BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **14**, No 2

Sofia • 2014

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2014-0015

Natural Language Generation and Semantic Technologies

Kamenka Staykova

Institute of Information and Communication Technologies, 1113 Sofia
Email: staykova@iinf.bas.bg

Abstract: *The paper presents a survey of the domain of Natural Language Generation (NLG) with its models, techniques, applications, and investigates how the semantic technologies are drawn into text generation. The idea and facilities of Semantic Web initiative are discussed in connection with the new opportunities offered to the Natural Language Generation.*

Keywords: *Natural Language Generation, semantic technologies, semantic Web.*

1. Introduction

Natural Language Generation (NLG) is classified as a subfield of both areas: Computational Linguistics and Artificial Intelligence. The research in NLG is concentrated on building computer systems or applications producing understandable texts in natural languages. NLG systems usually take as input some non-linguistic information representations, use knowledge about the natural language and knowledge about a particular domain to create documents, reports, explanations, summaries and other types of texts in natural language.

Producing natural language texts is a knowledge-intensive problem. NLG systems need knowledge about the application area of the future text (what should be included in the text, is the information appropriate), knowledge about the particular natural language (vocabulary, grammar, semantics of the language), strategic rhetorical knowledge (how to achieve given communicative goals, how to build different types of texts and so on). Based on scientific literature and discussions in the NLG area[1] Reiter and Dale [47] point out six "classical" tasks connected to the process of Natural Language Generation:

[1] **http://aclweb.org/aclwiki/index.php?title=Natural_Language_Generation_Portal**

- *Content Selection:* It is assumed that NLG systems have at disposal a large volume of input information for the NLG process. This task concerns decisions about the information to be mentioned in the text and the information to be skipped.
- *Document* (*or Discourse*) *Structuring:* This task concerns decisions about how the text should be organized and structured on the whole.
- *Lexicalization*: It is a choice of particular words or phrases which are the most appropriate to convey the specific semantics of information in the given natural language and context.
- *Aggregation*: Making decisions on how to compose the information in groups of sentences.
- *Referring Expression Generation*: Determination of the properties of a given linguistic element, which to be used when the element is mentioned again. The task includes the decisions about using pronouns and anaphors.
- *Linguistic* (*or Surface*) *Realization*: Mapping the already specified descriptions of sentences to sequences of words which present syntactically, morphologically and orthographically correct sentences in the corresponding natural language.

The users need NLG systems which are sophisticated, fast, reliable, stable and intelligent, so that NLG systems have to "pay attention" to the semantics of the texts they produce. Operating with semantics, NLG systems or applications are expected to create meaningful chunks of natural language statements or whole documents. As a knowledge-intensive area, NLG has a natural tight connection with semantics. It is interesting how this connection is understood by NLG researchers during the years of NLG development. Two ways of attitude could be distinguished: researchers who work substantially with the linguistic nature of texts, and researchers who accept texts more like technical carriers of semantics. Outside the NLG area, *semantic technologies* attempting to resolve "the-lack-of-semantics problem" of computer performance have received significant attention. Semantic technologies have been developed in different fields of computer science: Natural Language Processing, Data Mining, Artificial Intelligence, etc. Some WWW-technologies have also their grounds in semantic methodologies. Semantic Web is an initiative introducing special Web standards[2], so that the-lack-of-semantics in human-machine interaction to be reduced. Semantic Web standards for data presented on the Web ease applications of Natural Language Generation and stimulate a new wave of Web based applications of NLG.

The present paper is an attempt to examine a two-fold picture: natural connection between the semantic technologies and NLG and the influence of semantic technological standards on a recent bunch of NLG applications – NLG from Semantic Web data. Section 2 is a short presentation of the NLG area with basic views to the NLG process, the techniques and methodologies applied to NLG. The exposition of NLG process is organized around the view treating natural language texts mainly as a product of human linguistic culture. Section 3 is an attempt to identify the interweaving between NLG and semantic technologies in

[2] **http://www.w3.org/2001/sw/wiki/Main_Page**

general. Section 4 is based on an investigation of NLG applications generating texts from Semantic Web data which are discussed after a short presentation of Semantic Web idea. Section 5 gives a conclusion reporting some of the directions in the future of Natural Language Generation.

2. Natural Language Generation

According to the definition[3] in [1]: "*Natural Language Generation is concerned with constructing computational systems that automatically generate texts of all kinds based on some non-linguistic input (such as a data-base) and some communicative goals. This requires both linguistic knowledge – ranging from morphology to text structure – and computational knowledge, for making the entire thing actually work. Natural Language Generation is therefore of both theoretical and practical interest.*"

Most often the input data for the generation process is described as "a kind of semantic representation". The types of input data vary from internal machine representation of information, through the different formats of data-bases and up to diverse semantic representations experimented over the years (first order logics, Schank's scripts, Sowa's conceptual graphs, RDF-graphs of the Semantic Web and others). If a common abstract representation of NLG input is desired, there is a need for abstract specification of the range of possible input data.

The output of the NLG process is the generated text. Texts could be short or of considerable length, could be structured documents, or a couple of paragraphs, or simply some sentences; they could be strings of characters or the result of more sophisticated text-formatting, hypertext and so on. Within one generalized picture, there is a necessity for specification which allows each particular case to be localized and described.

2.1. Abstract models of the NLG process

Two abstract models of the NLG process as a whole are popular in NLG research society: the *Abstract Generation System* has its roots in Functional Linguistics, while the *Abstract Referential Model* is a result of the project "Reference Architecture for Generation Systems" (RAGS) [14] which is aiming at standards for NLG architecture. The current section continues with short presentation of both models.

Abstract Generation System: The model of NLG process, called *Abstract Generation System*[4] (Fig. 1) is suggested by John Bateman. It is a result of analysis of many NLG systems from a perspective of the range of linguistic phenomena which the different systems deal with. The linguistic notion of *stratification* is used for more precise description of linguistic variations during the NLG process.

[3] **http://www.fb10.uni-bremen.de/anglistik/langpro/webspace/jb/info-pages/nlg-root.htm**

[4] **http://www.fb10.uni-bremen.de/anglistik/langpro/webspace/jb/info-pages/nlg/ATG01/ATG01.html**

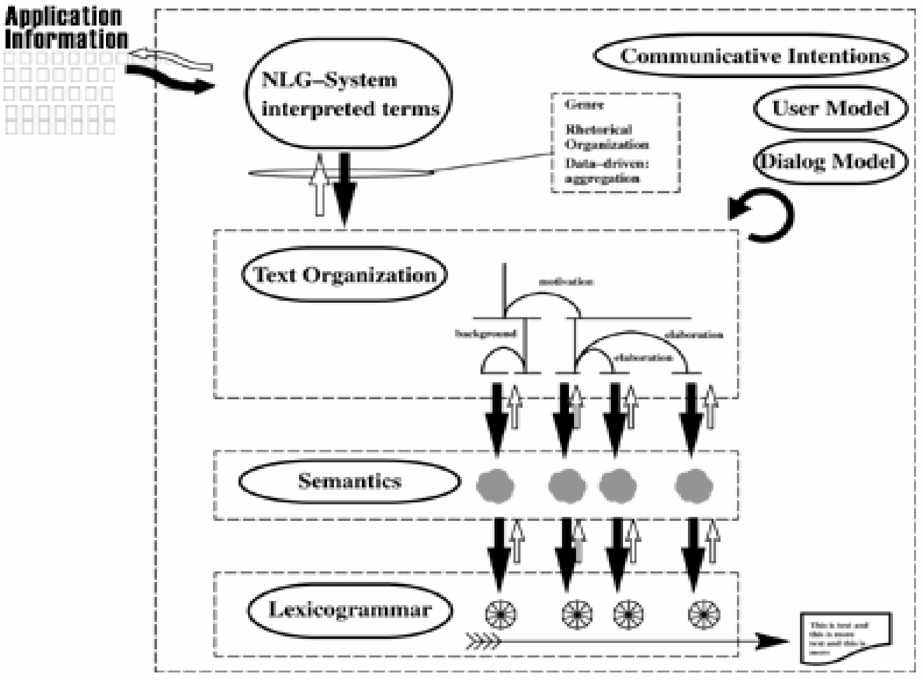

Fig. 1. Abstract Generation System

Stratification concerns the types of information needed on the different abstraction levels of text organization. It underlies three different layers in this abstract model of NLG process. The lower levels of linguistic abstraction, such as phonetics, morphology, syntax form together the layer of Lexico-grammar, the middle abstraction layer is the level of the text meaning or Semantics, and the highest level of linguistic abstraction concerns the text structure and style (Text Organization).

Considerable similarities could be observed between the lexico-grammatical and semantic forms of linguistic organization, which are independent of the application. The highest level of linguistic abstraction, Text Organization, requires a minimally necessary set of text organizational forms on the two lower levels, Lexico-grammar and Text Semantics. Lexico-grammar gives restricting organization of the text by possible grammatical structures used in the language and words, which could occur in the structures. Text Semantics restricts the choice among the possible structures and words only to those, which are appropriate for the particular text meaning.

There are many theoretical models of grammar developed in linguistics and a large part of them has computational implementations. In contrast, more intensive research in the field of semantics is a fact of recent decades. Semantic control of lexico-grammars is particularly useful for NLG, because NLG process requires not only the possible linguistic structures to be fixed, but also the decision points to be accessed, where the appropriate structures for the particular communicative goal are chosen. There are four general groups of linguistic semantic representations, which could give variations in text generation [22]: propositional descriptions, interpersonal variations, textual decisions and discourse semantics consistency.

The *propositional content* of individual sentences is the most natural source of possible choices in each clause. This is the traditional “who did what to whom (when and how)”. Content selection guides the lexico-grammatical choice of basic elements (clause/ noun phrase/ adverbial phrase and so on), as far as it is a choice of grammatical structures used in these categories. Regardless of the propositional content representation, this information can never constrain the lexico-grammatical options of expression sufficiently. A distinguishing feature of NLG is the necessity for considering much more varieties of meaning types than those traditionally addressed in Natural Language Processing.

Interpersonal Choices are concerned with the attitude of the speaker to the content being expressed. There must be assurance in the NLG systems that they do not select inappropriate words in cases when the systems work with many word meanings from large-scale lexicons. The force of the interpersonal choices is important for correct realization of orders, requests, statements, etc., when the particular choices influence substantially the grammatical selections and words selections.

The lexico-grammatical variations of textual meanings present a critical range of diverse choices to be controlled during the NLG process. Almost all existing examples of generated texts have problems related to the fact that there are always places where the human-creator of the text would have made different selections which turn the generated text into a natural and understandable production. The main task here is to isolate and describe the sources of constraints for the lexico-grammatical decisions which make texts “natural”. The complexity of this task depends on the richness of the lexico-grammatical resources proportionally, and the result of the task is a number of textually appropriate lexico-grammatical structures.

Many researchers recognize the task of *discourse semantic consistency* as the main one in NLG. It guarantees that the sequences of constraint specifications produced by the lexico-grammars are created and represented in a mutually consistent way and together constitute the text. For text generation it is not sufficient that the NLG system offers the means for controlling lexico-grammars – such a system must also be able to select from the theoretically available possibilities those that are most appropriate for the particular text. There are attempts in some approaches the texts to be organized on the basis of their intrinsic logic (for example, instructional texts are lists of instructions), but the practice shows that this is often rigid. There is an extensive body of work on the discourse semantic in the NLG area and the basic notions are those of discourse structure defined by the popular Rhetorical Structure Theory (RST) [32]. A picture of a typical RST schema is used to present the Text Organization level in the Fig. 1.

Following the idea of linguistic variations organized by stratification in the model of Abstract Generation System, each one of the “classical” NLG tasks (see Section 1) could be identified in respect to particular level(s) of linguistic abstraction: the task belongs to the level at which the linguistic variation connected to it has been offered or controlled.

Abstract Referential Model is a generalized model of the NLG process, proposed by the project “Reference Architecture for Generation Systems” (RAGS) [14], which aim is to offer standards for NLG architecture. The RAGS model suggests a collection of modules within a NLG system that follow Reiter’s proposal [46] for sequential architecture of NLG system, shown in Fig. 2.

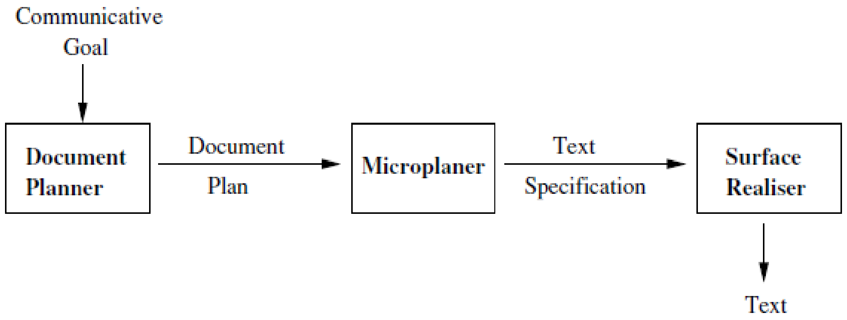

Fig. 2. Sequential architecture of NLG system

According to RAGS model, the NLG process begins with a particular communicative goal. To satisfy the goal, the six NLG tasks mentioned in Section 1 should be successively solved by three basic modules of the sequential NLG architecture. The first module is *Document Planner* (also called Macro-planner or Strategic Planning). It concerns the decisions which information to be expressed by the text and how the chosen content to be organized in a coherent whole. The module produces a Document Plan containing a representation of the whole text and is often used as a specification of the discourse structure of the text. The second module is *Sentence Planner* (known also as *Micro-planner* or *Tactic planning*). It deals with mapping of the document plan to linguistic structures and groups the information into sentences. The module solves Aggregation and Lexicalization tasks. For each sentence of the future text the Micro-planner produces an elaborated lexico-grammatical specification (Text Specification). The third module, called Surface Realiser, transforms the lexico-grammatical specification of each sentence to a sequence of symbols representing the sentence in a form appropriate for people (end-users). The output text is complete after the surface realization.

The above described manner of taking decisions successively within the sequential architecture assumes that there is no mechanism to change a decision taken in one of previous modules if the NLG process has reached a more distant place within the sequence of modules. However, it is clear that the tasks solved by the particular modules of the sequential architecture are not independent from each other. For example, Lexicalization and Generation of Referring Expressions are solved by Micro-planning, but they could require Content Selection, which is a part of a task already solved by the module of Macro-planner. Another characteristic of the sequential architecture of NLG system is that it does not draw on a specification of the linguistic problems faced during the text creation. This is the reason to continue the overview of NLG techniques following the model of Abstract Generation System (Fig. 1), and paying attention to the linguistic nature of the NLG output product – the text as such.

2.2. NLG techniques

Reading a generated text, a NLG user could not determine whether the techniques applied during the generation process are simple or very sophisticated. The complete text could be a result of one ready to print expression, or the text as a whole could be fixed and only some particular elements (names, years, places) to be dynamically installed. The user could have a notion of the real techniques used to build one NLG system only by taking into account the range of texts which could be generated by the system. So, the techniques applied must be evaluated according to the required linguistic variation. The more flexibility of the texts is demanded, the more general linguistic knowledge is required in order to get from the given input the desired range of linguistic productions.

Different techniques could be applied at the different levels of the Abstract Generation System model. The classification presented here follows [1], where the techniques are grouped in four categories: content selection and interpretation techniques, techniques of interaction between tasks and modules, techniques for text organization and techniques of lexico-grammatical realizations. Each group of techniques is shortly discussed in turn.

2.2.1. Content Selection and Content Interpretation Techniques

As it has been already mentioned, the Content Selection task (see Section 1) is to determine which information out of the whole amount of available input data should be presented in the future text. The solution of this complex task depends on many different aspects, which could be clustered in the following three groups:

1) General properties of the intended interaction NLG system – audience: what kind of users is envisaged, what kind of a subject matter is at issue, etc.

2) Situations, in which a particular part of the text must be additionally explained or extended; some information about an already referred objects has to be given, and so on.

3) Variations arise from the expectations concerning the nature of the particular text: whether particular texts conform to generic expectations that allow their readers to recognize them. In linguistics this phenomenon is referred to as ***genre*** or ***text type.***

Representing of the distinct classes of users is generally done by ***user model, which includes information about*** what the user is expected to know, what has been communicated previously in the text, what the user generally likes to know, etc. Representing and using knowledge of the text types is usually done by explicitly providing an account of the text ***macrostructure.*** Particular types of texts have particular structures that should be adhered to during generation and this offers a useful organizing framework for NLG. Generic structures of this kind have been introduced into NLG as ***text schemas*** [36] and have always played an important role in the NLG process.

Text schemas and rule-based templates are the basis of the so called *closed planning paradigm,* term coined in [16]. When NLG approaches adopt a network representation of the input data and exploit the network topology to perform search of the most relevant nodes, they belong to the *open planning paradigm*. Techniques of open planning paradigm (for example [27, 43]) work with content presented in a content graph where the nodes correspond to facts in a knowledge base or to database cells, while edges indicate selection constraints between pairs of nodes. The nodes and edges of the content graph can be assigned weights that modulate the strength of the constraints or quantify the degree of the user interest to certain types of content, as encoded in a user model. Weights support the application of optimization algorithms and graph-based algorithms to solve the content selection problem. The techniques of open planning manipulate large datasets.

Both user modeling and macro text structure definitions provide a partial answer to the question of what information is ***relevant*** for the generated text. Relevance can only be assessed relative to an information need: user modeling defines information needs in terms of the user, while the text structure defines information needs in terms of norms of a language culture.

As it was above discussed, the NLG process requires sequences of semantic specifications to be constructed that provide sufficient information for their expression in grammatical units. The problem of adequate representations of linguistic statements is connected to the much studied *interlingua problem* of Machine Translation [18], which is equally relevant for NLG, and particularly for multilingual NLG. It may be not even theoretically possible “an adequate presentation” to exist for all languages. The experience with *interlingua* suggests that searching for an unique representation is not a successful strategy [44]. It is necessary to adopt techniques that simplify the transition from the available information to the information needed in any text. The techniques which are currently in use include default selections made on the basis of user models and text types.

The easiest technique for solving Content Interpretation task is to create mapping tables from the domain categories to the linguistic resources of the generator [40]. This approach guarantees the modularity of the input data and maintains the possibility of flexible realization when the linguistic representation is sufficiently abstract. The problem with domain-specific encoding of the information remains. It is especially problematic when the task is to interface a domain-independent general NLG system with a particular application domain.

A technique that has found wide application is to reduce the possible mappings between a domain and NLG-terms only to the relation of ***logical subsumption***. This is facilitated when any particular instances of facts that occur in the application can be classified in terms of an ontological model, a hierarchy of general objects and relations. The terms of the hierarchy must behave systematically with respect to their possible linguistic realizations. Such an ontological model is, for example, ***The Upper Model*** [4] which is discussed in more details in Section 3.3.

2.2.2. Techniques of interactions between tasks and modules

Normally NLG systems apply various structures corresponding to the distinct strata of linguistic abstraction. This is not a logical necessity – approaches that adopt a single linguistic account for all representations have also been attempted (for example, [3]). However, there are clear differences in most of the approaches between the techniques applied for lexico-grammar, for semantic information and for text organization, which can be manifested in different representation and processing.

Two techniques of interaction between modules of a NLG system are in use: ***blackboard*** architecture [41] and ***unification*** [20]. The **blackboard** *architecture* leaves the processes to run independently, posting their results and requirements for other processes in a commonly accessible data structure, called *a blackboard*. In case of **unification**, partial structures are combined (unified) opportunistically as they become increasingly specified according to given constraints. The unification is theoretically and formally the most attractive approach, although it is computationally expensive and needs fine control in order to remain usable [20]. Other control strategies that attempt to avoid the problems of strict temporal sequencing are models, involving the feedback or revision from one component to another [49].

2.2.3. Techniques at the level of Text Organization

The task of Text Organization is to design, plan and construct the content into an appropriate text structure. The approaches applied to this task differ largely in flexibility and computational complexity.

The simplest approach to macro-organization of texts is provided by ***templates.*** Template generation sacrifices most of flexibility, but not all applications require flexible solutions. Fixing various aspects of the full generation process corresponds to the basic property of the language itself: when people produce texts they often reuse collections of choices made previously, rather than always make those choices all over again. A relatively flexible type of text templates is represented by McKeown in terms of transition networks [36]. This approach has become one of the most widespread techniques for text organization in NLG despite its clear limitations.

A more flexible text construction is possible by using a theory of text organization, such as Rhetorical Structure Theory (RST) above mentioned [32]. RST provides a general description of the relations among text segments and shows whether or not these relations are grammatically or lexically signaled. Texts described by RST are hierarchically decomposed into a nested set of related text spans: the so called satellites and nucleuses. RST defines 25 relations which may hold between the spans, motivated originally on the detailed linguistic analyses of around 400 texts of varying content and genres. RST definitions bring constraints to bear on the kinds of meanings that the related text spans must carry. There are also constraints on the communicative effect achieved by the combined set of text spans. Constructing a discourse structure in terms of RST relations has proved itself to be useful for supporting selections of linking forms and textual connectives. RST is also used to constrain the relationship for recent anaphors, choices of theme, and selections of focus. RST has been incorporated in a wide range of NLG systems (for example [3, 4, 27]).

An open problem in text planning remains the feature of the texts to be dynamically delivered in time. It is not realistic to rely on models that show the production of a text only in terms of a completed structure, but this is the most straightforward interpretation of RST analyses [33]. To resolve this problem an approach to incremental discourse is developed on the base of Tree Adjoining Grammars (TAGs) [26]. It argues that discourse structures can be constructed dynamically using mechanisms similar to those used syntactically when generating sentences with TAGs.

2.2.4. Techniques at the level of lexico-grammatical realizations

The tasks performing lexico-grammatical realizations map the semantic configurations into well-formed surface strings, presenting one or more natural language sentences. There are many different techniques for the treatment of lexico-grammar with respect to their flexibility. On the one end of the scale the sets of fixed or parameterized structural templates could be placed; on the other end fully generic components stay, which are called grammars and attempt to map any well specified semantic input to a corresponding string or set of strings.

The parameterized templates method essentially pre-compiles the possible texts because the templates are usually written by hand. The main problem with this approach is the fact that the NLG system "knows" very little about the units it is manipulating and even the simplest grammatical variation requires additional altering of the template. Even when the templates are flexible enough to allow iteration, the generated texts remain repetitive and stylistically poor. The method of the template generation is simple, so that it has been pursued in many practically-oriented systems. The more sophisticated variant of this technique mixes fully generated text fragments with pre-stored templates and in this way both approaches have a role within single NLG system [49]. The surface realizations with templates are intrinsically monolingual, have poor reuse properties, and are often stylistically stilted. The templates also become difficult to maintain and manage when their number in a NLG system grows. The differences between full natural language generation and template-based surface generation are discussed in detail in [46].

Grammars introduce plenty of grammatical constructions and with the growth of their range the search space of possible grammatical realizations becomes very large. Hence, the issues concerning the appropriate navigation of this search space become crucial. The different kinds of grammatical presentations can bring different possibilities for traversing the search space.

A structural grammar is primarily organized around phrase structure descriptions. In this area the most well-established strategy is the semantic head-driven generation algorithm [50]. The algorithm generates strings from logical forms for a relatively wide class of grammar formalisms. The technique works by following chains of grammatical rules related by virtue of their syntactic heads sharing common semantics in order to reach applicable lexical entries. When lexical entries are found, the algorithm works back up the structure tree imposing the constraints found in the lexicon. Despite the elegance and formal specification of the algorithm, it is unused outside of theoretical sentence generation work.

An alternative technique, called *message-directed processing* [35] is applied within the MUMBLE lexico-grammatical component for English. The main idea is that the input specifications control deterministic and incremental phrase construction. The input specifications call for particular syntactic tree fragments to be selected. The syntactic trees are expressed in terms of Tree Adjoining Grammars [26]. This input identifies explicitly the particular grammatical constructions that have to be selected in the resulting sentence.

A similar style of input specification is required by a technique which implements the model of language developed by Mel'cuk and colleagues and the so called Meaning-Text Theory (MTT) [37]. The RealPro surface generator [29] takes representation that is a syntactic dependency structure and fills it out. The realization does not draw on the more abstract linguistic strata proposed by Mel'cuk, but as a consequence the RealPro generator is very fast.

A contrasting alternative, called *grammar-directed control* is offered by the generators working with systemic-functional grammars: Penman [30] and its descendent KPML [2]. Systemic grammars organize their search space around possible ***communicative functions*** rather than around grammatical structures. This is particularly efficient for the needs of NLG, because it requires an answer to the question “Why particular structures (syntactic, textual, etc.) have to be used?” This is the natural area of functional linguistics, which has exerted a far greater influence on the larger-scale NLG systems than on the natural language analysis. The main reason is the fact that the structural approaches to syntax are the norm in natural language analysis. Systemic-functional grammars (for example [23]) provide a straightforward interface between the higher levels of text organization and the grammatical component. Traditionally, more attention is paid in such grammars to the non-propositional aspects of meaning, namely to textual and interpersonal meaning.

The generation algorithm of Penman and KPML consists of successive traversals of the feature space, which increase the specificity of tree structures. Although very simple, the algorithm has the advantage that it is quite fast even for large grammars, such as the NIGEL grammar of English [31], the KOMET grammar of German [52], or the AGILE grammars for Russian, Czech and Bulgarian [6]. The algorithm does not perform backtracking, which is a disadvantage. It means that any choice made has to be the right one or else the generation starts again with a modified input or more constraints.

The Functional Unification Formalism (FUF) [19] provides a more powerful traversal of systemic-like feature spaces by employing non-deterministic expansion by unification. A very large coverage grammar of English, called SURGE (Systemic Unification Realization Grammar for English [20]) is written for use with FUF. The SURGE style of input resembles the input style of MUMBLE in that it is mostly syntactic, but, on the other hand, it resembles the Penman/KPML input style in that there is no direct statement of required linguistic structure for the output text. The generation process within FUF consists of taking such an input and unifying it with a similar definition of the entire grammar. The unification approach neutralizes to some extent the division between the grammar-directed and message-directed control since the path followed during unification is sensitive to both sources of constraint.

3. Semantic technologies and NLG process

NLG process includes substantial work with the semantic level of natural texts and with semantic representations. On their side, semantic technologies form their own broader area of research. *Semantic technologies* deal with the problem, called *lack of semantic* which arises when computers perform software programs. Computers "have no notion" about the meaning of the operational elements. The hypothesis which is the base of semantic technologies in computer science is the assumption that machines will demonstrate more intelligent behavior if they are equipped with explicit, formal descriptions introducing a semantic level of the operational elements. According to the popular definition [42] "*semantic technology is a software technology, that allows the meaning of and associations between information to be known and processed at execution time*."

In the lectures published by the Semantic University of Cambridge[5], the term "semantic technologies" represents a fairly diverse family of technologies that help to derive the meaning from information. The main areas of informatics where semantic technologies are used today include:

- Natural Language Processing technologies attempting to process unstructured text content and to extract the names, dates, organizations, events, etc. that are mentioned in the text.
- Data mining technologies employing pattern-matching algorithms to discover trends and correlations within large data sets.
- Artificial Intelligence (especially expert systems) technologies using elaborate reasoning models to answer complex questions automatically. Such systems often include machine-learning algorithms able to improve the system's decision-making capabilities over time.
- Classification technologies using heuristics and rules to tag data with categories to help with searching and analyzing information.
- Semantic search technologies allowing people to locate information by means of concepts instead of keywords or key phrases.

Characteristic for the "traditional" semantic technologies, above listed is that they do not have many common features. The technologies are implemented by means of different programming languages, they produce data in different formats, apply formalisms quite different from each other and rarely work together without a significant effort for their integration.

As far as NLG is a subfield of Natural Language Processing, the technologies used in NLG applications are allied to those of Natural Language Processing area. Semantic technologies typical for Natural Language Processing as knowledge extraction and semantic annotation are often used for creating, upgrading and maintaining resources for NLG. For example, the strategy Ontology-to-text relation is an approach that could be applied to knowledge acquisition and semantic annotation [51] to facilitate a future NLG process.

[5] **http://www.cambridgesemantics.com/semantic-university/**

The Abstract Generation System model of NLG (see Fig. 1) contains a level of linguistic abstraction in text organization called Semantics. The level concerns text semantics and is naturally connected to semantic representations and semantic technologies. Speaking about "classical" NLG tasks listed in Section 1, it is stated in [12] that there are semantically oriented NLG tasks, namely Content Selection, Discourse Structuring and Lexicalization, which depend on the type of semantic input structures used by the particular generation system. It is also argued that concerning semantics in NLG, there is a successful and promising approach: to use a high level ontological model in order to facilitate the mapping from a language independent domain representation onto a NLG-dependent linguistic representation. While such an ontology is language-oriented, and also domain and task-independent, it gives sufficient room for flexible verbalization.

A classical and highly evaluated ontology of this type is the Upper Model [4], defined for use within the Penman text generation system [30]. The classification of domain concepts in terms of the Upper Model allows domain terms to inherit their possibilities for linguistic expression. In principle this does not interfere with domain-internal or application-specific organizations and requires expertise neither in the lexico-grammar nor in the mapping between Upper Model and lexico-grammar. An application needs only to concern itself with the meaning of its own knowledge, and not with details of linguistic form. Responsibility for the correct realization of semantic types remains solely with the NLG system employing an upper ontology. The Upper Model is designed to be portable, reusable and set outside the NLG grammar. It could be thought as interconnection between domain specific information and the linguistic and grammatical kernel of a NLG system. In [4] it is stated that defining relations between the domain specific notions and the notions of the Upper Model significantly simplifies the main task of NLG as a whole.

The Penman Upper Model is a hierarchy of concepts organized originally in three sub-hierarchies: OBJECT, PROCESS and QUALITY. Placing a concept into a particular position of the hierarchy indicates how the concept is expressed in a given natural language. In general, one concept is a member of a class only if it is treated by the natural language in the same way as the other notions of the same class. Further evolution of the Upper Model, addressing simultaneous multilingual NLG resulted in the Generalized Upper Model [5]. This version is more consistent with the Systemic Functional theory exposed in [23].

The Generalized Upper Model is the core multilingual ontology of the multilingual NLG environment KPML [3]. During the years the Generalized Upper Model has been tested for consistency with different languages (English, German, Italian, Dutch, etc.)[6]. A multilingual generation of instructional texts in Russian, Bulgarian and Czech within KPML environment was investigated under AGILE project [28]. The evolution of the Generalized Upper Model is going on with elaborations in conceptual structure [7] or with the ontology translation to OWL – the ontological language of the Semantic Web.

[6] **http://www.fb10.uni-bremen.de/anglistik/langpro/kpml/genbank/generation-bank.html**

A different approach to use a high level ontological model in NLG process is the attempt to reduce the set of upper level concepts and not to introduce a full-blown upper model during the NLG process. Thus, in NLG applications MIAKT [11] and ONTOSUM [9] the surface generator HYLITE+ is applied after mapping the semantic representations to one of four generic and linguistically-motivated relations.

4. Semantic Web and NLG applications

The vision for Semantic Web [8] as a layer over the World Wide Web includes notions of structured and available knowledge, about mechanisms extracting answers from one Global Data Base and also the idea that the structured representation of knowledge is understandable for people, as well as for computers.

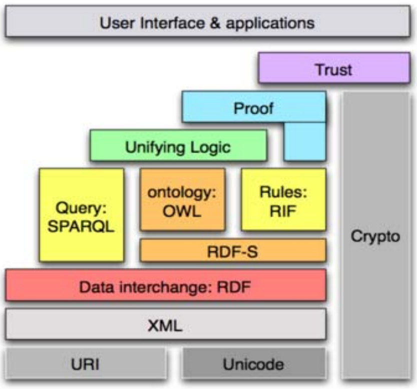

Fig. 3. Semantic Web technologies stack

The Semantic Web technologies (Fig. 3)[7] are formalized by family specific technological standards of the World Wide Web Consortium (W3C)[8]. The Semantic Web technologies are built over the basic Web technologies. The eXtensible Mark-up Language (XML) serves as syntactic base of semantics in the World Wide Web. On the basis of XML the basic data model of the Semantic Web is constructed – Resource Description Framework (RDF). Primary structural unit in this knowledge representation model is called RDF-triple and consists of Subject, Predicate and Object. Initially the goal of the Semantic Web has been Internet pages to be annotated with sets of meta-attributes and categories in order computers to be given the opportunity to interpret the natural language text and put it in some context. However, the annotation is complicated for users with average level of experience with computers. Despite the existing applications that support the annotation process, the approach turns to be unpromising. Its main goal is to ease the access of machines to the knowledge presented by humans via Internet pages, but computers need data bases with information without noise.

[7] Semantic Web Technologies Stack, Tim Berners-Lee, 2006.

[8] **http://www.w3.org/Consortium/**

The vision about the Web of Data is inspired by the fact that a big amount of structured data sets exist all over the world and contain diverse information. Usually a particular data set comprises knowledge about a certain domain, for example books, music, companies and enterprises and so on. The idea is the data sets to be connected with links like Internet pages and to enable machines to access this independent set of structured information without noise. The expected result is a massive accessible knowledge base that forms the basis for a new age applications and services as it is announced by Tim Berners-Lee in 2009.[9]

An important step in this direction is W3C project Linking Open Data[10]. The project follows the main directions of WWW design: simplicity, tolerance, modular design and decentralization. The data sets are organized in such a way that they re-use the existing ontologies, among which the most popular are: Dublin Core, Friend Of A Friend, Geospatial Ontology. Each data set offers access to its own knowledge bases as well as links to elements of other data sets. The number of the open data sets increases rapidly. More than 31.6 billion RDF-triples[11] are accessible now[12] via the project Linked Open Data, which is an immense amount of knowledge. Open data sets could be accessed through Semantic Web browsers or crawlers.

The reason for special interest by NLG researchers to the Semantic Web is the fact that having Linked Open Data initiative, the Semantic Web offers big amounts of RDF data for free, which could be used as input for NLG process. The facilities available by the Semantic Web for NLG are described in [12]:

“The codification of NLG-related knowledge, such as rules and templates has been facilitated by well known APIs and standard query languages (e.g., SPARQL for RDF) that query data which is structured following a standard syntax. Furthermore, heterogeneous knowledge sources, such as domain and domain communication knowledge and conceptual and discourse representations, can be modeled in separate ontologies and integrated using the OWL import mechanism, which provides a limited form of modularization of knowledge.”

Semantic Web data has the form of interconnected datasets or networks of RDF triples. The networks usually include ontologies with their terms/classes, axioms and constraints, and often are RDF transformations of big (relational) data bases.

There are several classifications of the approaches to NLG from Semantic Web data, which overlap to a great extent. In [10] distinction is made between two types of NLG applications from Semantic Web data: applications which help users who are not specialists in knowledge representation to understand and use ontologies, and applications which transform formal knowledge in natural language. A similar distinction is made between Natural Language Interfaces: for ontology engineering and for verbalization of knowledge structured in ontologies following a certain communicative goal. The current section is based on [12], where the

[9] **http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html**

[10] **http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData**

[11] Data is taken from the site of a working group called Linking Open Data Cloud, which monitors and offers catalog for the available data sets **http://datahub.io/group/about/lodcloud**.

[12] July, 2013.

existing paradigms and approaches to the semantically oriented tasks of NLG are discussed in more details.

4.1. Text planning for NLG from Semantic Web data

Analysis of the NLG applications from Semantic Web data shows that four different communicative goals could be distinguished: *say all* which describes all the characteristics of some input object (i.e., class, query, constraint, whole graph) contained in the model; verbalize the *content selected by the user*; verbalize the *most typical facts* found in real or virtual target texts; and verbalize the *most relevant facts* according to the context.

Say all is the communicative goal that subsumes a verbalization request for the entire input graph, the entire set of constraints or axioms in the domain model, all the queries of a term or all the axioms related to a class description in given ontology. In the first two cases, the content selection element is minimal and consists in eliminating redundancies. In the last two cases, the content selection consists mainly in selecting the queries or axioms to verbalize for the given term or class description.

Verbalize the content selected by the user is a communicative goal usually realized by *conceptual authoring*, which is an approach where a supporting natural language interface guides the user through the authoring process. The user selects or authors the concepts from the ontology to formulate a query or to edit the ontology. The access to the underlying knowledge representation is displayed to the user via an interface in terms of natural language statements generated automatically from the knowledge representation. The editing is done through substitutions in specific place-holder points in the rendered text, where the list of possible substitutions presented to the user is delimited by the system according to the underlying knowledge representation. After a confirmation from the user, that the instance is fixed its underlying representation is added to the semantic model and the feedback text is generated again to express the new instance and its (not yet specified) arguments. What makes *conceptual authoring* different from other ontology editors is that there is no need for language interpretation.

Say the most typical is a communicative goal realized by closed planning, which subsumes template-based and rule-based approaches that do not exploit a semantic network representation of the input data. Currently some NLG implementations use Semantic Web representations and technologies. In [13] SPARQL queries are used as content selection rules. In [17] relevant triples are retrieved from multiple data sets and ontologies using a single SPARQL end-point from which queries about museum artifacts can be formulated. In [53] a template-based approach is used for generating biographies of artists to combine text fragments with sentences generated dynamically from the facts. Both text fragments and facts are harvested from the web using information extraction technologies and then are reflected in ontology. In order to avoid repetition, the overlap of information between text fragments is monitored by a blackboard technique.

Say the most relevant is a communicative goal realized by open planning. Open planning approaches are particularly promising in NLG from large and heterogeneous Semantic Web datasets. In [38] and [39] Mellish and Pan address the problem of domain independent content determination (i.e., selecting the content and organizing it into a coherent whole) from an OWL ontology. They propose a new approach called “Natural Language Directed Inference” that selects axioms if their selection can be inferred from axioms already selected. Given an atomic concept of OWL ontology, they perform a depth-first search for more general concepts that can be used to describe the given concept. An approach named Semantic Network Language Generation [16] can be applied to generate texts from a generic semantic network. Starting from a node of interest in the input semantic network, additional nodes are iteratively selected according to a distance function. The resulting set of nodes is then mapped to tree structures according to some patterns which can be linguistically realized as a sentence each. An annotated corpus of texts is used to train the patterns used by the system components.

4.2. Sentence planning for NLG from Semantic Web data

Sentence planning approaches for NLG from Semantic Web data basically use simple rules and templates. Semantic grammars, SPARQL rules, XML templates, aggregation patterns based on entity-sharing between triples or axioms have been used for packaging and aggregating information into sentences. The main approaches to mapping Semantic Web content onto linguistic representation are: the consensus model approach, including the use of Controlled Natural Languages, approaches that annotate content with linguistic knowledge and approaches that use upper models as an intermediate representation between content and linguistic representation.

The Consensus Model is based on the assumption of linguistic expressibility of the ontology, which presumes direct mapping of content labels onto linguistic ones [45]. According to the proponents of the consensus model, the direct mapping loses in fluency of the output, but gains in domain independence and the simplicity of the engineering solution. Some researchers enforce naming conventions to restrict the grammatical category and composition of terms when authoring ontologies, but the efficiency of naming conventions seems difficult to be achieved. A popular strategy to reduce the amount of necessary task-specific knowledge is the reduction of the generated language constructions to a controlled subset, the so called Controlled Natural Language. An unambiguous mapping can be defined from the Semantic Web formal languages to the Controlled Natural Language. The consensus model exploits patterns in ontologies developed essentially in English, so the Controlled Natural Languages used are all subsets of English. Therefore, most approaches to ontology engineering verbalize in English. Perspectives for natural language generation in other languages or multilingual NLG applying the consensus model approach seem not promising.

Annotating content with linguistic knowledge comprises approaches which instead of keeping the lexicon separate from the domain data annotate the domain data with lexical information within the same Semantic Web representation. This is, for instance, the case of NaturalOWL [21], where classes and individuals in the OWL ontology are associated with noun phrases together with the gender of the head nouns and their singular and plural forms. Properties are assigned to microplans for sentence planning and information about the certain verb's inflection and valency. This tight integration of linguistic and domain knowledge, although probably motivated by implementation issues, raises some theoretical concerns and limits the reusability of the linguistic data in different application settings which may require different mappings.

5. Conclusion

There is a big potential for NLG applications from Semantic Web data, which is viewed as new push to NLG development [12]:

- Codification and modeling of NLG-relevant knowledge in Semantic Web standards can provide potential benefits for the interoperability between NLG modules, tasks, and for the reuse of linguistic resources across applications. It is useful to use cross-domain linguistically motivated upper ontologies created before the appearance of Semantic Web standards and already published in OWL language: the Generalized Upper Model [7], the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [34], also, the lexical data-bases which have also been converted to RDF/OWL: WordNet[13] and Framenet[14].
- The machine-oriented Web of Data provides a territory for NLG because of the growing need for easy user access. NLG offers means for presenting semantic data in an organized, coherent and human-accessible way thanks to its abilities to interpret the content even as multi-lingual information.
- NLG could be used for regeneration guided by Semantic Web data to produce a new textual material that may contain both dynamically generated (NLG) text and text fragments obtained from original Web documents.
- Encoding user model in Semantic Web formalisms can help in using the long NLG tradition of working with experimental techniques that adapt the output text to the context by varying the information communicated in the text, its general organization, the language, and so on, according to the preferences of the end user.
- Ingenious approaches for evaluation of NLG systems are encouraged on the base of availability of the vast number of hypertext documents, as for example, the Semantic Web oriented generation system described in [16] that compares its generated texts to corresponding Wikipedia items.

Present day NLG applications could be used for limited domains and for limited tasks, they demonstrate extraction and processing of meaningful content and also communication in a user-friendly way. The basic NLG components of such applications are still very specific and rely on simple text generation templates. If the involved researchers take the risk with more complicated, but also more expressive NLG techniques at the different levels of linguistic abstraction, if the researchers pay more attention to the linguistic nature of created text production there seems to be a fast move to the future of "embodied virtual characters" [15] and more satisfied human users.

[13] **http://wordnet.princeton.edu/**

[14] **https://framenet.icsi.berkeley.edu/**

Acknowledgements: This research is partially supported by project AComIn "Advanced Computing for Innovation", EC FP7 Grant No 316087, Capacity Programme, Research Potential of Convergence Regions (2012-2016).

References

1. Bateman, J. A. Natural Language Generation: An Introduction and Open-Ended Review of the State of the Art, 2002.
2. Bateman, J. A. Enabling Technology for Multilingual Natural Language Generation: The KPML Development Environment. – Natural Language Engineering, Vol. **3**, 1997, No 1, 15-55.
3. Bateman, J. A., E. Teich. Selective Information Presentation in an Integrated Publication System: An Application of Genre-Driven Text Generation. – Information Processing and Management, Vol. **31**, 1995, No 5, 753-767.
4. 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.
5. Bateman, J. A., B. Magnini, G. Fabris. The Generalized Upper Model Knowledge Base: Organization and Use, in Towards Very Large Knowledge Bases. N. Mars, Ed. Amsterdam, IOS Press, 1995, 60-72.
6. Bateman, J., E. Teich, I. Kruijff-Korbayová, G.-J. Kruijff, S. Sharoff, H. Skoumalová. Resources for Multilingual Text Generation in Three Slavic Languages. – In: Proceedings of 2nd International Conference on Language Resources and Evaluation (LREC'2000), European Language Resources Association (ELRA), Athens, Greece, 2000, 1763-1768.
7. Bateman, J. A., J. Hois, R. Ross, T. Tenbrink. A Linguistic Ontology of Space for Natural Language Processing. – Artificial Intelligence, Vol. **174**, 2010, No 14, 1027-1071.
8. Berners-Lee, T., J. Hendler, O. Lassila. The Semantic Web. – Scientific American, 2001, 29-37.
9. Bontcheva, K. Generating Tailored Textual Summaries from Ontologies. In the Semantic Web: Research and Applications. – In: Proceedings of 2nd European Semantic Web Conference (ESWC), Heraklion, Crete, Greece, 2005, 531-545.
10. Bontcheva, D. K., 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, in Conjunction with the Second International Semantic Web Conference (ISWC'03), Florida, USA, 2003, 89-96.
11. Bontcheva, K., Y. Wilks. Automatic Report Generation from Ontologies: The MIAKT Approach. In Natural Language Processing and Information Systems. – In: Proceedings of 8th International Conference on Applications of Natural Language to Information Systems, 2004. Lecture Notes in Computer Science, Vol. **3136**. Springer, 324-335.
12. Bouayad-Agha, N., G. Casamayor, L. Wanner. Natural Language Generation in the Context of the Semantic Web. – Journal the Semantic Web – Interoperability, Usability, Applicability. IOS Press, 2013. **http://www.semantic-web-journal.net/content/natural-language-generation-context-semantic-web**
13. Bouttaz, T., E. Pignotti, C. Mellish, P. Edwards. A Policy-Based Approach to Context Dependent Natural Language Generation. – In: Proceedings of 13th European Workshop on Natural Language Generation (ENLG'2011), Nancy, France, 2011, 151-157.
14. Cahill, L., C. Doran, R. Evans, R. Kibble, C. Mellish, D. Paiva, M. Reape, D. Scott, N. Tipper. Enabling Resource Sharing in Language Generation: An Abstract Reference Architecture. – In: Proceedings of 2nd International Conference on Language Resources and Evaluation, Athens, Greece, 2000, 155-159.

15. Cimiano, P., S. Kopp. Accessing the Web of Data through Embodied Virtual Characters. – Semantic Web 1, Vol. **1-2**, 2010, 83-88.

16. Dai, Y., S. Zhang, J. Chen, T. Chen, W. Zhang. Semantic Network Language Generation Based on a Semantic Networks Serialization Grammar. – World Wide Web, Vol. **13**, No 3, 307-341.

17. Dannells, D., M. Damova, R. Enache, M. Chechev. Multilingual Online Generation from Semantic Web Ontologies. – In: Proceedings of 21st International Conference Companion on World Wide Web, New York, USA, 2012, WWW'12 Companion, ACM, 239-242.

18. Dorr, B. J., E. H. Hovy, L. S. Levin. Machine Translation: Interlingual Methods in Natural Language Processing and Machine Translation. – In: Encyclopedia of Language and Linguistics. 2nd Ed. Amsterdam, Elsevier, Ltd., 2004.

19. Elhadad, M. Types in Functional Unification Grammars. – In: Proceedings of 28th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 1990, 157-164.

20. Elhadad, M., J. Robin. Controlling Content Realization with Functional Unification Grammars. – In: R. Dale, E. H. Hovy, D. Rösner and O. Stock, Eds. Aspects of Automated Natural Language Generation: 6th International Workshop on Natural Language Generation. Berlin/Heidelberg, Springer, 1992, 89-104.

21. Galanis, D., I. Androutsopoulos. Generating Multilingual Descriptions from Linguistically Annotated OWL Ontologies: the Natural OWL System. – In: Proceedings of the 11th European Workshop on Natural Language Generation, ENLG'07, 2007, 143-146.

22. Halliday, M. A. K. Language as Social Semiotic: The Social Interpretation of Language and Meaning. London, Edward Arnold, 1978.

23. Halliday, M. A. K. Introduction to Functional Grammar. Second Ed. London, Edward Arnold, 1994.

24. Henschel, R. Merging the English and the German Upper Model. Darmstadt, Germany, GMD/ Institute fur Integriente Publikation-and Informations Systeme, 1993.

25. Hovy, E. H., J. Lavid, E. Maier, V. Mittal, C. Paris. Employing Knowledge Resources in a New Text Planner Architecture. – In: R. Dale, E. Hovy, D. Rösner and O. Stock, Eds. Aspects of Automated Natural Language Generation, Berlin, Springer, 1992, 57-72.

26. 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. Boston/Dordrecht, Kluwer Academic Publishers, 1987.

27. Krahmer, E., S. Erk, A. Verleg. Graph-Based Generation of Referring Expressions. – Computational Linguistics, Vol. **29**, 2003, No 1, 53-72.

28. Kruijff, G.-J., E. Teich, J. Bateman, I. Kruijff-Korbayová, H. Skoumalová, S. Sharoff, L. Sokolova, T. Hartley, K. Staykova, J. Hana. A Multilingual System for Text Generation in Three Slavic Languages. – In: Proceedings of 18th International Conference on Computational Linguistics (COLING'2000), Saarbrücken, Germany, 2000, 474-480.

29. Lavoie, B., O. Rambow. A Fast and Portable Realizer for Text Generation Systems. – In: Proceedings of 5th Conference on Applied Natural Language Processing, Association for Computational Linguistics, Washington, D. C., 1997, 265-268.

30. Mann, W. C. An Overview of the PENMAN Text Generation System. – In: Proceedings of the National Conference on Artificial Intelligence, AAAI, 1983, 261-265.

31. Mann, W. C., C. M. I. M. Matthiessen. Demonstration of the Nigel Text Generation Computer Program. – In: J. D. Benson and W. S. Greaves, Eds. Systemic Perspectives on Discourse. Vol. **1**. Ablex, Norwood, New Jersey, 1985, 50-83.

32. Mann, W. C., S. A. Thompson. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. – Text, Vol. **8**, No 3, 243-281.

33. Martin, J. R. English Text: Systems and Structure. Benjamins, Amsterdam, 1992.

34. Masolo, C., S. Borgo, A. Gangemi, N. Guarino, A. Oltramari. Wonder Web Deliverable D18, Ontology Library (final), 2009. **http://www.academia.edu/2663572/WonderWeb_Deliverable_D18_Ontology_Library_final_**.

35. McDonald, D. D. Description Directed Control: Its Implications for Natural Language Generation. – Computers and Mathematics, Vol. **9**, 1983, No 1, 111-129.
36. McKeown, K. Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text. Cambridge, England, Cambridge University Press, 1985.
37. Mel'cuk, I. Dependency Syntax: Theory and Practice. Albany, SUNY Press, 1988.
38. Mellish, C., J. Pan. Finding Subsumers for Natural Language Presentation. – In: Proceedings of the International Workshop on Description Logics, The Lake District, England, 2006, 127-134.
39. Mellish, C., J. Pan. Natural Language Directed Inference from Ontologies. – Artificial Intelligence, Vol. **172**, June 2008, No 10, 1285-1315.
40. Meteer, M. W. Expressibility and the Problem of Efficient Text Planning. London, Pinter Publishers, 1992.
41. Nirenburg, S., V. Lesser, E. Nyberg. Controlling a Language Generation Planner. – In: Proceedings of 11th International Joint Conference on Artificial Intelligence, 1989, 1524-1530.
42. Polikoff, I., D. Allemang. Semantic Technology, Top Quadrant Technology Briefing. V1.1. September 2003. **https://lists.oasis-open.org/archives/regrep-semantic/200402/pdf00000.pdf**
43. O' Donnell, M., C. Mellish, J. Oberlander, A. Knott Ilex. An Architecture for a Dynamic Hypertext Generation System. – Natural Language Engineering, Vol. **7**, 2001, 225-250.
44. Polguère, A. Everything Has Not Been Said about Interlinguae: The Case of Multilingual Text Generation Systems. – In: Proceedings of Natural Language Processing Pacific Rim Symposium, Singapore, 1991.
45. 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, 132-136.
46. Reiter, E. NLG vs. Templates. – In: Proceedings of 5th European Workshop on Natural Language Generation, Faculty of Social and Behavioural Sciences, University of Leiden, Leiden, The Netherlands, 1995, 95-105.
47. Reiter, E., R. Dale. Building Natural Language Generation Systems (Studies in Natural Language Processing). Cambridge University Press, 2000.
48. Reiter, E., C. Mellish, J. Levine. Automatic Generation of Technical Documentation. – Applied Artificial Intelligence, Vol. **9**, 1995.
49. Robin, J. Evaluating the Portability of Revision Rules for Incremental Summary Generation. – In: Proceedings of 34th. Annual Meeting of Association for Computational Linguistics (ACL'96)', Santa Cruz, CA, 1996.
50. Shieber, S. M., G. Van Noord, F. C. N. Pereira, R. C. Moore. Semantic Head-Driven Generation. – Computational Linguistics, Vol. **16**, 1990, No 1, 30-42.
51. Staykova, K., G. Agre. Use of Ontology-to-Text-Relation for Creating Semantic Annotation. – In: Proceedings of 13th International Conference on Computer Systems and Technologies (CompSysTech'12), 22-23 June 2012, Ruse, Bulgaria, 64-71.
52. Teich, E. Systemic Functional Grammar in Natural Language Generation: Linguistic Description and Computational Representation. London, Cassell, 1999.
53. Weal, M., H. Alani, S. Kim, P. Lewis, D. Millard, P. Sinclair, D. De Roure, N. Shadbolt. Ontologies as Facilitators for Repurposing Web Documents. – International Journal of Human-Computer Studies, Vol. **65**, June 2007, No 6, 537-562.