

ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА „КОМПЮТЪРНА ИНФОРМАТИКА“

ДИМИТЪР БЛАГОЕВ БЛАГОЕВ

**СИСТЕМИ ЗА СЕМАНТИЧЕН АНАЛИЗ
И ИНФОРМАЦИОННО ТЪРСЕНЕ**

АВТОРЕФЕРАТ

на дисертационен труд

за присъждане на образователната и научна степен „Доктор“
в област на висше образование 4. Природни науки, математика и информатика;
професионално направление 4.6 Информатика и компютърни науки;
докторска програма Информатика

Научен ръководител:

проф. дмн. Георги Тотков

гр. Пловдив

2012 г.

ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА „КОМПЮТЪРНА ИНФОРМАТИКА“

ДИМИТЪР БЛАГОЕВ БЛАГОЕВ

**СИСТЕМИ ЗА СЕМАНТИЧЕН АНАЛИЗ
И ИНФОРМАЦИОННО ТЪРСЕНЕ**

АВТОРЕФЕРАТ

на дисертационен труд

за присъждане на образователната и научна степен „Доктор“
в област на висше образование 4. Природни науки, математика и информатика;
професионално направление 4.6 Информатика и компютърни науки;
докторска програма Информатика

Научен ръководител:

проф. дмн. Георги Тотков

Рецензенти:

доц. д-р Красимир Марков
доц. д-р Елена Сомова

гр. Пловдив

2012 г.

Дисертационният труд е обсъден и насочен за защита на заседание на катедра „Компютърна информатика“ при Факултета по математика и информатика на ПУ „Паисий Хилендарски“.

Дисертационният труд „Системи за семантичен анализ и информационно търсене“ съдържа 134 страници. Библиографията включва 127 източника. Списъкът на авторските публикации се състои от 9 заглавия.

Защитата на дисертационния труд ще се състои на 29.05.2012 г. от 13:00 часа в Заседателната зала на Нова сграда на ПУ „Паисий Хилендарски“.

Материалите по защитата са на разположение за интересувашите се в секретариата на ФМИ, нова сграда на ПУ, каб. 330, всеки работен ден от 8:30 до 17:00 часа.

СЪДЪРЖАНИЕ

УВОД.....	1
ГЛАВА 1. ОБЗОР НА СЪВРЕМЕННИ МЕТОДИ И СИСТЕМИ.....	3
1.1 ВЪВЕДЕНИЕ В ИНФОРМАЦИОННОТО ТЪРСЕНЕ.....	4
1.2 ОРГАНИЗАЦИЯ И МЕТОДИ.....	4
1.3 ХАРАКТЕРИСТИКИ НА ТЪРСЕЩИТЕ СИСТЕМИ.....	5
1.4 ОЦЕНКА НА КАЧЕСТВОТО.....	6
1.5 ИЗВЛИЧАНЕ НА ИНФОРМАЦИЯ.....	7
1.6 ТЪРСЕНЕ ПО ДОКУМЕНТ-ОБРАЗЕЦ.....	8
ГЛАВА 2. МЕТАМОДЕЛ ЗА ОПИСАНИЕ НА ПРОЦЕСИ ILNET.....	9
2.1 ПОТРЕБИТЕЛСКА ПЕРСПЕКТИВА.....	9
2.2 СЪСТАВНИ ЕЛЕМЕНТИ НА МОДЕЛА.....	11
2.3 СЪВМЕСТИМОСТ НА ILNET МОДЕЛА.....	12
2.4 УПОТРЕБА И ПРИЛОЖЕНИЕ НА МОДЕЛА.....	14
2.5 ИЗВОДИ.....	15
ГЛАВА 3. АРХИТЕКТУРА И РЕАЛИЗАЦИЯ НА СУРП.....	16
3.1 АНАЛИЗ НА ИЗИСКВАНИЯТА КЪМ РЕАЛИЗАЦИЯТА НА МОДЕЛА.....	16
3.2 ПРОЕКТИРАНЕ НА ПРОТОТИП НА СИСТЕМА ЗА УПРАВЛЕНИЕ НА РАБОТНИ ПРОЦЕСИ.....	19
3.3 ОПИСАНИЕ НА РЕАЛИЗАЦИЯТА.....	19
ГЛАВА 4. ПРИЛОЖЕНИЯ НА РАЗРАБОТЕНИЯ МОДЕЛ.....	26
4.1 ИЗВЛИЧАНЕ НА КОНТАКТНА ИНФОРМАЦИЯ ОТ ДОКУМЕНТИ НА БЪЛГАРСКИ ЕЗИК.....	26
4.2 АВТОМАТИЗИРАНО ГЕНЕРИРАНЕ НА МЕТАДАННИ ЗА УЧЕБНИ ОБЕКТИ.....	28
4.3 МОДЕЛИРАНЕ И УПРАВЛЕНИЕ НА ВИРТУАЛНИ АДАПТИВНИ КУРСОВЕ, БАЗИРАНИ НА MOODLE.....	30
ЗАКЛЮЧЕНИЕ.....	33
БЛАГОДАРНОСТИ.....	35
НАУЧНИ ТРУДОВЕ НА АВТОРА, СВЪРЗАНИ С ДИСЕРТАЦИЯТА.....	35
БИБЛИОГРАФИЯ.....	36

УВОД

Актуалност на темата

Търсенето на информация е относително утвърдена технология, която се използва за получаване на множество от свързани материали от огромна колекция. Типичен пример за търсене на информация е търсенето по ключова дума в Интернет, при което резултатът е страници, съдържащи тази ключова дума. Броят на намерените страници, обаче, може да бъде много голям, а и по всяка вероятност, не всичко на дадена страница ще е важно за потребителя. Оттук – преглеждането на всяка страница – резултат на търсенето, с цел откриване на необходимата информация, много често е трудоемка задача, изискваща значителен разход на време. Ето защо се налага процесът на търсене да се допълва с *извличане на информация*, при което в намерените документи се търсят определени фрази (представящи данни, факти и др.), които биха представлявали интерес за потребителя. С други думи, документи с неструктурирано съдържание се трансформират в структури от определен тип.

Създадени са много различни системи за информационно търсене и функции за търсене, вградени в различни приложения. Във всички тези случаи (независимо от математическия модел), идеите, софтуерните програми и структурите от данни са достатъчно прости. Съществува мнение, че всяко ново поколение програми е по-съвършено от предходното. Друг краен възглед се състои в това, че „всичко ново – това е добре забравено старо“. По всяка вероятност, истината, в случая на софтуерните системи за информационно търсене, е по средата.

Цели и задачи на дисертационния труд

Основна цел на дисертационното изследване е – да се моделира и създаде система за бързо проектиране на процеси в областта на информационното търсене и семантичният анализ.

Изведени са **четири основни** цели на дисертационното изследване:

1. Изследване на съществуващи методи и системи за информационно търсене и семантичен анализ;
2. Предлагане на общ модел за изграждане на системи за моделиране и процеси решаващи разглежданите задачи;
3. Изграждане на конкретна система, базирана на предложеният модел;
4. Създаване и експериментиране с конкретни описания на процеси, използвайки с реализираната система.

За постигане на поставените цели в изследването последователно се решават следните задачи:

1. Да се проучи общата теория за информационно търсене и семантичен анализ, да се класифицират съществуващите системи и методи;
2. Да се създаде концептуален модел за описание на процеси за информационно търсене и семантичен анализ;
3. Да се проектира система за описание и управление на работни процеси и да се разработи прототип, базиран на съвременни технологии;
 - 3.1. Да се проектира система за управление на работни процеси, базирана на описаният модел от точка 2;
 - 3.2. Да се разработи прототип на СУРП (система за управление на работни процеси);
 - 3.3. Да се опишат инструментите, които СУРП предоставя
4. Да се създадат приложения чрез разработения прототип на система за управление на работни процеси, които да докажат приложимостта му за решаване на конкретни практически проблеми;
 - 4.1. Създаване на система за извличане на контактна информация от документи на български език;
 - 4.2. Създаване на методи за автоматизирано генериране на метаданни за учебни обекти;
 - 4.3. Създаване на модул за моделиране и управление на виртуални адаптивни курсове, базирани на moodle.

Структура и обем на дисертацията

Работата се състои от увод, четири глави и заключение. Приложен е списък с използвана литература.

В **първа глава** е направен обзор в областта на информационното търсене и извличането на информация, който представя основните модели, задачи, методи и средства за тяхното прилагане и решаване, формулира най-важните изисквания към съвременните информационно-търсещи системи (ИТС), дискутира нерешени проблеми и систематизира съответната специализирана терминология. Специално внимание е обърнато на лингвистичните и статистически методи за анализ и оценяване на съдържанието на текстови документи. Във връзка с моделите за информационно търсене са разгледани основните методи за тематичен анализ и обработка на текстова информация.

Във **втора глава** е представен концептуален модел, който може да бъде използван за описание и бързо прототипизиране на лингвистични процеси.

В **трета глава** е проектирана система за описание и управление на работни процеси използваща предложеният модел. Извършен е анализ на функционалните и нефункционалните изисквания. Описани са основните архитектурни компоненти и реализацията на системата.

В **четвърта глава** са разгледани приложения на прототипа на система за управление на работни процеси. Обърнато е особено внимание на широката приложимост на разработеният модел. Описани са реализации на система за извличане на контактна информация от документи на български език, подходи за автоматизирано генериране на метаданни за учебни обекти и модул за моделиране и управление на виртуални адаптивни курсове, базирани на moodle.

В заключението е направено обобщение на архитектурата и реализираната система. Отбелязани са основните резултати и са очертани възможностите за бъдещо развитие.

Списъкът на използваната литература съдържа 127 заглавия, от които 10 на кирилица, 105 на латиница и 12 интернет-източници.

ГЛАВА 1. ОБЗОР НА СЪВРЕМЕННИ МЕТОДИ И СИСТЕМИ

Търсенето на информация е относително утвърдена технология, която се използва за получаване на множество от свързани материали от голяма колекция. Типичен пример за търсене на информация е търсенето по ключова дума в Интернет, при което резултатът е страници, съдържащи тази ключова дума. Броят на намерени страници, обаче, може да бъде много голям, а и по всяка вероятност, не всичко на дадена страница ще е важно за потребителя. Оттук – преглеждането на всяка страница – резултат на търсенето, с цел откриване на необходимата информация, много често е трудоемка задача, изискваща значителен разход на време. Ето защо се налага процесът на търсене да се допълва с *извличане на информация*, при което в намерените документи се търсят определени фрази (представящи данни, факти и др.), които биха представлявали интерес за потребителя. С други думи, документи с неструктурирано съдържание се трансформират в структури от определен тип.

В настоящата глава е направен преглед в областта на информационното търсене и извличането на информация, който представя основните модели, задачи, методи и средства за тяхното прилагане и решаване, формулира най-важните изисквания към съвременните информационно-търсещи системи (ИТС), дискутира нерешени проблеми и систематизира съответната специализирана терминология. Специално внимание е обърнато на лингвистичните и статистически методи за анализ и оценяване на съдържанието на текстови документи. Във връзка с моделите за информационно търсене са разгледани основните методи за тематичен анализ и обработка на текстова информация.

1.1 Въведение в информационното търсене

Във връзка с бурното развитие на Интернет и на приложните системи, свързани с обработка на големи масиви (колекции) от документи, все по-актуални стават проблемите за организация на ефективен достъп до информация. В днешните условия, решенията на задачи, свързани с информационно търсене и извличане на информация, стават не само приоритетни, но и жизнено необходими за осигуряване на своевременен достъп към информация, интересуваша потребителите. За около 4 години изследвания, списъкът на решаваните задачи значително се разшири, и сега включва въпроси за моделиране, класификация и кластеризация на документи, проектиране на архитектурата на търсещи системи и потребителски интерфейси, езици за заявки, и др.

- задаване/построяване на описания (модели) на всички тематики;
- построяване на описание на разглеждания документ;
- изчисляване на оценки за близост между описанията на тематики и описанието на документа, и избор на най-близки тематики
- непознаване на набора от ключови думи, еднозначно определящи търсения(те) документ(и);
- липса на достатъчен опит и квалификация за формиране на такива заявки;
- отсъствие на приета и недвусмислена терминология в областта, и др.
- начин за представяне на документите;
- начин за представяне на заявките за търсене;
- вид на критерия за релевантност на документи.
- отсъствието на възможности за аранжиране на намерените документи по степен на релевантност, тъй като липсват критерии за нейното оценяване;
- сложността на използване – далеч не всеки потребител може свободно да оперира с булеви оператори при формулиране на своите заявки.

1.2 Организация и методи

Създадени са стотици системи за информационно търсене и хиляди функции за търсене, вградени в различни приложения. Във всички тези случаи (независимо от математическия модел), идеите, софтуерните програми и структурите от данни са достатъчно прости. Съществува мнение, че всяко ново поколение програми е по-съвършено от предходното. Друг краен възглед се състои в това, че „всичко ново – това е добре забравено старо“. По всяка вероятност, истината, в случая на софтуерните системи за информационно търсене, е по средата.

Идеите на 60-те – 80-те години за итеративно уточняване на заявките за търсене, за разбиране на ЕЕ заявки, за търсене по смисъл, за генериране на свързан отговор, и т.н., обаче, все още са неосъществени мечти.

- автоматично определяне на езика, на който е написан документа;
- изключване на думи, които не носят информация (т. нар. стоп-думи);
- сегментиране (tokenisation) – отделяне на думи, дати, имена, изречения;
- лематизация (нормализация, привеждане на словоизменителните форми в основни, вкл. непознати за системата);
- разделяне на сложни думи;
- пълно или частично снемане на омонимия (disambiguates);
- разделяне на именни групи;
- автоматично резюмиране и определяне на ключови думи в текстови документи, и др.;
- формализация на заявката за търсене (представяне от потребителите, в един или друг вид, на техните информационни потребности);
- предварителен подбор на документи по формални признаци на наличие на интересувашата ни информация (например, наличие в текста на документа на една от думите в заявката, ако заявката е формулирана на естествен език);
- анализ на избраните документи (напр. лингвистичен, статистически, и т.н.);

1.3 Характеристики на търсещите системи

Във връзка с извеждане на основните характеристики на съвременните системи за информационно търсене, накратко ще разгледаме някои **специфични черти на големи динамични колекции от документи**, до която е предоставен он-лайн достъп (до голяма степен същото се отнася и до търсенето в Интернет).

Обем на информацията. Характерна особеност е огромния и динамично нарастващ обем на достъпни информационни ресурси (оценки в терабайтове - за Интернет); за нашия правен случай – брой на документите от порядъка на 10^4 . Във връзка с това възникват изисквания към използваните алгоритми (и структури от данни) за организация на търсенето.

Динамика. Дадената особеност значително затруднява използването на общи статистически характеристики на колекцията. Особено поява на

нови страници/документи или промяна на тяхното разположение; по някои статистически данни, средното време за живот на 50% от страниците в Интернет не е повече от 10 дни, ежемесечно около 40% от страниците се обновяват, а обемът на данните се увеличава 2 пъти на всеки 2 години.

Взаимосвързаност. Една от особеностите на документалното информационно пространство е взаимната връзка между представените в него документи/страници. Връзката се реализира със система от позовавания/референции, която може да бъде ползвана при реализацията на някои методи за търсене.

Потребители. Колекциите от документи/страници се ползват от потребители, които се различават значително по квалификация и подготовка. Нещо повече, много от тях не умеят грамотно и ефективно да формулират заявките си за търсене. Статистиката показва, че не по-малко от 60% от заявките за търсене в Интернет са съставени от 1-2 думи (в класическите информационно-търсещи системи това число е 7-9 думи [20]). В резултат се намира огромно количество документи, а резултатите са твърде далеч от реалните информационни нужди на потребителя. Други изследвания върху поведението на потребителите показват, че много от тях не са готови за продължително очакване на резултатите от търсене и анализ на резултатното множество за откриване на необходимите документи: 58% от потребителите се ограничават с изучаване на първия екран с резултати на търсенето, 67% не се опитват да модифицират своята първоначална заявка [20].

Мултиезиковост. Ясна тенденция за създаване на мултиезични информационни среди (Интернет, ЕС), с което на дневен ред се поставят задачите за мултиезиково и кросс-езиково търсене. Решаването на тези задачи предполага реализация на алгоритми за търсене, които са независими от езика на представяне на анализирания документ и от езика на информационните заявки на потребителите.

1.4 Оценка на качеството

Какъвто и да е моделът на търсещата система, тя се нуждае от оценка на качеството на информационното търсене. Оценката на качеството е фундаментална идея в теорията на търсенето, защото именно благодарение на тази оценка може да се говори за приложимост или не, на един или друг модел, както и да се обсъждат/сравняват техните теоретични аспекти и характеристики. Обикновено за оценка на качеството се използват две характеристики:

- точност (precision) – частта на релевантните материали в получения от ИТС отговор;

- пълнота (recall) – частта на релевантните документи в отговора, спрямо общият брой на релевантните документи в колекцията от документи, обект на информационното търсене.

За естествено ограничение на качеството, постигано от конкретни ИТС, служи наблюдението (Donna Harman [5]), че мненията на двама специалисти за релевантност на сравнявани документи съвпадат средно около 40%, като точността и пълнотата са около 65%. От тук следва и примерната горна граница за качеството на системите за търсене, тъй като това качество би могло да се измерва и чрез съпоставяне на мненията на експерти.

1.5 Извличане на информация

С бързото нарастване на източниците на информация и електронни документи, достъпни през Интернет, все повече изследвания се фокусират върху проблемите на извличане на информация [9, 13].

Методи за извличане на информация първоначално са приложени към идентификация на полезна (за потребител) информация от документи, написани на естествен език, с цел - преобразуване на извлечения текст в определен (например, в табличен) формат. С развитието на световната мрежа Интернет, през последните години, извличането на информация започна широко да се използва и за откриване на текстови фрази във Web-страници, които, за разлика от документите на естествен език, все пак са полуструктурирани (имат определена, дефинирана поне с тагове структура).

Според [8], въпреки че *системите за извличане на информация* се различават една от друга, в зависимост от естеството на решаваните от тях задачи, може да се открият поне **5 (пет) общи функционални елемента** в тяхната архитектура:

- сегментация и токенизация (необходими при всяка нормализация на входния текст), и поставяне на етикети (за частите на речта и техните семантични характеристики);
- предсинтактичен анализ на изречения (за идентифициране на синтактични фрази като подлог, пряко допълнение, предложна фраза и т.н., чрез провеждане на синтактичен анализ и разбор);
- търсене и извличане на специфични за съответната предметна област фрази;
- сливане (за идентификация на корелативни връзки с други елементи на текста);
- генериране на шаблони, с попълване на съответните им полета с данни (на базата на извлечените фрази) и създаване

на шаблони за съхраняване в база данни, за отпечатване, и др.

Съществуват два основни подхода за построяване на системи за извличане на информация [4] – експертен (knowledge engineering) и автоматично (само)обучение върху анотирани (или не) корпуси.

Техниките за машинно самообучение се различават една от друга по вида на обучаващите данни [3] - контролирани (двойки обучаващи данни – резултат) или неконтролирани (само обучаващи данни).

1.6 Търсене по документ-образец

Следвайки формулираните по-горе изисквания към системите за текстово търсене, това би означавало осигуряване на подобрени модели на смисловото съдържание на документи, потребителските заявки и информационното търсене, базирани на документи-образци. Анализът на изследвания, свързани с търсене на документи по образец, открива незначителен брой готови и апробирани решения в областта, което е следствие на липса на достатъчно утвърдена теория и сполучливи практики за решаване на задачите за тематичен анализ на неструктурирана естествено-езикова текстова информация. Практическото решаване на задачи за тематичен анализ на документи (дори и в по-тясна предметна област) е актуално не само за реализация на информационно-търсещите системи, но и при проектирането на системи за обработка и анализ на информация. В тази посока могат да се формулират и изследват широк спектър от задачи за интелектуална обработка на информация, вкл. задачи за извличане, идентификация, разпознаване и използване на смислово съдържание на неструктурирани текстови документи.

Частна задача на информационното търсене е задачата за търсене по документ-образец. Документът-образец е една от формите за представяне на информационните потребности на потребителя. Цел на търсенето е откриване на тематично близки документи. При това, като правило, става дума не за търсене на идентични или синтактически близки документи, а за търсене на документи, близки по съдържание и смисъл [2].

Тематиката отразява съдържанието на документа и включва в себе си множество ключови думи, намиращи се в някаква зависимост една от друга. Един от моделите на такава зависимост са теглови коефициенти, отразяващи значимостта на една или друга дума в конкретната тематика. Основен проблем е създаването на модел за оценяване (изчисляване) на тематичната близост на документи. Като правило, резултатът на търсенето са множество документи, в една или друга степен удовлетворяващи условията на (тематично) търсене. Именно моделът, приет за изчисляване на тематичната близост, в крайна сметка определя

релевантността на документите, извлечени в резултат на информационното търсене. Изчислявайки стойностите за тематична близост, тези документи може да се аранжират в зависимост от степента на значимост за потребителя.

ГЛАВА 2. МЕТАМОДЕЛ ЗА ОПИСАНИЕ НА ПРОЦЕСИ ILNET

В основата на архитектурата на ILNET седи лека работна рамка за изграждане на модели на процеси и цялостни решения за моделиране на бизнес процеси. Тя е продукт на дългогодишна разработка на серия от инструменти за различни видове лингвистичен анализ. Едно от най-важните й качества е, че въпреки, че е налице фиксиран модел, според който трябва да се описват и изпълняват процесите, съществува вграден механизъм, посредством който може да се адаптират както графичната репрезентация на моделите, така и начина, по който те се изпълняват, за да се дефинират и изпълняват модели на процеси, с които крайният потребител може да е по-запознат. Понастоящем е в тестова фаза моделирането на мрежи на Петри и подмножество от BPML елементите.

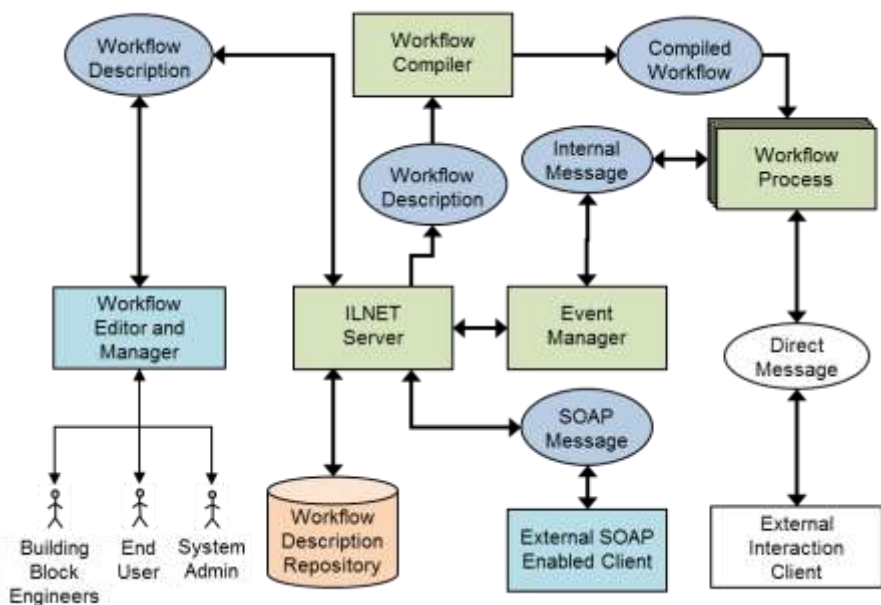
Първата версия на системата бе създадена с идеята да се използва основно в областта на компютърната лингвистика [7]. Тя е използвана успешно в системата СЛОГ за преобразуване на текст в говор [24] в етапа за нормализация на входният текст.

В [19] настоящата версия на системата, вече базирана на .NET, с наименование ILNET, се използва за създаването на прототип на архитектура за моделиране на процеси в е-обучението свързани с курсовият работен поток. На фиг. 1. е представена архитектурата на системата.

2.1 Потребителска перспектива

От потребителска перспектива системата *ILNET* се състои от следните компоненти:

- **Актьори** – те контролират и използват системата:
 - **Инженери на изграждащи блокове (ИИБ)** – дефинират и имплементират основните изграждащи блокове и работни под-потоци в системата;
 - **Крайни потребители (КП)** – използват изграждащите блокове, за да дефинират и изпълняват модели на работни потоци в конкретна област;



Фигура 1. Архитектура на софтуерната система ILNET

- Обекти:
 - **Хранилище с описания на работни потоци (ХОРП)** – централизирано хранилище за работни потоци. Улеснява съвместната работа между различни дизайнери на модели;
 - **Описание на работен поток (ОРП)** – спецификация за дефиниране на работни потоци, която включва описание на модела на процеса както и на начина на неговото изпълнение;
 - **Компилиран работен поток** – изпълнима инстанция на работен поток (ИРП);
 - **Процес на работен поток** – работеща инстанция на работен поток;
 - **Вътрешно съобщение** – вътрешно представяне на съобщение, което носи информация за конкретно събитие. Използва се за комуникация между работещите ИРП и мениджъра на събития (МС);
 - **Външна SOAP услуга** – външно за системата приложение или услуга, която може да комуникира с ИРП, посредством SOAP съобщения, които се обработват от МС;

- Външен интегриран клиент – външен клиент, който може да се свърже и да комуникира директно с ИРП, посредством комуникационни протоколи имплементирани от ИИБ.

Системата е модулна и се състои от пет основни компонента, които комуникират по между си, посредством вътрешни съобщения, или чрез SOAP-базирани уеб услуги. Тези модули са:

- **Компилятор на работни процеси**, който транслира описанията на процесите в изходен програмен код;
- **Сървър за изпълнение и управление** на така описаните процеси;
- **Графичен редактор** за визуално представяне и моделиране на процеси;
- **Мениджър на събития**, който обработва вътрешната комуникация между процесите и предоставя SOAP интерфейс, посредством който външни приложения могат да обменят съобщения с работещи инстанции на процеси;
- **Библиотека от изграждащи блокове**, която съдържа дефиниции на помощни под-процеси, които могат да бъдат използвани за по-бързото моделиране процеси. В допълнение, библиотеката предоставя и блокове, които служат като емулятори на структурни елементи в други съществуващи инструменти за моделиране.

2.2 Съставни елементи на модела

Първичният модел, който се разпознава от компилатора, представлява насочен граф, в който всеки връх отговаря на действие а всяка дъга – възможен преход. При успешно изпълнение на действието в даден връх всички изходящи дъги се проследяват паралелно и работният поток автоматично се разделя на толкова под-потока, колкото е броят на тези дъги.

Едно действие може да бъде описано по два начина – чрез програмен език, или чрез изпълняване на друг работен поток. Във вторият случай от основният поток се изпращат входни параметри на под-потока, след което от последният може да върне резултат под формата на изходни параметри. Всеки път, когато се извиква един под-процес от друг работен процес се създава нова инстанция на под-процеса, която се изпълнява.

2.3 Съвместимост на ILNET модела

Определение 1. Процес в ILNET е комбинация от параметри, работни данни и действия, и може да се определи формално с наредената n -торка

$ILNET = (N, F, V, C, V_0, n_{IN}, n_{OUT}, V_{IN}, V_{OUT})$, където:

- N е крайно множество от върхове, отговарящи на дейности или под-процеси (извикването на под-процес е специален вид дейност);
- $F \subseteq N \times N$ е множество от дъги, което определя последователността от изпълнение на дейностите;
 - $n^*(x) = \{y | (x, y) \in F\}$ е множество на изходните върхове на връх n ;
- $V = V_g \cup V_i \cup V_l$ е множество от променливи, като:
 - V_g е крайно множество от глобални променливи;
 - V_i е крайно множество от междинни променливи;
 - V_l е крайно множество от локални променливи;
- $C: N \rightarrow T$ е функция на трансформациите на дейностите. Трансформациите са процедури описани със средствата на конкретен език за програмиране, които променят състоянието на системата;
- $V_0: V \rightarrow D$ е изображение на променливите върху областта на техните типове, което определя началните им стойности;
- $n_{IN} \in N$ е началната дейност, която се изпълнява;
- $n_{OUT} \in N \cup \emptyset$ е крайната дейност;
- $V_{IN} \subseteq V$ е множество от променливи, определящо входните параметри на процеса;
- $V_{OUT} \subseteq V$ е множество от променливи, определящо изходните параметри на процеса.

Определение 2. Трансформациите в ILNET може да се представят като функции от вида:

$T: S \rightarrow (R \times S)$, където:

- $R = \{r_{success}, r_{fail}\}$ е множество на възможните резултати от изпълнение на трансформацията, според които се определя дали да се продължи към изпълнение на следващите дейности, при резултат $r_{success}$, или не, в противен случай;

- $S: V_M \rightarrow D$ е състояние на системата, съпоставящо стойност от областта D на всяка една от променливите в общото множество на променливите V_M ;
- $V_M = V_g \cup (V_i \times P) \cup (V_l \times P \times B)$ е общо множество на променливите в системата по време на нейното изпълнение, където P е множеството на всички инстанции на процеса, а B е множеството на всички възможни нишки на процесите. Тъй като всяко разклонение на пътя на модела води до създаването на нови негови нишки, то $B \subseteq N^*$, където N^* са всички пътища в насоченият граф $G = (N, F)$.

Една от целите на метамодела е да могат да се представят различни видове формализми за описание на процеси или граматика. Това позволява на така представените формализми да бъдат комбинирани и използвани за решаване на конкретна задача.

Твърдение 1. Мрежите на Петри могат да бъдат изразени използвайки елементите на предложеният метамодел.

Ще използваме формалната дефиниция за маркирани мрежи на Петри. Мрежа на Петри (маркирана) е наредената четворка (S, T, W, M_0) , при която:

- $S = \{s_1, s_2, \dots, s_n\}$ е крайно множество от позиции;
- $T = \{t_1, t_2, \dots, t_n\}$ е крайно множество от преходи;
- $S \cap T = \emptyset$;
- $W : (S \times T \cup T \times S) \rightarrow \mathbb{N}$ е мултимножество от дъги, (съпоставя цяло неотрицателно число на всяка дъга);
- $M_0 : S \rightarrow \mathbb{N}$ е началната маркировка на Петри, където под маркировка $M : S \rightarrow \mathbb{N}$ се разбира съпоставяне на определен брой ядра (tokens) на всяка позиция от S .

Релацията на потока може да се опише и като двойка матрици с размери $|S|$ на $|T|$:

- W^- , определена с $\forall s, t: W^-[s, t] = W^-(s, t)$;
- W^+ , определена с $\forall s, t: W^+[s, t] = W^-(t, s)$.

Един преход t е в готовност в една маркировка M , тогава и само тогава, когато $\forall s \in S: M(s) \geq W(s, t)$.

Ще се опитаме да създадем общ ILNET метамодел, за описание на мрежи на Петри. Един начин за описание на тази мрежа с ILNET метамодел, е като създадем моделът $ILNET = (N, F, V, C, V_0, n_{IN}, n_{OUT}, V_{IN}, V_{OUT})$ със следните елементи:

- $N = T \cup \{n_{init}\}$
- $F = \{(x, y) | x = y; x \neq n_{init}\} \cup \{(x, y) | x \neq y; x = n_{init}\}$, за $x, y \in N$
- $V = V_g \cup V_i \cup V_l$
- $V_g = \{v_{in}, v_{out}, v_{w^-}, v_{w^+}\}$
- $V_i = S_v \cup T_v$
- $V_l = \emptyset$
- $V_0 = \{S_v \mapsto M_0\} \cup \{T_v \mapsto (t_{v_1}, t_{v_2}, \dots, t_{v_k}) | k = |T|; t_{v_i} = true\}$
 $\cup \{v_{in} \mapsto \{t \mapsto s | W(s, t) > 0; t \in T\}\}$
 $\cup \{v_{out} \mapsto \{t \mapsto s | W(t, s) > 0; t \in T\}\}$
 $\cup \{v_{w^-} \mapsto W^-\} \cup \{v_{w^+} \mapsto W^+\}.$
- $n_{IN} = n_{init}$
- $n_{OUT} = \emptyset$
- $V_{IN} = \emptyset$
- $V_{OUT} = \emptyset$
- $C(n_{init}) = \{V_M \mapsto (r_{success}, V_M)\}$

Тъй като всички променливи в модела принадлежат на V_i , то $V_M = V_i \times P$, инстанциите на процесите се изпълняват напълно независимо, а нишките на една инстанция могат да се разглеждат като система от напълно симетрични паралелно протичащи под-процеси, които работят само върху глобални (за тях, междинни в системата на ILNET модела) променливи. Можем да опишем дейностите $C(n)$ за $n \in T$ чрез псевдокодът:

```

if  $T_v[n] = true$  and foreach  $s$  in  $v_{in}(n) : S_V[s] \geq v_w[s, n]$  then
    foreach  $t_{v_i}$  in  $T_v$  set  $t_{v_i} = true$ 
    foreach  $s$  in  $v_{in}(n)$  set  $S_V[s] = S_V[s] - v_{w^-}[s, n]$ 
    foreach  $s$  in  $v_{out}(n)$  set  $S_V[s] = S_V[s] + v_{w^+}[s, n]$ 
else
     $T_v[n] = false$ 
    if foreach  $t_{v_i}$  in  $T_v : t_{v_i} = false$  then return false;
return true;

```

2.4 Употреба и приложение на модела

Използвайки функционалността на ILNET модела за работни потоци, можем да създаваме блокове, които имплементират различни шаблони. По време на изпълнението под-потоците имат достъп до извикващия ги поток, ако има такъв, както и до контекста, в който те са извикани. Използвайки тази информация могат да се моделират сложни контекстно-зависими блокове, които да изпълняват действия на базата

на решения върху акумулирани серии от събития. Един такъв пример е шаблона за синхронизирано сливане (AND-Join, паралелно събиране) (фиг. 2.) имплементиран с използването на един единствен връх с действие описано в кратка C# функция и няколко глобални променливи за акумулиране на данни.

Всеки път, когато се извика върха *gateway*, от даден поток, за първи път, под-потока посредством рефлексия върху модела на извикващият поток намира броят на всички входящи дъги към върха, който извиква И-блока. От този момент, всеки следващ път, който навлезе в изпълнение на същият И-блок, *gateway* върха проверява коя входяща дъга в извикващият поток е била проследена и акумулира входа. В момента, в който е налице сигнал от всяка една от възможните входящи дъги, И-блока вместо само да акумулира сигнала, го пуска (генерира сигнал за успешно изпълнение на върха) и се изтриват акумулираните данни, позволявайки на следващи разклонения (цикли) на над-потока да използват отново същият И-блок (при положение че над-потока е правилно моделиран).

Следвайки тази процедура, може да се реализират редица от разнотипни изграждащи елементи за достъп до данни, обработка на съобщения, използването на уеб-услуги.

Предишната версия на системата [10] е използвана успешно като инструмент за описание на процеси при нормализацията на текст в системата за преобразуване на текст в говор SLOG [15].

В [17] е използвана ILNET системата за моделиране и управление потоците на курсовете в система за е-обучение.

Системата *ILNET* е интегрирана и в приложения, целящи разширяването на популярни системи за е-обучение с възможности за описание на работни потоци [18]. Решението използва уеб услугата на *ILNET*, за да свърже съществуваща система със сървър за изпълнение на работни потоци.

2.5 Изводи

Метамоделът *ILNET* предоставя метод за бързо имплементиране на прототипи на различни видове модели на работни потоци и шаблони. Интегрирайки обектно-ориентиран програмен език в първичният модел, и давайки възможност на потребителските функции да получават информация за структурата и данните на изпълняваният модел, могат да се реализират компактни и интелигентни изграждащи блокове.

Вграденият MC, комбиниран с приложен програмен интерфейс за използване като уеб услуга, позволява интеграцията на модели на работни потоци в почти всяка една съществуваща клиентска система.

Сред възможните бъдещи развития на разработката са:

- Имплементиране на изграждащи блокове и шаблони за изпълнение, които да позволят моделирането на известни стандарти за описание на процеси/потоци;
- ILNET имплементация на оцветени мрежи на Петри;
- Разработка на методи за автоматичен анализ на процесите с цел подобряване на производителността [21];
- Интеграцията на ILNET системата в други практически приложения (персонализирано и адаптивно обучение [11, 12], е-бизнес [23], административен контрол [22], и др.).

2.5.1 Решени задачи

Основните резултати получени във втора глава могат да се формулират по следния начин:

- Анализирани на предложения метамодел за моделиране и управление на процеси и реализирани на прототип на базата на този модел, който да включва.
- Дефиниране на изискванията на метамодела. Предложени са функционалните изисквания към реализацията на метамодела, както и нефункционалните изисквания.
- Проектиране на софтуерна архитектура и разработка на програмен прототип, базиран на тази архитектура с цел изясняване на изискванията към системата.

ГЛАВА 3. АРХИТЕКТУРА И РЕАЛИЗАЦИЯ НА СУРП

Целта на настоящата глава е да опише архитектурата на софтуерна платформа за моделиране и управление на процеси, следвайки метамодела от глава 2, и да покаже как различни информационни процеси могат да бъдат моделирани, реализирани и интегрирани в други решения.

Самата архитектура ще послужи за изграждане на два софтуерни прототипа. Единият софтуерен прототип, от тип *редактор*, ще позволява създаване на екземпляри на метамодела, а вторият софтуерен прототип – „*среда за изпълнение*”, ще позволява проиграване на екземпляри на метамодела, изградени с първия прототип.

3.1 Анализ на изискванията към реализацията на модела

С оглед разработване на софтуерна архитектурна рамка на описания метамодел за моделиране и управление на процеси, бяха съставени функционални и нефункционални изисквания към софтуерните системи, поддържащи този модел.

3.1.1 Функционални изисквания

Функционалните изисквания се поставят на база функционалността на отделните компоненти, изграждащи цялостното решение.

Могат да се поставят следните функционални изисквания към реализацията на метамодела:

- Създаване и редактиране модели на процеси:

Описание	Този случай на употреба описва процесите на създаване и променяне на модели
Участници	Потребител
Пред-условия	Трябва да е осъществена връзка с ILNET сървър
Пост-условия	Няма
Нормална последователност на събитията	<ol style="list-style-type: none">1. Потребителят преглежда хранилището на ILNET метамодела.2. Потребителят създава нов модел в избраната от него директория на хранилището.3. Потребителят зарежда създадения модел в графичният потребителски интерфейс.4. Потребителят редактира модела дефинирайки съставните му работни потоци (процеси) и описвайки техните елементи.5. Потребителят запазва модела, изпращайки го за съхраняване в хранилището на метамодела.6. Потребителят стартира работна инстанция на модела.

- Редактиране на работен поток:

Описание	Този случай на употреба редактирането на работен поток в рамките на конкретен ILNET метамодел
Участници	Потребител
Пред-условия	Трябва да е осъществена връзка с ILNET сървър
Пост-условия	Няма
Нормална последователност на събитията	<ol style="list-style-type: none">1. Потребителят отваря прозореца за редактиране на метамодела.2. Потребителят избира дали да добави или да изтрие елемент.3. При избор за добавяне на нов елемент потребителят избира между:<ul style="list-style-type: none">• Добавяне на възел• Добавяне на променлива• Добавяне на подпроцес

	<p>4. При избиране на съществуващ възел потребителят има възможност за промяна на неговата функционалност.</p> <p>5. Потребителят може да добави, изтрие или промени връзка между възли и подпроцеси.</p> <p>6. Потребителят може да променя свойствата на елементите.</p> <p>6.1. При промяна на свойствата на възел потребителят има избор между :</p> <ul style="list-style-type: none"> • Промяна на име • Промяна на кода • Избор дали елемента е начало на работния поток • Избор дали елемента е край на работния поток <p>6.2 При промяна на свойствата на променлива потребителят има избор между:</p> <ul style="list-style-type: none"> • Избор дали променливата е входен параметър • Избор дали променливата е изходен параметър • Промяна на тип • Промяна на стойност • Промяна на име • Промяна на област на действие : <ul style="list-style-type: none"> ○ Локална ○ Междинна ○ Глобална <p>6.3 При промяна на свойствата на подпроцес потребителят има избор между:</p> <ul style="list-style-type: none"> • Промяна на име • Промяна на името на подпроцеса • Избор дали елемента е начало на работния поток • Избор дали елемента е край на работния поток <p>7. Потребителят запазва модела, изпращайки го за съхраняване в хранилището на метамодела.</p> <p>8. Потребителят стартира работна инстанция на модела</p>
--	--

3.1.2 Нефункционални изисквания

Следните нефункционални изисквания се дефинират към реализацията на метамодела:

- Разширяемост – да е лесен за адаптиране и да позволява бъдещо разширяване;
- Скалируемост – възможност за едновременно ефективно обслужване на голям брой конкурентни потребители;
- Преносимост – трябва да бъде лесно преносим на различни софтуерни платформи;
- Надеждност – трябва да предоставя надежден достъп до съдържанието;
- Лесна поддръжка и модификация на компонента: чрез разделяне на бизнес логиката от съдържанието и, от друга страна, на разделяне на съдържанието от представянето му;
- Минимизиране на цената на разработка чрез използване на безплатни и базирани на отворен код софтуерни технологии и среди за разработка.

3.2 Проектиране на прототип на система за управление на работни процеси

В тази точка са изброени накратко основните функции на системата (бизнес логиката на приложението), и графичният потребителски интерфейс, който ги реализира. Изброени са отделните визуални компоненти, както и функциите, които ще изпълняват, в това число и действията, които ще се извършват при настъпването на определено събитие.

Проектирането на графичен интерфейс се състои, от една страна в дефинирането на визуалната част, чрез която клиентите ще използват системата (прозорци, изгледи, перспективи, таблици, текстови полета и т.н.), а от друга в дефинирането на конкретните действия, които ще извършва системата, когато клиентът предизвика определени събития, посредством графичните елементи за управление на действията (бутони, връзки и т.н.).

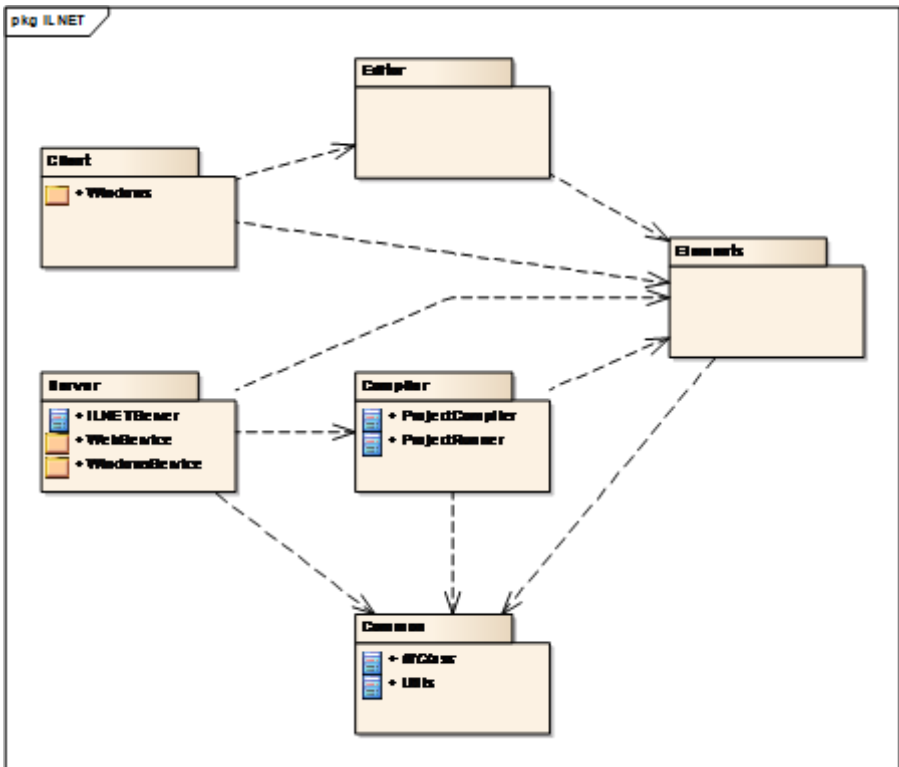
3.3 Описание на реализацията

3.3.1 Пакетна структура на приложението

Приложението е реализирано под формата на няколко пакета. Описанието на реализацията се прави общо за двата софтуерни прототипа. На фиг. 2. е представена пакетната структура на по-главните пакети.

Във всеки от тях се съдържат класове със строго определена функционална насоченост. Функционалността реализирана от отделните пакети е следната:

- Пакетът **Elements** съдържа реализацията на ILNET модела под формата на C# класове.
- Пакетът **Common** съдържа класове и дефиниции предоставящи обща помощна функционалност.
- Пакетът **Compiler** съдържа класове, които предоставят методи за компилиране и изпълнение на работни потоци описани чрез ILNET метамодела.
- Пакетът **Editor** съдържа класове, които предоставят методи за визуализация и редактиране на визуалните характеристики на ILNET метамодела.
- Пакетът **Server** съдържа приложения и библиотеки от класове, които реализират сървърната част на архитектурата на приложението.
- Пакетът **Client** съдържа класове, които реализират клиентската функционалност от архитектурата на приложението, в това число и графичният потребителски интерфейс за редактиране на ILNET модели.



Фигура 2. Пакетна структура на приложението

3.3.2 Пакет ILNET.Elements

В този пакет се съдържа реализацията на ILNET модела под формата на C# класове. Имената на класовете отговарят на имената дефинирани в XML схема файлът на модела (приложение 1). Тези класове са с проста структура и служат единствено за предоставяне на лесни методи за създаване и редактиране елементи на модела, сериализация на проектите от и към XML, както и за трансформиране на проектите до изпълним изходен код. Главен клас на пакета е *Project*, в който се хранява цялото описание на един ILNET метамодел. От него директно или косвено могат да се зададат и извлекат всички елементи на метамодела. Той съдържа и методите за сериализация и десериализация на метамоделите. Всички наследници съдържат метод за опилиране (трансформиране) на описанието на модела до текст, посредством подаден шаблон. Тези методи се използват за генерирането на изпълним изходен код от метамодела. Ролите на другите класове са следните:

- Класът **Process** представя един работен поток от метамодела и предоставя методи за редактиране на действията и променливите свързани с даден поток.
- Класът **Variable** представя една променлива в работен поток и предоставя методи за редактиране на името, обхвата, типа и стойността по подразбиране на променливата, както и дали е входяща (параметър на потока) и/или изходяща (резултат от изпълнението на работният поток).
- Класът **Node** представя едно действие (функция или под-поток) в работен поток и предоставя методи за редактиране на функцията или името и параметрите на под-потока.
- Класът **ProcessParameter** представя стойността на един параметър при извикване на под-процес от действие (Node) на текущият процес и предоставя методи за редактиране на името (посредством `Element.Name`) и стойността на параметъра.

3.3.3 Пакет ILNET.Common

Този пакет съдържа помощни класове, които предоставят обща помощна функционалност и дефиниции на делегати. Този пакет се компилира като самостоятелна библиотека и се използва от другите библиотеки на системата. По този начин става възможно дефинирането на обекти или делегати в една компилирана библиотека или програма, които безпроблемно да се подават към и използват от други библиотеки / програми заредени в рамките на един .Net приложен домейн. Ролите на класовете в пакета са следните:

- Класът **iliClass** представя един работен поток от метамодела и предоставя методи за редактиране на действията и променливите свързани с даден поток.
- Класът **Utils** предоставя методи за клониране и клониране в дълбочина на динамични структури от данни и методи за извличане и търсене със замяна на текст, които се използват за обработката на шаблоните при компилирането на метамодели.

3.3.4 Пакет ILNET.Compiler

Този пакет съдържа класове, които предоставят методи за компилиране и изпълнение на работни потоци описани чрез ILNET метамодела. Ролите на класовете в пакета са следните:

- Класът **ProjectCompiler** използва C#.NET компилаторът, за да компилира изпълнима библиотека от генерираният изходният код за даден ILNET метамодел. В резултат създава инстанция на класа ProjectRunner.
- Класът **ProjectRunner** предоставя методи за стартиране и изпълняване на инстанции на описаните в изходният метамодел работни потоци.

3.3.5 Пакет ILNET.Editor

Този пакет съдържа класове, които обгръщат класовете реализиращи ILNET метамодела, и предоставят методи за визуализация и редактиране на визуалните характеристики на ILNET метамодели. Класовете реализират функционалността за визуализиране взаимодействие с отделните видове елементи на ILNET метамодела както следва:

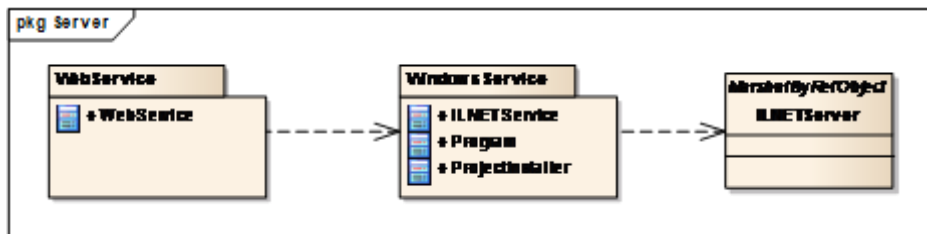
- Класът **GProject** съдържа списък от всички работни процеси в метамодела. Той няма визуална функционалност, а само редакционна – свързана с добавянето, премахването и преименуването на процеси.
- Класът **GProcess** предоставя методи за визуализация, редактиране и взаимодействие с работни потоци.
- Класът **GElement** е абстрактен клас, който реализира част от общата функционалност за изчертаване и обработка на събития от графичната среда и се наследява от всички класове на библиотеката, които имат визуално представяне.
- Класът **GNode** е абстрактен клас, който наследява **GElement**, и добавя методи и свойства, характерни за елементите от тип действие (Node) или променлива (Variable).
- Класът **GConnection** предоставя методи за визуализация, редактиране и взаимодействие с връзките между елементите.

Главен клас на пакета е *GProcess*, който е основна входна точка за визуализиране и редактиране на описания на процеси. Той делегира заявките за изчертаване и обработка на събития към останалите елементи на модела, ефективно капсулирайки тяхната функционалност. Всички връзки между елементите в един процес се индексират в класът *ConnectionMap*, който предоставя методи за добавяне и премахване на връзки, както и за търсене на връзки по начален или краен елемент.

Класът *GConnection* реализира визуализирането на връзките между елементите. Всяка връзка помни елементите (*GNode*) и местата на прикачване към тези елементи на началото и края на връзката. Към момента, стандартният начин за изчертаване на връзки използва свързани кубични криви на Безие, чиито контролни точки се съхраняват в структурата *Anchor*, и като метаданни към елемента прикрепен към тяхното начало.

3.3.6 Пакет ILNET.Server

Този пакет съдържа класове, които реализират сървърната част на архитектурата на приложението. Те включват пакети с класове реализиращи различни видове точки за достъп до сървъра и библиотеката *ILNETServer*. По-долу на фиг. 3. е представена пакетната структура на сървърните компоненти. Класът *ILNETServer* предоставя хранилище, средства за изпълнение и средства за обработка на съобщенията от/към и между изпълнявани инстанции на *ILNET* метамоделите. Другите пакети съдържат прокси класове, които предават получените заявки към съответните методи *ILNETServer*.



Фигура 3. Пакетна структура на сървърните компоненти

Функционалността реализирана от отделните пакети е следната:

- Пакетът **WindowsService** съдържа класове реализиращи Windows системна услуга (Service), която хоства инстанция на класа *ILNETServer* и регистрира слушател на TCP/IP порт, посредством което става възможност отдалечено извикване на методите на сървъра от други .NET Remoting приложения. Тази услуга предоставя бърз и удобен начин за свързване и

работа със сървъра, но е технологично ограничена само за приложения писани на .NET.

- Пакетът **WebService** съдържа реализацията на ASP.NET Web услуга, която пренасочва всички получени заявки към конкретна ILNET Windows системна услуга. Тази услуга дава възможност на широк набор от приложения да се свържат с ILNET сървъра, посредством предоставянето на WSDL дефиниция и използването на SOAP съобщения при комуникация с услугата.

3.3.6.1 Класът *ILNETServer*

Класът *ILNETServer* предоставя хранилище, средства за изпълнение и средства за обработка на съобщенията от/към и между изпълнявани инстанции на ILNET метамодели и методи за управление на тези инстанции от другите компоненти. Дефинирани са следните видове етоди за работа с инстанции на сървъра:

- Методи за работа с хранилището на ILNET метамодели. Хранилището използва локалната файлова система за съхраняване и извличане на XML описанията на моделите, и позволява групирането им в пакетна (директорна) структура.
- Методи за компилиране, зареждане в паметта и освобождаване на паметта от изпълними ILNET метамодели намиращи се в хранилището на сървъра.
- Методи за стартиране и прекратяване на инстанции на заредените в паметта изпълними метамодели.
- Методи за обработка на съобщенията от събитията, които настъпват в изпълняваните процеси
- Методи за извличане на получените резултати от изпълняваните инстанции на метамодели. Една инстанция може да произведе множество резултати по време на нейното изпълнение (всеки сигнал успешно преминал последното действие на основният работен поток на метамодела произвежда резултат).

3.3.7 Пакет *ILNET.Client.Windows*

Този пакет съдържа класове, които реализират графичният потребителски интерфейс на системата. Той е изграден на базата на приложният потребителски интерфейс на .NET библиотеката Windows Forms и с помощта на средствата на Visual Studio за потребителски интерфейс – форми, менюта, ленти с инструменти, събития и др., както и на някои библиотеки, които предоставят разширена функционалност и потребителски удобства при работа с клиента. Сред използваните външни библиотеки са:

- Библиотеката DockPanel Suite предоставя удобна рамка за реализирането на много-прозоречни потребителски интерфейси, които да се доближават по функционалност и гъвкавост до прозоречната система на средата Visual Studio.
- Библиотеката ScintillaNET, която предоставя .NET обвивка на компонентите на библиотеката Scintilla за оцветяване и редактиране на изходен програмен код на различни езици, включително и C#. Използва се за предоставяне на вграден редактор на кода на функциите в работните потоци, като по този начин се улеснява разработката на работни потоци от по-ниско ниво (изграждащи блокове).

В клиента е използван много-документен интерфейс – Multiple Document Interface (MDI). Това прави възможно едновременното отваряне и редактиране на няколко работни потока от различни ILNET метамодели.

3.3.7.1 Класът GraphDoc

Класът *GraphDoc* реализира визуализирането и редактирането на един работен поток от ILNET метамодел. Той съдържа графични компоненти от тип ленти с инструменти, панели, прозорец за изчертаване на изображения и панел за редактиране на изходният код на функциите.

В прозорецът за изчертаване, посредством класовете на ILNET.Editor пакета, се показва избраният работен поток, а събитията за взаимодействие (интеракция) с мишката се предават на инстанцията на класа *GProcess*. Класовете от пакета *PropertyClasses* служат за показване и редактиране на характеристиките на различните видове визуални елементи.

3.3.7.2 Класът RepositoryExplorer

Класът *RepositoryExplorer* реализира визуализирането и навигирането в хранилището на ILNET сървъра. Той съдържа един графичен компонент от тип *TreeView*, в който динамично зарежда и показва структура на хранилището и проектите в него. За всеки проект като под-елементи се показват всички негови работни потоци.

3.3.7.3 Класът PropertyWindow

Класът *PropertyWindow* реализира визуализирането и редактирането на характеристики на обекти. Той съдържа компонент от тип *PropertyGrid*, посредством който се реализира визуализирането и редактирането на полетата от елементите на ILNET метамодела. Дефинирани са помощни класове, които описват данните на отделните елементи, и обработват събитията свързани с промяна и опресняване на данните.

3.3.7.4 Класът MainForm

Класът *MainForm* реализира основният прозорец на графичният потребителски интерфейс на клиентското приложение. Той съдържа графични компоненти от тип ленти с инструменти, и инструментални панели и на свой ред обработва голяма част от събитията, които възникват в отворените документни прозорци. Използва една инстанция на класа *RepositoryExplorer* за показване на съдържанието на сървърното хранилище, и една инстанция на класа *PropertyWindow* за показване и редактиране на характеристиките на селектираният в някой от документните прозорци елемент. За всеки отворен работен поток създава нов документ от тип *GraphDoc*. Реализиран е много-документен интерфейс, посредством панела *DockPanel* на библиотеката *DockPanelSuite*.

ГЛАВА 4. ПРИЛОЖЕНИЯ НА РАЗРАБОТЕНИЯ МОДЕЛ

4.1 Извличане на контактна информация от документи на български език

Извличането на структурна информация от неструктурирани текстови документи е задача, чиито интерес постоянно се увеличава с нарастване количеството информация достъпна в неструктуриран формат (напр. Интернет). Разработена е система предназначена за улесняване потребителя при създаването на автомати и граматики за анализ на текстове, използвайки регулярни изрази, морфологичен анализатор, връзка с БД и вграден интерпретатор. Тази функционалност позволява софтуерният продукт да бъде използван като средство за описание процеса на извличане на информация. При проведен експеримент с извличане на контактна (адресна) информация са постигнати висока степен на откриване и прецизност на получените данни. Използваният разширен синтаксис на регулярни изрази и граматика спомагат за полесното имплементиране на необходимите за подобрението на анализа корекции и нововъведения.

В основата си системата се състои от изпълним модул, който съдържа в себе си библиотеките за интерпретиране на разширените граматики и регулярни изрази описани в предишните секции и код за синтезиране на XML код съдържащ откритите при анализ елементи в съответствие с използваната граматика.

Тъй като най-голямото хранилище на свободно достъпна информация е Интернет е добре да се обърне внимание на предварителната обработка на документите преди извършване на по-нататъшни анализи върху тях. На фиг. 4 е показан работният поток за анализ на хипертекстов документ използван в настоящата разработка.

Това са някои от най-съществените неща, за които трябва да се внимава при работа с хипертекстови документи:

- Кодовата таблица. Следните фактори са полезни при определяне правилната кодировка за документа: средна дължина на думите и брой азбучни символи. Те помагат за различаването на едно-байтови от мулти-байтови кодировки.
- Ненужни данни – това са скриптове, скрити полета, коментари и др. които не се използват по време на анализ или могат да доведат до смъкване прецизността.
- Специални символи – някои символи могат да присъстват повече от веднъж в дадена кодова таблица под различни кодове с различни семантики. Ако анализаторът не диференцира тези семантики е добре тези символи да се преобразуват до един единствен, за да се улесни писането на изразите.

При създаването на граматиката и регулярните изрази е използван процес подобен на подходите за гъвкава разработка на софтуер. Тъй като използваните разширения в граматиката и синтаксисът на регулярните изрази правят анализаторът лесен за разбиране и промяна във всеки един момент от неговото доразработване, се губи необходимостта от създаване на напълно точни изрази и граматика от самото начало. Вместо това е използван подход подобен на принципа на опита и грешката, при който се редуват коригиране на анализатора и тестване върху досегашните страници + нови.



Фигура 4. Работен поток за анализ на хипертекстов документ

На таблица 1 са показани процентът на разпознаване на различните под-елементи от търсената контактна информация при извършен анализ върху 149 фирмени сайта на български фирми. Всяка от разгледаните редакции представлява модифицирана версия на предишната. Ясно се вижда как с всяка модификация на граматиката и/или регулярните изрази в по-горна редакция се подобрява точността на работа на анализаторът. Това се дължи от части на условието че всяка корекция трябва да не влошава точността при пускане на анализаторът върху вече вкараните в тестовата група документи.

Последната 12-та редакция съдържа регулярни изрази с обща дължина 187 961 символа. По-голямата и дълга част от тях са автоматично генерирани изрази разпознаващи различни видове населени места в страната (184 117), които са използвани като вмъкнати под-изрази в регулярният израз за адрес.

Разработеният подход за извличане на информация от текстови документи използва средства, позволяващи лесното реализиране на различни модули за извличане на данни. Визуалната среда за описание на граматиката може да се използва и като инструмент за моделиране на процеси на анализ и обработка на извлечената информация а интуитивният интерфейс прави създаването и редакцията на граматиките лесно и бързо. Проведеният експеримент с прогресивно подобряване на анализаторът дава добри резултати.

Редакция	Адреси	Телефони	Имейли	Линкове
10	75.07%	85.82%	100.00%	80.00%
11	86.54%	96.65%	100.00%	85.00%
12	89.54%	96.54%	100.00%	90.63%

Таблица 1: Резултати при извличане на контактна информация от хипертекстови документи.

4.2 Автоматизирано генериране на метаданни за учебни обекти

С навлизането на е-обучението се появява необходимостта от създаване на софтуерни системи за автоматизирано генериране на електронни курсове. Изключително важно за всеки такъв курс е описанието на характеристиките на включените в него учебни обекти (LO - Learning objects), т.е. създаването и/или генерирането на метаданните.

Основен стандарт в областта на метаданните за обекти, използвани в обучението е LOM (Learning Object Metadata) [14], според който метаданните – на брой 58, са разпределени в девет групи: 1. Основни (General), 2. Продължителност на съществуване (Life Cycle), 3. Данни за метаданните (Meta-Metadata), 4. Технически характеристики (Technical), 5. Образователни (Educational), 6. Права (Rights), 7. Връзки (Relation), 8. Анотация (Annotation) и 9. Класификация (Classification). Всяка група съдържа определен брой полета, които я характеризират в детайли. Освен че описват съдържанието на учебния обект, метаданните се отнасят и до други елементи на процеса на обучение. С използване на метаданни, аташирани към съответните учебни обекти, се улеснява тяхното търсене и обмен, като значително се повишава качеството на администриране на системите за е-обучение. От друга страна, създаването и въвеждането на метаданни е трудоемка и високо квалифицирана експертна дейност, което налага търсене на различни начини за тяхното автоматизирано генериране.

Според [16], автоматизираното генериране на метаданни за даден учебен обект е резултат от машинна обработка, в която участват единствено разработчик, разпространител на софтуер и инициатор на процеса. Последното е валидно само при условие, че метаданните за даден LO са статични и неизменяеми при участие на дадения обект в различни процеси на виртуално обучение. Подобно, между другото, е и разбирането на повечето автори, които предполагат (явно или не), че метаданните, съпровождащи учебните обекти (вкл. тестовите единици), са относително неизменни – независимо от процесите, в които участват. Подобна предпоставка силно стеснява възможностите за удовлетворително решаване на проблема за автоматизирано генериране на метаданни за LO. Например, генерирането на метаданни, свързани с трудността, областта, степента на гранулираност, типична възраст на обучаемите, средното време, необходимо за усвояване на материала, и др. под. на даден LO, няма как да не зависи силно от конкретния контекст на провежданото обучение. Нещо повече, динамичният характер на редица метаданни за LO (напр. трудност), едва ли може да бъде поставен под съмнение (напр. извън зависимост от постиженията на конкретна група обучавани). Очевидно, че за решаване на посочения проблем, трябва да бъдат изследвани и други подходи.

Целта на разработката е да систематизира и изследва подходящи подходи за извличане на метаданни за LO. За провеждане на експерименти е избран стандарта LOM, поради неговото широко използване и факта, че ще се изследват материали от различни области. Проучването на различни методи за автоматизирано извличане на метаданни показва, че съществуват добри възможности за минимално участие на автора на LOS в съставянето на визитка на учебния обект. Както беше посочено, може да се разграничат 5 (пет) основни групи

подходи. Изследването на методи от първата група (акумулативни въпроси, генерирани в резултат на виртуално обучение с различни групи обучавани) се оказва особено продуктивно, и ще бъде предмет на следващи публикации. Особено предизвикателна е задачата по изграждане на курсове за е-обучение от нулата. Последното означава – стартиране на виртуално обучение от набор LO, свързани със съответната предметна област (дори при отсъствие на съпровождащи метаданни и банка от тестови въпроси), и постепенното генериране (на базата на подходящи акумулативни тестови единици и на метаданна за всеки използван в обучението LO, и на тестови въпроси от различен тип (на базата на естествени дистрактори – резултат от колекционирани и оценени отговори на обучаваните в резултат на зададените акумулативни въпроси). Перспективно направление на изследванията в областта на интелигентните системи за е-обучение е свързано с развитие на една друга идея – включване в процеса на виртуално обучение и на акумулативни въпроси, свързани с когнитивни равнища на знанието (напр. по таксономията на Блум).

4.3 Моделиране и управление на виртуални адаптивни курсове, базирани на moodle

Системите за управление на потокови структури (WMS) дават възможност за дефиниране на различни процеси или дейности, в които участват и си взаимодействат множество потребители. При достигане на всяка отделна стъпка на потока, WMS осигурява комуникация и включване на потребителите и ресурсите, необходими за изпълнението. След приключване на съответната стъпка (по един или друг начин – вкл. с обработка на грешки и частично приключване), WMS трябва да подготви необходимите данни и да осигури включване на лицата, отговорни за изпълнението на следващата (в зависимост от потоквата структура) стъпка.

Броят на софтуерните продукти, базирани на WFM-технологиите, постоянно нараства. Въпреки това, до момента, няма стандарти, които да позволяват съвместна работа на WFM-продукти, представящи различни процеси. Основаната през 1993 г. коалиция WfMC обединява проектантите, консултантите, аналитиците, университетските и изследователските групи с интереси в областта на потоквите структури и BPM. Дело на WfMC е създаването и разпространението на стандарти в изследваната област с цел осъществяване на комуникация между иначе несъвместими системи за управление на потоци и интегрирането им с различни ИТ-услуги.

Съвременните системи за електронно обучение (CeO) са фокусирани върху осигуряване на отделни виртуални учебни дейности, оставяйки на заден план процеса на обучение (в рамките на който се провеждат) и свързаните с тях изисквания и ограничения от логически и методически характер. Иначе казано, съвременните CeO не са подходящи за

моделиране на разнообразните форми на обучение, наблюдавани в практиката. Нещо повече, дори опитите за представяне на адаптивни виртуални учебни курсове в подобни системи среща определени трудности (вж. [25]).

Удовлетворяването на подобни **изисквания за адаптивност и приспособимост на виртуалния учебен процес** (едновременно със спазване на определени методически и ресурсни ограничения), обаче, поставя на дневен ред редица проблеми, свързани със създаването на адекватни средства за неговото моделиране, изпълнение в реално време и управление (вкл. адаптивни потребителски интерфейси, средства за автоматизация и др.). В последните години, за решаване на посочените проблеми опити се привличат технологии и средства за WMP.

В основата на настоящата разработка е общ подход, същността на който се заключава в изследване на възможността за интегриране в CeO с отворен код (*широко разпространена*) на елементи на WMS (*с широко предназначение*). Съществено е да се отбележи – какво се разбира тук под WMS с широко предназначение. В случая става въпрос за WMS, която (за разлика, например, от изброените по горе основни подходи) притежава средства за моделиране и управление не само на бизнес-процеси, но и на компонентите на самата потокова структура (дейности, подпроцеси, ресурси, ограничителни условия и др.). Иначе казано, в подобни системи могат предварително да се моделират елементите на бизнес-средата (в случая – е-обучението), които се използват при дефиниране, изпълнение и управление на съответните процеси, след което системата започва да действа като специализирана (за целите на е-обучението) WMS.

Предложеният за реализация подход има няколко важни **предимства**:

- преодолява синдрома още една CeO за проектантите и потребителите (интерфейса и функционалността на избраната популярна CeO се запазват);
- позволява реализация на бърз прототип на CeO с управление на процесите (не е необходимо да се проектира и реализира от нула – използват се възможностите на избраната популярна CeO, като специфичните компоненти на е-обучението се моделират със средства на WMS с широко предназначение);
- разширява функционалността на конкретната CeO в посока на използване на технологии и средства за моделиране и управление на виртуални учебни процеси;
- с развитието на избраната CeO (напр. с включване на нови виртуални учебни дейности и събития) автоматично се обогатява функционалността на системата като цяло (в

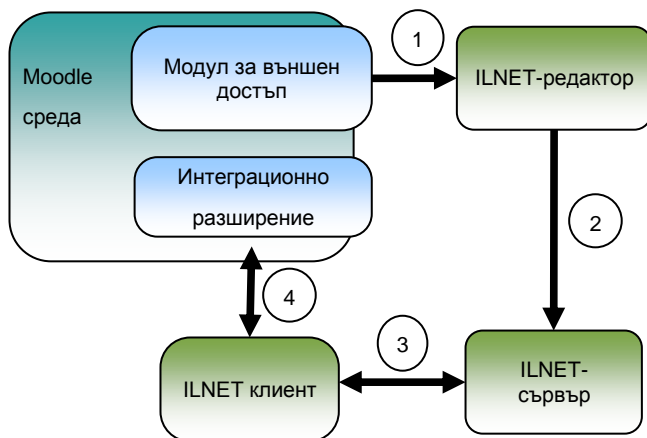
модела на процеса може да се включват и успешно управляват и новите дейности);

- възможно е мултиплициране на резултатите (избор на друга СеО като основа), и др.

Създаденият на тази основа бърз прототип на WMS за е-обучение, наречен *BEST 3.M* (за друг подход в реализацията – *BEST 2.0* вж. [17]) предоставя инструменти за моделиране на конкретни е-курсове под формата на сложни (в общия случай нелинейни и адаптивни) потокови структури от учебни дейности (типични за *Moodle*) и управление на съответното обучение за множество от студенти.

Основните компоненти на системата за моделиране и управление на виртуални курсове *BEST 3.M* са: *разширена Moodle-среда* (включително ILNET-клиент), *ILNET-редактор* и *ILNET-сървър*. Фиг. 5. дава обща представа за комуникацията между отделните компоненти на *BEST 3.M*.

В отделните етапи на разработката, модулите на *BEST 3.M* са тествани първоначално с контролни данни, а по-късно - и с реални.



Фигура 5. Общ изглед на архитектурата на BEST 3.M

Перспективно изследователско направление за развитие на системата е в посока на създаване на методи, софтуерни средства и дружелюбен потребителски интерфейс за моделиране, изучаване и управление на други аспекти от процеса на обучение (например автоматизирана оценка на качеството, методика на е-обучението, провеждане на адаптивни тестове, и др.)

ЗАКЛЮЧЕНИЕ

В дисертационния труд е направен преглед на типовете системи за семантичен анализ и информационно търсене. Разгледани са съвременните направления в разработката на системи за семантичен анализ и информационно търсене. Изводите от направения обзор са използвани за създаване на концепция на модел за описание на процеси за информационно търсене и семантичен анализ. Изведена е архитектура, удовлетворяваща поставените изисквания. Направена е реализация на архитектурата. Моделът е приложен успешно за разработка на система за извличане на контактна информация от документи на български език, методи за автоматизирано генериране на метаданни за учебни обекти и модул за моделиране и управление на виртуални адаптивни курсове, базирани на moodle.

Поставените в увода задачи са постигнати.

Основните приноси на дисертационния труд са:

1. Предложен е модел за описание на процеси за информационно търсене и семантичен анализ, сравним, в частност с мощността на мрежите на Петри;
2. Проектирана е архитектура на система за управление на работни процеси, базирана на описаният модел;
3. Реализиран е прототип на системата;
4. Системата е апробирана, като са описани различни процеси от областта на информационното търсене и семантичният анализ, и е използвана за извличане на контактна информация от документи на български език [1], автоматизирано генериране на метаданни за учебни обекти, моделиране и управление на виртуални адаптивни курсове и извличане на структурирани данни от уеб документи [6].

Връзките между приносите, целите, задачите, мястото на описание в дисертационния труд и направените публикации са описани в следната таблица:

Принос	Вид принос	Цел	Задачи	Параграф	Публикация
1	научно-приложен	2	2	(2.2), (2.3)	1,2,3,5
2	научно-приложен	3	3.1	(3.1)	6,7,9
3	приложен	3	3.2, 3.3	(3.2), (3.3)	
4	научно-приложен	4	4	(4.1), (4.2), (4.3)	4,7,8,9

Перспективи

Предложеният модел може да се изследва в следните насоки:

1. Сравняване на модела с други формализми за описание на процеси и/или структури в КЛ;
2. Описания на други процеси (от областта на информационната търсене, семантичният анализ и др.);

Софтуерната система може да се развива в следните насоки:

3. Имплементиране на изграждащи блокове и шаблони за изпълнение, които да позволят моделирането на известни стандарти за описание на процеси/потоци;
4. Интеграцията на системата в други практически приложения (персонализирано и адаптивно обучение [11, 12], е-бизнес [23], административен контрол [22], и др.);
5. Реализация на уеб-базиран клиент на системата;
6. Разширяване на архитектурата на сървърният компонент с възможност работа в клъстер.

Апробация

Резултати, получени от изследването, са използвани в следните национални проекти:

1. D002 308, Автоматизирано генериране на метаданни за спецификации и стандарти на Е-документи, 2008-2011
2. МИ-203, Моделиране на учебните процеси и управление на проекти за е-обучение, 2007-2010

Част от резултатите, получени в дисертационния труд са докладвани на следните национални и международни конференции:

1. 3-та научна конференция за студенти, докторанти и млади научни работници, 25.4.2009, Пловдив, България.
2. International Conference on Information Research and Applications (i.Tech 2009), 02.09.-05.09.2009, гр. Мадрид, Испания
3. 10th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing 2009, June 18-19, 2009, Rouse, Bulgaria.
4. Национална конференция "Образованието в информационното общество", АРИО, 26-27 май 2011, Пловдив, България

БЛАГОДАРНОСТИ

Бих искал да изразя голямата си признателност към моя научен ръководител проф. дмн. Георги Тотков за всестранныя помощ и проявените интереси и внимание към моята работа, за добрите напътствия и съвети. Благодаря и на колегите от ФМИ при ПУ „П. Хилендарски“ за тяхната подкрепа и насоки. Искам сърдечно да благодаря и на семейството си за обичта и голямата подкрепа.

НАУЧНИ ТРУДОВЕ НА АВТОРА, СВЪРЗАНИ С ДИСЕРТАЦИЯТА

1. Totkov G., D. Blagoev, *Regular Expressions Builder and Parser for Unicode Systems*, Information Technologies and Control, Year III, №1/2005, 40-45.
2. Тотков Г., Д. Благов, В. Ангелова, *За озвучаването на компютърен български текст*, Национална научна конференция „10 години катедра Компютърни системи към ТУ – София, филиал Пловдив“, ноември 2003, 119-131.
3. Blagoev D., G. Totkov, *Visual Parser Builder*, Proc. of the International Conference Recent Advances in Natural Language Processing RANLP'05, 21-23 September 2005, Borovetz, 112-116.
4. Благов Д., *Извличане на контактна информация от документи на български език*, в Сборник доклади на 3-та научна конференция за студенти, докторанти и млади научни работници, 25.4.2009, Пловдив, 339-343.
5. Blagoev D., G. Totkov, M. Staneva, Kr. Ivanova, Kr. Markov, *Indirect Spatial Data Extraction from Web Documents*, International Book Series Information Science & Computing, New Trends in Intelligent Technology, Number 14, ITNEA, 2009, 89-100.
6. Indzhov Hr., D. Blagoev, G. Totkov, *Executable Petri Nets: Towards Modelling and Management of e-Learning Processes*, ACM International Conference Proceeding Series; Vol. xxx, Proc. of the 10th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing 2009, Rousse, Bulgaria, June 18-19, 2009. IIIA.12.1 IIIA.12.6.
7. Благов Д., Хр. Инджов, Г. Тотков, *Моделиране и управление на виртуални адаптивни курсове, базирани на Moodle*, Сборник материали на конференция на НБУ, В. Търново, 30.9.2010 г. (в печат).
8. Христина Костадинова, Георги Тотков, Димитър Благов, *Автоматизирано генериране на метаданни за учебни обекти*, Сборник доклади на Национална конференция "Образованието в информационното общество" Пловдив, АРИО, 26-27 май 2011, стр. 044-052
9. Г. Тотков и кол., *Е-обучението в информационното общество: технологии, модели, системи, достъпност и качество*, ISBN 978-954-423-651-9, българска, първо издание, Университетско издателство „Пасий Хилендарски“ – гр. Пловдив, 2010г.

Забелязани цитирания

Indzhov Hr., D. Blagoev, G. Totkov, *Executable Petri Nets: Towards Modelling and Management of e-Learning Processes*

1. Korečko Št., Br. Sobota; R. Janošo, *Evaluation of Parallel Raytracing Strategy Improvements by Petri Nets*, Journal of Computer Science and Control Systems, Vol. 3, No. 1, May 2010, 87-92.
2. A. Guerriero, C. Pasquale, F. Quaranta, F. Ragni, *Java Grid Workflow management with fuzzy clustering approach*, in proceeding of CIMSA 2010 - IEEE International Conference on Computational Intelligence for Measurement System and Applications Taranto, Italy, 6-8 September 2010.

Г. Тотков и кол., *Е-обучението в информационното общество: технологии, модели, системи, достъпност и качество*

3. Габриела Кирякова, Надежда Ангелова, Лина Йорданова, *Свободният софтуер в образованието*, Сборник на 4-та Нац. конференция „Образованието в информационното общество“, (ред. Г. Тотков и Ив. Койчев), 26-27 май 2011 г., Пловдив, Асоциация „Развитие на информационното общество“, София, ISSN 1314-0752, 96-103
4. Габриела Кирякова, Надежда Ангелова, Лина Йорданова, *Web технологии и инструменти за създаване и управление на съдържание*, Сборник на 4-та Нац. конференция „Образованието в информационното общество“ (ред. Г. Тотков и Ив. Койчев), 26-27 май 2011 г., Пловдив, Асоциация „Развитие на информационното общество“, София, ISSN 1314-0752, 157-166
5. Николай Касъкчиев, *Сравнителен анализ на информационни системи за висши училища*, Сборник на 4-та Нац. конференция „Образованието в информационното общество“ (ред. Г. Тотков и Ив. Койчев), 26-27 май 2011 г., Пловдив, Асоциация „Развитие на информационното общество“, София, ISSN 1314-0752, 139-148
6. Илияна Чакърова, *Към дистанционно обучение във филиала на ПУ „Паисий Хилендарски“ – град Смолян*, Сборник на 4-та Нац. конференция „Образованието в информационното общество“ (ред. Г. Тотков и Ив. Койчев), 26-27 май 2011 г., Пловдив, Асоциация „Развитие на информационното общество“, София, ISSN 1314-0752, 167-176

БИБЛИОГРАФИЯ

- [1] Благоев Д., *Извличане на контактна информация от документи на български език*, в Сборник доклади на 3-та научна конференция за студенти, докторанти и млади научни работници, 25.4.2009, Пловдив, 339-343.
- [2] Некрестьянов И., Пантелеева Н. *Системи текстового поиска для Веб // Программирование*. – 2002. – N4.
- [3] Alpaydin E. *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, October 2004.
- [4] Appelt D. E., D. J. Israel. *Introduction to Information Extraction Technology*, 1999. A tutorial prepared for IJCAI-99.
- [5] Bergman M. *The Deep Web: Surfacing Hidden Value*. Journal of Electronic Publishing, 7, Aug. 2001.
- [6] Blagoev D., G. Totkov, M. Staneva, Kr. Ivanova, Kr. Markov, *Indirect Spatial Data Extraction from Web Documents*, International Book Series Information Science & Computing, New Trends in Intelligent Technology, Number 14, ITHEA, 2009, 89-100.
- [7] Blagoev D., G. Totkov, Visual Parser Builder, RANLP'05, Borovetz, 112-116.

- [8] Cardie C. *Empirical Methods in Information Extraction*. AI Magazine, 18(4):65–80, 1997.
- [9] Cowie J., W. Lehnert. *Information Extraction*. Commun. ACM, 39(1):80–91, 1996.
- [10] D. Blagoev, G. Totkov. *Regular Expressions Builder and Parser for Unicode Systems*, Information Technologies and Control, Year III, №1/2005, 40-45.
- [11] D. Lamboudis, A. Economides, C. Papas. *Applying Work Flow Reference Model in Adaptive Learning*, In G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, 2005, pp. 856-859.
- [12] D. Sampson, C. Karagiannidis, C. Kinshuk. *Personalised Learning: Educational, Technological and Standardisation Perspective*, Interactive Educational Multimedia, number 4 (April 2002), pp. 24-39.
- [13] Eikvil L.. *Information Extraction from World Wide Web - A Survey*. Technical Report 945, Norwegian Computing Center, 1999.
- [14] *Final LOM Draft Standard*, <http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html>.
- [15] G. Totkov, D. Blagoev, V. Angelova. *SLOG 1.0: Bulgarian text to speech transformation system*, International Joint Conf. on Computer, Information, Systems Sciences and Engineering (CIS2E'05), Bridgeport (CT), Dec 10-20, 2005.
- [16] Greenberg, J., K. Spurgin, A. Crystal, *Functionalities for Automatic-Metadata Generation Applications: A Survey of Metadata Experts' Opinions*. International Journal of Metadata, Semantics, and Ontologies. 2006, Vol. 1, No. 1, 2006 3
<http://www.inderscience.com/storage/f121932106117458.pdf>.
- [17] Hr. Indzhov, D. Blagoev, G. Totkov. *Executable Petri Nets: Towards Modelling and Management of e-Learning Processes*, ACM International Conference Proceeding Series; Vol. 375, Proc. of the 10th Int. Conf. on Computer Systems and Technologies CompSysTech'09, Rouse, Bulgaria, June 18th - 19th, 2009, pp. IIIA.12.1 - IIIA.12.6.
- [18] Hr. Indzhov, R. Doneva, D. Blagoev, G. Totkov. *Modelling and Carrying on e-Learning Processes with a Workflow Management Engine* (this volume).
- [19] Indzhov Hr., D. Blagoev, G. Totkov, *Executable Petri Nets: Towards Modelling and Management of e-Learning Processes*, ACM International Conference Proceeding Series; Vol. 375, Proc. of the 10th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing 2009, Rouse, Bulgaria, June 18-19, 2009. IIIA.12.1 IIIA.12.6.
- [20] Jansen B. J., Spink A., Saracevic T. *Real life, real users, and real needs: a study and analysis of user queries on the web*. Information Processing and Management, 36(2):207-227, 2000.
- [21] Korečko S., B. Sobota, C. Szabó, Performance analysis of processes by automated simulation of Coloured Petri nets, 10th International Conference on Intelligent Systems Design and Applications (ISDA), 2010, Cairo, Nov. 29 2010-Dec. 1 2010, p176-181.
- [22] S. Williams. *Business Process Modelling Improves Administrative Control*, In: Automation, December, 1967, pp. 44 - 50.
- [23] T. Dufresne, J. Martin. *Process Modelling for E-Business*, INFS 770 Methods for Information Systems Engineering: Knowledge Management and E-Business, Spring 2003.
- [24] Totkov G., D. Blagoev, V. Angelova, SLOG 1.0: Bulgarian text to speech transformation system, International Joint Conf. on Computer, Information, Systems Sciences and Engineering (CIS2E'05), Bridgeport (CT), Dec 10-20, 2005.
- [25] Totkov G., E. Somova, M. Sokolova, *Modelling of e-Learning Processes: an Approach Used in Plovdiv e-University*, Int. Conf. on Computer Systems and Technologies (e-learning), CompSysTech'04, 17-18 June 2004, Rouse, IV.12.1 – IV.12.6.