On Creating Metadata With Authoring Tools

Sheremetyeva and **Alexei Pervuchin** and **Vladislav Trotsenko** and **Alexei Tkachev** [1]

Abstract. The paper addresses issues of knowledge markup with authoring tools in which users construct representations of their knowledge. It attempts to contribute to the problem by suggesting a methodology to author new documents in a way that they contain markup directly. The methodology is illustrated with a case study, - knowledge markup with an AutoPat tool. AutoPat is an application for authoring technical documents, such as patent claims that guides users towards expert ways of thinking. It consists of two stages, - a semantic authoring module for interactive elicitation of technical knowledge about invention and a syntactic authoring module that automatically generates a legal (syntactically complex) claim text. The semantic authoring module is implemented as a user-friendly interface that can be used as a stand-alone markup tool. The markup includes morpho-syntactic information augmented by semantic data, such as concept (semantic class) and predicate-argument (case-role) structure. A particular focus is set on an easy-to-use environment for markup automation.

1 INTRODUCTION

A knowledge-capturing problem is a major focuse in the research about content-oriented intelligent applications. A wide range of activities can be found in the area of developing annotated corpora resources, markup languages and tools targeted for experiments in NLP, see, e.g., [1]-[4]. Development of such resources is usually done or at least supervised by highly qualified specialists, requires a lot of time and effort and thus is very expensive. It would be a clear advantage to have a tool based on a methodology, which could provide a much cheaper way of metadata acquisition. The methodology should be universal in the sense that it could be applied to any language and require no skilled labour of professionals.

Constructing general-purpose mark-up guidelines and tools, such as XML, SGML, etc. that can be shared by the community is a popular topic of interest nowadays. It is also recognized that though many increasingly convivial, more widely distributed and hardware-independent applications softwares are currently available for this purpose it is difficult to find a system that matches exactly the end-user requirements [5]. It seems highly problematic (at least nowadays) to be able to develop metadata suitable for all kinds of applications. If, however, the concept of genericity is considered as applied to a family of applications, i.e., applications sharing tasks and domains, one can probably suggest particular approaches to solve the problem. In this paper we attempt just that.

We suggest knowledge markup with authoring tools in which users construct representations of their knowledge. The approach is illustrated on the example of knowledge markup with an AutoPat tool, an application for authoring technical documents, such as patent claims. It consists of two stages, - a semantic authoring module for interactive elicitation of technical knowledge about invention and a syntactic authoring module that automatically generates a legal (syntactically complex) claim text**.** The knowledge elicited from the user is the knowledge about invention to be patented, the tool further annotates user's raw input with syntactico-semantic information to create an internal representation of the claim content that serves as an input to the generator. We thus get semantically annotated knowledge as a by-product of an AutoPat main user (who is a patent officer or inventor) authoring session. The tool can also be used directly for knowledge markup. The specificity of our approach is that semantic annotation of knowledge contained in a claim text is created before such text comes into being rather than after as in many other applications. In fact, our annotation does not necessarily require a text as a starting point for markup. In what follows we first discuss linguistic aspects of semantic annotation in specific IT applications, we then present the context of the AutoPat application and overview its procedures and components relevant for knowledge markup, - an analyzer and user interface.

[1] LanA Consulting, Madvigs Alle, 9, 2 tv, DK-1829 Copenhagen, Denmark

2 MARKUP LANGUAGE

We use Autopat internal knowledge representation language to annotate the technical knowledge of the patent domain. The problem of semantic knowledge representation (and annotation) is directly connected with the decisions on the depth of semantic descriptions. Deeper descriptions promise better results but require a greater acquisition effort. Practical considerations make it reasonable to follow the demand-side approach to knowledge acquisition which places a premium on what must be done (vs. what can be done) to put together a useful working application [6]. We based our approach on the ideas of lexicalism that shifts all linguistic descriptions to lexicon [7]. This move, in turn, has led to an increased interest in argument structure - the representation and characterization of argument-taking properties of predicates [8], [9] that proved to be useful to encode a wide rande of information. In our system technical knowledge about invention described by a claim text (see Figure 2) is represented as a set of filled predicate templates in the form:

```
text::={ template){template}*
template::={predicate-class predicate ((case-role)(case-role}*)
case-role::= (rank status value)
value::= {word tag}*
```

where *predicate-class* is a label of an ontological concept, *predicate* is a string corresponding to one of predicates from the system lexicon, *case-roles* are *ranked* according to the frequency of their co-ocurreence together with each predicate in the training corpus, *status* is the semantic status of a case-role, such as agent, theme, place, instrument, etc., *value* is a string which fills a case-role. *Tag* is a label, which conveys morphological information (such as POS, number and inflection type) and semantic information, a concept, defining word membership in a certain semantic class (such as object, process, substance, etc.). For example, the tag Nf means that a word is a noun in singular (N), means a process (f), and does not end in *–ing*. This tag will be assigned, for example, to such words as *activation* or *alignment.* At present we use 23 tags that are combinations of 1 to 4 features out of a set of 19 semantic, morphological and syntactic features for 14 parts of speech. For example, the feature structure of noun tags is as follows:

```
Tag [ POS[Noun [object   [plural, singular]
               process [-ing, other[plural, singular]]
               substance [plural, singular]
                other      [plural, singular]]]]]
```

The number of semantic classes (concepts) and case-roles is domain based and is rather small but can be easily augmented. In general, our annotation can be classified, following the definition of the Network Working Group (www.landfield.com) as "out-of-band" which convey the textual content by metadata or hyperstructure of some sort.

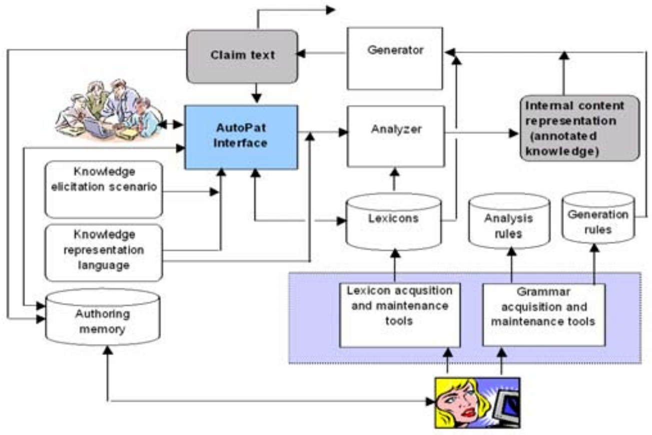
3 THE AUTOPAT TOOL

3.1 Overview

Claims are parts of patents that contain crucial information about the invention and are the subject of legal protection. They must be formulated according to a set of precise rules and so as to make patent infringement difficult. Composing a patent claim that meets all legal requirements to its structure is a complex task, even for experts (see Figure 2 for a sample claim text). AutoPat is designed to reduce composition effort, time and costs. It can also be used for training patent attorneys**.**

AutoPat is an NLP application[1] that consists of an interactive semantic authoring module for technical knowledge elicitation with a sophisticated but easy-to-use interface at the user end, analysis module and fully automatic text generation module.

.



Claim text
Generator
AutoPat Interface
Analyzer
Internal content representation (annotated knowledge)
Knowledge elicitation scenario
Knowledge representation language
Lexicons
Analysis rules
Generation rules
Authoring memory
Lexicon acqusition and maintenance tools
Grammar acquisition and maintenance tools

Figure 1. The overall architecture of AutoPat

The architecture of AutoPat with integrated development environment is given in Figure 1. Superficially, the architecture of our system conforms to the standard emerged in natural language generation, in that it includes the stages of content specification, text planning and surface generation (realization), as expressed, for instance in [10]. However, there are some important differences. Unlike the typical content specification modules, our system relies on an authoring workstation environment equipped with scenario for joint human-computer content specification. The latter starts with the user supplying *natural language phrases* into the system in the process of computer interview and (after passing through the analyzer) results in production of a "draft" claim. This draft is a list of proposition-level structures ("templates") specifying the proposition head with its semantic class membership, a set of its case-roles, and case-role values filled by tagged word strings. The tags mark both POS and semantic class membership of the words (see Section MARKUP LANGUAGE). The draft is then submitted to automatic generator, which produces a claim text in a legally accepted format (see Figure 2).

The draft of a nascent claim is in fact annotated knowledge about an invention or (which is the same) a syntactically and semantically annotated claim text, the legal format of which is produced at a later stage. The AutoPat knowledge base is corpus-based and draws heavily on the sublanguage. It contains AutoPat inherient knowledge and authoring memory (cf. "translation memory"). The inherent knowledge includes *a shallow lexicon* of lexical units simply listed with their class membership that is a morpho-semantic classification of words and phrases (this lexicon is used for content support in claim composition and for morphological analysis of the input), and *a deep (information-rich) lexicon* of predicates (heads of predicative phrases describing essential features of an invention). This lexicon is the main part of the AutoPat static knowledge and covers the lexical, semantic and syntactic knowledge and is the basis of knowledge representation . It is used both to provide content support for technical knowledge elicitation and for generation heuristics. The user can customize these lexicons. Authoring memory is created by the user in the course of authoring sessions.

3.2 Elicitation/Annotation Module

Analyzer. In AutoPat knowledge annotation takes place in the course of knowledge elicitation domain-dependent mixed-initiative interview and is, in fact, semantico-syntactic analysis of the user's input. Our analyzer differs from many other application analyzers in that the morphological analysis module switches on after rather then before syntactic analysis. The early application of syntactic analysis allows the morphological analyzer to avoid overgeneration and produce unambiguous results.

The knowledge elicitation scenario consists of the system requesting the user, in English, to supply information about the invention by offering the user a choice for lexical selection of what amounts to heads (predicates) of phrases in the nascent text, a significant amount of knowledge about predicate subcategorization and argument properties is available to the system from the lexicon. The user is presented with a predicate template (see Figure 3) based on knowledge about the case-roles (semantic arguments) of the selected dictionary item and fills appropriate slots – "What", "Where", "How", and so forth. Filling case-roles in a predicate template during knowledge elicitation procedure is, in fact, an interactive semantico-syntactic analysis and knowledge annotation. The system marks the boundaries of the fillers (syntax) and their case-role status (semantics) to be used later for morphological disambiguation.

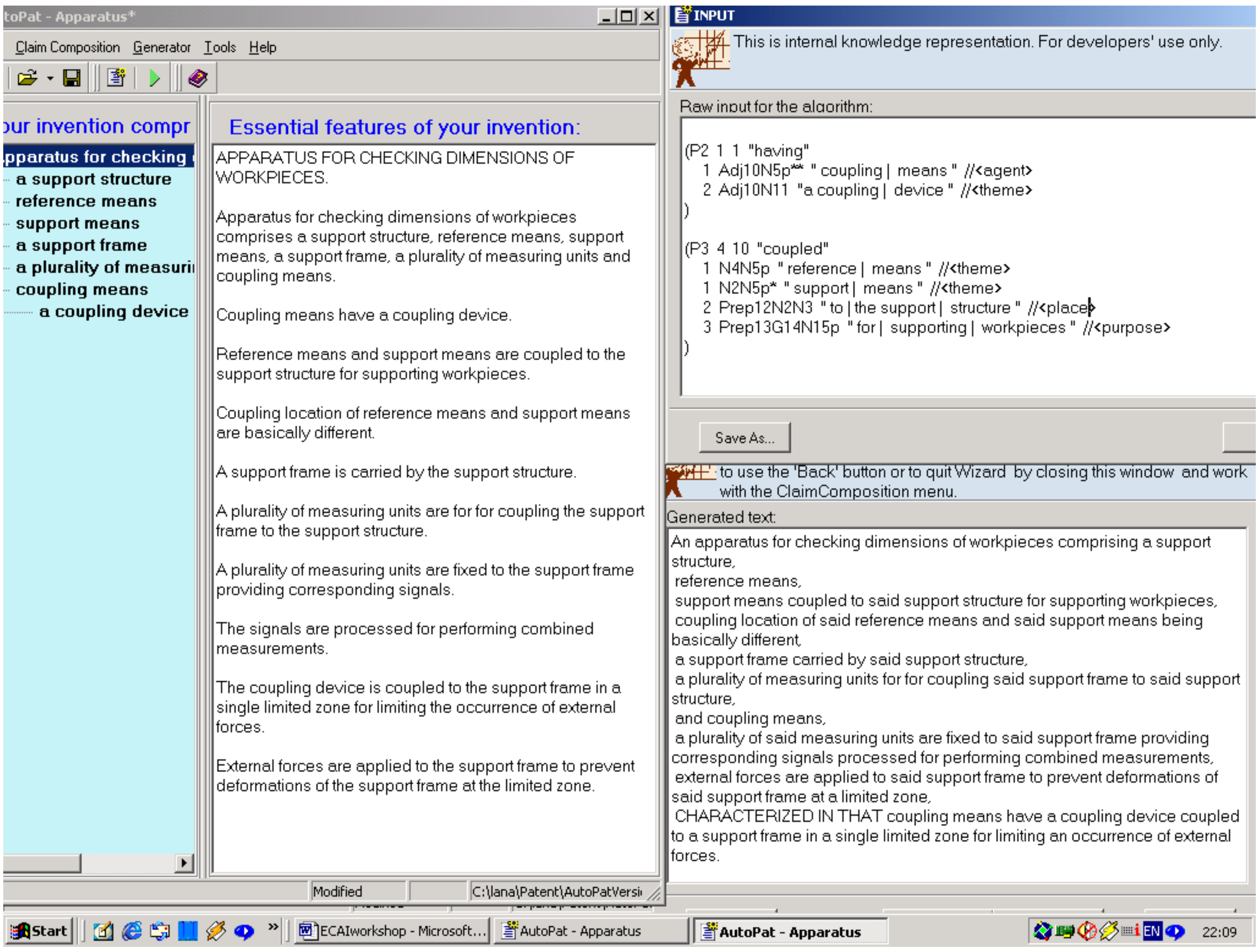
Morphological analysis is applied to the case-role fillers and consists in tagging proper which assigns all possible tags to words. To help resolve POS ambiguities all lexical units in our analyzer are put into classes specified by tags marking not only POS but also such morphological features as number and inflection type. Some of the tags include semantic features (see Section 2)**.** After the set of tags is assigned to a word a disambiguation procedure

[1] AutoPat is a product developed from the prototype described in [11]. It is a 32-bit Windows application developed to run in a number of operating environments: Windows 95/98/2000/NT. It covers claims about apparatuses.

switches on. Discarding faulty readings of ambiguously tagged words are done in two passes. During the first pass the analyzer applies disambiguation rules of a more general character relying only on the knowledge in the morphological lexicon and a 5-word window context with the tag in question in the middle. If there are still ambiguities pending after this first step of disambiguation the second step of disambiguation is applied. It uses knowledge about case-role boundaries and their semantic status delivered by the interactive module of the analyzer. The output of the knowledge elicitation procedure is an out-of-band annotated claim (shallow content representation) that when stored in a knowledge base provides a resource where syntactic information is augmented by semantic knowledge.

3.3 Interface

AutoPat features a sophisticated but extremely user-friendly interface that can be adapted to different user profiles (beginners and experts) and has two modes: a) Wizard that guides a user through an ordered step-by step procedure of describing the invention and b) Professional that allows for more speed and flexibility when authoring a claim, - the user may freely navigate among the stages of claim composition authoring them in any order.



ɔur invention compr

pparatus for checking

- a support structure
- reference means
- support means
- a support frame
- a plurality of measuri
- coupling means
  - a coupling device

Essential features of your invention:

APPARATUS FOR CHECKING DIMENSIONS OF WORKPIECES.

Apparatus for checking dimensions of workpieces comprises a support structure, reference means, support means, a support frame, a plurality of measuring units and coupling means.

Coupling means have a coupling device.

Reference means and support means are coupled to the support structure for supporting workpieces.

Coupling location of reference means and support means are basically different.

A support frame is carried by the support structure.

A plurality of measuring units are for for coupling the support frame to the support structure.

A plurality of measuring units are fixed to the support frame providing corresponding signals.

The signals are processed for performing combined measurements.

The coupling device is coupled to the support frame in a single limited zone for limiting the occurrence of external forces.

External forces are applied to the support frame to prevent deformations of the support frame at the limited zone.

Modified | C:\Ilana\Patent\AutoPatVersi

This is internal knowledge representation. For developers' use only.

Raw input for the algorithm:

```
(P2  1  1  "having"
   1  Adj10N5p**  " coupling |  means "  //<agent>
   2  Adj10N11  "a coupling |  device "  //<theme>
)

(P3  4  10  "coupled"
   1  N4N5p  " reference |  means "  //<theme>
   1  N2N5p*  " support |  means "  //<theme>
   2  Prep12N2N3  " to | the support |  structure "  //<place>
   3  Prep13G14N15p  " for |  supporting |  workpieces "  //<purpose>
)
```

to use the 'Back' button or to quit Wizard by closing this window and work with the ClaimComposition menu.

Generated text:

An apparatus for checking dimensions of workpieces comprising a support structure,
reference means,
support means coupled to said support structure for supporting workpieces,
coupling location of said reference means and said support means being basically different,
a support frame carried by said support structure,
a plurality of measuring units for for coupling said support frame to said support structure,
and coupling means,
a plurality of said measuring units are fixed to said support frame providing corresponding signals processed for performing combined measurements,
external forces are applied to said support frame to prevent deformations of said support frame at a limited zone,
CHARACTERIZED IN THAT coupling means have a coupling device coupled to a support frame in a single limited zone for limiting an occurrence of external forces.

Figure 2. A screen shot of the Autopat user interface at the final stage of knowledge elicitation. It displays a fragment of annotated knowledge, (top right) that is also represented in two unannotated texts, - a) a set of simple sentences (middle) corresponding to quantums of technical knowledge elicited from the user and generated for user content control, and b) a single sentence (bottom right) meeting legal requirements to the claim format. The left part of the screen shows the element tree of the invention.

As was mentioned above the knowledge elicitation scenario consists of the system requesting the user, in English, to supply information about the invention. Using common graphical interface tools (mouse support, dialogue boxes, menus, templates and slide bars) the interface draws the user through a step-by step procedure of describing essential features of the invention. If the content appears incorrect, the user can undo the latest quantum or acquisition and do it again correctly. He can also easily edit the knowledge elicited at any of the earlier stages using the second, Professional mode of interface.

The interface has two main components, - the *background window* were the results of elicitation procedure stay displayed through the whole session and a set of *pop-up windows* corresponding to elicitations steps (see Figures 2 and 3). The two modes of the interface share the background window while the sets of pop-up windows are mainly different. All pop-up windows in both modes can be moved freely around the screen to allow the user to see any part of the background window at any time.

Background Window (Figures 2 and 3). The left pane of this window is headed "Your invention comprises" and displays a graphical representation of the hierarchy of all main elements and sub-elements after the user supplies the knowledge about them into the system. The names of the elements at its nodes can be transferred to any of pop-up windows by simply clicking on them. The right pane is headed "Essential features of your invention". It displays the title of the invention and every essential feature of the invention in the form of a simple sentence (see Figure 2) that is generated every time the user supplies a quantum of technical knowledge. Visualization of the results of the elicitation procedure in the form of simple sentences is only done to make it possible and convenient for the user to control the results of authoring/annotation session. The simple sentences correspond to statements in the system's internal knowledge representation language that are created following the knowledge elicitation procedures (see Figure 2). At the stage of eliciting knowledge about relations of invention elements a new section headed "Your terminology" appears in the bottom of the left pane. Form now on all phrases used in relation descriptions stay displayed and "clickable" there for further reuse.

A brief description of interface windows and functionalities is given below.

Title. Helps the user to select the most appropriate title for the invention. This window contains a title template. The slots of this template contain menus of words and phrases for optional inclusion in the title.

Main Elements and *Sub-Elements.* Prompt the user to describe the element hierarchy of the prototype of the invention. These windows display a template of menus similar to that in the *Title* window.

Element with Novel Characteristics. Makes the user specify the element whose novel properties (that, according to Patent Law, can only be its shape or material) it is necessary to include in the claim.

Shape/Material. Prompts the user to describe novel shapes of materials of the elements specified in the previous window background window).

Relations. Within the procedure the user selects two or more objects in the element tree then specify the relation between them. The initial setup in this window involves two menus, one listing names of relation types (semantic classes) and another listing words (predicates) that can describe these relations. One can start by first selecting a relation type and then, after a semantic class is selected the second menu displays predicates which belong to this class for further selection. By checking a corresponding radio button it is possible to start directly with selecting a predicate among all the predicates included in the AutoPat knowledge base and listed in the predicate menu. In case the selected predicate belongs to more than one semantic classes, these classes appear in the semantic class menu and the user is asked to select one of them to specify the meaning of the predicate. Selecting a predicate constitutes lexical selection, whereupon the system determines the roles played by the highlighted elements .

Relation Specification. Presents the user with a predicate (sentence) template based on knowledge about the case-roles (semantic arguments) of the semantic class underlying the selected dictionary item. The user fills appropriate slots – "What", "Where", "How", and so forth (Figure 3). To make this easier apart from clickable nodes in the element tree and in phrases in "Your terminology" section every template slot has a pop-up menu of auxiliary phrases from the underlying predicate dictionary entry.

Co-reference. Highlights coreference candidates and ask to mark any elements that are coreferential among them. The coreference candidates are searched by morphosyntactic analyzer and are noun lexemes regardless of their grammatical form

Main Claim Format. Presents a "checkable" menu of all generated sentences-features. The user can either check the novel features of the invention to thus have a final claim text containing generic and difference parts with the "characterized in that" expression between them (as in Figure 1), which is a must according to the European Patent Office, or skip this stage. In the latter case the final claim text will be generated without generic and difference parts in the format accepted by the US Patent Office. The underlying knowledge will be accordingly marked as referring to novel or prototype features of the invention. This is relevant if annotated knowledge base is searched in a new patent novelty examination.

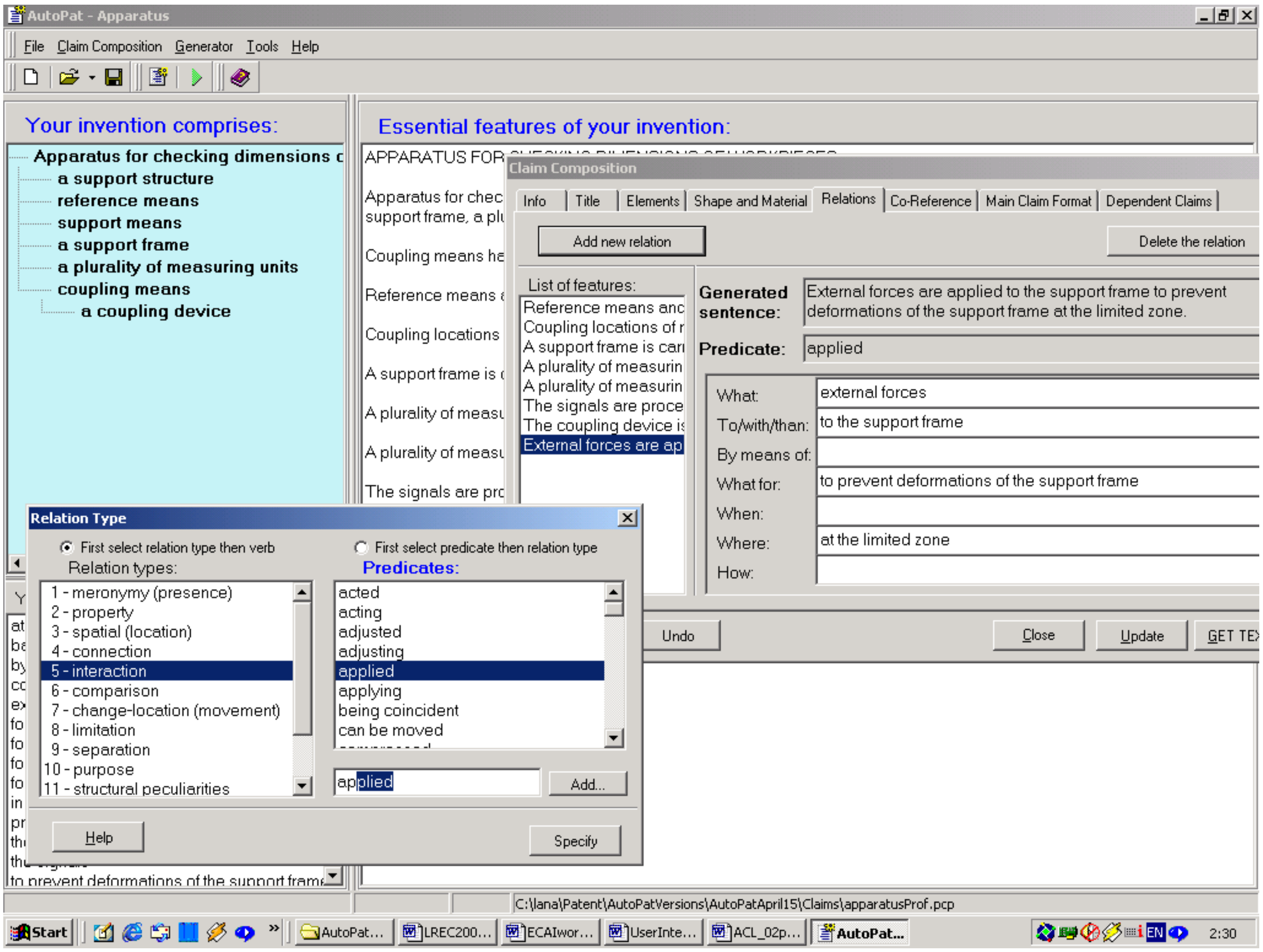
Main Claim Text. Presents the output of the Auto generator, - the claim text in legally acceptable format, as shown in Figure1. If necessary the user may edit the text right in this interface window. This window is accessible through the "GET TEXT" button.

Dependent Claims and *Dependent Claim Text.* The former appears only if called by the user who wants to compose a dependent claim, it elicits information upon which of other claims the current one depends. The latter displays generated text of the dependent claim as it should appear in a patent document.

All user-computer communication is done in a natural language. It provides content, composition and terminology consistency maintenance support through choices of standing and pull-down menus. These menus supply access to words and phrases required in a claim. Though the user is encouraged to use the AutoPat controlled language given in the menus s/he has always a choice to type in active text areas of interface windows. If a word is in a menu it will be automatically completed right after the first characters are typed. The interface looks for spelling errors and, more important, provides for lexicons customization so as not to require any linguistic skills. In case a word cannot be found in the knowledge of the system the user will be asked to add it through an easy-to-use pop-up entry box. The interface automates tedious tasks such as typing and propagating changes through document and, what is more important, it has effective means to control knowledge supplied to the system. The user can check the content elicited so far in an output window where the immediate results of each quantum of acquisition are displayed in the form of simple sentences (see Figure 2).

4 CONCLUSIONS

We suggest knowledge markup with authoring tools in which users construct representations of their knowledge and illustrated our approach on the example of knowledge markup with an AutoPat tool, an application for authoring patent claims. Annotated knowledge is created as a by-product of an AutoPat main user (a patent officer or inventor) authoring session. The tool can also be used directly for knowledge markup. In general, our annotation can be classified, as "out-of-band" which convey the textual content by metadata mainly stored in the predicate lexicon of the system. Annotated knowledge is stored as a formal shallow content representation. Annotation marks both patent domain technical knowledge and linguistic data about patent sublanguage on morpho-syntactic and semantic levels. The annotated knowledge can be stored in the domain knowledge base without its unannoteted text form, the latter can always be generated by Autopat. A knowledge base of such annotations is beneficial for other applications of AutoPat family such as domain-tuned machine translation, information retrieval, summarization, etc. It can also be used for constructing linguistic metadata. We have also descried an interactive procedure which allows for simultenuous elicitation tecnical knowledge from the user and its annotation by the system. Both the metodology of annotation and components of AutoPat relevant to annotation (analyzer and user interface) can be portable to other domains, languages and applications.


AutoPat - Apparatus

File Claim Composition Generator Tools Help

Your invention comprises:

- Apparatus for checking dimensions c
  - a support structure
  - reference means
  - support means
  - a support frame
  - a plurality of measuring units
  - coupling means
    - a coupling device

Essential features of your invention:

APPARATUS FOR

Apparatus for chec
support frame, a plu

Coupling means ha

Reference means a

Coupling locations

A support frame is c

A plurality of measu

A plurality of measu

The signals are prc

Claim Composition

Info | Title | Elements | Shape and Material | Relations | Co-Reference | Main Claim Format | Dependent Claims

Add new relation — Delete the relation

List of features:

Reference means anc
Coupling locations of r
A support frame is carı
A plurality of measurin
A plurality of measurin
The signals are proce
The coupling device is
External forces are ap

Generated sentence: External forces are applied to the support frame to prevent deformations of the support frame at the limited zone.

Predicate: applied

What:	external forces
To/with/than:	to the support frame
By means of:	
What for:	to prevent deformations of the support frame
When:	
Where:	at the limited zone
How:	

Undo — Close — Update — GET TEX

Relation Type

(●) First select relation type then verb ( ) First select predicate then relation type

Relation types:

1 - meronymy (presence)
2 - property
3 - spatial (location)
4 - connection
5 - interaction
6 - comparison
7 - change-location (movement)
8 - limitation
9 - separation
10 - purpose
11 - structural peculiarities

Predicates:

acted
acting
adjusted
adjusting
applied
applying
being coincident
can be moved

applied — Add...

Help — Specify

to prevent deformations of the support frame

C:\lana\Patent\AutoPatVersions\AutoPatApril15\Claims\apparatusProf.pcp

Start — AutoPat... — LREC200... — ECAIwor... — UserInte... — ACL_02p... — AutoPat... — EN — 2:30

Figure 3.A screen shot of the user interface at the stage of describing relations between the elements of an invention. The window on the right displays filling case-roles in a predicate template as a result of choices made during the "Relation Type" step of knowledge elicitation scenario shown on the left.

REFERENCES

[1] T. McEnery, A. Wilson, F.Sunchez-Leon, A.Nieto-Serrano, 'Multininguual resources in European Languages, Contributions of the CRATER Project', *Literary and Linguistic Computing 12/4 (1997).*

[2] T.Erjavec, N.Ide. 'The MULTEXT-East Corpus', *Proceedings of the MT Summit VII, 1999, Singapore.*

[3] C.M. Sperberg-McQueen, L. Burnard, 'Guidelines for Electronic Text Encoding and Interchange, *Chicago and Oxford. (1994).*

[4] P.Anick, J.Pustejovsky, 'An application of lexical semantics to knowledge acquisition of corpora', *COLING-90.Proceedings of the 13th International Conference on Computational Linguistics. Helsinki. 2 (1990)*

[5] P.Degoulet, F.C.Jean, U.Engelmann, HP.Meinzer, R.Baud, B.Sadbald, O.Wigertz, R.Le Meur, CA.Jagermann, *The component-based architecture of the HELIOS medical software .(1994)*

[6] S. Nirenburg, V. Raskin, 'Ten Choices for Lexical Semantics. New Mexico State University', *Computing Research Laboratory Technical Report (1996 MCCS 96-304.(1996).*

[7] B. Y. Ooi, Computer Corpus Lexicography. *Edinburgh University Press (1996).*

[8] Ch.J. Fillmore. 'Subjects, speakers and roles'. *Synthese. 21/3/4. (1970).*

[9] B. Levin. 'English Verb Classes and Alternations'. *University of Chicago Press, Chicago . (1993).*

[10] Reiter, E.B. 'Has a consensus natural language generation architecture appeared and is it psycholinguistically plausible?' *In Proceedings of the 7th International Workshop on Natural Language Generation. (1994)*

[11] S.Sheremetyeva, S.Nirenburg.. 'Interactive Knowledge Elicitation in a Patent Expert's Workstation', *IEEE Computer. Vol.7.(1996).*