



**БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И КОМУНИКАЦИОННИ
ТЕХНОЛОГИИ**

Каменка Атанасова Стайкова

**ЛИНГВИСТИЧНИ И СЕМАНТИЧНИ РЕСУРСИ
ПРИ КОМПЮТЪРНО ГЕНЕРИРАНЕ И АНОТИРАНЕ
НА БЪЛГАРСКИ ТЕКСТОВЕ**

ДИСЕРТАЦИЯ

за придобиване на образователната и научна степен „доктор“

по специалност 01.01.12 Информатика

професионално направление 4.6 Информатика и компютърни науки

Научен ръководител:
доцент д-р Данаил Дочев

София, 2015 г.

СЪДЪРЖАНИЕ

Списък на използвани съкращения и означения.....	3
Увод	4
Актуалност на темата	4
Обзор на основните резултати в областта	4
Цели и задачи на дисертацията.....	6
Методология на изследването	7
Структура на съдържанието.....	8
Глава 1. Компютърно генериране на текст, лингвистични и семантични ресурси...9	
1.1 Компютърно генериране на текст – обща постановка	9
1.2 Техники при КГТ	16
1.2.1 Техники за селектиране и интерпретиране на съдържание	16
1.2.2 Техники за планиране на текста	21
1.2.3 Техники за лексико-граматически реализации	23
1.3 Компютърно генериране на текст, семантични технологии и ресурси	30
Глава 2. Подготовка на лексико-граматически ресурс за компютърно генериране на български текстове	44
2.1 Моделиране на някои езикови явления в термините на Системично-функционалната граматика	44
2.1.1 Ниво изречение	47
2.1.2 Ниво сложно изречение.....	57
2.2 Лексико-граматически ресурс за генериране на български текстове: Приложна българска системично-функционална граматика	66
2.2.1 Принципи на създаване	66
2.2.2 Елементи от приложната граматика и продукции	69
Глава 3. Реализация на процеса на генериране на български текстове	89
3.1 Представяне на знания за генерирането на български текстове	89
3.2 Определяне на съдържанието	92
3.3 Планиране на текста	93
3.4 Лексико-граматическа реализация	98
Глава 4. Работа с български текстове, семантични технологии и ресурси	102
4.1 Формализиране на знания от областта на иконографията.....	102
4.2 Семантично аотиране на специализирани български текстове.....	110
Заклучение.....	121
Списък на публикациите по дисертацията	122
Апробация на резултатите	123
Основни научни и научно-приложни приноси	123
Декларация за оригиналност.....	124
Благодарности	124
Литература	125
Приложение 1: Модел на предметна област „CAD/CAM приложения“	130
Приложение 2: Формално представяне на съдържанието на процедурен текст ...	132
Приложение 3: Онтологии в OWL на предметна област „Иконография“	140
Приложение 4: Аотационни граматика	169
Приложение 5: Резултати от семантичното аотиране	177

Списък на използвани съкращения и означения

Термин на български език	Съкр. на бълг. език	Термин на английски език	Съкр. на англ. език
Извличане на знания		Knowledge Extraction	
Извличане на информация (от текст)		Information Extraction	IE
Извличане на информация на базата на онтологии		Ontology-Based Information Extraction	
Изкуствен интелект	ИИ	Artificial Intelligence	AI
Компютърна лингвистика		Computational Linguistics	
Компютърно генериране на текст	КГТ	Natural Language Generation	NLG
Контролиран естествен език		Controlled Natural Language	CNL
Концептуален референтен модел на CIDOC		CIDOC Conceptual Referent Model	CIDOC CRM
Обобщен модел		Generalized Upper Model	GUM
Обработване на естествен език	OEE	Natural Language Processing	NLP
Онтологичен език на Глобалната мрежа		Web Ontology Language	OWL
правене на изводи		reasoning, inferencing	
Представяне на знания	ПЗ	Knowledge Representation	KR
Протокол за трансфер на хипер-текст		Hyper-Text Transfer Protocol	HTTP
Протокол и език за RDF- запитвания		SPARQL Protocol and RDF Query Language	SPARQL
Разпознаване на наименовани понятия		Named Entities Recognition	NER
Разширен език за маркиране		Extensible Mark-up Language	XML
Свързване на отворени данни		Linking Open Data	LOD
Семантична мрежа		Semantic Web	
Семантично аотиране		Semantic Annotation	
Системично- функционална граматика	СФГ	Systemic-Functional Grammar	SFG
Среда за многоезиково генериране KPML		Multilingual text generation environment KPML	KPML
Теория на реторичната структура	TPC	Rhetorical Structure Theory	RST
Унифициран идентификатор на ресурс (в Глобалната мрежа)		Uniform Resource Identifier	URI

Увод

Актуалност на темата

Компютърното генериране на текст (Natural Language Generation) е научна дисциплина, чиято цел е продуцирането на разбираеми текстове на естествен език. Генерирането и аотирането на естествено-езикови текстове е особено актуално в съвременния свят, когато комуникацията е изключително бърза благодарение на Глобалната мрежа. От една страна съществува необходимост информация във вътрешно компютърно представяне да бъде поднесена в разбираем за хората вид, от друга страна, огромният обем достъпна текстова информация би била по-разбираема и полезна ако е систематизирана и подредена по някакви (семантични) критерии. В първия случай са приложими стратегиите на компютърното генериране на текст, във втория случай са полезни техниките за обработване на естествен език, като една от тях е семантичното аотиране.

В Глобалната мрежа са достъпни текстови представяния на всички съвременни езици и това прави обработването на естествен език научна област от изключителна важност и с необозрим потенциал. Предмет на настоящата дисертация са ресурсите за генериране и аотиране на български текстове. Научната общност занимаваща се с компютърни обработки на български език все още се нуждае от стабилни и пре-използваеми ресурси, отворени за допълване и развитие. В този смисъл лексико-граматическият ресурс за генериране на български текстове разглеждан в настоящата дисертация дълго ще бъде ценен и актуален с предлаганите възможности за по-нататъшно разширяване и развитие. Семантичните ресурси използвани за аотиране на български текстове също представят една от най-актуалните теми в обработването на естествено-езикови текстове, а именно- семантичните технологии. Освен, че са свързани с едни от най-интересните съвременни научни изследвания такива ресурси биха могли да имат и други приложения, тъй като по природа са пре-използваеми.

Обзор на основните резултати в областта

Компютърното генериране на текст се определя като „задача интензивно използваща знания” (a knowledge-intensive problem) тъй като изисква много ресурси и различни видове знания. За компютърно генериране на текст са необходими знания за предметния свят описван в текстовете, знания за естествения език, на който се генерира (лексика, граматика, семантика), стратегически реторически знания (как се постигат определени комуникационни цели, как се построяват различни видове текст, стил на текста) и т.н. От една страна за компютърното генериране на текст са важни солидните теоретични изследвания свързани с лингвистичната същност на изходния продукт- текстът на естествен език. От друга страна, своите аргументи има и едно по-технократско отношение към компютърното генериране на текст, което въвлича по специфичен начин някои от съвременните семантични технологии. Най-многобройни са създадените системи и приложения за компютърно генериране на текст на английски език. Все още продукции на друг естествен език са сериозно изследователско предизвикателство.

Абстрактно описани, задачите на компютърно генериране на текст имат две измерения на лингвистична спецификация. От една страна, информацията в бъдещия текст може да бъде по-конкретна или по-абстрактна (стратификация с нива: регистър/ стил на текста, семантика, лексико-граматика, графология/фонология). От друга страна, информацията може да кореспондира с различни смислови стойности или мета-функции (пропозиционална, интерперсонална или текстова).

Така се очертава представата за същността на генерирането на текст и се определят присъщите задачи при компютърно генериране:

- 1) селектиране и интерпретиране на съдържанието,
- 2) планиране на текста,
- 3) лексико-граматическа реализация на текста.

Техниките за селектиране на съдържанието се свързват с вземането на решение коя част от представената на входа информация да се включи в подготвяния текст и коя- да се пропусне. Класическият начин за определяне на съдържанието е въведен със схемите на текста [McKeown, 1985] и играе важна роля при структурирането на текстове в много от системите за КГТ. Някои от съвременните приложения за КГТ ползват съвсем различни техники за селектиране на съдържание, когато са свързани с т.н. «отворено планиране на текста». Тези техники зависят пряко от специфичното представяне на знанията на входа, а именно представяне в RDF-графи.

Техниките за интерпретиране на съдържанието варират от създаването на таблици съпоставящи понятията от приложната област и лингвистичните ресурси на генератора, до различни по сложност онтологични модели. Ролята на онтологиите тук е да свързват понятията от предметната област на текста с лингвистичната природа на езиковите единици, чрез които се изразяват тези понятия в текста. В [Bouayad-Agha et.al., 2013] се твърди, че една от най-обещаващите такива онтологии е Обобщеният модел (Generalized Upper Model), чиято еволюция продължава повече от 20 години [Bateman et.al, 1990], [Bateman et.al., 2010].

Класически техники за планиране на текста са шаблонните структури, които могат да бъдат използвани за текстове със стереотипни конструкции. Сравнително гъвкав тип текстови шаблони са предложени от [McKeown, 1985] и представени в термините на мрежи на преходите (transition networks). Този подход е една от най-разпространените техники за организиране на текст въпреки явните си ограничения. Създаване на текстови конструкции с по-голяма гъвкавост е възможно чрез прилагане на Теорията за реторичната структура (TPC) [Mann and Thompson, 1988], която предлага общо описание на релациите съществуващи между текстови сегменти като показва дали тези релации са граматически или лексикално сигнализирани.

По отношение на лексико-граматическата реализация на текста съществуващите техники прилагани в КГТ могат да бъдат класифицирани в скала, като в единия ѝ край са множества от шаблони, а в другия- реализации с граматики.

Структурните шаблони могат да бъдат фиксирани или параметризирани, и се задействат от специфични комуникационни цели или семантични спецификации.

Граматиките имат за цел да съпоставят на всеки добре форматиран семантичен вход кореспондиращ му низ или последователност от низове представящи генерирания изходен текст. Подходите за тази задача съдържат огромна част от работата по КГТ. Най- известните формализми и съответстващи им граматически ресурси за компютърно генериране на текст са: Опорна фразова граматика [Pollard and Sag, 1994] и семантичното генериране управлявано от опорния елемент (semantic head-driven generation) [Shieber et.al, 1990], Модел на смисъла на текста разработен от Мелчук и повърхностния генератор RealPro [Lavoie and Rambow, 1997], формализмът за функционално унифициране (Functional Unification Formalism, FUF) [Elhadad, 1990] и граматиката за английски език SURGE (Systemic Unification Realization Grammar for English), генериране направлявано от съобщението (message-directed processing) [McDonald, 1983] и лексико-граматическия ресурс MUMBLE, генериране с контрол направляван от граматиката, предложен за системата Penman [Mann, 1983] и граматиките в Средата за многоезиково генериране KPML [Bateman, 1997].

Парадигмата на Системично-функционалната лингвистика е една от основните най-успешни постановки за компютърно генериране на текст. Като теоретичен подход тя предлага

изключителни възможности за паралели при системично-функционалните описания на различни естествени езици. В приложния си аспект такива паралели позволяват по-бързо и по-ефективно формализиране на лингвистичните знания при създаване на система за компютърно генериране на текст за новоразглеждан език.

Специфичното при системично- функционалния възглед за КГТ е организирането на целия процес на генериране около комуникационните цели, а не около граматическите структури на конкретния естествен език. На това се базира идеята за изграждане пространство на търсене от системични мрежи за генериране, което представлява приложен ресурс на системично-функционалната граматика за даден език. За реализиране на такава граматиката е необходимо моделиране на (някои) лингвистични феномени на естествения език по отношение на Системично-функционалната лингвистика.

Интересна характеристика на компютърното генериране е фактът, че от изходните текстове не може да се оцени какви технологии са използвани в процеса на генериране - елементарни текстови обработки или дълбочинни методи с богати ресурси. Затова е важен обхватът езикови феномени, с които работи генераторът на естествено-езиков текст. Този обхват дава представа за богатството на възможните вариации на изходните текстове.

Новите тенденции в компютърно генериране на текст са свързани с нарастващото значение на семантичните технологии в компютърните обработки. Естествено за компютърното генериране на текст е използването на онтологии за предметната област на генерирането, а също и обобщаващи онтологични модели от по-висок ред за лингвистичните знания. Следователно, работата с онтологии е до голяма степен свързана и полезна за компютърното генериране на текст. Семантичното аотиране на текстове на естествен език има приложение в областта на генерирането като практически значима подпомагаща дейност. Семантичното аотиране представлява интерес особено за естествени езици различни от английски поради по-малкия брой изследвания и работещи реализации.

Цели и задачи на дисертацията

Основна цел на настоящата дисертация са изследвания и експерименти за реализация на компютърно генериране на текстове на български език чрез създаване на лингвистични ресурси за основни обекти на българския език и на семантични ресурси за конкретни предметни области.

Изследователските задачи на дисертационния труд са конкретизирани на базата на анализа на съвременните подходи, модели и техники за формиране на лингвистични и семантични ресурси, отчитащи проблемите на съвременното генериране на текстове, отразен в обзорната глава. Те са съобразени и с нуждите на изследователската среда, в която е работил авторът.

Задачите на дисертацията са следните:

1. Създаване на формални описания на основни обекти на българския език в рамките на Системично- функционалната теория на Халидей с цел компютърно генериране на технически текстове.
2. Реализация на създадените формални описания във вид на компютърен ресурс в Средата за многоезиково генериране KPML.
3. Аprobация на разработения ресурс за компютърно генериране на кохерентни технически текстове на български език в избрана предметна област.
4. Разработване и реализация на схема за аотиране на специализирани текстове на български език на базата на съвременни семантични технологии.

В дисертацията са използвани съществено научните изследвания извършени с активното участие на автора по проектите AGILE и СИНУС.

AGILE: „Automatic Generation of Instructions in Languages of Eastern Europe” (Автоматично генериране на инструкции на три източно- европейски езика) е изследователски проект финансиран от Европейската комисия и реализиран по програмата *INCO-Copernicus* през 1998-2001г. от партниращи си организации от пет държави: Великобритания, Германия, Чехия, България и Русия.

СИНУС: „Семантични технологии за Интернет-услуги и технологично поддържано обучение” е изследователски проект № Д-002-189 финансиран от Националния фонд „Научни изследвания” през 2009-2012г. Резултатите от проекта са достъпни на адрес: sinus.iinf.bas.bg.

Методология на изследването

Компютърното генериране на текст (Natural Language Generation) е под-област на Обработването на естествен език (Natural Language Processing). Научните изследвания и разработки в компютърното генериране на текст са съсредоточени върху създаването на компютърни системи продуциращи разбираем текст на естествен език. Започвайки обикновено от някакво семантично представяне на информацията на входа, системите генериращи естествен език използват знания за езика и знания за приложната област, за да създадат и оформят документи, отчети, рапорти, обяснения, помощни съобщения или други видове естествено-езикови текстове.

В настоящата дисертация са представени научно-приложни изследвания свързани с процеса на компютърно генериране на български текстове. От гледна точка на информатиката основни за компютърното генериране на текст са формализацията и обработката на лингвистични знания или лексико-граматиката. В настоящата дисертация някои лингвистични феномени на българския език са обект на анализ и моделиране в парадигмата на Системично-функционалната граматика на [Halliday, 1994]. Показано е изграждането на проложен лексико-граматически ресурс за генериране на български език. Достъпът до Средата за многоезиково генериране KPML подпомага съществено работата по създаване, тестване и настройка на приложна системично-функционална граматика за генериране на български език.

Системично- функционалните граматики създадени на базата на теорията на [Halliday, 1994] кодират семантичните връзки в текста във функционална форма и са насочени към директни съответствия между по-високите нива на организация на текста и граматическия компонент. Това е предпоставка за научно-приложна изследователска работа по генериране на различни типове/ стилове на изходни текстове от едно и също входно представяне.

С развитието на семантичните технологии в последните години се наблюдават интензивни научно-приложни изследвания свързани с подпомагане компютърното генериране на текст, например за по-удобно представяне на знания в онтологични конструкции или по-ефективни семантични обработки на естествено-езикови текстове. Изследователско направление в настоящата дисертация е семантичното аотиране на специализирани текстове на български език чрез прилагане на подхода „Релация: От онтология към текст“. Основна идея на метода е специално разработени ресурси (терминологичен лексикон, анотационна граматика) да се използват в комбинация при разпознаването на срещания на онтологичните понятия от дадена онтология в текст. Изследването е реализирано с частични граматики на базата на регулярни изрази разработени в системата CLaRK.

Структура на съдържанието

Първата глава на дисертацията представлява обзор, който представя по същество компютърното генериране на текст (КГТ) като научно-приложна област и описва лингвистичните и семантичните ресурси, които се използват в процеса на генериране. Първите два раздела 1.1 и 1.2 се занимават с постановката на задачата за КГТ и със съвременните техники и методологии използвани в процеса на КГТ. В раздел 1.3 са разгледани накратко семантичните технологии характерни за областта Обработка на естествен език, тъй като КГТ е нейна под-област. Коментирани са също съвременните идеи и перспективи на компютърното генериране на текст от данни на Семантичната мрежа.

Втора глава представя работата по създаването на ресурс за генериране на български изречения. В раздел 2.1 са показани теоретични модели за някои граматически явления в българския език на базата на Системично-функционалната граматика на Халидей. В раздел 2.2 са описани принципите за създаване на приложна българска системично-функционална граматика. Показани са резултати от автоматичното генериране на български изречения със създадената приложна българска системично-функционална граматика.

Трета глава демонстрира използването на приложната граматика като лексико-граматически ресурс за генериране на български език в конкретна приложна област. Описан е процесът на планиране на текст. Показани са резултати от автоматично генериране на кохерентни текстове на български език в различни стилове.

В четвърта глава се обръща внимание на семантичните технологии като представяне на знания и семантично аотиране, които биха подпомогнали подготвителните работи при компютърно генериране на текст. Използвана е конкретна приложна област за изследване на семантично аотиране с частични граматика на базата на регулярни изрази.

Глава 1. Компютърно генериране на текст, лингвистични и семантични ресурси

1.1 Компютърно генериране на текст – обща постановка

Компютърното генериране на текст (Natural Language Generation) е под-област на компютърната обработка на естествен език (Natural Language Processing). Това е дисциплина, в която научните изследвания са съсредоточени върху създаването на компютърни системи, чиито продукт са разбираеми текстове на естествен език. За да разграничим Генериране на естествен език от научната област Синтезиране на говор, в тази дисертация ще използваме превода Компютърно генериране на текст (КГТ).

Дефиниране на задачата

Най-общо основната задача на компютърното генериране на текст може да се формулира като превръщане на някакъв вид не-лингвистична информация чрез работата на компютърна система в писмен текст на естествен език, подходящ за възприемане от човека. Робърт Дейл¹ дава следната дефиниция²:

“Компютърното генериране на текст е процес на целенасочено построяване на текст на естествен език, за да бъдат постигнати определени комуникационни цели.”

КГТ има приложен аспект с определена практическа стойност. Тъй като компютърните системи използват вътрешни представяния на информацията (бази-данни, счетоводна информация, бази-знания, данни от тестове за работата на машини и съоръжения и т.н.), то съществува необходимост от компютърни програми, които да представят такава информация или обобщения на най-важните ѝ характеристики в разбираем вид за не-специалисти в конкретната област. Технологиията за КГТ се използва за представяне на данните в текстов вид в подходящ и удобен за човека формат. Дефинирането на конкретната задача насочва прилагането на технологията за КГТ или към автоматично генериране без участието на човека или към полуавтоматично генериране с участие на потребителя.

От научно-изследователска гледна точка КГТ се определя като „задача интензивно използваща знания“ (a knowledge-intensive problem) тъй като изисква много ресурси и различни видове знания. За КГТ са необходими знания за предметния свят описван в текстовете, знания за естествения език (лексика, граматика, семантика на съответния естествен език), стратегически реторически знания (как се постигат определени комуникационни цели, как се построяват различни видове текст, стил на текста) и т.н.

Понякога „компютърно генериране на текст“ се наричат и някои прости процеси с примерни стъпки: (1) задава се фиксиран текстов шаблон; (2) запазва се списък с предварително подготвени текстови отрязъци; (3) текстовете се копират и поставят на нужните места в шаблона; (4) ако е необходимо, отделни пасажии се свързват със „слепващ“ текст, оформя се изходният текст. Резултатите от такова „генериране“ са задоволителни за някои прости приложения като машини за хороскопи или генератори на персонализирани бизнес писма. Смята се, обаче, че една система за КГТ трябва да притежава най-малко нива за планиране и сливане на информация, за да успее да генерира текст, който изглежда естествен и без повторения [Bateman, 2002].

Можем да добием представа за съвременното ниво на компютърното генериране на текст като разгледаме продукцията на някоя достъпна система за КГТ на английски език. Групата на

¹ Робърт Дейл е професор в Macquarie University, Австралия: <http://web.science.mq.edu.au/~rdale/>

² <http://web.science.mq.edu.au/~rdale/teaching/esslli/part01.pdf>

изследователите със специален интерес към КГТ организирана от Асоциацията за компютърна лингвистика³ (ACL) поддържа списък на достъпните системи за КГТ⁴, повечето от които генерират английски текстове. Нека вземем системата наречена „Прогноза за полените в Шотландия” създадена в Университета на Абърдийн [Turner et.al., 2006]. Входни данни за нея са шест числа, които представляват предвижданите стойности за нивата на полените в различните части на Шотландия. От тези числа системата генерира като изход кратко текстово описание на очакваните поленови нива. Например, ако се използват данните от 1ви юли 2005г. (6,6,7,6,4,4) системата генерира следния текст:

Grass pollen levels for Friday have increased from the moderate to high levels of yesterday with values of around 6 to 7 across most parts of the country. However, in Northern areas, pollen levels will be moderate with values of 4.

Нивата на тревните полени за петък са покачени от средните до високи вчерашни нива, със стойности около 6 до 7 за повечето части на страната. В северните райони, обаче, нивата на полените ще бъдат средни, със стойности 4.

За сравнение авторите на системата за КГТ предлагат текст съставен от специалист- метеоролог от същите данни:

Pollen counts are expected to remain high at level 6 over most of Scotland, and even level 7 in the south east. The only relief is in the Northern Isles and far northeast of mainland Scotland with medium levels of pollen count.

Поленовите стойности се очаква да останат високи на ниво 6 в по-голямата част от Шотландия, и дори на ниво 7 на югоизток. Благоприятна промяна има единствено за Северните острови и далечния североизток на Шотландия със средни нива на поленова стойност.

Сравнението между двата текста илюстрира сложността на изборите, които трябва да бъдат направени от системата за КГТ на различни етапи от нейната работа. Тези етапи са дефинирани в [Reiter and Dale, 2000].

Определяне на съдържанието	Да се прецени дали изрично да се спомене цифрата 7 като ниво на полените на югоизток: „ <i>and even level 7 in the south east/ и дори на ниво 7 на югоизток</i> ” е изрично споменато в текста на експерта.
Структуриране на документа	Да се прецени дали първо ще се изредят областите с високо ниво на полените или обратно- първо ще се говори за областите с ниски нива на полените.
Агрегация	Системата е извършила сливане на подобни изречения, за да се подобри естествения изказ и четивността. Така се е получило изречението: <i>Grass pollen levels for Friday have increased from the moderate to high levels of yesterday with values of around 6 to 7 across most parts of the country./ Нивата на тревните полени за петък са покачени от средните до високи вчерашни нива, със стойности около 6 до 7 за повечето части на страната.</i>
Лексикализация	При описване на средното ниво на полени 4, трябва да се направи избор между думите <i>medium</i> и <i>moderate</i> .
Генериране на референтни изрази	В текста на експерта са използвани фразите <i>the Northern Isles/ Северните острови</i> и <i>far northeast of mainland Scotland/ далечния</i>

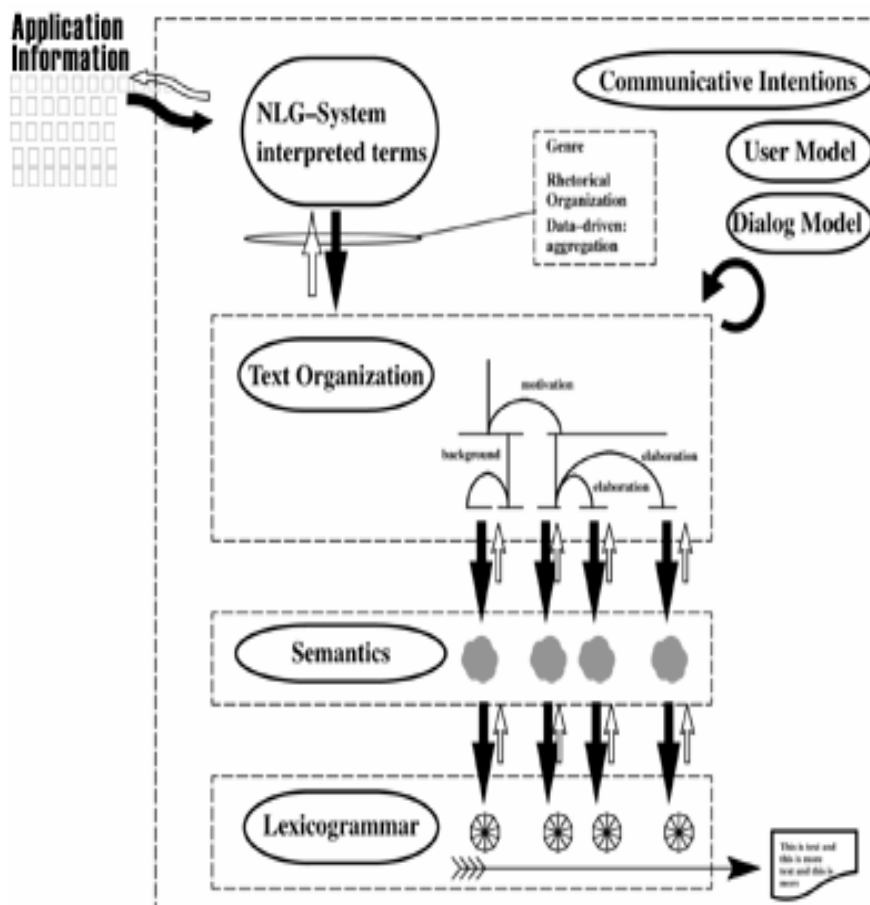
³ <http://www.siggen.org/>

⁴ http://aclweb.org/aclwiki/index.php?title=Downloadable_NLG_systems

	североизток на Шотландия за рефериране на конкретна област в Шотландия: трудно за реализиране от системата.
Повърхностна реализация	Избори за използването на нужните граматически форми за съответните части на речта, например <i>will be</i> за бъдеще време на <i>to be</i> .

Модел на процеса за КГТ

Предложената от Джон Бейтман Абстрактна система за КГТ⁵ представена на Фигура 1, е резултат от анализ на голям брой приложни системи за КГТ от перспективата на обхвата езикови феномени, с които системите се занимават.



Фигура 1: Абстрактна система за КГТ, [Bateman, 2002]

При КГТ не е установен общ, външно мотивиран източник на генерирането. Съществува консенсус, че „входът“ трябва да бъде *някакво семантично представяне*. За различните системи решенията за „вход“ са съвсем различни. Входните данни могат да бъдат, например, минимално структурирани числови данни, статистически рапорти, съдържанието на бази-данни. Разнообразни семантични представяния са били изпробвани през годините – логики от първи ред и свързаните с тях формализми [Basile and Bos, 2011], сценарии на Шанк (Schank's scripts) [Novy, 1988], концептуални графи на Сова [Nogier and Zock, 1992], различни фреймови представяния, а напоследък и RDF-представянията типични за Семантичната мрежа.

С изхода на процеса на генериране ситуацията е сходна. Генерираните текстове са разнообразни: с голяма дължина или кратки, самостоятелни параграфи или единични изречения

⁵ <http://www.fb10.uni-bremen.de/anglistik/langpro/webpace/jb/info-pages/nlg/ATG01/ATG01.html>

(напр. отговорите на запитвания към бази-данни), могат да са низове или да е приложено по-сложно текст-форматиране (пунктуация, оформяне на страници), могат да имат линейна структура, да бъдат част от диалог или да бъдат организирани като хипертекст. Текстовете могат да бъдат насочени към аудитории различаващи се по опит, знания, интереси или по когнитивно натоварване; може да се изисква изход на различни естествени езици и т.н. Систематично представяне на цялостната картина за входа и изхода на системите за генериране на текст все още не е предложено.

Техниките, които се използват за генериране на текст могат да бъдат съпоставени и оценени спрямо изискваното езиково многообразие в изходните текстове. Колкото повече гъвкавост на текстовете е търсена, толкова по-обща лингвистични знания са необходими и е по-голяма мотивацията за обръщане към характерните техники на КГТ. Разглеждането на приложните системи от перспективата на обхвата обработвани езикови феномени позволява вариациите на изходните текстове да бъдат поставени в общ план, като се абстрахираме от реализационното ниво на системите.

За по-точно описание на езиковите вариации при КГТ се използва лингвистичното понятие за стратификация (разслоеност). Стратификацията се отнася до различните видове информация на различните нива на абстракция, които допринасят за организацията на текста. Можем да говорим за по-ниски нива на езикова абстракция, като фонетика, морфология, синтаксис (общо лексико-граматика), за ниво на смисъла или семантиката на текста, и за по-високите нива на абстракция, като структура и стил на текста (виж Фигура 1: Абстрактна система за КГТ). Текстовете могат да се отличават спрямо различни текстови структури, различни синтактични структури и т.н. Независимо от действителните входни данни на системата за КГТ и контекста на приложение, задачите в процеса на генериране могат да се идентифицират по отношение на нивото на лингвистична абстракция. Задачата принадлежи на това ниво, в което е била предложена вариацията.

За да се обясни как текстът „пренася” нужната информация са необходими най-малко двете нива: лексико-граматика и семантично ниво. Лексико-граматиката дава възможните граматически структури на езика и свързаните с тях думи, които могат (или не могат) да се срещат в структурите. Това е ограничаваща организация, която трябва да се спазва във всеки текст, защото изреченията не са произволни комбинации от думи. Семантиката ограничава избора измежду възможните структури и думи само до тези, които са необходими за конкретния текст. Текстовете не са произволни комбинации от изречения.

В лингвистиката са известни различни теоретични подходи към граматиката и голяма част от тях имат приложни компютърни еквиваленти, докато в областта на семантиката се работи интензивно едва в последните десетилетия. Подходящ за КГТ е подходът на *семантичен контрол*, от който има нужда една лексико-граматика. За генерирането е необходимо не само да се определят възможните структури в езика, но и да се достигне до „точките на решение”: кои от възможните структури да бъдат избрани при конкретна комуникационна цел. Системите за КГТ винаги ще се различават по формата на входа си, но конкретните представяния на лингвистичната семантика дават добра основа за разглеждане на алтернативите независимо от приложението. Съществуват четири групи възможни вариации и избори при съставяне на изходния текст: определяне на съдържанието, интерперсонални вариации, семантика на текста, съгласуваност на дискурсна семантика.

Определяне на съдържанието е най-естественият източник на възможни избори. Това е пропозиционалното съдържание, нарича се и съдържание на твърдението или традиционно „Кой направи какво кому кога и как?”. Определянето на съдържанието ръководи лексико-граматическия подбор на основните единици, напр. изречение или група на съществителното

или наречна група. То служи и за подбор на основните граматически структури използвани в рамките на съответните категории, напр. преходна/ непреходна структура на изречението.

Ако целта е да се генерира, например, изречението *Потребителят чертае линия.*, то *чертая'* (*потребител'*, *линия'*).

е примерна логическа форма, която представя съдържанието на изречението. В логическата форма участват три понятия, които системата за КГТ трябва да има в своите регистри. Подадена като вход на лексико-граматиката логическата форма трябва да доведе след поредица от избори до конструиране на транзитивна граматическа структура изразяваща желаното съдържание.

Важен е фактът, че само подадената информация за пропозиционалното съдържание никога не може да ограничи в достатъчна степен възможностите за изразяване, които притежава един естествен език. Необходимо е да се вземат предвид и други “видове семантика”, за да се съберат достатъчно ограничения за ръководене изборите на лексико-граматиката. Чрез лексико-граматическите избори се селектират конкретните повърхностни низове за конкретния контекст на употреба и комуникационно намерение.

Отличителна черта на КГТ е необходимостта да се има предвид по-широк спектър от смислови видове отколкото традиционно разглежданите в Обработването на естествен език. Една основна разлика между анализ и генериране на естествен език е фактът, че докато семантичната логическа форма представлява изречение обикновено е основната цел на анализа, то в КГТ тази форма заема позиция сравнително близо до повърхностната реализация на текста. Повечето трудности и изследователски задачи в КГТ са свързани с генерирането на подходящи семантични логически форми на базата на комуникационните цели на съответната система за КГТ.

Интерперсоналните избори или отношението на говорещия към изразяваното съдържание влияе пряко върху смисъла на генерирания текст. За конкретно изречение, напр. *Потребителят чертае линия.*, интерперсоналните избори са необходимият минимум, за да се мотивира контролът на вариации от вида:

(а) *Потребителят чертае ли линия?*

(б) *Потребителят чертае линия, нали?*

(в) *Потребителят наистина чертае линия.*

Изречение (а) променя речевата функция от изказване във въпрос; изречение (б) променя речевата функция, за да се поиска потвърждение от слушателя и изречение (в) въвежда повече съдържание и новата информация се отразява на силата на речевия акт.

Интерперсоналните вариации имат решаващо влияние върху някои аспекти на текста. Изразяването на лично отношение на говорещия може да повлияе лексикалния избор: "художник" или "шарлатанин", "произведение на изкуството" или "боклук". Когато системите за КГТ използват големи лексикони и имат достъп до различни варианти при лексикализацията, трябва да съществува сигурност, че поне не са направени неподходящи избори. Силата на интерперсоналното послание е от особено значение за правилната вербализация на искания, заповеди, молби и това има значителни последици за граматическите селекции.

Решения за семантиката на текста са необходими, защото диапазонът на текстовата семантика предоставя много голям обем от лексико-граматически вариации за контролиране в процеса на КГТ. Почти всички примери на текстове действително генерирани от съвременните системи за КГТ демонстрират несъвършенства свързани с изразяването на текстовата семантика. Това означава, че има места, където човек съставящ текста би направил селекции, които водят до по-естествен и разбираем текст, както в примера за полените по-горе.

Основната задача тук е да се изолират и опишат източниците на ограничения за лексико-граматическите решения, които повлияват текстовата семантика и да се контролира лексико-граматиката. Колкото по-богати са представянията на лексико-граматически ресурси, толкова по-сложна става задачата. Ако лексико-граматиката предлага само няколко възможности, то нуждата от контрол е по-малка, но свободното изразяване или *флуентността на текста* са съответно намалени. Затова много от съвременните подходи към КГТ се концентрират върху намиране на подходящи контролиращи ограничения за общите ресурси вместо да разчитат на изкуствените вътрешни ограничения на ресурсите сами по себе си.

Съгласуваността на дискурсната семантика също изисква подходящ контрол на лексико-граматиката и много допълнителни ограничения за нивото на пропозиционалното съдържание. Предоставянето на такава информация прави възможно широко покритие на лексико-граматиката, за да се продуцират текстово подходящи лексико-граматически структури. Много изследователи от областта на КГТ смятат задачата за съгласуване на дискурсната семантика за основна в КГТ, защото тя гарантира, че последователности от подходящи спецификации на ограничения се продуцират по такъв начин, че спецификациите са взаимно съгласувани и събрани заедно те съставляват кохерентен текст.

В КГТ не е достатъчно, че като цяло системата за КГТ предлага средства за контролиране на лексико-граматиките, за да се продуцират синтактично валидни конструкции. Системата за КГТ трябва също да може да селектира от теоретично достъпните възможности тези, които са най-подходящи за конкретния текст. Въпреки първоначалния оптимизъм в опитите да се структурират текстове по вътрешно присъща за тях организация на изразяваното знание, това се оказва сковаващо и пречи на ефективното представяне при промяна на комуникационните цели. По-гъвкави методи на представяне в този аспект са крайно необходими за съвременното КГТ.

От лингвистична гледна точка всеки текст демонстрира комбинация от различни *нагласи* или *пре-диспозиции* (predispositions) на говорещия, които непрекъснато влияят както на избора каква информация ще бъде избрана и изразена, така и на формата на самото изразяване. Тази съгласуваност в селектираните семантични избори е общ въпрос за всички нюанси на смисъла представени в горните три задачи.

Пропозиционалните, интерперсоналните и текстовите селекции демонстрират едно специфично единство. Един начин да се определи тази съгласуваност по-прецизно в теоретични термини е предложен от Якобс през 1987г. в Теорията на регистрите. Теорията е изградена върху сходството между такива пре-диспозиции и системично-функционалното понятие регистър (*register*)⁶, което е използвано за структуриране обхвата на пре-диспозициите отговарящи на пропозиционалните, интерперсоналните и текстовите селекции.

По същество, Теорията за регистрите твърди, че езикът показва систематични вариации в зависимост от ситуациите на употреба и, следователно, съгласуваността със ситуацията изисква съгласуваност при подбора на семантични вариации. Това дава също и теоретична интерпретация на тези системи за КГТ, които не поддържат вариантност на пре-диспозициите за своите текстове: те са системи с един регистър и създават текстове подходящи за използване в една конкретна ситуация на употреба. Регистърът може да се смята за един следващ, по-абстрактен слой на лингвистичната система. Пример за това е настройването на текстовете според експертното ниво на потребителите. Опит да се реализира моделиране на регистъра в процеса на генериране е описан в [Bateman and Paris, 1989], като по-късно такъв опит се демонстрира от системата AGILE, чиито продукт са процедурни текстове-инструкции на три езика: български, руски и чешки [Kruijff et.al., 2000].

⁶ Понятието *регистър* е въведено от М. А. К. Халидей в *Language as Social Semiotic: The Social Interpretation of Language and Meaning*, Edward Arnold, London, 1978.

Много изследователи свързват текстовата семантика с понятията за структуриране на дискурса въведени от Теорията на реторичната структура (Rhetorical Structure Theory, RST) [Mann and Thompson, 1988], която е разгледана по-подробно нататък в *Раздел 1.2.2*.

Абстрактно описани задачите на КГТ имат две измерения на лингвистична спецификация: разслояване/ стратификация (stratification) и мета-функция (metafunction). Тези две измерения са извлечени от функционалния семиотичен подход на Системично- функционалната лингвистика: информацията може да бъде по-малко абстрактна или много абстрактна (стратификация с нива: регистър, семантика, лексико-граматика, графология/фонология), а също така информацията може да се отнася за различни смислови видове (мета-функции: пропозиционална, интерперсонална или текстова). Така се очертава представата за същността на генерирането на текст и се определят присъщите му задачи:

- 1) селектиране и интерпретиране на съдържанието,
- 2) планиране на текста,
- 3) лексико-граматическа реализация на текста.

За реализиране на КГТ трябва да бъдат направени съвместими избори пропозиционално, интерперсонална и текстово, за да се конструират последователности от семантични спецификации, които поддържат текстовата семантика и я изразяват в граматически конструктори. Без значение как е реализирана една система за КГТ, тези абстрактни задачи са съществени за генерирането. Ако някаква част от поставените от задачите проблеми не са решени, то гъвкавостта на системата за КГТ е ограничена по отношение на текстовото разнообразие.

На Абстрактната система за КГТ (виж Фигура 1 по-горе) можем да погледнем и като на приложен модел на процеса на КГТ. Входът за процеса на КГТ е наречен *информация за приложението* и е отбелязан в горния ляв ъгъл на изображението на Абстрактната генерираща система. Можем да приемем текст-планировчика и лексико-граматиката като отделни модули. След селектиране на съдържанието текст-планировчикът разполага със целево ориентирани техники конструирани последователности от не-строги семантични спецификации с размерите на изречение. Лексико-граматиката преработва такива сравнително абстрактни спецификации в не толкова абстрактни граматически конфигурации и свързан с тях лексически материал. На изхода имаме последователност от граматически спецификации превърната в текст съставен от символите на дадения естествен език.

Взаимодействието между компонентите е показано със стрелки в двете посоки, въпреки че в много приложни архитектури не е поддържана такава двупосочност. Много разлики между подходите за КГТ се дължат на гъвкавостта, която те придават на съпоставянето на представянията във вътрешните нива. Отделни листа на текстовата структура в някои системи се съпоставят на отделна семантична спецификация, а другаде- на последователност от семантични спецификации. По същия начин, отделни семантични спецификации могат да съответстват на последователности от лексико-граматически спецификации, но на практика най-често са реализирани прости връзки и съответствия.

За целите на настоящата дисертация моделът на Абстрактната генерираща система показан на Фигура 1 е използван като основа за изследване на различните нива на лингвистична абстракция на текстове- инструкции на български език. Основната мотивация на изследването е на практика да бъде реализирано генериране на текстове на български език, като се създадат нужните за това ресурси. Настоящият обзор е насочен към преглед на прилаганите техники за решаване на задачите на КГТ поставени от лингвистичната природа на изходния продукт – текстът като такъв.

1.2 Техники при КГТ

Основно изискване към системите за КГТ е да конструират различни видове структури съответстващи на различните нива на лингвистична абстракция. Типичните техники обикновено се различават в зависимост от естеството на използваните структури и реализацията избрана за приложната програма. Най-често лексико-граматическите структури са третираны по доста различен начин от текстовите структури или от семантичните структури, но са известни и подходи, при които е адаптирана една и съща лингвистична методика за всички представяния, например в [Bateman and Teich, 1995]. При по-голямата част от системите съществува ясно разграничаване между техниките за лексико-граматиката, за семантичната информация и за организацията на текста. Това обикновено се проявява чрез разликите в представянията, в обработките и тяхната последователност.

В настоящия раздел са разгледани конкретните техники използвани на всяко от нивата на лингвистична абстракция: определяне и интерпретация на съдържанието, планиране на текста, лексико-граматически реализации. Накрая се обръща внимание на техниките осигуряващи взаимодействие между модулите от различни нива на абстракция.

1.2.1 Техники за селектиране и интерпретиране на съдържание

Селектиране на съдържание

Обикновено системите за КГТ разполагат с голям обем входна за генерирането информация. Селектиране на съдържанието е вземането на решение коя част от информацията да се включи в подготвяния текст и коя- да се пропусне. Това е сложна задача зависеща от много фактори като представата какво желае да знае целевата аудитория или степента на детайлизиране на текста изисквано от аудиторията. Тези фактори могат да бъдат групирани в три области:

1. Общи свойства на предвижданото взаимодействие потребител - система за КГТ: какви потребители се имат предвид, каква е темата на интеракцията по същество и т.н.
2. Специфични ситуации на взаимодействие с потребителя, които възникват по време на генерирането, като например, когато някоя конкретна част от текста трябва да бъде обяснена или разширена. Ако потребителят е поставен в конкретна експертна аудиторна група, той ще има нужда от по-малко обяснения за термините в текста.
3. Конкретните текстове се приспособяват към естествените очаквания на потребителя, за да бъдат разпознати. В лингвистиката това се нарича жанр или тип на текста. Например, текст- прогноза за времето има разпознаваеми лингвистични характеристики, които го отличават от текст- техническо ръководство или текст от речник.

Представянето на различни класове потребители се прави със средствата за моделиране на потребителя. Обикновено системите за КГТ поддържат модел на потребителя включващ информация за това какво се очаква потребителят да знае, каква информация е вече спомената в текста, какво потребителят би желал да научи и т.н.

Представянето и използването на знания за типа на текста обикновено се прави чрез явно задаване на макроструктура на текста. Това се дължи на факта, че конкретни типове текст имат често срещани конкретни структури, към които трябва да се придържа и системата за генериране. Това създава полезна организационна схема за процеса на КГТ.

Изключително важно за селектиране на съдържанието е това, че конкретни части от макроструктурата на текста изискват изразяването на определен тип информация. Характерни структури от този вид са въведени в КГТ като схеми на текста [McKeown, 1985] и играят важна роля при структурирането на текстове в много от системите за КГТ. Този традиционен начин за определяне на съдържанието със шаблони на базата на правила прилага парадигмата на т.н.

„затворено планиране”, както например в [Duboue and McKeown, 2003]. Парадигмата на „отворено планиране” е приложена, както например в [Dai et.al., 2010], когато входните данни имат мрежово представяне в граф и съществено се използва топологията на графа за “информирано търсене” на релевантни възли.

В подходите с отвореното планиране съдържанието се представя чрез граф на съдържанието, където възлите съответстват на атоми на съдържанието (факти от база-знания или клетки от база-данни), докато дъгите на графа представят ограничения при селектирането на двойка възли. В някои случаи ограниченията на селектирането са изведени от връзки между данните в съдържанието и служат за индикатори на свързани данни, които могат да бъдат избрани заедно.

В други случаи ограниченията са извлечени от специални релации между факти поставени в плана на дискурса. На възлите и дъгите на графа на съдържанието, върху който оперират стратегиите на отвореното планиране могат да бъдат присвоени тегла. Те модулират силата на ограниченията или оценяват степента на интерес от страна на потребителя към определени типове от съдържанието. Теглата подпомагат прилагането на оптимизационни алгоритми и алгоритми базирани на графи за решаване на проблема със селектирането на съдържание. Теглата могат да бъдат присвоени ръчно, както в подхода на [Demir et.al., 2010] или да бъдат изведени статистически от текстов корпус равен с данните.

Моделирането на потребителя, дефинициите на текстови структури и теглата при мрежовите структури на отвореното планиране частично отговарят на въпроса за информацията релевантна за бъдещия текст. Релевантността може да се оцени само относно нуждата от информация. Моделирането на потребителя дефинира информационните нужди по отношение на потребителя, докато текстовите структури дефинират информационните нужди по отношение на нормите на езиковата култура.

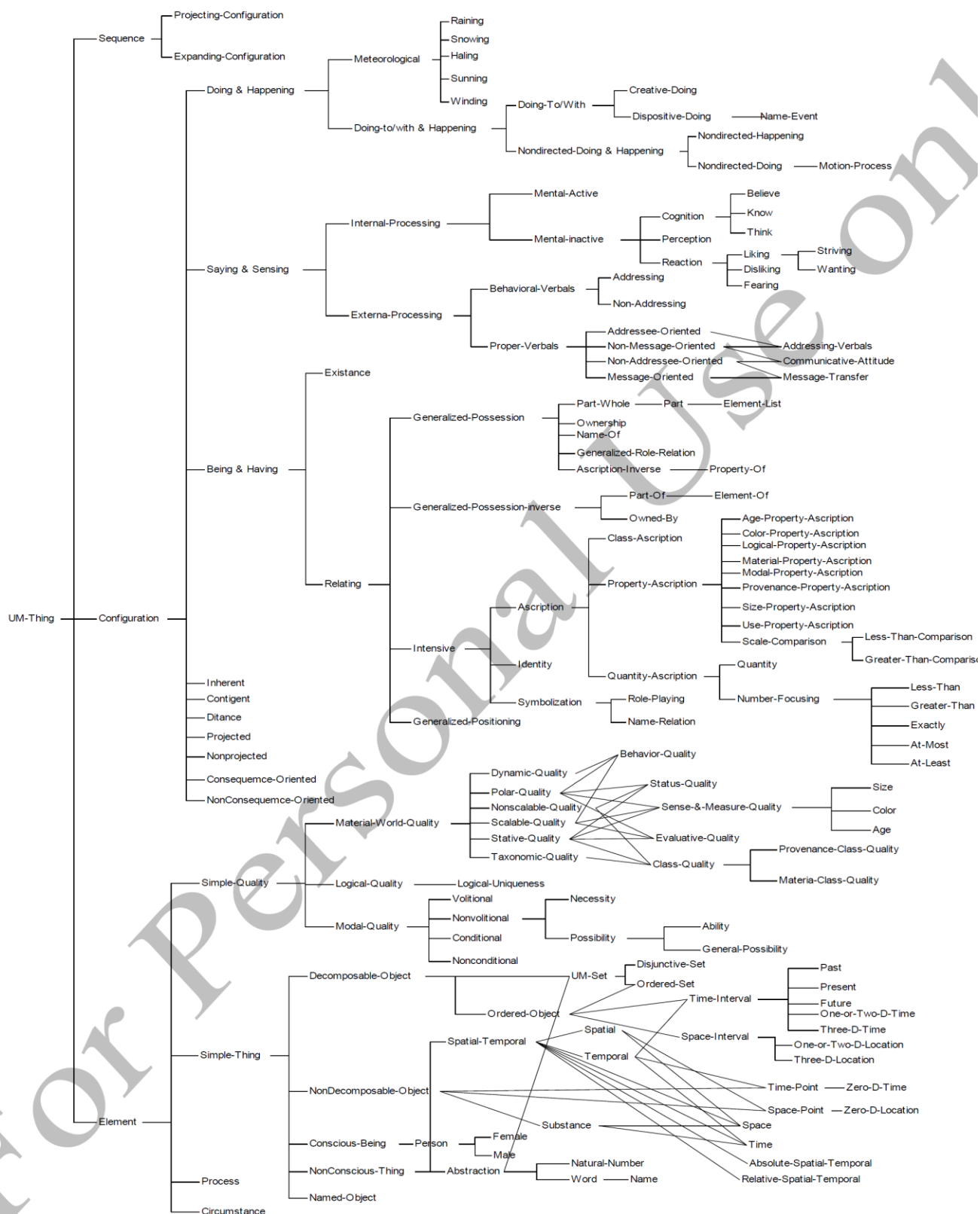
Интерпретиране на съдържанието

Най-простият подход към интерпретиране на съдържанието е да се създадат таблици съпоставящи понятията от приложната област и лингвистичните ресурси на генератора, например като в [Meteer, 1992]. Това гарантира модуларност на информацията за приложението. Ако адаптираното лингвистично представяне е достатъчно абстрактно, то е осигурена и възможност за гъвкава реализация.

Проблематично е специфичното за приложната област кодиране на информацията. Това кодиране изисква задълбочени знания от страна на създателите на системата за КГТ за типовете информация характерни за приложната област на входа. Следователно възниква сложен проблем да се създаде удобна комуникация между независимите от приложната област, преизползваеми и общи представяния на системата за КГТ и конкретната приложна област. Вътрешната за приложението информация трябва да бъде свързана със стратегиите за нейното адекватно изразяване.

Една техника намерила широко приложение е ограничаването на възможните съпоставяния между термините от приложната област и тези от КГТ само до релацията наречена логическо включване (logical subsumption). Това се осигурява, когато всеки конкретен екземпляр на факт, състояние, ситуация и т.н., които се срещат в приложението могат да бъдат класифицирани в термините на йерархия от общи понятия и релации, които имат систематично поведение по отношение на възможната за тях лингвистична реализация.

Използването на онтология от високо ниво при генерирането на текст се определя в [Bateman et.al, 1990] като необходимост при опростяване на интерфейса между знанията специфични за приложната област и лингвистичните ресурси с високо абстрактно ниво прилагани в процеса на КГТ. Предлагащата от онтологията класификационна система поддържа генерирането и



Фигура 2: Йерархия на Обобщения модел UM-Thing, [Bateman, 1997]

същевременно е независима както от приложната област, така и от конкретната задача за генериране. По прагматични съображения някои изследователи използват вместо напълно построен висш онтологичен модел подмножество от понятия на такава фундаментална онтология. Пример за това са системите за генериране на доклади от медицински онтологии на Семантичната мрежа MIAKT [Bontcheva and Wilks, 2004] и ONTOSUM [Bontcheva, 2005].

Пример за цялостна висша йерархия с търсените качества представлява онтологията описана в [Bateman et.al, 1990] и наречена Висш модел (The Upper Model), която е извлечена от системично-функционалната парадигма за естествения език [Halliday, 1994]. Висшият модел е всъщност изчислителен ресурс за представяне на знания специално разработен за компютърно генериране на текст. Абстрактната организация на знанията е лингвистично мотивирана, за да бъде ограничена лингвистичната реализация при генерирането [Bateman, 1990]. Висшият модел е проектиран така, че да бъде преносим, пре-използваем и външен за лексико-граматиката. Той може да бъде считан за вътрешна връзка между специфична за приложната област информация и лингвистичното граматическо ядро на една система за КГТ. Неговата интерфейсна техника поддържа две съществени функционалности:

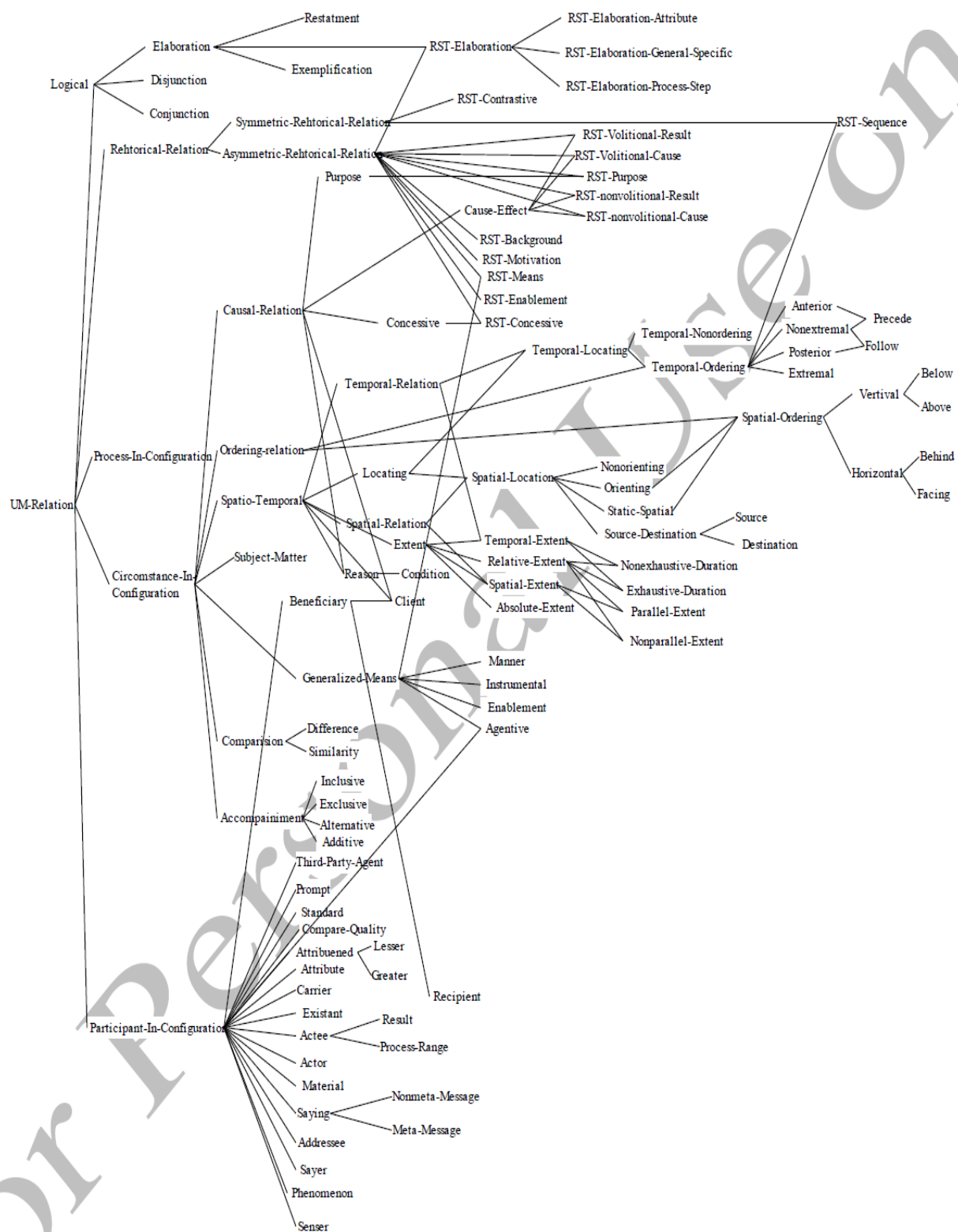
1. Обменът на данни със системата за КГТ е опростен, тъй като голяма част от информацията специфична за езиковите обработки е отделена от изискваните входни спецификации;
2. Нуждата от специфични за приложната област правила за лингвистични обработки е намалена, тъй като Висшият модел предлага независима от областта, обща и пре-използваема организация, класифицираща всички знания специфични за приложната област при лингвистичните обработки.

В първоначалния си вид Висшият модел е бил успешно използван от системата Penman [Matthiessen and Bateman, 1991] за генериране на английски език. Негова модификация получена чрез метод описан в [Novy and Nirenburg, 1992] е приложена за генериране на текстове на английски, немски и холандски език в проекта КОМЕТ⁷ [Henschel, 1993]. Обобщеният модел [Bateman et.al., 1994] поддържа лингвистичните конструкции и за италиански език. Версията документирана в [Bateman et.al., 1995] е най-последователна по отношение на теорията на системично-функционалната лингвистика. Обобщеният модел се състои от две йерархии: една за понятията и една за релациите [Bateman et.al., 1995], както са показани на Фигура 2 и Фигура 3.

Еволюцията на Обобщения модел (Generalized Upper Model) продължава в посока детайлизиране на някои от под-йерархиите [Bateman et.al., 2010] или превеждане на онтологията на онтологичния език на Семантичната мрежа OWL. Обобщеният модел тепърва привлича интереса на изследователите занимаващи се с генериране от данни на Семантичната мрежа, като в [Bouayad-Agha et.al., 2013] се твърди, че той е най-обещаващата многоезикова лингвистично ориентирана онтология поддържаща процеса на текст-генериране.

Обобщеният модел е основната онтология от високо ниво на средата за многоезиково генериране KPML [Bateman, 1997]. Направени са анализи за адекватността на онтологията и към други езици като напр. арабски [Al-Muhtaseb and Mellish, 1997], както и частични анализи от прагматични съображения при генериране на текст в средата KPML. В [Kruijff et.al, 2000] се докладва за реализираното по проекта AGILE многоезиково генериране на текстове-инструкции на руски, български и чешки в средата KPML извършено със съществено прилагане на Обобщения модел за тези езици. Именно частичният анализ направен за проекта AGILE на

⁷ <http://www.nlg-wiki.org/systems/KOMET>



Фигура 3: Йерархия на релациите на Обобщения модел, [Bateman, 1997]

конструкции от българския език по отношение на Обобщения модел е описан по-долу в раздел 3.1.

1.2.2 Техники за планиране на текста

Задачата за планиране на текста представлява организиране на избраното съдържание в подходящи текстови структури, които да бъдат трансформирани в кохерентен текст. Прилаганите подходи към тази задача се различават много по гъвкавост и изчислителна сложност.

Най-простият подход към организацията на текста като цяло са фиксираните структури или шаблоните. Почти всяка текстова структура може да бъде замразена. При генериране с шаблони се жертва огромна част от гъвкавостта на текста, но не всички приложения за КГТ имат нужда от голяма гъвкавост. Замразяването на различни аспекти от цялостния процес на КГТ съответства точно на основните свойства на самия език, тъй като при съставянето на текстове хората понякога използват наготово определени набори от направени лингвистични избори, вместо винаги да правят тези избори „на живо”. Това може да варира от изречения-формули и цели текстове, през различни степени на идиоматичност (на синтаксиса, на начина за структуриране на аргументи, на използваните семантични конфигурации и т.н.). Затова структуриращите шаблони могат да бъдат разглеждани като частично замръзнали резултати от текст планирането.

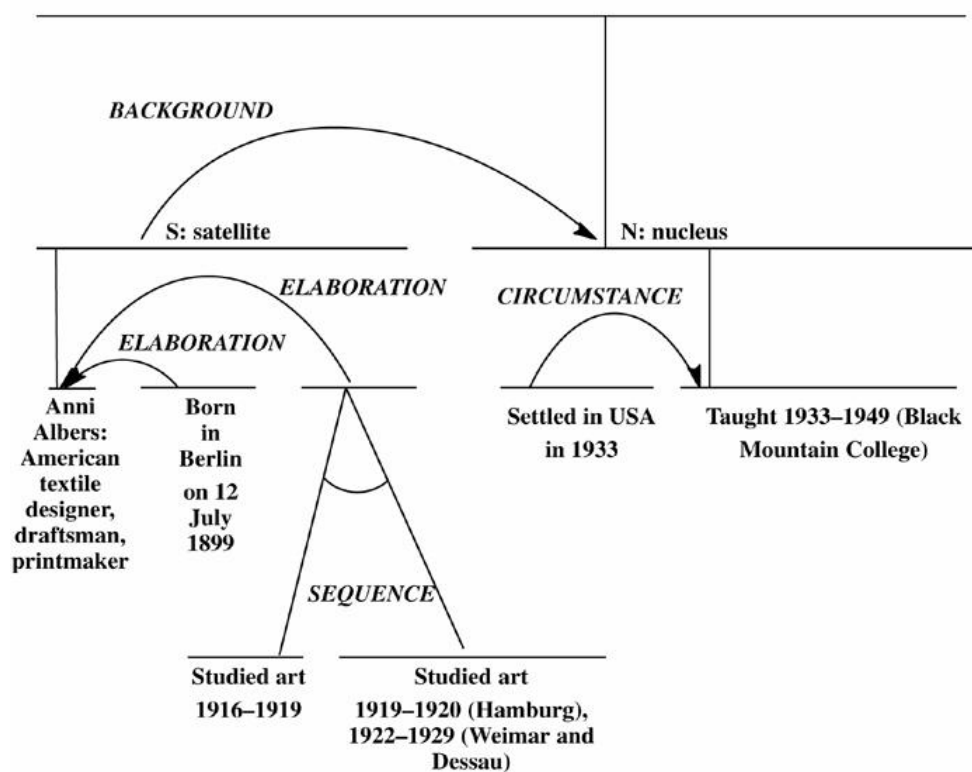
Теоретично генерирането на текст трябва да предлага възможност за свободно движение между повече или по-малко фиксирани образци, комбинирайки ги по понятен начин в изходните текстове. Разработки, които могат да ограничат своите основни описания и да компилират полезни шаблони са обещаващи за спестяване на време и труд. Шаблонната структура може да бъде използвана за текстове, които демонстрират стереотипни конструкции. Сравнително гъвкав тип текстови шаблони са предложени от [McKeown, 1985] и представени в термините на мрежи на преходите (transition networks). Този подход се е превърнал в една от най-разпространените техники за организиране на текст въпреки явните си ограничения.

Създаване на текстови конструкции с по-голяма гъвкавост е възможно чрез прилагане на Теорията за реторичната структура (TPC) [Mann and Thompson, 1988], която предлага общо описание на релациите съществуващи между текстови сегменти като показва дали тези релации са граматически или лексикално сигнализираны. Текстовете анализирани чрез TPC са йерархично декомпозирани на вложени множества от свързани текстови елементи (spans). Теорията за реторичната структура дефинира около 25 релации, които важат между текстовите елементи и са първоначално мотивирани на базата на описателен лингвистичен анализ на около 400 текста с разнообразно съдържание и от различни жанрове. На Фигура 4 по-долу е показан TPC-анализ на биографичен текст илюстриращ някои от дефинираните релации.

Дефинициите на Теорията на реторичната структура поставят изискването да се поддържа връзка със смисъла, който трябва да носи даден елемент на кохерентния текст. Има също ограничения върху комуникационния ефект постигнат чрез комбинираното множество от текстови елементи. Конструирането на дискурсна структура на базата на TPC има доказан ефект при поддържане на селекции от свързващи форми и текстови съюзи. Ефектът се получава, когато дискурсът е разгърнат по конкретен начин, за който е подходяща конкретна форма. Когато една система за КГТ разполага с текстови планове на базата на TPC, то тя разполага и с допълнителната информация мотивираща определени не-пропозиционални семантични избори. Теорията на реторичната структура е използвана също, за да ограничи близки анафорични връзки, избор на тема, селекциите на фокус в текста.

TPC има своето компютърно представяне [Moore and Paris, 1988] и е използвана в много системи за КГТ. Обикновено TPC се прилага за генерирането на текст чрез представяне на реторичните релации като тип комуникационни цели и прилагане на стандартни за изкуствения интелект планиращи стратегии, за да се продуцират текстови структури.

При популярните напоследък приложения генериращи текст от Семантичната мрежа планирането на текста се извършва или след селектиране на съдържанието, или едновременно със селектирането на съдържанието. По време на претърсването на входните данни, които представляват RDF-граф, под внимание се вземат само тези възли, които могат да бъдат свързани с вече селектирани възли чрез дискурсни релации и така се осигурява кохерентността на дискурса. Дискурсните релации между възли от съдържанието са въведени директно в графа на съдържанието преди самото генериране или чрез оператори на плана по време на текст-планирането, или чрез проекции на семантични релации зададени между възли на съдържанието.



Фигура 4 TPC-анализ на биографичен текст, [Bateman, 2002]

Подходите с отворено планиране са типични с демонстрацията на контрол направляван от данните (data-driven methods) при КГТ от обемистите и хетерогенни множества от данни на Семантичната мрежа. Мелиш и Пан [Mellish and Pan, 2006], [Mellish and Pan, 2008] предлагат техника за планиране на текст наречена Natural Language Directed Inference, която избира аксиоми от дадена онтология ако селекцията може да бъде изведена от аксиоми, които са вече селектирани. При подхода на [Dai et.al., 2010] наречен „Генериране на езика на семантична мрежа” (Semantic Network Language Generation) като се започне от възел по предпочитане в дадената семантична мрежа итеративно се избират допълнителни възли в зависимост от

функция отчитаща разстоянието между възли в графа и това определя дискурса на бъдещия текст.

Тези техники са принципно различни от Теорията за реторичната структура, която предлага възможности за адекватно моделиране на комуникационните цели и достигане до по-изразителни лингвистични конструкции на естествения език. Чрез инструментите за прилагане на ТРС в процеса на генериране се постига изразяване на семантична информация извън буквалното пропозиционално съдържание, несравнимо по-голяма кохерентност и гъвкавост на генерирания текст.

1.2.3 Техники за лексико-граматически реализации

Задачата на лексико-граматиката е да съпостави на семантичните конфигурации добре оформени низове на естествен език. Съществуват разнообразни техники за разработване на лексико-граматика. По отношение на гъвкавостта в единия край на скалата са множествата от фиксирани или параметризирани структурни шаблони, които се задействат от специфични комуникационни цели или семантични спецификации. В другия край на скалата са компоненти, които се стремят да съпоставят на всеки добре форматиран семантичен вход кореспондиращ му низ или последователност от низове представящи генерирания изходен текст. Подходите за тази задача съдържат огромна част от работата по КГТ.

Шаблони

Най-простият метод за конструиране на изречения представляват параметризираните шаблони. За дискутирания биографичен текст от горния раздел такъв шаблон може да бъде следната конструкция:

?X was born in ?Y on ?Z.

Символите ?X, ?Y и ?Z са попълнени в конструкцията, за да отговорят на текущата заявка на потребителя. Този метод предварително компилира възможните текстове, тъй като шаблоните обикновено се пишат на ръка. Проблематичен е фактът, че системата за КГТ „знае” твърде малко за елементите, с които работи. Дори най-простата граматическа вариация изисква допълнителни техники за настройване на шаблона. Ако шаблоните са достатъчно гъвкави да позволяват итерации (например, пре-използване на всички места, където е учила Ани Алберс), то резултатните текстове са стилистично незадоволителни. Райтър [Reiter, 1995] детайлно проучва разликите и сравнява ползите от пълно, дълбоко КГТ и шаблонните повърхностни реализации.

Заради простотата си шаблонното генериране е прилагано в някои практически ориентирани системи за КГТ. Вече усложнена тази техника смесва текстови фрагменти – резултат от пълно генериране с предварително заложен шаблон, като по този начин и двата метода имат място в една и съща система за КГТ. Генерирането на базата на шаблони трябва много внимателно да се оценява по отношение на изходните текстове, които са само на един конкретен език, имат твърде елементарни свойства за пре-използване, могат да бъдат стилистично сковани. Шаблоните стават трудни за работа с увеличаване на броя им в една система за КГТ. Повечето съвременни широкопектърни генератори (large-scale generators) предлагат възможности за смесване на пълното генериране с части специфицирани от потребителя.

Реализации с граматик

Гръбнака на граматиките са приложните граматически конструкции. С увеличаването им в пространството на търсене се увеличават и възможните граматически реализации. Решаваща роля имат решения, които предлагат подходяща навигация за това пространство на търсене. Различните видове граматически описания могат да доведат до различни възможности за претърсване на пространството.

Структурната граматика е по същество организирана около описания на фразовата структура. Тя обикновено е претърсвана за приложими правила ограничени от семантиката, която трябва да бъде реализирана. Най-солидната стратегия тук е алгоритъмът за семантично генериране управлявано от опорния елемент (semantic head-driven generation) [Shieber et.al, 1990]. Този алгоритъм генерира стрингове от логически форми за сравнително широк клас граматически формализми.

Техниката работи по същество следвайки последователности от граматически правила свързани чрез техните синтактични главни или опорни елементи (heads), които споделят обща семантика, за да достигнат до приложими лексикални елементи. Ако не са намерени такива правила или последователността стигне до края, се избира някое правило, което декомпозира семантиката недетерминистично. В случай, че се стигне до лексикални елементи, алгоритъмът работи обратно „нагоре” по дървовидната структура като налага ограниченията намерени в лексикона.

Приема се, че елементите на лексиката предлагат най-богатия източник на ограничения за синтактичната структура и затова те са търсени първи, за да се избегне построяването на неприложими структури. Въпреки елегантността и формалната спецификация на алгоритъма той не е използван извън формалното теоретично генериране на изречения. Остават много отворени въпроси по отношение на работата му с големи лексико-граматики със съществени не-пропозиционални семантични изисквания. Недетерминизмът на алгоритъма също е критикуван в средите на КГТ като неподходящо свойство при реално генериране.

Алтернативата наречена *обработка направлявана от съобщението* (message-directed processing) [McDonald, 1983] е предпочетена за построяване на лексико-граматическия ресурс MUMBLE за английски език. Тук детерминистичната и инкрементална конструкция на фразата се контролира директно от входните спецификации. Тези входни спецификации изискват конкретни синтактични фрагменти на дървовидната структура изразени чрез Tree Adjoining Grammars: [Joshi, 1987]. Такъв вход експлицитно идентифицира конкретните граматически конструкции, които трябва да бъдат селектирани за резултатното изречение.

В подобен стил са входните спецификации за повърхностния генератор RealPro [Lavoie and Rambow, 1997]. Входът му представлява представяне, което е синтактична структура на зависимостите (syntactic dependency structure), генераторът я попълва, за да я превърне в напълно определено изречение. Тази техника е повлияна от многослойния Модел на смисъла на текста разработен от Мелчук и колеги. Генераторът RealPro не наследява по-дълбоките и по-абстрактни лингвистични нива предложени от Мелчук, но като следствие е много бърз.

Контрастираща алтернатива, наречена контрол направляван от граматиката, се предлага от Penman [Mann, 1983] и неговия наследник KPML [Bateman, 1997], които са генератори за системично-функционални граматика. Системичните граматика организират своето пространство на търсене около възможни комуникационни цели, а не около граматически структури. Фрагменти на структурите са локализирани в това пространство на характеристики и сами по себе си имат много ограничен статут. Това е изключително ефективно за нуждите на КГТ.

КГТ изисква описания на причините защо да се използват дадени структури (синтактични, текстови и т.н.), а не само формални описания на използваните структури. Това е естествена територия за функционалната лингвистика, която оказва далеч по-голямо влияние върху широкопоспектърните системи за КГТ, отколкото върху анализа на естествен език, където са норма структурните подходи към синтаксиса. Системично-функционалните граматика, с теоретична основа изложена в [Halliday, 1994], се фокусират точно върху прекодиране на връзките във функционална форма и, следователно, са насочени към директен интерфейс между по-високите нива на организация на текста (планиращите процеси) и граматическия компонент.

Традиционно в тези граматика се обръща повече внимание на не-пропозиционалните (текстовите и интерперсонални) аспекти на смисъла [Matthiessen and Bateman, 1991].

Системично-функционалната граматика на Халидей се базира основно на Системно-структурната теория на Фърт и наследява много от идеите на Пражката лингвистична школа. Основно градивно понятие в теорията на Халидей за езика е понятието “система”, което се използва с функционалната му парадигма и е доразвито във формален план чрез възможността да се изграждат “системични мрежи”. Системичните мрежи представляват теория за езика като ресурс за създаване на смисъл. Всяка система в мрежата предлага определен избор и се състои от (1) входно условие, което показва къде се прави избора, (2) множество от възможни изходи, и (3) реализации, които показват какви са структурните следствия в езика за всеки от възможните изходи на системата.

Принципите на функционалността според Халидей са изложени в следните три твърдения:

1. Всеки текст, т.е. всичко, което е казано или написано спада към определен контекст на използване.
2. Всички езици са организирани около два основни смислови типа: експериментен (рефлексивен) смисъл и интерперсонален (активен) смисъл.
3. Всеки езиков елемент е разтълкуван по отношение на неговата функция в лингвистичната система като цяло.

Двете основни функции на езика, експериментната и интерперсоналната, са в основата на всяка езикова употреба: или да се разбере околната среда, или да се въздейства на другите в средата. Тези две базови функции са наречени мета-функции в терминологията на Функционалната лингвистика. Към тях се присъединява като трети компонент текстовата мета-функция, която осъществява връзката между първите две. Въведена е и логическата мета-функция, която изразява логически отношения между езиковите елементи.

За английския език Халидей твърди, че „изречението и думата са двете граматически единици, които се разпознават от народната лингвистика, и в това има значителна доза здрав разум. Вътре в изречението са типични конструиращите отношения от частите към цялото, като частите са езиковите елементи “по-малки” от изречението. При езиковите елементи “по-големи” от изречението рядко се случва да доминират подобни конструиращи отношения и само някои конкретни текстове притежават изграждащи елементи, каквито са структурните единици вътре в изречението. Следователно, изречението заема важна гранична позиция и това е причината то да бъде поставено в центъра на системично-функционалния анализ.”

Според [Halliday, 1994, стр.36] на мета-функциите са съпоставени различни статуси на изречението, както е изобразено в Таблица 1 Мета-функциите и отразяването им в граматиката по-долу. Логическата мета-функция няма реализация като статус на изречението, тъй като тя е вградена не в простото, а в сложното изречение.

Метафункция	Дефиниция (вид значение)	Съответстващ статус на изречението	Предпочитан тип структура
Експериментна	Конструира модела на опита	Изречението като представяне	Сегментна (основана на конституентността)
Интерперсонална	Изразява социалните отношения	Изречението като обмен	Прозодична
Текстова	Създава релевантност за контекста	Изречението като съобщение	Кулминативна
Логическа	Структурира логическите отношения		Итеративна

Таблица 1 Мета-функциите и отразяването им в граматиката

Изречението като съобщение. Според Системичната функционална лингвистика в английския език, както и в много други езици, изречението е организирано като съобщение чрез присвояване на специален статут на една негова част, наречена *тема*. Някой елемент на изречението се избира да бъде *тема* и след това се комбинира с останалата част от изречението наречена *рема*, така че двете части заедно образуват съобщение.

Задълбоченият анализ на английското изречение показва, че не винаги този, който конструира изречението (Говорителят) прави съзнателен избор като поставя определен елемент на първо място. Съществуват правила на словореда за конкретния език, които фиксират някой от елементите на изречението в началото му и така го превръщат в немаркирана тема. Например, запазената първа позиция за подлога в английското декларативно изречение често превръща елемента с роля на подлог в немаркирана тема.

Друга важна характеристика на темата е това, че тя може да бъде изградена от няколко елемента. Различни тематични елементи са типични за различните мета-функции. Например, типичен тематичен елемент на текстовата мета-функция са съюзите.

In this job	Anne	we're working with silver
Тема		Рема

Пример 1 Анализ на английско изречение като съобщение

Изречението като обмен. В процеса на общуване говорещият взема за себе си конкретна речева роля и така придава на слушателя роля, която допълва неговата и която по-късно говорещият би искал да заеме. По отношение на интер-персоналната си роля изречението може да се раздели на две части: Елемент на модуса и Остатък.

В Елемента на модуса е съсредоточено цялото съдържание на отношението Говорител-Слушател за конкретния речеви акт. Елементът на модуса се състои от Подлог реализиран чрез група на съществително и Финитив, който е част от глаголната група. Финитивът изразява модалност или граматическо време. Има случаи, в които Финитивът и лексикалният глагол са слети и тогава лексикалният глагол е натоварен с функцията на Финитив.

Остатъкът се състои от три типа функционални елементи: Предикатор, Допълнение и Адюнкт. В английското изречение може да има само един Предикатор, едно или две Допълнения и неопределен брой Адюнкти (около 7).

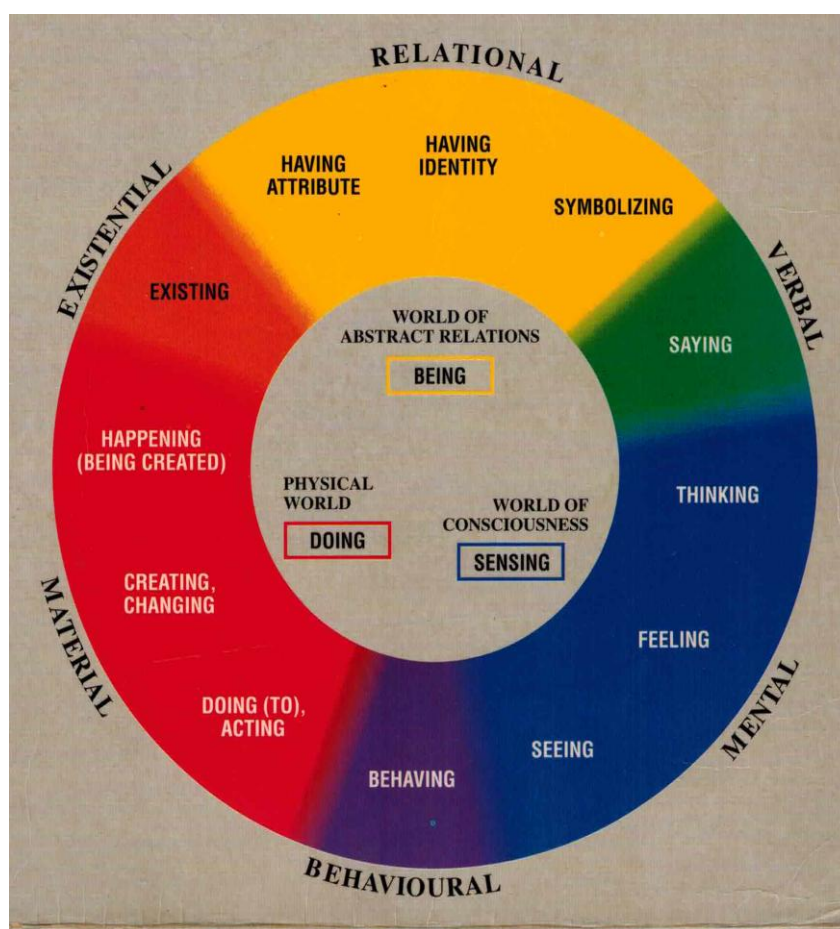
Предикаторът е тази част от глаголната група, която остава при премахване на Финитива. Предикаторът носи вторично време, ако такова е изразено в изречението, определя допълнителни аспекти и фази на процеса, определя залога (активен или пасивен) и самия процес, изпълняван от Подлога. Допълнението се характеризира с това, че има потенциал да бъде подлог на изречението, но не е. Адюнктите нямат потенциал да бъдат подлог и обикновено са реализирани с наречна или предложна група.

In this job	Anne	we	're	working	with silver
Адюнкт		Подлог	Финитив	Предикатор	Адюнкт
Остатък	Вокатив	Елемент на модуса	Елемент на модуса	Остатък	Остатък

Пример 2 Анализ на английско изречение като обмен

Изречението като представяне. Халидей твърди, че „езикът дава на хората възможност да изграждат ментална картина на реалността, да вложат смисъл в това, което става около тях или вътре в тях. Тук отново изречението играе основна роля, защото то възплъщава основния принцип за моделиране на опита, а именно принципа, че реалността е изградена от процеси.” [Halliday, 1994, стр.106]

Според системично- функционалната лингвистика статутът на процес поставен в граматиката на изречението се изгражда по проста схема: всеки процес се състои от три компонента: (1) самият процес, (2) участници в процеса, (3) обстоятелства, свързани с процеса. Такава тристранна интерпретация на процесите е нещо, което се простира отвъд граматическото разграничаване на класовете думи като глаголи, съществителни и останалите думи. Това е шаблонът, който в една или друга форма е най-вероятно универсален за всички езици. Трите компонента на процеса се реализират с различни групи на изречението: самият процес- с глаголна група, участниците- с група на съществителното и обстоятелствата- с наречна или предложна група.



Фигура 5 Типове процеси в Системично-функционалната граматика, [Halliday, 1994]

Излагайки доказателства от английската граматика, Халидей определя четири основни типа процеси и два междинни на базата на транзитивната граматическа система. Предложена е визуална метафора- кръг изобразен на Фигура 5 по-долу, в който типовете процеси са подредени като съседството между различните типове е съществено.

Между материалните процеси, описващи физическия свят и менталните процеси от света на съзнанието Халидей поставя поведенческите процеси, които демонстрират навън вътрешните състояния. Между менталните процеси и релационните процеси, изграждащи света на абстрактните отношения е различима групата на словесните процеси, които са характерни със

съзнателното изграждане на словесни конструкции, за да бъдат изразени абстрактни релации. Между релационните и материалните процеси стои групата на екзистенциалните процеси, чрез които всички видове феномени са просто разпознавани, че „са”, че „съществуват”. Така кръгът е затворен.

Всеки отделен тип процес се свързва с определени характеристики на участниците в него, затова участниците имат свои функционални наименования в зависимост от типа на процеса. Таблица 2 показва най-общо процесите и участниците в тях, както те са извлечени от анализите на английски език.

Тип процес	Смисъл на категорията	Участници
Материален Действие Събитие	правя правя случва се	Актор, Цел
Поведенчески	държа се	Изпълнител
Ментален Перцептивен Афективен Когнитивен	усещам виждам чувствам мисля	Усещащ, Феномен
Словесен	казвам	Казващ, Идея
Релационен Атрибутивен Идентификация	съм съм някакъв съм нещо/някой	Носител, Атрибут Измерван, Стойност Идентифициран, Идентификатор
Екзистенциален	съм, съществувам	Съществуващ

Таблица 2 Типове процеси и ключови участници

Другите функции, които Халидей дефинира за английското изречение като представяне са Бенефициент, Област, Степен, Място, Начин, Причина, Условност (Случайност), Съпровод, Роля, Същност, Гледна точка.

In this job	Anne	we	're working	with silver
Място: абстрактно		Актор	Процес: материален	Начин: средство

Пример 3 Анализ на английско изречение като представяне

Навътре в изречението. Изречението е съставено от различни групи: група на съществителното, глаголна група, наречна група, предложна група като трите компонента (експериентен, интерперсонален и текстов) могат да се разпознаят и в групите. Логическата структура на нивото на групата представя най-общите логически отношения в езика и така се обосновава конструирането на сложни групи.

Над изречението. Сложните изречения са единствените структури по-големи от изречението, които Системично-функционалната граматика разпознава. Съществуват две системични измерения на интерпретацията на логическите връзки между изреченията съставлящи сложното изречение. Едното е измерението на зависимостта, или тактическата система: паратаксис и хипотаксис. Другото е измерението на логико-семантичната система, експанзия и проекция, което е всъщност отношение между процеси изразено в граматиката най-често с комплекси от изречения. Двете измерения дават функционалната рамка за описване на сложните изречения.

Допълнително към изречението. Според Халидей, информационна единица е тази част от речта или текста, която е завършена и цялостна по отношение на предаваната от говорещия

информация. Различните информационни единици в речта или текста могат да бъдат разграничени чрез анализ на ритъма и интонацията естествени за дадения език. Информационната единица не съответства точно на нито един от елементите на граматиката на изречението. Най-близкият граматически елемент е самото изречение.

Около изречението са дефинирани понятията кохезия и дискурс. Според Халидей те са тясно свързани с Темо-Ремната структура на изречението и с функциите Дадено и Ново. Халидей твърди, че „голям брой семантични отношения са вградени чрез влаганията в сложните изречения. Но за да конструираме дискурс ни е необходимо да разполагаме с възможност за изграждане на допълнителни връзки в текста, връзки, които да могат да въвлечат елементи от всякаква големина, и по-малки и по-големи от изречението, от отделни думи до големи текстови пасажии. Това не може да бъде постигнато с граматическа структура, то зависи от ресурс със съвсем различна природа. Тези неструктурни ресурси на дискурса са обозначени с термина кохезия.

Това кратко описание предлага един най-общ поглед към основните постановки и термини в Системично-функционалната граматика. На тази основа се обляга системично-функционалният анализ на всеки естествен език, както и изследванията свързани с генериране на текст в Системично-функционалната парадигма.

Практическа възможност за разработване на приложни системично-функционални граматики дава Средата за многоезиково генериране KPML [Bateman, 1997]. Тя предлага стабилна платформа за работа с широкообхватни граматики и е специално ориентирана към многоезиковото генериране на текст. Основна идея за създаването на Средата KPML е да се предложат ресурси за реалистично, но същевременно широкообхватно генериране, при което се търси както гъвкавост на изходните текстове, така и бързина при генерирането.

В KPML са разработени граматики за над 15 езика между които английски, немски, холандски, китайски, испански, руски. Английската граматика е най-обемистата и изчерпателна системично-функционална граматика NIGEL [Mann and Matthiessen, 1985], върху която се работи от 80-те години на миналия век. Архитектурата за генериране на естествен език реализирана в KPML е всъщност по-нататъшно усъвършенстване на компонента за генериране на изречение на системата за генериране на текст Penman [Matthiessen and Bateman, 1991].

Алгоритъмът за генериране в Средата KPML се състои от преходи през пространство на характеристики на генерирания елемент, като преходите са последователни и с нарастваща специфичност. Всеки такъв преход създава множество ограничения определящи един структурен фрагмент. Този фрагмент може да включва граматически конституенти, които да изискват по-нататъшни преходи, за да се специфицират. Макар много прост, алгоритъмът има предимството, че е доста бърз дори за големи граматики като NIGEL за английски език, граматиката KOMET за немски език [Teich, 1999] или AGILE-граматиките за български, руски и чешки [Bateman et.al., 2000]. В алгоритъма няма връщане назад (backtracking).

Входните данни за системата илюстрират опростяването, което внася за генерирането използването на висш онтологичен модел. Докато за контрола направляван от съобщението на MUMBLE е нужна явна синтактична информация от ниско ниво, такава информация е игнорирана в средата KPML чрез свързването на понятията от приложната област с подходящите понятия от Висшия онтологичен модел.

Стилът на алгоритъма за генериране на Penman има един основен недостатък. При него всеки направен избор трябва да бъде правилен, в противен случай генерирането стартира отново с модифициран вход или повече ограничения. Тази особеност е особено критикувана от разработващите лексикалната част, където може да се случи така, че несистематични пропуски в лексиката да направляват чисто семантичната процедура.

Отчасти в отговор на тези проблеми Формализмът за функционално унифициране (Functional Unification Formalism, FUF) [Elhadad, 1990] предлага по-мощно трасиране на пространството от системични характеристики чрез използване на не-детерминистична експанзия с унификация. Недетерминизмът е направен по-ефективен чрез няколко допълнителни механизма за направляване процеса на унификация [Elhadad and Robin, 1992].

Граматиката за английски език SURGE (Systemic Unification Realization Grammar for English) има много голямо покритие и е създадена за прилагане на FUF. Форматът на нейната нотация е рекурсивна структура от двойки състоящи се от имена на атрибути и техните стойности. Споделяне на структурата или използване на едни и същи стойности за атрибутите е често срещана практика. Стилът на входа за SURGE напомня, от една страна, този на MUMBLE с това, че е по-скоро синтактичен (включващ синтактични категории, лексикални спецификации, решения за граматическото време, членуването и т.н.), а от друга страна- този на Penman/KPML с това, че не се декларира явно какъв вид лингвистична структура се изисква и е представена семантична информация.

Процесът на генериране с FUF се състои от унифициране на такъв вход с подобни на него дефиниции от граматиката. Входните данни направляват процеса на унификация, за да се намерят тези части от граматиката, с които те са съпоставими и да се специфицира нататък структурата в зависимост от ограниченията поставени от граматиката. Подходът с унификация неутрализира до известна степен разделението между контрол направляван от граматиката и контрол направляван от съобщението, доколкото избираният път при унификацията е чувствителен и към двата източника на контрол.

В заключение можем да отбележим, че експерименти за КГТ на най- много различни естествени езици са правени в Средата за многоезиково генериране KPML. Това се дължи на характерния системично- функционален стил при анализа на естествения език улесняващ и насърчаващ аналозиите и паралелните изводи като се спазва посоката от най-общите семантични дефиниции към спецификацията на по-фини нюанси изразявани в конкретния естествен език. Този стил на теоретично и особено на реализационно ниво подпомага пре-използването на базови семантични конструкции и улеснява построяването на лексико-граматики за различните естествени езици. В средата за многоезиково генериране KPML са построени лексико-граматически ресурси за английски, немски, холандски, китайски, испански, руски и други от естествените езици. На базата на този изследователски опит е изграден и ресурсът за генериране на български език Приложна българска системично-функционална граматика.

1.3 Компютърно генериране на текст, семантични технологии и ресурси

Семантичните технологии атакуват проблема формулиран като “липса на семантика при изпълняване на компютърните програми от машините”. Компютрите „не разбират”, „нямат понятие” за смисъла на елементите, с които оперират. Хипотезата, върху която се градят семантичните технологии в компютърната наука е предположението, че компютрите ще демонстрират „по-интелигентно поведение” ако са снабдени с явни, формални описания, въвеждащи семантично ниво на оперативните единици.

Според популярната дефиниция в [Polikoff and Allemang, 2003] семантичните технологии са такива софтуерни технологии, които позволяват смисълът на информацията и асоциациите между информационни единици да бъдат достъпни за обработка по време на изпълнение на програмата.

Семантичните технологии кодират смисъла в софтуерните приложения, както отделно от програмния код, така и отделно от данните, от файловете със съдържание. Това позволява на

компютрите подобно на хората „да разбират”, т.е. да имат интерпретация за целите на софтуерното приложение, „да споделят информация” и да правят изводи в процеса на изпълнение на дадена програма.

Като под-област на Обработването на естествен език, компютърното генериране на текст е повлияно от семантичните технологии в областта, а те са: извличане на информация от текст, представяне на знания и онтологични конструкции, извличане на знания. В настоящия раздел се разглеждат накратко семантичните технологии специфични за Обработването на естествен език, като целта е да оценим ползата от приложенията на семантичните технологии и семантичните пре-използваеми ресурси за КГТ.

Извличане на информация от текст (Information Extraction) се нарича задачата за автоматично извличане на структурирана информация от неструктурирани или полу-структурирани електронни документи. Най-често това е дейност по обработване на текстове на естествен език.

Поради голямата трудност на задачата съвременните методи за извличане на информация от текст са фиксирани върху тясно ограничени приложни области, например, извличане на информация от официални рапорти в корпоративните новини, или извличане на медицинска терминология за конкретна медицинска област и т.н.

Извличането на информация от текст се появява през 70те години на двадесети век заедно с първите опити за обработване на естествен език. В средата на 80те за Ройтерс е създадена първата комерсиална система JASPER, чиито създатели от Carnegie Group имат за цел да предоставят финансови новини в реално време на дилърите на финансовите борси. В днешно време изключителната важност на извличането на информация от текст е следствие от увеличаващия се обем неструктурирана информация достъпна чрез Интернет. Един интелигентен агент наблюдаващ непрекъснатия поток новинарски данни, например, трябва да разполага с технологиите за извличане на информация, за да трансформира неструктурираните данни във вид удобен за правене на изводи. Типично приложение на извличане на информация от текст е сканирането на множество от документи на естествен език и попълването на база-данни с извлечената информация.

Прилагането на извличане на информация от текст е свързано с проблема за опростяване на текста. Основната цел е да се създаде структуриран вид на информацията представена в свободния текст, което да улеснява компютърните обработки. Под-задачите на извличането на информация от текст включват извличане на наименовани понятия, извличане на полу-структурирана информация и анализ на езика и лексикона.

Извличането на наименовани понятия обикновено се състои от разпознаване на наименовани понятия (Named Entities Recognition) т.е. известни имена на хора, фирми и др., наименования на места и териториални области, времеви изрази, и т.н.; разрешаване на кореференции (Coreference Resolution), извличане на релации (Relationship Extraction), конструиране модел на текста (Template Construction). Създаването на подробен модел на текста се извършва, когато вече са разпознати наименовани понятия и релации.

Извличането на полу-структурирана информация се използва за възстановяване на документи, например, при изгубена структура на текста. Най-често това включва извличане на таблици или извличане на коментари от съдържанието на документа.

Анализ на езика и лексикона включва извличане на терминология, което представлява разпознаване на термини свързани с даден текстов корпус. Извличането на терминология се среща в литературата на английски като Terminology Mining, Term Extraction, Term Recognition или Glossary Extraction. Целта на извличането на терминология е автоматично да се направят достъпни термините от даден текстов корпус.

В наше време се увеличава броят на общностите комуникиращи чрез Интернет и на свързаните в мрежи предприятия и организации. Различни приложения на Глобалната мрежа имат за цел да осигурят информационните нужди на тези виртуални общности и затова се създават модели на общностите. Моделите се захранват с подходяща информация, например, теми за организираните по теми Интернет- кроуълри (web crawlers), наименования на услуги за Интернет- услугите (web services), наименования на продуктите за системите препоръчващи продукти (recommender systems) и т.н. Една от първите стъпки при моделирането на областта на знания (knowledge domain) на една виртуална общност е съставянето на речника от свързаните с приложната област термини, които представляват повърхностна лингвистична реализация на понятията от областта.

Обикновено подходите за автоматично извличане на термини използват лингвистични обработки като тагиране на части на речта (part of speech tagging), разделяне на фразите (phrase chunking) и т.н., за да извлекат кандидати за термини измежду подходящи групи на съществителното (напр. кредитна карта) или измежду подходящи предложни групи (напр. борд на директорите). След това кандидатите за термини се филтрират чрез използването на статистически методи или машинно обучение. Вече филтрирани термините не са толкова противоречиви и са достатъчно специфични, за да се използват за концептуализация и изграждане на онтология на приложната област.

Съвременните подходи за извличане на информация от текст са три: прилагане на ръчно написани регулярни изрази, използване на генеративни или дискриминационни класификатори, моделиране на последователности (модели: Hidden Markov Model, CMMs/MEMMs, Conditional Random Fields). Съществуват и редица хибридни техники за извличане на информация от текст, които комбинират някои от стандартните подходи.

Представяне на знания (ПЗ). Научните изследвания по представяне на знания (Knowledge representation) включват анализи как се разсъждава правилно и ефективно, а също и кой е най-добрият начин да се използва множество от символи за представяне на множество от факти в дадена приложна област. За да стане възможно правенето на логически изводи (inferences) за елементи на конкретното представяне на знания, а също и създаването на нови твърдения в това представяне, се използват речник от символи и логическа система. Логика се използва също за дефиниране как да се прилагат операторите и да се преформулират знанията. Примери за оператори и операции са отрицание, конюнкция, допълнения, адюнкти, кванфикатори и модални оператори. Логиката е интерпретационна теория. Трите елемента: символи, оператори и интерпретационната теория дават смисъла на последователностите от символи в представянето на знания.

В прилагането на системи за представяне на знания при реални задачи сложността на задачата може да надхвърли ограниченията на ресурса или възможностите на системата за представяне на знания. Съществуват няколко основни технологии за представяне на знания: фрейми, правила, тагиране/ маркиране, семантични мрежи. Новите разработки в ПЗ включват идеята за Семантичната мрежа и разработването на фамилия езици и стандарти за представяне на знания базирани на Разширения език за маркиране (XML). Тъй като знанията се използват за постигане на интелигентно поведение, то основната цел на представянето на знания е да осигури разсъжденията и правенето на изводи. Доброто представяне на знания трябва да бъде едновременно декларативно и процедурно.

Какво представлява представянето на знания може добре да бъде осмислено в термините на пет различни роли, които то изпълнява [Davis et.al., 1993] като всяка роля може да бъде ключова за конкретната задача:

1. Представянето на знания е най-вече абстракция, заместител на нещото като такова.
2. Представянето на знания е множество от онтологични уговорки (ontological commitments).
3. Представянето на знания е фрагментарна теория за интелигентно правене на изводи, изразено в термините на три компонента: (i) основната концепция за интелигентно правене на изводи присъща на представянето; (ii) множеството от изводи, които се санкционират от представянето; и (iii) множеството от изводи, които се насърчават от представянето.
4. Представянето на знания е среда за прагматични ефективни изчисления.
5. Представянето на знания е среда на естествено за хората изразяване, например езикът, на който комуникираме и описваме света.

Онтологии. Представянето на знания чрез онтологични конструкции е класически подход във философията, който през последните години получава подобаващо внимание в информатиката. Онтологиите са особено актуални в стремежа за разрешаване на проблема с липсата на семантика при компютърното обработване на огромния обем информация в Интернет.

В компютърната наука и информатиката всяка онтология представя формално даден обем знания като множество от понятия/ концепти в конкретната приложна област и релации между такива понятия/ концепти. Онтологията може да бъде използвана за моделиране на приложната област и за основа на правенето на изводи (reasoning) за онтологичните елементи. Общоприетата дефиницията на онтология в информатиката е: „формална, експлицитна спецификация на споделяна концептуализация” [Gruber, 1993].

Онтологията предоставя общ, споделян от специалистите в областта лексикон (vocabulary) и таксономия, която моделира приложната област чрез дефинирането на обекти/ концепти и техните свойства и релации. Онтологиите са структурни модели за организиране на информацията и се използват като форма за представяне на знания за света или за някаква част от него.

Съвременните онтологии имат много структурни прилики без значение какъв е езикът, на който са изразени. Повечето онтологии описват индивиди (екземпляри), класове (понятия), атрибути и релации. Индивидите са екземпляри или обекти представляващи основното ниво онтологични елементи. Класовете са множества, които могат да бъдат интерпретирани като колекции, понятия, програмни класове, типове обекти и т.н. Атрибутите представляват аспекти, свойства, характеристики или параметри, които обектите могат да притежават. Релациите са начините, по които класовете и индивидите могат да бъдат свързани едни с други. Необходими елементи за описване на онтологии са също функционалните термини, ограниченията, правилата, аксиомите, събитията от процеса на онтологично моделиране.

Онтологичен език е формален език, който се използва за кодиране на онтологии. Съществуват много такива езици, както частни, така и създадени на базата на стандарти.

Висши онтологии и онтологии на приложни области. Онтологията на дадена приложна област моделира конкретна част от света. Конкретните значения на термините от дадената приложна област се дефинират от нейната онтология.

Висша онтология или фундаментална онтология е модел на общи онтологични обекти приложими в голям брой онтологии на конкретни приложни области. Висшата онтология използва базов речник, който съдържа термините и асоциираните описания на обекти като те могат да се използват в различни релевантни приложни области. В днешно време най-често се използват няколко от създадените достъпни стандартизирани висши онтологии: Dublin Core (DC), Онтология за географско разположение (Geospatial Ontology - GEO),

OpenCyc/ResearchCyc, SUMO, Описателната онтология за лингвистично и когнитивно инженерство (DOLCE), Концептуален референтен модел за историческото наследство (CIDOC-CRM).

Извличане на знания (ИЗ) в информатиката е конструирането на знания в някакъв формализъм за представяне на знания, като данните са взети от структурирани електронни източници (релационни бази-данни, XML-данни) или от неструктурирани електронни източници (текст, документи, изображения). Получените като резултат знания са във формат удобен за компютърни обработки и имат такова представяне, което да осигурява правенето на изводи (inferencing).

Критерият, който може да превърне извличането на информация в извличане на знания е стремежът за пре-използване на вече съществуващи фрагменти от някакво представяне на знания. Като методология извличането на информация и извличането на знания си приличат, но резултатът от извличане на знания е не просто структурирана информация или информация трансформирана в релационна схема. При извличането на знания се пре-използват вече формализирани знания (например, идентификатори на ресурси URI или елементи от онтологии) или се генерират такива конструкции на базата на първичните данни.

Извличането на знания от източници на естествен език винаги е било предизвикателство за изследователите. Според дадената по-горе дефиниция на Извличане на знания неструктурираната информация не е подходяща за тази дейност и се налага прилагането на по-сложни методи, които при това в общия случай дават по-лоши резултати, отколкото ако е възможно използването на структурирана информация. Обемът на придобитите извлечени знания би трябвало да компенсира нарастващата сложност и намаляващото качество на извличането. Затова работещите в тази област приемат източниците на естествен език като данни в неструктуриран вид, като неструктурирани текстове.

Основната задача на семантичните технологии свързани с традиционното извличане на информация е структурирането на информацията по подходящ начин. Големият проблем е, че трябва да се определи в модел какъв да бъде видът информация, която ще се идентифицира преди да започне процесът по извличането ѝ, което прави цялата дейност зависима от приложната област.

Извличане на информация на базата на онтологии [Wimalasuriya and Dou, 2010] е подобласт на извличането на информация, при което се използва поне една онтология водеща процеса на извличане на информация от текстове на естествен език. Следователно, системите за извличане на информация на базата на онтологии използват съществено традиционното Извличане на информация от текст, за да идентифицират понятия, екземпляри и релации на онтологиите, с които се работи върху текста, като извлечените понятия се структурират в изходна онтология след приключване на процеса. Така самите входни онтологии представляват модел на информацията, която предстои да бъде извлечена.

Извличането на онтологии от текст (Ontology Learning/ Ontology Extraction/ Ontology Generation/ Ontology Acquisition) е под-област на извличането на информация от текст. Целта на извличането на онтологии от текст е от даден текстов корпус или от друг вид зададени множества данни да се извлекат полуавтоматично подходящи понятия и релации и да се конструира онтология.

Автоматичното създаване на онтологии е задача включваща много дисциплини. Обикновено процесът започва с извличане на термини и понятия от групи на съществителното от текстовете чрез методите на Извличане на терминология. Това най-често включва лингвистични обработки, например тагиране на частите на речта, разделяне на фразите и т.н. След това се прилагат статистически технологии, или символни технологии за извличане на сигнатури на релациите.

Търсените характеристики на приложната област се формализират в онтология. Чрез методите на Извличането на онтологии от текстове на естествен език полуавтоматично се получават големи онтологични конструкции, които могат да бъдат приложени в онтологичното инженерство.

При *семантичното аотиране* на текстове на естествен език според дефиницията в [Erdmann et.al, 2000] към текстовете се прикачат метаданни, които трябва да направят семантиката на термините в текста „разбираема” за машините. При този процес, който е по принцип полуавтоматичен, се извличат знания, в смисъл, че между лексически термини от текста и, например, онтологични понятия се установява връзка. Така се придобиват знания, чиито смисъл е бил търсен в обработвания контекст.

Полуавтоматичното семантично аотиране може да бъде разделено на две под-задачи: извличане на терминология и свързване на елементи. При извличането на терминология се извличат всъщност лексически единици от текста. За целта се използват токънизатори (tokenizers), които първо определят отделните думи и разпознават съкращенията. След това от текста се извличат термините, които съответстват на понятия от специфичен лексикон за приложната област, за да се свържат на следващата стъпка.

При свързването на елементи, както например в [Rao et.al., 2011], се установява връзката между извлечените лексически термини от текста и понятията от онтологията. Това става чрез разпознаване на няколко понятия- кандидати за различни значения на термина, отново с помощта на лексикон. Накрая се анализира контекста, в който е използван терминът, за да се предвиди най-вероятното значение, да се разреши противоречивостта (disambiguation) и да се свърже терминът с правилното понятие от онтологията.

При подхода за семантично аотиране на текст предложен в [Simov and Osenova, 2007] и доразработен в [Simov and Osenova, 2008] към семантичното аотиране се подхожда в перспектива от онтологията към текста, оттам и наименованието му „Релация: От онтология към текст”. При него се използват лексикони, основани на онтология (Ontology-Based Lexicons).

Основната идея на метода е специално разработени ресурси, терминологичен лексикон и анотационна граматика, да се използват в съчетание при разпознаването на срещания на онтологичните понятия в текст. Дефинираната задача никак не е тривиална, тъй като (1) не всички онтологични понятия задължително имат лексикализация, (2) онтологичните понятия не винаги се срещат в текстовете във вида, в който са лексикализирани в онтологията от експерти или специалисти в предметната област и (3) едни и същи онтологични понятия могат да бъдат изразени в естествено-езиков текст по множество различни начини и представени по смисъл със свободни фрази.

Терминологичният лексикон е множество от лексикални еквиваленти на понятията от дадена онтология. Той изпълнява двустранна роля. Първо, лексиконът свързва понятията на онтологията с лексическото знание, използвано от граматиката за разпознаване на ролята на понятието като езиков елемент на текста. Второ, лексиконът представлява основа за създаване на удобен интерфейс между потребителя и онтологията, който позволява онтологията да бъде представена по естествен за потребителя начин.

Всеки елемент на лексикона се състои от термин, контекстна информация за употребата на термина и граматически характеристики, определящи синтактичната реализация в текста. За получаване на последните два елемента са необходими допълнителни езикови ресурси освен лексикализациите съдържащи се в онтология. По тази причина е необходим и корпус от подходящи текстове, където термините са използвани в различни синтактични съчетания. Корпусът се използва за „обучение” на лексикона, при което лексиконът се обогатява с фрази и словосъчетания за всеки термин. При използване на адекватна граматика това повишава

вероятността за разпознаване на съответните термини в целевите текстове. Събирането на такива фрази и словосъчетания за термините е първият етап на процес на обучение при подхода „Релацията: От онтология към текст”.

Анотационната граматика е средство за разпознаване на онтологични термини в целевите текстове. В идеалния случай тя представлява разширение на една обща дълбочинна граматика за даден език, адаптирана към конкретната анотационна задача. Като минимум анотационната граматика представлява частична граматика за аотиране на понятията с добавени правила за разрешаване на многозначност. Частичната граматика съдържа за всеки термин от лексикона най-малко едно граматическо правило за разпознаване на понятието. За да работи анотационната граматика е необходима предварителна обработка на текста, т.е. аотирането му с граматически характеристики и лематизация. Настройването на граматиката може да се счита за втори етап на обучение при прилагане на подхода „Релация: От онтология към текст”.

В сравнение с класическия метод за семантично аотиране при следване на релацията „От онтология към текст” се използват съществено онтологии като ресурс за съставяне и обогатяване на терминологичния лексикон, което е много удачно при работа с текстове класифицирани тематично в определена област на знанието. КГТ изисква представяне на знанията в тематичната област на бъдещите текстове, затова семантично аотиране с метода „От онтология към текст” е подходяща подготвителна дейност за генерирането. Ползата от преизползване на семантично аотирани текстове би била по-голяма, както за настройване на процеса на КГТ, така и за оценяването му.

Можем да направим извода, че семантичните технологии като цяло реализират алгоритми и решения, които дават семантична структура на информацията, за да се ползва тя и от хората и от компютрите. Това е интересна перспектива по отношение на Компютърното генериране на текст, като се има предвид важността на семантичните представяния за процеса на КГТ. От направения кратък преглед на семантичните технологии в областта Обработка на естествен език можем да заключим, че те са полезни при предварителните обработки на големи обеми от текстова информация и могат да послужат за подготовка на ресурси за КГТ, като например, онтологично базирани лексикони.

Компютърно генериране на текст от Семантичната мрежа

Технологиите на Семантичната мрежа са създадени да въведат стандарт при представянето на знания по принцип и особено в Интернет-пространството. Визията за Семантичната мрежа като „надстройка” на Глобалната мрежа [Berners-Lee et.al., 2001] включва представата за структурирано и достъпно знание, за механизми извличащи отговори от тази „Глобална база-знания” и най-вече представата, че структурираната форма на представените знания е достъпна и „разбираема” както за хората, така и за компютрите.

Технологиите на Семантичната мрежа са формализирани чрез фамилия специфични технологични стандарти на консорциума W3C⁸, които са създадени да свързват данните в Глобалната мрежа и в локалните мрежи на организации, правителства, фирми, общества и т.н.

Първоначално (до 2009г.) идеята на Семантична мрежа е била да се аотират Интернет-страниците с множество мета-атрибути и категории, за да се даде възможност на компютрите чрез такова семантично аотиране да интерпретират текста и да го поставят в някакъв контекст. Интернет-страниците, обаче, съдържат много информация, която не е релевантна за компютрите и трябва да бъде филтрирана, тъй като компютрите работят най-добре с информация без шум.

⁸ <http://www.w3.org/Consortium/>

Идеята за Мрежа на данните възниква като резултат от тези ограничения и от факта, че съществуват огромен брой структурирани множества от данни разпръснати в различни точки на света и съдържащи най-разнообразна информация. Обикновено едно множество от данни съдържа знания за конкретна приложна област, напр. книги, музика, енциклопедични данни, компании и предприятия, и т.н. Идеята е те да се свържат с линкове като Интернет-страници и компютрите да могат да достъпват тази независима мрежа от структурирана информация без шум, за да събират знания от различни приложни области. Очакваният резултат според Тим Бърнарс-Лий⁹ е масивна и достъпна база-знания, която да оформи основата на ново поколение приложения и услуги.

Сериозна стъпка в тази посока е проектът на W3C „Свързване на отворени данни” (Linking Open Data, LOD)¹⁰. Проектът следва основните направления в дизайна на Глобалната мрежа: простота, толерантност, модуларност и децентрализация. Множествата от данни са организирани така, че да пре-използват съществуващите онтологии, сред които най-популярни са Dublin Core (DC), Friend Of A Friend (FOAF), Simple Knowledge Organization System (SKOS), Geospatial Ontology (GEO), споменати и по-горе. Всяко множество от данни позволява достъп до своите бази-знания и своите линкове към елементи на други множества от данни.

Чрез проекта „Свързване на отворени данни” са достъпни 337 такива множества¹¹ с повече от 31.6 билиона RDF-тройки¹², което представлява огромен обем знания. Множествата от данни могат да бъдат достъпвани чрез браузър на Семантичната мрежа или чрез машини за семантично търсене (кроуълъри).
Фигура 6: Диаграма на Облака от отворени свързани данни

представлява диаграма на „Облака от отворени свързани данни”. Различните множества от данни са представени в различни цветове за различните приложни области (медийни данни, географски данни, правителствена информация и т.н.).

Основанието за големия интерес на работещите върху Компютърното генериране на текст към Семантичната мрежа е фактът, че Семантичната мрежа чрез инициативата „Свързване на отворени данни” предлага свободно големи обеми данни в RDF-формат, които могат да служат за входно представяне при КГТ. За работа с големия обем хетерогенни данни на Семантичната мрежа се изисква подходите към КГТ да бъдат адаптивни и стабилни. Затова за изследователите става предизвикателство да се намерят решения за по-дългосрочните задачи пред КГТ като преносимост, стабилност и адаптивност на системите, а също и възможност за оценяване на генерираните текстове. В [Bouayad-Agha et.al., 2013] са формулирани предимствата при генериране от данни на Семантичната мрежа:

-Кодирането на знания свързани с КГТ като правила и шаблони е осигурено от популярни приложни интерфейси и стандартния език за заявки на RDF – SPARQL. Специфичните за КГТ знания за дискурса и знания от приложната област могат да бъдат моделирани в отделни онтологии и интегрирани чрез механизма за импорт на онтологичния език на Глобалната мрежа (OWL).

-Оценяването на данните и интерпретирането на знанията от предметната област са до голяма степен осигурени от програмите за правене на изводи и машините за правила, които изпълняват стандартни онтологични операции.

Съвременните подходи към КГТ от данни на Семантичната мрежа имат няколко функционални класификации, които до голяма степен съвпадат. В [Bontcheva and Cunningham, 2003] се разграничават два типа приложения на КГТ ориентирани към Семантичната мрежа: такива,

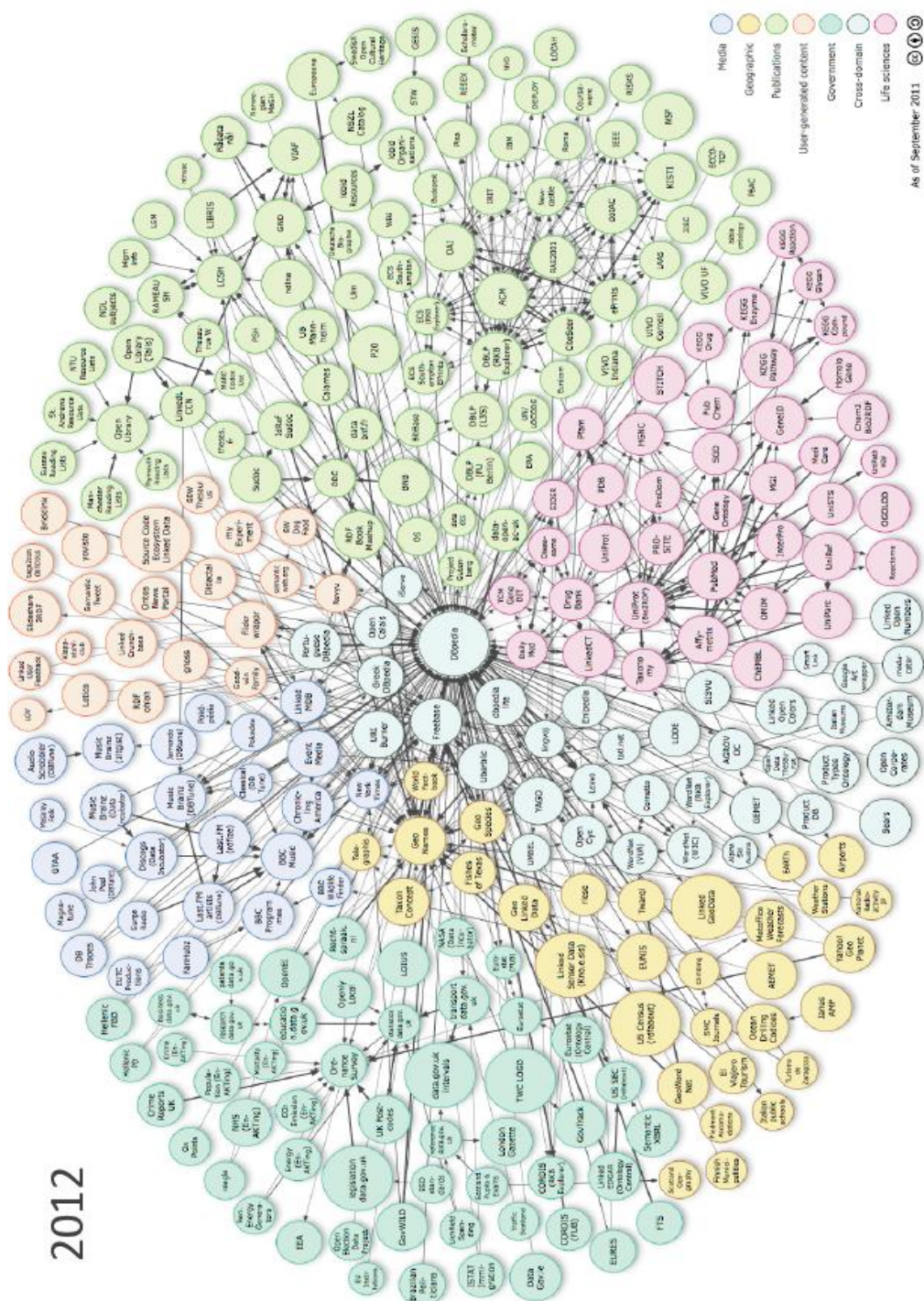
⁹http://www.ted.com/talks/tim_berniers_lee_on_the_next_web.html

¹⁰ <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

¹¹ Юли, 2013г.

¹² Данни на работната група Linking Open Data Cloud (Облак от отворени свързани данни), която каталогизира достъпните в Интернет множества от данни <http://datahub.io/group/about/lodcloud>

Фигура 6: Диаграма на Облака от отворени свързани данни



които „подпомагат потребители- неспециалисти по представяне на знания да разберат и използват онтоологиите” и приложения, които да продуцират „текстове на естествен език като доклади, писма и т.н., което означава да представят структурирана информация по начин удобен за потребителя.” Подобно разграничаване се прави и между интерфейсите на естествен език: за онтологично инженерство и за вербализиране на онтологично- представено знание с конкретна комуникационна цел чрез КГТ [Gardent et.al., 2011].

“Планиране на текст” и “Планиране на изречение” са класифицирани в [Bouayad-Agha et.al., 2013] като двете семантично обвързани задачи при съвременните приложения за КГТ от данни на Семантичната мрежа. Някои от подходите са стандартни за КГТ с допълнително адаптиране към данните на Семантичната мрежа, докато други са настроени изцяло към характеристиките на данните на Семантичната мрежа. Най-представителните приложения за КГТ от данни на Семантичната мрежа могат да бъдат разделени в две групи- такива, създадени за подпомагане работата с онтологии като интерфейси на естествен език, и такива, които служат за вербализиране на знания.

При планирането на текст са формулирани четири комуникационни цели, които могат да бъдат изпълнени при КГТ от данни на Семантичната мрежа: (1) генерирай всичко, което може да се каже по отношение на входния обект (клас, задание, граф), както например в [Bontcheva and Wilks, 2004]; (2) вербализирай съдържанието, което е интерактивно зададено от потребителя, например в [Power, 2009]; (3) вербализирай най-характерните факти намерени в реални или виртуални целеви текстове, например в [Dannells et.al., 2012]; и (4) генерирай в естествено-езиков текст най-подходящите факти според контекста, например в [Dai et.al., 2010].

При планиране на изречение за КГТ от данни на Семантичната мрежа се използват основно прости правила и шаблони. Затова за комплектоване и агрегация на изреченията се прилагат семантични граматиките [Dannells et.al., 2012], правила на SPARQL [Bouayad-Agha et.al., 2013], шаблони на XML, модели за агрегация на базата на аксиоми.

По отношение на интерпретирането на съдържание сложността на прилаганите подходи е по принцип пропорционална на флуентността на изходния текст като в началото на скалата е директното съпоставяне на етикетите от онтоологиите и лингвистичните реализации. В [Davis et.al., 2008], например, имената на онтологичните класове се третират като лични имена, както в изречението „SeniorResearcher attends Conference.”. Това предположение за лингвистична изразност на онтологията, когато аксиомите се изразяват чрез изречения построени според граматика и атомарните понятия се вербализират с отделни елементи на лексикона, е наречено в [Power, 2010] Модел на консенсуса.

Такъв консенсус е потвърден от две различни изследвания в [Mellish and Sun, 2006] и в [Power and Third, 2010], които анализират в колекции от достъпни онтологии от една страна зависимостите при наименоване на свойства и класове, и от друга – сложността и лингвистичната експресивност на аксиомите на OWL. Според привържениците на Модела на консенсуса, това, което подходите с директно съпоставяне губят по отношение на флуентността на изходния текст е всъщност печалба по отношение на независимостта от предметната област и простотата на инженерното решение. Така се намалява значително цената за лексикони зависими от приложната област и от онтологията. При това за подобряване флуентността на текстовете могат да се използват нетипични обработки за КГТ като автоматична проверка за изписването на думите, премахване или наблюдение на повторения, представяне на части от изречението чрез изреждане в списък.

Тук накратко са разгледани различни подходи към интерпретирането на съдържание на Семантичната мрежа и лингвистичното му представяне: подходи на базата на Модела на

консенсуса, включително използващи контролиран естествен език, подходи за аотиране на съдържанието с лингвистични знания, и подходи, които използват висши онтологични модели за интерпретиране на съдържанието и съпоставяне между съдържанието и лингвистичната реализация.

При прилагане Модела на консенсуса флуентността на лингвистичния изход може евентуално да бъде подобрена чрез проучване за лингвистични шаблони за имената на свойствата и класовете. При идентифициране на шаблони в онтологиите могат да се използват он-лайн лексически ресурси като WordNet¹³ или FrameNet¹⁴, за да се асоциират тези шаблони с информация за валентността в изречението.

Има изследователи, които предлагат автоматичното извличане на лексико-граматически единици от етикетите в онтологиите да се използва само като резервен вариант, когато няма ръчно въведени аотации [Stevens et.al., 2011]. Други смятат, че при създаването на онтологиите трябва да се спазват конвенции за наименованията, които да ограничават граматическите категории и композиции от термини [Hewlett et.al., 2005]. Междувременно, начинът да се заобиколи липсата на конвенции за наименованията е въвеждането на потребителите на системите за КГТ в оформянето на лингвистичните ресурси за конкретната приложна област, както се предлага чрез концептуалното авторство [Hielkema et.al., 2007].

Често използвана стратегия за намаляване на знанията специфични за конкретната задача, за да се генерира текст на естествен език е ограничаването на генерираните конструкции до едно контролирано множество, наречено контролиран естествен език (Controlled Natural Language CNL). За такъв език може да бъде дефинирано непротиворечиво съпоставяне с формалните езици на Семантичната мрежа, както в [Davis et.al., 2008]. Подходът е подходящ при вербализиране на онтологии, където генерираният текст трябва да бъде непротиворечив и няма изисквания за кохерентност на текста. Важно е да се отбележи, че моделът на консенсуса използва шаблони от онтологии, които са разработени най-вече на английски език. Всички използвани контролирани естествени езици са също подмножества на английския език, затова и повечето подходи за онтологично инженерство генерират английски текстове.

Аотирането на съдържанието с лингвистични знания е решение, при което имаме аотиране на данните от приложната област с лексикална информация вместо поддържане на отделен лексикон, напр. в NaturalOWL [Galanis and Androustopoulos, 2007]. Класове и индивиди в OWL-онтологията са асоциирани с групи на съществителното заедно с рода на основното съществително име и неговите форми за единствено и множествено число. На свойствата в онтологията са присвоени микро-планове за планиране на изречение. Тези микро-планове дефинират абстрактна структура на изречение с главен глагол и информация за склонението на глагола и валентността му. Такова тясно обвързване на лингвистични знания и знания от приложната област определено среща категорични контра мнения, тъй като ограничава преизползването на лингвистични данни в различни постановки за приложението, които биха изисквали различно съпоставяне.

Използване на висша онтология за интерпретиране на съдържание. Използването на представяния независими от заданието и от приложната област, като, например, Обобщеният модел (GUM) [Bateman et.al., 1995], осигурява съпоставянето на съдържанието и лингвистичното представяне.

По прагматични съображения, някои изследователи предлагат вместо напълно построен висш модел, да се използва подмножество от понятия на такава фундаментална онтология, за да се осигури както съпоставимостта, така и преносимост на системата за КГТ. Пример за това е

¹³ <http://wordnet.princeton.edu/>

¹⁴ <https://framenet.icsi.berkeley.edu/>

МІАКТ [Bontcheva and Wilks, 2004] - една от първите системи за генериране на доклади от медицински онтологии на Семантичната мрежа. Повърхностният генератор на МІАКТ HYLITE+ се прилага след като всички релации от входните онтологии са съпоставени на една от четири същностни и лингвистично мотивирани релации, за които съществува вградена поддръжка на лингвистичното представяне. Този подход е разширен в ONTOSUM [Bontcheva, 2005], където текстът и модулите за планиране на изречения могат да работят с всяка онтология, която съдържа същите четири релации от високо ниво.

Налага се изводът, че както на идейно така и на реализационно ниво използването на висша онтология като Обобщения модел подробно представен в раздел 1.2.1 е печелившо решение и има бъдеще с най-новите приложения за генериране на текст от данни на Семантичната мрежа.

Една от оригиналните и далечни цели на Семантичната мрежа е да стане възможно изпълняването на сложни задачи от агенти вместо от потребителите, чрез извличане и обработване на смислено съдържание и чрез комуникация по начин удобен за потребителя [Berners-Lee et.al., 2001]. Такива приложения вече се появяват, разработени в ограничени приложни области и за ограничен кръг задачи. Компонентите на КГТ в тези приложения са много специфични и разчитат на прости шаблони за генериране на текст.

Все още сме твърде далече от интелигентна система с изкуствен интелект, която да може да представи в разговор информационно съдържание отчитащо контекста [Cimiano and Kopp, 2010]. Най-вероятно е това да се получи ако се обърне повече внимание на лингвистичната природа на текста. КГТ на основата на Системично-функционалната лингвистика използва именно това чрез функционалния анализ на естествено-езиковата продукция.

Съществуващият опит и съществуващите ресурси за различни естествени езици са обещаващо начало при евентуални изследвания на КГТ от Семантичната мрежа със средствата на приложената Системично-функционална лингвистика. Интересен е и въпросът за многоезичността, която е естествена за Глобалната мрежа с универсалните ѝ стандарти за всички езици. Респективно многоезиковото КГТ би могло да спечели много използвайки потенциала на Семантичната мрежа, а парадигмата на Абстрактната генерираща система е изключително подходяща за многоезиково КГТ. Приложенията отчитащи семантиката на езика на всяко ниво имат обещаващо бъдеще, затова в изследванията на настоящата дисертация е избрана тази парадигма.

Изводи

От направения обзор е видно, че компютърното генериране на текст е област със специфична теоретична платформа и полезни прагматични реализации, област, която тепърва предстои да разгърне потенциала си. От една страна за КГТ са важни солидните теоретични изследвания свързани с лингвистичната същност на изходния продукт- текстът на естествен език, от друга страна своите аргументи има и едно по-технократско отношение към КГТ, което въвлича по специфичен начин някои от съвременните семантични технологии. Безспорно е, че и двата подхода изискват солидни лингвистични ресурси представени в една или друга форма според избрания метод за генериране. Без такова представяне на лингвистични знания, което по същество са лингвистичните ресурси, не може да съществува компютърно генериране на текст.

Най-богати и детайлизирани са наличните лингвистичните ресурси, чрез които се продуцират текстове на английски език и все още КГТ на друг естествен език е сериозно изследователско предизвикателство. Поради това основна идея на настоящата дисертация е изследването как българският език може да бъде обект на анализ и моделиране, за да се създаде лингвистичен ресурс за КГТ на български език.

Паралелното изследване на няколко естествени езика през призмата на дадена теоретична парадигма, както и подпомагането при паралелно създаване на приложни лингвистични ресурси е изключително полезно. На базата на един съществуващ солиден лингвистичен ресурс повторението на процедурата по анализ и формализиране на езиковите феномени е значително улеснена. Това дава възможност за обръщане на специално внимание на разликите в естествените езикови системи и фина настройка на специфичните явления за всеки добавен език.

За настоящата дисертация като теоретична основа на КГТ на български език е избрана парадигмата на Системично-функционалната лингвистика, тъй като тя е една от основните най-успешни постановки за КГТ. Системично-функционалната лингвистика предлага изключителни възможности за паралели при системично-функционалните описания на различни естествени езици, по-бърза и по-ефективна формализация и реализация на лингвистични знания при създаване на система за КГТ за новоразглеждан език. Специфичното при системично-функционалния възглед за КГТ е организирането на целия процес на генериране около комуникационните цели, а не около граматическите структури на конкретния естествен език. На това се базира идеята за изграждане пространство на търсене от системични мрежи за генериране на български език, което да представлява ресурс наречен Приложна системично-функционалната граматика за български език. За реализиране на граматиката е необходимо моделиране на някои лингвистични феномени на българския език по отношение на Системично-функционалната лингвистика. На базата на теорията на [Halliday, 1994] системично-функционалните граматики са фокусирани върху кодиране на семантичните връзки в текста във функционална форма и са насочени към директни съответствия между по-високите нива на организация на текста, т.е. планиращите процеси, и граматическия компонент. Това е предпоставка за научноприложно изследване за генериране на няколко типа текстове-технически инструкции на български език. Достъпът до Средата за многоезиково генериране KPMIL подпомага работата по създаване, тестване и настройки на приложни системично-функционални граматики на различни естествени езици.

Както става ясно от направения обзор на областта, компютърното генериране на текст не може да бъде реализирано без семантични ресурси. Естествено за КГТ е използването на онтологии за предметната област на генерирането, а също и висши онтологични модели за лингвистичните знания, така че онтологичното инженерство е до голяма степен свързано с КГТ. Семантичното аотиране на текстове на естествен език има приложение в КГТ като практически значима

подпомагаща дейност. Семантичното аотиране представлява интерес особено за естествени езици различни от английски поради по-малкия брой изследвания и работещи реализации. Поради това изследователско направление в настоящата дисертация е представяне на знания в онтологични модели и семантично аотиране на специализирани текстове на български език.

В резултат на горните изводи, както и отчитайки насоките на изследователските колективи, в които е работил авторът, задачите на настоящата дисертация са формулирани така:

1. Създаване на формални описания на основни обекти на българския език в рамките на Системично- функционалната теория на Халидей с цел компютърно генериране на технически текстове.
2. Реализация на създадените формални описания във вид на компютърен ресурс в Средата за многоезиково генериране KPML.
3. Аprobация на разработения ресурс за компютърно генериране на кохерентни технически текстове на български език в избрана предметна област.
4. Разработване и реализация на схема за аотиране на специализирани текстове на български език на базата на съвременни семантични технологии.

В дисертацията са използвани съществено научните изследвания извършени с активното участие на автора по проектите AGILE и СИHУC.

AGILE: „Automatic Generation of Instructions in Languages of Eastern Europe” (Автоматично генериране на инструкции на три източно- европейски езика) е изследователски проект финансиран от Европейската комисия и реализиран по програмата *INCO-Copernicus* през 1998-2001г. от партниращи си организации от пет държави: Великобритания, Германия, Чехия, България и Русия.

СИHУC: „Семантични технологии за Интернет-услуги и технологично поддържано обучение” е изследователски проект № Д-002-189 финансиран от Националния фонд „Научни изследвания” през 2009-2012г. Резултатите от проекта са достъпни на адрес: sinus.iinf.bas.bg.

Глава 2. Подготовка на лексико-граматически ресурс за компютърно генериране на български текстове

Интерес към компютърното генериране на български език са проявявали изследователи от областта на компютърната лингвистика още от времето на първите по-сериозни приложни опити в обработването на естествен език. Идеята на Руслан Митков за генерираща система [Mitkov, 1990] е свързана с генериране на описания на основните геометрични термини на български език. По отношение на моделирането на българския език, особено през последните години има интересни разработки свързани с машинния превод, например [Ranta, Angelov, Hallgren, 2010]. Съществува опит в генерирането на изречения на български език от концептуални графи [Bontcheva and Angelova, 1996]. Генериране на текстове на български език, при това като един от естествените езици в паралелно многоезиково генериране, е реализирано по проекта AGILE (1998-2001), което ще бъде обсъдено тук по-подробно.

Проектът AGILE „Автоматично генериране на инструкции на три източно- европейски езика“ има за цел разработване на софтуерни инструменти, които да подпомагат работата на автори на технически ръководства. Избраната област за лингвистичната научно-приложна работа по проекта са CAD/CAM ръководствата на български, чешки и руски език. Реализиран е процес на автоматично генериране на кохерентни текстове, които представляват процедури-инструкции за ползване на CAD/CAM софтуер. Интересен от изследователска гледна точка е опитът да се генерират текстове на различни езици от едно и също концептуално представяне на съдържанието. Теоретична основа на реализираното многоезиково генериране е Системично-функционалната лингвистика. В последната фаза на процеса по генериране системата AGILE използва за лексико-граматическата реализация средата за многоезиково генериране KPML, в която са създадени ресурси за генериране на текстове на трите езика.

Настоящата глава от дисертационния труд се базира на извършената научно-изследователска работа за построяване на българския ресурс за системата AGILE, а именно Приложна българска системично-функционална граматика. В раздел 2.1 се описват теоретичните съображения за моделирането на елементи от системично- функционална граматика за български език, което подпомага създаването на приложния ресурс. Самата реализация на приложната българска системично-функционална граматика за целите на автоматичното генериране в системата AGILE е описана в раздел 2.2.

2.1 Моделиране на някои езикови явления в термините на Системично-функционалната граматика

В първата глава на настоящата дисертация споменахме, че за да се генерира автоматично отделно изречение на даден език е достатъчно генераторът да разполага само с нужната лексическа и граматическа информация за реализираните езиковите явления в това изречение. По тази логика е определен неголям корпус от целеви текстове по отношение на генерирането, за да се изследват специфичните езикови явления проявени конкретно в избрания тип текстове, а също да се фиксира нужният лексически материал за генериране в избраната област-инструкции за CAD/CAM софтуер.

Текстовият корпус съдържа за българската си част 9 процедури, 194 изречения (клаузи), 1219 думи. Следният процедурен текст представлява част от този корпус:

Създаване стил на мултилия

Първо отворете диалоговия прозорец с име *Multiline Styles*, като използвате един от следните методи:

Windows От функционалния ред с име *Object Properties* или менюто *Data* изберете *Multiline Style*.

DOS и UNIX От менюто *Data* изберете *Multiline Style*.

1. Изберете *Element Properties*, за да прибавите елементи към стила.
2. В диалоговия прозорец *Element Properties* въведете отместването на елемента на мултилията.
3. Изберете *Add*, за да добавите елемента.
4. Изберете *Color*. След това изберете цвета на елемента от диалоговия прозорец *Select Color*.
5. Изберете *Linetype*. След това изберете вида линия на елемента от диалоговия прозорец *Select Linetype*.
6. Повторете стъпки 2-5, за да дефинирате друг елемент.
7. Изберете *OK*, за да запишете характеристиките на елемента на мултилията и излезете от диалоговия прозорец *Element Properties*.

Извършен е детайлен анализ на текстовия корпус в три стъпки:

Първо, определяне на граматическите конструкции използвани в процедурите.

Второ, проучване на начините за описване на езиковите явления в термините на Системично-функционалната лингвистика.

Трето, разработване на формални спецификации за основните граматически функции.

Като резултат от проучването на CAD/CAM корпуса са определени следните функционални полета в езика, които да бъдат моделирани: транзитивност на процесите, диатезност, модалност, темпоралност, завършеност на процесите, системично-функционални конструкции при сложни изречения, а също някои езикови явления свързани със словореда, членуване, съгласуване и номинализация в групата на съществителното.

Тези функционални полета се намират на различни нива в организацията на граматиката, които нива са фиксирани от входната система за системично-функционалната мрежа. Входната система е наречена Ранг (Rank) и е изобразена по-долу. Според зададените нива от системата Ранг са организирани и подразделите в изложението показващи моделиране на прости и сложни изречения.

Разполагайки с дадения обем моделирани езикови явления работата нататък е организирана в посока създаване на принципно построено множество от ресурси за описание на една цялостна граматика, като това множество представлява детайлизация, работеща в избраната предметна област. Такъв подход е възможен, защото на естествения език и неговата граматика се гледа като на интегрална система и моделирането понякога обхваща елементи, които може да не присъстват в текстовия корпус, но са свързани с граматическите системи описващи реализирани граматически конструкции от областта.

Системично-функционалната граматика като системична мрежа дава възможност да се определят регионите, в които се сглобяват типичните конструкции на предметната област и в същото време да се моделират така, че да бъдат пре-използваеми за всеки друг процес на генериране, тъй като отразяват принципен граматически модел. Да се развиват обаче изчерпателно всички клонове на достигнатите граматически системи е подход, който отнема

много време и затова в рамките на проекта AGILE не се цели построяване на цялостни приложни граматика. Това е причината предложеното тук моделиране на езиковите явления да е частично, но обхващащо всички регистрирани в текстовия корпус явления, за да бъде възможно автоматичното генериране на текст на български език.

Мета-функциите на Системично-функционалната граматика изложени в обзора (Таблица 1, раздел 1.2.3) експериментна, интерперсонална, текстова, важат и за българския език, тъй като те са дефинирани на високо ниво като се твърди, че са валидни за всички естествени езици. Изречението в българската реч има същия статут, както изречението в английския език, например, и заема същата „гранична позиция” между по-малките единици на граматическия анализ, групите и фразите, и по-големите речеви отрязъци. В стила на системично-функционалния анализ изречението има трипластова структура, по един слой за всяка от мета-функциите. В Таблица 3 по-долу е показан анализ на изречението “*След това от менюто Data изберете Linestyle.*”.

Мета-функция	<i>След това</i>	<i>от менюто Data</i>	<i>изберете</i>	<i>Linestyle.</i>
Експериментна/ Идейна		Място:абстрактно	Процес	Цел
Интер- персонална	Адюнкт	Адюнкт	Финитив: императив	Допълнение
Текстова	Тема: свързваща	Тема:маркирана	Рема	

Таблица 3 Системично-функционален анализ на изречение на български език

В [Стайкова и Пенчев, 2000] е направена първата малка стъпка за въвеждане на терминологията за системично-функционален анализ на български език, но тук ще се придържаме към утвърдената английска терминология за описване на езиковите явления като системични мрежи. Както беше споменато и в обзора (раздел 1.2.3), теорията на системичните мрежи представя естествения език като ресурс за създаване на смисъл. Всяка система в системичната мрежа дава избор да се изрази един или друг смисъл на разглеждания в системата аспект. Системата се състои от (1) входно условие, което показва къде се прави избора, (2) множество от възможни изходи, и (3) реализации, които показват какви са структурните следствия в езика за всеки от възможните изходи на системата. Като пример за записване на система нека вземем входа към дадена граматиката, който представлява система определяща какъв по ранг е описваният езиков елемент.

RANK:

(start) →

[clauses],

[groups-phrases],

[words] (+Stem) (Stem:Morphemes),

[morphemes] (+Head).

Изобразената система има 4 възможни изхода и в зависимост от направения избор се описва или генерира изречение, група или фраза, дума или морфема. В случаите на дума или морфема като следствие от направения избор се въвежда нов функционален елемент. Коренът/ Stem на думата има под-тип Морфема/ Morphemes.

В следващата Таблица 4 са изобразени знаците, които се използват за означаване на различните елементи на системите нататък в текста.

Обозначен елемент	Пример	Коментар
Функционални елементи	Actor, Subject	наименования на конституенти използвани в системите
Наименование на система	MOOD	Класификации на характеристики според парадигмата
Граматически характеристики	[характеристика]	Характеристики според които се класифицират функционалните елементи
Елементи на селекцията:		
прецизиране	[feature-x : feature-y,...]	Отразява релацията тип-This captures the type-подтип
Симултантност	[feature-x & feature-y,...]	Избор на няколко характеристики
Реализационни изрази:		Изрази ограничаващи начините на проявяване на характеристиката в естествения език
вмъкване	+Subject	Вмъкване на конституент
съвпадение	Subject/Actor	Сливане на два конституента в един
разширение	Mood(Finite)	разширение
подредба	Subject ^ Finite	Нареждане на конституентите непосредствено един до друг
Предварителна селекция	Subject:nominal-group	Селектиране на избор от друго ниво
Лексически ограничения:		
класификация	Process::doing-verb	Ограничения за класификацията на функционален елемент
инфлексия	Noun::singular	Ограничения за инфлексните характеристики на функционалния елемент
лексика	Subordinator ! in-order-to	Избор на лексически елемент реализиращ функционалния елемент
съгласуване	Thing ~ Quality	Разпространяване на характеристики от един функционален елемент към друг

Таблица 4 Обозначения на елементите на системите

2.1.1 Ниво изречение

Трите мета-функции водят до три различни интерпретации на изречението, както е показано в обзора (раздел 1.2.3): изречението като представяне, изречението като обмен и изречението като съобщение. За всяко от регистрираните в корпуса езикови явления на нивото на изречението е отделено внимание като заглавието на под-раздела показва на коя мета-функция принадлежи съответното функционално поле.

2.1.1.1 Изречението като представяне. Процеси и участници. Типове процеси в текстовия корпус

Според Системично-функционалната методология с транзитивността са свързани три компонента:

1. Самият процес
2. Участниците в процеса
3. обстоятелствата свързани с процеса.

В Таблица 2 (раздел 1.2.3) са посочени типовете процеси и ключовите участници според Системично- функционалната лингвистика. Основните типове процеси са четири: [материален], [ментален], [релационен] и [словесен], като те се реализират структурно чрез участниците в тях: Актор-Цел за материалния процес, Усещаш- Феномен за менталния процес, Носител-Атрибут за релационния процес, Казващ-Идея за словесния процес. Други типични участници в материалните процеси са Агент, Бенефициент, Адресат. обстоятелствата са частично независими от типа на процеса, те включват функциите Съпровод, Причина, Начин, Роля и други и са описани в региона обстоятелства на граматиката.

В разглеждания текстов корпус се срещат два типа процеси: материални и релационни. Пример за реализация на материален процес е изречението: *Натиснете бутона ОК.*, а за релационен процес, например, изречението: *Полилинията е съставена от отсечки и дъги.* Разглеждаме първо материалните процеси, а след това – релационните. Така изглежда граматическата система, която прави избор между различните типове процеси:

PROCESS-TYPE:

(transitivity-unit) →

[material]	(Process::do-verb),
[mental]	(Process::experience-verb),
[verbal]	(Process::symbolic-verb),
[relational]	(Process::relational-verb).

Най-многобройни са срещанията на материални процеси. От системата за избор на типа процес е видно, че глаголите, които реализират материални процеси трябва да имат характеристика „do-verb“. В текстовия корпус можем да различим две групи глаголи с такава характеристика: ефективни глаголи/ effective-verbs, които се характеризират с двама участници в процеса (*някой натиска клавиш, някой добавя точка, някой начертава отсечка*) и средни глаголи/ middle-verbs, които имат само един участник (*Плаващото меню се отваря. Отсечката се появява на екрана.*). Изборът за тази характеристика се прави в системата AGENCY.

AGENCY:

(transitivity-unit) →

[middle]	(Process::middle-verb),
[effective]	(Process::effective-verb).

Участници в процесите, които моделираме имат своите функционални роли в зависимост от типа на процеса. Така за ефективните материални процеси участниците са Актор и Цел, а при средните процеси единственият участник получава функцията Медиум/ Medium.

В системично-функционалния анализ на английския език са въведени функциите Бенефициент/Beneficiary и Област/Range, които обозначават допълнителни участници в

процеса. Ако се има предвид материален процес, където Актора е логически субект, а Цел е логически пряк обект на процеса, то Бенефициентът/Beneficiary е наречен „логически непряк обект“, а Областта/Range е „логически сроден обект“ в процеса [Halliday, 1994, стр.144]. Като процес с Бенефициент в текстовия корпус е идентифициран процеса „добавяне“ (*за да добавите още елементи към мултилинията*).

В текстовия корпус са идентифицирани и релационни процеси, например „състои се от“ (*Полилинията се състои от отсечки и дъги.*). Този процес е класифициран като Приписващ_атрибути/Ascriptive. Следващата система TYPE-OF-BEING дава представа и за алтернативните релационни процеси.

TYPE-OF-BEING:

(relational) →

[identifying] (+Token) (+Value) (+Identified) (+Identifier)

(Identified/Subject) (Identifier/Directcomplement),

[existential] (+Existent) (Existent:Nominal-Group),

[ascriptive] (+Attribute) (Process::middle-verb)

[equative] (+Token) (+Value)

(Agent/Token) (Medium/Value) (Process::effective-verb).

Към езиковите явления свързани с транзитивността спадат и фазите при изразяване на процесите (например, *Започнете да чертаете.*). В текстовия корпус има срещания на фазовите глаголи *започна, завърша, продължа, приключа*, чиито конструкции се разпознават от следните граматически системи:

PHASE:

(transitivity-unit) →

[not-phase],

[phase](+Phase::phase-verb)

(+Phasedependent)

(Phase/Voice)

(Phasedependent/Lexverb).

PHASE-TYPE:

(phase) →

[beginning],

[repeating],

[ongoing],

[ending].

PHASEDEPENDENT-TYPE:

{beginning, repeating, ongoing, ending} →

[phase-infinitive] (Phasedependent:::present-form)

(+Da-phase)

(Da-phase ! Da)

(Da-phase^Phasedependent),

[nominal-group-phase] (Lexverb:Nominal-Group)

2.1.1.2 Изречението като представяне. Обстоятелства на процесите

Транзитивността свързана с обстоятелствата около даден процес (Адюнкти/Adjuncts) е наречена „транзитивност на обстоятелствата“. Системично-функционалната литература [Halliday, 1994, стр.151] предлага различни функции за обстоятелствата около процеса: Степен/Extent Място/Location, Начин/Manner, Причина/Cause, Условност/Contingency, Съпровод/Accompaniment, Роля/Role, Същност/Matter, Гледна точка/Angle. Както в английското, така и в българското изречение обстоятелствата могат да бъдат реализирани с предложни фрази или допълнения. Записано в граматическата система за вида на обстоятелствата това изглежда така:

TYPE-OF-CIRCUMSTANCE:

(transitivity-unit) →

[nonaccompaniment]
[accompaniment] (+Accompaniment)
(Accompaniment:prepositional-phrase)
(Accompaniment:accompaniment-process)
[noncause]
[cause] (+Cause)
[nonmanner]
[manner] (+Manner)
[nonmatter]
[matter] (+Matter) (Matter:matter-process),
[nonrole],
[role] (+Role) (Role:role-process),
[no-spatial-extent],
[spatial-extent] (+Spaceextent)
(Spaceextent:prepositional-phrase)
(Spaceextent:extent-process)
[no-spatial-location]
[spatial-location] (+Spacelocative)
[no-temporal-extent]
[temporal-extent] (+Timeextent)
(Timeextent:prepositional-phrase)
(Timeextent:extent-process)
[no-temporal-location]
[temporal-location] (+Timelocative)

Примерите за обстоятелства от текстовия корпус са най-често в материални процеси, към които е асоциирано обстоятелствено пояснение за място. Отбелязани са също срещания на изрази за релацията част-цяло (*крайна точка на дъгата*) и на изрази за релацията за съотнасяне (*Въведете ъгъл спрямо допирателната*.)

Типичните срещания на обстоятелствените пояснения за място (*до подсказващото съобщение, на функционалния ред, в диалоговия прозорец*) са класифицирани под [spatial-extent] като в изречението се добавя конституент за развиване на предложната фраза. Релацията за съотнасяне е класифицирана като [accompaniment] и тогава в изречението има конституент за предложната фраза изразяваща тази релация.

2.1.1.3 Изречението като представяне. Диатезност

Според системично-функционалния анализ ядро на функционално-семантичното поле Диатезност (Diathesis) е категорията залог. Залогът може да бъде описан като връзка между транзитивните функции изразени чрез ролите на участниците в процеса (виж раздел 2.1.1.1 по-горе) и функциите за дейтелността в процеса: Агент/Agent и Медиум/Medium [Halliday, 1994, стр.161].

Ако вземем за пример материалните процеси в английския език, то в деятелен залог трите функции Агент/Agent, Актор/Actor и Подлог/Subject съвпадат, както и трите функции Цел/Goal, Медиум/Medium и Пряко допълнение/Direct complement. В страдателен залог Медиумът/Medium изпълнява ролята на подлог в изречението и се групира с функцията Подлог/Subject, като функцията Агент/Agent се реализира с предложна фраза започваща с предлога *by*. Същата схема може да бъде идентифицирана и в българския език, както е показано в следващите примери.

<i>Потребителят</i>	<i>задава</i>	<i>точка.</i>
Агент/Agent	Процес:материален	Цел/Goal
Актор/Actor	Финитив	Медиум/Medium
Подлог/Subject	Залог: деятелен	Пряко допълнение/Direct complement

<i>Точката</i>	<i>е зададена</i>	<i>от потребителя.</i>
Цел/Goal	Процес:материален	Агент/Agent
Медиум/Medium	Финитив	Актор/Actor
Подлог/Subject	Залог: страдателен	

Освен чрез морфологичната категория за залог пасивност може да бъде изразена и чрез “средна конструкция”:

<i>Точката</i>	<i>се задава</i>	<i>от потребителя.</i>
Цел/Goal	Процес:материален	Агент/Agent
Медиум/Medium	Финитив	Актор/Actor
Подлог/Subject	Залог: страдателен	

Пасивната реализация на Агента с предложната фраза започваща с предлога *от* може да бъде пропусната (*Точката е зададена. Точката се задава. Задава се точка.*) Този анализ важи за т.н. ефективни материални процеси, чиито срещания са най-многобройни в текстовия корпус: въвеждам, задавам, прибавям, натискам....

В случая със средните процеси, като например: *Отваря се диалоговият прозорец.* или *Точката се появява на екрана.* имаме деятелен залог с единствен участник в процеса:

<i>Точката</i>	<i>се появява</i>	<i>на екрана.</i>
Агент/Agent		Обстоятелство
Медиум/Medium	Финитив	
Подлог/Subject	Залог: деятелен	

2.1.1.4 Изречението като обмен. Модалност

Модалността има най-характерно проявление на нивото на изречението в императивното или индикативно наклонение на глагола. В текстовия корпус броят на императивните и на индикативните изречения е почти изравнен заради стила на текстовете-инструкции, които го съставляват. В други текстове императивното наклонение на глаголите не би било толкова застъпено. Използвана е само учтивата заповедна форма на глагола, 2-ро лице, мн. число. В корпуса няма въпросителни изречения, нито реализации на условно наклонение.

Ключовата система е наречена MOOD-TYPE. Чрез нея изреченията се разделят на индикативни/ indicative и императивни/ imperative. В сравнение с английското императив, който не се променя по лице и число за българското императивно изречение глаголът се скланя, което е осигурено от добавянето на Финитива в системата.

MOOD-TYPE:

(independent-clause-simplex) →
[indicative],
[imperative] (+Finite).

Индикативни изречения. В българския език индикативните изречения се разделят на две групи: въпросителни или интеррогативни, /interrogative и декларативни/ declarative:

INDICATIVE-TYPE:

(indicative) →
[declarative],
[interrogative].

Декларативните изречения преобладават в текстовете, които са в т.н. безличен изявителен стил, като в показания отрязък от примерен текст:

Създаване стил на мултилия

Отначало се отваря диалоговият прозорец Multiline Styles, чрез използване на един от следните методи:

Windows: *От функционалния ред с име Object Properties или менюто Data се избира Multiline Style.*

DOS и UNIX: *От менюто Data се избира Multiline Style.*

- 1. Избира се Element Properties, за да се прибави елемент към стила.*
- 2. В диалоговия прозорец Element Properties се въвежда отместването на елемента на мултилията.*

При този стил всички изречения са декларативни и използват пасивна конструкция. Изреченията изразяващи странични ефекти също са декларативни. Това са изречения от вида:

Диалоговият прозорец Select Color се появява на екрана.

Главна система за индикативни изречения в системичната мрежа е системата наречена FINITE-CLAUSE, която въвежда елемента на модуса / Mood element. Реализацията Mood (Finite) обозначена в системата е получена чрез разширяване (виж Таблица 4 по-горе) и така е изградена структурата на индикативното изречение с конструкт модус / Mood, който съдържа Finite. Системата FINITE-CLAUSE е показана по-долу.

FINITE-CLAUSE:

(indirect-indicative; finite-enhancing; finite-elaborating;
finite-extending; finite-rankshift; indicative) →
[Finite-clause]
(+Mood, Mood(Finite)).

По-нататъшното развиване на елемента на модуса изисква въвеждане на конституенти за подлога/ Subjectи сказуемото/ Finite.

FINITE-INSERT:

(finite-clause; finite-in-imperative) →
[Finite-inserted] (+Finite).

Избора на граматическото лице за сказуемото става чрез системата INDICATIVE-MOOD-PERSON.

INDICATIVE-MOOD-PERSON:

(finite-clause,
[mental, verbal, creative, dispositive, identifying, relational-other,
(material, nonmeteorological)]) →
[indicative-interactant],
[indicative-noninteractant] (Finite::Thirdperson-form).

По-нататъшното специфициране на сказуемото/ Finite се прехвърля в други полета на системично-функционалната граматика като залог, граматическо време и т.н. Присъствието или отсъствието на подлог се балансира от логиката на три системи INDICATIVE-SUBJECT-PRESUMPTION, EXPLICIT-DECLARATIVE-SUBJECT и SUBJECT-INSERT.

INDICATIVE-SUBJECT-PRESUMPTION:

([(declarative, speaker-subject)
(interrogative, addressee-subject)]) →
[subject-explicit],
[subject-implicit].

Всички декларативни изречения от целевите текстове имат явно зададен подлог, следователно интересната характеристика за нас е subject-explicit.

EXPLICIT-DECLARATIVE-SUBJECT:

(declarative,
[addressee-subject; indicative-noninteractant;
metathing-subject; speaker-plus-subject]) →
[explicit-declarative-subject].

SUBJECT-INSERT:

([wh-subject; meteorological; existential;
indicative-noninteractant; metathing-subject;
subject-specified; explicit-subject-elaborating;
indirect-indicative; finite-enhancing; finite-extending;
finite-rankshift; explicit-subject;
explicit-declarative-subject; explicit-interrogative-subject;
imperative-other; imperative-subject-explicit; subject-explicit;
(imperative [oblativ; suggestive])])→
[subject-inserted] (+Subject)

В българския език подлогът и сказуемостта се съгласуват по лице и число като в зависимост от избора на граматическо време може да са съгласувани и по род, както, например в изречението:

<i>Полилинията</i>	<i>е</i>	<i>съставена</i>	<i>от отсечки и дъги.</i>
<i>Зл.,ед.ч.,ж.р.</i>	<i>Зл.,ед.ч.</i>	<i>ж.р.</i>	

Императивни изречения. Типично императивно изречение от текстовия корпус е, например, изречението:

Задайте началната точка на отсечката.

При системично-функционалния анализ се различават три вида императивни изречения: заповедни/ jussive, подканващи/ suggestive, задължаващи/ oblativ. Текстовият корпус съдържа само заповедни форми. Системата IMPERATIVE-INTERACTANT-SUBJECT е отговорна за избора между трите вида императивни изречения. В случая със заповедния изход глаголът е винаги във второ лице (secondperson-form) и в императивна форма (imperative-form).

IMPERATIVE-INTERACTANT-SUBJECT:

(imperative-interactant)→
[jussive]
(Finite:::secondperson-form, Finite:::imperative-form),
[suggestive]
(+Conjunct, Finite:::indicative-form),
[oblativ]
(+Conjunct, Finite:::indicative-form).

За уточняване дали се използва учтив форма на заповедното обръщение е необходимо да добавим системата IMPERATIVE-POLITENESS-TYPE. Тя има два изхода за лична императивна форма, която се реализира с единствено число и за учтив императивна форма, която се реализира с множествено число на глагола.

IMPERATIVE-POLITENESS-TYPE:

(jussive) →

[polite-imperative] (Finite::plural-form),
[personal-imperative] (Finite::singular-form).

Пропускането на подлога при императивните изречения в българския език се реализира чрез следната система:

IMPERATIVE-SUBJECT-PRESUMPTION:

(imperative) →

[imperative-subject-implicit],
[imperative-subject-explicit] (Subject : Addressee).

2.1.1.5 Изречението като обмен. Темпоралност

В съвременното езикознание се приема, че българският език притежава девет глаголни времена: сегашно, минало свършено, минало несвършено, минало неопределено, минало предварително, бъдеще, бъдеще в миналото, бъдеще предварително и бъдеще предварително в миналото [Бояджиев, Куцаров, Пенчев, 1999г.]. От тях само две се срещат в изследвания текстов корпус-сегашно време и минало неопределено. Срещанията на сегашно време са най-многобройни, като обикновено те са в описателните изречения за ефекта от дадени действия (*Щриховката се появява в посочената фигура.*) или в сложни изречения като дадените по-долу:

Линиите пресичат края на цялата мултилия, а външните дъги съединяват краищата на външните елементи.

Стартирайте командата LINE, като използвате един от следните методи:

Минало неопределено време се среща само 3 пъти в корпуса, като например в следното изречение:

След като вече сте записали линиите, начертани на ръка, можете да ги редактирате.

Системата, в която се определя граматическото време в системичната граматика е наречена TENSE-SYSTEM. Тя има три изхода, съответно за сегашно, минало и бъдеще време. Тъй като минало неопределено време не е типично за текстовете- инструкции не сме си поставяли за цел да моделираме и генерираме такива изречения, така че сме работили само със сегашно време. При изхода [present] е отбелязано, че Финитивът е в морфологичната форма на глагола за сегашно време.

TENSE-SYSTEM:

(clause-simplex) →

[past],
[future],
[present] (Finite::present-form).

Разгръщането на системичната мрежа за граматическото време започва от системата DEICTICITY представена по-долу. Входно условие е изречението да има Финитив, а изходите

са два: да се изрази модалност или да се изрази граматическо време чрез Финитива. Вариациите на наклонението в изследвания текстов корпус водят само през изхода за граматическо време, което е детайлизирано в следващия раздел Темпоралност.

DEICTICITY:

(finite-clause) →

[modal] (Finite::modal-aux),

[temporal] (+TemporalUnit, TemporalUnit/Finite).

Условието за преминаване през тази система е изпълнено за следните типове изречения от текстовия корпус:

1. Просто изречение- страничен ефект

Появява се диалоговият прозорец Multiline Style.

Характерен за тези случаи е несвършения вид на използваните глаголи.

2. Просто изречение в безличен стил

Задава се крайната точка на дъгата.

В такива изречения глаголите са в сегашно време, 3-то лице.

3. Подчинени изречения в някои сложни конструкции

Стартирайте командата ARC, като използвате един от следните методи:

Стартира се командата ARC, като се използва един от следните методи:

Използваното време в подчинените изречения е сегашно време. Повече за анализа на сложните изречения предлага изложението в раздел 2.1.3.

2.1.1.6 Изречението като обмен. Завършеност на процеса

Видът на глагола е лексикално-граматична категория характерна за славянските езици, в това число и за българския език. Всъщност, както е споменато в [Бояджиев, Куцаров, Пенчев, 1999], „...спорен е въпросът за категориалната характеристика на глаголния вид, който на практика се свежда до това, дали трябва да се приемат видови форми или глаголи от свършен и от несвършен вид.“ Авторите правят обзор на становищата на специалисти българисти и приемат, че видът на глагола е лексикално-граматична характеристика, като един от аргументите е, че по традиция в българската граматична литература се говори за глаголи от свършен и от несвършен вид.

В Системично-функционалната теория видът на глагола е функция на Предикатора, който от своя страна е част от т.н. Остатък (виж раздел 1.2.3) заедно с вторичното време, залога и типа на процеса. Важно е да се отбележи, че най-популярната лингвистика на Халидей не отделя задълбочено внимание на изучаването на глаголния вид, най-вероятно заради природата на основните езикови системи, на които е базирана.

За целите на нашата работа приемаме следния модел: Всеки глагол принадлежи на поне една от двете категории: свършен_вид/ perfective_verb и несвършен_вид/ imperfective_verb. Най-общо различаваме смисъла на двете категории по следния начин:

Несвършеният вид на глагола подчертава следните семантични характеристики в описанието на действието:

-продължителна, непрекъсваща природа на действието,

-незавършеност на действието,

- повтаряема, циклична, обичайна природа на действието,
- обща твърдения, общоприети истини.

Свършеният вид на глагола има следните характеристики:

- отнася се за действието като цяло,
- концентрира се върху завършеността на действието.

В бинарната категория вид на глагола несвършеният вид е немаркиран заради обхвата на използване и честотата на срещанията.

По отношение на използването на различни по вид глаголи в изследвания корпус може да се каже, че както несвършения вид, така и свършения вид се наблюдават в разглежданите текстове. Вида на глагола не се влияе от преобладаващото използване на повелително наклонение.

ASPECT е наименование на новата система за моделиране на тази характеристика и се отнася към Предикатора, следователно, трябва да бъде добавена на нивото на изречението:

ASPECT:

(clause-simplex) →

[perfective] (Process::perfective-verb)

[imperfective] (Process::imperfective-verb)

2.1.2 Ниво сложно изречение

В системично-функционалната граматика на Халидей се поддържа представата, че на сложното изречение/ clause complex може да се гледа като на комплекс от изречения, точно както групата може да бъде възприемана като „комплекс от думи“. Подобно на ситуацията в сложните групи, може да бъде открито *главно изречение*, което се модифицира от останалите изречения.

Системично-функционалната граматика отива и по-нататък с принципите си, че създаването на сложни изречения / clausecomplexity ни дава възможност да покрием целия спектър от възможни функционални организации за изреченията. Всъщност, Халидей дефинира изречението като комплекс от клаузи/ clause complex и този комплекс е единствената граматична единица разпознавана над изречението [Halliday, 1994, стр. 216].

Сложността в организацията на сложните изречения произтича от различните начини, по които простите изречения могат да бъдат свързвани. Връзките между изреченията се интерпретират в СФГ в смисъла на „логически компонент“ на лингвистичната система. Различават се две измерения, за да се предложат по-детайлни описания за това как точно простите изречения модифицират главното изречение.

Първото измерение се занимава със системата на взаимозависимост, дали е паратактична (съчинение) или хипотактична (подчинение).

Другото измерение е логико-семантично и се занимава с експанзията и проекцията. Те са разгледани по-подробно нататък в настоящия раздел като се подчертава идеята, че сложните изречения възникват като резултат от взаимодействието на двете измерения.

Хипотаксис и паратаксис

Ключова тук е системата *таксис* или *взаимозависимост*. Основна идея на таксиса е да изясни статута на връзката между главното изречение и модифициращите изречения. Както е обяснено в [Halliday, 1994, стр. 218], хипотаксис е връзката между зависим елемент и елемент, който го доминира, докато паратаксис е връзка между два елемента, като нито един не е зависим от

другия. Постулат на системично-функционалната граматика е твърдението, че всички „логически“ връзки в езика представляват или хипотаксис или паратаксис.

Всяка двойка изречения може да се смята за свързаност от изречения (clause nexus), при която има релация между първичното изречение и вторичното изречение. Таблицата по-долу показва как тези термини се отнасят към таксиса.

	Първично изречение	Вторично изречение
Паратаксис	1 (иницииращо)	2 (продължаващо)
Хипотаксис	α (доминиращо)	β (зависимо)

Таблица 5 Свързаност от изречения (nexus): тактическо измерение

Заради неравностойния статут на участниците в хипотактична връзка, релацията е асиметрична и не-транзитивна от логическа гледна точка. По естествен начин този белег може да бъде модифициран от естеството на логико-семантичните отношения описани по-нататък, но обикновено семантичните свойства са подредени колкото е възможно по-добре.

Елементите на хипотактичната структура са подредени според зависимостта си, но този ред е доста независим от линейността на изречението. Така може да имаме зависимо изречение (а) следващо доминиращото изречение, (б) предхождащо доминиращото изречение, (в) включено в доминиращото изречение или (г) включващо доминиращото изречение.

(а)	Файлът не е съхранен	докато не го запазите.		
	α	β		$\alpha^{\wedge}\beta$
(б)	Когато запазите файла,	той ще бъде съхранен.		
	β	α		$\beta^{\wedge}\alpha$
(в)	Запазете,	ако искате,	файла.	
	α	β	α	$\alpha \llbracket \beta \rrbracket$
(г)	Вие можете,	според указаното в ръководството,	да запазите файла.	
	β	α	β	$\beta \llbracket \alpha \rrbracket$

При паратаксис първичният и вторичният елемент на свързаността от изречения / clause nexus са с еднакъв статут. Както инициращият (първичен) елемент, така и продължаващият (вторичен) елемент могат да бъдат самостоятелно функциониращи единици, докато в случай на хипотаксис зависимият елемент не може да съществува без връзката си с доминиращия елемент. Тази относителна автономност може да бъде наблюдавана в логическите характеристики на паратактичните връзки, които са симетрични и транзитивни. Поради липсата на зависимост между инициращите и продължаващите изречения, при тях не съществува наредба различна от линейността на изречението, което контрастира със случая на хипотаксис.

Логико-семантични връзки

Вече беше отбелязано, че релациите между простите изречения в сложното изречение се характеризират и с логико-семантични връзки. СФГ поддържа класификация на логико-семантични връзки, която ги категоризира чрез малко на брой абстрактни типове, за да бъде

възможно да се обхванат многобройните вариации на отношенията между първичните и вторичните изречения в „свързаностите от изречения“.

Класификацията е основана на два фундаментални класа: експанзия и проекция. Понятието за експанзия изразява идеята, че изречението с неговата лингвистична функция като процес, обмен и съобщение може да влезе в конструкция с друго изречение, което го уголемява. Тогава двете изречения образуват сложно изречение. Понятието за проекция, от друга страна, обозначава тези конструкции, в които второто изречение е проектирано през първичното изречение. Първичното изречение въвежда второто като идиом или идея.

Тъй като корпусът на Agile не съдържа сложни изречения, които да класифицираме като проекция, то се ограничаваме нататък с разглеждане само на експанзията при логико-семантичните релации. Вторичните изречения могат да уголемяват първичните чрез допълване / *elaboration*, разширяване / *extension* или подчертаване / *enhancement*.

Експанзия- допълване

При експанзия- допълване вторичното изречение разширява значението на първичното изречение като го специфицира или описва в по-големи детайли. Вторичното изречение не въвежда нов референт в дискурса, а по- скоро предлага по- подробно описание на вече представения референт. Следователно, не е съществено, че вторичното изречение е логически свързано с първичното изречение като цялост, то може да пояснява само някаква част от него, някой от конституентите му.

В комбинация с паратаксис експанзия- допълване създава следните три типа:

1. **Експозиция/ Exposition** – «с други думи»
2. **Даване на примери/ Exemplification** – «например»
3. **Изясняване/ Clarification** – «по-точно»

При експозицията вторичното изречение представя гледната точка изразена в първичното изречение по друг начин, с други думи или подсилва първичното изречение. При даването на примери вторичното изречение изработва тезис на първичното изречение като изрежда различни примери. При изясняването вторичното изречение пояснява твърдението от първичното изречение със средствата на разяснителен коментар.

Комбинацията на експанзия- допълване и хипотаксис създава т.н. недефиниращо релативно изречение / *non-defining relative clause* в СФГ. Значението на вторичното изречение е неограничаващо или описателно по отношение на твърдението изразено в първичното изречение.

Интересно е, че дефиниращите релативни изречения, в контраст с недефиниращите, реализират включване / **embedding**. Включването е функционално различно от таксиса дефиниран в СФГ. „Докато паратаксис и хипотаксис са релации между изречения или елементи от друго ниво, то това не може да се каже за включването. Включването е механизъм, при които изречението или фразата започва да функционира като конституент вътре в структурата на група, която сама по себе си е конституент на изречение. Следователно, няма директна връзка между включеното изречение и изречението, в което то е включено [Halliday 1994, стр.242].

Експанзия- разширение

При експанзия- разширение вторичното изречение добавя нещо ново към съдържанието изразено в първичното изречение, обратно на разгледаната експанзия- допълнение. Добавеното

може да бъде или просто допълване или вариация. Това са двете разновидности на експанзия-разширение [Halliday 1994, стр.230].

Паратактичното разширение е известно също като *координация* между простите изречения, обикновено се изразява чрез съюзите *и*, *или*, *а*. Първият подтип означава добавяне. Тогава единият процес е просто присъединен към другия без никакви по-нататъшни причинно-следствени или времеви връзки между тях. Вторият подтип на паратактичното разширение включва вариации. При него изразеното значение на второто изречение продължава напред като частично или тотално замества тезата изразена от първичното изречение.

Категория		Значение
Добавяне	«и», добавяне: позитивно	X и Y
	«нито», добавяне: негативно	не X и не Y
	«а», противоположно	X и обратно Y
Вариация	«вместо», заместващо	не X а Y
	«освен», изваждащо	X но не всички X
	«или», алтернатива	X или Y

Таблица 6 Видове разширение при сложно изречение

И тук могат да се различат под-типове на разширението в зависимост от това дали релацията възниква като комбинация с паратаксис или с хипотаксис. При съвпадане с хипотаксис разширението води до добавяне, заместване и алтернатива, но в този случай вторичното изречение е зависимо, заради природата на хипотаксиса [Halliday 1994, стр.231].

Експанзия- подчертаване

При подчертаването едно изречение усилва смисъла на друго чрез квалифициране на второто изречение с референции за време, място, начин, причина или условие, т.е. *обстоятелствено* квалифициране. Таблица 7 предлага списък на основните видове експанзия- подчертаване [Halliday 1994, стр.234].

Експанзията- подчертаване също може да бъде комбинирана с паратаксис и хипотаксис. Паратактическото подчертаване води до вид координация, при която е инкорпорирана обстоятелствената характеристика. Подобно наблюдение се прави и при хипотактическото подчертаване, макар там да говорим за „изречения- обстоятелствени пояснения“ включващи описваните обстоятелства.

Категория		Значение
Времево	същото време различно време: по-късно различно време: по-рано	А докато В А след В А преди В
Пространствено	същото място	С където D
Начин	средство сравнение	N чрез M N е като M
Причинно-условно	причина: основание причина: цел условие: цел условие: отрицателно условие: отстъпително	поради Р резултат Q поради намерение Q действие Р ако Р то Q ако не Р то Q ако Р то обратно на очакванията Q

Таблица 7 Видове подчертаване при сложно изречение

Моделиране на сложни изречения

Системата СЛОЖНОСТ_НА_ИЗРЕЧЕНИЕ / CLAUSECOMPLEXITY има като изходи граматическите характеристики сложно_изречение / clause-complex и просто_изречение / clause-simplex. Първата характеристика служи след това за вход на системата ТИП_НА_ВЗАИМОЗАВИСИМОСТ /TYPE-OF-INTERDEPENDENCE, която води до по-нататъшната спецификация на видовете свързаност между изреченията в сложното изречение и е формално представена по следния начин:

TYPE-OF-INTERDEPENDENCE

(clause-complex) ->

[expansion],

[projection].

Тъй като в корпуса на AGILE няма срещания на проекция / projection, не се занимаваме нататък с този клон на системичната мрежа.

Чрез характеристиката експанзия / expansion се достига до две системи: за тип на експанзията EXPANSION-TYPE и за таксис на експанзията EXPANSION-TAXIS. Чрез разгърнатата нататък системична мрежа се реализира принципната постановка на СФГ, че сложността на модификациите на простите изречения в сложното изречение може да бъде обхваната чрез взаимодействието на таксиса и логико-семантичните връзки. Споменатите по-горе системи постигат точно това:

EXPANSION-TYPE:

(expansion) →

[enhancement] (Enhancement),
[extension] (Extension),
[elaboration] (Elaboration).

EXPANSION-TAXIS:

(expansion) →

[general-hypotactic-expansion],
[paratactic-expansion].

Измерението на таксиса има своето отражение нататък чрез следните системи:

PARATAXIS:

{paratactic-expansion, quoting} →

[parataxis] (+Initiating,
+Continuing,
Initiating...Continuing).

HYPOTAXIS-ALPHA-COMPLEXITY:

(general-hypotactic-expansion) →

[complex-alpha-hypotactic-expansion]
(+Dependent,
+Continuing,
+Terminant,
Terminant : full),
[hypotactic-expansion].

Чрез тези системи са дефинирани основните видове връзки на експанзия между изреченията в сложното изречение, а също така вида на тяхната взаимозависимост. За избраната релация на експанзия са въведени кореспондиращите й граматически функции в структурата на сложното изречение, например, *initiating* and *continuing* за паратаксис. Как се реализират конкретно споменатите две граматически функции в продукцията от генерирането е демонстрирано, например, от генерираната структура на сложното изречение показана на Фигура 22 от следващия раздел.

Граматическите характеристики са комбинирани според вида на сложното изречение, например, подчертаване/ *enhancement* заедно с паратактична експанзия / *paratactic-expansion* са входно условие на системата QUALIFYING-COORDINATION, или разширяване / *enhancement* заедно с основна_хипотактична_експанзия / *general-hypotactic-expansion*, които са входно условие за системата QUALIFYING-CONDITION:

QUALIFYING-COORDINATION:

(enhancement, paratactic-expansion) →

[qualifying-coordination]
(+Enhanced,
+Coordinator,
Coordinator^Continuing).

QUALIFYING-CONDITION:

(enhancement, general-hypotactic-expansion) →
[qualifying-condition]
(Dependent : dependent-clause,
Enhancement/Dependent).

EXTENDING-COORDINATION:

(extension, general-hypotactic-expansion) →
[extending-coordination]
(+Extended,
Initiating/Extended,
Continuing/Extension).

NON-DEFINING-RELATIVE:

(elaboration, general-hypotactic-expansion) →
[nondefining-relative]
(Elaboration : dependent-clause,
Elaboration/Dependent).

Определен е точният вид на сложното изречение. Тъй като са подробно класифицирани сложните изречения и съставлящите си прости изречения, това до известна степен влияе на определянето и на граматическите категории в тях.

Системи детайлизиращи експанзия-подчертаване:

QUALIFYING-CONDITION-TYPE:

(qualifying-condition) →
[concessive-condition]
(*Enhancement : concession-dependent),
[conditional-condition]
(*Enhancement : conditional-dependent),
[purposive-condition]
(*Enhancement : purpose-dependent),
[manner-condition]
(*Enhancement : manner-dependent),
[spatial-condition]
(*Enhancement : spatial-dependent),
[causal-condition]
(*Enhancement : causal-dependent),
[temporal-condition]
(*Enhancement : temporal-dependent).

QUALIFYING-COORDINATION-TYPE:

(qualifying-coordination) →

[causal-coordination]

(Enhancement/Initiating,

Enhanced/Continuing,

Coordinator ! So),

[temporal-coordination]

(Enhancement/Continuing,

Enhanced/Initiating,

Coordinator ! Then).

Системи детайлизиращи експанзия-разширение:

EXTENDING-HYPOTAXIS:

(extension, hypotactic-expansion) →

[extending-hypotaxis] (Dependent/Extension).

EXTENDING-COORDINATION-TYPE:

(extending-coordination) →

[additive-coordination]

(+Coordinator,

Coordinator^Extension,

Coordinator ! And),

[alternative-coordination]

(+Correlator,

Correlator^Extended,

Correlator ! Either,

+Coordinator,

Coordinator^Extension,

Coordinator ! Or),

[contrastive-coordination]

(+Coordinator,

Coordinator ! But,

Coordinator^Extension).

Във връзка с моделирането на транзитивността изложено в предишния раздел 2.1.1 важно е да се обърне внимание на системата TRANSITIVITY-UNIT дадена по-долу, като нейните алтернативни входни условия са hypotactic-extension или clause-simplex. Тя осигурява връзка с редица други системи, чрез които се детайлизират характеристките свързани с изречението като обмен (виж раздел 1.2.3).

TRANSITIVITY-UNIT:

{clause-simplex, hypotactic-expansion} →
[transitivity-unit]
(+Predicator,
+Process,
+Lexverb,
Lexverb/Process,
Predicator (Lexverb),
+Voice,
+Predicator (Voice)).

Накрая прилагаме Таблица 8, в която са изложени съюзите, които се срещат в обработения корпус, класифицирани според Системично-функционалната терминология.

Вид на сложното изречение	Съюз
подчертаване + хипотаксис: причина- цел	за, за да
подчертаване + хипотаксис: времево съвпадение	когато
подчертаване + хипотаксис: времево следване	след като, след като вече
подчертаване + хипотаксис: причина- условност	ако
допълване + хипотаксис: (неограничаващи релативни второстепенни изречения)	който, която, което, които
разширяване- вариация (алтернатива) + паратаксис	или
паратаксис	и, а, но

Таблица 8 Съюзи от текстовия корпус

В под-раздел 2.1 е разгледана системично-функционалната концепция за някои граматически явления в българския език. Това са явленията обхванати от анализирания корпус, съставен от текстове-инструкции, като прагматичната цел е да се направи моделиране на определените граматически явления, за да се обезпечи автоматично генериране на подобни текстове-инструкции. Използвана е парадигмата и формализмите на Системично-функционалната граматика на Халидей. Предложени са модели във вид на системични мрежи за следните граматически области от българския език:

Диатезност. Моделирани са деятелния и страдателния залог. Обърнато е внимание на средната страдателна конструкция.

Модалност. Моделирани са декларативни и императивни изречения.

Темпоралност. Констатирана е необходимост от моделиране само на сегашно време и е предложен системичен модел.

Завършеност на процеса. Предложен е системичен модел за свършен-несвършен вид на глагола в българския език.

Анализирани са сложните изречения в корпуса от текстове-инструкции. Обсъдени са системите изграждащи системична мрежа за моделиране на сложни изречения според особеностите на българския език. Заради семантичната абстракция при свързване на прости

изречения в паратактични или хипотактични комплекси, както и боравейки с логико-семантичното измерение, се твърди, че при съставянето на целевите сложни изречения в избрания контекст на текстове-инструкции няма специални случаи за езиковата система на българския език. Направен е изводът, че бихме могли до голяма степен да се възползваме и за българския език от разработената в СФГ теоретична системично-функционална мрежа.

2.2 Лексико-граматически ресурс за генериране на български текстове: Приложна българска системично-функционална граматика

2.2.1 Принципи на създаване

Приложната българска системично-функционална граматика е ресурс за генериране на български текстове създаден в средата за многоезиково генериране KPML като резултат от работата по проекта AGILE. В средата KPML при създаване на нови граматики се насърчава подход на сравнителен анализ с приложната английска граматика NIGEL или с подходящи приложни ресурси за други езици, например, немски или френски.

Изключително полезно се оказва сътрудничеството при сравнителния анализ на трите славянски езика- руски, български и чешки. Цел на изследователската работа е създаването на ресурси за трите славянски езика, които да бъдат пре-използваеми за генериране в различни предметни области. Изчерпателният анализ за всички възможни изходи на системичната мрежа отнема много време и труд, така че в рамките на проекта приложният резултат засяга основно идентифицираните функционални полета при анализа на конкретния текстов корпус. Този подход предлага възможност във всеки момент приложният ресурс да бъде детайлизиран и обогатен с нови възможности за генериране.

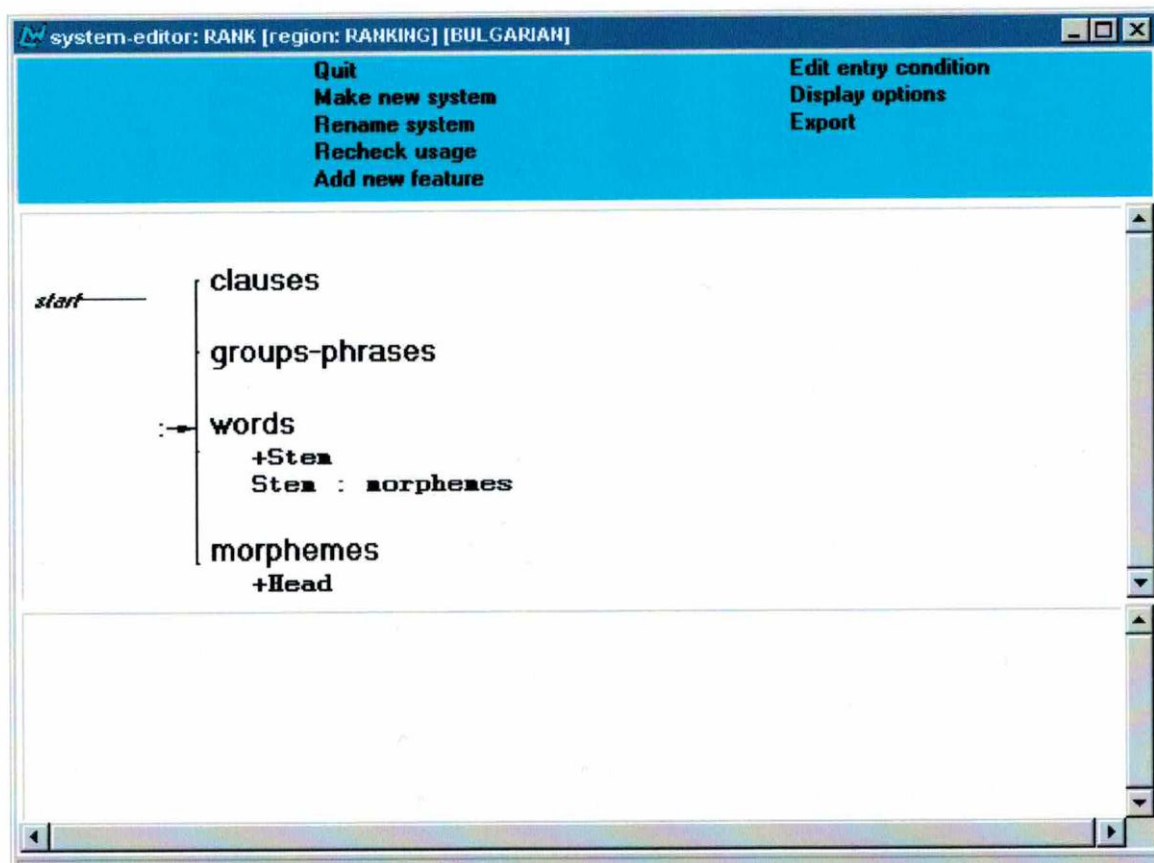
Преди проекта AGILE не са известни генериращи системи, които да предлагат пре-използване на ресурсите за генериране в нови приложни области, така че приложният подход към създаване на граматиките е от полза за цялата общност. Приложната българска системично-функционална граматика е достъпна в Банката по генериране¹⁵ поддържана за средата за многоезиково генериране KPML. Ресурсът притежава представително множество от генерирани примери, които дават представа за функционално-системичните полета, които са разработени.

Като изчислителен ресурс една приложна граматика в средата KPML представлява системична мрежа организирана в полета (regions) от свързани системи, като полетата са условно разделени и системите в едно поле имат отношение към едно и също граматическо явление. Начало на системичната мрежа е системата Ранг, в която се правят първите избори за подадения като вход семантичен израз.

Изборите се направляват от входните данни и от логиката за правене на автоматични избори, която се реализира чрез избирателите /choosers. Всяка система има свой избирател който насочва процеса на генериране през някой от изходите на системата. Понякога избирателите се нуждаят от специфична или динамична информация, за да направят избор и ползват свои проучватели/ inquiry, за доставяне на такава информация за текущия процес.

На следващата фигура е показана системата, от която обикновено започва процесът на генериране на текст в дадена приложна граматика на средата KPML.

¹⁵ <http://www.fb10.uni-bremen.de/anglistik/langpro/kpml/genbank/generation-bank.html>



Фигура 7 Визуализация на системата Ранг/RANK

```
(SYSTEM
:NAME    RANK
:INPUTS  START
:OUTPUTS ((0.2 CLAUSES)
          (0.2 GROUPS-PHRASES)
          (0.2 WORDS (INSERT STEM)
                     (PRESELECT STEM MORPHEMES))
          (0.2 MORPHEMES (INSERT HEAD)))
:CHOOSER RANK-CHOOSER
:REGION  RANKING
:METAFUNCTION LOGICAL)
```

Реализация на системата Ранг

Процесът е автоматизиран. Всяка система има Избирател за подходящия изход. Следва кода на Избирателя на системата RANK:

```

(CHOOSE
  :NAME RANK-CHOOSE
  :DEFINITION ((IDENTIFY WHERE-AM-I
    (WHERE-AM-I-ID)
    (IDENTIFY ONUS
      (MODIFICATION-SPECIFICATION-ID WHERE-AM-I ONUS))
    (IDENTIFY SPEAKER
      (SPEAKER-ID ONUS))
    (IDENTIFY HEARER
      (HEARER-ID ONUS))
    (ASK (EXIST-SPEECH-ACT-Q ONUS)
      (SPEECHACT (CHOOSE CLAUSES)
        (IDENTIFY SPEECHACT
          (SPEECH-ACT-ID ONUS))
        (IDENTIFY SPEECHACT
          (MODIFICATION-SPECIFICATION-ID WHERE-AM-I
            SPEECHACT))
        (IDENTIFY POLARITY
          (POLARITY-ID SPEECHACT))
        (IDENTIFY POLARITY
          (MODIFICATION-SPECIFICATION-ID WHERE-AM-I
            POLARITY))))
    (NOSPEECHACT
      (ASK (EXIST-SELECTED-STEM-Q ONUS)
        (NO-STEM (CHOOSE GROUPS-PHRASES))
        (STEM (COPYHUB ONUS STEM)
          (COPYHUB ONUS HEAD)
          (DEFAULTCHOOSE WORDS))))))

```

Реализация на избирател Ранг

Генерирането се инициира и направлява от системичните характеристики на очаквания генериран текст, зададени във входно описание –израз на езика за планиране на изречения Sentence Planning Language. Всички термини, които присъстват във входните изрази са дефинирани или в онтологията от високо ниво Обобщен модел (GUM) или в предметната онтология за конкретната приложна област. Следва примерен план на изречение (SPL).

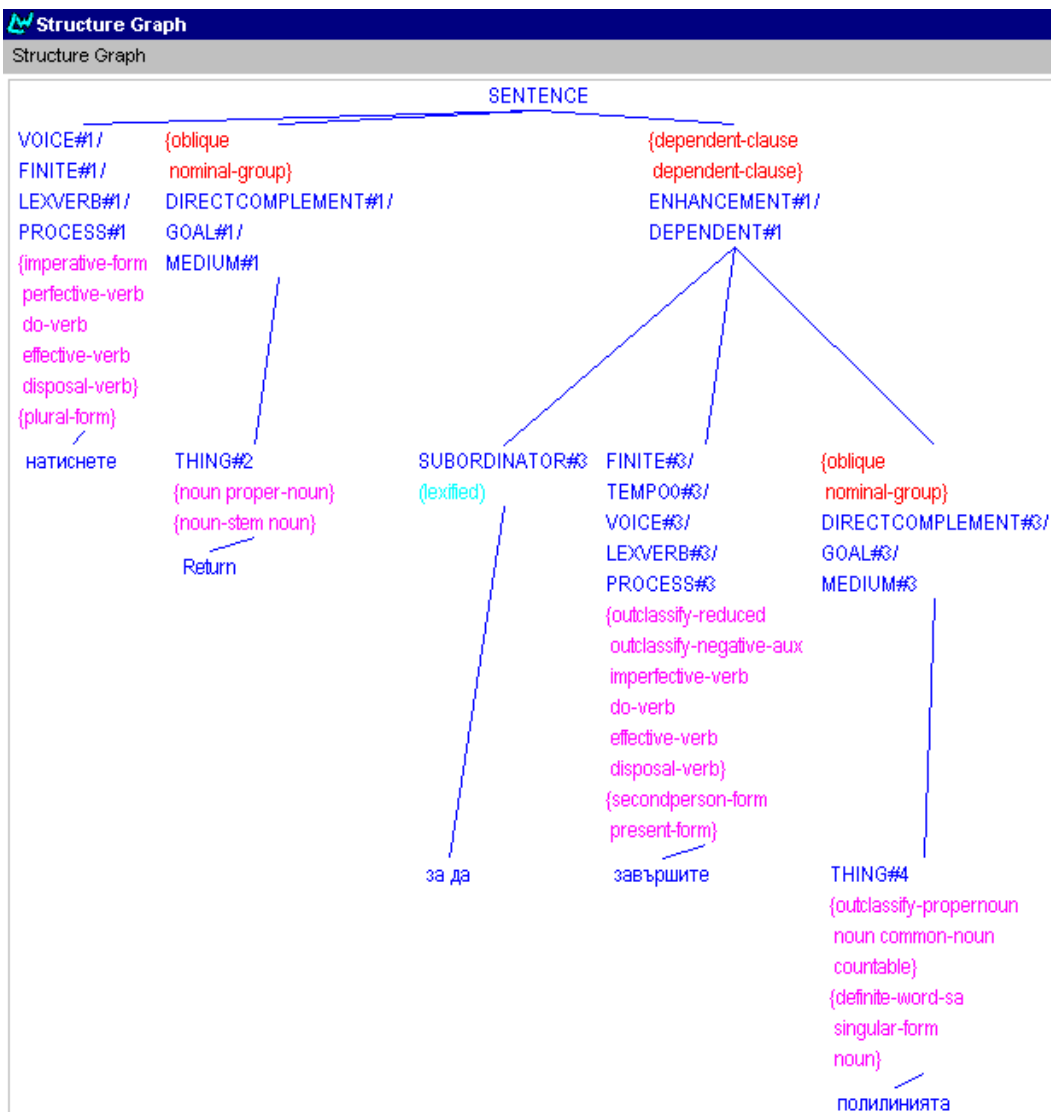
(EXAMPLE

```

:NAME PERS-TEXT2-14
:TARGETFORM " Натиснете Return, за да завършите полилинията."
:LOGICALFORM
  (R / RST-PURPOSE
    :DOMAIN
    (P / DIRECTED-ACTION :LEX NATISNA
      :SPEECHACT IMPERATIVE
      :ASPECT-Q PERFECTIVE-ASPECT
      :ACTEE (D / OBJECT :NAME RETURN))
    :RANGE
    (E / DIRECTED-ACTION :LEX ZAVURSHA
      :ASPECT-Q PERFECTIVE-ASPECT
      :ACTEE
      (P2 / OBJECT :LEX POLILINIA :IDENTIFIABILITY-Q IDENTIFIABLE)))
:SET-NAME PERS-TEXT2)

```

Резултат от генериране с приложната граматика при зададения SPL е слената синтактична структура на изречение.



Фигура 8 Генерирана структура за „Натиснете Return, за да завършите полилинията.“

2.2.2 Елементи от приложната граматика и продукции

В настоящия раздел са показани системи и избиратели от Приложната българска системично-функционална граматика за някои от специфичните езикови явления разгледани в раздел 2.1. Чрез различни примери е демонстрирано генерирането на изречения съдържащи разнообразни езикови явления, чието моделиране е показано в предходния раздел. Описаните модели са приложени в граматиката и тя ги обработва адекватно. Това включва генериране с деятелен и страдателен залог, със свършен и несвършен вид на глагола, с императивно и индикативно наклонение, номинализация на глаголната група, членуване, съгласуване на елементите в групата на съществителното по род и число. Накрая са показани синтактичните конструкции на генерирани сложни изречения.

Типове процеси. Транзитивност и диатезност (зalog)

Следва формалното програмно представяне на системата и избирателя за тип на процес, от които започва спецификацията в граматическото поле транзитивност, и по-точно транзитивност на не-релационни процеси (nonrelationaltransitivity).

```
(SYSTEM
:NAME      PROCESS-TYPE
:INPUTS    TRANSITIVITY-UNIT
:OUTPUTS   ((0.333333 MATERIAL
              (CLASSIFY PROCESS DO-VERB))
              (0.333333 MENTAL
              (CLASSIFY PROCESS EXPERIENCE-VERB))
              (0 VERBAL
              (CLASSIFY PROCESS SYMBOLIC-VERB))
              (0.333333 RELATIONAL
              (CLASSIFY PROCESS RELATIONAL-VERB)))
:CHOOSER   PROCESS-TYPE-CHOOSER
:REGION    NONRELATIONALTRANSITIVITY
:METAFUNCTION EXPERIENTIAL)

(CHOOSER
:NAME      PROCESS-TYPE-CHOOSER
:DEFINITION ((ASK (STATIC-CONDITION-Q PROCESS)
                  (STATIC
                   (ASK (MENTAL-PROCESS-Q PROCESS)
                        (MENTAL (IDENTIFY SENSER
                                (SENSER-ID PROCESS))
                                (COPYHUB SENSER MEDIUM)
                                (CHOOSE MENTAL))
                        (NONMENTAL
                         (ASK (VERBAL-PROCESS-Q PROCESS)
                              (VERBAL (IDENTIFY SAYER
                                        (SAYER-ID PROCESS))
                                        (CHOOSE VERBAL))
                              (NOTVERBAL (CHOOSE RELATIONAL))))))
                  (NONSTATIC
                   (ASK (VERBAL-PROCESS-Q PROCESS)
                        (VERBAL (IDENTIFY SAYER
                                (SAYER-ID PROCESS))
                                (CHOOSE VERBAL))
                        (NOTVERBAL
                         (ASK (MENTAL-PROCESS-Q PROCESS)
                              (MENTAL (IDENTIFY SENSER
                                        (SENSER-ID PROCESS))
                                        (COPYHUB SENSER MEDIUM)
                                        (CHOOSE MENTAL))
                              (NONMENTAL (CHOOSE MATERIAL))))))))))
```

Реализация на система и избирател Тип на процеса

Примерите за генериране на изречение с ефективен материален процес са най-често срещани в целевия корпус. С дадения по-долу SPL за изречението „Начертайте линия.“ се генерира синтактичната структура показана на Фигура 9 Генерирана структура за „Начертайте линия.“, където като характеристики на процеса присъстват do-verb и effective-verb. Първата характеристика е резултат от преминаване на процеса на генериране през изхода MATERIAL на системата Тип на процес, чието формално представяне е изложено по-горе.

(S/ DM::DRAW
 :SPEECHACT IMPERATIVE
 :ACTOR (HEARER / DM::USER IDENTIFIABILITY-Q IDENTIFIABLE)
 :ACTEE (AE / DM::LINE))



Фигура 9 Генерирана структура за „Начертайте линия.“

За характеристиката *effective-verb* е отговорна системата AGENCY, чиято програмна реализация в ресурса е изложена по-долу, заедно с нейния избирател AGENCY-CHOOSER.

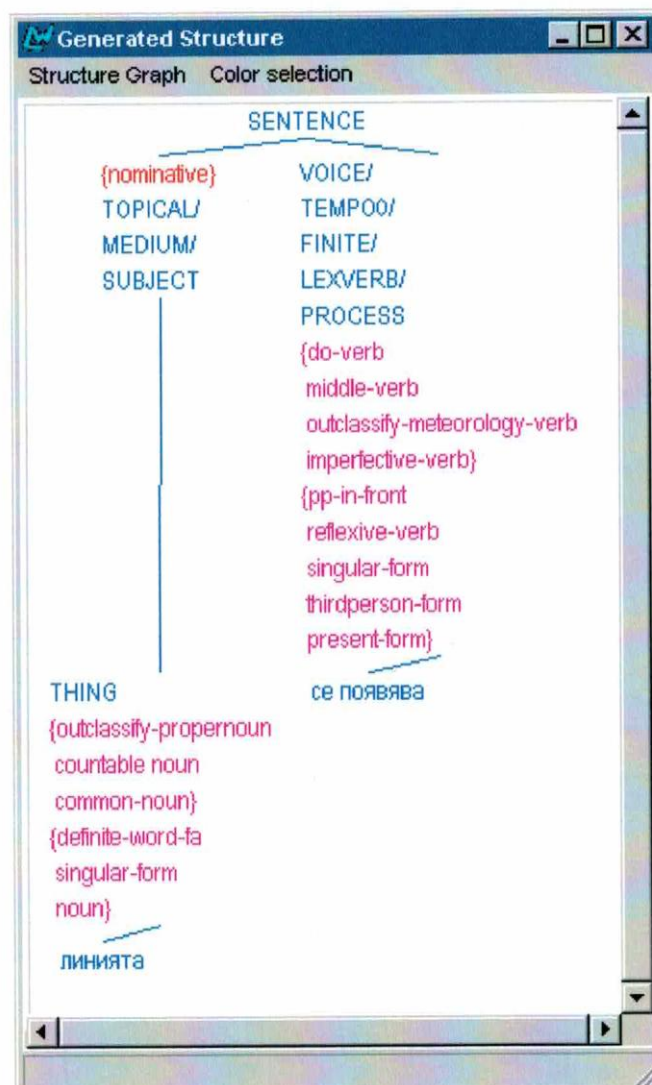
```
(SYSTEM
  :NAME      AGENCY
  :INPUTS    (OR MATERIAL MENTAL VERBAL)
  :OUTPUTS
    (0.5 MIDDLE (CLASSIFY PROCESS MIDDLE-VERB)
     (0.5 EFFECTIVE (CLASSIFY PROCESS EFFECTIVE-VERB)))
  :CHOOSER   AGENCY-CHOOSER
  :REGION    NONRELATIONALTRANSITIVITY
  :METAFUNCTION EXPERIENTIAL)

(CHOOSER
  :NAME      AGENCY-CHOOSER
  :DEFINITION ((ASK (VERBAL-PROCESS-Q PROCESS)
    (VERBAL
      (ASK (ADDRESSEE-ORIENTED-Q PROCESS)
        (ADDRESSEEORIENTED (COPYHUB SAYER AGENT)
          (CHOOSE EFFECTIVE)))
      (NOTADDRESSEEORIENTED (CHOOSE MIDDLE))))
    (NOTVERBAL
      (ASK (CAUSED-PROCESS-Q PROCESS)
        (INDEPENDENT (CHOOSE MIDDLE))
        (CAUSED (IDENTIFY AGENT
          (CAUSER-ID PROCESS))
          (ASK (CAUSATIVE-MENTION-Q AGENT PROCESS)
            (WITHHOLD (CHOOSE MIDDLE))
            (MENTION (CHOOSE EFFECTIVE))))))
    )))
```

Реализация на система и избирател AGENCY

За изречението „Начертайте линията.“ Избирателят е насочил процеса на генериране към изхода EFFECTIVE. Със следващия SPL се демонстрира преминаване на процеса на генериране през алтернативния изход: MIDDLE. Това е нужно, когато се реализират средни процеси, какъвто е случая с процеса *появявам-се /appear*. Характеристиката *middle-verb* е видна за генерирания процес в синтактичната структура на изречението „Линията се появява.“ на Фигура 10 по-долу.

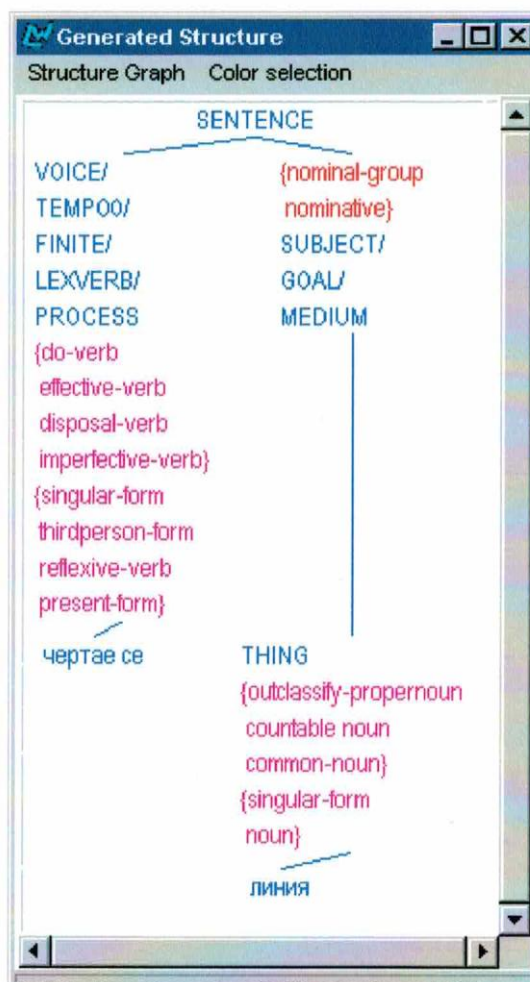
```
(S / DM::APPEAR
:ACTOR (L / DM::LINE
  :IDENTIFIABILITY-Q IDENTIFIABLE))
```

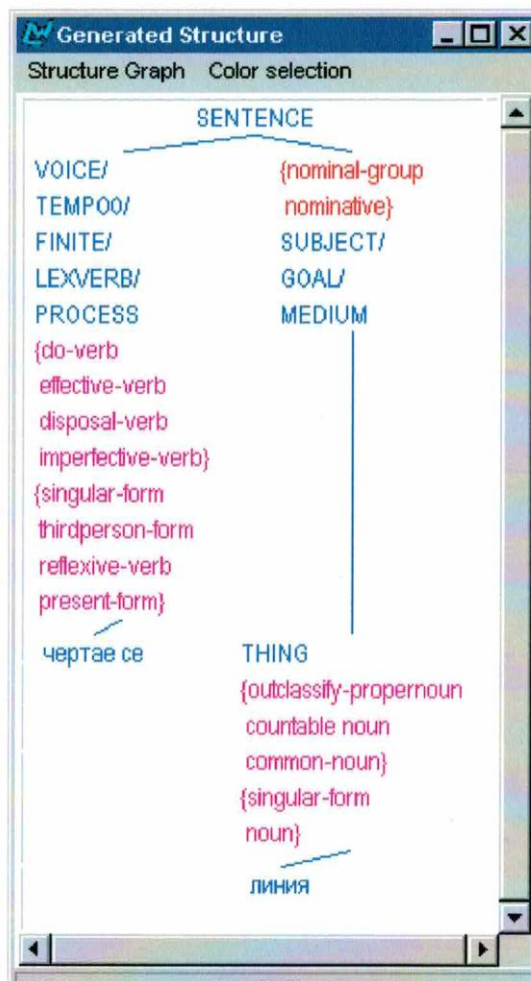
Фигура 10 Генерирана структура за „Линията се появява.“

Важно е да обърнем внимание на характеристиките за реализация на процес с middle-verb и reflexive-verb в горната синтактична структура. Както споменахме в предходния раздел процесът изразен с глагола „появявам се“ е среден процес, който има само един участник. Интересно е сравнението с често използваната в избрания регистър пасивна конструкция генерирана, например, в изречението „Чертае се линия.“ При нея процесът е ефективен с двама

участници, но реализиран в страдателен залог. Това личи от характеристиките на процеса do-



verb и effective-verb показани на следващата
Фигура 11.



Фигура 11 Генерирана структура за „Чертае се линия.“

Следва входното SPL- задание за генериране на показаното по-горе изречение „Чертае се линия.“

(S / DM:: DRAW

:PREFER-MENTION-AGENT-Q WITHHOLD

:ACTEE (AE / DM::LINE))

В този SPL директно е зададена стойност за проучвателя наречен PREFER-MENTION-AGENT-Q, именно стойност WITHHOLD, която направлява избора на страдателен залог за изразяване на входната семантика. Това може да бъде проследено на по-долу във формалната програмно представяне на системата и избирателя AGENTIVITY, където е видно как избирателя AGENTIVITY се обръща към проучвателя PREFER-MENTION-AGENT-Q, за да вземе решение за залога.

```

(SYSTEM
  :NAME      AGENTIVITY
  :INPUTS    (OR EFFECTIVE-MATERIAL EFFECTIVE-MENTAL EFFECTIVE-VERBAL)
  :OUTPUTS   ((0.9 NONAGENTIVE)
              (0.1 AGENTIVE))
  :CHOOSER   AGENTIVITY-CHOOSER
  :REGION    VOICE
  :METAFUNCTION EXPERIENTIAL)

(CHOOSER
  :NAME      AGENTIVITY-CHOOSER
  :DEFINITION ((ASK (PREFER-MENTION-AGENT-Q AGENT PROCESS)
                    (WITHHOLD (CHOOSE NONAGENTIVE))
                    (MENTION (CHOOSE AGENTIVE))))))

```

Реализация на система и избирател AGENTIVITY

Интересно по отношение спецификата на българския език е следващото уточнение за тип на пасивната конструкция, когато вече е избран страдателен залог. Възможните конструкции са на т.н. *participle-passive* и *medio-reflexive-passive* или между реализациите „съм чертал“ и „чертае се“. Програмната реализация на системата и избирателя от полето залог/voice отговорни за типа на пасивната конструкция са показани по-долу.

```

(SYSTEM
  :NAME      PASSIVE-CONSTRUCTION-TYPE
  :INPUTS    PASSIVE-PROCESS
  :OUTPUTS   ((0.5 PARTICIPLE-PASSIVE
              (CLASSIFY VOICE SAM-AUX)
              (OUTCLASSIFY VOICE NEGATIVE-AUX)
              (INSERT VOICEDPENDENT)
              (INFLECTIFY VOICEDPENDENT L-PARTICIPLE))
              (0.5 MEDIO-REFLEXIVE-PASSIVE
              (CONFLATE FINITE LEXVERB)
              (INFLECTIFY FINITE REFLEXIVE-VERB)
              (PRESELECT DEPENDENT NONAGENTIVE))))
  :CHOOSER   PASSIVE-CONSTRUCTION-TYPE-CHOOSER
  :REGION    VOICE)

(CHOOSER
  :NAME      PASSIVE-CONSTRUCTION-TYPE-CHOOSER
  :DEFINITION ((ASK (COMMAND-Q SPEECHACT)
                    (COMMAND (CHOOSE PARTICIPLE-PASSIVE))
                    (NOCOMMAND
                     (ASK (REPEATABLE-Q PROCESS)
                          (NONREPEATABLE (CHOOSE PARTICIPLE-PASSIVE))
                          (REPEATABLE (CHOOSE MEDIO-REFLEXIVE-PASSIVE)))))))

```

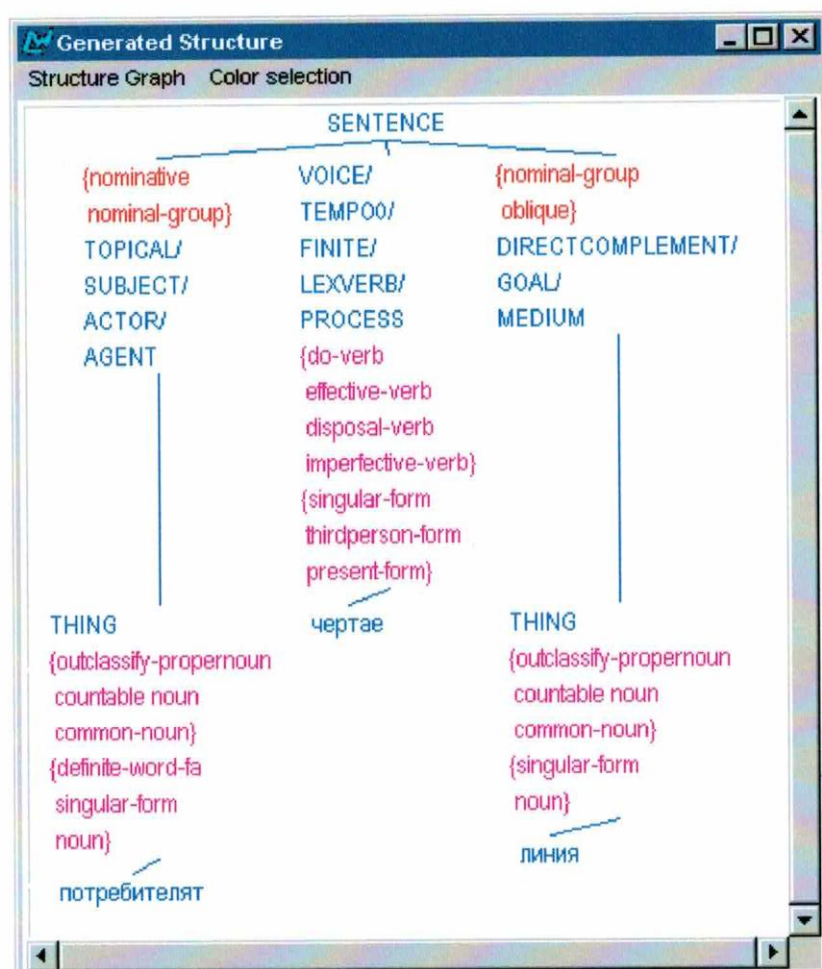
Реализация на система и избирател за тип на пасивната конструкция

Следващият SPL задава реализация на деятелен залог демонстриран от изречението “Потребителят чертае линия.” Синтактичната конструкция генерирана с този семантичен вход е показана на Фигура 12 по-долу.

```

(S / DM::DRAW
:ACTOR      (AR / DM::USER
             :IDENTIFIABILITY-Q IDENTIFIABLE)
:ACTEE      (AE / DM::LINE))

```



Фигура 12 Генерирана структура за „Потребителят чертае линия.“

Модалност. Наклонение

Както беше споменато в предходния раздел реализацията на речевия акт се отразява от елемента на Модуса: двойката Subject-Finite. Когато Акторът е и слушател при речевия акт в семантичния вход за генерирането задаваме роля на слушател.

(S / DM::DRAW

:ACTOR (HEARER / DM::USER
:IDENTIFIABILITY-Q IDENTIFIABLE)

:ACTEE (AE / DM::LINE))

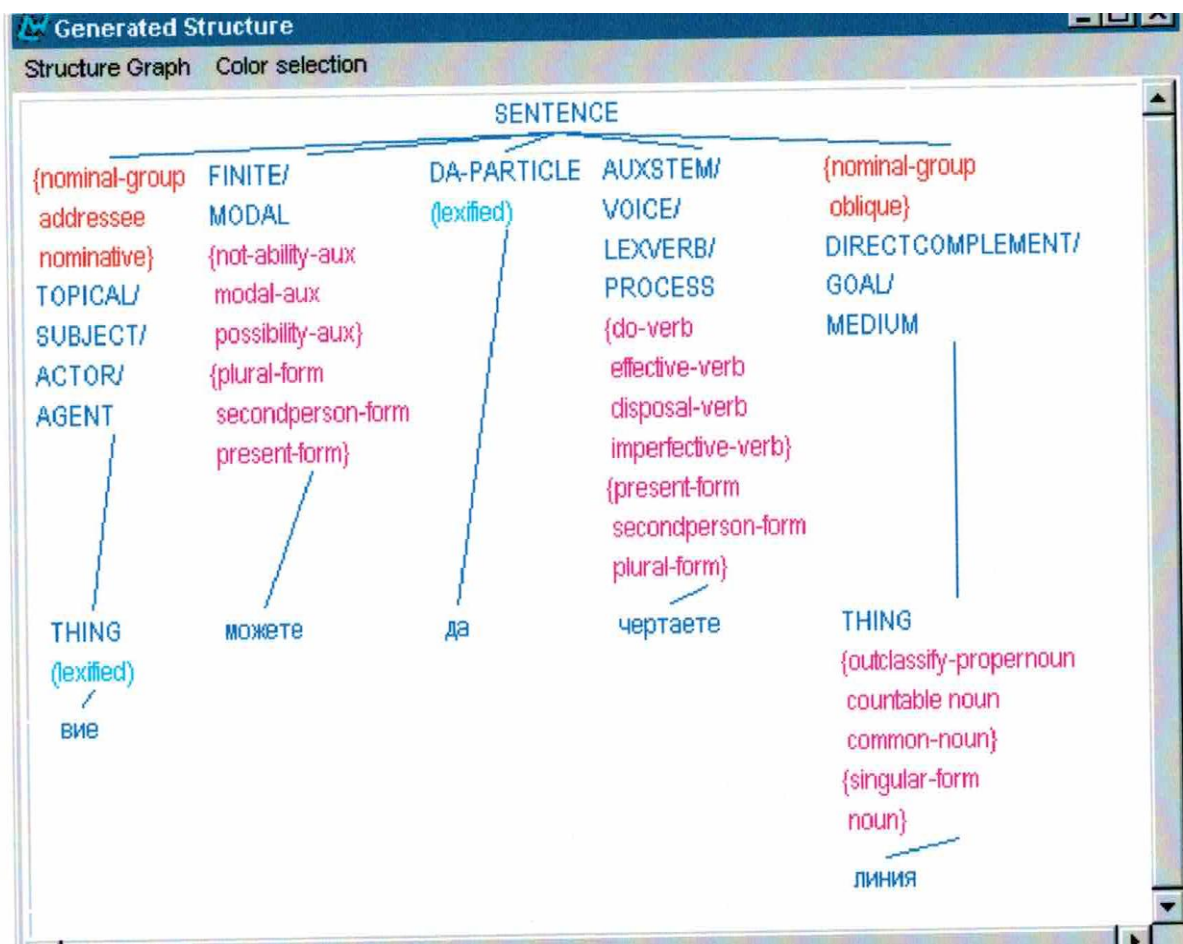
Резултат от генерирането е изречението „Вие чертаете линия.“, чиято синтактична структура е показана на следващата Фигура 13:



Фигура 13 Генерирана структура за „Вие чертаете линия.“

Настройването на българската приложна граматика дава възможност за генериране на функцията модалност, т.е. изразяване на възможност, необходимост и т.н. Въведена е типичната „да-конструкция“ за реализиране на модални глаголни групи като „мога да...“, „трябва да...“ и т.н. Доказателство за това е следващият семантичен вход и реализацията на изречението „Вие можете да чертаете линия.“ показана на Фигура 14 по-долу.

```
(S / DM::DRAW
:MODAL-PROPERTY-ASCRPTION GENERAL-POSSIBILITY
:ACTOR      (HEARER / DM::USER
              :IDENTIFIABILITY-Q IDENTIFIABLE)
:ACTEE      (AE / DM::LINE))
```



Фигура 14 Генерирана структура за „Вие можете да чертаете линия.“

Темпоралност

Характеристиката present-form е фиксирана за глаголите, реализиращи процеси във всеки от горните 5 примера.

Завършеност на процеса. Вид на глагола

По отношение на разнообразието може да се каже, че свършен и несвършен вид за един и същ процес водят към различни реализации. Например, за процеса DRAW са достъпни две реализации свързани с алтернативите за вид: за несвършен вид процесът се реализира с глагола „чертая“, за свършен вид се използва реализация с глагола „начертая“.

В SPL-а това се задава явно така:

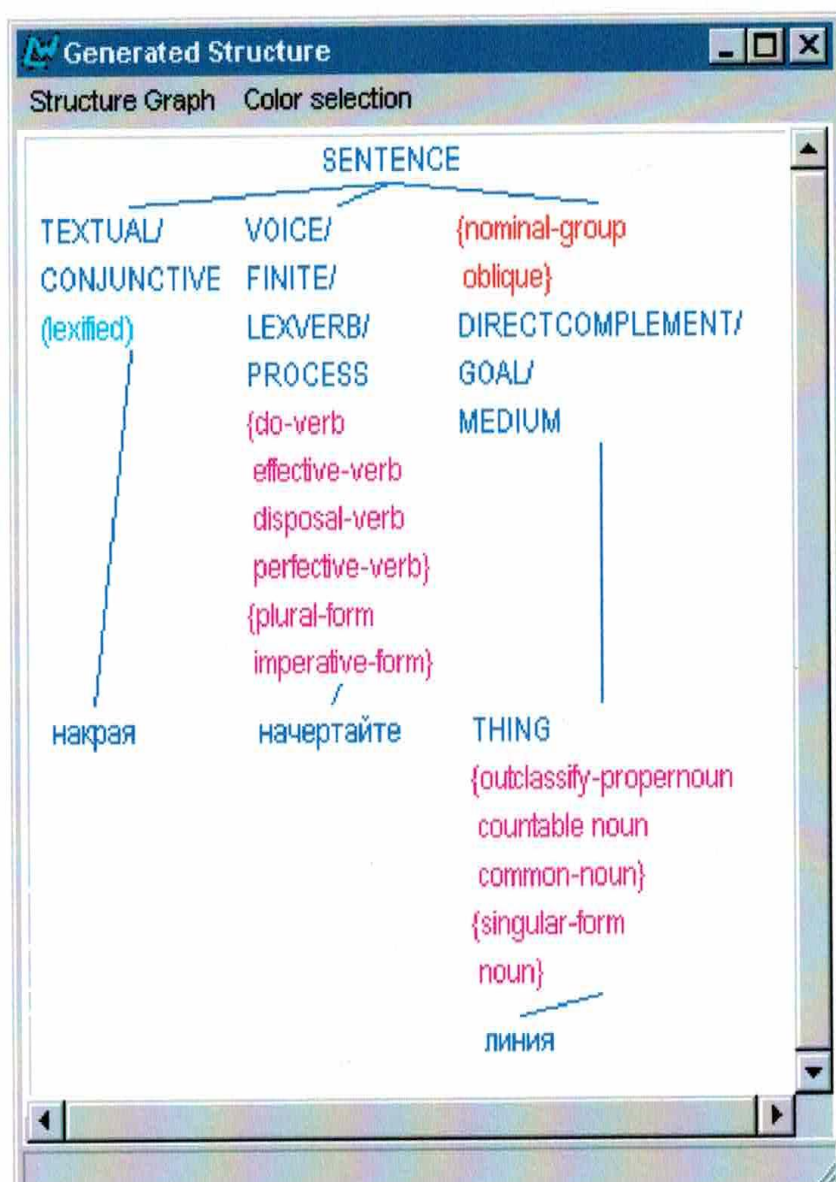
:ASPECT-Q PERFECTIVE-ASPECT

Съответните характеристики уточнени по време на генерирането могат да бъдат видяни при реализациите на изреченията „Начертайте линия.“ и „Чертае се линия.“ Съответно на Фигура 9 Генерирана структура за „Начертайте линия.“ имаме характеристика за процеса: perfective-verb, а на Фигура 13 Генерирана структура за „Вие чертаете линия.“ имаме характеристика за процеса: imperfective-verb.

Текстова кохерентност/ свързаност

Текстова кохерентност или свързаност реализирана с текстови съюзи / Textual Conjunction е демонстрирана чрез вариации на изречения като например следните: „Първо изберете точка.“, „След това въведете ъгъл“, „Накрая начертайте линия.“ Последното примерно изречение е показано на Фигура 15.

(S/ DM::DRAW
:SPEECHACT IMPERATIVE
:ACTOR (HEARER / DM::USER IDENTIFIABILITY-Q IDENTIFIABLE)
:ACTEE (AE / DM::LINE)
:CONJUNCTIVE SEQUENCE-LAST)



Фигура 15 Генерирана структура за „Накрая начертайте линия.“

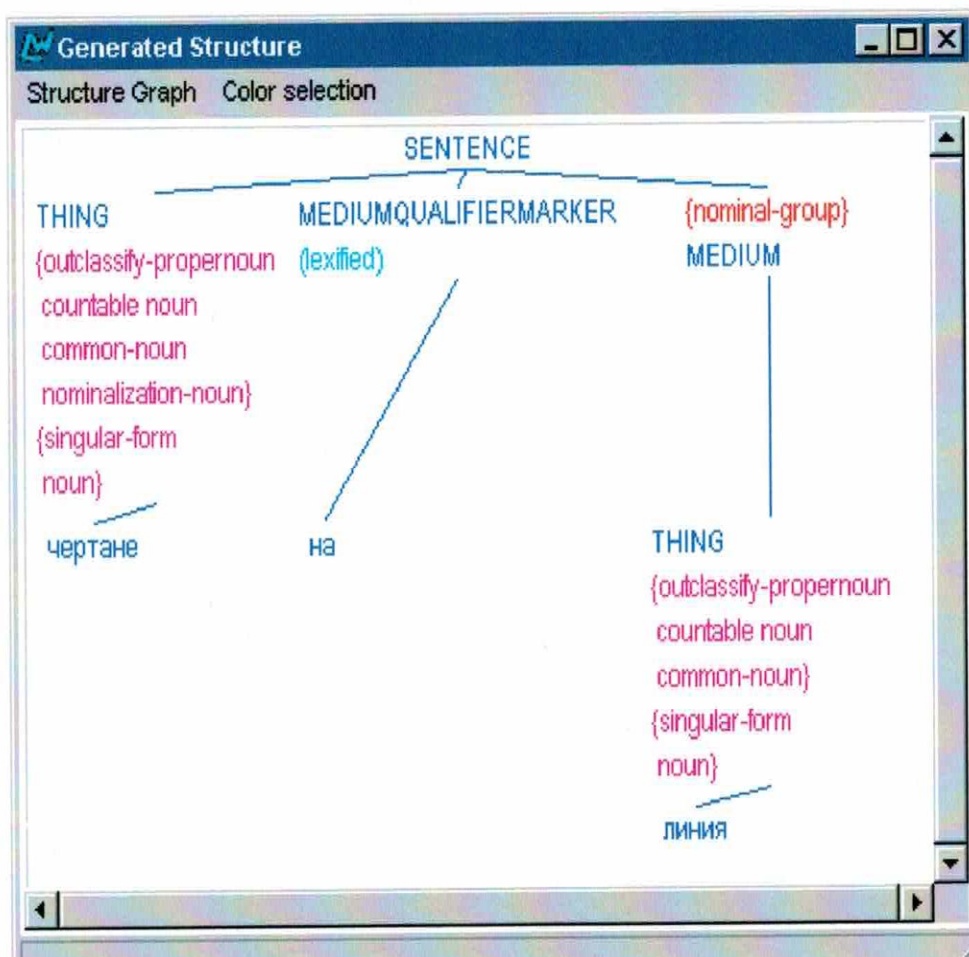
Номинализация

Генерираната структура на Фигура 16 показва реализацията на номинализираната глаголна група „чертая линия“.

(S / DM::DRAW

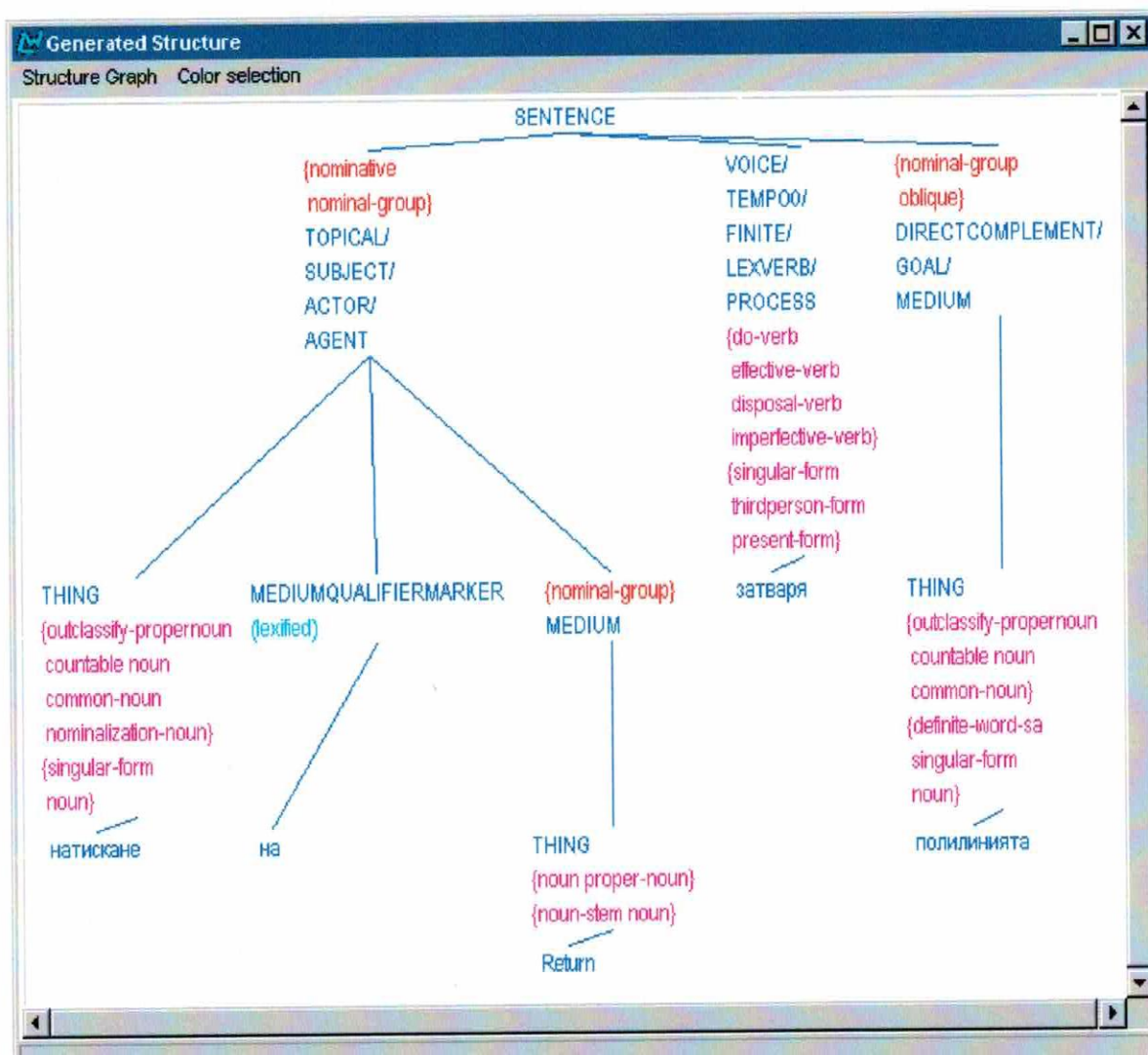
: EXIST-SPEECH-ACT-Q NOSPEECHACT

:ACTEE (AE / DM::LINE))



Фигура 16 Генерирана структура за „Чертане на линия“

Генерираната структура на Фигура 17 показва реализацията на номинализираната глаголна група „натискам Return“, като номинализацията е използвана за подлог в генерираното изречение.



Фигура 17 Генерирана структура за „Натискане на Return затваря полилинията.“

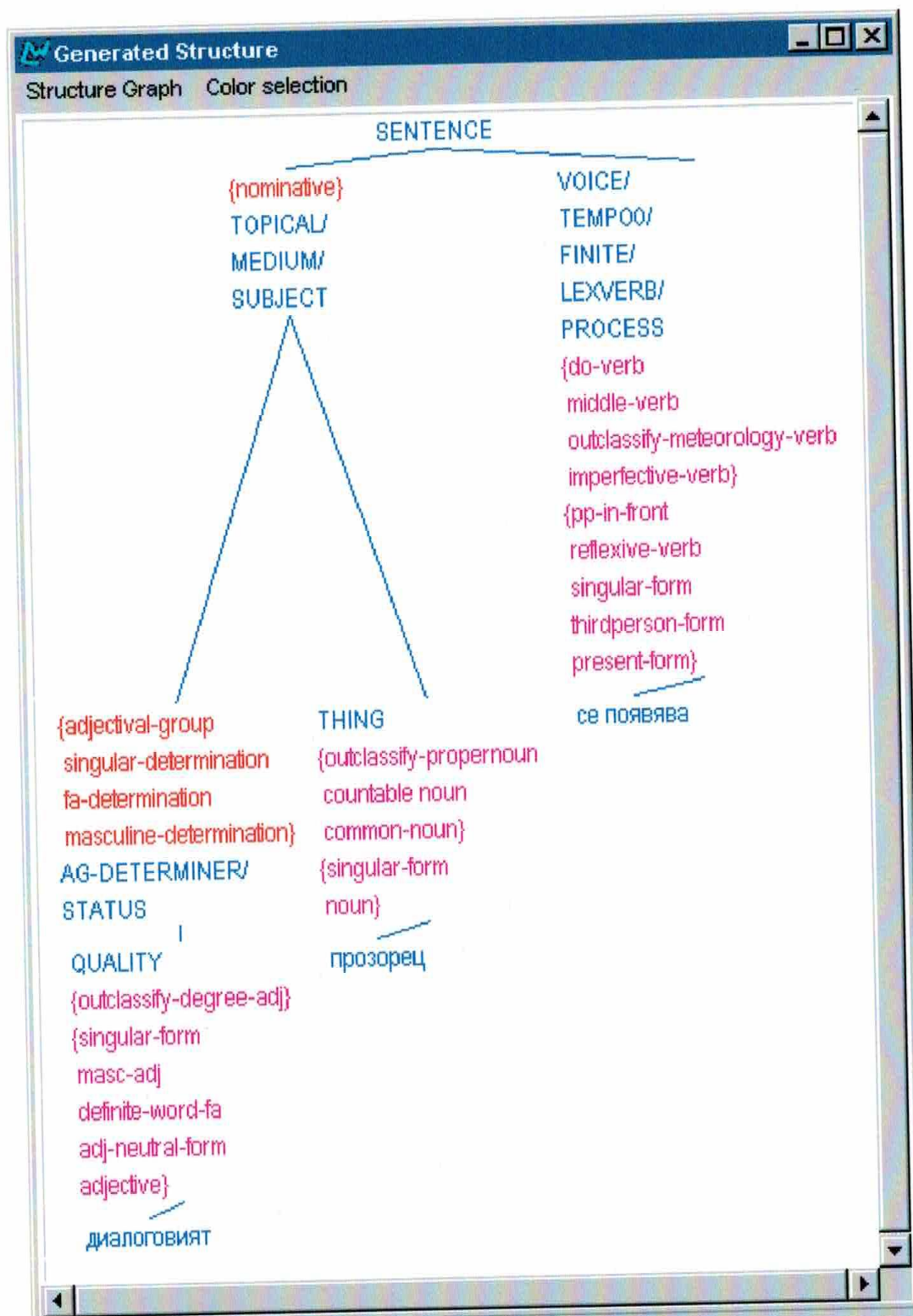
Членуване и съгласуване по род и число в групата на съществителното.

Показаната на Фигура 18 генерирана структура съдържа членуване на съществително от женски род.

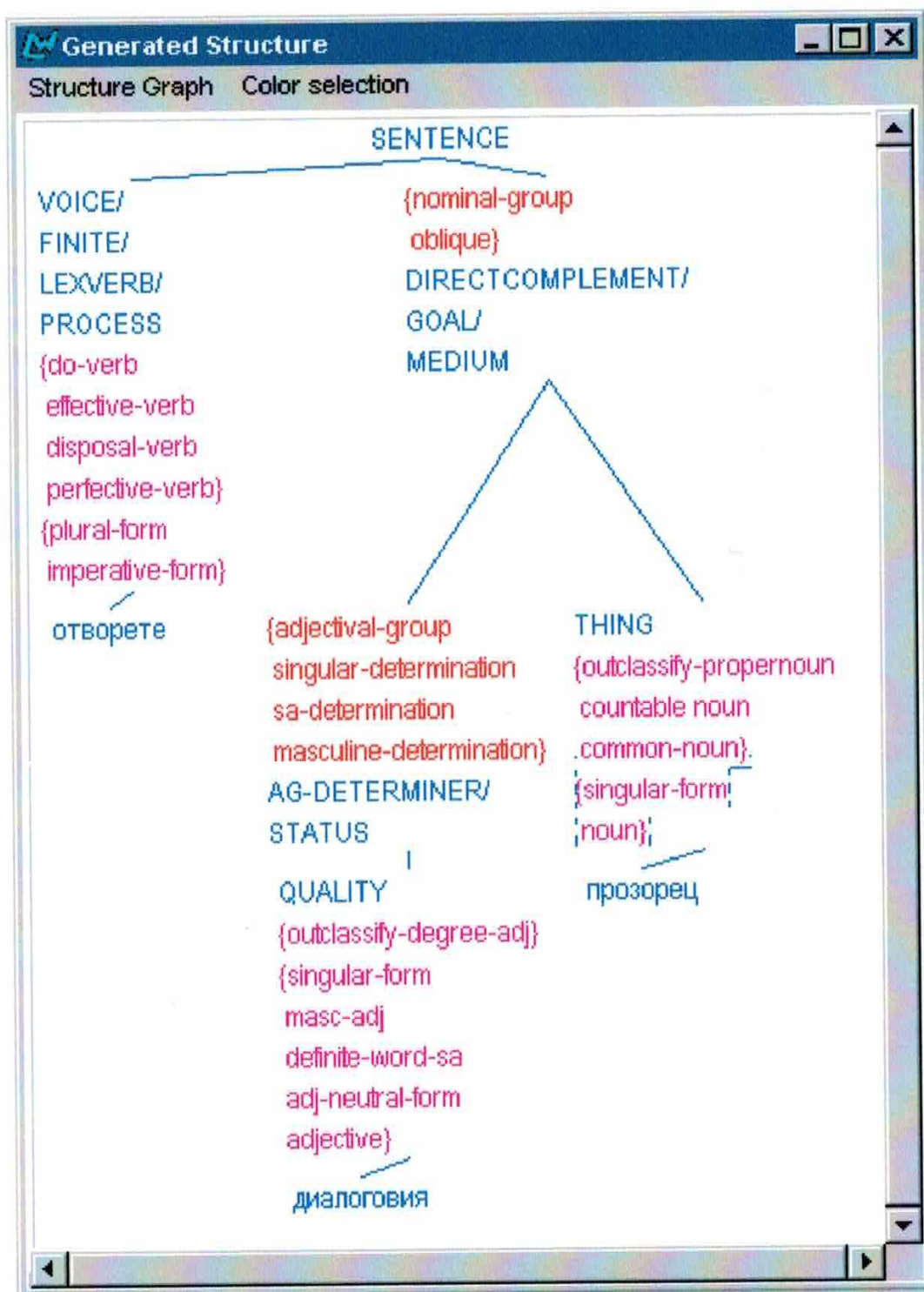


Фигура 18 Генерирана структура за „Начертайте линията.“

Генерираните структури на следващите две фигури 19 и 20 показват реализация на пълен и непълен член за прилагателно от мъжки род в групата на съществителното.



Фигура 19 Генерирана структура за „Диалоговият прозорец се появява.“

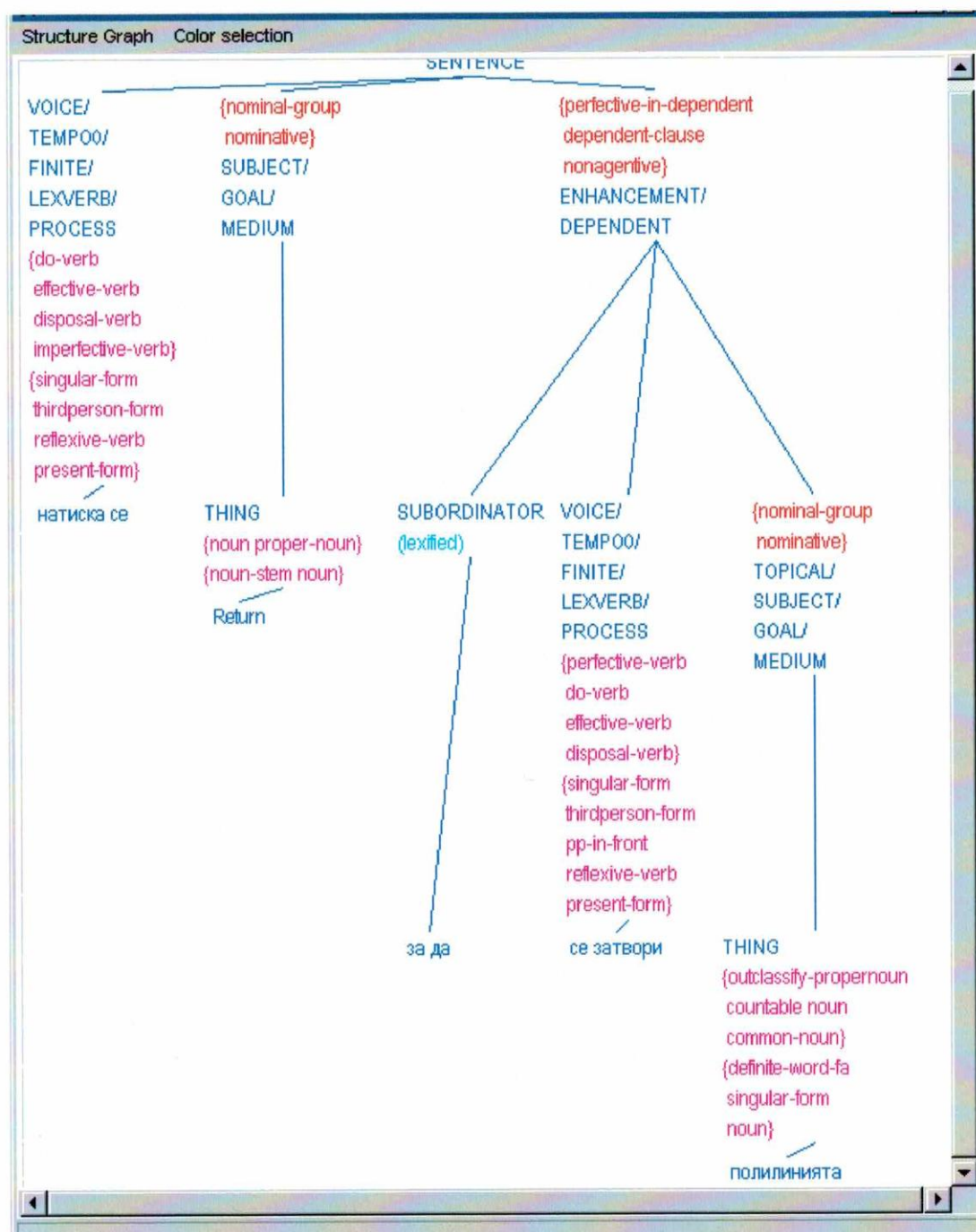


Фигура 20 Генерирана структура за „Отворете диалоговия прозорец.“

Сложни изречения

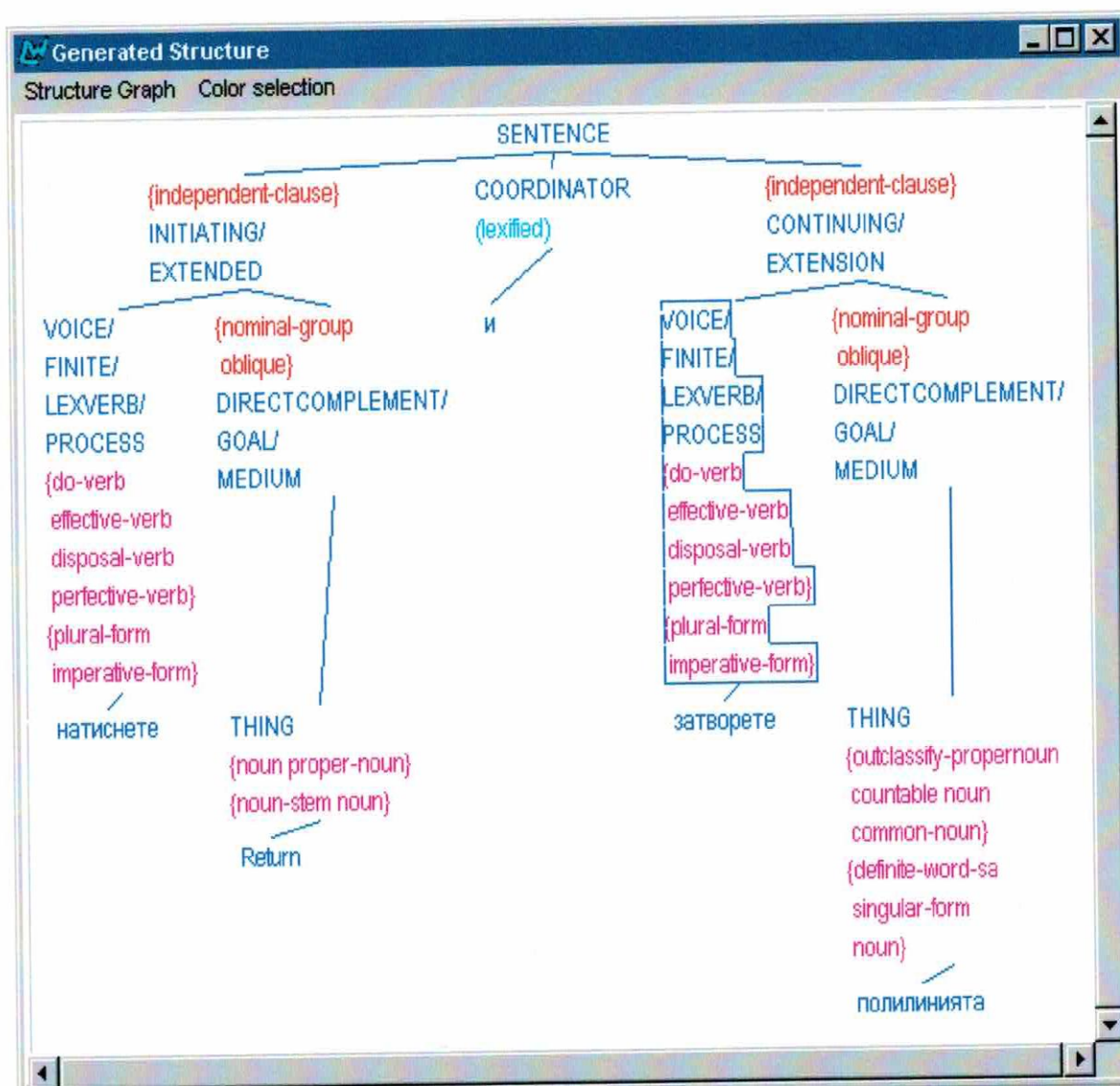
Полето от приложната граматика, което съдържа систмите, избирателите и проучвателите за генериране на сложни изречения се нарича **CLAUSECOMPLEX**. Премиваването на процеса на генериране през това поле е демонстрирано със следващите генерирани структури на сложни изречения на фигури 21, 22 и 23.

(S / RST-PURPOSE
:PREFER-MENTION-AGENT-Q WITHHOLD
:DOMAIN (D / DM::PRESS
:ACTOR (HEARER / DM::USER IDENTIFIABILITY-Q IDENTIFIABLE)
:ACTEE (AE1 / DM::RETURN))
RANGE (R / DM::CLOSE-SCREEN-OBJECT
:ACTOR (HEARER / DM::USER IDENTIFIABILITY-Q IDENTIFIABLE)
:ACTEE (AE1 / DM::POLYLINE IDENTIFIABILITY-Q IDENTIFIABLE)))



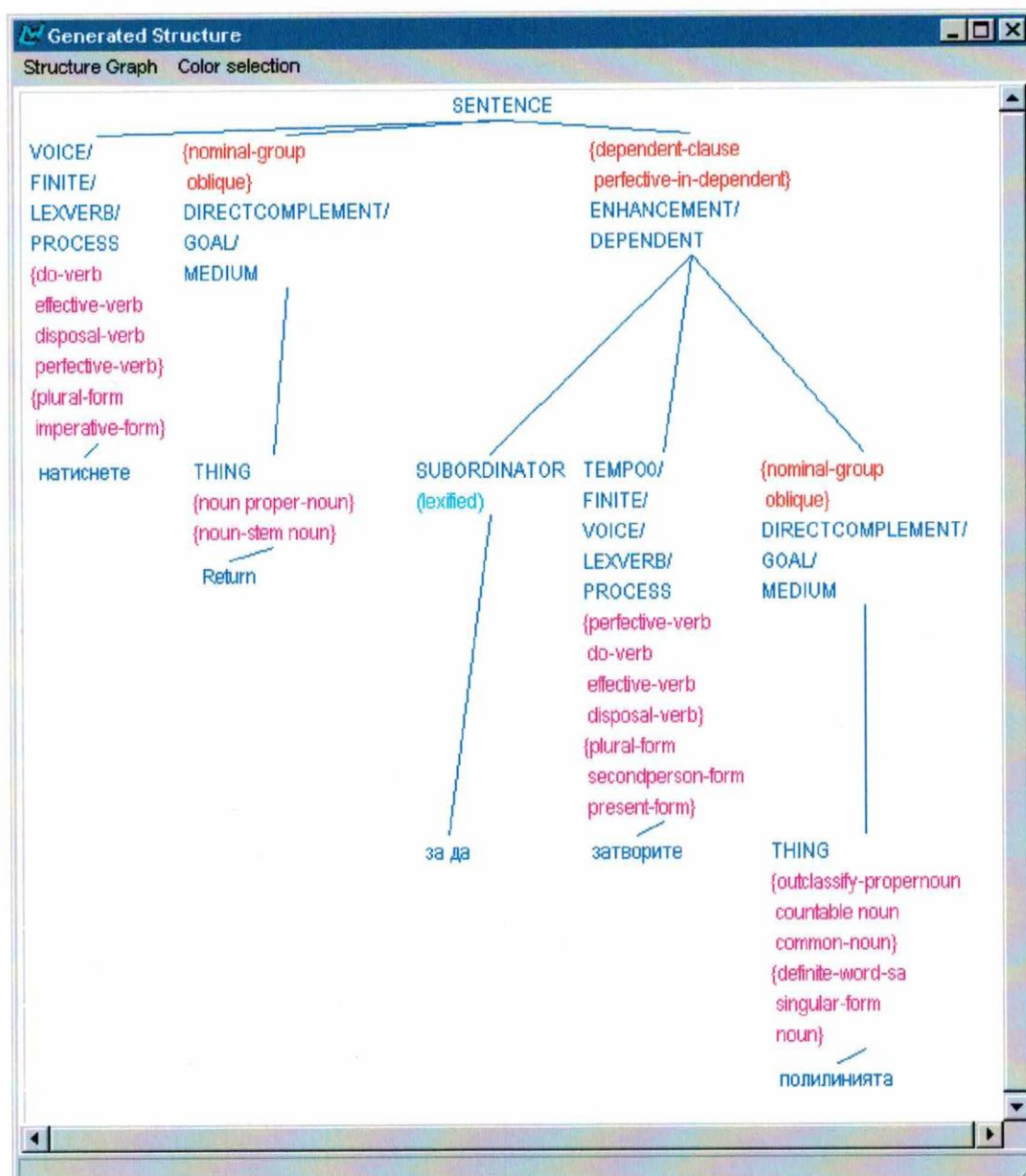
Фигура 21 Генерирана структура за „Натиска се Return, за да се затвори полилинията.“

(S / CONJUNCTION
 :SPEECHACT IMPERATIVE
 :DOMAIN (D / DM::PRESS
 :ACTOR (HEARER / DM::USER IDENTIFIABILITY-Q IDENTIFIABLE)
 :ACTEE (AE1 / DM::RETURN))
 RANGE (R / DM::CLOSE-SCREEN-OBJECT
 :ACTOR (HEARER / DM::USER IDENTIFIABILITY-Q IDENTIFIABLE)
 :ACTEE (AE1 / DM::POLYLINE IDENTIFIABILITY-Q IDENTIFIABLE)))



Фигура 22 Генерирана структура за „Натиснете Return и затворете полилинията.“

(S / RST-PURPOSE
 :SPEECHACT IMPERATIVE
 :DOMAIN (D / DM::PRESS
 :ACTOR (HEARER / DM::USER IDENTIFIABILITY-Q IDENTIFIABLE)
 :ACTEE (AE1 / DM::RETURN))
 RANGE (R / DM::CLOSE-SCREEN-OBJECT
 :ACTOR (HEARER / DM::USER IDENTIFIABILITY-Q IDENTIFIABLE)
 :ACTEE (AE1 / DM::POLYLINE IDENTIFIABILITY-Q IDENTIFIABLE)))



Фигура 23 Генерирана структура за „Натиснете Return, за да затворите полилинията.“

Раздел 2.2 показва принципите на създаване и резултати при генериране постигнати от Приложната българска системично-функционална граматика, която представлява ресурс за автоматично генериране на български изречения. Създадена е на базата на обширния ресурс за генериране на английски изречения NIGEL. Използвани са съществено изложените в предходния раздел 2.1 системично-функционални модели за някои граматически явления в българския език.

Демонстрирани са възможностите на създадения ресурс да генерира сравнително широк обхват от алтернативни граматически характеристики в изреченията (напр. деятелен и страдателен залог, императивно и декларативно изречение, просто и сложно изречение, няколко вариации на хипотактични и паратактични връзки в сложното изречение, номинализация на глаголна група и др.)

Постигнатият обхват на генериране е прагматично ограничен от конкретния стил на текстовете от корпуса: процедурни текстовете-инструкции. От друга страна, ресурсът е достъпен за преизползване и разширяване на генерираните вариации, тъй като се базира стриктно на системично-функционалното моделиране на българския език, което отчита лингвистичната природа на изходния продукт от генерирането- добре формулираното българско изречение.

Глава 3. Реализация на процеса на генериране на български текстове

В тази глава е описан самият процес на автоматично генериране на текстове-инструкции на български език, както той е конструиран и изпълнен от системата AGILE.

Процесът на задаване на параметри за автоматичното генериране започва с въвеждане на понятията от предметната област на бъдещите текстове, това представлява дефинирането на модела на предметната област на CAD/CAM-приложенията. Осъществява се свързване на тези понятия с Обобщения модел, онтологията от високо ниво, която осигурява съответствието между понятията от предметна област и тяхната системично-функционална проекция по отношение на българския език (раздел 3.1).

Определянето на съдържанието на бъдещия текст се извършва от потребителя на системата с помощта на текст-структуриращи елементи, които са описани в раздел 3.2.

След получаване на входните данни за процеса на автоматично генериране работи модулът за планиране на текст, а след него и планировчикът на изречение, които създават планове на всяко отделно изречение като част от цялостен кохерентен текст. Механизмът е разяснен в раздел 3.3 заедно с идеята за създаване на няколко текстови стилови разновидности на изходния текст.

Лексико-граматическата реализация на автоматичното генериране на текст се извършва в средата за многоезиково генериране KPML от приложната българска системично-функционална граматика, чието създаване беше описано в предходната глава на дисертацията. Описанието на реализирания процес на автоматично генериране на текстове- инструкции на български език е завършено с коментар на резултата от автоматичното генериране на 5 различни по стил текстове на български език от конкретен зададен вход за системата (раздел 3.4).

3.1 Представяне на знания за генерирането на български текстове

Избраната предметна област е свят на CAD/CAM термини и инструкции. Както беше споменато в предходната глава източник на текстове-инструкции за текстовия корпус са ръководствата за ползване на CAD/CAM софтуер. Тези ръководства се състоят главно от указания как да се създават различни видове проекти и чертежи, как да се съхраняват файловете на даден проект, как да се зареждат отново, да се редактират и т.н.

Базови понятия на тази предметна област са понятия като ФАЙЛ, КЛАВИШ, ЕКРАН, различни видове графични обекти: ЛИНИЯ, ДЪГА и т.н. основните действия в областта се извършват от потребителя: ЧЕРТАЯ, ЗАПАЗВАМ (файл), НАТИСКАМ (клавиш) и т.н. Моделът на предметната област съдържа също текст-структуриращи елементи, които са градивни блокове на бъдещите процедурни текстове.

Елементите от предметната област са разделени в няколко групи:

ПРОЦЕСИ:

ДЕЙСТВИЯ_на_ПОТРЕБИТЕЛЯ (виж списък в Приложение 1: Модел на предметна област „CAD/CAM приложения“)

СЪБИТИЯ

ПОЯВЯВАМ-СЕ

ИЗЧЕЗВАМ

НЕЩА:

КЛАВИШ

ЕКРАН

ЛИНИЯ

МУЛТИЛИНИЯ

ТОЧКА

ЕТИКЕТИ_на_ГИ (графичния интерфейс)

ОПЕРАЦИОННА_СИСТЕМА

Windows

DOS

UNIX

ПОТРЕБИТЕЛ

и т.н.

Изчерпателен списък на елементите от приложната област е предложен в Приложение 1: Модел на предметна област „CAD/CAM приложения“.

Текст-структуриращите елементи са четири на брой и предоставят контекст, в който действията и събитията могат да се случат. Текст-структуриращите елементи са: ПРОЦЕДУРА, МЕТОД, СПИСЪК_от_ПРОЦЕДУРИ и СПИСЪК_от_МЕТОДИ. Следват техните дефиниции.

ПРОЦЕДУРА

Има три слота:

ЦЕЛ (задължителен, попълва се с ДЕЙСТВИЯ_на_ПОТРЕБИТЕЛЯ)

МЕТОДИ (незадължителен, попълва се със СПИСЪК_от_МЕТОДИ)

СТРАНИЧЕН ЕФЕКТ (незадължителен, попълва се със СЪБИТИЯ)

МЕТОД

Има три слота:

ОГРАНИЧЕНИЕ (незадължителен, попълва се с ОПЕРАЦИОННА_СИСТЕМА)

УСЛОВИЕ (незадължителен, попълва се с ПРОЦЕДУРА)

ПОД-СТЪПКИ (задължителен, попълва се със СПИСЪК_от_ПРОЦЕДУРИ)

СПИСЪК_от_МЕТОДИ

Представява списък от елементи тип МЕТОД:

ПЪРВИ (задължителен, от тип МЕТОД)

ОСТАНАЛИ (незадължителен, от тип СПИСЪК_от_МЕТОДИ)

СПИСЪК_от_ПРОЦЕДУРИ

Представява списък от елементи тип ПРОЦЕДУРА:

ПЪРВИ (задължителен, от тип ПРОЦЕДУРА)

ОСТАНАЛИ (незадължителен, от тип СПИСЪК_от_ПРОЦЕДУРИ)

Дефиниции на атрибутите на текст- структуриращите елементи.

Атрибути на ПРОЦЕДУРА:

ЦЕЛ: Задава някакво действие изпълнявано от потребителя, което да се опише в детайлни стъпки от развитата нататък процедура. Може да служи за задаване на заглавие на основния процедурен текст, който да бъде генериран.

ДЕЙСТВИЯ_на_ПОТРЕБИТЕЛЯ са специфична група ПРОЦЕСИ дефинирани по-горе.

МЕТОДИ: Задава формален СПИСЪК_от_МЕТОДИ чрез които да се постигне целта на процедурата. Най-често този списък е преход към друго действие на потребителя.

СТРАНИЧЕН ЕФЕКТ: Задава връзка между въведени вече в процедурата ДЕЙСТВИЯ_на_ПОТРЕБИТЕЛЯ и възникващи СЪБИТИЯ като следствие от тези действия. Най-често срещаните примери са на екрана да се появи или изчезне изображението на някакъв графичен обект при чертаене или изтриване, съответно. СЪБИТИЯ са специфична група ПРОЦЕСИ дефинирани по-горе.

Атрибути на МЕТОД

ОГРАНИЧЕНИЕ: Дава възможност да се направи разлика между методи приложими под Windows и методи приложими под DOS или UNIX както е демонстрирано от следния типичен за текстовия корпус примерен текст:

Стартирайте командата PLINE, като използвате един от следните методи:

Windows: От плаващото меню Polyline на функционалния ред Draw изберете Polyline.

DOS и UNIX: От менюто Draw изберете Polyline.

Затова слотът за атрибута ОГРАНИЧЕНИЕ се попълва с ОПЕРАЦИОННА_СИСТЕМА.

УСЛОВИЕ: Изразява някакво състояние, което трябва да е налично, за да се реализира описваният метод. Попълва се с текст-структуриращия елемент ПРОЦЕДУРА.

ПОД-СТЪПКИ: Дава отделните действия, които описват метода. Попълва се с текст-структуриращия елемент СПИСЪК_от_ПРОЦЕДУРИ и съдържа очакване за последователност от по-прости стъпки, които да описват конкретния метод.

Атрибути на СПИСЪК_от_МЕТОДИ

Представява списък от елементи тип МЕТОД:

ПЪРВИ: от тип МЕТОД

ОСТАНАЛИ: от тип СПИСЪК_от_МЕТОДИ

Атрибути на СПИСЪК_от_ПРОЦЕДУРИ

Представява списък от елементи тип ПРОЦЕДУРА:

ПЪРВИ: от тип ПРОЦЕДУРА

ОСТАНАЛИ: от тип СПИСЪК_от_ПРОЦЕДУРИ

Използване на Обобщения модел

В обзорната част на настоящата дисертация (раздел 1.2.1) е посочена ролята и важността на онтологията от високо ниво Обобщен модел в процеса на интерпретиране на съдържанието при автоматично генериране на естествен език. Обобщеният модел е практически използван като посредник между нивата с по-висока абстракция в процеса на текст-генериране (виж Фигура 1: Абстрактна система за КГТ) и “по-ниските” нива на лексико-граматическата реализация за английски, немски, италиански език.

В работата по проекта е направен системично-функционален анализ на понятията от предметната област на CAD/CAM - приложения и беше установено, че понятията могат адекватно да бъдат класифицирани в онтологията по отношение на реализацията си с езикови конструкции на български език. Това означава, че не е необходима никаква корекция в Обобщения модел, за да бъде той използван в процеса на автоматично генериране на български текстове.

По отношение на един цялостен анализ за съответствие на конструкциите на Обобщения модел и езиковите конструкции на българския език е необходима още сериозна изследователска работа. Можем да твърдим единствено, че е направена първата малка стъпка с въвеждане на терминологията на Системично-функционалния анализ на български език в [Стайкова и Пенчев, 2000]. Това е съществено, тъй като конструкциите на Обобщения модел кореспондират пряко с постановките на Системично-функционалната граматика на [Halliday, 1994], както беше споменато и по-горе в раздел 1.2.3.

3.2 Определяне на съдържанието

Едно от предизвикателствата пред създателите на системата AGILE е автоматичното генериране на кохерентни текстове от съдържание зададено от потребителя на системата. На евентуалния потребител е предоставен интерфейс, чрез който да бъдат съчетани и подредени термини от модела на предметната област на CAD/CAM приложения. Това са понятия за елементи описани в раздел 3.1, например: ЛИНИЯ, ДЪГА, КЛАВИШ, ПОКАЗВАМ, ЧЕРТАЯ, НАТИСКАМ, ПОТРЕБИТЕЛ и т.н.

Потребителят на системата може да съставя текстове- инструкции. За да се групират и подредят някакъв набор от термини на предметната онтология, така че да изразяват търсено съдържание в модела на областта са предоставени и *текст-структуриращи елементи*. От анализа на текстовия корпус изложен в раздел 2.1 става ясно, че целевите текстове-инструкции най-често се състоят от определени изграждащи ги елементи, например цел, списък от стъпки за постигане на целта, странични ефекти и т.н. Чрез входния интерфейс потребителят на системата задава комбинация от определените текст-структуриращи блокчета като ги изпълва с определено съдържание - конкретни понятия от модела на предметната област на CAD/CAM приложенията. Така чрез конкретен набор от текст-структуриращите елементи се описва *какво* да бъде реализирано в изходния текст.

Определянето на съдържанието на бъдещите текстове в системата се извършва чрез затворено планиране, което означава, че то изцяло зависи от решенията и намесата на потребителя на генериращата система. Потребителят задава комбинация от текст-структуриращи елементи и ги запълва със желаното съдържание изразено чрез понятия дефинирани в онтологията на предметната област.

Следващият пример представя структурата от дефинираните текст-структуриращи елементи, с която се определя връзката цел-подцел в изречението „Изберете Save, за да запазите чертежа.“:

```

процедура
  ЦЕЛ 'запазвам чертеж'
  МЕТОДИ списък_от_методи
    ПЪРВИ метод
      ПОД-СТЪПКИ списък_от_процедури
        ПЪРВИ процедура
          ЦЕЛ 'избирам Save'

```

Формално представяне на съдържанието на примерната процедура наречена „Чертане на дъга“ е предоставено в Приложение 2: Формално представяне на съдържанието на процедурен текст ”.

3.3 Планиране на текста

От всеки зададен вход за процеса на генериране се очаква системата AGILE да предостави на изход автоматично генериран кохерентен текст, а ако е възможно и няколко типа автоматично генерирани текстове. Обработването на входното съдържание, т.е. на запълнените текст-структуриращи елементи зададени от потребителя, се извършва от модул за текст-структуриране, чиято работа има два етапа. Първо от зададеното съдържание се създава план на бъдещия изходен текст като цяло, а след това планировчикът на изречения създава и планове на отделните изречения (т.н. SPL-и) на базата на готовия текстов план.

Текст-структуриращите елементи описват само съдържанието на бъдещия текст, т.е. *какво* да бъде реализирано. За да се контролира *как* да бъде реализирано съдържанието са въведени текстови шаблони, които дефинират *стил на текста* като задават параметрите на реализацията в лексико-граматиката за определени елементи на текстовата структура. Характерен пример е заглавието на текста-инструкция съдържащо се в слота ЦЕЛ на текст-структуриращия елемент ПРОЦЕДУРА, което се реализира чрез номинализация на зададения глагол. В края на този раздел са изложени възможностите предлагани от системата AGILE за отразяване на определени стилови предпочитания за изходния текст.

Модулът за текст-структуриране работи с две групи понятия: текст-структуриращите елементи описани в предходния раздел и текстови шаблони, които представляват градивните единици на текстовия план. Текстовите шаблони дават разпределението на съдържанието и дефинициите им са извлечени от терминологията ориентирана към потребителя, към която обикновено се придържат авторите на технически ръководства.

Елементи на текстовите шаблони:

ДОКУМЕНТ-ЗАДАНИЕ

Елементът е дефиниран чрез два слота:

ЗАГЛАВИЕ_на_ЗАДАНИЕТО (задължително)

ИНСТРУКЦИИ_за_ЗАДАНИЕТО (задължително)

ИНСТРУКЦИИ_за_ЗАДАНИЕТО означава поне една инструкция.

ИНСТРУКЦИЯ

Елементът е дефиниран чрез три слота:

ЗАДАЧИ (задължително)

ОГРАНИЧЕНИЕ (незадължително)

ПРЕДУСЛОВИЕ (незадължително)

ЗАДАНИЕ

Елементът е дефиниран чрез два слота:

ИНСТРУКЦИИ (задължително)

СТАРНИЧЕН_ЕФЕКТ (незадължително)

Съпоставянето на текстовите шаблони и текст-структуриращите елементи на входния интерфейс за съдържанието дава възможност за създаване на правила за текст-структуриране.

ЗАГЛАВИЕ_на_ЗАДАНИЕТО	↔	ЦЕЛ на главната ПРОЦЕДУРА
ИНСТРУКЦИИ_за_ЗАДАНИЕТО	↔	МЕТОДИ на ПРОЦЕДУРА
СТАРНИЧЕН_ЕФЕКТ	↔	СТАРНИЧЕН_ЕФЕКТ на ПРОЦЕДУРА
ЗАДАНИЕ	↔	ЦЕЛ на ПРОЦЕДУРА
ИНСТРУКЦИИ	↔	МЕТОДИ на ПРОЦЕДУРА
ИНСТРУКЦИЯ	↔	МЕТОД
ОГРАНИЧЕНИЕ	↔	ОГРАНИЧЕНИЕ на МЕТОД
ПРЕДУСЛОВИЕ	↔	ПРЕДУСЛОВИЕ на МЕТОД
ЗАДАЧИ	↔	ПОД-СТЪПКИ на МЕТОД

Изречението „Изберете Save, за да запазите чертежа.“ може да бъде представено чрез елементите на текстовите шаблони по следния начин:

ЗАГЛАВИЕ_на_ЗАДАНИЕТО “запазване на чертеж”
ИНСТРУКЦИИ_за_ЗАДАНИЕТО
ИНСТРУКЦИЯ
ЗАДАЧИ
ЗАДАНИЕ “избирам Save”

Текстовите шаблони се използват за предварително определяне на избора на граматически средства за реализиране на съдържанието и по този начин може да бъде определян стила на текста. С такава цел в шаблоните могат да се въведат и допълнителни маркери на дискурса, за да се направи експлицитна текстовата структура. Независимо от конкретния текстов шаблон могат да съществуват избори, които да влияят на нивото на експлицитност, например, за информиране на потребителя на генерирания текст за наблюдаваните странични ефекти.

При анализа на текстовия корпус за системата AGILE са забелязани следните характеристики и варианти:

- Явно изразяване или скриване на агента извършващ действието;
- Различни начини да се изрази връзката с читателя – дали читателят да е специално адресиран или не;
- Режим на реализиране на инструкции;
- Сложност на лингвистичните изрази.

Прецизната работа с тези аспекти дава възможност за определяне на вариациите в текстовия стил. Въведени са два различни стила, в които могат да бъдат реализирани инструкциите: *персонален императивен* и *безличен изявителен*. По-конкретно, прави се разлика между заглавието на заданието и неговия стил на реализация от една страна и стъпките на заданието с тяхната реализация в конкретен стил- от друга страна. Така текстовите шаблони трябва да определят следните аспекти:

- Реализация на заглавие на текста чрез номинализация на глаголната група
- Реализация на стъпките-инструкции – ограничения върху TASK-INSTRUCTIONS, които съответстват на стила:

1. персонален императивен: императивно наклонение, учтива форма на императива (2-ро лице, мн. число)
2. безличен изявителен: индикативно наклонение, страдателен залог.

Определени са изразите, които се включват в SPL-заданията, за да настроят различните аспекти на определените стилове. Получаването на различни SPL-конструкции от зададения семантичен вход за различните стилове се контролира чрез средствата за текст-структуриране на средата KPML, а именно изрази REALIZE-WITH. Например, ако искаме да видим съдържанието на елемента TASK-TITLE реализирано с номинална група, то можем да зададем следния израз SPL-конструкцията:

```
(REALIZE-WITH TASK-TITLE
(:EXIST-SPEECH-ACT-Q NOSPEECHACT))
```

тъй като изрази :EXIST-SPEECH-ACT-Q NOSPEECHACT зададен в SPL насочва генерирането на глаголна група към номинализация (виж SPL за Фигура 16 Генерирана структура за „Чертане на линия“).

В допълнение към ограниченията за лексико-граматическите реализации средата KPML дава възможност да се специфицира *layout* за определени елементи на текстовата структура. Чрез функцията ANNOTATE, която може да бъде използвана в SPL-конструкциите определено съдържание там се аотира с елементи на множество от HTML-тагове. Реализацията на функцията осигурява вмъкването на таговете в изходния текст, което се използва при визуализиране на изходните текстове.

Следващата Таблица 9 дава съпоставянето, което се използва за подреждане на генерираните процедурни текстове в AGILE:

Текст-структуриращ елемент	Layout	HTML - тагове
TASK-TITLE	Ниво заглавие <i>N</i>	<HN> ... </HN>
(TASK-) INSTRUCTIONS	Подреден списък	 ...
INSTRUCTION	(Подреден) елемент от списък	
CONSTRAINT	Описателен елемент от списък	<DL> <DD>... </DL>
PRECONDITION	(Подреден) елемент от списък	
SIDE-EFFECT	Описателен елемент от списък	< DL> <DD>... </DL>
TASK	Текст	

Таблица 9 Спецификация на HTML-тагове за текст-структуриращите елементи

Освен ограниченията на текстовия шаблон можем да въвеждаме общи ограничения за реализацията, например, да се генерират ли страничните ефекти, или да останат имплицитни в изходния текст.

Явни странични ефекти:

1. Въведете **a**, за да превключите на режим Arc.
2. Появява се диалоговият прозорец на режима Arc.
3. Изберете ОК.

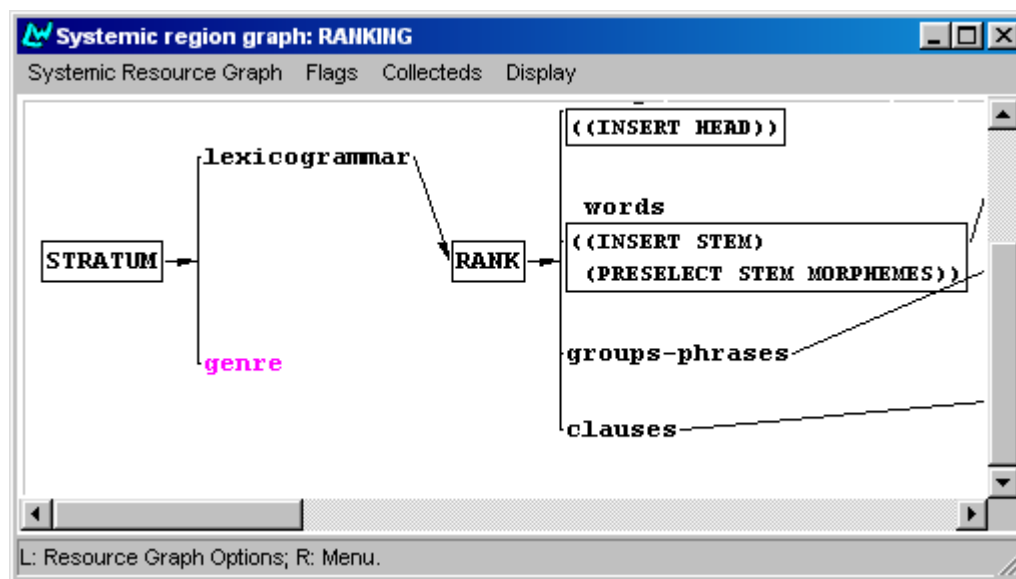
Имплицитни странични ефекти:

1. Въведете **a**, за да превключите на режим Arc.
2. Изберете ОК в диалоговия прозорец на режима Arc.

Модулът за текст-планиране е реализиран в стила на приложните системично-функционални граматики на средата за многоезиково генериране KPML. Създаден е регион CADCAM-INSTRUCTIONS, за да се изложат вариантите на текстовия план и LISP- функция, която претърсва този регион в зависимост от зададения потребителски вход за съдържание на текст.

Полето CADCAM-INSTRUCTIONS задава допълнително ниво на лингвистичния ресурс, нивото „жанр“, което дава възможност за построяване на текстовия план по начин подобен на този използван от граматиката, за да съставя граматически конструкции. Такъв регион за жанр е естествено да бъде поставен в самото начало на тактическия генератор и така да бъде интегриран с граматиките в средата за многоезиково генериране KPML.

На практика полето CADCAM-INSTRUCTIONS се зарежда след лексико-граматическия ресурс заедно с редактираното поле RANKING. За разлика от показаното в раздел 3.1 за начало на процеса на генериране (Фигура 7 Визуализация на системата Ранг), първият избор е вече избор на лингвистично ниво, не на граматически ранг. Фигура 24 по-долу показва региона RANKING, където изборът е между *жанр* (текст) и *лексико-граматика* (изречения).



Фигура 24 Първа система за процеса на генериране: STRATUM

Една част от системите в региона CADCAM-INSTRUCTIONS се занимават с интерпретацията на текст-структуриращите елементи от потребителския вход и водят до определяне на структурата на бъдещия текст, докато друга част от системите са предназначени за обработване на текстовите шаблони и регулиране особеностите на стила на бъдещия текст.

След като разполагаме с план на текста, който включва ограниченията за начина на реализация на неговите съставни части вече могат да бъдат създадени SPL-и за отделните изречения като се използват тези ограничения. Това се извършва от Планировчика на изречения.

Преди да започне работата на Планировчика на изречения на всяка отделна част на потребителския вход е съпоставен SPL-фрагмент за съдържанието, който е същностен (ideational) по отношение на това съдържание, т.е. изразява съдържанието и нищо повече. Към този същностен SPL-код се добавят определените ограничения, също изразени в SPL-термини и конструкции. Полученият SPL би трябвало да реализира съдържанието по начин предопределен от текстовия шаблон, в който е поставено това съдържание.

Резултатът от тази стъпка е едно множество SPL- планове за изречения, които предават съдържанието от потребителския вход в термините на генерираната текст-структура. За да се получи автоматично генерираният кохерентен текст, това множество от планове за изреченията се подава на тактическия реализатор- съответен езиков ресурс в средата за многоезиково генериране KPML.

Една интересна задача на Планировчика на изречения е агрегацията на изреченията. Тук агрегацията представлява всъщност комбиниране на два или повече SPL-фрагмента в по-голям фрагмент от SPL-код. Всяка отделна част определя плана за реализиране на елемент от потребителския вход, който може да има и зададен стил. На практика всеки SPL-фрагмент определя изречение. Когато става въпрос за комбиниране на SPL-и, то трябва да се вземе решение за сложността на изречението, което да изрази зададеното съдържание.

На задачата за агрегация може да се погледне като на задача за определяне границите на изречението. Текст-структуриращите елементи следват начина, по който съдържанието е йерархично зададено чрез входния интерфейс. Да сложим граница на изречението означава да изоставим рекурсивното влагане и да започнем наново от мястото, където сме прекъснали. Така всички вече конструирани SPL-фрагменти ще бъдат комбинирани в един SPL. Определени са два критерия за спиране на процеса на агрегация на SPL-фрагменти:

- ПРОЦЕДУРА, която има слотове ЦЕЛ, МЕТОДИ и СТРАНИЧЕН_ЕФЕКТ
- МЕТОД, който има слотове ОГРАНИЧЕНИЕ, ПРЕДУСЛОВИЕ и ПОД-СТЪПКИ

Планировчикът на изречения преминава през листата на създадената дървовидна структура на текста последователно като създава SPL-код за семантиката идентифицирана в тези листа.

Първо зададеното съдържание се превежда в чисто същностно съответстващ му SPL-код, след това се определят границите на отделните изречения и се формира SPL-код за всяко изречение.

След това се добавят ограниченията в реализацията изисквани от стила на текста чрез описаните по-горе специфични SPL-фрагменти. Така планове на изреченията носят информация не само какво да бъде реализирано, но и как да стане това. По този начин подадените за генериране от тактическия генератор SPL-и на изречения представляват всъщност код на кохерентен текст.

В следващия раздел е показан пример за вариантите на автоматично генерирани текстове на български език от едно конкретно зададено входно съдържание.

3.4 Лексико-граматическа реализация

Стиловото разнообразие на текстовете генерирани на български език е представено в следващата Таблица 10.

Стил на текста	Кратки команди номериран списък	По-дълги изречения	По-дълги изречения с обяснения
Персонален императивен	Вариант 1	Вариант 2	Вариант 4
Безличен изявителен		Вариант 3	+ допълнителна агрегация Вариант 5

Таблица 10 Стил на генерирания текст

Двете основни вариации представени от редовете на таблицата, персонален императивен стил и безличен изявителен стил, се отнасят за начина на реализиране на инструкциите в текстовете, например „Въведете ОК.“ или „Въвежда се ОК“.

Колоните отразяват различни варианти на организацията на текста - представянето на списък от команди като номериран списък с елементи записани всеки на отделен ред, или вариации на агрегацията. Вариациите на агрегацията са всъщност различно разпределение на съдържанието в изречения. Това може да включва генериране на „изречения с обяснения“, или, с други думи, генериране на информацията за страничните ефекти.

Следват генерираните изходни текстове в петте вариации за зададено едно и също примерно съдържание на желанния текст.

Вариант 1: Персонален-императивен текст с кратки команди

Чертане на полилиния, съставена от отсечки и дъги

1. Стартирайте командата PLINE.
2. Задайте началната точка на отсечка.
3. Задайте крайната точка на отсечката.
4. Въведете а. Изберете ОК.
5. Задайте крайната точка на дъгата.
6. Задайте трета точка на дъгата.
7. Въведете I. Изберете ОК.
8. Въведете дължина на отсечка.
9. Въведете ъгъл на отсечката спрямо допирателната в крайната точка на дъгата.
10. Натиснете Return.

Вариант 2: Персонален-императивен текст

Чертане на полилиния, съставена от отсечки и дъги

1. Стартирайте командата PLINE, като използвате един от следните методи:

Windows: От плаващото меню Polyline на функционалния ред Draw изберете Polyline.

DOS и UNIX: От менюто Draw изберете Polyline.

2. Задайте началната точка на отсечката.

3. Задайте крайната точка на отсечката.

4. Въведете **a**, за да превключите на режим Arc. След това изберете ОК в диалоговия прозорец на режима Arc.

5. Задайте крайната точка на дъгата.

6. Задайте трета точка на дъгата.

7. Въведете **l**, за да се върнете в режим Line. След това изберете ОК в диалоговия прозорец на режима Line.

8. Въведете дължина на отсечка от крайната точка на дъгата.

9. Въведете ъгъл на отсечката спрямо допирателната в крайната точка на дъгата.

10. Натиснете Return, за да завършите полилинията.

Вариант 3: Безличен-изявителен текст

Чертане на полилиния, съставена от отсечки и дъги

1. Стартира се командата PLINE, като се използва един от следните методи:

Windows: От плаващото меню Polyline на функционалния ред Draw се избира Polyline.

DOS и UNIX: От менюто Draw се избира Polyline.

2. Задава се началната точка на отсечката.

3. Задава се крайната точка на отсечката.

4. Въвежда се **a** за превключване на режим Arc. След това, в диалоговия прозорец на режима Arc се избира ОК.

5. Задава се крайната точка на дъгата.

6. Задава се трета точка на дъгата.

7. Въвежда се **l** за връщане в режим Line. След това, в диалоговия прозорец на режима Line се избира ОК.

8. Въвежда се дължината на отсечката от крайната точка на дъгата.

9. Въвежда се ъгълът на отсечката спрямо допирателната в крайната точка на дъгата.

10. Натиска се Return за завършване на полилинията.

Вариант 4: Персонален-императивен текст с обяснения

Чертане на полилиния, съставена от отсечки и дъги

Първо начертайте отсечката.

1. Стартирайте командата PLINE, като използвате един от следните методи:

Windows: От плаващото меню Polyline на функционалния ред Draw изберете Polyline.

DOS и UNIX: От менюто Draw изберете Polyline.

2. Задайте началната точка на отсечката.

3. Задайте крайната точка на отсечката.

4. Въведете **a**, за да превключите на режим Arc. Появява се диалоговият прозорец на режима Arc. Изберете OK.

5. Задайте крайната точка на дъгата.

6. Задайте трета точка на дъгата.

7. Въведете **l**, за да се върнете в режим Line. Появява се диалоговият прозорец на режима Line. Изберете OK.

8. Въведете дължината на отсечка от крайната точка на дъгата.

9. Въведете ъгъла на отсечката спрямо допирателната в крайната точка на дъгата.

10. Натиснете Return, за да завършите полилинията.

Вариант 5: Безличен-изявителен текст с обяснения и агрегация

Чертане на полилиния, съставена от отсечки и дъги

Първо се чертае отсечката.

1. Стартира се командата PLINE, като се използва един от следните методи:

Windows: От плаващото меню Polyline на функционалния ред Draw се избира Polyline.

DOS и UNIX: От менюто Draw се избира Polyline.

2. Задава се началната точка на отсечката и се задава крайна точка на отсечката.

3. Въвежда се **a** за превключване на режим Arc. Появява се диалоговият прозорец на режима Arc. Избира се OK.

4. Задава се крайната точка на дъгата и се задава трета точка на дъгата.

5. Въвежда се **l** за връщане в режим Line. Появява се диалоговият прозорец на режима Line. Избира се OK.

6. Въвежда се дължината на отсечка от крайната точка на дъгата и се въвежда ъгъла на отсечката спрямо допирателната в крайната точка на дъгата.

7. Натиска се Return за завършване на полилинията.

В настоящата глава от дисертацията е проследена логиката на реално организиран процес за генериране на текст. Създаденият ресурс за генериране на български изречения, Приложна българска системично-функционална граматика, е използван при тактическото генериране в края на процеса. Обърнато е внимание на семантичните технологии, които имат отношение към реалното конструиране и изпълнение на такъв процес за текст-генериране. Показан е нужния семантичен модел на предметната област на генерирането, показано е как се осигурява съответствие между понятията от този модел и висшата онтология кореспондираща с лингвистичните конструкции за лингвистична реализация на зададените понятия. В под-раздел 2.1 се докладва за изследване на съответствието на понятията от зададената предметна област (формален CAD/CAM свят) и понятията от Обобщения модел, с цел генериране на синтактични конструкции на български език. Споделен е изследователският опит при текст-планирането извършено в рамките на проекта AGILE, за да се генерират 5 различни типа текстове от едно и също зададено семантично съдържание. Демонстрирани са постигнатите резултати при генерирането на кохерентен текст, които са уникални като опит за автоматично генериране на български език.

Глава 4. Работа с български текстове, семантични технологии и ресурси

В тази глава се разглеждат приложения на семантичните технологии за семантично аотиране на специализирани текстове на български език, които биха подпомогнали процес на компютърно генериране на български текстове. Използвани са разработки по проекта СИНУС, който има за цел създаването на семантична платформа за технологично- поддържано обучение чрез динамично композиране на учебните материали. Това изисква да са на разположение поддържащи информационни модели за динамично създаване и адаптиране на учебни обекти, за да се осигури многократното им използване в процеса на обучение.

Сложността при създаването на учебно съдържание в съвременните адаптивни системи за обучение се корени преди всичко в сложността на моделите на предметните области в тези системи. Обещаваща методология е да се разчита на модели на областта, разработени от експерти и да се предоставят на преподавателите удобни средства за създаване на учебно съдържание, като се поддържа ефективното му индексирание [Dochew and Agre, 2009].

Семантичната платформа на проекта СИНУС е тествана с демонстрационни примери за използване на предлаганите от проекта технологии, които включват работа с мултимедийни учебни обекти аотирани с мета-данни за съдържанието им. Разглежданата тук научно-приложна задача се състои в това да се намерят методи за преодоляване на ограниченията на стандартните мета-данни за учебни обекти, а именно, да се използват онтологии, които позволяват пряка компютърна обработка на знанията кодирани в мета-данните.

Чрез платформата на проекта СИНУС се достъпват и използват различни съществуващи мултимедийни библиотеки като в разглежданите тук приложения мултимедийните обекти са от мултимедийната цифрова библиотека “Виртуална енциклопедия на Източно-Християнското изкуство” [Pavlova-Draganova et. al., 2007]. Това е мултимедийната библиотека, която съдържа мултимедийна информация за иконографски обекти (икони, миниатюри, стенописи и т.н.) създадени на територията на България от VII до XIX век. Един мултимедийен обект е съвкупност от дигитални изображения и различни описателни текстове.

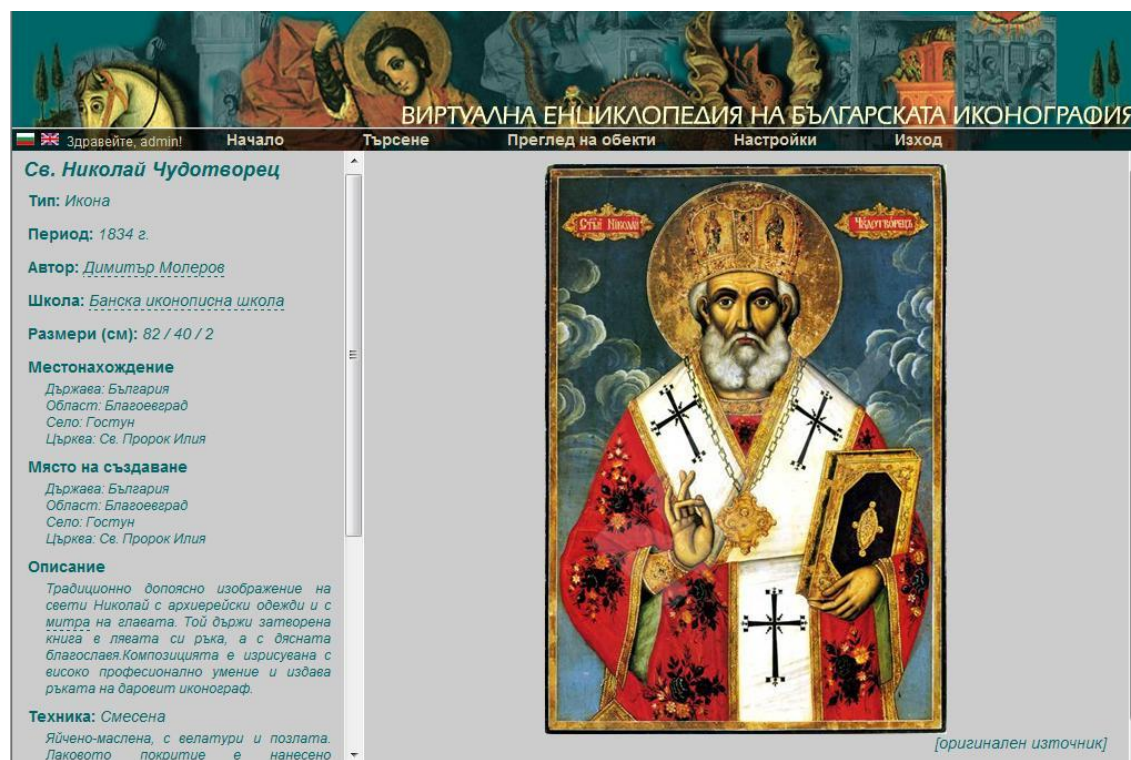
Семантичното търсене от платформата на проекта СИНУС в самата мултимедийна библиотека е осигурено от онтология, която за целите на проекта е наречена *базова онтология*. Възприетият подход осигурява намиране, визуализация и използване на мултимедийно съдържание чрез прилагане на различни схеми на мета-данните описващи мултимедийните обекти от различни перспективи според различните интереси на потребителите. Това се постига чрез формализирани експертни знания структурирани в онтологии, които са условно наречени *специализирани онтологии*.

В раздел 3.1 изложението продължава с описание на направеното по задачата за формализиране на необходимите знания и структурирането им в онтологични конструкции. В раздел 3.2 е описан процеса на семантично аотиране на специализирани български текстове, при което е използван подхода „От онтология към текст” и са приложени вече създадените онтологични конструкции.

4.1 Формализиране на знания от областта на иконографията

Задачата свързана с формализирането на знания от областта на иконографията е породена от задачата за създаване на семантични модели на основните мултимедийни обекти нужни за

обучителната среда на проекта СИНУС. Основните мултимедийни обекти използвани в демонстрационните примери на проекта са ИКОНА, СТЕНОПИС, МИНИАТЮРА. Изображението на Фигура 25 дава представа за тяхното представяне в използваната мултимедийна библиотека.



Фигура 25 Мултимедийен обект: икона „Св. Николай Чудотворец“

Достъпването на такива обекти от платформата на проекта СИНУС се осигурява от създадените в средата семантични модели на мултимедийните обекти. Семантичните модели от своя страна се изграждат върху онтологични конструкции. Спецификата на проекта, както и предварителният анализ на данните доведоха до решение онтологичните концептуални знания за мултимедийните обекти да се групират в една *базова онтология* и три *специализирани онтологии*.

Базовата онтология наречена “Онтология на български иконографски обекти” или ОБИО съдържа основните понятия свързани с иконографските обекти като такива (наименование на иконографско изображение върху обекта, автор на иконографския обект, времеви период на създаване, място на създаване и др.).

В специализираните онтологии се борави с много по-специфични понятия от областта на технологиите за създаване на иконографски обекти. Наименованията на специализираните онтологии са: Специализирана онтология за технология на иконографски обект, Специализирана онтология за лаково покритие и Специализирана онтология за основа на иконографски обект.

Базова онтология ОБИО

През последните години в областта на онтологичното инженерство е извършена значителна по обем работа, в резултат на която се увеличават областите на информационното пространство снабдени с онтологични стандарти при Интернет-обработки на информация. Такива области са медицината, културното наследство на човечеството, управлението на проекти и т.н. Основните

онтологични модели в съответните области се намират в определена фаза от жизнения цикъл на онтоологиите, като фазите са подробно описани в ... След като всеки от тези големи онтологични модели се признае за адекватен от съответната общност от експерти в неговия жизнен цикъл настъпва период на популяризация сред прилагащите го общности и много по-консервативно отношение към фиксираните в онтологията понятия.

За приложните изкуства и по-общо в областта на културното наследство на човечеството такъв фундаментален онтологичен модел е CIDOC – CRM, разработен от Групата по стандартизиране на документацията към Международния съвет на музеите (International Council of Museums - ICOM). От септември 2006-та година онтологията CIDOC CRM е приета за стандарт ISO 21127 на Международната организация по стандартизация (ISO).

В официалната страница на CIDOC CRM¹⁶ е записано, че „моделът представлява онтология за информацията свързана с културното наследство, т.е. той описва на формален език експлицитните и имплицитните понятия и релации свързани с документацията на културното наследство. Основната роля на CIDOC CRM е да служи като база за свързване на информацията за културното наследство и да представлява семантично „лепило“ необходимо при трансформирането на съвременните разпръснати локални информационни източници в кохерентен и стойностен глобален ресурс.”

Затова понятията от базовата онтология на проекта СИНУС- ОБИО са създадени да бъдат съвместими с понятията дефинирани в CIDOC CRM. Това е нужно при достъпване на хетерогенни данни от различни източници в Интернет-пространството. Така използваният в средата СИНУС семантичен модел е пре-използваем и достъпен за всеки проект базиран на CIDOC CRM, като публикуваните знания могат да се обработват от субекти в широкото Интернет-пространство.

В следващата йерархия на класовете на CIDOC-CRM, версия 5.0.4 [Crofts et.al, 2011] е посочено мястото на централното понятие за ОБИО-СИНУС Иконографски обект/ Iconographical Object.

E1 CRM Entity
E2 - Temporal Entity
E4 - Period
E5 - Event
E7 - Activity
E11 - Modification
E12 - Production
E13 - Attribute Assignment
E65 - Creation
E63 - Beginning of Existence
E12 - Production
E65 - Creation
E64 - End of Existence
E77 - Persistent Item
E70 - Thing
E72 - Legal Object
E18 - Physical Thing
E24 - Physical Man-Made Thing
E90 - Symbolic Object
E71 - Man-Made Thing
E24 - Physical Man-Made Thing
E22 - Man-Made Object
OBIO - Iconographical Object
E84 - Information Carrier

¹⁶ <http://www.cidoc-crm.org/>

E25 - - - - - Man-Made Feature
 E78 - - - - - Collection
 E28 - - - - - Conceptual Object
 E39 - - Actor
 E74 - - - Group
 E52 - Time-Span
 E53 - Place
 E54 - Dimension
 E59 Primitive Value

Следващото изложение е организирано в стила и според терминологията на CIDOC CRM [Crofts et.al, 2011]. Построяването на онтологията ОБИО е на практика специализация на класове на CIDOC CRM (отбелязвани тук както в специализираната литература с Е и число) и на релации на CIDOC CRM (отбелязвани с Р и число). За улесняване работата по концептуализацията на понятията от онтологията ОБИО в рамките на сложния модел на CIDOC CRM условно обособихме три ключови концептуални области за онтологията ОБИО:

- Свойства на *Иконографски обект* като материален обект;
- Знания за създаването на *Иконографски обект*;
- Религиозно значение на *Иконографски обект*.

Първата концептуална област формализираща същността на Иконографски обект като материален обект от описвания свят обхваща следните класове и релации:

- Клас Иконографски обект и неговите под-класове: Икона, Стенопис, Миниатюра, Витраж, Мозайка, Пластичен иконографски обект, както и под-класовете на Пластичен иконографски обект: Иконостас и Престол.
- Свойството P102 has title (is title of): E35 Title свързва Иконографски обект с неговото Наименование/Title.
- Свойството P129 is subject of (is about): E89 Propositional Object използваме, за да свържем Иконографския обект с Идентификационни бележки за него, с негово Описание и с текст за състоянието му Текущо състояние/ Condition State, които по същество са Лингвистични обекти (следователно и Пропозиционални обекти). Тези три типа текстови или лингвистични обекти въвеждаме като под-класове на E33 Linguistic Object.
- Свойството P43 has dimension (is dimension of): E54 Dimension използваме, за да свържем Иконографския обект с измеренията Височина, Ширина, Дебелина, които въвеждаме като под-класове на E54 Dimension. Всъщност, по-удачно е да свържем тези три измерения с клас Икона, тъй като другите под-класове на Иконографски обект (например Стенопис, Мозайка) могат да бъдат описани с други измерения. За измеренията Височина, Ширина, Дебелина използваме свойството P91 has unit (is unit of): E58 Measurement Unit, за да ги свържем с мерната единица cm използвана за съответните данни от базовата цифрова библиотека.
- Свойството P46 is composed of (forms part of): E18 Physical Thing дава възможност да въведем знания за съставни части на Иконографския обект като физически обект. Дефинираме Основа като под-клас на E24 Physical Man-Made Thing. Сега можем да свържем екземпляри от клас Основа с техни описания чрез свойството P129 is subject of (is about): E89 Propositional Object, като въведем под-клас на E89 Propositional Object (и на E33 Linguistic Object) Описание на основата/ Base Description.

- Свойството P45 consists of (is incorporated in): E57 Material е характерно за екземпляри на класа E18 Physical Thing, следователно може да бъде приложено към Иконографски обект, както и към Основа на иконографски обект. Предмет на детайлизация е класът E57 Material.
- Свойството P137 exemplifies (is exemplified by): E55 Type -> Iconographic School осигурява връзката между Иконографски обект и Иконописната школа, чиито продукт е обекта. Школата се разглежда като съвкупност от стилови характеристики, демонстрирани от нейните продукти. Иконописна школа е формално под-клас на E55 Type и съдържа като екземпляри различните школи (Тревненска, Банско-Разложка и т.н.).
- Свойството P53 has former or current location (is former or current location of): E53 Place свързва Иконографския обект с място където той се е намирал или се намира в момента.
- Свойството P108 was produced by (has produced) E12 Production свързва нещо, което е E24 Physical Man-Made Thing, включително Иконографски обект, с процеса на неговото създаване.
- Свойството P65 shows visual item (is shown by): E36 Visual Item, свързва физическия Иконографски обект с изображението върху него разглеждано като концептуално понятие. Иконографско изображение е под-клас на E38 Image.
- Свойството P138 has representation (represents) E36 Visual Item представя връзката на кой да е елемент от описвания свят с негово Изображение. Използваме това свойство, за да формализираме връзката между Иконографски обект и неговото цифрово изображение, което се пази в базовата цифрова библиотека. Цифрово изображение/ Digital Image е под-клас на E36 Visual Item.

Във втората концептуална област представяме знанията за създаването на Иконографски обект чрез класа Iconographical Object Production, под-клас на E12 Production. Всеки екземпляр на новия клас наследява свойствата на всички съдържащи го класове E12 Production, E11 Modification, E63 Beginning of Existence, E7 Activity, E5 Event, E4 Period, E2 Temporal Entity и накрая CRM Entity. За Света на ОБИО-СИНУС са съществени някои от тези свойства, например, природата на създаването на иконографски обект като Времеви елемент/ E2 Temporal Entity. Другите важни свойства са:

- Свойството P108 has produced (was produced by): E24 Physical Man-Made Thing, което представлява връзка между процеса на създаване на иконографски обект и конкретен резултат на този процес, т.е. Иконографски обект, като под-клас на E24 Physical Man-Made Thing.
- Свойството P14 carried out by (performed): E39 Actor свързва процеса на създаване на иконографски обект с неговия изпълнител. Класът E39 Actor има дефинирани под-класове E74 Group и E21 Person в CIDOC-CRM. Въвеждаме под-клас на E21 Person: Иконограф/ Iconographer и под-клас на E74 Group: Иконографски род/ Iconographical Clan. Принадлежността на даден иконописец към определен Иконографски род може да бъде изразена чрез свойството P107 has current or former member (is current or former member of), което свързва екземпляр на класа E74 Group с екземпляр на класа E39 Actor и показва принадлежност на личността към групата. Ако за даден иконописец разполагаме с негова биография в базовата цифрова библиотека можем да покажем това и в онтологията чрез свойството P129 is subject of (is about): E89 Propositional Object, като създадем под-клас на E33 Linguistic Object (който е под-клас на E89 Propositional Object): Биография / Biography.

- Свойството P32 used general technique (was technique of): E55 Type свързва процеса на създаване на иконографски обект с иконографската техника, използвана за него. Като под-клас на E55 Type въвеждаме Иконографска техника. Ако за дадена Иконографска техника разполагаме с нейно текстово описание в базовата цифрова библиотека можем да покажем това и в онтологията чрез свойството P129 is subject of (is about): E89 Propositional Object, като създадем под-клас на E33 Linguistic Object (който е под-клас на E89 Propositional Object): Описание на иконографска техника/ Iconographical Technique Description.
- Свойството P7 took place at (witnessed): E53 Place свързва процеса на създаване на иконографски обект с мястото, където е бил извършен този процес; това е място на създаване за Иконографски обект, създаден при конкретния процес на изработка.
- Свойството P10 falls within (contains): E4 Period дава възможност да посочим периода, през който е протичал процеса на създаване на иконографски обект. Ако имаме въведен екземпляр на E4 Period наречен Българско възраждане, можем да покажем, че даден Иконографски обект е Възрожденски иконографски обект, тъй като е създаден в този период.
- Свойството P4 has time-span (is time-span of): E52 Time-Span свързва процеса на създаване на иконографски обект с времеви интервал дефиниран с начало, край и продължителност.
- Третата условна концептуална област за Иконографски обект описва връзката на Иконографския обект със съдържанието на религиозното изображение, на което той е носител. Свойството свързващо Иконографски обект и Иконографско изображение е P65 shows visual item (is shown by).
 - За класа Иконографско изображение са важни свойствата:
- Свойството P137 exemplifies (is exemplified by): E55 Type -> Iconographic School характеризира Иконографското изображение чрез Иконописната школа, чиито продукт е изображението. Школата се разглежда като съвкупност от стилови характеристики и е формално под-клас на E55 Type.
- Всяко Иконографско изображение изобразява или личност значима за Източно-Християнската религия или събитие, значимо за Източно-Християнската религия. Използваме свойството P138 represents (has representation): E1 CRM Entity, за да покажем кои събития и кои личности са изобразени на Иконографски изображения. За целта създаваме двата класа Личности значими за Източно-Християнската религия и Събития значими за Източно-Християнската религия. Първият клас е под-клас на E21 Person/ Личност, вторият - на E5 Event. За да обособим концептуалното пространство на религиозния свят можем да свържем двата нови класа с нов под-клас на E55 Type Неща значими за Източно-Християнската религия чрез свойството P2 has type (is type of): E55 Type.
- Характерно за Иконографските изображения е, че те следват някакъв канон. Тази характеристика моделираме със свойството P2 has type (is type of): E55 Type, което води към нов под-клас на E55 Type -> Каноничен тип на иконографско изображение.

Екземпляри на този клас са Изображение на светец, Изображение на мъченик, Изображение на Дейсис с Апостолите, Свети Уброс, Христос Вседържител и т.н.

Обособени са под-класове на класа Иконографско Изображение:

- Еднофигурна Композиция
- Многофигурна Композиция
- Композиция от завършени композиции
- Свойството P148 has component (is component of): E89 Propositional Object дава възможност да се свържат завършените композиции, които съставят Композиция от завършени композиции. Същото свойство се използва, за да се покаже, че дадена личност е изобразена като участник в религиозна сцена.

С добавените специализации на класове за ОБИО-СИНУС йерархията на класовете изглежда така:

E1 CRM Entity
E2 - Temporal Entity
E4 -- Period
E5 --- Event
[OBIO -- Important Event for ECR](#)
E7 ---- Activity
E11 ----- Modification
E12 ----- Production
[OBIO ----- Iconographical Object Production](#)
E13 ----- Attribute Assignment
E65 ----- Creation
E63 ----- Beginning of Existence
E12 ----- Production
[OBIO ----- Iconographical Object Production](#)
E65 ----- Creation
E64 ----- End of Existence
E77 - Persistent Item
E70 -- Thing
E72 --- Legal Object
E18 ---- Physical Thing
E24 ----- Physical Man-Made Thing
E90 ---- Symbolic Object
E71 --- Man-Made Thing
E24 ----- Physical Man-Made Thing
[OBIO ---- Base of Iconographical Object](#)
E22 ----- Man-Made Object
[OBIO ----- Iconographical Object](#)
[OBIO ----- Icon](#)
[OBIO ----- Wall-Painting](#)
[OBIO ----- Miniature](#)
[OBIO ----- Mosaic](#)
[OBIO ----- Витраж](#)
[OBIO ----- Plastic Iconographical Object](#)
[OBIO ----- Иконостас](#)
[OBIO ----- Престол](#)
E84 ----- Information Carrier
E25 ----- Man-Made Feature
E78 ----- Collection
E28 ---- Conceptual Object
E90 ----- Symbolic Object
E73 ----- Information Object

E33 ----- Linguistic Object
 E35 ----- Title
 E36 ----- Visual Item
 E38 ----- Image
 OBIO ----- Iconographical Image
 OBIO ----- One Figure Composition
 OBIO ----- Many-figures Composition
 OBIO ----- Composition of Complete Compositions
 E41 ----- Appellation
 E42 ----- Identifier
 E35 ----- Title
 E89 ----- Propositional Object
 E73 ----- Information Object
 E33 ----- Linguistic Object
 E35 ----- Title
 OBIO ----- Biography
 OBIO ----- IO Identification Note
 OBIO ----- IO Description
 OBIO ----- Iconographical Technique Description
 OBIO ----- Base Description
 OBIO ----- IO Condition State
 E36 ----- Visual Item
 E38 ----- Image
 OBIO ----- Iconographical Image
 OBIO ----- One Figure Composition
 OBIO ----- Many-figures Composition
 OBIO ----- Composition of Complete Compositions
 E55 ----- Type
 E56 ----- Language
 E57 ----- Material
 E58 ----- Measurement Unit
 OBIO ----- Iconographic School
 OBIO ----- Iconographical Technique
 OBIO ----- Canonic Type of Iconographical Image
 E39 -- Actor
 E74 --- Group
 OBIO --- Iconographic Clan
 E21 --- Person
 OBIO --- Iconographer
 OBIO --- Important Person for ECR
 E52 - Time-Span
 OBIO - Year
 OBIO - Month
 OBIO - Day
 OBIO - Century
 OBIO - Part of Century
 E53 - Place
 OBIO - State
 OBIO - Region
 OBIO - Town
 OBIO - Village
 OBIO - Monastery
 OBIO - Church
 OBIO - Chapel
 OBIO - Museum
 OBIO - Gallery
 E54 - Dimension
 OBIO - Height
 OBIO - Width

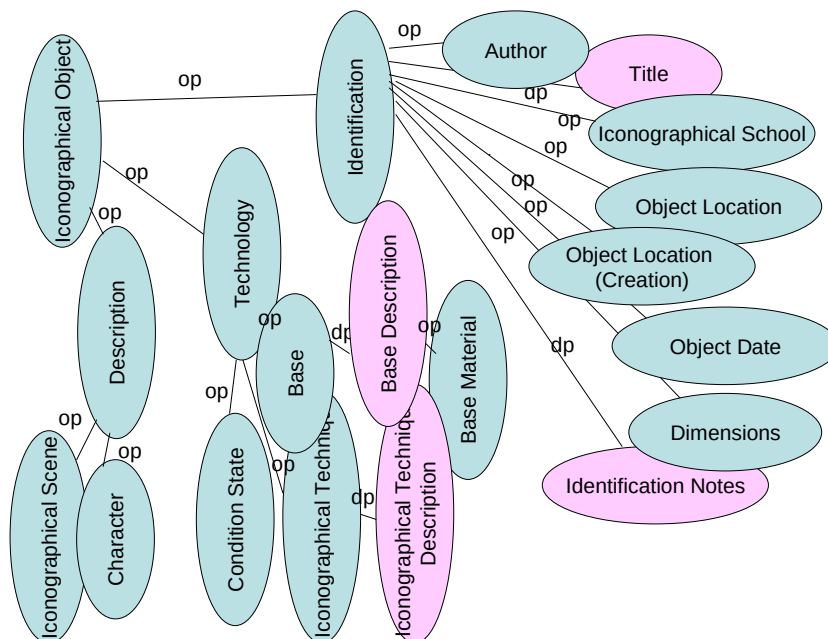
Приложение 3: Онтологии в OWL на предметна област „Иконография“ Приложение 3: Онтологии в OWL на предметна област „Иконография“ съдържа представянето на онтологията OBIO на онтологичния език OWL.

Специализирани онтологии

Конструкциите на трите специализирани онтологии са формирани с участието на експерти в областта на иконографията [Pancheva-Marinova et.al., 2010]. Приложение 3: Онтологии в OWL на предметна област „Иконография“ съдържа формалното програмно представяне на трите онтологии на езика OWL във формата, в който са използвани в сценария за експлоатация на средата Синус. Например, онтологията “Технология на иконографски обект” се състои от йерархия на класовете, йерархия на релациите и индивиди. Понятията в йерархията на класовете са 16, релациите са 14, индивидите на онтологията са 45. За разлика от базовата онтология понятията от специализираните онтологии не са свързани с понятията от онтологията CIDOC CRM и са използвани локално за експерименти с български текстове, както е показано в следващия раздел.

4.2 Семантично аотиране на специализирани български текстове

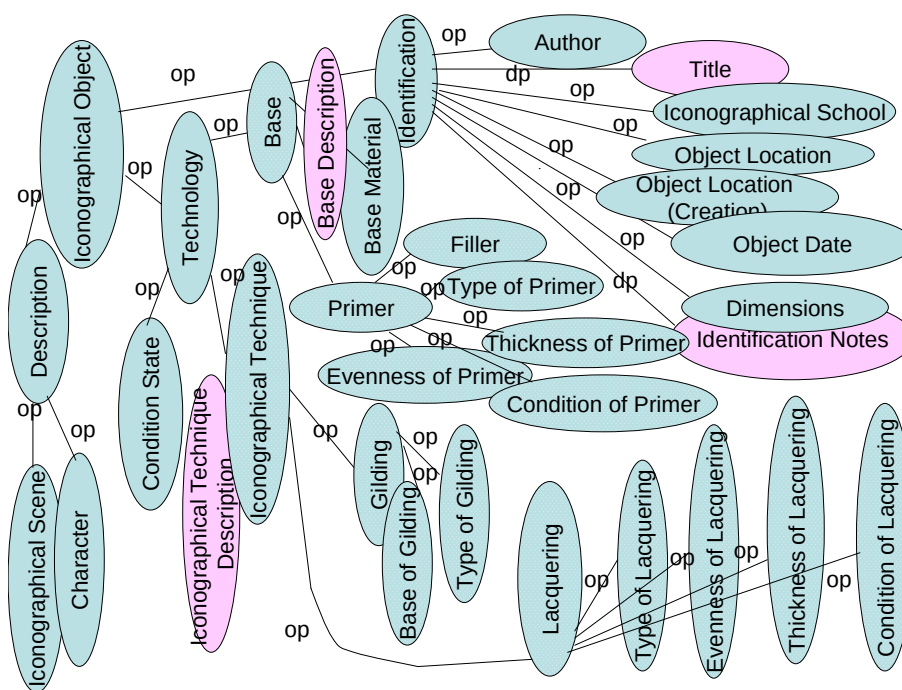
В този раздел е описано решението на задача за семантично аотиране на специализирани текстове на български език. В приложния контекст на проекта Синус задачата се дефинира така: Работим с мултимедийни обекти, които аотирани в семантичната платформа Синус представляват екземпляри на класа *Иконографски обект* от базовата онтология ОБИО.



Фигура 26 Базов семантичен модел

Разполагаме с Базов семантичен модел и Разширен семантичен модел на *Иконографски обект*, поддържани от описаните в предходния раздел онтологии. Фигура 26 и Фигура 27 изобразяват съответно Базовия семантичен модел и Разширения семантичен модел на понятието „иконаграфски обект“.

В Базовия семантичен модел връзките между понятията представляват или обектна релация или релация за данни. Обектната релация свързва две онтологични понятия. Релациите за данни предоставят достъп до текстови данни, които представляват кратки описателни текстове на български език за конкретния иконографски обект – икона, стенопис, миниатюра. Текстовете са част от описанията на мултимедийни обекти от цифровата библиотека „Виртуална енциклопедия на българската иконография [Pavlova-Draganova, et.al., 2007].



Фигура 27 Разширен семантичен модел

Задачата за семантично аотиране е насочена именно към тези текстове. Тя се състои в това в описателните текстове на български език да бъдат добавени анотации, които фиксират всички споменавания на онтологични понятия от специализираната онтология „Технология на иконографски обект“.

Тези семантични анотации след това позволяват полуавтоматично разширяване на семантичния модел на мултимедийния обект в семантичното пространство на средата Синус, където семантичното аотиране е част от по-голяма прагматична задача. Чрез разширения семантичен модел става възможно семантично търсене на аотирани мултимедийни обекти в платформа за технологично поддържано обучение [Agre, 2012], [Dochev and Agre, 2012].

Например, за показания по-горе мултимедийен обект описващ икона на Св. Николай Чудотворец, текстът достъпен от базовия семантичен модел чрез свойството за данни *Iconographical Technique Description* е следният:

„Позлатата по ореолите и по ограждащата рамка е с листово злато (варак).”

След извършване на семантичното аотиране на текста с понятията от специализираната онтология „Технология на иконографски обект” в текста са разпознати следните понятия:

```
<Позлатата> "#OWLClass_Gilding"  
по  
ореолите  
и  
по  
ограждащата  
рамка  
е  
с  
<листово  
злато> "#OWLIndividual_TypeOfGilding01"  
<(варак)> "#OWLIndividual_TypeOfGilding01"  
.
```

Това дава възможност в семантичното пространство на проекта Синус да се използва разширен семантичен модел на иконата. Така мултимедийният обект може да бъде търсен и по тази специфична характеристика (в случая: позлата с листово злато), която не е съществувала в неговата базова семантична аотация. Характеристиката е извлечена директно от описателния текст за иконографския обект чрез семантичното аотиране на текста.

Семантично аотиране на български текстове чрез следване на релацията “От онтология към текст”

Подходът за семантично аотиране чрез следване на релацията “От онтология към текст” е представен в обзорната част на дисертацията. Използването му в конкретната задача за аотиране дефинирана в проекта Синус е изключително подходящо, защото са изпълнени всички изисквания за прилагане на този подход:

- 1) в семантичното пространство на Синус-платформата се използва *онтологията* „Технология на иконографски обект”;
- 2) онтологичните й понятия са лексикализирани на български език и на тази основа е възможно създаването на *терминологичен лексикон* на български език;
- 3) могат да бъдат построени *частични граматики* на български език, които да разпознават споменавания на онтологичните понятия в българските текстове и да ги аотират като онтологични термини.

Настройване на средствата за семантично аотиране

Прилагането на метода “От онтология към текст” започва със съставяне на онтологичен лексикон. В *терминологичния лексикон* се записват лексикализациите на наименованията на онтологичните понятия в дефинициите на онтологията. Например, в онтологията е дефиниран клас с уникален идентификатор *#OWLClass_Lacquering* и лексикализации за наименованието на класа на български език: *лак* и *лаково покритие*.

```

<owl:Class rdf:about="#OWLClass_Lacquering">
  <rdfs:label xml:lang="bg">лак</rdfs:label>
  <rdfs:label xml:lang="bg">лаково покритие</rdfs:label>
  ...
</owl:Class>

```

Лексиконът има два дяла, като единият съдържа наименования на онтологичните индивиди, другият - наименованията на онтологичните класове. Към тези думи и словосъчетания са добавени някои вариации. Например, онтологичният индивид FillerPlaster е понятие лексикализирано като „гипс” в онтологичната дефиниция, докато в лексикона е добавено и словосъчетание „гипсов грунд”. Онтологичният индивид ThicknessOfPrimer01 е понятие лексикализирано в онтологичната дефиниция като „Тънко нанесен грунд”, докато в лексикона е добавена фразата „грунд # нанесен # тънко”, където символът # замества други възможни думи, разширяващи фразата.

Анотационни граматики

За реализиране на анотационните граматики е използвана системата CLaRK [Simov et.al., 2001] предназначена за създаване и работа с корпуси от XML-документи. Името на системата идва от съкращение на Computational Linguistics and Represented Knowledge. Разработването на CLaRK е дело на група изследователи от Лабораторията по лингвистично моделиране към Института по информационни и комуникационни технологии на Българската академия на науките, работили в тясно сътрудничество с колеги от Лингвистичния семинар на Университета на Тюбинген „Еберхард Карлс” в Германия.

Системата CLaRK разполага с инструмент за работа с крайни автомати, който се използва за проверка на валидността на XML-документите, в токанизаторите и в каскадните регулярни граматики. Регулярните граматики стандартно се формализират като множество от преписващи правила върху терминални и нетерминални символи. Граматиките се прилагат върху дума от терминални символи и я парсират с цел да се достигне до крайното състояние. Всички думи, които успешно са парсирани до крайното състояние, образуват един регулярен език. Регулярните граматики не са достатъчно изразителни, за да бъдат добър модел за естествените езици, но имат широко приложение в Обработването на естествен език и най-често се използват за моделиране на морфологични знания и за токанизация.

За системата CLaRK регулярна граматика е множество от правила, такива че регулярните изрази на правилата разпознават съответстващи си непресичащи се езици. Условието за непресичане е необходимо на граматиката, за да присвои уникална категория на всяка разпозната дума. Регулярната граматика работи над думи от букви/символи, наречени входни думи.

Граматиката сканира входната дума от ляво надясно и се опитва да намери първа под-дума, такава, която да принадлежи на езика на регулярните изрази представени от някое правило на граматиката. Ако няма такава дума, която да започва с първата буква на входната дума, граматиката извежда първата буква на изход и продължава по същия начин с втората буква от входната дума. Когато граматиката разпознае под-дума, тя дава на изход категорията от съответното правило и продължава сканирането от буквата след разпознатата под-дума.

Граматиката работи детерминистично над входната дума. Резултатът от прилагането ѝ е копие на входната дума, в което разпознатите под-думи са заменени с категории на граматиката. Резултатът е наречен изходна дума за граматиката. В този смисъл този вид регулярни граматики могат да бъдат наречени крайни трансдюсери [Simov et al., 2002].

Граматиките в системата CLaRK се състоят от правила, чиято структура е дефинирана със следния фрагмент от DTD (Дефиниция на тип документ):

```
<!ELEMENT line (LC?, RE, RC?, RM, Comment?)>
<!ELEMENT LC (#PCDATA)>
<!ELEMENT RC (#PCDATA)>
<!ELEMENT RE (#PCDATA)>
<!ELEMENT RM (#PCDATA)>
<!ELEMENT Comment (#PCDATA)>
```

Всяко правило е представено като линеен елемент и се състои от регулярен израз (RE) и маркер (RM = return markup). Регулярният израз се съпоставя на съдържанието на даден XML-елемент и би могъл да „разпознае“ отделни елементи и/или анотирани данни. Изходният маркер е представен като XML-фрагмент, който замества разпознатата част от съдържанието на елемента. Допълнително потребителят може да използва регулярни изрази, за да ограничи контекста, в който е осъществено съвпадение с регулярния израз. Елементът LE съдържа регулярен израз за левия контекст, а елементът RE- за десния контекст. За удобство на потребителите е добавен и елемент Comment.

За настройване на системата CLaRK е необходим корпус от текстове, за да бъде разширен лексикона и да бъде „обучена“ анотационната граматика в разпознаване на срещанията на лексическите елементи в целевите текстове. В общия случай е важно текстовете от корпуса действително да съдържат лексикализации на онтологичните понятия, които ще бъдат разпознавани.

Най-подходящо е обучителните текстова да са близки до тези, които ще бъдат анотирани, най-малко по отношение на формите и контекстите, в които се срещат лексически елементи на целеви онтологични термини. Това помага и при елиминиране на многозначността на различните нива (думи, фрази и т.н.). От тази гледна точка изключително полезни са тематично групирани текстове от съответната област на специализираната онтология.

В работата по настройване на анотационните граматики на CLaRK са използвани част от наличните описателни текстове на български език достъпни чрез свойствата за данни на базовия семантичен модел:

- 1)#OWLObjectProperty_base_has_Description с описания на основата на иконографски обект,
- 2)#OWLObjectProperty_iconographicalTechnique_has_Description с описания на иконографската техника характерна за иконографския обект,
- 3)#OWLObjectProperty_conditionState_has_Name с описание на текущото състояние на иконографски обект.

Корпусът се състои от 434 описателни текста, разпределени според съдържанието си както следва:

Описания на основата на иконографски обект	189 текста
Описания на иконографска техника	124 текста
Описания на състоянието на иконографски обект	121 текста

За съставянето на златен стандарт е извършено ръчно маркиране на срещанията на онтологичните термини в текстовете. Златният стандарт задава целта, към която се стремим с

настройването на програмата: маркираните срещания на онтологични елементи в текстовете от корпуса да се разпознаят автоматично от анотационната граматика.

В корпуса са отбелязани 899 срещания на търсените термини, т.е. общ брой лексикализации за индивиди и класове от онтологията „Технология на иконографски обект”. Следващата Таблица 11 показва езиковите вариации и броя срещания на термините в трите типа текстове (с точки е означено наличието на други думи, които не са цел на разпознаването).

URI	Срещания на термина в текстовете на български език	Брой срещания в различните типове текстове		
		Тип 1 189 текста	Тип 2 124 текста	Тип 3 121 текста
Primer	грунд, грунда, грундът грундовото покритие грундиран	102	2	17
Gilding	позлата, позлатата златното покритие позлатен, позлатени златен, златният (компонент) златна, златната (декорация) златно, златни злато златист	15	94	11
TypeOfGilding01	листово злато варак	–	104	–
Laquering	лаково покритие лаковото покритие лак, лака, лакът	–	89	12
FillerPlaster	гипс гипсов грунд, гипсов ... грунд грундът е гипсов ... грундовото покритие е гипсово грундовото покритие е гипсово ...	96	–	1
ThicknessOfPrimer 01	...грунд, нанесен тънко ...грунд, нанесен тънко... ...грунд, нанесен относително тънко... тънкослойно нанесен грунд грундът е ... нанесен тънкослойно грундът е ...положен тънкослойно грундът е гипсов, положен тънко грундът е ... нанесен тънко грундът е ... нанесен тънко ... грундът е ... нанесен на тънък пласт ... грундовото покритие ... е тънко нанесено грундовото покритие е ... нанесено тънко ...	74	–	–

ThicknessOfLacquer04	тънко лаково покритие лаковото покритие е нанесено тънко лаковото покритие е нанесено тънко... лаковото покритие ... е нанесено тънко... лаковото покритие е тънко... лакът е положен тънко ...	—	62	—
EvennessOfLacquer02	лаковото покритие е нанесено ... равномерно лаковото покритие ... е нанесено равномерно ... лаковото покритие е ... равномерно. лакът е положен ... равномерно	—	58	—
EvennessOfPrimer01	...грунд, нанесен ... равномерно ...грунд, нанесен равномерно грундът е нанесен равномерно ... грундът е ... нанесен ... равномерно грундът... е нанесен...равномерно грундовото покритие е...нанесено равномерно ...	38	—	—
BaseOfGilding01	полимент	—	19	—
EvennessOfLacquer01	лаковото покритие е нанесено неравномерно лаковото покритие е нанесено ... неравномерно лаковото покритие е нанесено неравномерно ... лаковото покритие е нанесено ... неравномерно ... лаковото покритие е ... неравномерно нанесено ... лаковото покритие е неравномерно нанесено ... лакът е нанесен на неравномерен слой.	—	16	1
ThicknessOfLacquer01	лаковото покритие е нанесено дебело ... лаковото покритие ... е нанесено сравнително дебело ... лаковото покритие е нанесено на дебел слой лаково покритие с много дебел слой	—	10	1
ThicknessOfPrimer03	... грунд, нанесен неколкратно грундът е ... нанесен многослойно грундът...е нанесен неколкратно грундовото покритие е ... нанесено неколкократно	10	—	—
TypeOfGilding02	точно злато точно трито злато	1	7	—
ConditionOfLacquer08	потъмнялото лаково покритие лакът е ... потъмнял ... лаковото покритие е потъмняло...	—	5	3

EvennessOfPrimer03	грундът е ... добре шлифован грундът ... е фино шлифован грундът е ... нанесен ... добре шлифовано грундовото покритие е ... добре шлифовано	7	–	–
ConditionOfLacquering02	лакът е силно замърсен ... лакът е ... силно замърсен лакът ... силно замърсени ... лаковото покритие ... е замърсено лаковото покритие е силно замърсено ...	–	3	3
Semi-OilPrimer	полумаслен свързвател	1	4	–
EmulsionPrimer	емулсията свързващата емулсия	–	4	–
ThicknessOfPrimer02	... грунд, нанесен ... дебело грундовото покритие е ... дебело ...	2	2	–
ConditionOfPrimer03	грундът е ... разрушен ... наранен грундът ... повреден грунда грундът е ... повреден ...	1	–	3
FillerChalk	грундът е кредов ... грундовото покритие е кредово ...	3	–	–
ThicknessOfLacquering02	лаковото покритие е нанесено многократно ... лаковото покритие е нанесено на два слоя лаковото покритие е нанесено ... на няколко слоя	–	3	–
ConditionOfPrimer05	грундът е ... сравнително здрав...	–	2	1
ConditionOfPrimer01	грундът ... е в добро състояние грундът е ... добре запазен	2	–	–
ConditionOfLacquering03	широки натициания от златистокафяв лак лаковото покритие е ... и се установяват натициания	–	2	–
ConditionOfLacquering09	лаковото покритие ... протрито лаковото покритие ... протрито...	–	2	–
OilPrimer	маслен свързвател	–	1	–
EvennessOfPrimer02	грундът е ... нанесен неравномерно ...	1	–	–
ConditionOfLacquering01	лаковото покритие е ... в добро състояние.	–	1	–
ConditionOfLacquering04	лаковото покритие е лошо нанесено ...	–	1	–

Таблица 12 Целеви срещания на лексикализациите на онтологичните термини в корпуса

Създаване на анотационни граматики

Както е споменато по-горе, граматиките в системата CLaRK се състоят от правила представени чрез регулярен израз и маркер. За решаване на конкретната задача са създадени две граматики - едната разпознава индивидите от онтологията, а другата – онтологичните класове. Изграждането на всяка от граматиките започва с елементите на съответния лексикон. Всеки

елемент на лексикона се лематизира с „Българския морфологичен лексикон” [Попов, Симов, Видинска, 1998] и лематизираната форма се превръща в регулярен израз на съответното граматическо правило.

Съществената част от настройването или „обучението” на граматиката се състои в лематизирането на регулярните изрази, които трябва да разпознават търсените фрази в текстовете. Най-простият случай е обобщаване или заместване на част от фразата с подходящ системен символ. Например, фразата „грубо обработен грунд”, която е лексикализация на онтологичния индивид `OWLIndividual_EvennessOfPrimer04`, може да се лематизира като:

```
<RE><"грубо">, <"обработен">, <"грунд"></RE>
```

Този регулярен израз, обаче, няма да разпознае, например, фразата „грубо нанесен грунд”. Следователно, уместно е да се направи следната лематизация:

```
<RE><"грубо">, <"#">*, <"грунд"></RE>
```

която осигурява разпознаване на съчетание от думите „грубо” и „грунд”, дори когато те са разделени и от повече думи/ елементи на изречението.

Интересен е случаят с често срещаните в текстовия корпус фрази от типа:

Лаковото покритие е нанесено тънко и равномерно.

Това са случаи, при които в една фраза трябва да бъдат разпознати два индивида на онтологията - за примера това са: `ThicknessOfLacquering04`, лексикализиран като „тънко лаково покритие” и `EvennessOfLacquering02`, лексикализиран като „равномерно лаково покритие”. Според логиката на крайните автомати, които са ядрото на системата `CLaRK`, двата регулярни изрази, построени за разпознаване на съответните индивиди:

```
<RE><"лаково">, <"покритие">, <"#">*, <"тънко"></RE>
```

```
<RE><"лаково">, <"покритие">, <"#">*, <"равномерно"></RE>
```

не могат едновременно да върнат нужния резултат и при еднократно парсиране на текста ще бъде избрано само едно от двете разпознавания. За да имаме и двете разпознавания правилата могат да бъдат разделени в различни граматики. Тогава парсираният с едната граматика текст ще бъде маркиран с първия маркер, а парсиран с втората граматика същият текст ще получи още един маркер.

Това решение се базира на използваната в системата `CLaRK` XML-технологията, която позволява подобни каскадни обработки на XML-документи по отношение на различни маркери и тагове в тях. Чрез отделяне на някои от правилата на граматиката за разпознаване на индивиди са създадени двете граматики: `SINUS TSO basic` и `SINUS TSO I-2`, а за разпознаване на класовете на онтологията е създадена граматиката `SINUS TSO Cl`. Листинг на трите граматики е предоставен в Приложение 4: Анотационни граматики.

Създаване на каскадно задание за претърсване на текст

Върху входния XML-документ съдържащ целеви текстове се извършват следните обработки формирани на етапа на обучението на системата:

- токанизация на текстовете
- вмъкване на XML-елемент за изходния списък с маркери
- парсиране с граматиката SINUS TSO basic
- парсиране с граматиката SINUS TSO I-2
- парсиране с граматиката SINUS TSO CI

Описаните обработки образуват едно каскадно задание в системата CLaRK, чиито изход е XML-документ, с добавени към него списъци с маркери, получени чрез разпознаването на заложените в граматиките фрази (регулярни изрази). Обученото състояние на системата CLaRK се запомня, за да бъде извиквано от процесите на средата СИНУС.

Резултати от работата върху текстовия корпус

Програмата за разпознаване на термините с така настроените граматиките е приложена върху всеки текст от корпуса. Резултатите, изложени в Приложение 5: Резултати от семантичното аотиране, показват за всеки онтологичен термин до каква степен е било ефективно “обучението” на граматиките. От обобщението е видно, че програмата разпознава правилно общо 757 срещания на термини и 12 пъти погрешно предлага разпознаване на термин. Неразпознати са 142 срещания на термини в текстовете от корпуса. С тези данни са пресметнати стойностите на Точност (Precision) и Покритие (Recall):

Точност = $757 / 769 = 0.984$

Покритие = $757 / 899 = 0.842$

За поставените прагматични цели определени в проекта Синус тези резултати са много добри.

Приложение 5: Резултати от семантичното аотиране" съдържа таблица с информация за неразпознатите срещания на онтологичните термини в корпуса. Те са групирани според уникалния идентификатор на онтологичния елемент (URI), който би трябвало да бъде разпознат и са подредени в низходящ ред според броя неразпознати срещания.

Анализът на грешките допуснати от анотационните граматиките дава възможност за евентуално още по-фино настройване на граматиките и подобряване на работата им. Понякога причина за допуснатите погрешни разпознавания и неразпознати срещания на онтологични термини са несъвършенства на онтологичния модел или липса на формализирани експертни знания. В част от случаите пропуснатите от анотационните граматиките срещания на лексикализации на онтологичните термини могат да бъдат обхванати с допълване на терминологичния лексикон.

Проведеното научно-приложно изследване показва, че макар да са възможни по-нататъшни подобрения, следването на релацията „От онтология към текст” води до ефективно решаване на поставената задача за семантично аотиране на български текстове.

Реализираната процедура разпознава най-често срещаните в корпуса онтологични термини и създава адекватни семантични анотации. Като основна насока за бъдеща работа по развитие на избрания метод можем да отбележим, че съществува широко поле за експерименти с различни стратегии за лематизиране на регулярните изрази на частичните анотационни граматиките. Съществен принос би дало пре-използването на създадените с метода ресурси във вид на терминологични лексикони, особено ако могат да са полезни и в други приложни области.

В раздел 4.1 е изложена работата по формализиране на експертни знания от областта на иконографията. Използвани са семантични технологии за представяне на знания в онтологичен

модел. Стойностното в докладваното изследване в раздел 4.2 е прагматичното използване на представения метод за семантично аотиране на специализирани български текстове. Семантичните аотации са определени от лексикализациите на понятията в построената специализирана онтология. Анализиранияте текстове са специализирани описания от областта на знание на онтологията. Подробно са описани основните характеристики на подхода „От онтология към текст“, както и етапите на реализация на семантичното аотиране. Представени са резултати от извършените експерименти с аотационни граматики, а именно, разпознаване на онтологични елементи и семантично аотиране на български текст. Обсъдени са основните параметри на получените резултати, според които прилагането на частични аотационни граматики е изключително подходящо при приложни задачи от описания тип. Идеята следвана от автора е, че семантичното аотиране на български текстове би могло да подпомага значително подготвителните работи по автоматично генериране на български език.

Заклучение

Настоящият дисертационен труд съдържа описание на научно-приложната и изследователска работа по подготовката и използването на ресурси за компютърно генериране на български текстове. Теоретичната парадигма за компютърното генериране е Системично- функционалната лингвистика на Халидей. Избраният приложен подход към генерирането е така наречения контрол направляван от граматиката. Както е показано в обзорната част, необходимите ресурси в конкретния формализъм за компютърно генериране на даден естествен език са лексико-граматически и семантични.

Показано е изграждането на приложна лексико-граматика за компютърно генериране на български език в последователните стъпки на процеса:

- 1) Частичен анализ и моделиране на някои езикови явления в българския език. Създадени са модели на езиковите явления, които са идентифицирани в корпус от целеви текстове за генерирането. Текстовете са процедури-инструкции от ръководство за ползване на CAD/CAM системи. Системично-функционалната граматика на Халидей е парадигмата на лингвистичното моделиране. В дисертацията са показани съставените модели за езиковите явления диатезност, модалност, темпоралност, завършеност на процеса, номинализация на глаголната група, членуване. Моделирани са и някои типични за корпуса връзки в сложните изречения.
- 2) На основата на лингвистичните модели и в паралел със съществуващ приложен ресурс за английски език е съставена приложна българска системично-функционална граматика. Приложният ресурс е изграден със средствата на Средата за многоезиково генериране KPML. Съществени характеристики на приложния ресурс са достъпност, отворената възможност за разширяване и надграждане, както и за пре-използване в други предметни области.

В дисертацията е описан процесът на генериране на български текстове със създадения лексико-граматически ресурс. Докладвана е работата с нужните семантични ресурси за генерирането: проверки за настройване на лингвистичната онтология от високо ниво Обобщен модел (Generalized Upper Model) и на модела на предметната област за работа с български език.

Показан е резултат от компютърно генериране на български език: 5 варианта на текстове-инструкции генерирани от едно и също представяне на информацията на входа.

Последната част на дисертацията показва работата по формализиране на знания в семантични модели свързани със семантично аотиране на български текстове. С идея, че семантичното аотиране би подпомогнало значително подготовката за компютърно генериране в нови предметни области е описан процес на семантично аотиране на описателни текстове на български език от областта на иконографията. Приложен е подходът „Релация: От онтология към текст“. Използвани са частични граматика на базата на регулярни изрази разработени в системата CLaRK.

Дисертацията инспирира идеи за бъдеща работа. Възможностите за експерименти и изследвания на компютърно генериране на български текстове с приложната системично-функционална граматика са огромни. Интересни са и перспективи като многоезиковото генериране или компютърно генериране от различни входни представяния на информацията.

Списък на публикациите по дисертацията

1. Kamenka Staykova: *Natural Language Generation and Semantic Technologies*, Cybernetics and Information Technologies, Volume 14, No2, 2014, pp. 3-24.
2. Kamenka Staykova, Gennady Agre: *Use of Ontology-to-Text Relation for Creating Semantic Annotation*, In Proceedings of 13th International Conference on Computer Systems and Technologies - CompSysTech 2012, Ruse, Bulgaria, June 22 - 23, 2012, pp. 64-71.
3. Kamenka Staykova, Petya Osenova, Kiril Simov: *New Applications of "Ontology-to-Text Relation" Strategy for Bulgarian Language*, Cybernetics and Information Technologies, Bulgarian Academy of Sciences, Sofia, Vol.12, No 4, 2012, pp. 43-52.
4. Kamenka Staykova, Gennady Agre, Kiril Simov, Petya Osenova: *Language Technology Support for Semantic Annotation of Iconographic Descriptions*, In Proceedings of the International Workshop "Language Technologies for Digital Humanities and Cultural Heritage", Sept. 2011, Hisar, Bulgaria, 16 Sept. 2011, pp. 51-57.
5. Kamenka Staykova: *Exercise in Conceptualization*, Cybernetics and Information Technologies, Bulgarian Academy of Sciences, Sofia, Vol. 5, No 2, 2005, pp. 69-83.
6. Kamenka Staykova and Sergey Varbanov: *The Globe: Representation of Linguistic Knowledge and Knowledge about the World Together*, In Proceedings of the Workshop "Language and Speech Infrastructure for Information Access in the Balkan Countries", part of Fifth International Conference on Recent Advances in Natural Language Processing, RANLP-2005, Borovetz, Bulgaria, 25 September 2005, pp. 68-74.
7. Danail Dochev, Kamenka Staykova: *A Multilingual System for Automatic Generation of Technical Manual Texts*, In Proceedings of the International Conference on Computer Systems and Technologies CompSysTech'2001, Sofia, 21-22 June 2001, pp. II.14.1-5.
8. Kamenka Staykova, Danail Dochev: *Development of Lexico-Grammar Resources for Natural Language Generation (Experience from AGILE Project)*, In: Cerry S. and D. Dochev (Eds.), Proceedings of the International Conference "Artificial Intelligence: Methodology, Systems, Applications 2000", Varna, September 2000, Lecturer Notes in Artificial Intelligence 1904, Springer-Verlag, 2000, pp. 242-251.
9. Kruijff GJ, E. Teich, J. Bateman, I. Kruijff-Korbayová, H. Skoumalová, S. Sharoff, L. Sokolova, T. Hartley, K. Staykova, J. Hana: *Multilinguality in a Text Generation System for Three Slavic Languages*, In Proceedings of the 18th Conference on Computational Linguistics - Volume 1, Saarbrücken, Germany, 2000, pp. 474 – 480..
10. Staykova K.: *Bulgarian Resource for Generation of Instructional Texts: Result of AGILE Project*, IIT Working Papers, IIT/WP-109, 2000.
11. Стайкова К., Й. Пенчев: *Системично-функционалната лингвистика и българският език*, "Български език", София, XLVIII, том 4-5, 1999/2000, стр. 5-24.
12. Dochev D., N. Gromova, K. Staykova - *Lexico-Grammatical Characteristics of Bulgarian Software Instructional Texts*. Problems of Engineering Cybernetics and Robotics, No 49, pp. 11-19, 1999.

Публикация 1. кореспондира с обзорната част на дисертацията, Глава 1, където се разглежда отношението на компютърното генериране на текст и съвременните семантични технологии. Публикации 10. и 11. са свързани с моделирането на някои езикови явления в българския език (раздел 2.1), докато публикации 7. и 9. отразяват работата по приложната граматика (раздел 2.2). Към изследванията описани в Глава 3 имат отношение публикации 5., 6. и 8. Това са доклади за реализирания процес на компютърно генериране на български текстове. Публикации 2., 3. и 4. се отнасят към Глава 4 и отразяват работата по семантично аотиране на специализирани български текстове.

Апробация на резултатите

Доклади по темата на дисертацията са изнесени на следните научни форуми и семинари:

- Международна конференция по компютърни системи и технологии CompSysTech'2012, Русе юни, 2012.
- Международен семинар "Language Technologies for Digital Humanities and Cultural Heritage"/ „Езикови технологии за електронната хуманитаристика и културното наследство“ Хисаря, септември, 2011.
- Международен семинар "Language and Speech Infrastructure for Information Access in the Balkan Countries", Езикова инфраструктура за достъп до информацията в Балканските страни, Боровец, септември 2005г.
- постер-сесия на международната конференция RANLP 2001, Цигов чарк, 2001.
- Международна конференция по компютърни системи и технологии CompSysTech'2001, София, юни 2001г.
- 18та международна конференция по компютърна лингвистика, Саарбрюкен, Германия, 2000.
- Девета международна конференция AIMSА: Изкуствен интелект- методология, системи, приложения, Варна, септември 2000г.
- Текущи семинари на секция "Изкуствен интелект" в Института по информационни технологии на БАН, 1999-2004.

Основни научни и научно-приложни приноси

В дисертацията са докладвани резултати от осъществяване на процес на компютърно генериране на български текстове. Реализирани са следните научни и научно-приложни приноси:

1. Направено е формално описание на базови обекти от българския език в съответствие с Системично-функционалната теория за естествения език. Разработеното описание позволява компютърно генериране на процедурни текстовете (технически инструкции) за създаване на техническа документация на български език.

2. Разработеното формално описание е реализирано във вид на приложен компютърен ресурс (Приложна системично-функционална граматика на български език). Реализацията е извършена в Средата за многоезиково генериране KPML и позволява пре-използване, разширяване и по-нататъшно развитие.

3. Разработената Приложна системично-функционална граматика на български език е реализирана като модул в многоезиковата система AGILE позволяваща компютърно генериране на технически текстове в различни стилове (персонален- императивен, безличен- изявителен). Системата е приложена за генериране на кохерентни текстове на български език при създаване на технически ръководства за CAD/CAM системи.

4. Въз основа на разработено онтологично описание (специализация на CIDOC- CRM) в областта на Източно-християнско иконографско изкуство е предложена и реализирана схема за семантично аотиране на специализирани текстове на български език. Реализацията използва частични граматики на базата на регулярни изрази.

Декларация за оригиналност

Декларирам, че настоящата дисертация съдържа оригинални резултати, получени при проведени от мен научни изследвания с подкрепата и съдействието на научните колективи на проектите AGILE и SINUS . Резултатите, които са получени, описани и публикувани от други учени, са надлежно и подробно цитирани в библиографията.

Настоящата дисертация не е прилагана за придобиване на научна степен в друго висше училище, университет или научен институт.

Благодарности

Благодаря на научния си ръководител Данаил Дочев за търпеливото, задълбочено и многократно редактиране. Благодарна съм на Йордан Пенчев, чиято роля при системично-функционалното моделиране за проекта AGILE оцених истински твърде късно. Искрени благодарности на колегите Петя Осенова и Кирил Симов за споделената работа по проекта СИНУС и за споделените идеи в последната фаза на писане на дисертацията. Благодаря на Геннадий Агре за подканващата му подкрепа за завършване на дисертацията. Благодаря на семейството си за търпението и на всички колеги и приятели за насърчаването в доста продължителната работа по тази дисертация.

Литература

- [Бояджиев, Куцаров, Пенчев, 1999] Бояджиев Т., И. Куцаров, Й. Пенчев: *Съвременен български език*, ИК Петър Берон, София, 1999.
- [Попов, Симов, Видинска, 1998] Попов Д., К. Симов и Св. Видинска: *Речник за правоговор, правопис и пунктуация*, Атлантис, 1998, София.
- [Стайкова и Пенчев, 2000] Стайкова К., Й. Пенчев: *Системично-функционалната лингвистика и българският език*, "Български език", София, XLVIII, том 4-5, 1999/2000, стр. 5-24.
- [Agre, 2012] Agre, G.: *SINUS – A Semantic Technology Enhanced Environment for Learning in Humanities*, Cybernetics and Information Technologies, Vol. 12, 2012, No 4, 5-24.
- [Al-Muhtaseb and Mellish, 1997] Al-Muhtaseb, Husni and Chris Mellish: From the Generalized Upper Model Towards an Arabic Upper Model, In Proceedings of The 4th IEEE International Conference on Electronics, Circuits and Systems ICECS'97, Cairo, Egypt, 1997.
- [Basile and Bos, 2011] Basile V. and J. Bos: Towards *Generating Text from Discourse Representation Structures*, In Proceedings of the 13th European Workshop on Natural Language Generation (Nancy, France, 2011), pp. 145–150.
- [Bateman, 1990] Bateman J.: *Upper Modelling: Organizing Knowledge for Natural Language Processing*, In Proceedings of the 5th International Workshop on Natural Language Generation, 3-6 June 1990, pp. 54–60.
- [Bateman, 1997] Bateman J. A.: *Enabling Technology for Multilingual Natural Language Generation: The KPML Development Environment*, Natural Language Engineering 3, 1 (1997), 15–55.
- [Bateman, 2002] Bateman J. A.: *Natural Language Generation: an Introduction and Open-Ended Review of the State of the Art*, 2002, <http://www.fb10.uni-bremen.de/anglistik/langpro/webpace/jb/info-pages/nlg/ATG01/ATG01.html> .
- [Bateman and Teich, 1995] Bateman, J. A. and Teich, E.: *Selective Information Presentation in an Integrated Publication System: an Application of Genre-Driven Text Generation*, Information Processing and Management, 1995, 31(5), 753-767.
- [Bateman and Paris, 1989] Bateman, J.A. and C.L. Paris: *Phrasing a Text in Terms the User Can Understand*, In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, IJCAI'89, Detroit, Michigan, 1989, pp. 1511-1517.
- [Bateman et.al, 1990] Bateman J., R. Kasper, J. Moore, R. Whitney: *A General Organization of Knowledge for Natural Language Processing: the Penman Upper Model*, Technical report, USC/Information Sciences Institute, Marina del Rey, California, 1990.
- [Bateman et.al., 1994] Bateman, J., B. Magnini and F. Rinaldi: *The Generalized {Italian, German, English} Upper Model*, The ECAI94 Workshop: Comparison of Implemented Ontologies, Amsterdam, 1994.
- [Bateman et.al., 1995] Bateman J. A., B. Magnini and G. Fabris: *The Generalized Upper Model Knowledge Base: Organization and Use*, In Towards Very Large Knowledge Bases, N. Mars, Ed. IOS Press, Amsterdam, 1995, pp. 60–72.
- [Bateman et.al., 2000] Bateman, J., Teich, E., Kruijff-Korbayová, I., Kruijff, G.-J., Sharoff, S. and Skoumalová, H.: *Resources for multilingual text generation in three Slavic languages*, in Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'2000), European Language Resources Association (ELRA), 2000, Athens, Greece, pp.1763-1768.
- [Bateman et.al., 2010] Bateman, J. A., J. Hois, R. Ross, and T. Tenbrink: *A Linguistic Ontology of Space for Natural Language Processing*, Artificial Intelligence 174, 14 (2010), 1027 – 1071.
- [Berners-Lee et.al., 2001] Berners-Lee T., J. Hendler and O. Lassila: *The Semantic Web*, Scientific American, 2001, pp. 29–37.

- [Bontcheva and Angelova, 1996] Bontcheva, K. and Angelova, G.: *Planning and Generating Hypertext Documentation*. In: Proceedings. of the Workshop "Gaps and Bridges in Natural Language Generation", European Conference on Artificial Intelligence ECAI-96, Budapest, Hungary, August 1996, pp. 25-28.
- [Bontcheva and Cunningham, 2003] Bontcheva, D. K. and H. Cunningham: *The Semantic Web: A New Opportunity and Challenge for Human Language Technology*, In Proceedings of the Workshop on Human Language Technology for the Semantic Web and Web Services, ISWC'03 ,Florida, USA, October 2003, pp. 89-96.
- [Bontcheva, 2005] Bontcheva K.: *Generating Tailored Textual Summaries from Ontologies*. In The Semantic Web: Research and Applications, Proceedings of the Second European Semantic Web Conference, ESWC, Heraklion, Crete, Greece, 2005, pp. 531-545.
- [Bontcheva and Wilks, 2004] Bontcheva K. and Wilks Y.: *Automatic Report Generation from Ontologies: the MIAKT Approach*. In Natural Language Processing and Information Systems, Proceedings of the Ninth International Conference on Applications of Natural Language to Information Systems (NLDB), 2004, vol. 3136 of Lecture Notes in Computer Science, Springer, pp. 324-335.
- [Bouayad-Agha et.al., 2012] Bouayad-Agha N., Casamayor G., Mellish C., and Wanner L.: Content Selection from Semantic Web Data. In Proceedings of the Seventh International Natural Language Generation Conference (INLG), Special Track on Future Generation Challenges Proposals (2012), pp. 146-149.
- [Cimiano and Kopp, 2010] Cimiano P. and S. Kopp: *Accessing the Web of Data through Embodied Virtual Characters*, Semantic Web 1, 1-2, 2010, 83-88.
- [Crofts et.al., 2011] Crofts, N., Doerr M., Gill T., Stead S., Stiff M.: *Definition of the CIDOC Conceptual Reference Model*, 2011, available: http://www.cidoccrm.org/docs/cidoc_crm_version_5.0.4.pdf
- [Dai et.al., 2010] Dai Y., S. Zhang, J. Chen, T. Chen, W. Zhang: *Semantic Network Language Generation Based on a Semantic Networks Serialization Grammar*, World Wide Web 13, 3, 307-341.
- [Dannells et.al., 2012] Dannells D., M. Damova, R. Enache, and M. Chechev: *Multilingual Online Generation from Semantic Web Ontologies*. In Proceedings of the 21st International Conference Companion on World Wide Web, New York, USA, 2012, WWW '12 Companion, ACM, pp. 239-242.
- [Davis et.al., 2008] Davis B., A. Iqbal, A. Funk, V. Tablan, K. Bontcheva, H. Cunningham, S. Handschuh: *RoundTrip Ontology Authoring*. In Proceedings of the International Semantic Web Conference (ISWC), vol. 5318 of Lecture Notes in Computer Science. Springer Berlin /Heidelberg, 2008, pp. 50-65.
- [Davis et.al., 1993] Davis R., Shrobe H., and Szolovits P.: *What Is a Knowledge Representation?*, AI Magazine Volume 14 Number 1, 1993, <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1029/947>
- [Demir et.al, 2010] Demir S., S. Carberry, and K. McCoy: *A Discourseaware Graph-based Content Selection Framework*, In Proceedings of the 6th International Natural Language Generation Conference (INLG) (2010), pp. 17-27.
- [Dochev and Agre, 2009] Dochev D., Agre G.: *Towards Semantic Web Enhanced Learning*, In: Proceedings of the International Conference on Knowledge Management and Information Sharing, Madeira, 2009, pp. 212-217.
- [Dochev and Agre, 2012] Dochev D. and G. Agre: *Supporting Learning-by-Doing Situations by Semantic Technologies*, In: Stefanoui D., J Culita (Eds.). Proceedings of 17th Annual Conference on Media and Web Technology EUROMEDIA'2012, Bucharest, April, 2012, ISBN 978-90-77381-69-4, pp. 49-53.
- [Duboue and McKeown, 2003] P. A. Duboue and K. R. McKeown: *Statistical acquisition of content selection rules for natural language generation*. In Proceedings of 2003 Conference on Empirical Methods for Natural Language Processing, (EMNLP 2003), Sapporo, Japan, July.
- [Elhadad, 1990] Elhadad, M.: *Types in Functional Unification Grammars*, in Proceedings of the 28th. Annual Meeting of the Association for Computational Linguistics, ACL, 1990, pp. 157-164.

- [Elhadad and Robin, 1992] Elhadad, M. and Robin, J.: *Controlling Content Realization with Functional Unification Grammars*, In R. Dale, E. H. Hovy, D. Rösner and O. Stock, editors, “Aspects of automated natural language generation”, 6th. international workshop on natural language generation, Springer, Berlin/Heidelberg, 1992, pp. 89-104.
- [Erdmann et.al, 2000] Erdman M., A. Maedche, H.-P. Schnurr, S. Staab: *From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools*, In P. Buitelaar and K. Hasida (eds) Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content, 2000.
- [Galanis and Androutsopoulos, 2007] Galanis, D. and I. Androutsopoulos: *Generating Multilingual Descriptions from Linguistically Annotated OWL Ontologies: the NaturalOWL System*, in Proceedings of the 11th European Workshop on Natural Language Generation, ENLG '07, pp. 143–146, 2007.
- [Gardent et.al., 2011] Gardent C., E. Banik, and L. Perez-Beltrachini: *Natural Language Generation and Natural Language Interfaces to Knowledge Bases*, Tutorial at the Sixth International Conference on Knowledge Capture (K-CAP), 2011.
- [Gruber, 1993] Gruber, T.: *A translation approach to portable ontology specifications*, Knowledge Acquisition 5(2), pp. 199–220, 1993.
- [Halliday, 1994] M A K Halliday: *Introduction to Functional Grammar*, Edward Arnold, London, Second Edition, 1994.
- [Henschel, 1993] Renata Henschel: *Merging the English and the German Upper Model*, Darmstadt, Germany, GMD/Institute für Integrierte Publikation- und Informationssysteme, 1993.
- [Hewlett et.al., 2005] Hewlett D., A. Kalyanpur, V. Kolovski, C. Halaschek-Wiener: *Effective NL Paraphrasing of Ontologies on the Semantic Web*. In Workshop on End-User Semantic Web Interaction, 4th International Semantic Web Conference (ESWC), Galway, Ireland, 2005.
- [Hielkema et.al., 2007] Hielkema F., C. Mellish, P. Edwards: *Using WYSIWYM to Create an Open-ended Interface for the Semantic Grid*, In Proceedings of the Eleventh European Workshop on Natural Language Generation, ENLG '07, Stroudsburg, PA, USA, Association for Computational Linguistics, 2007, pp. 69–72.
- [Hovy, 1988] Hovy, E.: *Generating Natural Language under Pragmatic Constraints*. Lawrence Erlbaum, Hillsdale, New Jersey, 1988.
- [Hovy and Nirenburg, 1992] Eduard Hovy and Sergei Nirenburg: *Approximating an Interlingua in a Principled Way*, the DARPA Speech and Natural Language Workshop, Arden House, New York, 1992.
- [Joshi, 1987] Joshi, A. K.: *The Relevance of Tree Adjoining Grammar to Generation*, In G. Kempen, ed., “Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics”, Kluwer Academic Publishers, Boston/Dordrecht, 1987.
- [Kruijff et.al., 2000] Kruijff, G.-J., Teich, E., Bateman, J., Kruijff-Korbayová, I., Skoumalová, H., Sharoff, S., Sokolova, L., Hartley, T., Staykova, K. and Hana, J., *A multilingual system for text generation in three Slavic languages*, in Proceedings of the 18th. International Conference on Computational Linguistics (COLING'2000)', Saarbrücken, Germany, 2000, pp. 474-480.
- [Lavoie and Rambow, 1997] Lavoie, B. and Rambow, O.: *A fast and portable realizer for text generation systems*, in Proceedings of the 5th. Conference on Applied Natural Language Processing, ACL, Washington, 1997, pp. 265-268.
- [Mann, 1983] Mann, W.C., *An overview of the PENMAN text generation system*, in Proceedings of the National Conference on Artificial Intelligence, AAAI, 1983, pp.261-265.
- [Mann and Matthiessen, 1985] Mann, W.C. and Matthiessen, C.M.I.M.: *Demonstration of the Nigel text generation computer program*, in J.D.Benson and W.S.Greaves, eds, “Systemic Perspectives on Discourse”, Volume 1, Ablex, Norwood, New Jersey, 1985, pp.50-83.
- [Mann and Thompson, 1988] Mann, W.C. and S.A. Thompson: *Rhetorical structure theory: Toward a functional theory of text organization*, Text 8(3), 243-281.

- [Matthiessen and Bateman, 1991] Matthiessen C. and J. Bateman: *Text Generation and Systemic-Functional Linguistics: Experiences from English and Japanese*, Frances Pinter Publishers and St. Martin's Press, London and New York, 1991.
- [McDonald, 1983] McDonald, D.D. *Description directed control: its implications for natural language generation*, Computers and Mathematics, 9(1), 1983, 111-129.
- [McKeown, 1985] McKeown, K: *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*, Cambridge University Press, Cambridge, England, 1985.
- [Mellish and Pan, 2006] Mellish C., and J. Pan: *Finding Subsumers for Natural Language Presentation*. In Proceedings of the International Workshop on Description Logics, The Lake District, England, 2006, pp. 127–134.
- [Mellish and Pan, 2008] Mellish C., and J. Pan: *Natural Language Directed Inference from Ontologies*. Artificial Intelligence 172, 10, June 2008, 1285–1315.
- [Mellish and Sun, 2006] Mellish C., and X. Sun: *The Semantic Web as a Linguistic Resource: Opportunities for Natural Language Generation*. Knowledge-Based Systems 19, 5, Sept. 2006, 298–303.
- [Meteer, 1992] Meteer, M. W.: *Expressibility and the Problem of Efficient Text Planning*, Pinter Publishers, London, 1992.
- [Mitkov, 1990] Mitkov, R.: *Generating Explanations of Geometrical Concepts*, Computers and Artificial Intelligence, 1990, 9(6), 579-589.
- [Moore and Paris, 1988] Moore, J. D. and Paris, C. L.: *Constructing Coherent Texts Using Rhetorical Relations*, In Proceedings of the Tenth Annual Conference of the Cognitive Science Society', Cognitive Science Society, 1988.
- [Nogier and Zock, 1992] Nogier J.-F. and M. Zock, *Lexical Choice as Pattern Matching*, Knowledge Based Systems 5, 3 (1992), 200–212.
- [Paneva-Marinova et.al., 2010] Paneva-Marinova D., R. Pavlov, M. Goynov, L. Pavlova-Draganova, L. Draganov: *Search and Administrative Services in Iconographical Digital Library*, In Proceedings of the International Conference “Information Research and Applications”, July 2010, Varna, Bulgaria, 177-187.
- [Pavlova-Draganova, et.al., 2007] Pavlova-Draganova L., V. Georgiev, L. Draganov: *Virtual Encyclopaedia of Bulgarian Iconography*, Information Technologies and Knowledge, Vol. 1, 2007, No 3, 267-271.
- [Pollard and Sag, 1994] Carl J. Pollard and Ivan A. Sag: *Head-Driven Phrase Structure Grammar*, University of Chicago Press, Chicago, Illinois, USA, 1994.
- [Polikoff and Allemang, 2003] I. Polikoff and D. Allemang: *Semantic Technology*, TopQuadrant Technology Briefing, v1.1, September 2003, <https://lists.oasis-open.org/archives/regrep-semantic/200402/pdf00000.pdf> .
- [Power, 2009] Power, R.: *Towards a Generation-based Semantic Web Authoring Tool*, in Proceedings of the 12th European Workshop on Natural Language Generation ENLG '09, Stroudsburg, USA, ACL, 2009, pp. 9–15.
- [Power, 2010] Power R.: *Complexity Assumptions in Ontology Verbalisation*, In Proceedings of 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, Stroudsburg, PA, USA, 2010, ACLShort '10, pp. 132–136.
- [Power and Third, 2010] Power R. and A. Third: *Expressing OWL Axioms by English Sentences: Dubious in Theory, Feasible in Practice*, In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Stroudsburg, PA, USA, 2010, COLING'10, Association for Computational Linguistics, pp. 1006–1013.
- [Ranta, Angelov, Hallgren, 2010] Ranta, A., K. Angelov, T. Hallgren: *Tools for Multilingual Grammar-Based Translation on the Web*, In Proceedings of the ACL 2010 System Demonstrations, pp. 66-71, 2010.
- [Rao et.al., 2011] Rao D., P. McNamee, M. Dredze: *Entity Linking: Finding Extracted Entities in a Knowledge Base*, Multi-source, Multi-lingual Information Extraction and Summarization, 2011.

- [Reiter, 1995] Reiter, E., *NLG vs. Templates*, in Proceedings of the Fifth European Workshop on Natural Language Generation, Faculty of Social and Behavioural Sciences, University of Leiden, Leiden, The Netherlands, 1995, pp. 95-105.
- [Reiter and Dale, 2000] Reiter E. and R. Dale: *Building Natural Language Generation Systems (Studies in Natural Language Processing)*, Cambridge University Press, 2000.
- [Shieber et.al., 1990] Shieber, S.M., van Noord, G., Pereira, F. C. N. and Moore, R.C., *Semantic head-driven generation*, Computational Linguistics 16(1), 1990, 30-42.
- [Simov and Osenova, 2007] Simov K., P. Osenova: *Applying Ontology-Based Lexicons to the Semantic Annotation of Learning Objects*. In: Proceedings of the Workshop on NLP and Knowledge Representation for eLearning Environments, RANLP-2007, 49-55.
- [Simov and Osenova, 2008] Simov K., P. Osenova: *Language Resources and Tools for Ontology-Based Semantic Annotation*, In: Al. Oltramari, L. Prévot, Chu-Ren Huang, P. Buitelaar, P. Vossen, Eds. Proc. of the OntoLex Workshop at LREC'2008, 2008, 9-13.
- [Simov et.al., 2001] Simov K., Z. Peev, M. Kouylekov, A. Simov, M. Dimitrov, A. Kiryakov: *CLaRK - an XML-based System for Corpora Development*, In: Proceedings of the Corpus Linguistics Conference, 2001, pp. 558-560.
- [Simov et.al., 2002] Kiril Simov, Milen Kouylekov, Alexander Simov, *Cascaded Regular Grammars over XML Documents*, In: Proceedings of the 2nd Workshop on NLP and XML (NLPXML-2002), Taipei, Taiwan. September, 2002. pages 51-58.
- [Staykova, 2014] K. Staykova: *Natural Language Generation and Semantic Technologies*, Cybernetics and Information Technologies, Volume 14, No2, 2014, pp. 3-24.
- [Staykova and Agre, 2012] K. Staykova, G. Agre: *Use of Ontology-to-Text Relation for Creating Semantic Annotation*, In Proceedings of 13th International Conference on Computer Systems and Technologies - CompSysTech 2012, Ruse, Bulgaria, June 22 - 23, 2012, pp. 64-71.
- [Staykova et.al., 2012] K. Staykova, P. Osenova, K. Simov: *New Applications of "Ontology-to-Text Relation" Strategy for Bulgarian Language*, Cybernetics and Information Technologies, ISSN 1311-9702, Bulgarian Academy of Sciences, Sofia, Vol.4, 2012, pp. 43-52.
- [Stevens et.al., 2011] Stevens R., J. Malone, S. Williams, R. Power, A. Third: *Automating Generation of Textual Class Definitions from OWL to English*. Journal of Biomedical Semantics 2, 2:S5, 2011.
- [Teich, 1999] Elke Teich: *Systemic functional grammar in Natural Language Generation: linguistic description and computational representation*, Cassell, London, 1999.
- [Turner et.al., 2006] Turner R., S. Sripada, E. Reiter, I. Davy: *Generating Spatio-Temporal Descriptions in Pollen Forecasts*. In Proceedings of the 11th Conference of European Chapter of the Association for Computational Linguistics, Trento, Italy, EACL06, 2006, pp. 163-166.
- [Wimalasuriya and Dou, 2010] Wimalasuriya D. C. and D. Dou: *Ontology- based Information Extraction: An Introduction and a Survey of Current Approaches*, Journal of Information Science, 36 (3), 2010, pp. 306-323.

Приложение 1: Модел на предметна област „CAD/CAM приложения“

Моделът на предметната област на CAD/CAM приложенията е съставен единствено за целите на проекта AGILE и няма претенции за изчерпателност по отношение на CAD/CAM областта.

DM::ADD ДОБАВЯМ
DM::APPEAR ПОЯВЯВАМ_СЕ
DM::APPLY-PROPERTY ПРИЛАГАМ
DM::CONSIST-OF СЪСТОЯ-СЕ-ОТ
DM::CHANGE-COMPONENT ПРОМЕНЯМ
DM::CHOOSE ИЗБИРАМ
DM::CLOSE-SCREEN-OBJECT ЗАТВАРЯМ
DM::CLICK ПОСОЧВАМ
DM::CREATE SAZDAVAM
DM::DEFINE DEFINIRAM
DM::DISAPPEAR ИЗЧЕЗВАМ
DM::DISPLAY ПОКАЗВАМ
DM::DRAW ЧЕРТАЯ
DM::EDIT РЕДАКТИРАМ
DM::END-LINE ЗАВЪРШВАМ
DM::ENTER ВЪВЕЖДАМ
DM::JUSTIFY ПОДРАВНЯВАМ
DM::NOT-SEE СКРИВАМ
DM::OPEN ОТВАРЯМ
DM::PRESS НАТИСКАМ
DM::PRINT ПЕЧАТАМ
DM::REPEAT-STEPS ПОВТАРЯМ
DM::RETURN-TO ВЪЗСТАНОВЯВАМ
DM::SAVE ЗАПАЗВАМ
DM::SEE ВИЖДАМ
DM::SNAP ПРИКАЧВАМ
DM::SPECIFY-COMPONENT ЗАДАВАМ
DM::START-LINE ЗАПОЧВАМ
DM::START-TOOL СТАРТИРАМ
DM::SWITCH-MODE ПРЕВКЛЮЧВАМ
DM::USE ИЗПОЛЗВАМ
DM::QUIT-TOOL ИЗКЛЮЧВАМ

DM::ANGLE ЪГЪЛ
DM::ARC ДЪГА
DM::BUTTON КЛАВИШ
DM::CADCAM-LABEL ЕТИКЕТ
DM::COMMAND-LINE КОМАНДА
DM::CONSTRAINT ОГРАНИЧЕНИЕ
DM::DATA-NAME ИМЕ
DM::DESCRIPTION ОПИСАНИЕ
DM::DIALOG-BOX ДИАЛОГОВ-ПРОЗОРЕЦ

DM::DISTANCE РАЗСТОЯНИЕ
DM::DOCUMENT ДОКУМЕНТ
DM::DRAWING ЧЕТРЕЖ
DM::END-POINT КРАЙНА-ТОЧКА
DM::ELEMENT ЕЛЕМЕНТ
DM::FIELD ПОЛЕ
DM::FILE ФАЙЛ
DM::HATCH ЦРИХОВКА
DM::JUSTIFICATION ПОДРАВНЯВАНЕ
DM::ICON ИКОНА
DM::KEY КЛАВИШ
DM::LINE ЛИНИЯ
DM::LINE-SEGMENT ОТСЕЧКА
DM::MENU МЕНЮ
DM::MULTILINE МУЛТИЛИНИЯ
DM::MODE РЕЖИМ
DM::OBJECT ОБЕКТ
DM::OPTION ОПЦИЯ
DM::POINT ТОЧКА
DM::POLYLINE ПОЛИЛИНИЯ
DM::PROGRAM ПРОГРАМА
DM::ROOT КОРЕН
DM::SCALE МАЩАБ
DM::SCREEN-LIST СПИСЪК
DM::SOFTWARE-COMMAND КОМАНДА
DM::SPECIFICATION СПЕЦИФИКАЦИЯ
DM::START-POINT НАЧАЛНА-ТОЧКА
DM::STYLE СТИЛ
DM::TANGENT ДОПИРАТЕЛНА

Приложение 2: Формално представяне на съдържанието на процедурен текст

Процедура „Чертане на дъга“

PROCEDURE a1

GOAL:

DRAW a2

ACTEE:

GRAPHICAL-ACTEE a3

RANGE:

ARC a4

ACTOR:

USER-ACTOR a5

RANGE:

USER a6

METHODS:

METHOD-LIST a7

FIRST:

METHOD* a8

PRECONDITION:

PROCEDURE a9

GOAL:

START-TOOL a10

ACTEE:

SOFTWARE-ACTEE a11

RANGE:

SOFTWARE-COMMAND a12

LABEL:

GUI-ARC a13

ACTOR:

USER-ACTOR a14

RANGE:

USER a6

METHODS:

METHOD-LIST a15

FIRST:

METHOD* a16

CONSTRAINT:

OPERATING-SYSTEM a17

LABEL:

GUI-WINDOWS a18

SUBSTEPS:

PROCEDURE-LIST a19

FIRST:

PROCEDURE a20

GOAL:

CLICK a21

ACTEE:

DISPLAYED-ACTEE a22

RANGE:

ICON a23

LABEL:

GUI-THREE-POINTS a24

LOCATION:

GUI-ACTEE a25
RANGE:
FLYOUT a26
LOCATION:
TOOLBAR a27
LABEL:
GUI-DRAW a28
LABEL:
GUI-ARC a29
ACTOR:
USER-ACTOR a30
RANGE:
USER a6
REST:
METHOD-LIST a31
FIRST:
METHOD* a32
CONSTRAINT:
OPERATING-SYSTEM a33
LABEL:
GUI-DOS-AND-UNIX a34
SUBSTEPS:
PROCEDURE-LIST a35
FIRST:
PROCEDURE a36
GOAL:
CHOOSE a37
ACTEE:
DISPLAYED-ACTEE a38
RANGE:
OPTION a39
LABEL:
GUI-ARC a40
OPTIONS:
GUI-ACTEE a41
RANGE:
MENU a42
LABEL:
GUI-DRAW a43
ACTOR:
USER-ACTOR a44
RANGE:
USER a6
REST:
PROCEDURE-LIST a45
FIRST:
PROCEDURE a46
GOAL:
CHOOSE a47
ACTEE:
DISPLAYED-ACTEE a48
RANGE:
OPTION a49
LABEL:
GUI-THREE-POINTS a50
OPTIONS:
GUI-ACTEE a51
ACTOR:
USER-ACTOR a52

RANGE:
 USER a6
 SUBSTEPS:
 PROCEDURE-LIST a53
 FIRST:
 PROCEDURE a54
 GOAL:
 SPECIFY-COMPONENT a55
 ACTEE:
 CONFIGURED-ACTEE a56
 RANGE:
 START-POINT a57
 OWNER:
 ARC a4
 LOCATION:
 DISPLAYED-ACTEE a58
 ACTOR:
 USER-ACTOR a59
 RANGE:
 USER a6
 METHODS:
 METHOD-LIST a60
 FIRST:
 METHOD* a61
 SUBSTEPS:
 PROCEDURE-LIST a62
 FIRST:
 PROCEDURE a63
 GOAL:
 ENTER a64
 ACTEE:
 CONFIGURED-ACTEE a65
 RANGE:
 COMMAND-LINE a66
 LABEL:
 GUI-ENDP a67
 LOCATION:
 GUI-ACTEE a68
 RANGE:
 COMMAND-LINE-PROMPT a69
 ACTOR:
 USER-ACTOR a70
 RANGE:
 USER a6
 REST:
 PROCEDURE-LIST a71
 FIRST:
 PROCEDURE a72
 GOAL:
 CHOOSE a73
 ACTEE:
 DISPLAYED-ACTEE a74
 RANGE:
 LINE a75
 OPTIONS:
 GUI-ACTEE a76
 ACTOR:
 USER-ACTOR a77
 RANGE:

USER a6
 SIDE-EFFECT:
 SNAP a78
 LOCATION:
 GRAPHICAL-ACTEE a79
 RANGE:
 END-POINT a80
 OWNER:
 LINE a75
 ACTEE:
 DISPLAYED-ACTEE a81
 RANGE:
 ARC a82
 ACTOR:
 USER-ACTOR a83
 RANGE:
 USER a6
 REST:
 PROCEDURE-LIST a84
 FIRST:
 PROCEDURE a85
 GOAL:
 SPECIFY-COMPONENT a86
 ACTEE:
 CONFIGURED-ACTEE a87
 RANGE:
 POINT a88
 OWNER:
 ARC a4
 NUMBER:
 TWO a89
 LOCATION:
 DISPLAYED-ACTEE a90
 ACTOR:
 USER-ACTOR a91
 RANGE:
 USER a6
 METHODS:
 METHOD-LIST a92
 FIRST:
 METHOD* a93
 SUBSTEPS:
 PROCEDURE-LIST a94
 FIRST:
 PROCEDURE a95
 GOAL:
 ENTER a96
 ACTEE:
 CONFIGURED-ACTEE a97
 RANGE:
 COMMAND-LINE a98
 LABEL:
 GUI-POI a99
 LOCATION:
 GUI-ACTEE a100
 RANGE:
 COMMAND-LINE-PROMPT a101
 ACTOR:
 USER-ACTOR a102

RANGE:
 USER a6
 REST:
 PROCEDURE-LIST a103
 FIRST:
 PROCEDURE a104
 GOAL:
 CHOOSE a105
 ACTEE:
 DISPLAYED-ACTEE a106
 RANGE:
 POINT a107
 OPTIONS:
 GUI-ACTEE a108
 ACTOR:
 USER-ACTOR a109
 RANGE:
 USER a6
 SIDE-EFFECT:
 SNAP a110
 LOCATION:
 GRAPHICAL-ACTEE a111
 RANGE:
 POINT a107
 ACTEE:
 DISPLAYED-ACTEE a112
 RANGE:
 ARC a113
 ACTOR:
 USER-ACTOR a114
 RANGE:
 USER a6
 REST:
 PROCEDURE-LIST a115
 FIRST:
 PROCEDURE a116
 GOAL:
 SPECIFY-COMPONENT a117
 ACTEE:
 CONFIGURED-ACTEE a118
 RANGE:
 END-POINT a119
 OWNER:
 ARC a4
 LOCATION:
 DISPLAYED-ACTEE a120
 ACTOR:
 USER-ACTOR a121
 RANGE:
 USER a6
 REST:
 PROCEDURE-LIST a122
 FIRST:
 PROCEDURE a123
 GOAL:
 END-LINE a124
 ACTEE:
 GRAPHICAL-ACTEE a125
 RANGE:

- POLYLINE a126
- ACTOR:
 - USER-ACTOR a127
 - RANGE:
 - USER a6
- METHODS:
 - METHOD-LIST a128
 - FIRST:
 - METHOD* a129
 - SUBSTEPS:
 - PROCEDURE-LIST a130
 - FIRST:
 - PROCEDURE a131
 - GOAL:
 - PRESS a132
 - ACTEE:
 - HARDWARE-ACTEE a133
 - RANGE:
 - KEY a134
 - LABEL:
 - GUI-RETURN a135
 - ACTOR:
 - USER-ACTOR a136
 - RANGE:
 - USER a6
- REST:
 - PROCEDURE-LIST a137
 - FIRST:
 - PROCEDURE a138
 - GOAL:
 - SAVE a139
 - ACTEE:
 - CONFIGURED-ACTEE a140
 - RANGE:
 - DRAWING a141
 - LOCATION:
 - DATA-ACTEE a142
 - ACTOR:
 - USER-ACTOR a143
 - RANGE:
 - USER a6
 - METHODS:
 - METHOD-LIST a144
 - FIRST:
 - METHOD* a145
 - SUBSTEPS:
 - PROCEDURE-LIST a146
 - FIRST:
 - PROCEDURE a147
 - GOAL:
 - CHOOSE a148
 - ACTEE:
 - DISPLAYED-ACTEE a149
 - RANGE:
 - OPTION a150
 - LABEL:
 - GUI-SAVE a151
 - OPTIONS:
 - GUI-ACTEE a152

RANGE:
 MENU a153
 ACTOR:
 USER-ACTOR a154
 RANGE:
 USER a6
 SIDE-EFFECT:
 SEE a155
 SENSER:
 SEE-SENSER a156
 RANGE:
 USER a6
 PHENOMENON:
 SEE-PHENOMENON a157
 RANGE:
 DIALOGUE-BOX a158
 LABEL:
 GUI-SAVE a159
 REST:
 PROCEDURE-LIST a160
 FIRST:
 PROCEDURE a161
 GOAL:
 ENTER a162
 ACTEE:
 CONFIGURED-ACTEE a163
 RANGE:
 DATA-NAME a164
 OWNER:
 DRAWING a141
 LOCATION:
 GUI-ACTEE a165
 RANGE:
 FIELD a166
 LABEL:
 GUI-NAME a167
 ACTOR:
 USER-ACTOR a168
 RANGE:
 USER a6
 REST:
 PROCEDURE-LIST a169
 FIRST:
 PROCEDURE a170
 GOAL:
 ENTER a171
 ACTEE:
 CONFIGURED-ACTEE a172
 RANGE:
 DESCRIPTION a173
 LOCATION:
 GUI-ACTEE a174
 RANGE:
 FIELD a175
 LABEL:
 GUI-DESCRIPTION a176
 ACTOR:
 USER-ACTOR a177
 RANGE:

USER a6
REST:
PROCEDURE-LIST a178
FIRST:
PROCEDURE a179
GOAL:
CLOSE-SCREEN-OBJECT a180
ACTEE:
DISPLAYED-ACTEE a181
RANGE:
DIALOGUE-BOX a158
LABEL:
GUI-SAVE a159
ACTOR:
USER-ACTOR a182
RANGE:
USER a6
METHODS:
METHOD-LIST a183
FIRST:
METHOD* a184
SUBSTEPS:
PROCEDURE-LIST a185
FIRST:
PROCEDURE a186
GOAL:
CLICK a187
ACTEE:
DISPLAYED-ACTEE a188
RANGE:
BUTTON a189
LABEL:
GUI-OK a190
LOCATION:
GUI-ACTEE a191
ACTOR:
USER-ACTOR a192
RANGE:
USER a6
SIDE-EFFECT:
NOT-SEE a193
SENDER:
SEE-SENDER a194
RANGE:
USER a6
PHENOMENON:
SEE-PHENOMENON a195
RANGE:
DIALOGUE-BOX a158
LABEL:
GUI-SAVE a159

Приложение 3: Онтологии в OWL на предметна област „Иконография“

Онтология ОБИО представена в OWL

Съдържанието на файл OBIO-SINUS_PD4.2.3 .owl е показано частично поради големия му обем. В предложената „разпечатка“ присъстват специализираните класове на онтологията ОБИО-СИНУС, както и съществените за ОБИО-СИНУС класове и свойства наследени от онтологията от високо ниво CIDOC-CRM.

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY cidoc_crm_v5 "http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#" >
]>

<rdf:RDF xmlns="http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#"
  xml:base="http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:cidoc_crm_v5="http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about=""/>

  <!--
  ////////////////////////////////////////////////////////////////////
  //
  // Object Properties
  //
  ////////////////////////////////////////////////////////////////////
  -->

  <!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P100B -->

  <owl:ObjectProperty rdf:about="#P100B">
    <rdfs:label xml:lang="en">died in</rdfs:label>
    <rdfs:label xml:lang="fr">est mort par</rdfs:label>
    <rdfs:domain rdf:resource="#E21"/>
    <rdfs:range rdf:resource="#E69"/>
    <rdfs:subPropertyOf rdf:resource="#P93B"/>
  </owl:ObjectProperty>

  <!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P100F -->

  <owl:ObjectProperty rdf:about="#P100F">
    <rdfs:label xml:lang="en">was death of</rdfs:label>
    <rdfs:range rdf:resource="#E21"/>
    <rdfs:domain rdf:resource="#E69"/>
    <rdfs:subPropertyOf rdf:resource="#P93F"/>
  </owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P101B -->

```
<owl:ObjectProperty rdf:about="#P101B">
  <rdfs:label xml:lang="en">was use of</rdfs:label>
  <rdfs:domain rdf:resource="#E55"/>
  <rdfs:range rdf:resource="#E70"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P101F -->

```
<owl:ObjectProperty rdf:about="#P101F">
  <rdfs:label xml:lang="en"
    >had as general use</rdfs:label>
  <rdfs:range rdf:resource="#E55"/>
  <rdfs:domain rdf:resource="#E70"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P102B -->

```
<owl:ObjectProperty rdf:about="#P102B">
  <rdfs:label xml:lang="en">is title of</rdfs:label>
  <rdfs:domain rdf:resource="#E35"/>
  <rdfs:range rdf:resource="#E71"/>
  <rdfs:subPropertyOf rdf:resource="#P1B"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P102F -->

```
<owl:ObjectProperty rdf:about="#P102F">
  <rdfs:label xml:lang="en">has title</rdfs:label>
  <rdfs:range rdf:resource="#E35"/>
  <rdfs:domain rdf:resource="#E71"/>
  <rdfs:subPropertyOf rdf:resource="#P1F"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P106B -->

```
<owl:ObjectProperty rdf:about="#P106B">
  <rdfs:label xml:lang="en">forms part of</rdfs:label>
  <rdfs:domain rdf:resource="#E90"/>
  <rdfs:range rdf:resource="#E90"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P106F -->

```
<owl:ObjectProperty rdf:about="#P106F">
  <rdfs:label xml:lang="en">is composed of</rdfs:label>
  <rdfs:range rdf:resource="#E90"/>
  <rdfs:domain rdf:resource="#E90"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P107B -->

```
<owl:ObjectProperty rdf:about="#P107B">
  <rdfs:label xml:lang="en"
    >is current or former member of</rdfs:label>
  <rdfs:domain rdf:resource="#E39"/>
  <rdfs:range rdf:resource="#E74"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P107F -->

```

<owl:ObjectProperty rdf:about="#P107F">
  <rdfs:label xml:lang="en"
    >has current or former member</rdfs:label>
  <rdfs:range rdf:resource="#E39"/>
  <rdfs:domain rdf:resource="#E74"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P108F -->

<owl:ObjectProperty rdf:about="#P108F">
  <rdfs:label xml:lang="en">has produced</rdfs:label>
  <rdfs:domain rdf:resource="#E12"/>
  <rdfs:range rdf:resource="#E24"/>
  <rdfs:subPropertyOf rdf:resource="#P31F"/>
  <rdfs:subPropertyOf rdf:resource="#P92F"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P10F -->

<owl:ObjectProperty rdf:about="#P10F">
  <rdfs:label xml:lang="en">falls within</rdfs:label>
  <rdfs:domain rdf:resource="#E4"/>
  <rdfs:range rdf:resource="#E4"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P128F -->

<owl:ObjectProperty rdf:about="#P128F">
  <rdfs:label xml:lang="en">carries</rdfs:label>
  <rdfs:domain rdf:resource="#E24"/>
  <rdfs:range rdf:resource="#E73"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P129F -->

<owl:ObjectProperty rdf:about="#P129F">
  <rdfs:label xml:lang="en">is about</rdfs:label>
  <rdfs:range rdf:resource="#E1"/>
  <rdfs:domain rdf:resource="#E89"/>
  <rdfs:subPropertyOf rdf:resource="#P67F"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P137F -->

<owl:ObjectProperty rdf:about="#P137F">
  <rdfs:label xml:lang="en">exemplifies</rdfs:label>
  <rdfs:domain rdf:resource="#E1"/>
  <rdfs:range rdf:resource="#E55"/>
  <rdfs:subPropertyOf rdf:resource="#P2F"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P138F -->

<owl:ObjectProperty rdf:about="#P138F">
  <rdfs:label xml:lang="en">represents</rdfs:label>
  <rdfs:range rdf:resource="#E1"/>

```

```

    <rdfs:domain rdf:resource="#E36"/>
    <rdfs:subPropertyOf rdf:resource="#P67F"/>
  </owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P143F -->

  <owl:ObjectProperty rdf:about="#P143F">
    <rdfs:label xml:lang="en">joined</rdfs:label>
    <rdfs:range rdf:resource="#E39"/>
    <rdfs:domain rdf:resource="#E85"/>
    <rdfs:subPropertyOf rdf:resource="#P11F"/>
  </owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P144F -->

  <owl:ObjectProperty rdf:about="#P144F">
    <rdfs:label xml:lang="en">joined with</rdfs:label>
    <rdfs:range rdf:resource="#E74"/>
    <rdfs:domain rdf:resource="#E85"/>
    <rdfs:subPropertyOf rdf:resource="#P11F"/>
  </owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P148F -->

  <owl:ObjectProperty rdf:about="#P148F">
    <rdfs:label xml:lang="en">has component</rdfs:label>
    <rdfs:range rdf:resource="#E89"/>
    <rdfs:domain rdf:resource="#E89"/>
  </owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P14F -->

  <owl:ObjectProperty rdf:about="#P14F">
    <rdfs:label xml:lang="en">carried out by</rdfs:label>
    <rdfs:range rdf:resource="#E39"/>
    <rdfs:domain rdf:resource="#E7"/>
    <rdfs:subPropertyOf rdf:resource="#P11F"/>
  </owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P2F -->

  <owl:ObjectProperty rdf:about="#P2F">
    <rdfs:label xml:lang="en">has type</rdfs:label>
    <rdfs:domain rdf:resource="#E1"/>
    <rdfs:range rdf:resource="#E55"/>
  </owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P32F -->

  <owl:ObjectProperty rdf:about="#P32F">
    <rdfs:label xml:lang="en">
      >used general technique</rdfs:label>
    <rdfs:range rdf:resource="#E55"/>
    <rdfs:domain rdf:resource="#E7"/>
    <rdfs:subPropertyOf rdf:resource="#P125F"/>
  </owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P33F -->

```



```

<owl:ObjectProperty rdf:about="#P33F">
  <rdfs:label xml:lang="en"
    >used specific technique</rdfs:label>
  <rdfs:range rdf:resource="#E29"/>
  <rdfs:domain rdf:resource="#E7"/>
  <rdfs:subPropertyOf rdf:resource="#P16F"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P3F -->

<owl:ObjectProperty rdf:about="#P3F">
  <rdfs:label xml:lang="en">has note</rdfs:label>
  <rdfs:domain rdf:resource="#E1"/>
  <rdfs:range rdf:resource="#&rdfs;Literal"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P43F -->

<owl:ObjectProperty rdf:about="#P43F">
  <rdfs:label xml:lang="en">has dimension</rdfs:label>
  <rdfs:range rdf:resource="#E54"/>
  <rdfs:domain rdf:resource="#E70"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P45F -->

<owl:ObjectProperty rdf:about="#P45F">
  <rdfs:label xml:lang="en">consists of</rdfs:label>
  <rdfs:domain rdf:resource="#E18"/>
  <rdfs:range rdf:resource="#E57"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P46F -->

<owl:ObjectProperty rdf:about="#P46F">
  <rdfs:label xml:lang="en">is composed of</rdfs:label>
  <rdfs:range rdf:resource="#E18"/>
  <rdfs:domain rdf:resource="#E18"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P4F -->

<owl:ObjectProperty rdf:about="#P4F">
  <rdfs:label xml:lang="en">has time-span</rdfs:label>
  <rdfs:domain rdf:resource="#E2"/>
  <rdfs:range rdf:resource="#E52"/>
</owl:ObjectProperty>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P53F -->

<owl:ObjectProperty rdf:about="#P53F">
  <rdfs:label xml:lang="en"
    >has former or current location</rdfs:label>
  <rdfs:domain rdf:resource="#E18"/>
  <rdfs:range rdf:resource="#E53"/>
</owl:ObjectProperty>

```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P55F -->

```
<owl:ObjectProperty rdf:about="#P55F">
  <rdfs:label xml:lang="en"
    >has current location</rdfs:label>
  <rdfs:domain rdf:resource="#E19"/>
  <rdfs:range rdf:resource="#E53"/>
  <rdfs:subPropertyOf rdf:resource="#P53F"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P5F -->

```
<owl:ObjectProperty rdf:about="#P5F">
  <rdfs:label xml:lang="en">consists of</rdfs:label>
  <rdfs:range rdf:resource="#E3"/>
  <rdfs:domain rdf:resource="#E3"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P62F -->

```
<owl:ObjectProperty rdf:about="#P62F">
  <rdfs:label xml:lang="en">depicts</rdfs:label>
  <rdfs:range rdf:resource="#E1"/>
  <rdfs:domain rdf:resource="#E24"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P65F -->

```
<owl:ObjectProperty rdf:about="#P65F">
  <rdfs:label xml:lang="en"
    >shows visual item</rdfs:label>
  <rdfs:domain rdf:resource="#E24"/>
  <rdfs:range rdf:resource="#E36"/>
  <rdfs:subPropertyOf rdf:resource="#P128F"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P67F -->

```
<owl:ObjectProperty rdf:about="#P67F">
  <rdfs:label xml:lang="en">refers to</rdfs:label>
  <rdfs:range rdf:resource="#E1"/>
  <rdfs:domain rdf:resource="#E89"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P7F -->

```
<owl:ObjectProperty rdf:about="#P7F">
  <rdfs:label xml:lang="en">took place at</rdfs:label>
  <rdfs:domain rdf:resource="#E4"/>
  <rdfs:range rdf:resource="#E53"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P86F -->

```
<owl:ObjectProperty rdf:about="#P86F">
  <rdfs:label xml:lang="en">falls within</rdfs:label>
  <rdfs:domain rdf:resource="#E52"/>
  <rdfs:range rdf:resource="#E52"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P88F -->

```
<owl:ObjectProperty rdf:about="#P88F">
  <rdfs:label xml:lang="en">consists of</rdfs:label>
  <rdfs:range rdf:resource="#E53"/>
  <rdfs:domain rdf:resource="#E53"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P89F -->

```
<owl:ObjectProperty rdf:about="#P89F">
  <rdfs:label xml:lang="en">falls within</rdfs:label>
  <rdfs:domain rdf:resource="#E53"/>
  <rdfs:range rdf:resource="#E53"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P91B -->

```
<owl:ObjectProperty rdf:about="#P91B">
  <rdfs:label xml:lang="en">is unit of</rdfs:label>
  <rdfs:range rdf:resource="#E54"/>
  <rdfs:domain rdf:resource="#E58"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P91F -->

```
<owl:ObjectProperty rdf:about="#P91F">
  <rdfs:label xml:lang="en">has unit</rdfs:label>
  <rdfs:domain rdf:resource="#E54"/>
  <rdfs:range rdf:resource="#E58"/>
</owl:ObjectProperty>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#P9F -->

```
<owl:ObjectProperty rdf:about="#P9F">
  <rdfs:label xml:lang="en">consists of</rdfs:label>
  <rdfs:range rdf:resource="#E4"/>
  <rdfs:domain rdf:resource="#E4"/>
</owl:ObjectProperty>
```

<!--

//

//

// Classes

//

//

-->

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E1 -->

```
<owl:Class rdf:about="#E1">
  <rdfs:label xml:lang="en">CRM Entity</rdfs:label>
</owl:Class>
```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E11 -->

```
<owl:Class rdf:about="#E11">
  <rdfs:label xml:lang="en">Modification</rdfs:label>
```

```

<rdfs:subClassOf rdf:resource="#E7"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E12 -->

<owl:Class rdf:about="#E12">
  <rdfs:label xml:lang="en">Production</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E11"/>
  <rdfs:subClassOf rdf:resource="#E63"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E13 -->

<owl:Class rdf:about="#E13">
  <rdfs:label xml:lang="en"
    >Attribute Assignment</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E7"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E14 -->

<owl:Class rdf:about="#E14">
  <rdfs:label xml:lang="en"
    >Condition Assessment</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E13"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E15 -->

<owl:Class rdf:about="#E15">
  <rdfs:label xml:lang="en"
    >Identifier Assignment</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E13"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E16 -->

<owl:Class rdf:about="#E16">
  <rdfs:label xml:lang="en">Measurement</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E13"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E18 -->

<owl:Class rdf:about="#E18">
  <rdfs:label xml:lang="en">Physical Thing</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E72"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E19 -->

<owl:Class rdf:about="#E19">
  <rdfs:label xml:lang="en">Physical Object</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E18"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E71 -->

```

```

<owl:Class rdf:about="#E71">
  <rdfs:label xml:lang="en">Man-Made Thing</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E70"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E72 -->

<owl:Class rdf:about="#E72">
  <rdfs:label xml:lang="en">Legal Object</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E70"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E73 -->

<owl:Class rdf:about="#E73">
  <rdfs:label xml:lang="en"
    >Information Object</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E89"/>
  <rdfs:subClassOf rdf:resource="#E90"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E74 -->

<owl:Class rdf:about="#E74">
  <rdfs:label xml:lang="en">Group</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E39"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#E78 -->

<owl:Class rdf:about="#E78">
  <rdfs:label xml:lang="en">Collection</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E24"/>
</owl:Class>

...

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000000424934700918 -->
<owl:Class rdf:about="#OWLClass_00000000424934700918">
  <rdfs:label
    >Iconographical Object</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E22"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002136549768196 -->
<owl:Class rdf:about="#OWLClass_00000002136549768196">
  <rdfs:label>Height</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E54"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002169024510098 -->
<owl:Class rdf:about="#OWLClass_00000002169024510098">
  <rdfs:label>Width</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E54"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002192581145089 -->

```

```

<owl:Class rdf:about="#OWLClass_00000002192581145089">
  <rdfs:label>Thickness</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E54"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002223537732030 -->
<owl:Class rdf:about="#OWLClass_00000002223537732030">
  <rdfs:label>State</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E53"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002772576796212 -->
<owl:Class rdf:about="#OWLClass_00000002772576796212">
  <rdfs:label>Region</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E53"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002829675576644 -->
<owl:Class rdf:about="#OWLClass_00000002829675576644">
  <rdfs:label>Town</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E53"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002849480856365 -->
<owl:Class rdf:about="#OWLClass_00000002849480856365">
  <rdfs:label>Village</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E53"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002862124476993 -->

<owl:Class rdf:about="#OWLClass_00000002862124476993">
  <rdfs:label>Monastery</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E53"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002879633319318 -->
<owl:Class rdf:about="#OWLClass_00000002879633319318">
  <rdfs:label>Church</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E53"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002913694890272 -->
<owl:Class rdf:about="#OWLClass_00000002913694890272">
  <rdfs:label>Chapel</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E53"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002922563191411 -->
<owl:Class rdf:about="#OWLClass_00000002922563191411">
  <rdfs:label>Museum</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E53"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002944528725019 -->
<owl:Class rdf:about="#OWLClass_00000002944528725019">

```

```

    <rdfs:label>Gallery</rdfs:label>
    <rdfs:subClassOf rdf:resource="#E53"/>
  </owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000002958172437279 -->
<owl:Class rdf:about="#OWLClass_00000002958172437279">
  <rdfs:label>Year</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E52"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000003083097374386 -->
<owl:Class rdf:about="#OWLClass_00000003083097374386">
  <rdfs:label>Month</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E52"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000003093917614389 -->

  <owl:Class rdf:about="#OWLClass_00000003093917614389">
    <rdfs:label>Day</rdfs:label>
    <rdfs:subClassOf rdf:resource="#E52"/>
  </owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000003110703056394 -->
<owl:Class rdf:about="#OWLClass_00000003110703056394">
  <rdfs:label>Century</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E52"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000003124532387775 -->
<owl:Class rdf:about="#OWLClass_00000003124532387775">
  <rdfs:label>Part of Century</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E52"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000003154894411719 -->

  <owl:Class rdf:about="#OWLClass_00000003154894411719">
    <rdfs:label
      >Important Event for East-Christian Religion</rdfs:label>
    <rdfs:subClassOf rdf:resource="#E5"/>
  </owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000003501580836111 -->
<owl:Class rdf:about="#OWLClass_00000003501580836111">
  <rdfs:label
    >Iconographical Object Production</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E12"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000003690882131108 -->
<owl:Class rdf:about="#OWLClass_00000003690882131108">
  <rdfs:label
    >Iconographic Clan</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E74"/>
</owl:Class>

```

```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000003982102192038 -->
<owl:Class rdf:about="#OWLClass_00000003982102192038">
  <rdfs:label>Iconographer</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E21"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004048214992966 -->
<owl:Class rdf:about="#OWLClass_00000004048214992966">
  <rdfs:label
    >Important Person for East- Christian Religion</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E21"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004093765617976 -->
<owl:Class rdf:about="#OWLClass_00000004093765617976">
  <rdfs:label
    >Base of Iconographical Object</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E24"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004360414445690 -->
<owl:Class rdf:about="#OWLClass_00000004360414445690">
  <rdfs:label>Icon</rdfs:label>
  <rdfs:subClassOf rdf:resource="#OWLClass_0000000424934700918"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004502161475271 -->
<owl:Class rdf:about="#OWLClass_00000004502161475271">
  <rdfs:label>Wall-Painting</rdfs:label>
  <rdfs:subClassOf rdf:resource="#OWLClass_0000000424934700918"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004525609852928 -->
<owl:Class rdf:about="#OWLClass_00000004525609852928">
  <rdfs:label>Miniature</rdfs:label>
  <rdfs:subClassOf rdf:resource="#OWLClass_0000000424934700918"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004542883540505 -->
<owl:Class rdf:about="#OWLClass_00000004542883540505">
  <rdfs:label>Mosaic</rdfs:label>
  <rdfs:subClassOf rdf:resource="#OWLClass_0000000424934700918"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004565471008186 -->
<owl:Class rdf:about="#OWLClass_00000004565471008186">
  <rdfs:label>&#1042;&#1080;&#1090;&#1088;&#1072;&#1078;</rdfs:label>
  <rdfs:subClassOf rdf:resource="#OWLClass_0000000424934700918"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004590533859750 -->
<owl:Class rdf:about="#OWLClass_00000004590533859750">
  <rdfs:label
    >Plastic Iconographical Object</rdfs:label>
  <rdfs:subClassOf rdf:resource="#OWLClass_0000000424934700918"/>
</owl:Class>

```



```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004621508865232 -->
<owl:Class rdf:about="#OWLClass_00000004621508865232">
<rdfs:label>&#1048;&#1082;&#1086;&#1085;&#1086;&#1089;&#1090;&#1072;&#1089;</rdfs:label>
<rdfs:subClassOf rdf:resource="#OWLClass_00000004590533859750"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004646661990560 -->
<owl:Class rdf:about="#OWLClass_00000004646661990560">
<rdfs:label>&#1055;&#1088;&#1077;&#1089;&#1090;&#1086;&#1083;
</rdfs:label>
<rdfs:subClassOf rdf:resource="#OWLClass_00000004590533859750"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004663774326904 -->
<owl:Class rdf:about="#OWLClass_00000004663774326904">
<rdfs:label
>Iconographic School</rdfs:label>
<rdfs:subClassOf rdf:resource="#E55"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004863040895700 -->
<owl:Class rdf:about="#OWLClass_00000004863040895700">
<rdfs:label
>Iconographical Technique</rdfs:label>
<rdfs:subClassOf rdf:resource="#E55"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004898929473159 -->
<owl:Class rdf:about="#OWLClass_00000004898929473159">
<rdfs:label
>Canonic Type of Iconographical Image</rdfs:label>
<rdfs:subClassOf rdf:resource="#E55"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000004940415430452 -->
<owl:Class rdf:about="#OWLClass_00000004940415430452">
<rdfs:label
>Iconographical Image</rdfs:label>
<rdfs:subClassOf rdf:resource="#E38"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000005144401276419 -->
<owl:Class rdf:about="#OWLClass_00000005144401276419">
<rdfs:label
>One- Figure Composition</rdfs:label>
<rdfs:subClassOf rdf:resource="#OWLClass_00000004940415430452"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000005167514487455 -->
<owl:Class rdf:about="#OWLClass_00000005167514487455">
<rdfs:label
>Many- Figures Composition</rdfs:label>
<rdfs:subClassOf rdf:resource="#OWLClass_00000004940415430452"/>
</owl:Class>

```

```

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000005199296306932 -->
<owl:Class rdf:about="#OWLClass_00000005199296306932">
  <rdfs:label
    >Composition of Complete Compositions</rdfs:label>
  <rdfs:subClassOf rdf:resource="#OWLClass_00000004940415430452"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000005249932647026 -->
<owl:Class rdf:about="#OWLClass_00000005249932647026">
  <rdfs:label>Biography</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E33"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000005477965338611 -->
<owl:Class rdf:about="#OWLClass_00000005477965338611">
  <rdfs:label
    >IO Identification Note</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E33"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000005546340470925 -->
<owl:Class rdf:about="#OWLClass_00000005546340470925">
  <rdfs:label>IO Description</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E33"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000005572844740043 -->
<owl:Class rdf:about="#OWLClass_00000005572844740043">
  <rdfs:label
    >Iconographical Technique Description</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E33"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000005670227659101 -->
<owl:Class rdf:about="#OWLClass_00000005670227659101">
  <rdfs:label
    >Base Description</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E33"/>
</owl:Class>

<!-- http://cidoc.ics.forth.gr/rdfs/cidoc_crm_v5.0.1.rdfs#OWLClass_00000005706645450748 -->
<owl:Class rdf:about="#OWLClass_00000005706645450748">
  <rdfs:label
    >Iconographical Object Condition State</rdfs:label>
  <rdfs:subClassOf rdf:resource="#E33"/>
</owl:Class>

<!-- http://www.w3.org/2000/01/rdf-schema#Literal -->
<owl:Class rdf:about="#<rdfs:Literal"/>
</rdf:RDF>

<!-- Generated by the OWL API (version 2.2.1.1138) http://owlapi.sourceforge.net -->

```

Онтология „Технология на иконографски обект“ представена в OWL

Листинг на OWL реализацията на специализираната онтология SINUS_TechnologySpecOntology

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY SINUSBasicOntology "http://www.semanticweb.org/ontologies/2011/1/SINUSBasicOntology.owl#" >
  <!ENTITY SINUS_TechnologySpecOntology "http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#" >
]>

<rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#"
  xml:base="http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:SINUSBasicOntology="http://www.semanticweb.org/ontologies/2011/1/SINUSBasicOntology.owl#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:SINUS_TechnologySpecOntology="http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about=""/>

  <!--
  ////////////////////////////////////////////////////////////////////
  //
  // Object Properties
  //
  ////////////////////////////////////////////////////////////////////
  -->

  <!--
  http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_base_has_Component
  -->
  <owl:ObjectProperty rdf:about="#OWLObjectProperty_base_has_Component">
    <rdfs:label xml:lang="bg">има компонент</rdfs:label>
    <rdfs:label xml:lang="en">base has Component</rdfs:label>
    <rdfs:domain rdf:resource="#OWLClass_Base"/>
    <rdfs:range rdf:resource="#OWLClass_Primer"/>
  </owl:ObjectProperty>

  <!--
  http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_gilding_has_Base
  -->
  <owl:ObjectProperty rdf:about="#OWLObjectProperty_gilding_has_Base">
    <rdfs:label xml:lang="en">gilding has Base</rdfs:label>
    <rdfs:comment xml:lang="bg">има основа</rdfs:comment>
    <rdfs:range rdf:resource="#OWLClass_BaseOfGilding"/>
    <rdfs:domain rdf:resource="#OWLClass_Gilding"/>
  </owl:ObjectProperty>

  <!--
  http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_gilding_has_Type
  -->
  <owl:ObjectProperty rdf:about="#OWLObjectProperty_gilding_has_Type">
    <rdfs:label xml:lang="bg">е от тип</rdfs:label>
    <rdfs:label xml:lang="en">gilding has Type</rdfs:label>
    <rdfs:domain rdf:resource="#OWLClass_Gilding"/>
```

```

<rdfs:range rdf:resource="#OWLClass_TypeOfGilding"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_iTechnique_uses_Gilding
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_iTechnique_uses_Gilding">
<rdfs:label xml:lang="bg">използва позлата</rdfs:label>
<rdfs:label xml:lang="en">iconographical Technique uses Gilding</rdfs:label>
<rdfs:range rdf:resource="#OWLClass_Gilding"/>
<rdfs:domain rdf:resource="#OWLClass_IconographicalTechnique"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_iTechnique_uses_Lacquering
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_iTechnique_uses_Lacquering">
<rdfs:label xml:lang="bg">използва лак</rdfs:label>
<rdfs:label xml:lang="en">iconographical Technique uses Lacquering</rdfs:label>
<rdfs:domain rdf:resource="#OWLClass_IconographicalTechnique"/>
<rdfs:range rdf:resource="#OWLClass_Lacquering"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_lacquering_has_Condition
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_lacquering_has_Condition">
<rdfs:label xml:lang="bg">е в състояние</rdfs:label>
<rdfs:label xml:lang="en">lacquering has Condition</rdfs:label>
<rdfs:range rdf:resource="#OWLClass_ConditionOfLacquering"/>
<rdfs:domain rdf:resource="#OWLClass_Lacquering"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_lacquering_has_Evenness
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_lacquering_has_Evenness">
<rdfs:label xml:lang="bg">е с равномерност</rdfs:label>
<rdfs:label xml:lang="en">lacquering has Evenness</rdfs:label>
<rdfs:range rdf:resource="#OWLClass_EvennessOfLacquering"/>
<rdfs:domain rdf:resource="#OWLClass_Lacquering"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_lacquering_has_Thickness
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_lacquering_has_Thickness">
<rdfs:label xml:lang="bg">е с дебелина</rdfs:label>
<rdfs:label xml:lang="en">lacquering has Thickness</rdfs:label>
<rdfs:domain rdf:resource="#OWLClass_Lacquering"/>
<rdfs:range rdf:resource="#OWLClass_ThicknessOfLacquering"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_lacquering_has_Type
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_lacquering_has_Type">
<rdfs:label xml:lang="bg">е от тип</rdfs:label>
<rdfs:label xml:lang="en">lacquering has Type</rdfs:label>
<rdfs:domain rdf:resource="#OWLClass_Lacquering"/>
<rdfs:range rdf:resource="#OWLClass_TypeOfLacquering"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_primer_has_Condition

```

```

-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_primer_has_Condition">
  <rdfs:label xml:lang="bg">е в състояние</rdfs:label>
  <rdfs:label xml:lang="en">primer has Condition</rdfs:label>
  <rdfs:range rdf:resource="#OWLClass_ConditionOfPrimer"/>
  <rdfs:domain rdf:resource="#OWLClass_Primer"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_primer_has_Evenness
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_primer_has_Evenness">
  <rdfs:label xml:lang="bg">е с равномерност</rdfs:label>
  <rdfs:label xml:lang="en">primer has Evenness</rdfs:label>
  <rdfs:range rdf:resource="#OWLClass_EvennessOfPrimer"/>
  <rdfs:domain rdf:resource="#OWLClass_Primer"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_primer_has_Filler
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_primer_has_Filler">
  <rdfs:label xml:lang="bg">е с пълнител</rdfs:label>
  <rdfs:label xml:lang="en">primer has Filler</rdfs:label>
  <rdfs:range rdf:resource="#OWLClass_Filler"/>
  <rdfs:domain rdf:resource="#OWLClass_Primer"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_primer_has_Thickness
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_primer_has_Thickness">
  <rdfs:label xml:lang="bg">е с дебелина</rdfs:label>
  <rdfs:label xml:lang="en">primer has Thickness</rdfs:label>
  <rdfs:domain rdf:resource="#OWLClass_Primer"/>
  <rdfs:range rdf:resource="#OWLClass_ThicknessOfPrimer"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLObjectProperty_primer_has_Type
-->
<owl:ObjectProperty rdf:about="#OWLObjectProperty_primer_has_Type">
  <rdfs:label xml:lang="bg">е тип</rdfs:label>
  <rdfs:label xml:lang="en">primer has Type</rdfs:label>
  <rdfs:domain rdf:resource="#OWLClass_Primer"/>
  <rdfs:range rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:ObjectProperty>

<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_Base
-->
<owl:Class rdf:about="#OWLClass_Base">
  <rdfs:label xml:lang="bg">Основа</rdfs:label>
  <rdfs:label xml:lang="en">Base</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#OWLObjectProperty_base_has_Component"/>

```

```

<owl:onClass rdf:resource="#OWLClass_Primer"/>
<owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#OWLClass_BaseOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_ConditionOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ConditionOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_EvennessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_EvennessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_Filler"/>
<owl:disjointWith rdf:resource="#OWLClass_Gilding"/>
<owl:disjointWith rdf:resource="#OWLClass_IconographicalTechnique"/>
<owl:disjointWith rdf:resource="#OWLClass_Lacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
<owl:equivalentClass rdf:resource="http://www.semanticweb.org/ontologies/2011/1/SINUSBasicOntology.owl#OWLClass_Base"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_BaseOfGilding
-->
<owl:Class rdf:about="#OWLClass_BaseOfGilding">
<rdfs:label xml:lang="bg">Основа на позлата</rdfs:label>
<rdfs:label xml:lang="en">Base of Gilding</rdfs:label>
<owl:disjointWith rdf:resource="#OWLClass_ConditionOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ConditionOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_EvennessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_EvennessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_Filler"/>
<owl:disjointWith rdf:resource="#OWLClass_Gilding"/>
<owl:disjointWith rdf:resource="#OWLClass_IconographicalTechnique"/>
<owl:disjointWith rdf:resource="#OWLClass_Lacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_ConditionOfLacquering
-->
<owl:Class rdf:about="#OWLClass_ConditionOfLacquering">
<rdfs:label xml:lang="bg">Състояние на лака</rdfs:label>
<rdfs:label xml:lang="bg">Състояние на лаково покритие</rdfs:label>
<rdfs:label xml:lang="en">Condition of Lacquering</rdfs:label>
<owl:disjointWith rdf:resource="#OWLClass_ConditionOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_EvennessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_EvennessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_Filler"/>
<owl:disjointWith rdf:resource="#OWLClass_Gilding"/>
<owl:disjointWith rdf:resource="#OWLClass_IconographicalTechnique"/>
<owl:disjointWith rdf:resource="#OWLClass_Lacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

```

```

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_ConditionOfPrimer
-->
<owl:Class rdf:about="#OWLClass_ConditionOfPrimer">
<rdfs:label xml:lang="bg">Състояние на грунда</rdfs:label>
<rdfs:label xml:lang="en">Condition of Primer</rdfs:label>
<owl:disjointWith rdf:resource="#OWLClass_EvennessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_EvennessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_Filler"/>
<owl:disjointWith rdf:resource="#OWLClass_Gilding"/>
<owl:disjointWith rdf:resource="#OWLClass_IconographicalTechnique"/>
<owl:disjointWith rdf:resource="#OWLClass_Lacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_EvennessOfLacquering
-->
<owl:Class rdf:about="#OWLClass_EvennessOfLacquering">
<rdfs:label xml:lang="bg">Равномерност на лака</rdfs:label>
<rdfs:label xml:lang="bg">Равномерност на лаково покритие</rdfs:label>
<rdfs:label xml:lang="en">Evenness of Lacquering</rdfs:label>
<owl:disjointWith rdf:resource="#OWLClass_EvennessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_Filler"/>
<owl:disjointWith rdf:resource="#OWLClass_Gilding"/>
<owl:disjointWith rdf:resource="#OWLClass_IconographicalTechnique"/>
<owl:disjointWith rdf:resource="#OWLClass_Lacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_EvennessOfPrimer
-->
<owl:Class rdf:about="#OWLClass_EvennessOfPrimer">
<rdfs:label xml:lang="bg">Равномерност на грунда</rdfs:label>
<rdfs:label xml:lang="en">Evenness of Primer</rdfs:label>
<owl:disjointWith rdf:resource="#OWLClass_Filler"/>
<owl:disjointWith rdf:resource="#OWLClass_Gilding"/>
<owl:disjointWith rdf:resource="#OWLClass_IconographicalTechnique"/>
<owl:disjointWith rdf:resource="#OWLClass_Lacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_Filler
-->
<owl:Class rdf:about="#OWLClass_Filler">
<rdfs:label xml:lang="bg">Пълнител</rdfs:label>
<rdfs:label xml:lang="en">Filler</rdfs:label>

```

```

<owl:disjointWith rdf:resource="#OWLClass_Gilding"/>
<owl:disjointWith rdf:resource="#OWLClass_IconographicalTechnique"/>
<owl:disjointWith rdf:resource="#OWLClass_Lacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_Gilding
-->
<owl:Class rdf:about="#OWLClass_Gilding">
<rdfs:label xml:lang="bg">Позлатя</rdfs:label>
<rdfs:label xml:lang="en">Gilding</rdfs:label>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_gilding_has_Type"/>
<owl:onClass rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_gilding_has_Base"/>
<owl:onClass rdf:resource="#OWLClass_BaseOfGilding"/>
<owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#OWLClass_IconographicalTechnique"/>
<owl:disjointWith rdf:resource="#OWLClass_Lacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_IconographicalTechnique
-->
<owl:Class rdf:about="#OWLClass_IconographicalTechnique">
<rdfs:label xml:lang="bg">Иконографска техника</rdfs:label>
<rdfs:label xml:lang="en">Iconographical Technique</rdfs:label>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_iTechnique_uses_Gilding"/>
<owl:allValuesFrom rdf:resource="#OWLClass_Gilding"/>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_iTechnique_uses_Lacquering"/>
<owl:allValuesFrom rdf:resource="#OWLClass_Lacquering"/>
</owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#OWLClass_Lacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>

```



```

<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
<owl:equivalentClass rdf:resource="http://www.semanticweb.org/ontologies/2011/1/SINUSBasicOntology.owl#OWLClass_IconographicalTechnique"/>
</owl:Class>

```

```

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_Lacquering
-->
<owl:Class rdf:about="#OWLClass_Lacquering">
<rdfs:label xml:lang="bg">Лак</rdfs:label>
<rdfs:label xml:lang="bg">Лаково покритие</rdfs:label>
<rdfs:label xml:lang="en">Lacquering</rdfs:label>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_lacquering_has_Type"/>
<owl:onClass rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_lacquering_has_Condition"/>
<owl:someValuesFrom rdf:resource="#OWLClass_ConditionOfLacquering"/>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_lacquering_has_Thickness"/>
<owl:someValuesFrom rdf:resource="#OWLClass_ThicknessOfLacquering"/>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_lacquering_has_Evenness"/>
<owl:someValuesFrom rdf:resource="#OWLClass_EvennessOfLacquering"/>
</owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#OWLClass_Primer"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

```

```

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_Primer
-->
<owl:Class rdf:about="#OWLClass_Primer">
<rdfs:label xml:lang="bg">Грунд</rdfs:label>
<rdfs:label xml:lang="en">Primer</rdfs:label>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_primer_has_Thickness"/>
<owl:onClass rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_primer_has_Evenness"/>
<owl:onClass rdf:resource="#OWLClass_EvennessOfPrimer"/>
<owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>

```

```

<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_primer_has_Type"/>
<owl:onClass rdf:resource="#OWLClass_TypeOfPrimer"/>
<owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_primer_has_Filler"/>
<owl:onClass rdf:resource="#OWLClass_Filler"/>
<owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#OWLObjectProperty_primer_has_Condition"/>
<owl:onClass rdf:resource="#OWLClass_ConditionOfPrimer"/>
<owl:qualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:qualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

```

```

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_ThicknessOfLacquering
-->
<owl:Class rdf:about="#OWLClass_ThicknessOfLacquering">
<rdfs:label xml:lang="bg">Дебелина на лака</rdfs:label>
<rdfs:label xml:lang="en">Thickness of Lacquering</rdfs:label>
<owl:disjointWith rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

```

```

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_ThicknessOfPrimer
-->
<owl:Class rdf:about="#OWLClass_ThicknessOfPrimer">
<rdfs:label xml:lang="bg">Дебелина на грунда</rdfs:label>
<rdfs:label xml:lang="en">Thickness of Primer</rdfs:label>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfGilding"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

```

```

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_TypeOfGilding
-->
<owl:Class rdf:about="#OWLClass_TypeOfGilding">
<rdfs:label xml:lang="bg">Тип позлата</rdfs:label>
<rdfs:label xml:lang="en">Type of Gilding</rdfs:label>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfLacquering"/>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

```

```

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_TypeOfLacquering
-->
<owl:Class rdf:about="#OWLClass_TypeOfLacquering">
<rdfs:label xml:lang="bg">Тип лак</rdfs:label>

```

```

<rdfs:label xml:lang="bg">Тип лаково покритие</rdfs:label>
<rdfs:label xml:lang="en">Type of Lacquering</rdfs:label>
<owl:disjointWith rdf:resource="#OWLClass_TypeOfPrimer"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLClass_TypeOfPrimer
-->
<owl:Class rdf:about="#OWLClass_TypeOfPrimer">
<rdfs:label xml:lang="bg">Тип грунд</rdfs:label>
<rdfs:label xml:lang="en">Type of Primer</rdfs:label>
</owl:Class>

<!-- http://www.w3.org/2002/07/owl#Thing -->
<owl:Class rdf:about="http://www.w3.org/2002/07/owl#Thing"/>

<!--
//
//
// Individuals
//
//
-->

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_BaseOfGilding01
-->
<OWLClass_BaseOfGilding rdf:about="#OWLIndividual_BaseOfGilding01">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">полимент</rdfs:label>
</OWLClass_BaseOfGilding>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_BaseOfGilding02
-->
<OWLClass_BaseOfGilding rdf:about="#OWLIndividual_BaseOfGilding02">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">бонус</rdfs:label>
</OWLClass_BaseOfGilding>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_BaseOfGilding03
-->
<OWLClass_BaseOfGilding rdf:about="#OWLIndividual_BaseOfGilding03">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">без основа</rdfs:label>
</OWLClass_BaseOfGilding>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_CombinedPrimer
-->
<OWLClass_TypeOfPrimer rdf:about="#OWLIndividual_CombinedPrimer">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">Комбиниран грунд</rdfs:label>
<rdfs:label xml:lang="en">Combined Primer</rdfs:label>
</OWLClass_TypeOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering01
-->
<OWLClass_ConditionOfLacquering rdf:about="#OWLIndividual_ConditionOfLacquering01">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">добро състояние</rdfs:label>
</OWLClass_ConditionOfLacquering>

```

```

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering02
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfLacquering02">
<rdf:type rdf:resource="#OWLClass_ConditionOfLacquering"/>
<rdfs:label xml:lang="bg">замърсено лаково покритие</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering03
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfLacquering03">
<rdf:type rdf:resource="#OWLClass_ConditionOfLacquering"/>
<rdfs:label xml:lang="bg">лаково покритие с натичания</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering04
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfLacquering04">
<rdf:type rdf:resource="#OWLClass_ConditionOfLacquering"/>
<rdfs:label xml:lang="bg">лошо състояние</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering05
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfLacquering05">
<rdf:type rdf:resource="#OWLClass_ConditionOfLacquering"/>
<rdfs:label xml:lang="bg">напукано лаково покритие</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering06
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfLacquering06">
<rdf:type rdf:resource="#OWLClass_ConditionOfLacquering"/>
<rdfs:label xml:lang="bg">опало лаково покритие</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering07
-->
<OWLClass_ConditionOfLacquering rdf:about="#OWLIndividual_ConditionOfLacquering07">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">пожълтяло лаково покритие</rdfs:label>
</OWLClass_ConditionOfLacquering>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering08
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfLacquering08">
<rdf:type rdf:resource="#OWLClass_ConditionOfLacquering"/>
<rdfs:label xml:lang="bg">потъмняло лаково покритие</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering09
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfLacquering09">
<rdf:type rdf:resource="#OWLClass_ConditionOfLacquering"/>
<rdfs:label xml:lang="bg">протрито лаково покритие</rdfs:label>
</owl:Thing>

<!--

```

```

http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfLacquering10
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfLacquering10">
<rdf:type rdf:resource="#OWLClass_ConditionOfLacquering"/>
<rdfs:label xml:lang="bg">разпрашен лак</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfPrimer01
-->
<OWLClass_ConditionOfPrimer rdf:about="#OWLIndividual_ConditionOfPrimer01">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">добро състояние</rdfs:label>
</OWLClass_ConditionOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfPrimer02
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfPrimer02">
<rdf:type rdf:resource="#OWLClass_ConditionOfPrimer"/>
<rdfs:label xml:lang="bg">лошо състояние</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfPrimer03
-->
<owl:Thing rdf:about="#OWLIndividual_ConditionOfPrimer03">
<rdf:type rdf:resource="#OWLClass_ConditionOfPrimer"/>
<rdfs:label xml:lang="bg">повреден грунд</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfPrimer04
-->
<OWLClass_ConditionOfPrimer rdf:about="#OWLIndividual_ConditionOfPrimer04">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">добре запазен грунд</rdfs:label>
</OWLClass_ConditionOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ConditionOfPrimer05
-->
<OWLClass_ConditionOfPrimer rdf:about="#OWLIndividual_ConditionOfPrimer05">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">здрав грунд</rdfs:label>
</OWLClass_ConditionOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_EmulsionPrimer
-->
<OWLClass_TypeOfPrimer rdf:about="#OWLIndividual_EmulsionPrimer">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">Емулсионен грунд</rdfs:label>
<rdfs:label xml:lang="en">Emulsion Primer</rdfs:label>
</OWLClass_TypeOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_EvennessOfLacquering01
-->
<owl:Thing rdf:about="#OWLIndividual_EvennessOfLacquering01">
<rdf:type rdf:resource="#OWLClass_EvennessOfLacquering"/>
<rdfs:label xml:lang="bg">неравномерно лаково покритие</rdfs:label>
</owl:Thing>

<!--

```

```

http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_EvennessOfLacquering02
-->
<owl:Thing rdf:about="#OWLIndividual_EvennessOfLacquering02">
<rdf:type rdf:resource="#OWLClass_EvennessOfLacquering"/>
<rdfs:label xml:lang="bg">равномерно лаково покритие</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_EvennessOfPrimer01
-->
<owl:Thing rdf:about="#OWLIndividual_EvennessOfPrimer01">
<rdf:type rdf:resource="#OWLClass_EvennessOfPrimer"/>
<rdfs:label xml:lang="bg">равномерно нанесен грунд</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_EvennessOfPrimer02
-->
<OWLClass_EvennessOfPrimer rdf:about="#OWLIndividual_EvennessOfPrimer02">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">неравномерно нанесен грунд</rdfs:label>
</OWLClass_EvennessOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_EvennessOfPrimer03
-->
<owl:Thing rdf:about="#OWLIndividual_EvennessOfPrimer03">
<rdf:type rdf:resource="#OWLClass_EvennessOfPrimer"/>
<rdfs:label xml:lang="bg">шлифован грунд</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_EvennessOfPrimer04
-->
<OWLClass_EvennessOfPrimer rdf:about="#OWLIndividual_EvennessOfPrimer04">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">грубо обработен грунд</rdfs:label>
</OWLClass_EvennessOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_FillerChalk
-->
<owl:Thing rdf:about="#OWLIndividual_FillerChalk">
<rdf:type rdf:resource="#OWLClass_Filler"/>
<rdfs:label xml:lang="bg">Креда</rdfs:label>
<rdfs:label xml:lang="en">Chalk</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_FillerPlaster
-->
<OWLClass_Filler rdf:about="#OWLIndividual_FillerPlaster">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">Гипс</rdfs:label>
<rdfs:label xml:lang="en">Plaster</rdfs:label>
</OWLClass_Filler>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_GumPrimer
-->
<owl:Thing rdf:about="#OWLIndividual_GumPrimer">
<rdf:type rdf:resource="#OWLClass_TypeOfPrimer"/>
<rdfs:label xml:lang="bg">Лепилен грунд</rdfs:label>
<rdfs:label xml:lang="en">Gum Primer</rdfs:label>
</owl:Thing>

```

```

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_OilPrimer
-->
<owl:Thing rdf:about="#OWLIndividual_OilPrimer">
<rdf:type rdf:resource="#OWLClass_TypeOfPrimer"/>
<rdfs:label xml:lang="bg">Маслен грунд</rdfs:label>
<rdfs:label xml:lang="en">Oil Primer</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_PolymericPrimer
-->
<OWLClass_TypeOfPrimer rdf:about="#OWLIndividual_PolymericPrimer">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">Полимерен грунд</rdfs:label>
<rdfs:label xml:lang="en">Polymeric Primer</rdfs:label>
</OWLClass_TypeOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_Resin
-->
<OWLClass_TypeOfLacquering rdf:about="#OWLIndividual_Resin">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">Смола</rdfs:label>
<rdfs:label xml:lang="en">Resin</rdfs:label>
</OWLClass_TypeOfLacquering>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_Semi-OilPrimer
-->
<OWLClass_TypeOfPrimer rdf:about="#OWLIndividual_Semi-OilPrimer">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">Полумаслен грунд</rdfs:label>
<rdfs:label xml:lang="en">Semi-Oil Primer</rdfs:label>
</OWLClass_TypeOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ThicknessOfLacquering01
-->
<owl:Thing rdf:about="#OWLIndividual_ThicknessOfLacquering01">
<rdf:type rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<rdfs:label xml:lang="bg">дебело лаково покритие</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ThicknessOfLacquering02
-->
<owl:Thing rdf:about="#OWLIndividual_ThicknessOfLacquering02">
<rdf:type rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<rdfs:label xml:lang="bg">многократно нанесено лаково покритие</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ThicknessOfLacquering03
-->
<owl:Thing rdf:about="#OWLIndividual_ThicknessOfLacquering03">
<rdf:type rdf:resource="#OWLClass_ThicknessOfLacquering"/>
<rdfs:label xml:lang="bg">средно дебело лаково покритие</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ThicknessOfLacquering04
-->
<OWLClass_ThicknessOfLacquering rdf:about="#OWLIndividual_ThicknessOfLacquering04">

```

```

<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">тънко лаково покритие</rdfs:label>
</OWLClass_ThicknessOfLacquering>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ThicknessOfPrimer01
-->
<OWLClass_ThicknessOfPrimer rdf:about="#OWLIndividual_ThicknessOfPrimer01">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">тънко нанесен грунд</rdfs:label>
</OWLClass_ThicknessOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ThicknessOfPrimer02
-->
<OWLClass_ThicknessOfPrimer rdf:about="#OWLIndividual_ThicknessOfPrimer02">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">дебело нанесен грунд</rdfs:label>
</OWLClass_ThicknessOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ThicknessOfPrimer03
-->
<owl:Thing rdf:about="#OWLIndividual_ThicknessOfPrimer03">
<rdf:type rdf:resource="#OWLClass_ThicknessOfPrimer"/>
<rdfs:label xml:lang="bg">многослойно нанесен грунд</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_ThicknessOfPrimer04
-->
<OWLClass_ThicknessOfPrimer rdf:about="#OWLIndividual_ThicknessOfPrimer04">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">релефен грунд</rdfs:label>
</OWLClass_ThicknessOfPrimer>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_TypeOfGilding01
-->
<owl:Thing rdf:about="#OWLIndividual_TypeOfGilding01">
<rdf:type rdf:resource="#OWLClass_TypeOfGilding"/>
<rdfs:label xml:lang="bg">вапак</rdfs:label>
<rdfs:label xml:lang="bg">листово злато</rdfs:label>
<rdfs:label xml:lang="en">gold-leaf</rdfs:label>
</owl:Thing>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_TypeOfGilding02
-->
<OWLClass_TypeOfGilding rdf:about="#OWLIndividual_TypeOfGilding02">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">течно злато</rdfs:label>
</OWLClass_TypeOfGilding>
<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_TypeOfGilding03
-->
<OWLClass_TypeOfGilding rdf:about="#OWLIndividual_TypeOfGilding03">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">злато на прах</rdfs:label>
</OWLClass_TypeOfGilding>

<!--
http://www.semanticweb.org/ontologies/2011/2/SINUS_TechnologySpecOntology.owl#OWLIndividual_VegetalLacquer
-->
<OWLClass_TypeOfLacquering rdf:about="#OWLIndividual_VegetalLacquer">

```



```
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:label xml:lang="bg">Растителен лак</rdfs:label>
<rdfs:label xml:lang="en">Vegetal Lacquer</rdfs:label>
</OWLClass_TypeOfLacquering>
</rdf:RDF>

<!--
Generated by the OWL API (version 2.2.1.1138) http://owlapi.sourceforge.net
-->
```

Приложение 4: Анотационни граматики

Разпечатка на първата от каскадните граматики: SINUS TSO basic

```
<!--
This Document is created with the Clark System! http://www.bultreebank.org
-->
<Grammars>
<grammar direction="right_to_left">
<name>SINUS TSO basic</name>
<line>
<RE>
(<"позлата"|"позлатата"|"позлати"|"позлатите">, <"са"|"е"|"беше">?, <"без">, <"основа"|"основата"|"основи"|"основите">)
</RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_gilding_has_Base" uri="OWLObjectProperty_gilding_has_Base" rvalue="OWLIndividual_BaseOfGilding03"/></OF>
</RM>
</line>
<line>
<RE>
<"позлата"|"позлатата"|"позлати"|"позлатите">, <"няма">, <"основа">
</RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_gilding_has_Base" uri="OWLObjectProperty_gilding_has_Base" rvalue="OWLIndividual_BaseOfGilding03"/></OF>
</RM>
</line>
<line>
<RE><"болус"|"болуса"|"болусът"|"болуси"|"болусите"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_gilding_has_Base" uri="OWLObjectProperty_gilding_has_Base" rvalue="OWLIndividual_BaseOfGilding02"/></OF>
</RM>
</line>
<line>
<RE><"варак"|"варака"|"варакът"|"вараци"|"вараците"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_gilding_has_Type" uri="OWLObjectProperty_gilding_has_Type" rvalue="OWLIndividual_TypeOfGilding01"/></OF>
</RM>
</line>
<line>
<RE><"гипс"|"гипса"|"гипсът"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Filler" uri="OWLObjectProperty_primer_has_Filler" rvalue="OWLIndividual_FillerPlaster"/></OF>
</RM>
</line>
<line>
<RE>
<"гипсов"|"гипсовия"|"гипсовият"|"гипсова"|"гипсовата"|"гипсово"|"гипсовото"|"гипсови"|"гипсовите">
</RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Filler" uri="OWLObjectProperty_primer_has_Filler" rvalue="OWLIndividual_FillerPlaster"/></OF>
</RM>
</line>
<line>
<RE><"дебело"|"дебелото">, <"лаково">, <"покритие"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_lacquering_has_Thickness" uri="OWLObjectProperty_lacquering_has_Thickness" rvalue="OWLIndividual_ThicknessOfLacquering01"/></OF>
</RM>
</line>
<line>
<RE>

```

```

<"лаково"|"лаковото">, <"покритие">, <"#">*, <"дебело">
</RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_lacquering_has_Thickness"
rvalue="OWLIndividual_ThicknessOfLacquering01"/></OF>
</RM>
</line>
<line>
<RE><"дебело">, <"#">*, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Thickness"
rvalue="OWLIndividual_ThicknessOfPrimer02"/></OF>
</RM>
</line>
<line>
<RE><"грунд"|"грунда"|"грундът">, <"#">*, <"дебело"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Thickness"
rvalue="OWLIndividual_ThicknessOfPrimer02"/></OF>
</RM>
</line>
<line>
<RE>
<"емулсионен"|"емулсионния"|"емулсионният">, <"грунд">
</RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Type" rvalue="OWLIndividual_EmulsionPrimer"/></OF>
</RM>
</line>
<line>
<RE><"емулсия"|"емулсията"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Type" rvalue="OWLIndividual_EmulsionPrimer"/></OF>
</RM>
</line>
<line>
<RE><"злато"|"златото">, <"е">?, <"на">, <"прах"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_gilding_has_Type" rvalue="OWLIndividual_TypeOfGilding03"/></OF>
</RM>
</line>
<line>
<RE><"комбиниран">, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Type" rvalue="OWLIndividual_CombinedPrimer"/></OF>
</RM>
</line>
<line>
<RE><"креда"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Filler" rvalue="OWLIndividual_FillerChalk"/></OF>
</RM>
</line>
<line>
<RE>
<"лаково"|"лаковото">, <"покритие">, <"#">+, <"тънко">
</RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_lacquering_has_Thickness"
rvalue="OWLIndividual_ThicknessOfLacquering04"/></OF>
</RM>
</line>
<line>
<RE><"лепилен">, <"грунд"></RE>
<RM>

```

```

<OF>\w<property domain="OWLCClass_Primer" uri="OWLObjectProperty_primer_has_Type" rvalue="OWLIndividual_GumPrimer"/></OF>
</RM>
</line>
<line>
<RE><"листово"|"листовото">, <"злато"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Gilding" uri="OWLObjectProperty_gilding_has_Type" rvalue="OWLIndividual_TypeOfGilding01"/></OF>
</RM>
</line>
<line>
<RE><"маслен"|"масления"|"масленият">, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Primer" uri="OWLObjectProperty_primer_has_Type" rvalue="OWLIndividual_OilPrimer"/></OF>
</RM>
</line>
<line>
<RE><"масло"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Primer" uri="OWLObjectProperty_primer_has_Type" rvalue="OWLIndividual_OilPrimer"/></OF>
</RM>
</line>
<line>
<RE><"многократно">, <"#">*, <"лаково">, <"покрытие"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Thickness"
rvalue="OWLIndividual_ThicknessOfLacquering02"/></OF>
</RM>
</line>
<line>
<RE><"многослойно">, <"#">*, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Primer" uri="OWLObjectProperty_primer_has_Thickness"
rvalue="OWLIndividual_ThicknessOfPrimer03"/></OF>
</RM>
</line>
<line>
<RE>
<"грунд"|"грунда"|"грундът">, <"#">*, <"многослойно">
</RE>
<RM>
<OF>\w<property domain="OWLCClass_Primer" uri="OWLObjectProperty_primer_has_Thickness"
rvalue="OWLIndividual_ThicknessOfPrimer03"/></OF>
</RM>
</line>
<line>
<RE><"полимент"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Gilding" uri="OWLObjectProperty_gilding_has_Base" rvalue="OWLIndividual_BaseOfGilding01"/></OF>
</RM>
</line>
<line>
<RE><"полимерен">, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Primer" uri="OWLObjectProperty_primer_has_Type" rvalue="OWLIndividual_PolymericPrimer"/></OF>
</RM>
</line>
<line>
<RE><"полумаслен">, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Primer" uri="OWLObjectProperty_primer_has_Type" rvalue="OWLIndividual_Semi-OilPrimer"/></OF>
</RM>
</line>
<line>
<RE><"полумаслен">, <"свързвател"></RE>
<RM>

```

```

<OF>\w<property domain="OWLClass_Primer" uri="OWLObjectProperty_primer_has_Type" rvalue="OWLIndividual_Semi-OilPrimer"/></OF>
</RM>
</line>
<line>
<RE><"растителен">, <"лак"></RE>
<RM>
<OF>\w<property domain="OWLClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Type" rvalue="OWLIndividual_VegetalLacquer"/></OF>
</RM>
</line>
<line>
<RE><"релефен">, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLClass_Primer" uri="OWLObjectProperty_primer_has_Thickness"
rvalue="OWLIndividual_ThicknessOfPrimer04"/></OF>
</RM>
</line>
<line>
<RE><"смола"></RE>
<RM>
<OF>\w<property domain="OWLClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Type" rvalue="OWLIndividual_Resin"/></OF>
</RM>
</line>
<line>
<RE><"средно">, <"дебело">, <"лаково">, <"покритие"></RE>
<RM>
<OF>\w<property domain="OWLClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Thickness"
rvalue="OWLIndividual_ThicknessOfLacquering03"/></OF>
</RM>
</line>
<line>
<RE><"течно"|"течното">, <"злато"></RE>
<RM>
<OF>\w<property domain="OWLClass_Gilding" uri="OWLObjectProperty_gilding_has_Type" rvalue="OWLIndividual_TypeOfGilding02"/></OF>
</RM>
</line>
<line>
<RE><"тънко"|"тънкото">, <"лаково">, <"покритие"></RE>
<RM>
<OF>\w<property domain="OWLClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Thickness"
rvalue="OWLIndividual_ThicknessOfLacquering04"/></OF>
</RM>
</line>
<line>
<RE>
<"лаково"|"лаковото">, <"покритие">, <"#">+, <"тънко">
</RE>
<RM>
<OF>\w<property domain="OWLClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Thickness"
rvalue="OWLIndividual_ThicknessOfLacquering04"/></OF>
</RM>
</line>
<line>
<RE><"тънко">, <"#">*, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLClass_Primer" uri="OWLObjectProperty_primer_has_Thickness"
rvalue="OWLIndividual_ThicknessOfPrimer01"/></OF>
</RM>
</line>
<line>
<RE><"грунд"|"грунда"|"грундът">, <"#">*, <"тънко"></RE>
<RM>
<OF>\w<property domain="OWLClass_Primer" uri="OWLObjectProperty_primer_has_Thickness"
rvalue="OWLIndividual_ThicknessOfPrimer01"/></OF>
</RM>
</line>

```

```
</grammar>
</Grammars>
```

Разпечатка на втората от каскадните граматики: SINUS TSO I-2

```
<!--
This Document is created with the Clark System! http://www.bultreebank.org
-->
<Grammars>
<grammar direction="right_to_left">
<name>SINUS TSO I-2</name>
<line>
<RE><"грубо">, <"#">*, <"грунд"></RE>
<RM>
<OF><w<property domain="OWLClass_Primer" uri="OWLObjectProperty_primer_has_Evenness"
rvalue="OWLIndividual_EvennessOfPrimer04"/></OF>
</RM>
</line>
<line>
<RE><"добре">, <"запазен">, <"грунд"></RE>
<RM>
<OF><w<property domain="OWLClass_Primer" uri="OWLObjectProperty_primer_has_Condition" rvalue="OWLIndividual_ConditionOfPrimer04"/></OF>
</RM>
</line>
<line>
<RE><"добро">, <"състояние">, <"на">, <"лак"></RE>
<RM>
<OF><w<property domain="OWLClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering01"/></OF>
</RM>
</line>
<line>
<RE>
<"добро">, <"състояние">, <"на">, <"грунд"|"грунда">
</RE>
<RM>
<OF><w<property domain="OWLClass_Primer" uri="OWLObjectProperty_primer_has_Condition" rvalue="OWLIndividual_ConditionOfPrimer01"/></OF>
</RM>
</line>
<line>
<RE><"замърсено">, <"лаково">, <"покритие"></RE>
<RM>
<OF><w<property domain="OWLClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering02"/></OF>
</RM>
</line>
<line>
<RE><"лак">, <"#">*, <"замърсен"></RE>
<RM>
<OF><w<property domain="OWLClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering02"/></OF>
</RM>
</line>
<line>
<RE><"здрав">, <"грунд"></RE>
<RM>
<OF><w<property domain="OWLClass_Primer" uri="OWLObjectProperty_primer_has_Condition" rvalue="OWLIndividual_ConditionOfPrimer05"/></OF>
</RM>
</line>
<line>
<RE><"грунд"|"грунда"|"грундът">, <"#">*, <"здрав"></RE>
<RM>
<OF><w<property domain="OWLClass_Primer" uri="OWLObjectProperty_primer_has_Condition" rvalue="OWLIndividual_ConditionOfPrimer05"/></OF>
</RM>
```

```

</line>
<line>
<RE><"лошо">, <"състояние">, <"на">, <"грунда"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Condition" rvalue="OWLIndividual_ConditionOfPrimer02"/></OF>
</RM>
</line>
<line>
<RE>
<"лошо">, <"състояние">, <"на">, <"лаковото">, <"покритие">
</RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering04"/></OF>
</RM>
</line>
<line>
<RE><"напукано">, <"лаково">, <"покритие"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering05"/></OF>
</RM>
</line>
<line>
<RE><"неравномерно">, <"лаково">, <"покритие"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_lacquering_has_Evenness"
rvalue="OWLIndividual_EvennessOfLacquering01"/></OF>
</RM>
</line>
<line>
<RE>
<"лаково"|<"лаковото">, <"покритие">, <"#">*, <"неравномерно">
</RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_lacquering_has_Evenness"
rvalue="OWLIndividual_EvennessOfLacquering01"/></OF>
</RM>
</line>
<line>
<RE><"неравномерно">, <"#">*, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Evenness"
rvalue="OWLIndividual_EvennessOfPrimer02"/></OF>
</RM>
</line>
<line>
<RE><"опадало">, <"лаково">, <"покритие"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering06"/></OF>
</RM>
</line>
<line>
<RE><"повреден">, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Condition" rvalue="OWLIndividual_ConditionOfPrimer03"/></OF>
</RM>
</line>
<line>
<RE><"пожълтяло">, <"лаково">, <"покритие"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering07"/></OF>
</RM>

```

```

</line>
<line>
<RE><"потъмняло">, <"лаково">, <"покрытие"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering08"/></OF>
</RM>
</line>
<line>
<RE><"лак">, <"#">*, <"потъмнял"></RE>
<RM>
<OF>\w<property domain="OWLCClass_" uri="OWLObjectProperty_" rvalue="OWLIndividual_ConditionOfLacquering08"/></OF>
</RM>
</line>
<line>
<RE><"протрито">, <"лаково">, <"покрытие"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering09"/></OF>
</RM>
</line>
<line>
<RE><"лаково">, <"покрытие">, <"#">*, <"протрито"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering09"/></OF>
</RM>
</line>
<line>
<RE>
<"равномерно"|"равномерното">, <"лаково">, <"покрытие">
</RE>
<RM>
<OF>\w<property domain="OWLCClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Evenness"
rvalue="OWLIndividual_EvennessOfLacquering02"/></OF>
</RM>
</line>
<line>
<RE>
<"лаково"|"лаковото">, <"покрытие">, <"#">+, <"равномерно">
</RE>
<RM>
<OF>\w<property domain="OWLCClass_Lacquering" uri="OWLObjectProperty_lacquering_has_Evenness"
rvalue="OWLIndividual_EvennessOfLacquering02"/></OF>
</RM>
</line>
<line>
<RE><"равномерно">, <"#">*, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLCClass_Primer" uri="OWLObjectProperty_primer_has_Evenness"
rvalue="OWLIndividual_EvennessOfPrimer01"/></OF>
</RM>
</line>
<line>
<RE>
<"грунд"|"грунда"|"грундът">, <"#">*, <"равномерно">
</RE>
<RM>
<OF>\w<property domain="OWLCClass_Primer" uri="OWLObjectProperty_primer_has_Evenness"
rvalue="OWLIndividual_EvennessOfPrimer01"/></OF>
</RM>
</line>
<line>
<RE><"разпрашен">, <"лак"></RE>
<RM>

```



```

<OF>\w<property domain="OWLObjectProperty_lacquering_has_Condition"
rvalue="OWLIndividual_ConditionOfLacquering10"/></OF>
</RM>
</line>
<line>
<RE><"шлифован">, <"грунд"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Evenness"
rvalue="OWLIndividual_EvennessOfPrimer03"/></OF>
</RM>
</line>
<line>
<RE><"грунд"|"грунда"|"грундът">, <"#">*, <"шлифован"></RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Evenness"
rvalue="OWLIndividual_EvennessOfPrimer03"/></OF>
</RM>
</line>
<line>
<RE>
<"грунд"|"грунда"|"грундът">, <"#">*, <"шлифовано">
</RE>
<RM>
<OF>\w<property domain="OWLObjectProperty_primer_has_Evenness"
rvalue="OWLIndividual_EvennessOfPrimer03"/></OF>
</RM>
</line>
</grammar>
</Grammars>

```

Разпечатка на третата от каскадните граматки: SINUS TSO Cl:

```

<!--
This Document is created with the Clark System! http://www.bultreebank.org
-->
<Grammars>
<grammar direction="right_to_left">
<name>SINUS TSO Cl</name>
<line>
<RE><"позлата"|"позлатата"|"позлати"|"позлатите"></RE>
<RM><OF>\w<class uri="OWLObjectProperty_gilding"/></OF></RM>
</line>
<line>
<RE><"злато"|"златен"|"златно"|"златни"></RE>
<RM><OF>\w<class uri="OWLObjectProperty_gilding"/></OF></RM>
</line>
<line>
<RE><"лак"|"лака"|"лакът"></RE>
<RM><OF>\w<class uri="OWLObjectProperty_lacquering"/></OF></RM>
</line>
<line>
<RE><"лаково"|"лаковото">, <"покритие"></RE>
<RM><OF>\w<class uri="OWLObjectProperty_lacquering"/></OF></RM>
</line>
<line>
<RE><"грунд"|"грунда"|"грундът"></RE>
<RM><OF>\w<class uri="OWLObjectProperty_primer"/></OF></RM>
</line>
</grammar>
</Grammars>

```

Приложение 5: Резултати от семантичното аотиране

Резултати от използването на анотационните граматики в системата CLaRK за семантично аотиране на специализирани текстове от иконографията с онтологични термини на предметните онтологии представени в Приложение 3.

URI	Срещания на термина в текстовете на български език	Брой срещания на термина		
		Целеви срещания	Правилно разпознати	Неправилно разпознати
Gilding	позлата, позлатата златното покритие позлатен, позлатени златен, златният (компонет) златна, златната (декорация) златно, златни злато златист	120	116	—
Primer	грунд, грунда, грундът грундовото покритие грундиран	121	108	—
TypeOfGilding01	листово злато варак	104	104	—
Laquering	лаково покритие, лаковото покритие лак, лака, лакът	101	100	—
ThicknessOfLacquering04	тънко лаково покритие Лаковото покритие е нанесено тънко. лаковото покритие е нанесено тънко... лаковото покритие ... е нанесено тънко... лаковото покритие е тънко... лакът е положен тънко ...	62	59	—
ThicknessOfPrimer01	...грунд, нанесен тънко. ...грунд, нанесен тънко... ...грунд, нанесен относително тънко... тънкослойно нанесен грунд грундът е ... нанесен тънкослойно. грундът е ...положен тънкослойно. грундът е гипсов, положен тънко. грундът е ... нанесен тънко. грундът е ... нанесен тънко ... грундът е ... нанесен на тънък пласт ... грундовото покритие ... е тънко нанесено. грундовото покритие е ... нанесено тънко ...	74	58	1
FillerPlaster	гипс гипсов грунд, гипсов ... грунд грундът е гипсов ... Грундовото покритие е гипсово. грундовото покритие е гипсово ...	97	56	—

EvennessOfLacquering02	лаковото покритие е нанесено ... равномерно. лаковото покритие ... е нанесено равномерно ... лаковото покритие е ... равномерно. лакът е положен ... равномерно.	58	56	—
EvennessOfPrimer01	...грунд, нанесен ... равномерно. ...грунд, нанесен равномерно. грундът е нанесен равномерно ... грундът е ... нанесен ... равномерно. грундът ... е нанесен ... равномерно. грундовото покритие е...нанесено равномерно ...	38	36	—
BaseOfGilding01	полимент	19	19	—
EvennessOfLacquering01	лаковото покритие е нанесено неравномерно. лаковото покритие е нанесено ... неравномерно. лаковото покритие е нанесено неравномерно ... лаковото покритие е нанесено ... неравномерно ... лаковото покритие е ... неравномерно нанесено ... лаковото покритие е неравномерно нанесено ... лакът е нанесен на неравномерен слой.	17	16	—
ThicknessOfLacquering01	лаковото покритие е нанесено дебело ... лаковото покритие ... е нанесено сравнително дебело ... лаковото покритие е нанесено на дебел слой лаково покритие с много дебел слой.	11	9	—
TypeOfGilding02	течно злато течно трито злато	8	5	—
EmulsionPrimer	емулсията свързващата емулсия	4	4	—
Semi-OilPrimer	полумаслен свързвател	5	4	—
EvennessOfPrimer03	грундът е ... добре шлифован. грундът ... е фино шлифован. грундът е .. нанесен ... добре шлифовано. грундовото покритие е ... добре шлифовано.	7	3	—
ThicknessOfPrimer02	... грунд, нанесен ... дебело. грундовото покритие е ... дебело ...	4	2	—
ThicknessOfPrimer03	... грунд, нанесен неколkokратно ... грундът е ... нанесен многослойно ... грундът ... е нанесен неколkokратно. грундовото покритие е ... нанесено неколkokратно.	10	1	—
ConditionOfPrimer05	грундът е ... сравнително здрав...	3	1	—
ConditionOfLacquering08	... потъмнялото лаково покритие. лакът е ... потъмнял ... лаковото покритие е потъмняло ...	8	—	—
ConditionOfLacquering02	лакът е силно замърсен ... лакът е ... силно замърсен лакът ... силно замърсени ... лаковото покритие ... е замърсено лаковото покритие е силно замърсено ...	6	—	—
ConditionOfPrimer03	грундът е ... разрушен ... наранен грундът. ... повреден грунда. грундът е ... повреден ...	4	—	—

FillerChalk	грундът е кредов ... грундовото покритие е кредово ...	3	–	–
ThicknessOfLacquering02	лаковото покритие е нанесено многократно ... лаковото покритие е нанесено на два слоя. лаковото покритие е нанесено ... на няколко слоя.	3	–	–
ConditionOfPrimer01	грундът ... е в добро състояние. грундът е ... добре запазен.	2	–	–
ConditionOfLacquering03	широки натициания от златисто-кафяв лак лаковото покритие е ... и се установяват натициания.	2	–	–
ConditionOfLacquering09	лаковото покритие ... протрито. лаковото покритие ... протрито ...	2	–	–
EvennessOfPrimer02	грундът е ... нанесен неравномерно ...	1	–	–
OilPrimer	маслен свързвател масло	1	–	11
ConditionOfLacquering10	лакът е разпрашен ...	1	–	–
ConditionOfLacquering06	лакът е ... опаднал.	1	–	–
ConditionOfLacquering01	лаковото покритие е ... в добро състояние.	1	–	–
ConditionOfLacquering04	лаковото покритие е лошо нанесено ...	1	–	–
Общо		899	757	12

Таблица с неразпознати срещания на онтологичните термини

URI	Неразпознати срещания на термина в текстовете от корпуса	Брой неразпознати срещания на термина
FillerPlaster	гипсов грунд, ... грундът е гипсов ... грундовото покритие е гипсово	41
ThicknessOfPrimer01	...грунд, нанесен тънко... тънкослойно нанесен грунд грундът е ... нанесен тънкослойно грундът е ...положен тънкослойно грундът е гипсов, положен тънко грундът е ... нанесен на тънък пласт ... грундовото покритие ... е тънко нанесено грундовото покритие е ... нанесено тънко ...	16
Primer	грундовото покритие грундиран	13
ThicknessOfPrimer03	... грунд, нанесен неколkokратно ... грундът е ... нанесен многослойно ... грундовото покритие е ... нанесено неколkokратно	9
ConditionOfLacquering08	лакът е ... потъмнял ... лаковото покритие е потъмняло ...	8

ConditionOfLacquering02	лакът е силно замърсен ... лакът е ... силно замърсен лакът ... силно замърсени ... лаковото покритие ... е замърсено	6
ConditionOfPrimer03	грундът е ... разрушен грундът е ... повреден наранен грундът ... повреден грунда	4
Gilding	златното покритие златист	4
EvennessOfPrimer03	грундът ... е фино шлифован грундът е .. нанесен ... добре шлифовано грундовото покритие е ... добре шлифовано	4
ThicknessOfLacquering02	лаковото покритие е нанесено многократно ... лаковото покритие е нанесено на два слоя. лаковото покритие е нанесено...на няколко слоя.	3
ThicknessOfLacquering04	лаковото покритие е тънко... лакът е положен тънко ...	3
TypeOfGilding02	течно трито злато	3
FillerChalk	грундът е кредов ... грундовото покритие е кредово ...	3
EvennessOfLacquering02	лакът е положен ... равномерно	2
ConditionOfPrimer01	грундът е ... добре запазен	2
ConditionOfLacquering03	широки натичания от златисто-кафяв лак лаковото покритие е ... и се установяват натичания.	2
ConditionOfLacquering09	лаковото покритие ... протрито. лаковото покритие ... протрито ...	2
EvennessOfPrimer01	грундовото покритие е...нанесено равномерно ...	2
ThicknessOfPrimer02	грундовото покритие е ... дебело ...	2
ConditionOfPrimer05	грундът е ... сравнително здрав...	2
ThicknessOfLacquering01	лаковото покритие е нанесено на дебел слой лаково покритие с много дебел слой	2
Laquering	лаково покритие лаковото покритие	1
EvennessOfLacquering01	лакът е нанесен на неравномерен слой	1
ConditionOfLacquering01	лаковото покритие е ... в добро състояние	1
ConditionOfLacquering04	лаковото покритие е лошо нанесено ...	1
ConditionOfLacquering06	лакът е ... опаднал	1
ConditionOfLacquering10	лакът е разпразен ...	1
OilPrimer	маслен свързвател масло	1
Semi-OilPrimer	полумаслен свързвател	1
EvennessOfPrimer02	грундът е ... нанесен неравномерно ...	1
Общо		142