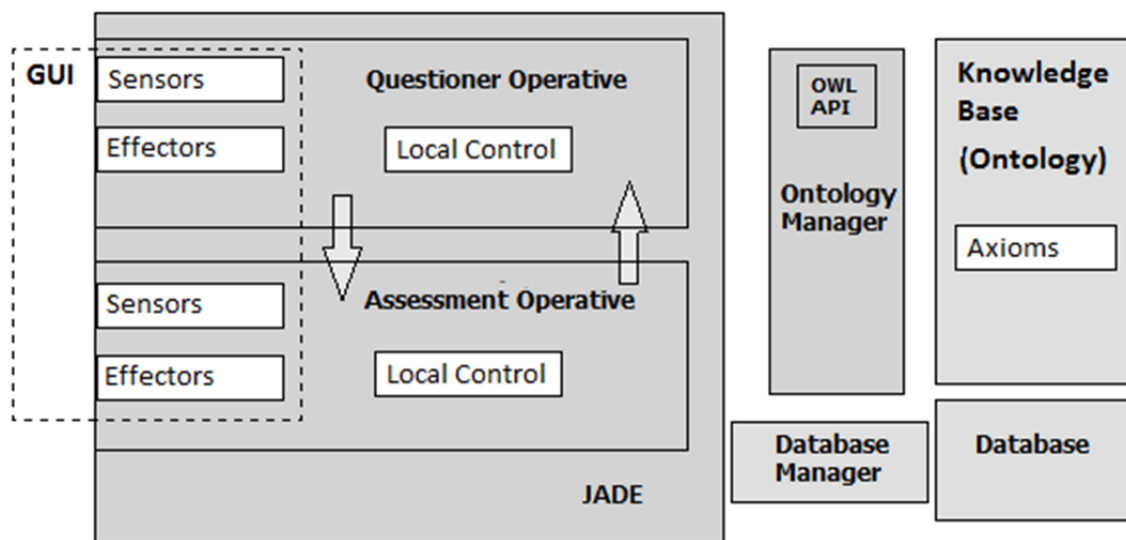
Тестовите предоставяни от средата са разделени на теми в областта според различните пакети в езика UML и последователността им в учебната програма. Така системата разполага с 11 последователни теста, които са съставени от тематични въпроси. Темите и техния ред са следните: “Use case“ пакет, “Interaction“ пакет, “Class” пакет, “State machine” пакет, “Activity” пакет, “Composite Structure” пакет, “Component” пакет, “Deployment” пакет, Връзки между UML конструкции, Диаграми, Обобщение. Последният тест „Обобщение“ включва 20 въпроса от цялата област като тези въпроси са персонално подбрани от средата. Въпросите от този тест се генерират на база на това как се е представил потребителят на изминалите тематични тестове.

Имплементацията и резултатите от средата за електронно тестване са публикувани в [23], [24].

Архитектура

Средата за електронни тестове е изградена като мултиагентна система. Основните компоненти в нейната архитектура (Фигура 6) са два оперативни асистента, база знания и база данни. Двата оперативни асистента в средата са Question Operative (QO) и Assessment Operative (AO). Всеки от тях има своите специфични задачи, но архитектурата им е идентична. Те са реализирани като JADE (Java Agent DEvelopment Framework) [18], [19] агенти, които обработват структурирано образователно съдържание. Двата агента споделят база данни, база знания и графичен интерфейс и имат своите сензори и ефектори.

Те биват използвани от оперативните асистенти, за да осъществяват взаимодействие със своята околна среда, която е графичният интерфейс на средата (Graphical User Interface – GUI). Чрез сензорите и ефекторите агентите достъпват GUI. Сензорите улавят промените в околната среда, необходими за работата на агентите. В случая това са всички външни промени настъпващи в графичния интерфейс, породени от действията на потребителя. Ефекторите имат за цел да влияят на околната среда и да нанасят промени в нея. Това включва всичко необходимо за визуализиране пред потребителя и представяне на информация, даване на възможност за избор на действия.



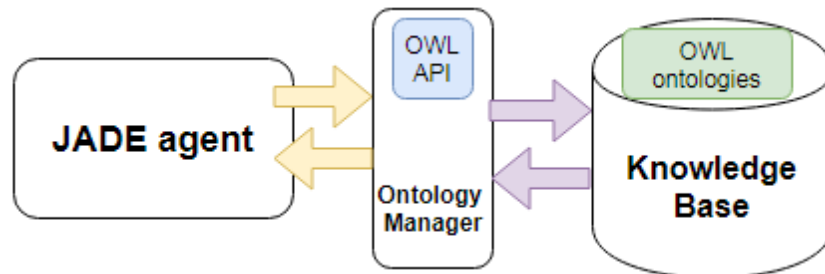
Фигура 6. Архитектура на средата за електронни тестове

Въпреки своите специфични функционалности, за изпълнение и координиране на задачите си оперативните асистенти имат нужда да комуникират помежду си по време на теста. По този начин те могат да изискат един от друг някакво действие или да извършат такова при поискване. Тази комуникация е реализирана според FIPA спецификацията (The Foundation for Intelligent Physical Agents) [20]. QO и AO изпращат асинхронни съобщения помежду си, за да комуникират, когато е нужно.

Базата знания е важна част от архитектурата на средата. Тя съхранява структурирано образователно съдържание. Нейната идея е да съдържа OWL онтологии в различни области и дисциплини, които да представят това съдържание и да бъдат използвани от оперативите. Към момента базата знания съдържа вече описаната в глава 4 онтология в областта на UML, представяща правилата изграждащи езика.

За да се осъществи връзката между оперативните асистенти и онтологията в базата знания, е имплементиран онтологичен мениджър

(Ontology Manager Entity - OME). Неговата идея е да работи като интерфейс между агентите и онтологията (Фигура 7). OME е реализиран като Java клас, който използва OWL API [22] като библиотека за достъпване на OWL онтологиите. Подходът за осъществяване на връзката между оперативните асистенти и онтологията е публикуван в [21].



Фигура 7. Връзка между оперативните асистенти и базата знания

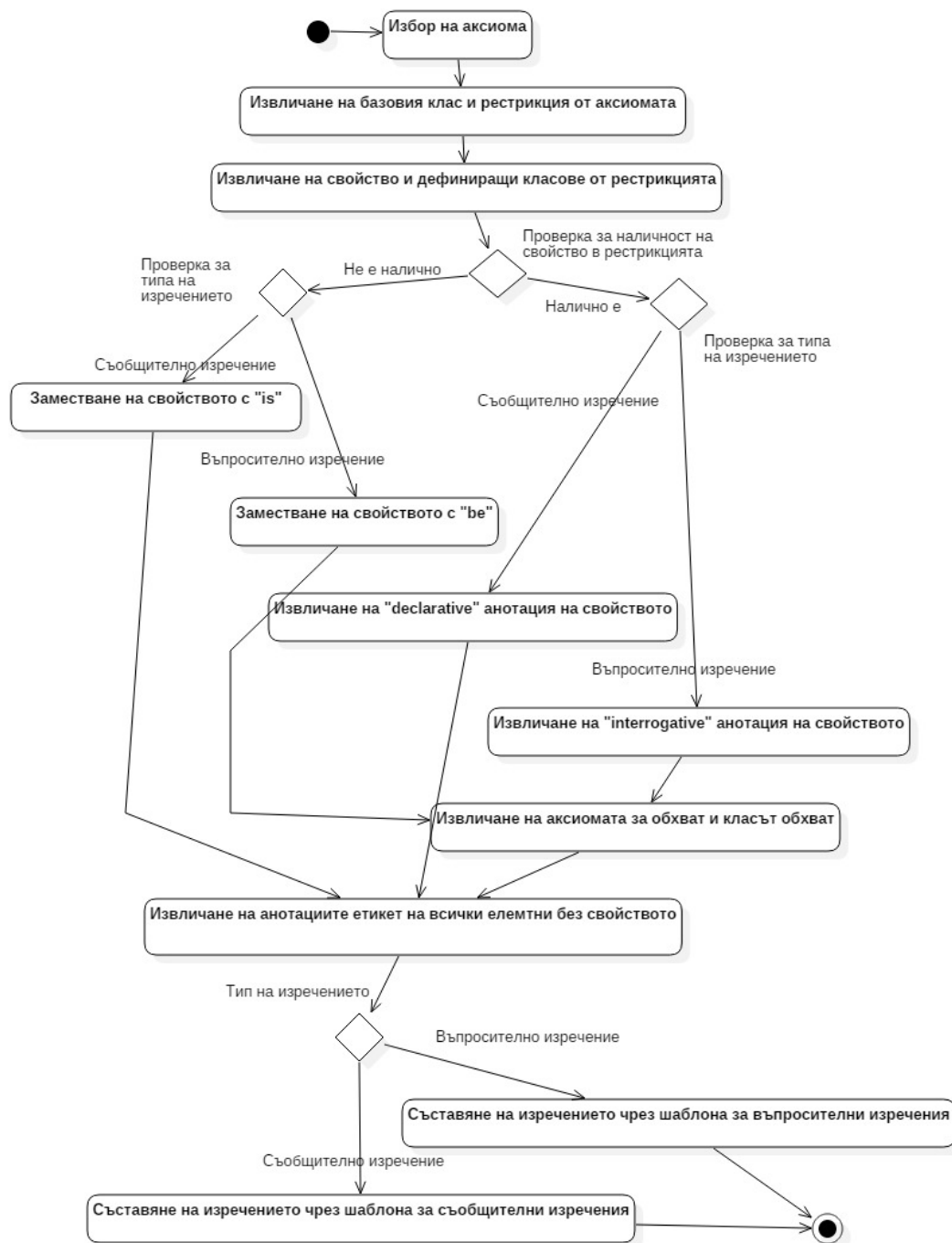
Базата данни в средата е реляционна база данни, която съхранява данни за темите на тестовете и техните резултати. Тези данни се използват от оперативните асистенти за определяне на темата на теста.

Автоматично генериране на въпроси

Автоматичното генериране на въпроси в средата за електронно тестване е реализирано като е приложен формалния модел за генериране на тестови въпроси, описан в Глава 3. Генерирането на въпроси е функционалност на единия агент в средата – QO. За съставяне на въпросите се използват знания от UML онтологията.

Първата стъпка е извличане на множество аксиоми от онтологията по даден критерий. Това множество се състои от аксиоми от видовете Equivalent Classes и Sub Class, за които рестрикциите трябва да бъдат екзистенциални и/или универсални в случай, че съдържат такива. Критерият зависи от типа на въпроса, така че да отговоря на изискванията за този тип.

Всеки въпрос, независимо от типа му, се състои от една или повече съставни части. Всяка част представлява изречение. Изреченията могат да бъдат въпросителни или съобщителни, а това зависи от вида на въпроса. Така генерацията на всеки въпрос включва съставяне на определен брой изречения. Алгоритъмът за генериране на изречение е показана на Фигура 8 чрез активити диаграма. Всяко изречение се генерира от точно една аксиома.

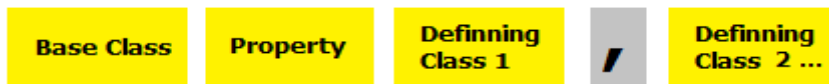


Фигура 8. Активити диаграма на алгоритъма за генериране на изречение

За това генерирането на изречение от всеки от двата вида започва с избор на аксиома от вече наличното множество. След това се извличат различните елементи от аксиомата. За да бъде съставено изречението се използва шаблон, според който те намират мястото си в него. Той е различен според типа изречение. За въпросително изречение шаблонът е показан на Фигура 9, а за съобщителни изречения – Фигура 10.

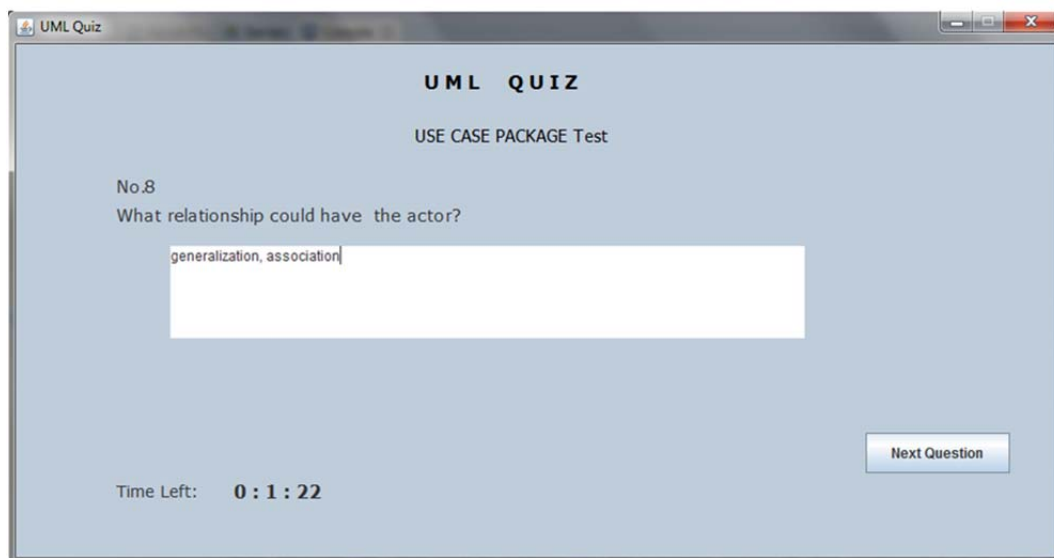


Фигура 9. Шаблон за въпросително изречение



Фигура 10. Шаблон за съобщително изречение

Алгоритъмът за изречения се използва за генериране на съставните части на различни типове въпроси. „Отворен въпрос“ е съставен от едно въпросително изречение със свободен отговор. Изглед от системата при задаване на такъв въпрос е показан на Фигура 11.



Фигура 11. Изглед от средата за въпрос от вида „Отворен въпрос“

„Липсващи думи“ се състои от едно съобщително изречение с липсващи понятия. „Вярно/Невярно“ е съставен от едно съобщително изречение, генерирано като грешно или правилно като част от понятията могат да се заместят с такива, които не удовлетворяват, използваната аксиома. „Затворен въпрос с един верен отговор“ се състои от едно въпросително изречение и множеството на възможните отговори, от които един е верен, а два са грешни. Верният отговор представлява понятие, представено от един от дефиниращите класове в рестрикцията на аксиомата. Двама неверни отговора се извличат от disjoint аксиоми – несъвместими класове. „Затворен въпрос с няколко верни отговори“ представлява въпрос с 4 възможни отговора, от които два са верни. Особеното тук е, че е нужно да се постави ограничение при избора на аксиома за генерацията. Аксиомата трябва да съдържа рестрикция, къ-

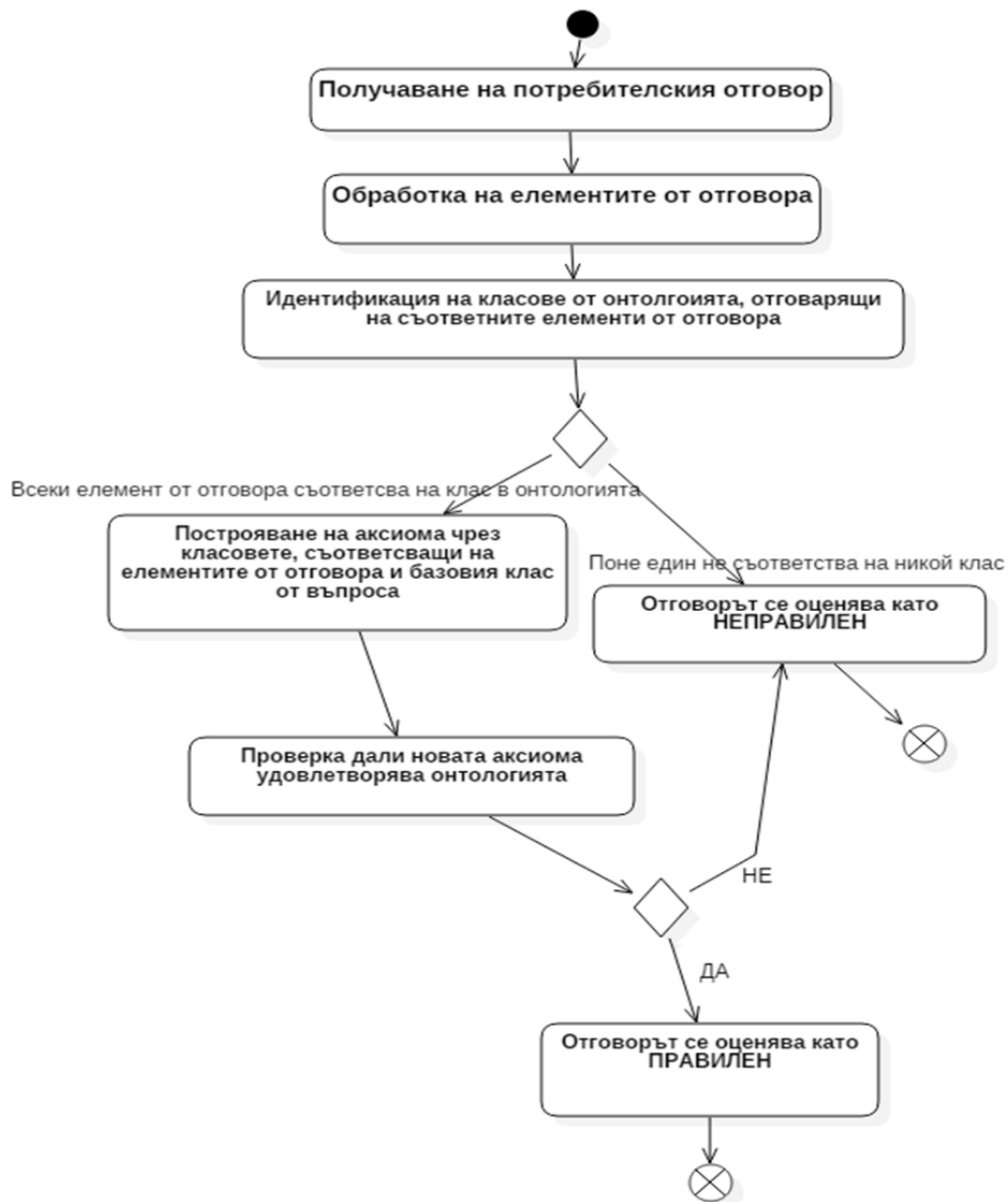
дето дефиниращите класове са поне два, т.е. съдържа множество класове от типа сечение или обединение. Това дава гаранция, че изискването за два верни възможни отговори, може да бъде осигурено. „Текстови въпрос“ е въпрос, представляващ кратък текст, в който част от понятията са заместени с няколко опционални отговора, където един от тях е подходящ за това място в текста. Този тип въпрос се състои от няколко съобщителни изречения. За да бъде генериран текст, те трябва да бъдат смислово свързани и да представят една тематика. По тази причина множеството от аксиоми, нужно за генерация, се избира, така че всяка аксиома да описва определено понятие.

Съставяне на тестове

Съставянето на различните тестове също се осъществява от QO. За тази цел се използва информация от база данни. Преди стартирането на всеки тест се определя какъв ще бъде той, каква е темата му. Тази информация се взема от базата данни персонално за всеки потребител. Темите за тестовете имат определена последователност. Така, имайки последния решен тест, се определя следващият. Темата на теста определя основния критерий при извличане на множеството аксиоми от онтологията, които могат да участват в генерацията на въпросите от този тест. Финалният тест използва различен подход от останалите. Тук отново се използва информация от базата данни. Тя представлява списък от понятията, които текущия потребител е сгрешил във въпроси от предишни тестове, т.е. отразява ненаучения материал. В този тест множеството от аксиоми се състои от всички аксиоми с базов клас, представящ едно от тези понятия. Така финалният тест е персонализиран за всеки потребител.

Проверка на потребителските отговори

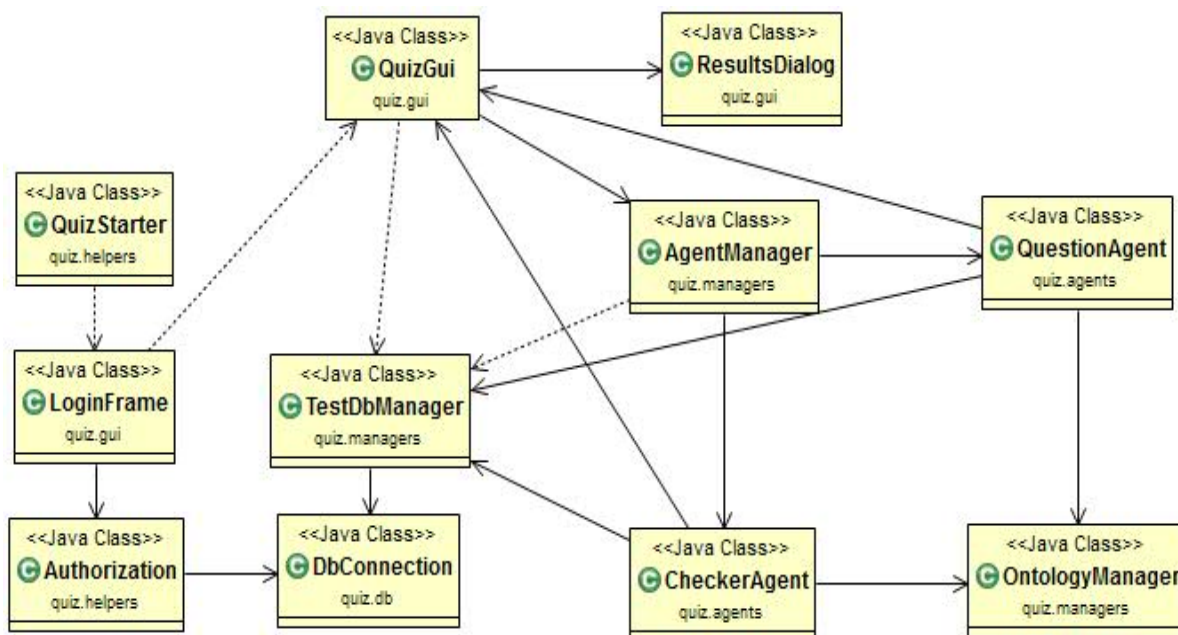
В тестовата среда е развит и алгоритъм за проверка на отговорите, дадени от потребителя по време на теста. АО изпълнява този алгоритъм. Проверката е максимално автоматизирана и възможно най-независима от генерацията на въпросите. Правилните отговори не са определени предварително и не се оценяват като просто статично сравнение за съвпадение. Този процес отново се извършва в съответствие с онтологията. Това се осъществява като верността на всеки отговор се проверява в онтологията. На Фигура 12 е изобразена активити диаграма на базовия алгоритъм за проверка.



Фигура 12. Активити диаграма на алгоритъма за проверка на отговор

Реализация на средата за електронни тестове

Класовете, реализиращи мултиагентната система, са организирани в 6 Java пакета. На Фигура 13 е представена клас диаграма на основните класове. Тяхната принадлежност към пакетите също е представена. Таблица 1 дава информация за съответствията между архитектурните компоненти и реализираните класове.



Фигура 13. Клас диаграма на основните класове, реализиращи средата

Таблица 1. Съответствия между архитектурните компоненти и класовете

Компонент	Клас	Функционалност
Database Manager	DbConnection, TestDbManager	управление на базата данни
Ontology Manager	OntologyManager	базови операции върху онтологията
GUI	QuizGui, LoginFrame, ResultsDialog	Графичен потребителски интерфейс
QO	QuestionAgent	агентът, който съставя тестовете и генерира въпросите
Сензори	AgentManager	агентът, получава информацията, че потребителят преминава към следващия въпрос
Ефектори	QuestioningService, ReceiveCorrectAnswersService	агентът визуализира необходимите данни за теста и въпроса чрез ГПИ.
АО	CheckerAgent	агентът, който проверява отговорите на потребителя
Сензори	ReceiveAnswerService	получава отговорите на потребителя от ГПИ
Ефектори	ReceiveAnswerService	подготвя ГПИ за следващ въпрос

Апробация на средата за електронни тестове

За проверка на ефективността на средата за електронни тестове са проведени два експеримента: тестване на средата при работа с друга онтология, приобщена за целта и апробация на средата за електронни тестове.

Първият експеримент е проведен с цел да се провери гъвкавостта на средата и до колко предложените и реализирани алгоритми могат да бъдат универсални и независими от дисциплината и тематиката на тестовете. Използвана е онтология в областта на софтуерните технологии, съдържаща знания за базови понятия в областта. Тя е обогатена с необходимите за целта анотации, за да съдържа мета знанията, нужни за реализираните алгоритми. Този процес е описан в точка 4.3 от дисертацията. Онтологията е поставена в базата знания и е използвана за функционалностите на средата – генериране и проверка на въпроси, съставяне на тестове. При тестването на средата със знания от новата онтология не са забелязани проблеми, произхождащи от използването на друга онтология. Средата изпълнява своите функционалности според очакванията и не се наблюдават разлики при изпълнението и резултатите, дължащи се на промяната в базата знания.

Вторият експеримент е апробация на средата за електронни тестове. Проведен е тест на системата с избрани студенти по време на практикума по Софтуерни технологии за езика UML, включен в бакалавърската програма „Информатика“ на ФМИ на Пловдивския университет. Системата е приета добре от студентите. Събрана е информация за възникнали проблеми при използването ѝ. Като основна критика се откроява това, че системата и генерираните въпроси са на английски език. За въпроси на български език се изисква превод на знанията и най-вече мета знанията в онтологията. Това е трудоемка задача и изисква да бъдат преведени голям брой технически термини от областта. Освен това е нужна и работа по алгоритъмът за генериране на въпроси, поради граматичните разлики в двата езика. Установено е, че тази задача е подходяща и необходима за бъдещото развитие на средата.

Като обобщение може да се заключи, че средата за електронно тестване е независима от областите на онтологиите в базата знания. Изпълнява функционалностите и постига добри резултати при използване в реална среда.

ЗАКЛЮЧЕНИЕ (Глава 6)

В дисертацията е представено изследване в областта на технологиите за електронно обучение. По-конкретно е обърнато внимание на начините за създаване на електронни тестове и възможностите за използване на данни със семантична стойност за автоматизиране на процеса. Като основни приноси на дисертацията се определят следните:

1. Предложение за формализирани правила за автоматично генериране на тестови въпроси
2. Създаване на прототипна онтология с цел използването и в електронното тестване и дефиниране на правила за създаване на семантични модели за генериране на въпроси
3. Разработка на прототипна среда за автоматично електронно тестване

В Таблица 2 е направена връзката между приносите и поставените цели на дисертацията, посочени са съответстващите секции в дисертацията и направените публикации.

Таблица 2. Връзка на приносите с целите на дисертацията и публикациите

Принос	Цел	Секция в дисертацията	Публикации от приложение 1
1	1	Глава 3	1, 2
2	2	Глава 4	2, 3, 5, 6
3	3	Глава 5	2, 3, 4

В дисертацията е дефиниран формален модел, който описва правила и алгоритми за автоматично генериране на тестови въпроси. На базата на този модел е реализиран прототип на среда за провеждане на персонализирани тестове с автоматично генерирани и проверявани въпроси. Средата е разработена като мултиагентна система, използваща база знания от онтологии. Дефинирана е и концепция за изграждане на онтологии за целите на автоматично създаване на тестове. Тя е реализирана чрез изграждането на онтология в областта на езика UML и е използвана за базата знания на прототипната среда. Предложен е и вариант за обновяване на съществуващи онтологии, така че да могат бъдат включени в базата знания на средата.

Отделни части от работата по дисертацията и междинни резултати са представени в редица публикации в специализирани списания,

международни и национални конференции, в катедрени семинари и в отчетите на научно-изследователски проекти, в които авторът участва.

При работата си върху дисертацията авторът потвърждава впечатлението си, че тематиката за електронното обучение става все всеобхватна. Във времена, в които физическото присъствие в учебните заведения не е наложително, а понякога и невъзможно, е важно тази област да се развива и автоматизира, така че да бъдат създавани полезни и ефективни системи, предоставящи всички необходими услуги. Това предполага необходимост от различни възможности за електронно тестване, както и усъвършенстването му до степента на пълно автоматизиране.

ПУБЛИКАЦИИ ПО ДИСЕРТАЦИОННИЯ ТРУД

1. Nina Stancheva, Asya Stoyanova-Doycheva, Stanimir Stoyanov, Ivan Popchev, Vanya Ivanova. A Model for Generation of Test Questions, *Compt. Rend. Acad. Bulg. Sci.*, tome 70 No 5, 2017, pp. 619 – 630, ISSN 1310–1331, ISSN 2367-5535 (Online), **IF 2015=0.233, Q4**
2. Nina Stancheva, Asya Stoyanova-Doycheva, Stanimir Stoyanov, Ivan Popchev, Vanya Ivanova. “An Environment for Automatic Test Generation”. *Cybernetics and Information Technologies*, Volume 17, ISSN: 1311-9702, ISSN (Online) 1314-4081, DOI: <https://doi.org/10.1515/cait-2017-0025> – **Списание то е с импакт ранг, SCImago Journal Rank (SJR) 2016: 0.203. (Индексирана в SCOPUS)**
3. Nina Stancheva, Asya Stoyanova-Doycheva, Stanimir Stoyanov, Ivan Popchev. Automatic generation of test questions by software agents using ontologies. In: *Proc. of IEEE 8th International Conference on Intelligent Systems*, Sofia, Bulgaria, 4-6 September 2016, pp. 741-746, DOI: 10.1109/IS.2016.7737395, Electronic ISBN: 978-1-5090-1354-8, USB ISBN: 978-1-5090-1353-1
4. Vladimir Valkanov, Jordan Todorov, Borislav Daskalov, Nina Stancheva, Stanimir Stoyanov. Personal assistant with ontology based knowledge. (2017). *Conference: International Conference Big Data, Knowledge and Control Systems Engineering*. Sofia, Bulgaria, ISSN: 2367 – 6450
5. Asya Stoyanova-Doycheva, Mariya Miteva, Nina Stancheva. Разработване на интелигентна среда за създаване на електронно учебно съдържание по „Софтуерни технологии“ (2015). Сборник с доклади на научна конференция „Иновационни ИКТ: Изследвания, разработка и приложения в бизнеса и обучението“, гр. Хисар, 11 – 12 ноември 2015 г., ISBN: 978-954-8852-56-7
6. Asya Stoyanova-Doycheva, Nina Stancheva, Vanya Ivanova, Stanimir Stoyanov, “Structure of an Ontology Used in a Test Generation Environment”, *American Institute of Physics Conference Proceedings*, 2020 (to print)

БЛАГОДАРНОСТИ

Искам да изразя своята благодарност към хората, без които този дисертационен труд би бил невъзможен.

На първо място бих искала да изкажа огромната си благодарност към научния си ръководител, доц. д-р Ася Стоянова-Дойчева, за идеите и ценните съвети, за подкрепата и всеотдайността, които получавах през цялото време. Благодаря за вдъхнатия кураж и мотивацията, от които имах нужда!

Изказвам специална благодарност и на проф. Станимир Стоянов за безрезервната помощ, насоките, градивните критики и мотивиращите отзиви.

Благодаря и на своето семейство за това, че винаги ме подкрепят и ми дават нужната позитивна енергия за всички начинания.

Благодаря!

БИБЛИОГРАФИЯ

- [1] IMS Question & Test Interoperability Specification, <https://www.imsglobal.org/question/index.html>
- [2] HAMMER, M., AND MCLEOD, D. 1981. Database description with SDM: A semantic database model. *ACM Trans. Database Syst.* 6,3 (Sept.), 351 – 386.
- [3] R. Hull and R. King. Semantic database modeling: survey, applications, and research issues. *ACM Computing Surveys*19(3), 201 – 260 (1987).
- [4] Chen, Peter (March 1976). “The Entity-Relationship Model - Toward a Unified View of Data”. *ACM Transactions on Database Systems.* 1 (1): 9 – 36
- [5] KERSCHBERG, L., AND PACHECO, J. E. S. 1976. A functional data base model. Tech. Rep., Pontificia Univ. Catolica do Rio de Janeiro, Rio de Janeiro, Brazil.
- [6] Gruber, T. (1993). “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”. *International Journal of Human-Computer Studies.* 43 (5 – 6): 907 – 928. doi:10.1006/ijhc.1995.1081
- [7] OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition) <https://www.w3.org/TR/owl2-syntax/>
- [8] Karadimce, Aleksandar (2013). Quality Estimation of E-learning Semantic Web Ontology. *ICT Innovations 2013 Web Proceedings* ISSN 1857-7288
- [9] Justin, Sheeba (2018). Semantic Web to E-Learning Content. *International Journal of Advanced Research in Computer Science and Software Engineering.*
- [10] Ibrahim Teo, Noor Hasimah & Joy, Mike (2017). Validation of Course Ontology Elements for Automatic Question Generation. *E-Learning, E-Education, and Online Training. Third International Conference, eLEOT 2016, Dublin, Ireland.* 10.1007/978-3-319-49625-2_17.
- [11] S. Stoyanov, A. Stoyanova-Doycheva, T. Glushkova, E. Doychev, Virtual Physical Space – An Architecture Supporting Internet of Things Applications, 20th International Symposium on Electrical Apparatus and Technologies (SIELA), 3 – 6 June 2018, DOI: 10.1109/SIELA.2018.8447156, <https://ieeexplore.ieee.org/abstract/document/84471>

- [12] Todorka Glushkova, Maria Miteva, Asya Stoyanova-Doycheva, Vanya Ivanova, Stanimir Stoyanov, Implementation of a Personal Internet of Thing Tourist Guide, American Journal of Computation, Communication and Control, Publication Date: May 10, 2018, Pages: 39 – 51, ISSN 2375-3943, <http://www.aascit.org/journal/archive?journalId=901&issueId=9010502>
- [13] Stanimir Stoyanov, Todorka Glushkova, Asya Stoyanova-Doycheva and Vanya Ivanova, A Reference Architecture Supporting Smart City Applications, 22nd International Conference on Business Information Systems, University of Seville, 26 – 28 June 2019, Seville, Spain (to print) Реферирана в SCOPUS, SJR = 0,243
- [14] S. Stoyanov, T. Glushkova, E. Doychev, Cyber-Physical-Social Systems and Applications, Part 1: Reference architecture, LAMBERT Academic Publishing, 2019, ISBN: 978-620-0-31825-1
- [15] Nina Stancheva, Asya Stoyanova-Doycheva, Stanimir Stoyanov, Ivan Popchev, Vanya Ivanova. A Model for Generation of Test Questions, Compt. Rend. Acad. Bulg. Sci., tome 70 No 5, 2017, pp. 619 – 630, ISSN 1310–1331, IF 2015=0.233
- [16] Grady Booch, James Rumbaugh, Ivar Jacobson. Unified Modeling Language User Guide, The (2 ed.). Addison-Wesley. 2005. p. 496. ISBN 0321267974
- [17] Asya Stoyanova-Doycheva, Mariya Miteva, Nina Stancheva. Разработване на интелигентна среда за създаване на електронно учебно съдържание по „Софтуерни технологии“. (2015). Conference: „Иновационни ИКТ: Изследвания, разработки и приложения в бизнеса и обучението“
- [18] Developing multi-agent systems with JADE by Fabio Bellifemine, Telecom Italia, Italy; Giovanni Caire, Telecom Italia, Italy; Dominic Greenwood, Whitestein Technologies AG, Switzerland; 2007
- [19] Java Agent Development Framework. <http://jade.tilab.com/>
- [20] FIPA ACL Message Structure Specification. FIPA TC Communication. 2002/12/03.
- [21] Vladimir Valkanov, Jordan Todorov, Borislav Daskalov, Nina Stancheva, Stanimir Stoyanov. Personal assistant with ontology based knowledge (2017). Conference: International Conference Big Data, Knowledge and Control Systems Engineering. Sofia, Bulgaria
- [22] OWL API. <http://owlapi.sourceforge.net/>
- [23] Nina Stancheva, Asya Stoyanova-Doycheva, Stanimir Stoyanov, Ivan Popchev, Vanya Ivanova. “An Environment for Automatic Test Generation”. Cybernetics and Information Technologies, Volume 17,

ISSN (Online) 1314-4081, DOI: <https://doi.org/10.1515/cait-2017-0025> – Списанието е с импакт ранг, SCImago Journal Rank (SJR) 2016: 0.203. (Индексирана в SCOPUS)

- [24] Nina Stancheva, Asya Stoyanova-Doycheva, Stanimir Stoyanov, Ivan Popchev. Automatic generation of test questions by software agents using ontologies. In: Proc. of IEEE 8th International Conference on Intelligent Systems, Sofia, Bulgaria, 4-6 September 2016, pp. 741-746
- [25] Asya Stoyanova-Doycheva, Nina Stancheva, Vanya Ivanova, Stanimir Stoyanov, " Structure of an Ontology Used in a Test Generation Environment", American Institute of Physics Conference Proceedings, 2020 (to print)

