Generating Patent Claims From Interactive Input

Svetlana Sheremetyeva[1] lana@crl.nmsu.edu
Sergei Nirenburg[1] sergei@crl.nmsu.edu
Irene Nirenburg[2] nirenburg@cgi.com

[1]Computing Research Laboratory, New Mexico State University, Las Cruces, NM 88003, (505) 646-5466
[2]Carnegie Group, Inc., 5 PPG Place, Pittsburgh, PA 15222

Abstract

Patent claims are the subject of legal protection. They must be formulated according to a set of precise syntactic, lexical and stylistic guidelines. Composing patent claims is a complex task, even for experts. In this paper we report about an implemented system for supporting authoring claims for patents describing apparatuses. The system generates claim texts from the input specified partly by the stored conceptual text schemata and partly by the input from the user. The result of the interactive content acquisition stage is a shallow-level representation which can be considered a draft to be automatically revised into the final text of the claim.

Subject Keywords: interactive, automatic, generation, conceptual schema, template, patent claim

1 Introduction

Patent law guidelines impose rather rigid constraints on the structural composition of the text of the informationally central and legally crucial part of a patent disclosure, the claim. Figure 1 illustrates a rather simple claim text (claims can be over a page long). The claim must consist of a single, albeit possibly very complex, sentence with a well-specified conceptual, syntactic and stylistic/rhetorical structure. For instance, if the invention is an apparatus it must be described in a static state, without reference to its operation.

A cassette for holding excess lengths of light waveguides in a splice area comprising

> a cover part and a pot-shaped bottom part having a bottom disk and a rim extending
>
> > perpendicular to said bottom disk, said cover and bottom parts are superimposed to enclose
> >
> > jointly an area forming a magazine for excess lengths of waveguides, said cover part being rotatable in said bottom part,
>
> two guide slots formed in said cover part, said slots being approximately radially directed,
>
> guide members disposed on said cover part,
>
> a splice holder mounted on said cover part to form a rotatable splice holder.

Figure 1. The text of the example claim generated by the system.

As can be seen from the figure, composing claims can be a difficult task even for a patent expert, let alone an inventor who is typically an engineer and not a technical writer. Note that the difficulty of the task is not constrained to syntax and style. A claim must be composed so as to make patent infringement difficult. We have developed a system which helps an inventor to compose patent claims. The system has an interactive and an automatic components. Knowledge about the invention is elicited from the inventor interactively. Most of text planning and realization is carried out automatically.

Superficially, the architecture of our system conforms to the standard emerged in natural language generation (NLG) (as expressed, for instance in Reiter, 1994) in that it includes the stages of content specification, text planning and surface generation (realization). However, there are some important differences. Unlike the typical content specification modules (e.g., Kukich, 1983; Kittredge et al. 1986), our system relies on an authoring workstation environment equipped with a knowledge elicitation scenario for joint human-computer content specification (see Sheremetyeva et al., *submitted* 1996, for the details of the knowledge elicitation scenario). Lexical selection and some other text planning tasks are interleaved with the process of content specification. The latter results in the production of a "draft" claim. This draft, while not yet an English text, is a list of proposition-level structures ("templates") specifying the proposition head and case role values filled by POS-tagged word strings. The draft is then submitted to an automatic text planner which outputs an hierarchical structure of templates which is ordered according to rhetorical and stylistic requirements. This process resembles revision-oriented generation (Meteer, 1991, Robin, 1994, Gabriel, 1988, Inui et al., 1992). Using the set of distinctions by Robin, our approach is content-preserving (no extra content is added) and performs revisions on a shallow representation. The realization stage linearizes the plan and takes care of the ellipsis, conjoined structures, punctuation and morphological forms. The architecture of the system is illustrated in Figure 2. In what follows we describe each stage of our system in turn and illustrate it with a single example of generating the claim of Figure 1.

2 Content Specification

The input to our system is quite unlike the inputs to other generators. McDonald (1993) lists several kinds of possible inputs -- numerical data, structured objects used by a reasoning system or logical formulae based on lexical predicates (p. 191). A large part of our input is, in fact, in the mind of the inventor. The system simply helps the inventor express this input. In this intent, this system is similar to the DRAFTER system (e.g., Paris et al., 1995).

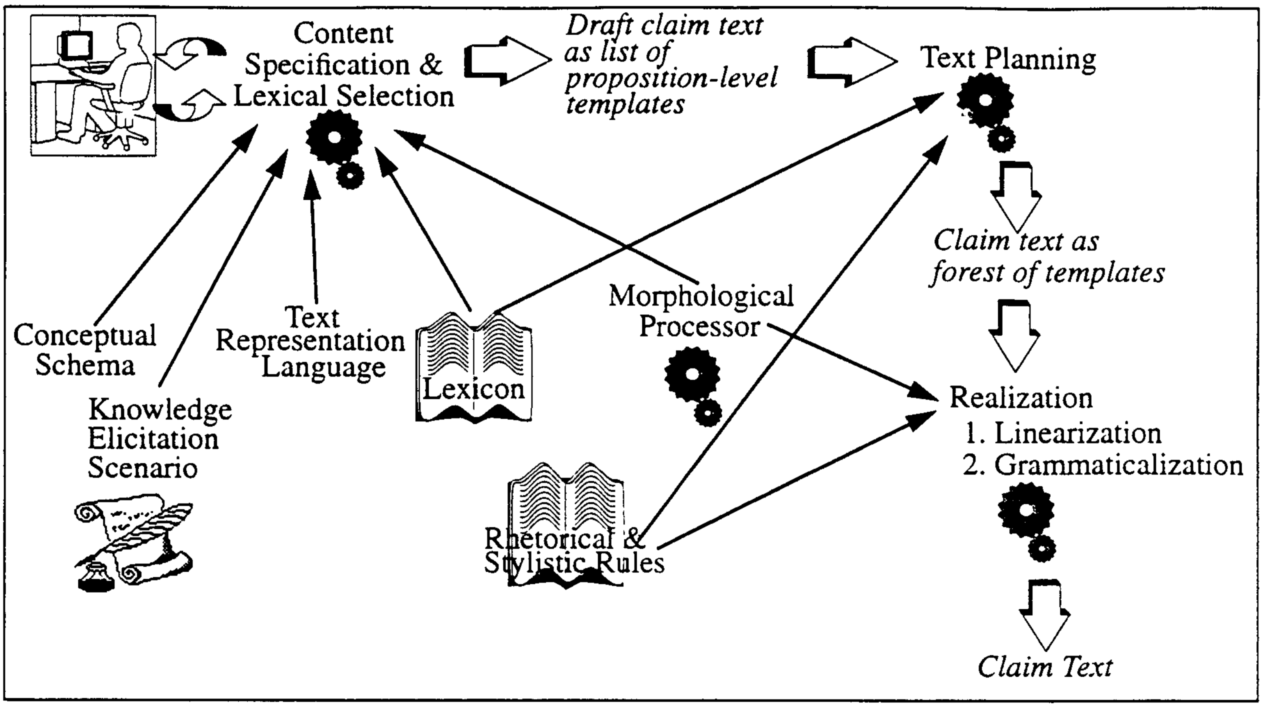

Figure 2. The architecture of the patent claim generator.

Content specification in our system is a process of interactive traversal of a conceptual schema of patents about apparatuses. We built a representation of this schema based on our study of a training corpus of U.S. patents. Patent law prescribes that an invention is described by specifying, in order, a) the title of invention; b) its components (and components of components, as required); c) properties ("attributes") of components (shape, material, dimensions, etc.); and d) relations among the components (spatial, connection, purpose, etc.). In graphical terms, this schema can be represented as a tree, with nodes representing invention components and arcs, the basic meronymic ("has-as-part") relations. Every concrete invention is represented as an instance of the general schema. The schema for our example invention is illustrated in Figure 3.

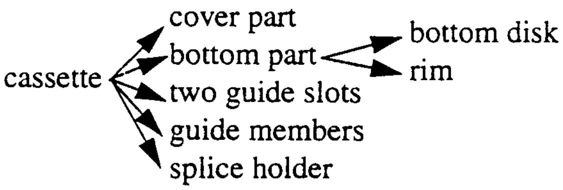

Figure 3. Instance of the conceptual schema tree.

Using common graphical user interface tools (such as dialogue boxes, menus, templates, slide bars etc.), the system guides the user through the paces of describing every essential feature of the invention. Language support is provided through access to vocabularies of suggested verbs and terminological nominal compounds. The inclusion of a human into the process simplifies the task of the system. Notably, it allows us to avoid using a deep knowledge representation language for describing the invention. It is easier for users to manipulate natural and not artificial language. The knowledge elicitation scenario consists in the system requesting the user, in English, to supply information about the invention, its components, their properties and relations among them. The user-supplied information is recorded using a simple text representation language:

```
text          ::= {template}{template}*
template      ::= (label predicate-class
              predicate ({case-role}{case-
              role}*)
case-role     ::= (rank ((label-string)
              value))
```

where *label* is a unique identifier of the template (by convention, marked by the number of its predicate), *predicate-class* is the label of a synonym set of predicate-type words, see below, *predicate* is a string corresponding to one of the predicates from the system lexicon, case roles are *ranked*[1] based on their frequency of cooccurrence with each predicate in the training corpus[2] and *value* is the string which fills a case role. A *label-string* consists of a grammatical class symbol (see Table 1) and a unique ordinal number for each distinct string. Labels are assigned by a morphological analysis module to strings in input templates and nodes in conceptual schema instances. (see Figure 4 and compare it with Figure 3).

Figure 5 illustrates an input template. The labels are used so that we can operate with words and phrases irrespective of the actual inflectional form in which they appear in the user-supplied input or will appear in the final text. All manipulations at the text planning stage are performed on labels. It is at the realization stage that we reintroduce the actual strings and determine their required inflectional forms.

In order to assign label strings to case role values, the values must be analyzed morphologically. As a result of morphological analysis, The interactive input specification (knowledge elicitation) stage provides information about the boundaries of case role values, which makes it possible to use special simplifying morphological rules (thus, for instance, if the filler of the theme case role consists of a single word, it must be a noun). The output of the morphological analysis involves the assignment of the word class and an inflectional form.

Class Label	Class Composition	Examples
P	Predicates: finite forms of verbs and participles in predicative uses	mount, is mounted, mounting
N	All nouns except NE (see below)	device, assembly, camshaft, cage
Adj	Adjectives, participles, ordinal numerals (in attributive uses)	horizontal, each, moving, second
Num	Cardinal numerals	two
DAdv	Discourse-related adverbs	only, at least
Adv	Domain-related adverbs	pivotally, coaxially
Quant	Quantifiers	one of, each of
Inf	Infinitival expressions	to cover
Ger	Gerundive expressions	for moving
Prep	Prepositions	in, on, inside
NE	Nominal events	movement, rotation

Table 1: Lexical Categories in the Sublanguage of Patent Claims

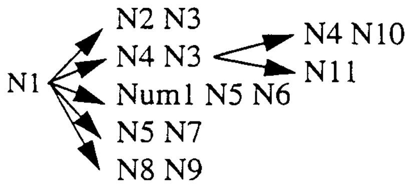

Figure 4. Labeling the conceptual schema tree.

Every question in the knowledge elicitation scenario is connected to one of the 11 synonym sets of English predicates, arranged in the decreasing order of their frequency of occurrence in the training corpus. The appropriate list is presented to the user for selecting the most appropriate realization

[1] Case roles are labeled in the lexicon entry for a particular predicate and the correspondence between the label and the rank for this word is established there, see description of lexicon entry below. The list of case roles for the sublanguage is as follows: *agent, theme, co-theme, place, manner, purpose, means, condition, time.*

[2] A training corpus of over 1,000 U.S. patents was used in this work.

of the content to be conveyed. Once a predicate is selected, the system proceeds requesting information about the values of the case roles of this predicate. The values of these case roles are supplied by the user.[1] This division of labor makes our system immediately practical, because it need not rely on a very large lexicon of terminological terms in the subject area. The internal lexicon of the system must include only a detailed specification of predicative words (mostly, verbs) and some closed-class items, such as prepositions and conjunctions. The following considerations guided our lexicon work.

```
(P6 3 "is mounted"
        (1 ((N8 N9) "the splice holder"))
        (2 ((Prep2 N2 N3) "on the cover part"))
        (4 ((Infl Adj1 N8 N9) "to form a rotatable splice
holder")))
```

Figure 5. A sample template using "mounted."

The patent sublanguage is a union of a legal sublanguage and a sublanguage of the domain of the invention. Our system is devoted to patents about apparatuses. Therefore, its technological sublanguage is that of machines and mechanisms. The sublanguage for such a system has two crucial peculiarities. First, the number of senses for each lexeme is, on average, much smaller than in language as a whole. This is a property of any sublanguage. The second peculiarity seems inherent only to the legal sublanguage. So as best to protect the rights of the inventor, it is desirable to use lexical units whose meanings are *as broad as possible* (see, e.g., Lawson, 1983) without making untrue statements about the invention. Therefore, at the lexical selection stage of the generation of a patent claim the system must be able to choose that member of the synonym set of candidates whose meaning is the broadest. For our system we determined the breadth of meaning of word senses by calculating the relative occurrence frequencies of every word sense the training corpus. Our hypothesis was that this measure is appropriate because the patents were written by expert patent specialists who actually used the words with broadest senses. These frequencies are marked in the system's dictionary only for verbs and take the form of the verb's rank in its semantic class. For example, if the synonym set for lexical selection is as follows: *engage, hold, attach, lock, join, clamp, fasten,* the system will present this list to the user in the descending order of frequencies, with the idea that the user would prefer to select the first applicable word on the list.

Verb entries in the system's lexicon consist of a number of zones as follows:

- **Zone 1** lists all morphological forms of the verb in which it is expected to occur in patent texts. The most frequent form is marked.
- **Zone 2** contains the verb's semantic class label. The classes defined for claims about apparatuses include: meronymy, spatial, connection, change-state, change-location, apply-force, purpose and others.
- **Zone 3** lists the verb's *frequency rank* in the list of all the verbs belonging to its semantic class. It is necessary to motivate the order of verb realization in the text at the generation stage.
- **Zone 4** contains the correspondence between the verb's case frame labels and their ranks.
- **Zone 5** contains a frequency-ordered list of linearized cooccurrences of the verb with a particular subset of case roles. Thus, in Figure 4 the *linearization pattern* (1 * 2 4) (where 1, 2 and 4 are case role ranks and "*" shows the position of the predicate) will match, for example, the following phrase from an actual claim: *(1: the splice holder) is mounted (2: on the cover part) (4: to form a rotatable splice holder).*

A sample lexicon entry is illustrated in Figure 6.

```
MOUNTED[2]
  Zone 1: MOUNTED(*), IS MOUNTED, ARE MOUNTED,
BEING MOUNTED
   Zone 2: spatial
   Zone 3: 1
   Zone 4: 1 agent; 2 place; 3 manner; 4 purpose; 5 means
```

[1] The system includes a number of additional knowledge sources to help the user in the choice of the responses, including access to a world model, or ontology (see Mahesh et al., 1995).

[2] To simplify the processing, it was decided to consider active and passive forms of verbs as separate dictionary entries.

```
Zone 5: (1 * 2), (1 3 * 2), (1 * 2 4), (1 * 2 3), (1 * 3), (1 * 4),
(1 * 2 5)
```

Figure 6. The lexicon entry for "mounted."

The output of the content specification stage (and input into the generation stage) consists of a list of filled templates in which the templates with the title of the invention in their subject slot are marked. A subset of the templates created for our example is given in Figure 7. The set of templates can be considered a draft text of the patent claim. If an English version is generated directly, it will produce a list of individual sentences describing the invention. In fact, our system performs this kind of generation for the purposes of allowing the user to check the draft before it is submitted to the claim generation stage (in this way it is guaranteed that the list of templates contains all the required information). However, we do not use this list of simple sentences in our generation (or revision). This situation is akin to the one described by Meteer (1990) in her Spokesman system design. We use the draft as the input to the process of stylistic and rhetorical text planning and realization.

```
(p1 2 "comprises"
 (1 ((N1)
   "A cassette for holding excess
    lengths of light waveguides in a splice area"))
 (2 ((N2 N3) "a cover part")
   ((N4 N3) "a bottom part")
   ((Num1 N5 N6 ) "two guide slots")
   ((N5 N7) "guide members")
   ((N8 N9) "a splice holder")))

(p2 1 "is pot-shaped"
  (1 ((N4 N3) "the bottom part")))

(p3 3 "are directed"
  (1 ((Num1 N5 N6) "the two guide slots"))
  (3 ((Adv1 Adv2) "approximately radially")))

(p4 7 "is rotatable"
  (1 ((N2 N3) "the cover part"))
  (2 ((Prep1 N4 N3) "in the bottom part")))

(p5 2 "has"
  (1 ((N4 N3) "the bottom part"))
  (2 ((N4 N10) "a bottom disk")
    ((N11)    "and a rim")))

(p6 3 "is mounted"
  (1 ((N8 N9) "the splice holder"))
  (2 ((Prep2 N2 N3) "on the cover part"))
  (4 ((Inf1 Adj1 N8 N9) "to form a rotatable
                           splice   holder")))

(p7 3 "extends"
  (1 ((N11) "the rim"))
  (3 ((Adv3 N4 N10) "perpendicular to the
                      bottom disk")))

(p8 3 "are superimposed"
  (1 ((N2 N3) "the cover part")
    ((N4 N3) "the bottom part"))
  (2 ((Adv4 Inf2 N12) " to jointly enclose
                         an area")))

(p9 11 "forms"
  (1 ((N12) "the area"))
  (2 ((N13) "a magazine")))

(p10 10 "for"
  (1 ((N13) "the magazine"))
  (2 ((Adj2 N14 Prep3 N15) "excess lengths
                             of waveguides")))

(p11 3 "are disposed"
  (1 ((N5 N7) "the guide members"))
  (2 ((Prep2 N2 N3) "on the cover part")))

(p12 3 "are formed"
  (1 ((Num1 N5 N6) "the two guide slots"))
  (2 ((Prep1 N2 N3) "in the cover part")))
```

Figure 7. The templates for the example claim.

3 Claim Text Planning

The planning stage is guided both by constraints on the patent claim sublanguage and the general constraints on style. The former determines the global ordering of the claim text while the latter deals with local text coherence. The global structure of the claim text plan follows the structure of the conceptual schema of a claim, with one important difference. The conceptual schema tree has invention components as nodes, whereas the claim text plan has in its nodes clusters of templates which *describe* the corresponding invention components. The plan structure is obtained by first clustering input templates according to the conceptual schema node to which they belong, building an hierarchical structure (a tree or a forest) for templates in each cluster and, finally, hierarchically connecting all such structures.

3.1 Clustering templates at conceptual schema nodes

This step is not straightforward because a template can be connected to its conceptual schema through more than one case role value string, so that a preference method must be suggested for these cases. For instance, Template P4 (see Figure 7) can be connected to its corresponding node of the conceptual schema, realized as Template P1, either through case role value (N2 N3) or case role value (N4 N3). We define four levels of preference, based on the quality of match between the node label and a string in a template case role (case roles can have a set of strings as their values; such values are called *compound*). (In the description below the template for which linking is attempted is referred to as "current.")

- Quality I match occurs when a) the string in the tree node is *identical* to a case-role string in the current template and b) the rank of the case role in the current template is 1;
- Quality II match occurs when a) the two strings have a nonempty intersection which includes the last element of the string[1] and b) the rank of the case role in the current template is 1 (if the procedure finds more than one match of Quality II, it will select the one with the largest intersection);
- Quality III match occurs when a) the two strings are identical and b) the rank of the case role in the current template is not 1;
- Quality IV match occurs when a) the two strings have a nonempty intersection, as in Quality II match, and b) the rank of the case role in the current template is not 1.

The procedure applies to simple case role values or to components of the compound case role values. The latter can occur both in the conceptual schema tree nodes and in the values of the template slots.

If there is a single candidate, the procedure finds it. If there is more than one candidate, the procedure finds the best one. If no match is possible with conceptual schema node labels, the procedure matches the candidate template case role not with a conceptual schema tree node label but rather with case roles of templates in each cluster, in turn. This activity is based on the expectation that the exposition in a patent claim is one coherent entity, without a possibility of unconnected threads.

3.2 From the Conceptual Schema to a Text Plan

This stage marks the shift from the conceptual to the rhetorical. The conceptual schema tree is transformed into a text plan tree representing the rhetorical structure of the claim text. The nodes in the text plan tree are labeled with the input templates, not invention components. We transform every node from the conceptual schema tree into a subtree whose nodes are templates and whose structure is determined by stylistic and rhetorical considerations typical of text planning. The subtrees are connected into the text plan tree following the links established in the conceptual schema tree, only these links will be between case role values in the templates which are the content of the nodes in the text plan tree.

The cluster-level subtrees of the text plan tree are organized by grouping the templates into what will become sets of siblings at different levels in the text plan tree. Templates which were assigned to a cluster through a match against the same string (either the label of the conceptual schema node or a case role in one of the templates inside the cluster)[2] are grouped into sets of siblings. The hierarchical structure among these sets is established based on the position in the tree of the template against which this match occurred. For example (see Figure 7), the templates P4 and P8 are siblings because they contain a case role value (N2 N3) which represents a node in the conceptual schema (see Figures 4 and 3).

Next, the procedure orders the siblings left to right, in preparation for eventual linearization. The sorting function used for ordering is based on heuristics such as: "the statement which describes more

[1] In our system the last word in a string is practically always the syntactic head of the phrase.

[2] To be precise, in the case of compound case role values the match may have occurred with the same *component* of the case role value.

than one component of the invention should appear as early as possible," "if a content element is described by a single template, it might be amenable to realization as à prenominal modifier; such elements should appear as early as possible," etc. A full set of heuristics see in Sheremetyeva et al., 1996.

After the initial sorting, the procedure checks for occurrence in the sibling templates of the same predicates. If found, they are all moved to form a continuous string at the position of the rightmost occurrence. This is done in expectation of an elliptical realization. The procedure also moves all templates whose predicates are prepositions to the leftmost positions in the string, in order to facilitate their realization without introducing a full clause. The actual text plan tree for our example is illustrated in Figure 8.

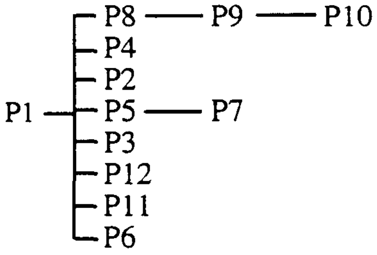

Figure 8. The text plan tree for our example.

The final step of the creation of the text plan tree is to test this tree for complexity and depth. The reasoning behind this is stylistic and syntactic, as claim texts must be both legible and syntactically unambiguous. If either the nesting depth or the number of potentially conjoined structures becomes excessive, the procedure reorders the subtrees of the text plan tree to produce te of acceptable-length and complexity output text chunks. The "counter" of complexity is incremented during the linearization stage and a text chunk is "wrapped up" at the point when the counter reaches a maximum and linearization starts a new text chunk.

4 Realization

4.1 Traversal and linearization of the trees

This stage takes as input a forest of templates and results in the production of a bracketed string of predicate and case role symbols. Two procedures are involved. First, every template is linearized, that is an order of appearance its predicate and case roles is established. Second, the order of templates in the output string is established. The text plan tree is traversed in a top-down, depth-first fashion. Templates can be *concatenated* to the end of the string which resulted from the linearization process of the template processed immediately before the current one or *inserted* into the string corresponding to its parent template, immediately following the case role of the parent template on which the child is linked. In the final string, the boundaries between the templates are retained (the string is bracketed). The result of linearization for our example is illustrated in Figure 9.

[1:N1 **P1** 2:N2N3 2:N4N3
[**P2** 1:N4N3]
[1:N4N3 **P5** 2:N4N10 2:N11
[1:N11 **P7** 3:ADV3N4N10]]
[1:N2N3 1:N4N3 **P8** 2:INF2ADV4N12
[1:N12 **P9** 2:N13
[1:N13 **P10** 2:ADJ2N14PREP3N15]
[1:N2N3 **P4** 2:PREP1N4N3]]] 2:NUM1N5N6
[1:NUM1N5N6 **P12** 2:PREP1N2N3]
[1:NUM1N5N6 **P3** 3:ADV1ADV2 P3] 2:N5N7
[1:N5N7 **P11** 2:PREP2N2N3] 2:N8N9
[1:N8N9 **P6** 2:PREP2N2N3 4:INF1ADJ1N8N9]]

Figure 9. A linearized tree for a claim text. The numbers marked with colons are case roles ranks.

4.2 Grammaticalization

The input to this stage is the bracketed string of English strings. In order to produce cohesive text, it will be necessary to a) select inflectional forms of predicates to facilitate continuity of exposition (e.g., using a participial form instead of a regular finite form to connect two phrases); b) to treat coreference issues by either pronominalization or ellipsis (our system does not use definite descriptions); and c) to realize discourse relations through inserting punctuation. and conjunctions. Correspondingly, the right-hand sides of the realization rules include instructions to carry out the above types of actions.

Realization is carried out left to right segment by segment. A segment is a substring between any two brackets, whether opening or closing. The property determining realization is adjacency, not hierarchical relations; therefore, the orientation of the brackets and their nesting is immaterial. In fact, the first action of this procedure is to substitute demarcation points for any cluster of brackets in the string. Realization of a segment (S_0) depends on its similarities to its preceding segment S_{-1} (occasionally, two preceding segments, S_{-1} and S_{-2}) as well as on the actual realization of the preceding segment(s). The first segment is realized in a standard fashion --- the predicate is always realized as the present participle and no pronominalization or ellipsis occurs.

The left-hand sides of the realization rules contain:

1. contextual constraints in the form of patterns for two (seldom, three) consecutive segments of linearized trees, a *context* of the rule (for instance, the case role values at the end of one segment and at the beginning of the other are identical);
2. lexical constraints in the form of knowledge from dictionary entries for predicates (for instance, that the most frequent form of a predicate is a present participle); and
3. control constraints in the form of knowledge about the system's prior decisions (for instance, that the predicate processed immediately before the current one was realized as a past participle).

The contexts are characterized by a) existence of matching elements in the two segments; b) quality of the match; c) the position in the segments of the matching elements and d) the relative position of partially matched strings. Ten distinct context types were defined for English. A few sample rules are illustrated below.

Rule 1:
Contextual Constraint: the segments S_0 and S_{-1} do not have case roles with identical values
Lexical Constraint: the most frequent form of the current predicate is present simple, passive voice
Control Constraint: none
Action: realize the predicate as a verb in present simple, passive voice

Rule 2:
Contextual Constraint: the first case role value of segment S_0 matches, at Quality I, the last case role value of segment S_{-1}
Lexical Constraint: the most frequent form of the current predicate is past participle
Control Constraint: there is no conjunction *and* between S_{-2} and S_{-1}
Action: realize the predicate as a past participle; remove brackets between the segments and delete the matching case role value in the current segment

5 Conclusion and Future Developments

We have described an implemented generation system with an interactive content specification stage which operates in a conceptually and stylistically constrained environment. Text planning in this system can be considered as content-preserving revision of a shallow "draft" representation produced by content specification. Lexical choice is interactively carried out during content specification, with the system offering the user several kinds of aid in the choice of terminological entities and the lexical realization of relations among them.

A distinguishing feature of this system is its partially interactive character. Borrowing a type distinction from the area of machine translation (MT), we can classify this system as that of human-aided NLG as opposed to fully-automatic NLG.

We intend to a) extend the system into multilingual generation (we have already acquired a lexicon and grammar of Russian for the patent disclosure sublanguage). Another direction of work is developing the interactive authoring support with human-computer interaction in a variety of languages (this could be called "software localization"); b) develop a patent search facility on the basis of the patent disclosure sublanguage and the information retrieval and extraction infrastructure developed in the TIPSTER project (Grishman, 1995); and c) combine the claim text generator with the analysis mc dules of the MikroKosmos project (Onishkevich et al., 1994) to develop a system of automatic translation of patent claims.

References

Gabriel, R. 1988. Deliberate writing. In D.D. McDonald and Bolc L., editors, *Natural Language Generation Systems*. Springer-Verlag, New york, NY.

Grishman, R. 1995. Tipster Phase II Architecture Design Document, version 1.52. TIPSTER Architecture Working Group.

Inui, K., Tokunaga, T., and Tanaka, H. 1992. Text revision: a model and its implementation. In Dale, R., Hovy, E., Roesner, D., and O. Stock, editors, *Aspects of Automated Natural Language generation*, pages 215-230. Springer-Verlag.

Kittredge, R., Polguere, A., and Goldberg, E. 1986. Synthesizing weather forecasts from formatted data. In *Proceedings of the 11th International Conference on Computational Linguistics*, pages 563-565. COLING.

Lawson, V. The Language of Patents. 1983. A Typology of Patents, with Particular Reference to Machine Translation. *In Lebende Sprachen Nr.2*, pages 58-61.

McDonald, D. 1993 Issues in the Choice of a Source for Natural Language generation. In *Computational Linguistics* (19)1, pages 191-197.March 1993.

Kukich, K. 1983. *Knowledge-Based report generation: a knowledge engineering approach to natural language report generation.* Ph.D. thesis, University of Pittsburgh.

Mahesh, K. and Nirenburg, S. 1995. A situated ontology for practical NLP. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, August 1995.

Meteer, M.W. 1990. *The generation gap: the problem of expressibility in text planning*. Ph.D. thesis, University of Massachusetts at Amherst. Also available as BBN technical report No. 7347.

Meteer, M.W. 1991. The implications of revisions for natural Language generation. In Paros, C., Swartout. W., and Mann, C., editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Kluwer Academic Publishers, Boston.

Onyshkevych, B., and Nirenburg S. The Lexicon in the Scheme of KBMT Things. Thecnical Report MCCS-94-227. US Department of Defense and Carnegie Mellon University. Computing Research Laboratory. New Mexico State University

Paris, C., K. Vander Linden, M.Fischer, A.Hartley, L.Pemberton, R.Power and D.Scott. 1995. A Support Tool for Writing Multilingual Instircutions. Proceedings of IJCAI-95, pages 1398 - 1404.

Reiter, E.B. 1994. Has a consensus natural language generation architecture appeared and is it psycholinguistically plausible? In *Proceedings of the 7th International Workshop on Natural Language Generation*, pages 163-170.

Robin, J. 1994. Revision-Based Generation of Natural Language Summaries Providing Historical Background. Corpus-Based Analysis, Design, Implementation and Evaluation. Technical Report CUCS-034-94.

Sheremetyeva, S., and Nirenburg S. 1996. Interactive Knowledge Elicitation in a Patent Expert's Workstation. Submitted to *International Journal of Corpus Linguistics*.

Sheremetyeva, S., and Nirenburg S. 1996. Semi-Automatic Authoring of Patent Claims. Technical Report MCCS-96-290. Computing Research Laboratory. New Mexico State University.